Practical Markov Boundary Learning without Strong Assumptions

Xingyu Wu[1], **Bingbing Jiang**[2], **Tianhao Wu**[3], **Huanhuan Chen**[1]*

[1]School of Computer Science and Technology, University of Science and Technology of China,
[2]School of Information Science and Engineering, Hangzhou Normal University,
[3]School of Data Science, University of Science and Technology of China,
xingyuwu@mail.ustc.edu.cn, jiangbb@hznu.edu.cn, thwu@mail.ustc.edu.cn, hchen@ustc.edu.cn.

Abstract

Theoretically, the Markov boundary (MB) is the optimal solution for feature selection. However, existing MB learning algorithms often fail to identify some critical features in real-world feature selection tasks, mainly because the strict assumptions of existing algorithms, on either data distribution, variable types, or correctness of criteria, cannot be satisfied in application scenarios. This paper takes further steps toward opening the door to real-world applications for MB. We contribute in particular to a practical MB learning strategy, which can maintain feasibility and effectiveness in real-world data where variables can be numerical or categorical with linear or nonlinear, pairwise or multivariate relationships. Specifically, the equivalence between MB and the minimal conditional covariance operator (CCO) is investigated, which inspires us to design the objective function based on the predictability evaluation of the mapping variables in a reproducing kernel Hilbert space. Based on this, a kernel MB learning algorithm is proposed, where nonlinear multivariate dependence could be considered without extra requirements on data distribution and variable types. Extensive experiments demonstrate the efficacy of these contributions.

Introduction

As a basic concept in statistical machine learning, the Markov boundary (MB) is the smallest variable set that renders the rest of the variables independent of the target (Pearl 1988). Naturally, MB discovery is a principled solution for feature selection as MB carries all predictive information about the class attribute (Guyon, Aliferis et al. 2007). In the causality research, it has been proved that MB consists of the direct causes, direct effects, and other direct causes of direct effects in a causal graph (Aliferis et al. 2010a). Hence, MB could imply the local causal mechanism around a variable, and MB learning algorithms are also alternatively named causal feature selection (Guyon, Aliferis et al. 2007). Existing MB discovery methods can be roughly divided into constraint-based and score-based approaches (Yu et al. 2020a). Constraint-based approaches, as the focus of MB research, learn the MB by identifying the conditional independence (CI) relationships between variable pairs. And the other strategy, score-based approaches, obtains the MB via learning the local Bayesian network structure around the target with a scoring criterion. With the continuous improvement of the methodology, state-of-the-art algorithms achieve ideal performance in a certain type of data satisfying specific assumptions, while few algorithms could remain practical in real-world data due to the violation of some assumptions.

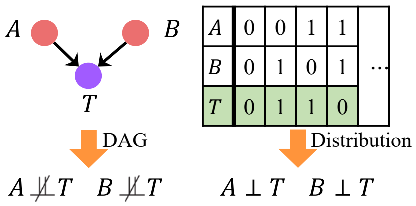


Figure 1: Example of multivariate dependence.

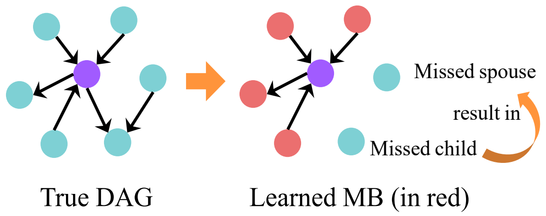


Figure 2: Example of cascading errors in MB discovery.

A basic assumption of MB learning is the faithfulness (Spirtes, Glymour, and Scheines 2000) between underlying distributions and directed acyclic graphs (DAG), which requires that every CI presented in the distribution is entailed by the DAG and Markov condition (Liu and Liu 2018). This assumption defaults that all considered CIs are pairwise relationships while ignoring the multivariate relationships that are ubiquitous but subtle in real-world applications. Multivariate dependence occurs when multiple variables jointly influence the target but none of them is dependent on the target, e.g., logical operation 'XOR' in Figure 1. To remain tractable, existing approaches only consider pairwise relationships and thus suffer from performance degradation as shown in some empirical studies (Yu et al. 2018; Lin and Zhang 2020). Moreover, existing methods further assume the correctness of all CI tests or structure learning process

*Corresponding author: Huanhuan Chen.

(Tsamardinos et al. 2019), which is also hard to be satisfied. The correctness of these techniques is affected by multiple factors such as the number and diversity of instances, and the CI test is additionally limited by the scale of the conditioning set. When the true MB is large and the training samples are insufficient, existing methods easily go astray. Even worse, the results of CI tests or structure learning processes are interrelated and affect subsequent decisions in MB discovery, leading to frequent cascading errors as the example in Figure 2. Existing methods can not deal with situations violating these assumptions, making the efficacy of the discovered MBs significantly reduce in applications. For example in feature selection tasks, MB learning algorithms often ignore true positives (critical features) (Wu et al. 2020b).

In addition to the basic assumptions of MB discovery, criteria or measurements used by each algorithm family usually assume that the data meet certain conditions to ensure theoretical reliability. For constraint-based approaches, the commonly used λ^2-test and G^2-test assume that the variables are discrete (McDonald 2009), whereas the Fishers Z-test is designed for continuous variables with linear relations and additive Gaussian errors (Pena 2008). Analogously, the representative scoring criteria used by score-based approaches have strict requirements on data distributions and variable types. For instance, BDeu (Buntine 1991) and BDe (Heckerman, Geiger, and Chickering 1995) scores are implemented on discrete data, but AIC (Akaike 1974), BIC (Lam and Bacchus 1994), and MDL (Lam and Bacchus 1994) scores are suitable for linear Gaussian parameterization. Other more recently proposed techniques (Sokolova et al. 2014) for CI test and DAG score also have certain requirements on the data to guarantee their robustness on the data satisfying their assumptions. These limitations narrow the applicability of algorithms, making them impractical for real-world data where not all of these assumptions can be fully satisfied.

