Accurate Markov Boundary Discovery for Causal Feature Selection

Xingyu Wu, Bingbing Jiang, Kui Yu, Chunyan Miao, and Huanhuan Chen, *Senior Member, IEEE*

***Abstract*—Causal feature selection has achieved much attention in recent years, which discovers a Markov boundary (MB) of the class attribute. The MB of the class attribute implies local causal relations between the class attribute and the features, thus leading to more interpretable and robust prediction models than the features selected by the traditional feature selection algorithms. Many causal feature selection methods have been proposed, and almost all of them employ conditional independence (CI) tests to identify MBs. However, many datasets from real-world applications may suffer from incorrect CI tests due to noise or small-sized samples, resulting in lower MB discovery accuracy for these existing algorithms. To tackle this issue, in this article, we first introduce a new concept of PCMasking to explain a type of incorrect CI tests in the MB discovery, then propose a cross-check and complement MB discovery (CCMB) algorithm to repair this type of incorrect CI tests for accurate MB discovery. To improve the efficiency of CCMB, we further design a pipeline machine-based CCMB (PM-CCMB) algorithm. Using benchmark Bayesian network datasets, the experiments demonstrate that both CCMB and PM-CCMB achieve significant improvements on the MB discovery accuracy compared with the existing methods, and PM-CCMB further improves the computational efficiency. The empirical study in the real-world datasets validates the effectiveness of CCMB and PM-CCMB against the state-of-the-art causal and traditional feature selection algorithms.**

***Index Terms*—Bayesian network (BN), causal feature selection, Markov boundary (MB), PCMasking.**

I. Introduction

CAUSAL feature selection is to identify a Markov boundary (MB) of a class attribute for building accurate prediction models. The MB was first defined and discussed by Pearl in a Bayesian network (BN) [1]. Under the faithfulness assumption (refer to Definition 5 in Section III), the MB of a variable in a BN consists of its parents, children, and spouses (the other parents of the children of the variable), and given the MB of a variable, all other variables will be independent of this variable [1]. Thus, the MB provides a complete picture of the local causal structure around a variable [2]. In addition, in theory, the MB of the class attribute is the optimal solution of the feature selection problem [3], [4].

Recent years have witnessed the proliferation of the causal feature selection methods since they can select features not only predictive but also causal informative. Existing algorithms can be roughly divided into two different types. The first type is to directly discover the MB of a target variable of interest, which sacrifices the MB discovery accuracy to improve the computational efficiency. The early causal feature selection methods, such as the growth and shrink algorithm (GS) [5] and the Koller–Sahami (KS) [6] algorithms, belong to the first type. The later methods, incremental association MB (IAMB) and its variants [7], improve the GS by reordering the variables each time the MB set changes. However, IAMB and its variants require large amount of data to guarantee the accuracy. To solve this problem, the second type of algorithms is proposed by employing a divide-and-conquer strategy to improve the MB discovery accuracy. Min–max MB (MMMB) [8] is the first divide-and-conquer-based method, later algorithms, such as HITON-MB [9] and parents–children-based MB (PCMB) [10], are improved on MMMB, which first find the parents and children (PC) of a target, and then identify the spouse (SP) of the target. MBOR [11] combines the two types of methods, which employs the first type of method to obtain an initial MB first and then finds more MB variables with the divide-and-conquer strategy.

To improve the MB discovery accuracy, the existing causal feature selection algorithms mainly focus on how to remove the false positives during the MB search process, but rarely consider the true positives discarded due to incorrect conditional independence (CI) tests, leading to low true positive discovery accuracy, especially in the presence of insufficient or noise data samples. For example in Fig. 1, by conducting the experiment on a benchmark Alarm BN with different sample sizes, we found that the recall of the existing causal feature selection algorithms is much smaller than their precision. Specifically, given 1000 samples, the average precision of these algorithms is 0.92, but the average recall is only 0.81 (a more detailed comparison on accuracy can be found in Fig. 5 of Section VI). The low recall value makes the existing causal

Manuscript received May 5, 2019; revised August 17, 2019; accepted August 26, 2019. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1000905, in part by the Beijing Municipal Science and Technology Commission under Grant Z171100000117017, and in part by the National Natural Science Foundation of China under Grant 61876206, Grant 91846111, and Grant 91746209. This article was recommended by Associate Editor X. Wang. *(Corresponding author: Huanhuan Chen.)*

X. Wu, B. Jiang, and H. Chen are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: xingyuwu@mail.ustc.edu.cn; jiangbb@mail.ustc.edu.cn; hchen@ustc.edu.cn).

K. Yu is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China (e-mail: yukui@hfut.edu.cn).

C. Miao is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: ascymiao@ntu.edu.sg).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2019.2940509

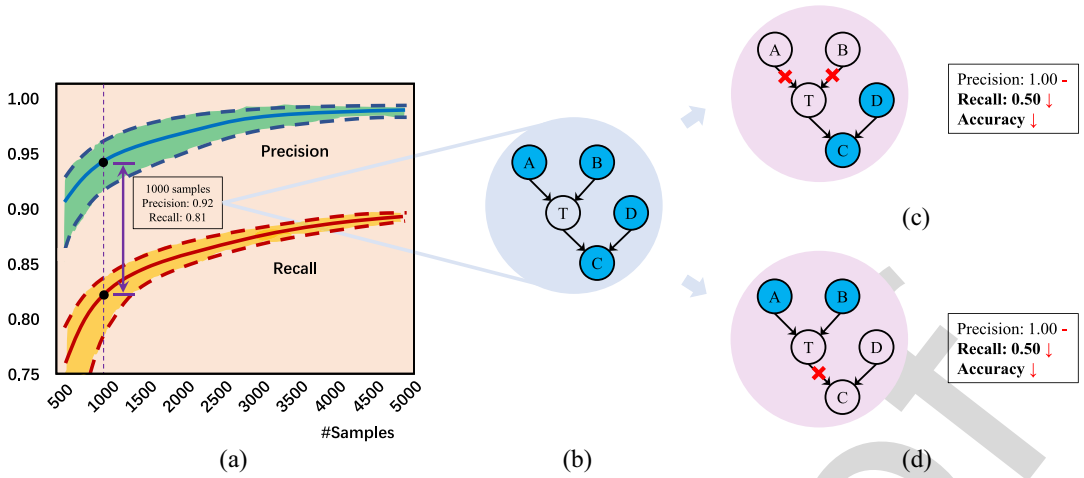


Fig. 1. Existing causal feature selection algorithms have higher precision but lower recall. A series of experiments was conducted on Alarm BN with different sample scale and averaging over several state-of-the-art algorithms (IAMB, PCMB, MBOR, and STMB). (a) Average precision and recall variation curves with respect to the number of samples. (b)–(d) Take a DAG as an example to illustrate the higher precision and lower recall of MB discovery when there exists the PCMasking phenomenon. (b) Correct result and highlight the true MB of T in blue, that is, parents A and B, child C, and spouse D. (c) and (d) The two wrong cases where the PCMasking phenomenon occurs. The wrong MBs of T in (c) and (d) are highlighted in blue, and the red "X" symbol denotes the independence relations between T and its parents or children.

feature selection algorithms ineffective for accurate prediction in practical applications.

Motivated by the above issue, we experimented on different benchmark BN datasets using existing causal feature selection algorithms to explore why some true positives are discarded. We found a type of incorrect CI tests that makes PC of a target mask each other, and we call it PCMasking. Specifically, PCMasking denotes a target and its children may be independent conditioning on its parents and vice-versa. For example, we use a directed acyclic graph (DAG) in Fig. 1(b) to illustrate the PCMasking phenomenon. Assuming that the target T and its parents $\{A, B\}$ are independent conditioning on its child C [as Fig. 1(c)] and, meanwhile, T and C are independent conditioning on $\{A, B\}$ [as Fig. 1(d)]. The incorrect tests will make $\{A, B\}$ or C not be added to the final output of the existing algorithms. In Fig. 1(b), the MB of T should be $\{A, B, C, D\}$. However, if we apply existing algorithms to the dataset and the type of incorrect CI tests occurs, the output of the algorithms is $\{A, B\}$ [as highlighted in Fig. 1(c)] or $\{C, D\}$ [as highlighted in Fig. 1(d)] due to the PCMasking phenomenon [highlighted with red "x" in Fig. 1(c) and (d)].

Besides empirical analysis, we also analyze the mechanism of the PCMasking phenomenon from the perspective of information theory, and further find that PCMasking phenomenon breaks the symmetry between the PC variables, which leads to some true PC variables discarded by the existing algorithms. Therefore, if there is no extra strategy to deal with the PCMasking phenomenon, then the MB discovery methods will ignore some direct causes (parents) and direct effects (children), resulting in the performance degradation. Furthermore, since the algorithms complete the PC discovery first and then search for the spouses based on the PC set, incomplete PC set will cause cascading errors in the SP discovery. However, the PCMasking phenomenon has attracted little attention, resulting in ineffectiveness of the existing causal feature selection algorithms on real-world datasets. To tackle this issue, the main contributions of this article are summarized as follows.

1) We formally present a new concept, called PCMasking, to describe a type of incorrect CI tests in the MB discovery process, and theoretically analyze the mechanism behind this type of CI tests.
2) Based on the theoretical analyses, we propose the cross-check and complement MB discovery (CCMB) algorithm to tackle the PCMasking phenomenon and improve the true-positive discovery accuracy. Moreover, to improve the computational efficiency of CCMB, we further design a PM-CCMB algorithm, which is more accurate and efficient compared with all other MB discovery algorithms.
3) We conduct a series of experiments on synthetic and real-world datasets, to validate the effectiveness and efficiency of the proposed algorithms against the state-of-the-art causal and traditional feature selection algorithms.

The remainder of this article is organized as follows. Section II reviews the related work and Section III introduces the basic notations and definitions. In Section IV, we propose the new concept of PCMasking and explain its mechanism. The proposed CCMB and PM-CCMB algorithms are described in Sections V. The experimental results and analyses are presented in Section VI. Finally, Section VII concludes this article and describes possible future work direction.

II. Related Work

As a dimensionality reduction technique, feature selection algorithms try to find a lower-dimensional representation of data via removing irrelevant features without altering the original feature space [12]–[17]. Traditional feature selection methods can be grouped into three categories, that is, filter, wrapper, and embedded approaches. Filter approaches

rank the features with correlation coefficients first and then select the most suitable features. Peng *et al.* proposed a minimal redundancy and maximal relevance [18] algorithm, which selects relevant features and simultaneously removes redundant features according to the mutual information. Another filter method, fast correlation-based filter [19], exploits symmetrical uncertainty for feature selection. Wrapper approaches apply a heuristic search strategy to determine feature subsets and evaluate them based on the classification performance. For example, Maldonado and Weber [20] proposed a wrapper method for feature selection problems using support vector machines (SVMs). The feature selection process and classification model are separated and independent in the filter and wrapper methods, and the wrapper methods might suffer from high computational complexity especially for high-dimensional data [21]. Embedded methods combine the advantages of the filter and wrapper methods, which perform the feature selection as part of its classicization process and obtain the feature subsets by optimizing the objective function, such as a sparse Bayesian-based feature selection method [16]. More recently, an evolutionary-computation-based [22] self-adaptive particle swarm optimization algorithm was proposed for large-scale feature selection problem [23].

However, most of the traditional feature selection algorithms ignore the cause-effect relationships between features and the class attribute, and thus do not lend themselves to make predictions of the results of actions or interventions [24], [25]. To build better interpretability for data and robustness against noise, causal feature selection methods were proposed, which discover the MB of a target [1]. Pellet and Elisseeff [26] theoretically proved that MB is the optimal solution for feature selection problem under the faithfulness condition. Therefore, causal feature selection methods based on MB have attracted more and more attention in recent years.

The first MB discovery algorithm for feature selection is the KS [6] algorithm proposed by Margaritis and Thrun [5] KS discovers MBs by minimizing the cross-entropy loss without theoretical guarantees to soundness. The GS [5] was the first sound MB discovery algorithm, whose framework with the growing phase and shrinking phase has become the basic strategy for the following algorithms. Tsamardinos *et al.* proposed IAMB [7] to improve the GS by reordering the variables in each iteration, which significantly improves the accuracy. Based on IAMB, many variants have been developed, including inter-IAMB [7], fast-IAMB [27], and KIAMB [10]. These algorithms are time efficient but require the number of samples to be exponential to the size of the MB, which means that insufficient samples will result in the performance degradation.

To improve the data efficiency while maintaining a reasonable time cost, a divide-and-conquer strategy for MB discovery is proposed, that is, first finding the PC of a target, then identifying spouses of the target. The MMMB [8] adopts the divide-and-conquer strategy to search MBs, in which the data requirement is dependent on the topological structure rather than the size of a variable set. Another early method, HITON-MB [9], interweaves the growing phase and the shrinking phase in the PC discovery process, so that the false PC variables can be excluded as early as possible. Pena *et al.* pointed out some errors in the PC discovery in the MMMB and HITON-MB and, then, they added the double check strategy to the MMMB framework and presented the PCMB [10] algorithm. Based on PCMB, iterative parent–child-based search of MB (IPCMB) [28] improves the time efficiency by connecting target to all other variables and removing the false variables in each iteration. De Morais and Aussem [11] proposed the MBOR, which uses a weak MB learner (a fast but data-inefficiency algorithm) to obtain the initial MB first and then corrects the MB through a divide-and-conquer search, which further improves the accuracy and data efficiency of the MB discovery. Recently, Gao and Ji [29] discovered the coexistence of the spouses and the false parent–child variables, and proposed a relatively efficient algorithm, simultaneous MB (STMB), which improves the time efficiency of the MB discovery.

Although existing causal feature selection algorithms improve the data efficiency and accuracy, there are still several true positives that cannot be identified, especially, with the noise or small-sized samples [12], [30]. In this article, we will focus on a type of incorrect CI tests occurred in the MB discovery, and further improve the accuracy of MB discovery through tackling this problem.

III. Notations and Definitions

In this article, the capital letters (such as X, Y) represent the random variables and the lowercase letters (such as x, y) represent their values, the capital bold italic letters (such as $\boldsymbol{U}$, $\boldsymbol{Z}$) denote variable sets. Specifically, let T denote the target variable, and $\boldsymbol{U}$ denote the (discrete random) variable set.

Definition 1 (CI): Variables X and Y are conditionally independent given a variable set $\boldsymbol{Z}$ if $P(X, Y|\boldsymbol{Z}) = P(X|\boldsymbol{Z})P(Y|\boldsymbol{Z})$, denoting as $X \perp Y|\boldsymbol{Z}$. Similarly, $X \not\perp Y|\boldsymbol{Z}$ represents that X and Y are conditionally dependent given a variable set $\boldsymbol{Z}$.

Existing MB discovery algorithms use the G^2-test [2] to implement the CI test. In this article, we use the symbol $\text{dep}(X, Y|\boldsymbol{Z})$ to represent the degree of the dependence between X and Y conditioned on $\boldsymbol{Z}$.

Definition 2 (Bayesian Network) [1]: Let $\mathbb{P}$ denote the joint probability distribution over a variable set $\boldsymbol{U}$ of a DAG $\mathbb{G}$. The triplet $\langle \boldsymbol{U}, \mathbb{G}, \mathbb{P} \rangle$ constitutes a BN, if $\langle \boldsymbol{U}, \mathbb{G}, \mathbb{P} \rangle$ satisfies the Markov condition: every variable is independent of any subset including its nondescendant variables given its parents in $\mathbb{G}$. In $\langle \boldsymbol{U}, \mathbb{G}, \mathbb{P} \rangle$, the joint probability $\mathbb{P}$ can be decomposed into a product of conditional probabilities as follows:

$$P(\boldsymbol{U}) = \prod_{X \in \boldsymbol{U}} P(X|Pa(X))$$

in which $Pa(X)$ denotes the parents of X.

Some terms in the BN need to be declared here. If there exists an edge from a variable (or node) X to Y, for example, $X \rightarrow Y$, then X is a parent of Y and Y is a child of X. A variable X is a spouse of Y if they share common child. In this article, we denote by $\boldsymbol{PC}(X)$, the set of parent–child variables of X, and $\boldsymbol{SP}(X)$, the set of spouse variables of X. For convenience, we will abbreviate parent–child and spouse as PC and SP, sometimes. Based on the definition of BN, some basic definitions in BN will be presented in the following.

Definition 3 (Blocked Path) [1]: A path π from variable A to B is blocked by a variable set $\boldsymbol{Z}$ iff: 1) π contains a chain $A \to X \to B$ or $A \leftarrow X \to B$ with the middle variable $X \in \boldsymbol{Z}$ and 2) π contains a collider $A \to X \leftarrow B$ with $X \notin \boldsymbol{Z}$.

Definition 4 (d-Separation) [1]: In a DAG $\mathbb{G}$, variable set $\boldsymbol{Z} \subset \boldsymbol{U}$ d-separates variables X and Y iff $\boldsymbol{Z}$ blocks every path from X to Y, denoting as $d\text{-}sep(X, Y|\boldsymbol{Z})$.

With Definitions 3 and 4, we give the definition of faithfulness condition.

Definition 5 (Faithfulness) [2]: Given a BN $\langle \boldsymbol{U}, \mathbb{G}, \mathbb{P} \rangle$, $\mathbb{P}$ is faithful to $\mathbb{G}$ when for any $X, Y \in \boldsymbol{U}$ and $\boldsymbol{Z} \subseteq \boldsymbol{U} - \{X, Y\}$, $X \perp Y|\boldsymbol{Z}$ in $\mathbb{P}$ iff $d\text{-}sep(X, Y|\boldsymbol{Z})$ in $\mathbb{G}$.

Definition 5 shows that CI and d-separation are equivalent if the dataset and its underlying BN are faithful to each other. Thus, we have Theorem 1 as follows.

Theorem 1: In BN $\langle \boldsymbol{U}, \mathbb{G}, \mathbb{P} \rangle$, for $X, T \in \boldsymbol{U}$, there is an edge between X and T iff $X \not\perp T|\boldsymbol{Z}$ for $\forall \boldsymbol{Z} \subseteq \boldsymbol{U} - \{X, T\}$.

Theorem 1 illustrates that if X is a PC variable of T, X and T are conditionally dependent for $\forall \boldsymbol{Z} \subseteq \boldsymbol{U} - \{X, T\}$. Theorem 1 will help us to design the algorithm to discover the PC variables. With Definition 5, we give the definition (Definition 6) and property (Theorem 2) of MB in a faithful BN.

Definition 6 (Markov Boundary) [1]: In a faithful BN $\langle \boldsymbol{U}, \mathbb{G}, \mathbb{P} \rangle$, the MB of a target variable T in $\mathbb{G}$ is unique and consists of its parents, children, and spouses.

Theorem 2 [1]: Given the $\boldsymbol{MB}(T)$, X is independent of T for any $X \in \boldsymbol{U} - \boldsymbol{MB}(T) - \{T\}$, that is, $X \perp T|\boldsymbol{MB}(T)$.

According to Definition 6, for each variable, its MB can be easily "read" from the structure of the corresponding faithful BN. To understand the intuition in the perspective of causal learning, we consider that the MB includes the direct causes (parents), direct effects (children), and other direct causes of direct effects (spouses) of the class attribute [24].

Theorem 3 [3], [4]: The MB is the optimal solution for the feature selection problem.

Theorem 3 presents the significance of MB research, which confirms that we can transfer the feature selection problem into the MB discovery of the class attribute in a faithful BN.

IV. PCMasking: Type of Incorrect CI-Tests in Markov Boundary Discovery

In this section, we focus on analyzing the three following questions: 1) Which variables are false negatives? 2) Why are these variables discarded? and 3) Can we seek a theoretical solution to support the improvement of the algorithm? To answer these questions, we first give two examples of the PCMasking phenomenon using the existing algorithms in Section IV-A, and then analyze the mechanism behind the PCMasking phenomenon in Section IV-B. And finally, we analyze the effect of the phenomenon on existing causal feature selection algorithms in Section IV-C.

A. Motivation

By running existing causal feature selection algorithms (PCMB, MBOR, and STMB) using the benchmark BN datasets with 1000 samples, we analyzed the incorrect CI tests in the discovered PC sets. We found that most of the false negatives (undetected PC variables) are independent of the target conditioning on other variables which have a strong correlation with the target (e.g., other PC variables). To further demonstrate this phenomenon, we take two subnetworks in the benchmark Alarm BN [31] as examples as follows.

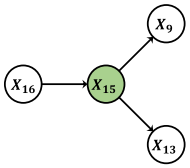
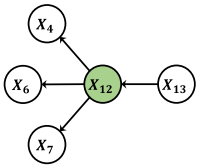
	Example 1	*Example* 2
Network		
Independence test	$X_{15} \perp X_{16}\|\{X_9, X_{13}\}$ $X_{15} \perp \{X_9, X_{13}\}\|X_{16}$	$X_{13} \perp X_{12}\|\{X_4, X_6, X_7\}$ $X_{12} \perp \{X_4, X_6, X_7\}\|X_{13}$
MaskingPC	$\{X_{16}\}$ and $\{X_9, X_{13}\}$	$\{X_{13}\}$ and $\{X_4, X_6, X_7\}$
Errors in Algorithm	When X_{16} is included by $PC(X_{15})$ first, X_9 and X_{13} will be ignored. When X_9 and X_{13} are included by $PC(X_{15})$ first, X_{16} will be ignored.	When X_4, X_6 and X_7 are included by $PC(X_{12})$ first, X_{13} will be ignored. When X_{13} are included by $PC(X_{12})$ first, X_4, X_6 and X_7 will be ignored.

Fig. 2. Examples: two subnetworks of Alarm BN to demonstrate the incorrect CI tests using the existing algorithms. The figure shows the corresponding DAG with targets highlighted in green, the errors in the CI tests, the MaskingPC, and the errors in the algorithm caused by PCMasking.

Consider Example 1 in Fig. 2 and take X_{15} as the target, then X_{16}, X_9, and X_{13} are PC variables. According to Theorem 1, $X_{16} \not\perp X_{15}|\{X_9, X_{13}\}$ and $\{X_9, X_{13}\} \not\perp X_{15}|X_{16}$ hold. However, in our experiments, we have found that the target X_{15} is independent of its parent X_{16} conditioning on its children $\{X_9, X_{13}\}$, and the target X_{15} is independent of its children $\{X_9, X_{13}\}$ conditioning on its parent X_{16}.

Using the other benchmark BN datasets, we also find many similar phenomenons in the experiments. Therefore, if we do not tackle the type of incorrect CI tests, many true PC variables may be discarded, leading to a low true positive discovery accuracy. Furthermore, since the second type of causal feature selection algorithms identifies the PC variables first, and then finds the SP variables, this type of incorrect CI tests will lead to the cascading errors during the SP discovery. Thus, it is important to address the problem to improve the true positive discovery accuracy. Before addressing this phenomenon, we first give a formal definition as follows.

Definition 7 (PCMasking): In a variable set $\boldsymbol{U}$, let $\boldsymbol{PC}(T)$ denote the parent–child set for the target T. $\boldsymbol{PC}_{S1}$ and $\boldsymbol{PC}_{S2}$ are the subsets of $\boldsymbol{PC}(T)$, and $\boldsymbol{PC}_{S1} \cap \boldsymbol{PC}_{S2} = \varnothing$. $\boldsymbol{PC}_{S1}$ and $\boldsymbol{PC}_{S2}$ are *PCMasking* for T if the following conditions hold:

$$T \perp \boldsymbol{PC}_{S1}|\boldsymbol{PC}_{S2}, T \perp \boldsymbol{PC}_{S2}|\boldsymbol{PC}_{S1}.$$

We call $\boldsymbol{PC}_{S1}$ and $\boldsymbol{PC}_{S2}$ as *MaskingPCs*.

Note that $\boldsymbol{PC}_{S1}$ and $\boldsymbol{PC}_{S2}$ in Definition 7 can not only be single variables but also be sets with several variables. The definition of PCMasking describes the results of the CI tests rather than the mechanism shown in the BN, that is to say, if the CI tests of an MB algorithm obtain the above independent relationship, then there exists the PCMasking phenomenon. For example, in Fig. 2, $\{X_{16}\}$ and $\{X_9, X_{13}\}$ are PCMasking for X_{15}, and $\{X_{13}\}$ and $\{X_4, X_6, X_7\}$ are PCMasking for X_{12}.