To promote the application of MB, this paper aims to propose a practical MB learning strategy, which can maintain feasibility and effectiveness in real-world data where variables can be either numerical or categorical with linear or nonlinear, pairwise or multivariate relationships. As we know, kernel-based methods have demonstrated their capacity to handle nonlinear problems in many learning scenarios (Schölkopf et al. 2002). Hence, we intend to adopt this technique to convert the multivariate nonlinear relations into pairwise linear relations by mapping the variables from a Euclidean space to a reproducing kernel Hilbert space (RKHS). Nevertheless, the variable relationships in the high-dimensional RKHS cannot provide a direct reference for MB learning as MB variables ultimately need to be selected from the original space. To make the (in)dependence relationships obtained from the RKHS usable in the original space, the association between the MB and variable dependence in RKHS is constructed by the feat of the conditional covariance operator (CCO) (Fukumizu, Bach, and Jordan 2004). We theoretically prove that the MB is equivalent to the feature subset minimizing the CCO.

However, it is not advisable to learn MB by directly minimizing the CCO since the error in the CCO estimation would make the discovered MB inaccurate. Besides, the minimization of the CCO is a combinatorial optimization problem whose solution space contains variable combinations of all scales (Wu et al. 2020c). As a result, neither brute-force ergodic methods nor suitable optimization approaches can be used. Even so, we observe that the minimum CCO reflects the consistency of conditional variance captured by MB and the entire feature set in kernel space, which inspires us to design the objective function by evaluating the predictability of the mapping variables in the RKHS so that both linear and nonlinear relationships are taken into account without any assumptions about data distributions and variable types. Based on these contributions, a Kernel MB learning algorithm (KMB) is proposed. Before learning the MB, KMB first reorders the candidates by evaluating the variation of CCO, to improve time efficiency. Then, KMB solves the proposed optimization function on a reduced scale and efficiently recovers the MB around the target, whose results are proved to be theoretically equivalent to the direct minimization of the CCO. Extensive experiments validate the superiority of KMB in both synthetic and real-world data.

Background

This section briefly provides some key notations and definitions. We first introduce the background knowledge in MB as well as existing MB learning algorithms, then introduce the CCO in RKHS, which is involved in the proposed MB learning strategy.

Notations

In this paper, common upper-case letters denote random variables and upper-case bold letters denote variable sets. Specifically, $\boldsymbol{X} = \{X_1, X_2, \cdots, X_n\}$ represents the entire variable (or feature) set, and T represents the target variable (or class attribute). $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$ and $\mathbf{t} = \{\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_m\}$ represent the m training samples and their target values. If variables X_i and X_j are (in)dependent conditioned on variable set $\mathbf{Z}$, then we denote the relation as $X_i \not\perp X_j|\mathbf{Z}$ ($X_i \perp X_j|\mathbf{Z}$). The Greek letter $\mathcal{X}$ is used to denote the original space (Euclidean space) of variables, and $\mathcal{H}_{\mathcal{X}}$ represents the mapping feature space from $\mathcal{X}$. Bilinear mapping function $\langle \mathbf{x}, \mathbf{y} \rangle$ denotes the inner product between vectors $\mathbf{x}$ and $\mathbf{y}$. Furthermore, a feature mapping from Euclidean space $\mathcal{X}$ to Hilbert space $\mathcal{H}_{\mathcal{X}}$ is denoted as $\Phi_{\mathcal{X}}(X) : \mathcal{X} \rightarrow \mathcal{H}_{\mathcal{X}}$, where kernel function satisfies $k_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi_{\mathcal{X}}(\mathbf{x}_i), \Phi_{\mathcal{X}}(\mathbf{x}_j) \rangle$. For given samples and a kernel function, the matrix obtained by inputting all samples into the kernel function in pairs is the kernel matrix or Gram matrix $(\boldsymbol{K}_{\mathcal{X}})_{ij} = k_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j)$.

MB and Causal Feature Selection Algorithms

MB is a concept from the structural causal model (Aliferis et al. 2010a), where the MB of a target includes the direct causes (parents), direct effects (children), and other direct causes of direct effects (spouses) in the causal graph (Aliferis et al. 2010a). Hence, MB provides a complete picture of the local causal structure around the target variable, and MB learning could be taken as the first step in causal learning, where the skeleton of the causal model without orientation is constructed by MB (Aliferis et al. 2010a,b; Pellet and Elisseeff 2008; Wu et al. 2022a,b,c). Besides the causal concept, the statistical concept of MB claims that:

Definition 1. *(Markov boundary) For variable subset $\boldsymbol{Z} \subset \boldsymbol{X}$ and target T, if all other variables $X \in \boldsymbol{X} - \boldsymbol{Z}$ are independent of T conditioned on $\boldsymbol{Z}$, and any subsets of $\boldsymbol{Z}$ do not satisfy the condition, then $\boldsymbol{Z}$ is the Markov boundary of T (Tsamardinos and Aliferis 2003).*

It shows that the MB of a variable is deemed to contain the causal information about the variable and possesses the most predictive knowledge when the discussion is in a causal graph. Guyon et. al. applied the MB in feature selection techniques and presented "causal feature selection" (Guyon, Aliferis et al. 2007), and Yu et. al. theoretically provide a unified view of causal and non-causal feature selection methods (Yu, Liu, and Li 2021). In recent years, MB has been widely applied to various complex application scenarios, such as multi-label data (Wu et al. 2020a, 2022e), multi-source data (Yu et al. 2020b), and streaming data (Wu et al. 2023), which achieves more advanced performance than traditional algorithms (Zhong et al. 2021; Jiang et al. 2022a,b). Existing MB discovery approaches are roughly divided into constraint-based and score-based algorithms (Yu et al. 2020a; Wu et al. 2022d). As the focus of MB learning research, constraint-based algorithms (Borboudakis and Tsamardinos 2019; Tsamardinos et al. 2019; Wu et al. 2020b, 2021, 2022f; Guo et al. 2022a,b) account for the majority of MB discovery algorithms, which learn MB through mining the CI relations. As previously mentioned, the MB learning process with a large conditioning set and small-scale training samples easily suffers from incorrect CI tests. Moreover, these CI tests are designed for pair-wise dependence, and can not mine the multivariate relations. Score-based algorithms (Niinimaki and Parviainen 2012; Gao and Ji 2017; Li, Korb, and Allison 2022) adopt score-based structure learning approaches to learn the MB set, where a greedy search method is used to maximize the fitness (scoring function) between the causal graph and training data. Score-based algorithms are far less numerous. Similarly, they consider one more variable in each iteration, and thus they can not identify the multivariate relationships in the distribution.

Conditional Covariance Operator in RKHS

The cross-covariance operator Σ_{YX} can be taken as a generalization of the covariance matrix over the feature maps $\Phi_{\mathcal{X}}(X)$ and $\Phi_{\mathcal{Y}}(Y)$, whose definitions is:

Definition 2. *(Conditional covariance operator) Let $(\mathcal{H}, k)$ denote an RKHS $\mathcal{H}$ with a positive definite kernel k on $\mathbb{R}$, then the cross-covariance operator for the random variable pair (X, Y) is the mapping from $\mathcal{H}_{\mathcal{X}}$ to $\mathcal{H}_{\mathcal{Y}}$, and for each $f \in \mathcal{H}_{\mathcal{X}}$ and $g \in \mathcal{H}_{\mathcal{Y}}$: $\langle g, \Sigma_{YX} f \rangle_{\mathcal{H}_Y} = Cov[f(X), g(Y)]$. The conditional covariance operator on $\mathcal{H}_{\mathcal{X}} \to \mathcal{H}_{\mathcal{Y}}$ is defined as $\Sigma_{YX|Z} = \Sigma_{YX} - \Sigma_{YZ}\Sigma_{ZZ}^{-1}\Sigma_{ZX}$.*

The independence relationship $X \perp Y$ in the probability distribution can be described in the RKHS with the participation of the cross-covariance operator, i.e., $\Sigma_{YX} = 0 \Leftrightarrow Cov[f(X), g(Y)] = 0$. Nevertheless, the correlation cannot reflect relationships closer to the essence, such as causality that MB focuses on. Hence, CCO is defined to describe the CI, which is employed to learn MB in this paper.

MB Representation in RKHS

Since the MB is defined in the original space, kernel MB learning aims to adopt the multivariate relationships contained in the RKHS to find the MB variables in the original space, instead of directly selecting the MB from the mapping features in the RKHS. Then, the CI between variables evaluated in RKHS is the premise of kernel-based MB learning. This section will utilize the CCO to capture these relations in RKHS, and analyze the connection between MB and CCO to propose a preliminary kernel-based MB learning idea.

According to Definition 1, the variable subset $\boldsymbol{Z} \subset \boldsymbol{X}$ is an MB of the target T if it satisfies two conditions: (1) independence: $\boldsymbol{X} - \boldsymbol{Z} \perp T|\boldsymbol{Z}$ in the distribution, and (2) minimality: $\boldsymbol{Z}$ is the minimal set satisfying (1). In the following, we first give a property of MB described with CCO.

Theorem 1. *Let $\boldsymbol{MB} \subset \boldsymbol{X}$ denote the MB of variable $T \in \boldsymbol{X}$. Mapping target space $\mathcal{T}$ and feature space $\mathcal{X}$ into the RKHS $\mathcal{H}_{\mathcal{T}}$ and $\mathcal{H}_{\mathcal{X}}$ with two measurable positive definite kernels, then for $\forall g \in \mathcal{H}_{\mathcal{T}}$, $\langle g, \Sigma_{TT|\boldsymbol{MB}} g \rangle_{\mathcal{H}_{\mathcal{T}}} = \langle g, \Sigma_{TT|\boldsymbol{X}} g \rangle_{\mathcal{H}_{\mathcal{T}}}$.*

The detailed proof is provided in the Appendix[1]. Theorem 1 describes that the CCO on the MB $\Sigma_{TT|\boldsymbol{MB}}$ is equal to the CCO on the entire variable set $\Sigma_{TT|\boldsymbol{X}}$. According to the characterization of the residual error of $g \in \mathcal{H}_Y$ by the CCO, we can observe a monotonic partial order relation on $\Sigma_{TT|\boldsymbol{Z}}$ as the scale of $\boldsymbol{Z}$ increases. This monotonicity combined with Theorem 1 leads us to solve MB by maximizing or minimizing CCO. Herein, we follow this thinking to prove the relationship between MB and CCO, as shown in Theorem 2.

Theorem 2. *The MB $\boldsymbol{MB} \subset \boldsymbol{X}$ of variable $T \in \boldsymbol{X}$ is the variable subset $\boldsymbol{Z}$ minimizing $\Sigma_{TT|\boldsymbol{Z}}$.*

The proof is provided in the Appendix, according to which the independence and minimality property is proved to be satisfied when a feature subset could minimize the CCO. Based on Theorem 2, the MB learning problem can be transformed to the minimization of the $\Sigma_{TT|\boldsymbol{Z}}$, and the $\boldsymbol{Z}$ minimizing $\Sigma_{TT|\boldsymbol{Z}}$ is the MB of T. Formally,

$$MB(T) = \arg\min_{\boldsymbol{Z} \subset \boldsymbol{X}} \ \Sigma_{TT|\boldsymbol{Z}} \tag{1}$$

where the value of $\Sigma_{TT|\boldsymbol{Z}}$ could be estimated by its sum or product of the eigenvalues, corresponding to the determinant and trace, respectively. When the matrix has multiple small eigenvalues, the calculated determinant is also small and may exceed the range of precision. Hence, trace-based measurement has the advantage of yielding relatively reliable results with a simple theoretical analysis.