B. Mechanism Analysis

This section focuses on explaining the mechanism of PCMasking. The direct reason for PCMasking is the complete dependence between variables which can be mathematically described as $P(X = x|Y = y) = 1$. Obviously, complete dependence is a sufficient condition for PCMasking, which can be intuitively understood from an example. For Example 1 in Fig. 2, $P(X_{15} = 0|X_{16} = 0) = 1.0$ and $P(X_9 = 0, X_{13} = 1|X_{15} = 0) = 1.0$, then $\{X_{16}\}$ and $\{X_9, X_{13}\}$ are PCMasking for X_{15} according to Definition 7. In the following, we prove that complete dependence is a necessary and sufficient condition for PCMasking.

Theorem 4: In a dataset with variable set $\boldsymbol{U}$, variable set $\boldsymbol{X}, \boldsymbol{Y} \subseteq \boldsymbol{U}$ are the subsets of $\boldsymbol{PC}(T)$, and $\boldsymbol{X} \cap \boldsymbol{Y} = \varnothing$. $\boldsymbol{X}$ and $\boldsymbol{Y}$ are PCMasking for T if and only if $\exists \boldsymbol{x}, \boldsymbol{y}, t$ (values of $\boldsymbol{X}$, $\boldsymbol{Y}$, T) such that $P(\boldsymbol{X} = \boldsymbol{x}|T = t) = P(\boldsymbol{Y} = \boldsymbol{y}|T = t) = 1$ or $P(T = t|\boldsymbol{X} = \boldsymbol{x}) = P(T = t|\boldsymbol{Y} = \boldsymbol{y}) = 1$.

Proof: First, we prove the sufficiency of the condition. Since $\exists \boldsymbol{x}, \boldsymbol{y}, t$ s.t. $P(\boldsymbol{X} = \boldsymbol{x}|T = t) = P(\boldsymbol{Y} = \boldsymbol{y}|T = t) = 1$, then $\boldsymbol{X} \perp T$ and $\boldsymbol{Y} \perp T$ by Definition 1. Due to the complete dependence between $\boldsymbol{X}$ and T, T is independent with any others given a variable set $\boldsymbol{X}$. Therefore, $\boldsymbol{Y} \perp T|\boldsymbol{X}$. And $\boldsymbol{X} \perp T|\boldsymbol{Y}$ can be proved in the same way. Consequently, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are PCMasking for T. Similarly, we can prove that if $\exists \boldsymbol{x}, \boldsymbol{y}, t$ s.t. $P(T = t|\boldsymbol{X} = \boldsymbol{x}) = P(T = t|\boldsymbol{Y} = \boldsymbol{y}) = 1$, then we have $\boldsymbol{X} \perp T|\boldsymbol{Y}$ and $\boldsymbol{Y} \perp T|\boldsymbol{X}$.

Second, we utilize the concept of entropy in information theory to prove the necessity of the condition. Let $H(X)$ denote the information entropy of X. Since $\boldsymbol{X} \perp T|\boldsymbol{Y}$ and $\boldsymbol{Y} \perp T|\boldsymbol{X}$, then the conditional information entropy can be simplified as follows:

$$\begin{cases} H(\boldsymbol{X}, T|\boldsymbol{Y}) = H(\boldsymbol{X}|\boldsymbol{Y}) + H(T|\boldsymbol{Y}) \\ H(\boldsymbol{Y}, T|\boldsymbol{X}) = H(\boldsymbol{Y}|\boldsymbol{X}) + H(T|\boldsymbol{X}). \end{cases} \tag{1}$$

According to the additivity of the information entropy, we have

$$\begin{cases} H(\boldsymbol{X}, \boldsymbol{Y}, T) - H(\boldsymbol{Y}) = H(\boldsymbol{X}, \boldsymbol{Y}) - H(\boldsymbol{Y}) + H(T|\boldsymbol{Y}) \\ H(\boldsymbol{X}, \boldsymbol{Y}, T) - H(\boldsymbol{X}) = H(\boldsymbol{X}, \boldsymbol{Y}) - H(\boldsymbol{X}) + H(T|\boldsymbol{X}). \end{cases} \tag{2}$$

By solving the simultaneous equations in (2), we immediately obtain

$$H(T|\boldsymbol{X}) = H(T|\boldsymbol{Y}). \tag{3}$$

Obviously, there exist two cases that satisfy (3).

1) $H(T|\boldsymbol{X}) = H(T|\boldsymbol{Y}) \neq 0$: Since information entropy is based on a complete probability distribution, we suppose that $P(T = t_i|\boldsymbol{X} = \boldsymbol{x}_j) = p_{ij}$, $P(T = t_i|\boldsymbol{Y} = \boldsymbol{y}_k) = q_{ik}$, and $P(T = t_i|\boldsymbol{X} = \boldsymbol{x}_j, \boldsymbol{Y} = \boldsymbol{y}_k) = h_{ijk}$ $(i \in [1, |T|], j \in [1, |\boldsymbol{X}|], k \in [1, |\boldsymbol{Y}|])$, where $|\boldsymbol{X}|$ denotes the domain of $\boldsymbol{X}$. We try to solve h_{ijk} via equation set

$$\begin{cases} \sum_{k=1}^{|\boldsymbol{Y}|} h_{ijk} = p_{ij} \\ \sum_{j=1}^{|\boldsymbol{X}|} h_{ijk} = q_{ik} \end{cases} \tag{4}$$

where p_{**} and q_{**} are used as the parameters. Note that there are $(|T - 1|)(|\boldsymbol{X}| + |\boldsymbol{Y}|)$ equations and $(|T - 1|)(|\boldsymbol{X}|)(|\boldsymbol{Y}|)$ dependent variables. Therefore, the solution of (4) can be obtained iff $H(T|\boldsymbol{X}) = H(T|\boldsymbol{Y}) = 0$, contradicting the condition of the case. Consequently, there exists no $P(T = t_i|\boldsymbol{X})$ and $P(T = t_i|\boldsymbol{Y})$ satisfying the condition. As a result, case 1) does not hold true.

2) $H(T|\boldsymbol{X}) = H(T|\boldsymbol{Y}) = 0$: Since the condition of $H(X) = 0$ is that there exists x s.t. $P(X = x) = 1$, then we can conclude that $\exists \boldsymbol{x}, \boldsymbol{y}, t$ s.t. $P(T = t|\boldsymbol{X} = \boldsymbol{x}) = 1$ and $P(T = t|\boldsymbol{Y} = \boldsymbol{y}) = 1$. According to the symmetry of the complete dependence, we can further conclude that $P(\boldsymbol{X} = \boldsymbol{x}|T = t) = 1$ and $P(\boldsymbol{Y} = \boldsymbol{y}|T = t) = 1$.

Summarizing: Theorem 4 is true. (Q.E.D.) ■

Case	*A*	*B*	*C*	
1	0	1	0	Training Samples
2	0	1	0	
3	0	1	0	
4	0	1	0	
5	1	0	1	
6	1	0	1	
7	1	0	1	
8	1	0	1	
9	0	0	0	Missing Samples
10	1	1	0	

Fig. 3. Example of crucial samples: when the last two samples in the dataset are missing (highlighted in blue), the dependency between A and B (C) will change from an uncertain relationship to a deterministic relationship.

Theorem 4 shows that if there exists PCMasking phenomenon, then it must be due to the complete dependence between variables. Note that standard BN datasets (e.g., Alarm) all satisfy the faithfulness condition, which makes the complete dependence not allowed in the original probability distribution in BN. The question will be, why does PCMasking phenomenon exist in the standard BN datasets?

Actually, the direct factor in determining the correctness of CI test is the underlying probability distribution in the samples, instead of the original probability distribution in the BN, we need to focus on the influence from the samples. Note that the size of samples used in our experiments is 1000. Although it is not small for the Alarm BN with 37 variables, the lack of some crucial samples may cause a strong change in the probability distribution of the variables, which makes the datasets not satisfy the original distribution in the DAG. The most significant change is that the dependency between variables changes from an uncertain relationship to a deterministic relationship. For example, suppose A, B, and C each have space $\{0, 1\}$, and we have these samples in Fig. 3. We can conclude from Fig. 3 that A and B (and C) are dependent and the relationship between them is stochastic, that is, given the value of A, we cannot determine the value of B or C. However, when we suppose our sample contains the first eight data items and the cases 9 and 10 (highlighted in blue) have been lost, then variable dependency will become deterministic, that is, $P(B = 1|A = 0) = 1$ and $P(C = 1|B = 0) = 1$.

Due to the loss of some crucial samples, the complete dependence appears in the underlying probability distribution of the samples, leading to the PCMasking phenomenon. Obviously, it will further interfere with the MB discovery as we discussed above. Unfortunately, there is no efficient way to test the complete dependence for MB discovery. Therefore, we need to exploit a variable-relationship-based method to detect the MaskingPCs, instead of a dependence-test-based way.

C. Influence and Solution

According to the examples analyzed above, the PCMasking phenomenon will interfere with the recognition of PC variables, making existing methods discover false PC and SP sets. Given the problem of existing algorithms, we propose Theorem 5 to formalize the influence of PCMasking phenomenon on MB discovery and give an insight on how to detect the MaskingPCs in BN.

Theorem 5: In variable set $\boldsymbol{U}$, let $\boldsymbol{PC}_R(A) \subset \boldsymbol{U}$ denote the real parent–child variable set of variable A, and $\boldsymbol{PC}_O(A) \subset \boldsymbol{U}$ denote the parent–child variable set of variable A found by the MB algorithm Ω. If a variable $Y \in \boldsymbol{PC}_R(T)$ and a variable set $\boldsymbol{X} \subseteq \boldsymbol{PC}_R(Y)$ are PCMasking for T, then: 1) T might be ignored by Ω, that is, there exists the situation that $T \notin \boldsymbol{PC}_O(Y)$ and 2) $Y \in \boldsymbol{PC}_O(T)$.

Proof: We first prove that there exists the situation that $T \notin \boldsymbol{PC}_O(Y)$. $\boldsymbol{X}$ and Y are PCMasking for T, we have $T \perp Y|\boldsymbol{X}$ based on Definition 7. Since $\boldsymbol{X} \subseteq \boldsymbol{PC}_R(Y)$, when the variables in $\boldsymbol{X}$ are all selected into the $\boldsymbol{PC}_O(Y)$ before T, the existing algorithms will misjudge T as a non-PC variable of Y according to Theorem 1. Therefore, we prove that $T \notin \boldsymbol{PC}_O(Y)$. Second, we prove $Y \in \boldsymbol{PC}_O(T)$. Whatever order the variables in $\boldsymbol{PC}_R(T)$ are selected to $\boldsymbol{PC}_O(T)$, the variable Y will be added to $\boldsymbol{PC}_O(T)$ and will not be deleted according to Theorem 1. Therefore, $Y \in \boldsymbol{PC}_O(T)$. ■

In Theorem 5, Proposition 1) shows that existing algorithms do not guarantee the correct output in the BN with PCMasking phenomenon. If the PC subset of target T is a MaskingPC, then existing algorithms may fail to discover the PC set of T under the influence of false CI tests. Taking Fig. 2 as an example. Due to the PCMasking, we have $X_{16} \perp X_{15}|\{X_9, X_{13}\}$. According to Theorem 1 (criteria for identifying PC variables), when variables X_9 and X_{13} are selected into $\boldsymbol{PC}_O(T)$ in advance, X_{16} will not be included by $\boldsymbol{PC}_O(T)$ since $T \perp X_{16}|\{X_9, X_{13}\}$ although $X_{16} \in \boldsymbol{PC}_R(T)$. Proposition 2) describes that the MaskingPC will break the symmetry between the PC variables. Also taking Fig. 2 as an example. Since there is no PCMasking phenomenon, then we have $X_{15} \in \boldsymbol{PC}_O(X_{16})$, while $X_{16} \notin \boldsymbol{PC}_O(X_{15})$ according to the above analysis. Therefore, the symmetry between X_{15} and X_{16} is broken in the output of the existing algorithms. Proposition 2) shows the possibility that we can detect the MaskingPCs and simultaneously recover the discarded PC variables via the symmetry property, which would be used for the improvement of MB discovery algorithm.