Based on Eq. (1), we propose an initial kernel MB learning algorithm (KMB$-$) with a simple idea in Algorithm 1. KMB$-$ directly solves the optimization problem and finds the variable subset $\boldsymbol{Z}$ minimizing the $\Sigma_{TT|\boldsymbol{Z}}$. According to Theorem 2, KMB- is theoretically correct and the learned

[1] Appendix: http://home.ustc.edu.cn/~xingyuwu/AAAI23.pdf

Algorithm 1: The KMB$-$ Algorithm.

```
Input: Target $T$, features set $\boldsymbol{X}$.
Find the variable subset $\boldsymbol{Z}$ minimizing Eq. (1)
$\boldsymbol{MB} \leftarrow \boldsymbol{Z}$
Output: MB set $\boldsymbol{MB}$.
```

$\boldsymbol{Z}$ is just the MB of T. However, as an immature approach, KMB$-$ does not possess scalability. Firstly, directly estimating the $\Sigma_{TT|\boldsymbol{Z}}$ is susceptible to estimation errors, leading to unfaithful results. Additionally, Eq. (1) is hard to solve, which can not be solved by commonly used optimization algorithms such as gradient descent. Therefore, KMB$-$ is only a theoretically feasible solution, and we need to further explore another practical method.

Kernel MB Learning

Since directly minimizing the CCO may lead to suboptimal solutions due to inaccurate estimation of CCO, this section exploits another more effective solution for MB discovery based on the above analyses. According to Theorem 1, we can reach the following corollary as $\Sigma_{TT|\boldsymbol{X}} = \Sigma_{TT|\boldsymbol{Z}}$:

Corollary 1. *Let $\boldsymbol{MB} \subset \boldsymbol{X}$ denote the MB of variable $T \in \boldsymbol{X}$. Mapping target space $\mathcal{T}$ and feature space $\mathcal{X}$ into the RKHS $\mathcal{H}_\mathcal{T}$ and $\mathcal{H}_\mathcal{X}$ with two measurable positive definite kernels, then for $\forall g \in \mathcal{H}_\mathcal{T}$, $E_{\boldsymbol{MB}}[D_{T|\boldsymbol{MB}}[g(T)|\boldsymbol{MB}]] = E_{\boldsymbol{X}}[D_{T|\boldsymbol{X}}[g(T)|\boldsymbol{X}]]$.*

Corollary 1 demonstrates that, to predict any-dimensional vector $\Phi_\mathcal{T}(T)$ in $\mathcal{H}_\mathcal{T}$, the predictive model constructed on $\Phi_\mathcal{X}(\boldsymbol{MB})$ could theoretically achieve the same performance as that on all features. In the following, we evaluate the likelihood of a subset to be MB by measuring its predictive ability in RKHS. Since kernel methods have transformed nonlinear problems into linear ones, we directly model the mapping target in the mapping feature space employing the most commonly used linear model, whose general formulation is:

$$\Phi_\mathcal{T}(T) = \Phi_\mathcal{X}(\boldsymbol{Z})\boldsymbol{W} + \varepsilon \tag{2}$$

where $\boldsymbol{W}$ and ε represent the feature projection matrix and the regression error vector, respectively. Note that the dimensions of the mapped vectors $\Phi_\mathcal{T}(T)$ and $\Phi_\mathcal{X}(\boldsymbol{Z})$ are unknown, even infinite, it is difficult to calculate the predictability of the model in Eq. (2) to measure the candidate subset. The main challenge comes from the unknown dimension of $\Phi_\mathcal{T}(T)$, while the dimension problem of $\Phi_\mathcal{X}(\boldsymbol{Z})$ will be finally eliminated by calculating the inner product between the mapping vectors, which will be shown in the following analyses. According to the classical representer theorem (Schölkopf, Herbrich, and Smola 2001; Wahba 1990), a target could be represented with the linear combination of the mapped training samples. Hence, the information in the $\Phi_\mathcal{T}(T)$ is equivalent to the information in vector function $\mathbf{f}_k(\mathbf{t})$ defined as:

$$\mathbf{f}_k(\mathbf{t}) = \begin{bmatrix} \langle \Phi_\mathcal{T}(\mathbf{t}_1), \Phi_\mathcal{T}(\mathbf{t}) \rangle_{\mathcal{H}_\mathcal{T}} \\ \langle \Phi_\mathcal{T}(\mathbf{t}_2), \Phi_\mathcal{T}(\mathbf{t}) \rangle_{\mathcal{H}_\mathcal{T}} \\ \vdots \\ \langle \Phi_\mathcal{T}(\mathbf{t}_m), \Phi_\mathcal{T}(\mathbf{t}) \rangle_{\mathcal{H}_\mathcal{T}} \end{bmatrix}^T = \begin{bmatrix} k_\mathcal{T}(\mathbf{t}_1, \mathbf{t}) \\ k_\mathcal{T}(\mathbf{t}_2, \mathbf{t}) \\ \vdots \\ k_\mathcal{T}(\mathbf{t}_m, \mathbf{t}) \end{bmatrix}^T \tag{3}$$