V. CCMB and PM-CCMB Algorithms

This section presents the proposed MB discovery algorithms, CCMB in Section V-A and PM-CCMB in Section V-B.

A. CCMB

This section focuses on the specific design of the CCMB algorithm, including an innovative PC discovery process, in which there exists the handling of PCMasking phenomenon. The pseudocode of the CCMB is shown as Algorithm 1, which consists of three parts. In part 1 (line 3), CCMB discovers the PC variables using a subroutine called *FindPC* (detailed in Algorithm 2). *FindPC* will find all the true positives except the PC variables with PCMasking phenomenon. Part 2 (lines 4–11) in CCMB detects the MaskingPCs and recovers the discarded variables. Based on the discovered PC sets in parts 1 and 2, CCMB calls part 3 (lines 12–21) to discover the SP set.

Algorithm 1 *CCMB(T)*: Discover the MB of T

1: **Input:** Target variable T, variables set $\boldsymbol{U}$.
2: Initialize the PC variable set, SP variable set and PCMasking table $\boldsymbol{PC}, \boldsymbol{SP}, \boldsymbol{PCMTab} \leftarrow \varnothing$.
{*Part 1: Discover the PC variables.*}
3: $\boldsymbol{PC} \leftarrow FindPC(T)$
{*Part 2: Detect the MaskingPCs and eliminate the effect of them.*}
4: **for** each $X \in \boldsymbol{U} - \{T\}$ **do**
5: **if** $T \in FindPC(X)$ and $X \notin \boldsymbol{PC}$, **then**
6: $\boldsymbol{PCMTab} \leftarrow \boldsymbol{PCMTab} \cup \{[T, X]\}$
7: **end if**
8: **end for**
9: **for** each $[T, X] \in \boldsymbol{PCMTab}$ **do**
10: $\boldsymbol{PC} = \boldsymbol{PC} \cup \{X\}$.
11: **end for**
{*Part 3: Discover the spouse variables.*}
12: **for** each $Y \in \boldsymbol{PC}$ **do**
13: **for** each $X \in \boldsymbol{PC}_Y$ **do**
14: **if** $X \notin \boldsymbol{PC}$, **then**
15: find $\boldsymbol{Z}$ s.t. $T \perp X|\boldsymbol{Z}$ and $T, X \notin \boldsymbol{Z}$.
16: **if** $T \not\perp X|\boldsymbol{Z} \cup \{Y\}$, **then**
17: $\boldsymbol{SP} \leftarrow \boldsymbol{SP} \cup \{X\}$.
18: **end if**
19: **end if**
20: **end for**
21: **end for**
22: **Output**: The Markov boundary of T, $\boldsymbol{MB} \leftarrow \boldsymbol{PC} \cup \boldsymbol{SP}$.

CCMB shares the same basic hypotheses with existing algorithms, that is, faithfulness and causal sufficiency. They will be used for each theorem and its proof without restatement. In the following text, we will describe the process of CCMB step by step, and give some theoretical analyses.

The **part 1** in CCMB (Algorithm 1, line 3) identifies the PC variables except the MaskingPCs, whose process is detailed in Algorithm 2. *FindPC* consists of three steps in an iteration.

Step 1 (Lines 4–14 of Algorithm 2): Build a candidate PC set $\boldsymbol{CanPC}$. Different from the existing methods, we exploit a two-phase process to remove the false PC variables from the current candidate PC set $\boldsymbol{CanPC}$, which could effectively reduce the number of iteration. The method is extracted from Theorem 1: given variable set $\boldsymbol{Z}$, if X and T are conditionally independent, then X will be removed from the $\boldsymbol{CanPC}$. Phase I (lines 4–9) traverses the $\boldsymbol{Z}$ from current $\boldsymbol{SPC}$. In line 5, Phase I finds a conditioning set $\boldsymbol{Sep}[X]$ for each variable X, which can minimize the conditional dependence between X and T, if X and T are independent conditioned on $\boldsymbol{Sep}[X]$, then X will be removed from the $\boldsymbol{CanPC}$ (lines 6–8). Phase II (lines 10–14) selects the $\boldsymbol{Z}$ out of the $\boldsymbol{SPC}$ and specially considers a general d-separation in BN, that is, the dependency is blocked by a

Algorithm 2 *FindPC(T)*: Search the PC Subset of T

1: **Input:** Target variable T, variables set $\boldsymbol{U}$.
2: Initialize the PC variable subset $\boldsymbol{SPC} \leftarrow \varnothing$, and the candidate PC variable set $\boldsymbol{CanPC} \leftarrow \boldsymbol{U} - \{T\}$.
3: **while** $\boldsymbol{CanPC} \neq \varnothing$ **do**
{*Step 1: Find the candidate PC variables.*}
4: **for** each $X \in \boldsymbol{CanPC}$ **do**
5: $\boldsymbol{Sep}[X] = \arg\min_{\boldsymbol{Z} \subseteq \boldsymbol{SPC}} dep(T, X|\boldsymbol{Z})$.
6: **if** $T \perp X|\boldsymbol{Sep}[X]$, **then**
7: $\boldsymbol{CanPC} \leftarrow \boldsymbol{CanPC} - \{X\}$.
8: **end if**
9: **end for**
10: **for** each $X, Y \in \boldsymbol{CanPC}$ **do**
11: **if** $X \not\perp Y$ and $T \perp X|Y$, **then**
12: $\boldsymbol{CanPC} \leftarrow \boldsymbol{CanPC} - \{X\}$.
13: **end if**
14: **end for**
{*Step 2: Score the candidates and select the best.*}
15: **for** each $X \in \boldsymbol{CanPC}$ **do**
16: $Score[X] = dep(T, X|\boldsymbol{Sep}[X])$
17: **end for**
18: $Y = \arg\max_{X \in \boldsymbol{CanPC}} Score[X]$.
19: $\boldsymbol{SPC} \leftarrow \boldsymbol{SPC} \cup \{Y\}$, $\boldsymbol{CanPC} \leftarrow \boldsymbol{CanPC} - \{Y\}$.
{*Step 3: Delete the false variables.*}
20: **for** each $X \in \boldsymbol{SPC}$ **do**
21: **if** $\exists \boldsymbol{Z} \subseteq \boldsymbol{SPC} - \{X\}$ s.t. $T \perp X|\boldsymbol{Z}$, **then**
22: $\boldsymbol{SPC} \leftarrow \boldsymbol{SPC} - \{X\}$.
23: **end if**
24: **end for**
25: **end while**
26: **Output**: The PC subset $\boldsymbol{SPC}$.

single variable, which is also dependent on the two variables participating in CI test. In lines 11–13, Phase II removes variable X from the $\boldsymbol{CanPC}$ if X and Y are dependent on each other and Y blocks the dependency between T and X.

Theorem 6: Any true PC variable $X \in \boldsymbol{PC}_R(T)$ is included in $\boldsymbol{CanPC}$ until it is added to $\boldsymbol{SPC}$.

Proof: According to Theorem 1, if X is a PC variable of T, then $T \not\perp X|Z$ given $\forall \boldsymbol{Z} \subseteq \boldsymbol{U} - \{X, T\}$. Let us assume that variable $X \in \boldsymbol{PC}_R(T)$, and X is removed from $\boldsymbol{CanPC}$ at lines 4–9 or lines 10–14 in Algorithm 2, then $\exists \boldsymbol{Sep}[X]$ such that $T \perp X|\boldsymbol{Sep}[X]$. Thus, we have $X \notin \boldsymbol{PC}_R(T)$, contradicting the assumption. Therefore, any $X \in \boldsymbol{PC}_R(T)$ will not be removed from $\boldsymbol{CanPC}$ until it is added to $\boldsymbol{SPC}$. ■

Step 2 (Lines 15–19 of Algorithm 2): Use scoring function $Score[X]$ to estimate the variables in $\boldsymbol{CanPC}$. $Score[X]$ is the minimum of the conditional dependence between X and T, which has been calculated in line 5 of step 1. The dependence between X and T conditioned on $\boldsymbol{Sep}[X]$ can be used to score the candidates in $\boldsymbol{CanPC}$ (line 16). We would select the variable with the highest score and add it to the $\boldsymbol{SPC}$ (lines 18 and 19).

Step 3 (Lines 20–25 of Algorithm 2): Detect the false variables. As a heuristic method, the process above might introduce some false positives into the $\boldsymbol{SPC}$. Therefore, lines 20–24 will remove the false variables from the current $\boldsymbol{SPC}$ according to Theorem 1, that is, if there exists $\boldsymbol{Z} \subseteq \boldsymbol{SPC} - \{X\}$ such that $T \perp X|\boldsymbol{Z}$, then X is a false positive.

Theorem 7: For variable $X \in \boldsymbol{U} - \{T\}$: 1) if $X \in \boldsymbol{PC}_R(T)$ and there exists a variable set $\boldsymbol{S} \subseteq \boldsymbol{PC}_R(T)$ such that X and $\boldsymbol{S}$ are PCMasking for T, then $X \in \boldsymbol{SPC}$ or $X \notin \boldsymbol{SPC}$; 2) if $X \in \boldsymbol{PC}_R(T)$ and there is no MaskingPCs for T, then $X \in \boldsymbol{SPC}$; and 3) if $X \notin \boldsymbol{PC}_R(T)$, then $X \notin \boldsymbol{SPC}$.

Proof: According to 1) of Theorem 5, proposition 1) of Theorem 7 is true. In proposition 2), since there is no MaskingPCs, if $X \in \boldsymbol{PC}_R(T)$, then $T \not\perp X|\boldsymbol{Z}$ with $\forall \boldsymbol{Z} \subseteq \boldsymbol{SPC} - \{X\}$ according to Theorem 1. By Theorem 6, X will not be removed until it is added to $\boldsymbol{SPC}$. Therefore, proposition 2) of Theorem 7 is true. In proposition 3), since $X \notin \boldsymbol{PC}_R(T)$, then $\exists \boldsymbol{Z} \subseteq \boldsymbol{SPC} - \{X\}$ such that $T \perp X|\boldsymbol{Z}$. According to Theorem 6, proposition 3) is also true. ■

The **Part 2** in CCMB (Algorithm 1, lines 4–11) is the biggest improvement compared with other methods. Based on the output of *FindPC* (Algorithm 2), CCMB performs the cross-check and complement processes, which can shield the PCMasking phenomenon and simultaneously find more true positives. The cross-check process (lines 4–8) aims to detect the MaskingPCs and record the variables with its corresponding target into the PCMasking table $\boldsymbol{PCMTab}$. This process utilizes the property that the MaskingPCs could break the symmetry between the PC variables and the target (proposed in Theorem 5). Specifically, if T is included in the PC set of X while the PC set of T does not include X, then the cross-check process can detect the MaskingPCs by the asymmetry between X and T. After that, the complement process (lines 9–11) repairs the asymmetry and completes the PC set.

The **Part 3** in CCMB (Algorithm 1, lines 12–21) discovers the SP set. It uses the topology information of the BN to find the colliders in the PC set of T. The spouses are selected from the union of the PC sets of the PC variables of T ($\boldsymbol{PC}_Y$ in line 13 denotes the PC set of variable Y found by CCMB). To illustrate the correctness of CCMB, Theorem 8 is proposed and proved as follows.

Theorem 8: CCMB outputs the correct MB.