In this way, linear model in Eq. (2) can be materialized as:

$$\mathbf{f}_k(\mathbf{t}) = \Phi_\mathcal{X}(\boldsymbol{Z})\boldsymbol{W}_f + \boldsymbol{\varepsilon}_f \tag{4}$$

where $\boldsymbol{W}_f \in \mathbb{R}^{|\Phi_\mathcal{T}(\mathbf{t})| \times m}$ and $\boldsymbol{\varepsilon}_f \in \mathbb{R}^{m \times 1}$. Generally, the model in Eq (4) contains m independent sub-models, and the i-th sub-model is recorded as: $k_\mathcal{T}(\mathbf{t}_i, \mathbf{t}) = \Phi_\mathcal{X}(\boldsymbol{Z})(\boldsymbol{W}_f)_i + (\boldsymbol{\varepsilon}_f)_i$, where $(\boldsymbol{W}_f)_i$ denotes the i-th column in $\boldsymbol{W}_f$ and $(\boldsymbol{\varepsilon}_f)_i$ denotes the i-th element in $\boldsymbol{\varepsilon}_f$. Herein, the input to the MB learning problem is also adopted to train the model in Eq. (4). Each sample corresponds to m results for m sub-models, constructing a matrix:

$$\left[\mathbf{f}_k(\mathbf{t}_1)^T, \mathbf{f}_k(\mathbf{t}_2)^T, \cdots, \mathbf{f}_k(\mathbf{t}_m)^T\right]^T = \boldsymbol{K}_\mathcal{T} \tag{5}$$

which is a symmetric matrix. Then, the objective function is formulated as follows:

$$\min_{\boldsymbol{W}_f} \sum_{i,j} \|(\boldsymbol{K}_\mathcal{T})_{ji} - \Phi_\mathcal{X}(\boldsymbol{Z})_{\mathbf{x}_j}(\boldsymbol{W}_f)_i\|^2 + \lambda(\boldsymbol{W}_f)_i^T(\boldsymbol{W}_f)_i \tag{6}$$

where the $\Phi_\mathcal{X}(\boldsymbol{Z})_{\mathbf{x}_j}$ is the value of mapping features of $\boldsymbol{Z}$ on j-th sample $\mathbf{x}_j$. Since the mapping feature space could be very high-dimensional, we introduce a regularization term $\lambda(\boldsymbol{W}_f)_i^T(\boldsymbol{W}_f)_i$. Obviously, the prediction of $k_\mathcal{T}(\mathbf{t}_i, \mathbf{t})$ for different i is independent of each other, and the optimization problem could be solved by focusing on each sub-model.

Herein, we consider its i-th column $(\boldsymbol{K}_\mathcal{T})_i$ for the i-th sub-model. The solution of $(\boldsymbol{W}_f)_i$ can be obtained via the normal equation, which is directly given as:

$$(\boldsymbol{W}_f)_i = \left(\Phi_\mathcal{X}(\boldsymbol{Z})^T\Phi_\mathcal{X}(\boldsymbol{Z}) + \lambda I_d\right)^{-1}\Phi_\mathcal{X}(\boldsymbol{Z})^T(\boldsymbol{K}_\mathcal{T})_i \tag{7}$$

where $d \in (0, +\infty)$ is the dimension of each mapping sample, $\Phi_\mathcal{X}(\boldsymbol{Z}) \in \mathbb{R}^{m \times d}$ includes all mapping samples in RKHS, and I_d is a $d \times d$ identify matrix.

As shown in Eq. (7), some matrices with unknown dimensions are involved in the calculation of $\boldsymbol{W}_f$, it is still challenging to measure the predictability of the prediction model. To solve this problem, some transformations are made to Eq. (7). According to the matrix inversion lemma (Greub 2012) on block matrix $\begin{bmatrix}\boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D}\end{bmatrix}$, if $\boldsymbol{A}$ and $\boldsymbol{D}$ are square and invertible matrices, then the following identity holds: $\left(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{B}\left(\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}\right)^{-1}$. Let $\boldsymbol{A} = \lambda I_d$, $\boldsymbol{B} = \Phi_\mathcal{X}(\boldsymbol{Z})^T$, $\boldsymbol{C} = -\Phi_\mathcal{X}(\boldsymbol{Z})$, and $\boldsymbol{D} = I_m$, and substitute them into Eq. (7):

$$\begin{aligned}(\boldsymbol{W}_f)_i &= \frac{1}{\lambda}\Phi_\mathcal{X}(\boldsymbol{Z})^T\left(I_m + \frac{1}{\lambda}\Phi_\mathcal{X}(\boldsymbol{Z})\Phi_\mathcal{X}(\boldsymbol{Z})^T\right)^{-1}(\boldsymbol{K}_\mathcal{T})_i \\ &= \Phi_\mathcal{X}(\boldsymbol{Z})^T\left(\lambda I_m + \Phi_\mathcal{X}(\boldsymbol{Z})\Phi_\mathcal{X}(\boldsymbol{Z})^T\right)^{-1}(\boldsymbol{K}_\mathcal{T})_i\end{aligned} \tag{8}$$