Proof: First, we prove that CCMB can output the correct PC set. Based on proposition 3) of Theorem 7, CCMB can delete all false PC variables. According to proposition 1) of Theorem 7, some of the PCMasking variables may not be included in $\boldsymbol{SPC}$. Denoting one of them as X, then $T \in \boldsymbol{PC}_O(X)$ according to proposition 2) of Theorem 4. Therefore, the PCMasking variables can also be correctly found by lines 4–11 of Algorithm 1. The remaining PC variables can be found according to proposition 2) of Theorem 7. Second, we prove that CCMB can find the correct SP set. For a variable $X \in \boldsymbol{SP}$, X has a common child with T, which can be found by lines 12–21 in Algorithm 1. Therefore, CCMB can find the correct SP set. Summarizing, CCMB outputs the correct MB. ■

To further explain the algorithm, we will take Fig. 2 as an example and revisit the steps of CCMB that were given in Algorithms 1 and 2. Initially, $\boldsymbol{CanPC} = \{X_{16}, X_9, X_{13}\}$ in the first iteration and there is no variable removed from the $\boldsymbol{CanPC}$ in lines 6–8 of Algorithm 2. Let us suppose a special case to show the effect of detecting the MaskingPCs,

that is, $\text{dep}(X_9, X_{15}) > \text{dep}(X_{13}, X_{15}) > \text{dep}(X_{16}, X_{15})$ and $\text{dep}(X_{13}, X_{15}|\{X_9\}) > \text{dep}(X_{16}, X_{15}|\{X_9\})$. Then, variable X_{16} will be selected into the ***SPC*** in the first iteration, and X_9 and X_{13} will be removed from the ***CanPC*** according to the lines 11–13 due to the PCMasking phenomenon as mentioned in Section IV. We conclude that the outputs of Algorithm 2 are $\boldsymbol{SPC}[X_{15}] = \{X_{16}\}$ and $\boldsymbol{SPC}[X_9] = \{X_{15}\}$, $\boldsymbol{SPC}[X_{13}] = \{X_{15}\}$. Note that, existing algorithms will take three iterations to finish the PC discovery and obtain an incorrect result, while CCMB discovers the PC efficiently in only one iteration and continues to select the undetected MaskingPCs. In Algorithm 1, lines 4–8 traverse the outputs of Algorithm 2 and build the $\boldsymbol{PCMTab} = \{[X_{15}, X_9], [X_{15}, X_{13}]\}$, which means that the symmetry between X_{15} and X_9 (X_{13}) are broken due to the PCMasking phenomenon. Lines 9–11 repair the asymmetry and the outputs of Algorithm 1 are $\boldsymbol{SPC}[T] = \{X_9, X_{13}, X_{16}\}$. Hence, our proposed CCMB obtains the expected outputs.

B. Pipeline Machine: Acceleration Strategy

As the second type of causal feature selection algorithms, CCMB has better accuracy but lower efficiency. In this section, we will propose an acceleration strategy to guarantee that CCMB can be efficient in relatively large-scale datasets.

First, we point out the problems in the algorithms, and take the pseudocodes in CCMB as examples to illustrate our idea. The MB discovery algorithms are designed based on the CI tests, implemented by G^2-test, which is a time-consuming process. We found that there are a large number of iterative processes consisting of the same CI tests, which are executed multiple times by the algorithm. For example, when searching for the PC set of variable X, we need to perform CI tests between Y and X conditioned on different $\boldsymbol{Z}$, while the same tests will be performed when searching for PC set of variable Y. Lines 3–10 in Algorithm 1 are also the steps that need to be performed only once between a pair of variables. The redundant tests exist not only between different variables, but also between different iterations of the same variable. Take lines 20–24 of Algorithm 2 as an example. ***SPC*** is similar between adjacent iterations since only a small number of variables are added or removed. Therefore, there are a large number of duplicate CI tests. In addition, redundant tests also exist in the same iteration. For example lines 6 and 21 in Algorithm 2.

Inspired by the problems above, we propose an acceleration methodology for the MB discovery algorithm called pipeline machine (PM). The main idea of PM is using a buffer to organize the redundant CI test results involved in the MB discovery process, so that more efficient search can be exploited to replace the complex calculations.

The structure of PM is shown in Fig. 4. PM is essentially a linked list consisting of several variable cells (in the red dashed frame). Each variable cell points to a CI test Information Table (in the blue dashed frame), which stores the new CI test results involved in the iteration of the corresponding variable being added to the MB. For the convenience of searching, the CI test Information Table is organized into a two-level structure in which CI tests with the same size of conditioning sets are stored in an II-level Table, and the addresses of the II-level Table are stored in the I-level Table.

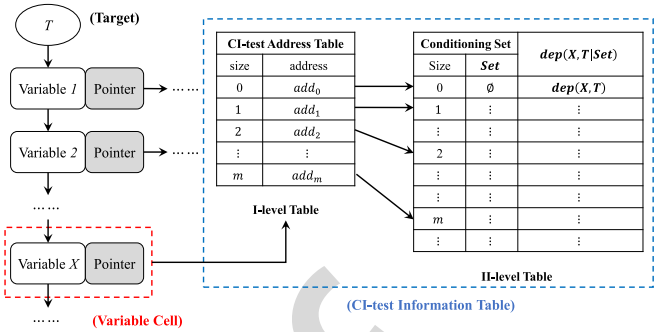


Fig. 4. PM for MB discovery algorithm. The skeletal structure is a linked list consisting of several variable cells. The red dashed part denotes the variable cell and the blue dashed part denotes the CI test Information Table which is used to store the results of the CI tests that have been executed.

When a new variable is added to the MB, the PM adds a variable cell to the end of the linked list. Before executing a CI test, the PM-algorithm first determines whether the CI test is already in the PM via the linked list structure of the PM. Specifically, if all the variables in the conditioning set (of the CI test) are in the linked list of the PM, then the CI test is in the PM, and its location is in the CI-test Information Table corresponding to the conditioning set variable closest to the end of the linked list. For example, in the line 6 of Algorithm 2, if all the variables in $\boldsymbol{Sep}[X]$ are in the linked list of the PM, then the result of $\text{dep}(X, T|\boldsymbol{Sep}[X])$ can be obtained from the CI-test Information Table corresponding to the variable in the $\boldsymbol{Sep}[X]$ closest to the end of the linked list. As can be seen from the mechanism, PM reduces the computational cost of redundant CI test, thus improving the efficiency of the algorithm.

In this article, we refer CCMB with PM as PM-CCMB, indicating that it is an enhanced version based on PM. Note that PM-CCMB could maintain the same accuracy with CCMB and simultaneously improve the time efficiency through employing the PM to store the repetitive computations, thus, PM-CCMB need to require additional memory for the PM. We will discuss the space complexity of PM-CCMB in the following. Assume that the largest size of conditioning sets during the PC search is m, then the *size* in the II-level Table (in Fig. 4) of variable T is between 0 and $\max\{m, |\boldsymbol{SPC}_T|\}$, where $\boldsymbol{SPC}_T$ denotes the current PC set after a certain iteration. Since there are $|\boldsymbol{PC}_T|$ II-level Tables, then there are $\sum_{i=0}^{|\boldsymbol{PC}_T|-1} \sum_{j=0}^{\max\{m,i\}} \binom{i}{j}$ CI-test results need to be stored. Therefore, PM-CCMB needs to store $\sum_{T\in\boldsymbol{U}} \sum_{i=0}^{|\boldsymbol{PC}_T|-1} \sum_{j=0}^{\max\{m,i\}} \binom{i}{j}$ CI-test results.

VI. Experimental Studies

In this section, we present the experimental studies of the proposed algorithms, CCMB and PM-CCMB. In Section VI-A, we first use the subnetworks of Alarm BN in Fig. 2 to demonstrate the effectiveness of CCMB to detect the MaskingPCs. Then, we perform experiments on 12 standard BN datasets in Section VI-B to evaluate several performance

aspects of the algorithms, including accuracy and time efficiency. In order to validate the performance in feature selection problem, extensive experiments and statistical analysis are performed on a newly developed electroencephalography (EEG) dataset, SEED [32], [33], to compare our algorithms with other state-of-the-art methods in Section VI-C.

To illustrate the effectiveness of the proposed methods, the four state-of-the-art MB discovery algorithms are compared.

1) *IAMB [7]:* IAMB is the first type of causal feature selection algorithms, focusing on time efficiency.
2) *PCMB [10]:* PCMB is the second type of causal feature selection algorithms, which aims to improve accuracy.
3) *MBOR [11]:* MBOR combines the idea of two types of algorithms to improve accuracy.
4) *STMB [29]:* STMB is a recently proposed algorithm, which incorporates the double check process of PCMB into the SP discovery to improve the time efficiency.

In addition, we also select two well-established information-theoretical-based feature selection algorithms as references in the experiment of emotion recognition task.

1) *FCBF [19]:* A fast correlation-based filter, which exploits the symmetrical uncertainty for feature selection.
2) *mRMR [18]:* An algorithm of removing redundant features while ensuring maximum correlation.

To measure the strength of the conditional dependence between variables, all the MB discovery algorithms use the G^2-test to implement the CI tests as previous work ([11], [29], etc.) at a significance level of 0.01. All codes are implemented in C++, and all experiments are conducted on a computer with Inter i5-8500 3.00-GHz CPU and 16-GB memory.

A. Alarm Subnetwork: The Effectiveness for Detecting MaskingPCs

The proposed CCMB has been proved to detect the MaskingPCs and shield its impact. This experiment selects the two subnetworks of Alarm BN in Fig. 2 to demonstrate the effectiveness of CCMB to solve the PCMasking phenomenon.

The two examples in Fig. 2 take variables X_{15} and X_{12} as the target, respectively. To validate the existence of PCMasking phenomenon, we test the CI with 1000 samples and obtain the results as follows:

$$X_{15} \perp X_{16} | \{X_9, X_{13}\}, X_{15} \perp \{X_9, X_{13}\} | X_{16}$$
$$X_{12} \perp X_{13} | \{X_4, X_6, X_7\}, X_{12} \perp \{X_4, X_6, X_7\} | X_{13}$$

which means that X_{16} and $\{X_9, X_{13}\}$ are PCMasking for X_{15}, and X_{13} and $\{X_4, X_6, X_7\}$ are PCMasking for X_{12}. According to Theorem 4, the existing algorithms cannot find out the correct PC sets of X_{15} and X_{12}. Table I provides the PC variables of X_{15} and X_{12} found by our algorithm and other MB discovery algorithms.

For Example 1 in Fig. 2, variables X_9 and X_{13} have stronger correlation with variable X_{15}, such that they will enter into $\boldsymbol{PC}(X_{15})$ in advance and then prevent variable X_{16}. When using $\{X_9, X_{13}\}$ as the conditioning set, we have $\text{dep}(X_{15}, X_{16} | \{X_9, X_{13}\}) < -1.0$. Therefore, X_{16} is misjudged as a non-PC variable according to Theorem 1. This error will occur in all other existing algorithms. Moreover, STMB and PCMB further misjudge that X_{15} is not a PC variable of X_{16}. In contrast, our proposed CCMB achieves the complete PC sets for variables X_{15} and X_{16} due to the cross-check and complement processes that can shield the influence of the PCMasking for X_{15}. Similarly, when using $\{X_4, X_6, X_7\}$ as the conditioning set in Example 2, X_{13} will be misjudged as a non-PC variable in all the existing algorithms while CCMB can avoid errors on MaskingPCs.