Different from the unknown dimension of $\Phi_{\mathcal{X}}(\mathbf{Z})^T\Phi_{\mathcal{X}}(\mathbf{Z})$ in Eq. (7), $\Phi_{\mathcal{X}}(\mathbf{Z})\Phi_{\mathcal{X}}(\mathbf{Z})^T \in \mathbb{R}^{m\times m}$ could be transformed into the form of a kernel matrix. Inputting test sample $\boldsymbol{x}$, the estimated $k_{\mathcal{T}}(\hat{\mathbf{t}}_i, \mathbf{t})$ is calculated as follows:

$$k_{\mathcal{T}}(\hat{\mathbf{t}}_i, \mathbf{t}) \\ = \mathbf{f}_k(\mathbf{t}_i)\left(\boldsymbol{K}_{\mathcal{X}}(\mathbf{Z}) + \lambda I_m\right)^{-1} \begin{bmatrix} \langle \Phi_{\mathcal{X}}(\mathbf{Z})_{\mathbf{x}_1}, \Phi_{\mathcal{X}}(\mathbf{Z})_{\boldsymbol{x}}\rangle_{\mathcal{H}_X} \\ \langle \Phi_{\mathcal{X}}(\mathbf{Z})_{\mathbf{x}_2}, \Phi_{\mathcal{X}}(\mathbf{Z})_{\boldsymbol{x}}\rangle_{\mathcal{H}_X} \\ \vdots \\ \langle \Phi_{\mathcal{X}}(\mathbf{Z})_{\mathbf{x}_m}, \Phi_{\mathcal{X}}(\mathbf{Z})_{\boldsymbol{x}}\rangle_{\mathcal{H}_X} \end{bmatrix} \tag{9}$$

where the result corresponds to the i-th mapping target of the sample $\boldsymbol{x}$.

To evaluate the performance of this model, training samples are used to learn the $\boldsymbol{W}_f$ and test samples are input to the learned model. The estimated values of all mapping targets on the j-th sample are obtained as:

$$\mathbf{f}_k(\hat{\mathbf{t}}_j)^T = \boldsymbol{K}_{\mathcal{T}}\left(\boldsymbol{K}_{\mathcal{X}}(\mathbf{Z}) + \lambda I_m\right)^{-1}(\boldsymbol{K}_{\mathcal{X}}(\tilde{\mathbf{Z}})^T)_j \tag{10}$$

where the $(\boldsymbol{K}_{\mathcal{X}}(\tilde{\mathbf{Z}})^T)_j = \Phi_{\mathcal{X}}(\mathbf{Z})\Phi_{\mathcal{X}}(\mathbf{Z})^T$ is calculated with mapping samples of training samples and test samples, respectively, and the $(\boldsymbol{K}_{\mathcal{X}}(\mathbf{Z}))_j$ denotes the j-the column of the Gram matrix of $\mathbf{Z}$. Since the $\mathbf{f}_k(\mathbf{t}_j)$ constructs the Gram matrix of target T and it is a symmetric matrix, the predicted results can be represented as:

$$\hat{\boldsymbol{K}_{\mathcal{T}}} = \boldsymbol{K}_{\mathcal{T}}\left(\boldsymbol{K}_{\mathcal{X}}(\mathbf{Z}) + \lambda I_m\right)^{-1}\boldsymbol{K}_{\mathcal{X}}(\tilde{\mathbf{Z}})^T \tag{11}$$

The performance of regression in Eq. (2) can reflect the congruent relationship between variable subsets and potential MB by Corollary 1, so the MB learning process can be transformed into the process of selecting the best regression model. We can measure the predictability of the model according to the trace of its cost matrix, i.e.:

$$\arg\min_{\boldsymbol{Z}\subset\boldsymbol{X}} Tr\left[(\tilde{\boldsymbol{K}_{\mathcal{T}}} - \hat{\boldsymbol{K}_{\mathcal{T}}})(\tilde{\boldsymbol{K}_{\mathcal{T}}} - \hat{\boldsymbol{K}_{\mathcal{T}}})^T\right] \tag{12}$$

where each element in the $\tilde{\boldsymbol{K}_{\mathcal{T}}}$ is calculated using a kernel function with a training sample and a test sample. To avoid overfitting when learning $\boldsymbol{W}_f$, K-fold cross-validation could be used here to measure the predictability of the regression model. Practically, the input data can be split into K blocks, and $K-1$ blocks are used to train the model and 1 block to test its performance, until K different results have been recorded, in which the average result is calculated. Moreover, Eq. (12) can be further rephrased so that some continuous optimization algorithms could be employed to solve it. Firstly, the parameter $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\} \in \{0,1\}^n$ is introduced to replace the decision variable $\boldsymbol{Z}$ and describes whether a feature is an MB variable, then $\boldsymbol{Z} = \Omega \odot \boldsymbol{X}$ describes the Hadamard product between Ω and $\boldsymbol{X}$, and $(\boldsymbol{K}_{\mathcal{X}}(\mathbf{Z}))_{ij} = k_{\mathcal{X}}(\Omega\odot\mathbf{x}_i, \Omega\odot\mathbf{x}_j)$. To use gradient descent for optimization, the domain of Ω can be relaxed to $\Omega \in [0,1]^n$ and a certain number of variables with the largest values are selected to the results.

Based on Eq. (12), a kernel MB learning (KMB) algorithm is presented in Algorithm 2. If the noise term follows a Gaussian distribution, the KMB and KMB$-$ can be proved to be equivalent, as shown in Theorem 3, whose detailed proof is provided in the Appendix.

Algorithm 2: The KMB Algorithm.

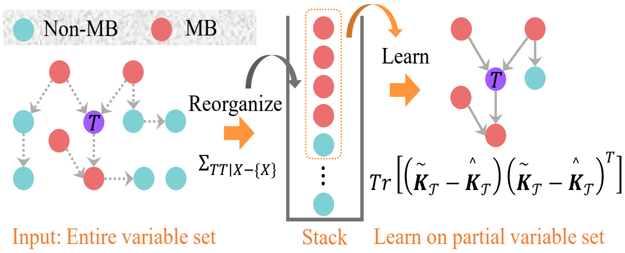
```
1: Input: Dataset $\mathbb{D}$, target $T$, features set $\boldsymbol{X}$, $k$.
2: Initialize $\boldsymbol{X}_s, \boldsymbol{MB}_T, \boldsymbol{DelX} \leftarrow \varnothing$
3: while $\boldsymbol{X} \neq \varnothing$ do
4:   $X \leftarrow \arg\min_{X\in\boldsymbol{X}} \Sigma_{TT|\boldsymbol{X}-\{X\}}$
5:   $\boldsymbol{X}_s \leftarrow \boldsymbol{X}_s \cup \{X\}$ using a stack, $\boldsymbol{X} \leftarrow \boldsymbol{X} - \{X\}$.
6: end while
7: for each $X \in \boldsymbol{X}_s - \boldsymbol{MB}_T - \boldsymbol{DelX} - \{T\}$ do
8:   $\boldsymbol{MB}_T \leftarrow$ Solve Eq.(12) on $\boldsymbol{MB}_T \cup \{X, T\}$.
9:   if $\boldsymbol{MB}_T$ does not change then
10:     $\boldsymbol{MB}_T \leftarrow$ Solve Eq.(12) on $\boldsymbol{MB}_T \cup \boldsymbol{DelX} \cup \{X, T\}$.
11:     $\boldsymbol{DelX} \leftarrow \boldsymbol{DelX} \cup \{X\} - \boldsymbol{MB}_T$
12:   end if
13:   If the $\boldsymbol{MB}_T$ does not change in $k$ steps, break.
14: end for
15: Output: MB set $\boldsymbol{MB}_T$.
```

Theorem 3. *If the noise term follows a Gaussian distribution, then KMB and KMB$-$ are equivalent.*

KMB does not directly solve the optimization problem on the entire variable set $\boldsymbol{X}$ but employs a stepwise optimization process. As shown in Figure 3, KMB roughly includes two steps: 1) Lines 3-6: Reorganize the variables in $\boldsymbol{X}$; and 2) Lines 7-14: Identify MB set $\boldsymbol{MB}_T$.

Step (1) evaluates the likelihood of each variable to be an MB variable according to the Eq. (1) and the monotonic partial ordering relation $\Sigma_{TT|\boldsymbol{X}} \preceq \Sigma_{TT|\mathbf{Z}}$ proved in Theorem 2. In other words, when a variable X is eliminated from the variable set, the smaller the value of $\Sigma_{TT|\boldsymbol{X}-\{X\}}$, the less likely X is the MB variable (line 4). By iteratively eliminating those variables that are irrelevant to T, the MB variables would be always included in the remaining variables, in which the multivariate effect on T is obviously considered in the backward elimination. As a result, the inverted order represents the relative importance of each variable, and thus KMB uses a data structure, stack $\boldsymbol{X}_s$ in line 5, to reorganize these ordered variables. By Step (1), only the first several variables in $\boldsymbol{X}_s$ need to be considered, significantly improving efficiency within the allowable precision range.

Step (2) learns the MB. In each iteration, KMB sequentially pops a variable from stack $\boldsymbol{X}_s$ (line 7) and learns the $\boldsymbol{MB}_T$ by repeatedly optimizing Eq. (12) on current $\boldsymbol{MB}_T \cup \{X, T\}$ (line 8). The introduced decision variable Ω could be solved by gradient descent as well as some commonly-used greedy search approaches. Since each iteration only considers one more variable, the strategy may lead to some multivariate relationships being ignored even though the optimization function has measured multivariate nonlinear relationships. When there exist other variables affecting the T together with the popped X, line 8 may mistake it for an irrelevant variable. To make up for this defect, line 10 does not directly discard it, but looks back at the previously invalid variables to consider the possible joint effects of these variables on the target T. If the possible synactic



Non-MB
MB
T
Reorganize
$\Sigma_{TT|X-\{X\}}$
Learn
T
$Tr\left[\left(\tilde{K}_T-\hat{K}_T\right)\left(\tilde{K}_T-\hat{K}_T\right)^T\right]$
Input: Entire variable set
Stack
Learn on partial variable set

Figure 3: Diagram of the KMB algorithm.

variables are not included in ***DelX***, then X will be added into ***DelX*** to wait for its possible synactic variables in line 11. Actually, in the implementation of KMB, this procedure (lines 9-12) does not need to be repeated in each iteration, but only considers the multivariate relations at one time after the algorithm has identified all pairwise relations, which will significantly improve the time efficiency. In line 13, a monitor is arranged at the end of each iteration with a predetermined parameter k. When the monitor finds that the $\boldsymbol{MB}_T$ has not changed in the last k iterations, then KMB will terminate the search of the $\boldsymbol{MB}_T$. This is because the variables in $\boldsymbol{X}_s$ are sorted according to their likelihood to be an MB variable, and thus MB variables are mostly found in the first few iterations of KMB.

Experiments

We first verify the effectiveness of KMB on synthetic datasets with foregone (in)dependence relationships to verify its superiority on mixed data. Then, KMB is conducted on real-world datasets to demonstrate the superiority against traditional feature selection algorithms and causal feature selection algorithms on the feature selection task.

MB Discovery on Mixed Data

The main superiority of KMB over existing MB learning algorithms is that it breaks through the limitations of all strong assumptions about data distributions and variable types. Therefore, the simulation data, sampled from synthetic Bayesian networks, are mainly to verify the robustness of KMB in various experimental environments, where the underlying data distributions and variable types are exactly known so that comparing methods can be evaluated in a controlled setting. Some classic or state-of-the-art algorithms in each type are chosen to compare with KMB, including four constraint-based algorithms IAMB (Tsamardinos et al. 2003), HITON-MB (Aliferis, Tsamardinos, and Statnikov 2003), CCMB (Wu et al. 2020b), and SRMB (Wu et al. 2021) and two score-based algorithms SLL (Niinimaki and Parviainen 2012) and S^{2}TMB (Gao and Ji 2017).

To validate the effectiveness of KMB in different experimental environments, each dataset is set with different controlled parameters: (1) proportion of the continuous variables p_c; (2) proportion of the nonlinear relationships (including multivariate relationships) p_n. The simulated Bayesian network includes 50 variables and 1000 training samples, which are the same in all experiment groups.

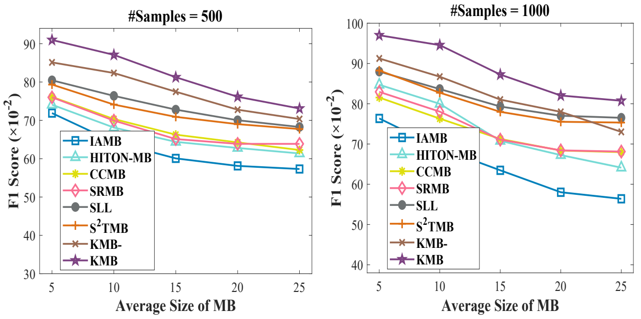


Figure 4: The F_1 score of MB discovery on synthetic mixed data with different MB scale.

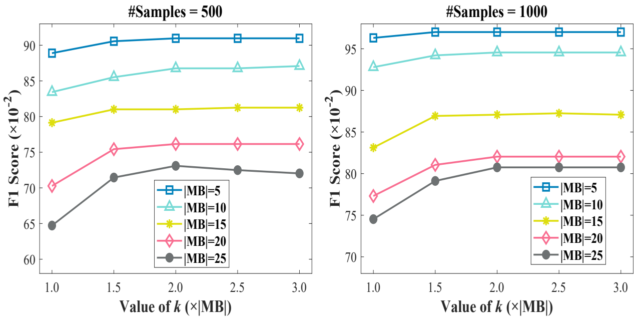


Figure 5: The F_1 score of KMB achieved by different k.

The Bayesian network is constructed by randomly choosing the MB variables for each target. Firstly, the variable type should be determined, where each variable becomes a continuous variable with probability p_c, followed by the random initialization of the variables without parent nodes. After that, the values of the remaining nodes are obtained through the operation of their parent nodes, and the nonlinear functions are selected according to the probability p_n. Note that linear operations on discrete variables will be implemented first since they can only be obtained by other discrete variables, the remaining operations are determined later. The experiment consists of two groups, and each group keeps one of the p_c and p_n invariant but changes the other, to show the performance of KMB and other comparing algorithms in different cases. Tables 1 and 2 compare the average F_1 score of discovered MB with respect to the proportion of nonlinear relationships and continuous variables in data, respectively. The details of comparing algorithms and experimental processes are provided in the Appendix.

Performance comparison with controlled settings: With the increase in the proportions of nonlinear relationships and continuous variables, existing algorithms witness a downward trend, albeit to widely varying degrees. While KMB remains steady regardless of the experimental environments and consistently achieves better performance, which shows the robustness of KMB. From the performance of other comparing algorithms and their variance changes, the impact of the nonlinear relationship on the performance of the algorithm is greater than that of continuous variables,

p_c	p_n	$p_n = 0.00$	$p_n = 0.25$	$p_n = 0.50$	$p_n = 0.75$	$p_n = 1.00$	$lg(t)$
$p_c = 0.00$	IAMB	83.74±0.98	77.25±1.32	71.44±1.97	63.13±3.55	56.37±3.93	0.26
	HITON-MB	90.74±0.74	85.36±0.99	78.73±1.56	68.12±2.03	61.39±2.58	1.16
	CCMB	93.06±0.69	86.99±1.21	77.04±1.99	70.57±2.85	63.31±3.79	1.26
	SRMB	92.94±0.71	87.02±1.26	78.19±2.09	70.27± 3.04	62.02±4.04	1.26
	SLL	92.98±1.02	88.45±1.97	84.73±2.99	79.62±4.05	73.10±5.02	3.88
	S^{2}TMB	91.72±1.14	86.01±2.05	81.39±3.13	75.42±4.20	69.98±5.31	3.19
	KMB−	93.12±2.06	91.08±2.12	87.94±2.64	85.03±3.07	82.73±3.05	1.36
	KMB	96.74±0.57	96.46±0.59	96.31±0.76	95.79±0.69	95.10±0.63	1.64
$p_c = 1.00$	IAMB	77.12±1.14	71.25±2.15	64.73±3.42	54.10±4.01	43.06±4.73	0.25
	HITON-MB	83.58±0.97	76.03±1.32	68.95±1.95	56.12±2.81	46.80±3.84	1.15
	CCMB	84.04±0.75	78.09±1.63	70.25±2.25	59.18±3.74	50.68±4.81	1.26
	SRMB	83.97±0.76	77.37±1.68	68.94±2.25	59.54±3.81	51.13±4.90	1.24
	SLL	81.69±1.25	77.45±2.70	72.10±3.81	68.43±4.94	62.77±5.73	3.88
	S^{2}TMB	82.04±1.27	78.09±2.81	71.16±3.94	65.79±5.03	60.15±6.25	3.23
	KMB−	92.18±3.01	90.09±3.14	86.12±3.62	83.97±3.65	80.59±3.93	1.37
	KMB	96.84±0.61	96.39±0.67	96.02±0.79	95.25±0.83	94.47±0.81	1.66

Table 1: Average F_1 score ($\times 10^{-2}$) of discovered MB with respect to the proportion of nonlinear relationships.

p_n	p_c	$p_c = 0.00$	$p_c = 0.25$	$p_c = 0.50$	$p_c = 0.75$	$p_c = 1.00$	$lg(t)$
$p_n = 0.00$	IAMB	83.74±0.98	82.37±1.03	81.06±0.99	79.25±1.10	77.12±1.14	0.37
	HITON-MB	90.74±0.74	89.25±0.76	87.31±0.87	84.97±0.83	83.58±0.