TABLE I
SEARCHED PC VARIABLES ON ALARM SUBNETWORK WITH PCMASKING PHENOMENON

Algorithms	Results of Example 1	Results of Example 2
IAMB	$X_{16} \notin \boldsymbol{MB}(X_{15})$ $X_{15} \in \boldsymbol{MB}(X_{16})$	$X_{13} \notin \boldsymbol{MB}(X_{12})$ $X_{12} \in \boldsymbol{MB}(X_{13})$
PCMB	$\boldsymbol{PC}(X_{15}) = \{X_9, X_{13}\}$ $\boldsymbol{PC}(X_{16}) = \varnothing$	$\boldsymbol{PC}(X_{12}) = \{X_4, X_6, X_7\}$ $\boldsymbol{PC}(X_{13}) = \varnothing$
MBOR	$\boldsymbol{PC}(X_{15}) = \{X_9, X_{13}\}$ $\boldsymbol{PC}(X_{16}) = \{X_{15}\}$	$\boldsymbol{PC}(X_{12}) = \{X_4, X_6, X_7\}$ $\boldsymbol{PC}(X_{13}) = \{X_{12}\}$
STMB	$\boldsymbol{PC}(X_{15}) = \{X_9, X_{13}\}$ $\boldsymbol{PC}(X_{16}) = \varnothing$	$\boldsymbol{PC}(X_{12}) = \{X_4, X_6, X_7\}$ $\boldsymbol{PC}(X_{13}) = \varnothing$
CCMB	$\boldsymbol{PC}(X_{15}) = \{X_9, X_{13}, \boldsymbol{X_{16}}\}$ $\boldsymbol{PC}(X_{16}) = \{X_{15}\}$	$\boldsymbol{PC}(X_{12}) = \{X_4, X_6, X_7, \boldsymbol{X_{13}}\}$ $\boldsymbol{PC}(X_{13}) = \{X_{12}\}$

TABLE II
STATISTICAL INFORMATION OF THE STANDARD BN DATASETS

Data set	Alarm	Alarm3	Child	Child3	Insurance	Insurance3
#Variables	37	111	20	60	27	81
#Edges	46	149	25	79	52	163
Data set	Alarm5	Alarm10	Child5	Child10	Insurance5	Insurance10
#Variables	185	370	100	200	135	270
#Edges	265	570	126	257	281	556

B. Standard BN Datasets: The Accuracy and Time Efficiency for MB Discovery

In this section, we conduct experiments on standard BN datasets [34] to evaluate the performance of CCMB and other algorithms. The standard BN data contains 12 network datasets. Table II provides the statistical information of these datasets, including the number of variables, the number of edges, and the training size in the experiments. The experiments consist of two parts, to verify the accuracy and time efficiency of the proposed algorithms, respectively.

We run all the MB algorithms for each variable and repeat these algorithms 20 times with different samples. The size of training samples turns from $\{500, 1000, \ldots, 4000\}$, respectively. We compare the accuracy and time efficiency of the algorithms under the same sets of samples and analyze the changes of the performance with increased training sample scale.

Accuracy: The frequently used metrics *Distance* [7], [10], [11], [29] is adapted to measure the accuracy of MB variables searching. The *Distance* measures the distance between the detected MB and the true MB, calculated by: $\text{Distance} = \sqrt{(1-\text{Precision})^2 + (1-\text{Recall})^2}$, where the *Precision* is the fraction of retrieved true positives over the total amount of detected MB variables, and the *Recall* is the fraction of retrieved true positives over the total amount of true MB variables. Thus, the lower *Distance* indicates the detected MB is closer to the true MB.

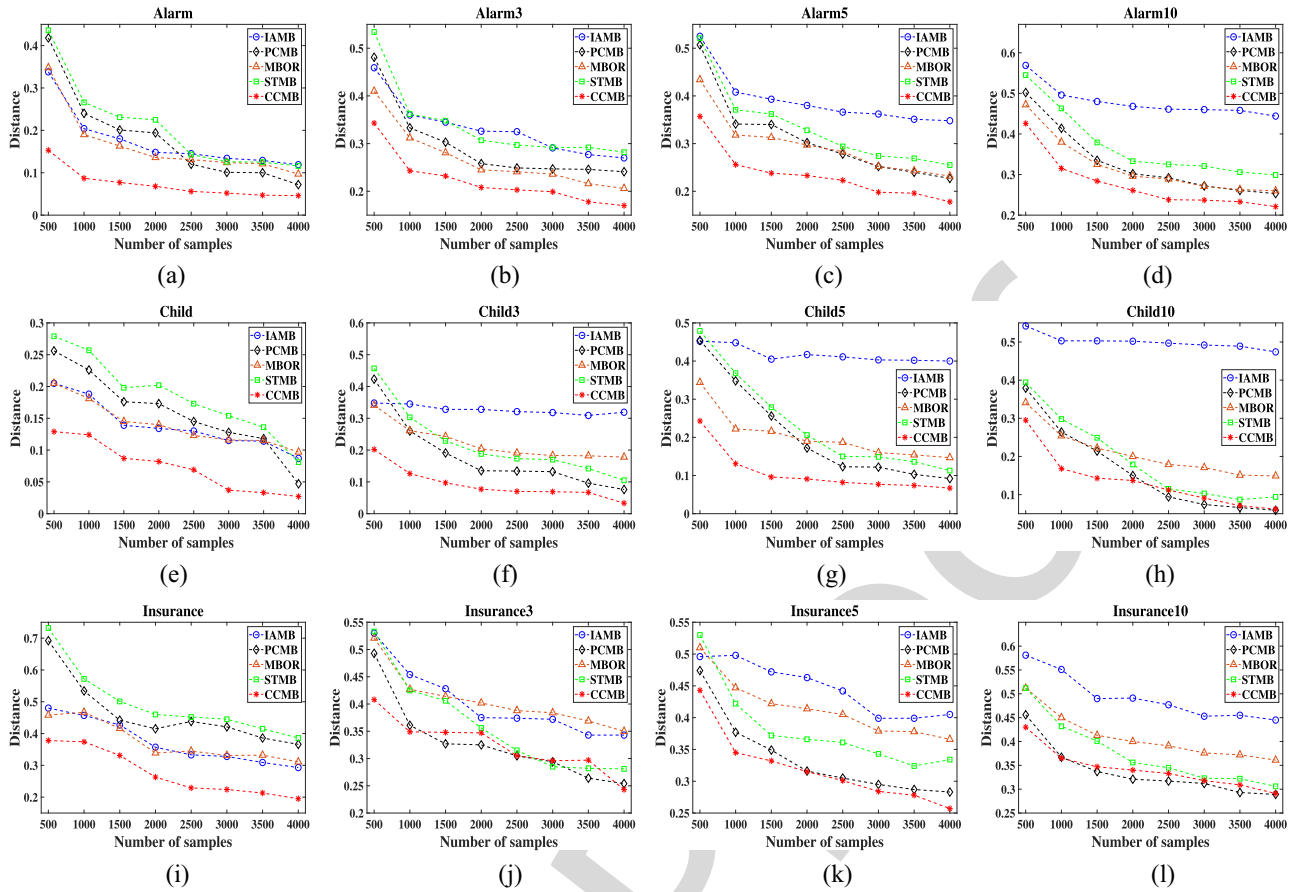


Fig. 5. Results of the MB discovery experiments for the accuracy of CCMB and other algorithms on the 12 standard BN datasets. Note that the curves of PM-CCMB and CCMB are identical since PM-CCMB only optimizes the algorithm structure of CCMB. Therefore, we do not make a distinction. (a) Alarm. (b) Alarm3. (c) Alarm5. (d) Alarm10. (e) Child. (f) Child3. (g) Child5. (h) Child10. (i) Insurance. (j) Insurance3. (k) Insurance5. (l) Insurance10.

Fig. 5 shows the average *Distance* variation curves of CCMB and other algorithms with respect to the number of samples. From Fig. 5, we can observe that all the algorithms trend to achieve a lower *Distance* with more samples, and CCMB consistently performs better than others with different scales of samples on 9 out of 12 datasets. Since CCMB considers the influence of PCMasking phenomenon on MB discovery, more true positives are detected, which improves the recall, thereby, improves the accuracy of the algorithm. The *Distance* of CCMB is similar with PCMB but also smaller than other algorithms on the Child 10, Insurance 3, and Insurance 10 with large-scale samples (#Variables $>$ 1000). This is because CCMB may introduce some false positives into the MB while finding more true positives, which causes the precision of CCMB to drop slightly, resulting in the accuracy of CCMB similar to PCMB. We also note that CCMB consistently outperforms the other four MB discovery algorithms on all datasets under small-scale samples (#Variables $\leqslant$ 1000), which demonstrates the significant superiority of CCMB in data efficiency. Especially, on the Child dataset, CCMB uses fewer samples (500 samples) to achieve a small *Distance* while other algorithms need 1500–2500 samples to achieve a similar *Distance*. With fewer samples, CCMB can find more accurate MBs, while other algorithms cause more errors due to insufficient samples. When the sample size reaches a certain scale, the existing algorithms can avoid some true positives being ignored, making their accuracy close to our proposed CCMB. Therefore, the significant superiority of CCMB under small-scale samples reflects that our proposed CCMB is more data-efficient. In general, CCMB can significantly improve the accuracy in comparison to the state-of-the-art algorithms.

We recall that PM-CCMB only optimizes the executing process of CCMB. Therefore, in the above experiments, the curves of PM-CCMB and CCMB are identical so that we do not make a distinction. In the following text, we will see the superiority of PM-CCMB over CCMB and other state-of-the-art algorithms.

Time Efficiency: We recorded the CPU time for each dataset in the above experiments. Fig. 6 shows the logarithmic time variation curves of CCMB, PM-CCMB, and other algorithms with respect to the number of samples. From Fig. 6, we can observe that the CPU time of CCMB is slightly higher than PCMB and other algorithms, it is mainly because CCMB introduces the cross-check and complement processes to find more true variables, which is a time-consuming step. To improve the time efficiency, we proposed the PM. Therefore, PM-CCMB consistently performs better than others with different scales of samples on 10 out of 12 datasets, which demonstrates that PM can significantly improve the computational efficiency of CCMB through organizing and storing the intermediate

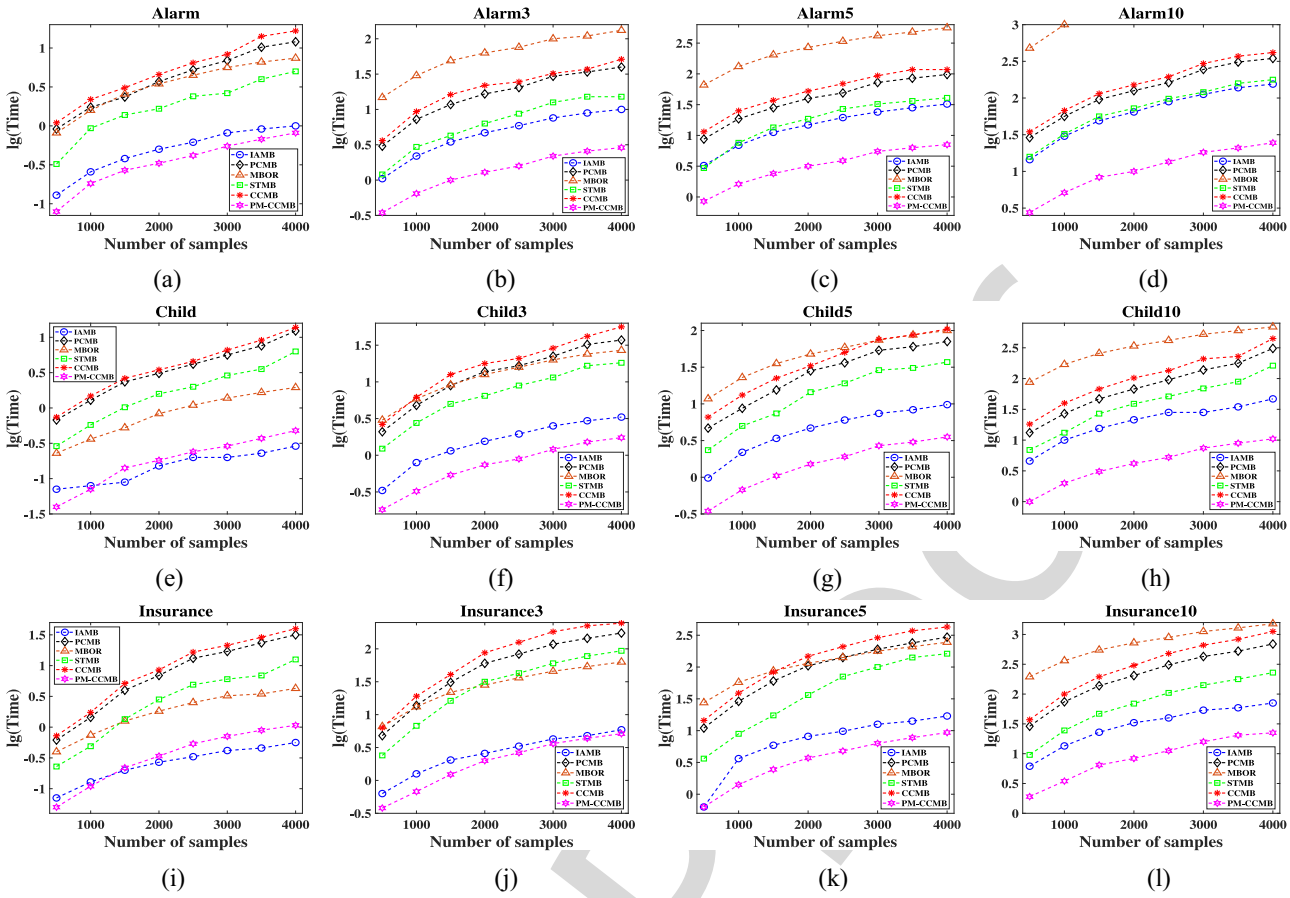


Fig. 6. Results of the MB discovery experiments for the time efficiency of CCMB, PM-CCMB, and other algorithms on the 12 standard BN datasets. (a) Alarm. (b) Alarm3. (c) Alarm5. (d) Alarm10. (e) Child (f). Child3. (g) Child5. (h) Child10. (i) Insurance. (j) Insurance3. (k) Insurance5. (l) Insurance10.

results. Especially, PM-CCMB is even faster than the most time-efficient IAMB. We also note that PM-CCMB is similar with or slightly higher than IAMB on the Child and Insurance. Mainly because, it will take more time to build the PM; then, the PM saves in some datasets with fewer variables (such as Child and Insurance). Even so, PM is still an effective accelerator for MB discovery algorithms. In general, PM-CCMB can significantly improve the time efficiency in spite of its impressive accuracy.

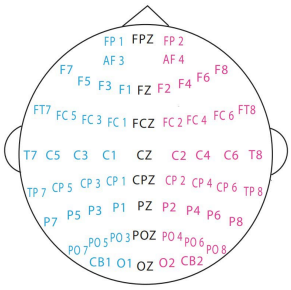


Fig. 7. Layout of 62 channel symmetrical electrodes on the EEG.

C. Emotion Recognition: The Effectiveness for Solving Feature Selection Problem

This section employs the proposed MB discovery algorithms in feature selection task to solve the emotion recognition problem. A newly developed EEG dataset, SEED [32], will be used to evaluate the performance of PM-CCMB. The SEED dataset contains the EEG signals of 15 subjects, which are regarded as 15 datasets. Each subject watched 15 emotional film clips while the EEG signals were recorded by 62-channel symmetrical electrodes (shown in Fig. 7). The features are extracted from five common frequency bands, namely, Delta (1–3 Hz), Theta (4–7 Hz), Alpha (8–13 Hz), Beta (14–30 Hz), and Gamma (31–50 Hz), and each frequency band has 62-channel neural signatures. Therefore, there are 310 features in the SEED. The emotional labels (negative, neutral, and positive) of the film clips are used as the target. Each subject performed the emotion experiments in three separate sessions with an interval of about one week or longer, and each session contains 3394 samples (1120 negative samples, 1104 neutral samples, and 1170 positive samples).

In our experiments, the differential entropy (DE) features are chosen for emotion recognition due to its better discrimination [32]. For each subject, we randomly choose 1000 samples as the training set, and 500 samples as the test set. We compare the emotion recognition and feature selection effectiveness of PM-CCMB with other algorithms on the test set.

We adopt three classifiers, that is, linear SVM, AdaBoost, and Naive Bayes to compute their classification accuracies

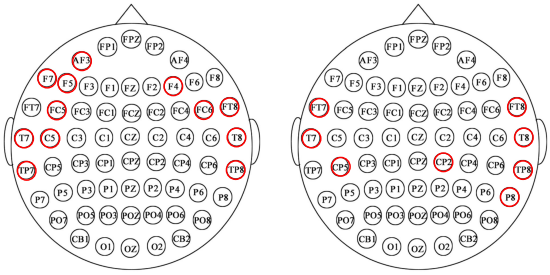
TABLE III
CLASSIFICATION ACCURACIES (MEAN±STANDARD DEVIATION %) ON THE EEG DATASETS ACHIEVED BY USING THREE CLASSIFIERS WITH THE FEATURES SELECTED BY DIFFERENT MB ALGORITHMS AND FEATURE SELECTION ALGORITHMS. THE BEST RESULTS FOR EACH DATASETS ARE HIGHLIGHTED IN BOLDFACE

Classifier	Subject	IAMB	PCMB	MBOR	STMB	FCBF	mRMR	**PM-CCMB**
LibSVM	#1	87.06±5.30	92.87±3.98	87.87±6.22	90.58±4.31	97.37±2.62	92.07±1.25	**99.91±0.22**
	#2	67.04±8.00	94.76±4.47	67.39±8.41	66.33±7.25	96.98±2.17	97.15±0.93	**99.99±0.01**
	#3	67.15±8.44	96.09±1.98	67.15±8.44	96.32±1.94	93.28±6.78	97.89±0.23	**99.99±0.01**
	#4	69.01±5.54	91.45±3.42	69.18±6.45	90.20±1.97	98.56±0.37	96.99±1.36	**100.00±0.00**
	#5	69.14±2.73	96.14±1.18	69.25±2.49	95.54±1.73	98.52±1.10	96.65±1.23	**100.00±0.00**
	#6	72.36±2.50	94.88±3.75	72.36±2.50	91.35±4.62	99.50±0.32	99.89±0.11	**100.00±0.00**
	#7	77.09±7.02	96.05±1.26	80.25±4.29	95.32±2.38	98.98±1.00	95.76±0.43	**99.99±0.01**
	#8	68.01±17.14	94.04±4.49	68.34±17.68	90.52±6.73	99.35±0.61	98.35±0.46	**100.00±0.00**
	#9	70.55±10.20	88.50±4.86	65.79±13.70	84.32±3.74	82.28±7.83	94.74±1.15	**99.90±0.10**
	#10	73.21±4.71	87.86±6.20	73.62±4.70	85.69±5.43	97.56±3.19	97.96±0.53	**99.97±0.02**
	#11	82.74±6.48	91.03±8.41	83.14±6.55	92.06±7.42	99.77±0.35	96.11±1.13	**100.00±0.00**
	#12	65.24±9.98	87.25±3.41	66.35±8.13	85.26±4.74	98.74±0.12	99.72±0.23	**100.00±0.00**
	#13	54.85±4.23	88.39±6.08	55.27±4.48	85.67±5.32	99.28±0.49	98.42±0.59	**100.00±0.00**
	#14	78.94±9.25	81.02±5.10	79.62±10.77	83.44±6.52	92.18±0.97	92.68±0.41	**99.98±0.01**
	#15	58.48±10.44	97.45±3.19	58.70±10.24	98.19±1.14	99.70±0.30	**100.00±0.00**	**100.00±0.00**
	average ranks	6.73	4.00	6.13	4.73	2.80	2.53	1.07
Adaboost	#1	82.15±8.97	84.60±7.64	83.82±6.71	82.12±6.99	86.94±7.25	90.10±1.18	**98.08±1.26**
	#2	58.08±2.99	71.21±5.82	57.94±3.18	69.20±5.97	82.22±5.64	82.54±0.76	**90.77±3.22**
	#3	58.13±3.84	72.04±5.50	58.13±3.84	73.06±5.60	78.67±1.37	**88.37±0.96**	87.31±0.95
	#4	65.53±4.32	69.89±4.34	66.33±3.75	66.35±4.50	90.80±6.10	88.83±1.35	**91.98±3.46**
	#5	67.29±2.70	77.49±2.36	67.03±2.97	78.25±2.26	82.94±8.71	81.77±0.96	**93.13±1.00**
	#6	75.41±3.89	83.38±7.38	75.41±3.89	81.25±5.69	92.17±2.65	94.83±2.62	**95.40±0.66**
	#7	77.44±7.22	90.08±4.26	77.47±7.19	88.25±6.31	91.31±0.37	96.15±0.79	**97.23±0.53**
	#8	66.63±15.76	77.30±8.19	66.63±15.76	76.46±8.37	91.26±2.65	93.69±1.12	**94.07±2.04**
	#9	67.96±8.10	74.08±7.20	66.34±10.17	71.52±8.12	79.12±4.77	90.18±2.25	**94.66±0.72**
	#10	65.60±2.22	67.83±5.81	65.83±2.56	63.26±4.32	82.10±2.96	**91.14±1.16**	86.79±0.97
	#11	78.99±6.00	77.81±4.88	79.11±5.95	76.42±5.23	91.56±1.28	90.53±0.51	**93.66±0.99**
	#12	64.15±8.97	81.22±6.35	67.32±9.63	80.79±4.53	88.74±1.12	89.48±0.92	**91.26±1.35**
	#13	53.67±4.04	75.53±11.70	53.64±3.26	72.56±10.25	90.69±4.51	94.56±1.01	**95.93±1.23**
	#14	61.72±9.43	73.59±5.32	61.28±10.52	75.26±6.48	91.25±0.74	88.95±2.74	**96.73±1.95**
	#15	59.23±13.37	95.45±2.64	59.42±13.19	96.42±2.76	98.53±0.29	97.51±0.63	**99.53±0.46**
	average ranks	6.20	4.40	6.27	5.13	2.67	2.20	1.13
Naive Bayes	#1	80.97±11.52	84.61±8.63	84.09±8.17	85.25±6.69	85.59±2.84	85.78±0.47	**92.73±2.83**
	#2	54.54±6.71	76.57±8.54	54.67±6.55	75.26±9.94	80.37±6.50	80.47±0.62	**85.96±2.79**
	#3	60.09±5.22	76.82±5.31	60.09±5.23	74.36±7.10	75.78±1.61	**85.37±1.12**	78.22±3.41
	#4	69.95±2.36	74.40±3.22	70.61±2.40	72.22±4.85	**92.74±5.58**	90.99±1.22	92.68±1.92
	#5	73.09±3.90	88.81±0.20	74.19±3.43	85.12±2.23	87.90±5.28	90.13±0.81	**95.30±0.96**
	#6	73.76±7.04	85.25±9.90	73.76±7.04	83.25±8.02	95.83±2.34	93.66±1.95	**96.77±1.88**
	#7	82.57±4.76	89.60±3.15	81.30±5.87	84.59±4.77	94.14±1.19	91.24±0.96	**97.30±0.23**
	#8	72.97±17.34	83.77±6.15	73.39±17.96	81.69±3.26	94.06±1.93	86.22±0.42	**100.00±0.00**
	#9	70.72±7.36	79.06±5.26	68.29±9.56	80.15±4.83	78.12±1.86	81.99±1.74	**85.76±1.62**
	#10	69.57±5.54	74.44±7.30	69.66±5.67	75.62±5.65	85.77±5.57	83.30±0.56	**91.01±1.19**
	#11	81.27±5.08	78.22±5.46	81.42±5.10	77.42±6.72	89.59±2.02	81.74±1.16	**91.75±1.08**
	#12	65.39±8.74	77.13±4.32	66.25±9.98	78.32±5.65	87.36±1.03	83.98±1.99	**91.28±1.14**
	#13	60.40±4.48	83.09±9.27	60.22±4.27	84.15±6.74	93.70±4.88	92.56±1.32	**99.20±0.56**
	#14	71.33±7.69	84.13±4.56	73.25±6.98	82.61±3.42	87.35±0.61	85.46±2.85	**89.15±1.36**
	#15	63.30±10.71	97.16±1.95	62.80±11.33	96.99±1.74	97.69±1.23	98.68±0.64	**99.67±0.71**
	average ranks	6.47	4.27	6.27	4.73	2.60	2.53	1.13

achieved by using the selected feature subsets. In this experiment, the regularization parameter of linear SVM is tuned from $\{10^{-2}, \ldots, 10^{2}\}$ by the grid search strategy, and the number of ensemble learning cycles is 50 in the AdaBoost. We first run PM-CCMB and other algorithms to search its MB (or feature subset) on the training samples, then use the selected feature subset to train these classifiers on the training samples and, finally, test the accuracy on the test samples. The experiments are repeated 20 times with different training and test samples, and the average classification accuracy of different classifiers using the feature subsets selected by different algorithms are given in Table III.

From Table III, we note that the classifiers using the features selected by PM-CCMB achieve the highest or highly competitive classification accuracy on most of the subjects, which shows the effectiveness of PM-CCMB to select the relevant features. Specifically, the features selected by PM-CCMB can train a completely correct SVM classifier, while any other algorithms cannot achieve the accuracy. Due to the influence of the PCMasking phenomenon, the dependence between the target and its relevant features may be blocked by other relevant features. Therefore, existing MB discovery algorithms may ignore some of the relevant features resulting in the low accuracy of classification. By detecting the MaskingPCs, our algorithm can find more relevant features and significantly improve the accuracy of features selection. Through comparing with two well-established feature selection algorithms, we can conclude that PM-CCMB can achieve similar or even better feature selection effectiveness.

To give a comprehensive performance comparison between PM-CCMB and others, the Friedman test [36] combining with the *post-hoc* tests at a 95% confidence level is used to make a statistical comparison of different algorithms over multiple datasets. The last row of each classifier in Table III shows the average ranks. Note that the traditional feature selection algorithms have better performances against the causal feature selection algorithms except PM-CCMB. It is mainly because all the causal feature selection algorithms are under



(a) (b)

Fig. 8. Profiles of top-20 features selected by PM-CCMB on the (a) Beta and (b) Gamma frequency bands, which is consistent with the previous findings in [35].

the faithfulness assumption, which cannot be satisfied in the real-world datasets. However, the proposed cross-check and complement processes can not only avoid the PCMasking phenomenon but also make the PM-CCMB more robust in the unfaithful datasets. Therefore, with the three classifiers, the overall classification accuracy of the proposed PM-CCMB is significantly better than all other algorithms, including the traditional feature selection methods.

Compared with the traditional feature selection methods, PM-CCMB can select useful features and simultaneously explain the causality between features and target. To illustrate the interpretability of PM-CCMB, we collect the features selected by PM-CCMB in all datasets and select the top-20 features with the highest frequency, whose positions are illustrated in Fig. 8. As depicted in the figure, the top-20 features are all from the Beta and Gamma frequency bands and located at the lateral temporal area except one of them, which is consistent with the previous findings in [35]. These results indicate that PM-CCMB can effectively select the relevant channels containing discriminative information and simultaneously eliminate irrelevant channels for emotion recognition.

VII. Conclusion

This article introduces the concept of PCMasking to the BN. Based on the PCMasking, we analyze the main reason that causes the lower accuracy in the existing MB discovery algorithms. To detect the MaskingPCs, we propose the cross-check and complement processes, in which the cross-check process can effectively detect the MaskingPCs and the complement process can repair the symmetry (between PC variables) broken by PCMasking phenomenon. On the basis of the cross-check and complement processes, we propose a topology-based MB discovery algorithm, CCMB, to find more true variables. To simultaneously improve the time efficiency, we propose an acceleration methodology for the MB discovery algorithm called PM. Embedding CCMB into PM, we propose PM-CCMB, which can maintain the same accuracy with CCMB and further improve the time efficiency. PM-CCMB is extensively evaluated and compared with the state-of-the-art MB discovery algorithms on different datasets. The results validate that PM-CCMB can shield the influence of PCMasking phenomenon effectively and simultaneously improve the accuracy and time efficiency of MB discovery.

In the future, there are three directions worth further investigation. One is to relax the causal sufficiency assumptions [37] and examine the validity of PM-CCMB when applied to real-world feature selection situations. The second direction is to extend PM-CCMB to BN structure learning, especially for large dimensional problems and small samples. The third direction is to develop a joint feature selection and classification algorithm with some Bayesian methods [38], [39].

References

[1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann, 1998.

[2] P. Spirtes *et al.*, *Causation, Prediction, and Search*. Cambridge, MA, USA: MIT Press, 2000.

[3] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos, "Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation," *J. Mach. Learn. Res.*, vol. 11, no. 1, pp. 171–234, 2010.

[4] K. Yu, L. Liu, and J. Li, *A Unified View of Causal and Non-Causal Feature Selection*. [Online]. Available: https://arxiv.org/abs/1802.05844

[5] D. Margaritis and S. Thrun, "Bayesian network induction via local neighborhoods," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 505–511.

[6] D. Koller and M. Sahami, "Toward optimal feature selection," in *Proc. 13th Int. Conf. Mach. Learn.*, Jul. 1996, pp. 284–292.

[7] I. Tsamardinos, C. F. Aliferis, A. R. Statnikov, and E. Statnikov, "Algorithms for large scale Markov blanket discovery," in *Proc. Florida Artif. Intell. Res. Soc. Conf.*, 2003, pp. 376–380.

[8] I. Tsamardinos, C. F. Aliferis, and A. Statnikov, "Time and sample efficient discovery of Markov blankets and direct causal relations," in *Proc. 9th ACM Int. Conf. Knowl. Disc. Data Min.*, 2003, pp. 673–678.

[9] C. F. Aliferis, I. Tsamardinos, and A. Statnikov, "HITON: A novel Markov blanket algorithm for optimal variable selection," in *Proc. Amer. Med. Informat. Assoc. Annu. Symp.*, 2003, pp. 21–25.

[10] J. M. Peña, R. Nilsson, J. Björkegren, and J. Tegnér, "Towards scalable and data efficient learning of Markov boundaries," *Int. J. Approx. Reason.*, vol. 45, no. 2, pp. 211–232, 2007.

[11] S. R. De Morais and A. Aussem, "A novel scalable and data efficient feature subset selection algorithm," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Disc. Databases*, 2008, pp. 298–312.

[12] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. Boca Raton, FL, USA: CRC Press, 2007.

[13] Y. Wang, X. Li, and R. Ruiz, "Weighted general group lasso for gene selection in cancer classification," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2860–2873, Aug. 2019.

[14] J. Cao, Z. Bu, Y. Wang, H. Yang, J. Jiang, and H. Li, "Detecting prosumer-community groups in smart grids from the multiagent perspective," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 8, pp. 1652–1664, Aug. 2019.

[15] Z. Bu, H.-J. Li, C. Zhang, J. Cao, A. Li, and Y. Shi, "Graph k-means based on leader identification, dynamic game, and opinion dynamics," *IEEE Trans. Knowl. Data Eng.*, to be published. doi: 10.1109/TKDE.2019.2903712.

[16] B. Jiang, C. Li, M. D. Rijke, X. Yao, and H. Chen, "Probabilistic feature selection and classification vector machine," *ACM Trans. Knowl. Disc. Data*, vol. 13, no. 2, pp. 1–27, 2019.

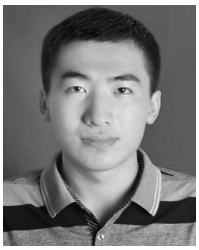
[17] B. Jiang, X. Wu, K. Yu, and H. Chen, "Joint semi-supervised feature selection and classification through Bayesian approach," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 3983–3990.

[18] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[19] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 856–863.

[20] S. Maldonado and R. Weber, "A wrapper method for feature selection using support vector machines," *Inf. Sci.*, vol. 179, no. 13, pp. 2208–2217, 2009.

[21] Y. Mohsenzadeh, H. Sheikhzadeh, and S. Nazari, "Incremental relevance sample-feature machine: A fast marginal likelihood maximization approach for joint feature selection and classification," *Pattern Recognit.*, vol. 60, no. 12, pp. 835–848, 2016.
[22] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.
[23] Y. Xue, B. Xue, and M. Zhang, "Self-adaptive particle swarm optimization for large-scale feature selection in classification," *ACM Trans. Knowl. Disc. Data*, vol. 13, pp. 1–28, Sep. 2019.
[24] I. Guyon *et al.*, "Causal feature selection," in *Computational Methods of Feature Selection*. Boca Raton, FL, USA: Chapman and Hall, 2007, pp. 75–97.
[25] K. Yu, J. Li, W. Ding, and T. D. Le, "Multi-source causal feature selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019, to be published. doi: 10.1109/TPAMI.2019.2908373.
[26] J.-P. Pellet and A. Elisseeff, "Using Markov blankets for causal structure learning," *J. Mach. Learn. Res.*, vol. 9, no. 7, pp. 1295–1342, 2008.
[27] S. Yaramakala and D. Margaritis, "Speculative Markov blanket discovery for optimal feature selection," in *Proc. 5th IEEE Int. Conf. Data Min.*, Houston, TX, USA, 2005, pp. 809–812.
[28] S. Fu and M. C. Desmarais, "Fast Markov blanket discovery algorithm via local learning within single pass," in *Proc. Conf. Can. Soc. Comput. Stud. Intell.*, 2008, pp. 96–107.
[29] T. Gao and Q. Ji, "Efficient Markov blanket discovery and its application," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1169–1179, May 2017.
[30] K. Yu, X. Wu, W. Ding, Y. Mu, and H. Wang, "Markov blanket feature selection using representative sets," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 11, pp. 2775–2788, Nov. 2017.
[31] I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper, "The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks," in *Proc. 2nd Eur. Conf. Artif. Intel. Med.*, 1989, pp. 247–256.
[32] R.-N. Duan, J.-Y. Zhu, and B.-L. Lu, "Differential entropy feature for EEG-based emotion classification," in *Proc. IEEE Int. Conf. Neural Eng.*, San Diego, CA, USA, 2013, pp. 81–84.
[33] W.-L. Zheng and B.-L. Lu, "A multimodal approach to estimating vigilance using EEG and forehead EOG," *J. Neural Eng.*, vol. 14, no. 2, pp. 17–26, 2017.
[34] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, "The max–min hill-climbing Bayesian network structure learning algorithm," *Mach. Learn.*, vol. 65, no. 1, pp. 31–78, 2006.
[35] W.-L. Zheng, J.-Y. Zhu, and B.-L. Lu, "Identifying stable patterns over time for emotion recognition from EEG," *IEEE Trans. Affective Comput.*, vol. 10, no. 3, pp. 417–429, Jul.–Sep. 2019. doi: 10.1109/TAFFC.2017.2712143.
[36] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 1–30, 2006.
[37] K. Yu, L. Liu, J. Li, and H. Chen, "Mining Markov blankets without causal sufficiency," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6333–6347, Dec. 2018.
[38] H. Chen, T. Peter, and X. Yao, "Probabilistic classification vector machines," *IEEE Trans. Neural Netw.*, vol. 20, no. 6, pp. 901–914, Jun. 2009.
[39] H. Chen, T. Peter, and X. Yao, "Efficient probabilistic classification vector machine with incremental basis function selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 356–369, Feb. 2014.



Xingyu Wu received the B.Sc. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2018. He is currently pursuing the M.Sc. degree with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China.

His current research interests include causal learning, feature selection, and multilabel learning.

Bingbing Jiang received the B.Sc. degree from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2014, and the Ph.D. degree in computer science from the University of Science and Technology of China, Hefei, China, in 2019.

His current research interests include Bayesian learning, semisupervised learning, and feature selection.

Kui Yu received the Ph.D. degree in computer science from the Hefei University of Technology, Hefei, China, in 2013.

He is currently a Professor with the Hefei University of Technology. From 2015 to 2018, he was a Research Fellow with the University of South Australia, Adelaide, SA, Australia. From 2013 to 2015, he was a Post-Doctoral Fellow with the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada. His current research interests include causal discovery and machine learning.

Chunyan Miao received the B.S. degree from Shandong University, Jinan, China, in 1988, and the M.S. and Ph.D. degrees from Nanyang Technological University (NTU), Singapore, in 1998 and 2003, respectively.

She is currently a Professor with the School of Computer Science and Engineering, NTU, and the Director of the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY). Her current research interests include infusing intelligent agents into interactive new media (virtual, mixed, mobile, and pervasive media) to create novel experiences and dimensions in game design, interactive narrative, and other real-world agent systems.

Huanhuan Chen (M'09–SM'16) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2004, and the Ph.D. degree in computer science from the University of Birmingham, Birmingham, U.K., in 2008.

He is currently a Full Professor with the School of Computer Science and Technology, USTC. His current research interests include neural networks, Bayesian inference, and evolutionary computation.

Prof. Chen was a recipient of the 2015 International Neural Network Society Young Investigator Award, the 2012 IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation Award, the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award (bestowed in 2011 and only one paper in 2009), and the 2009 British Computer Society Distinguished Dissertations Award. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE.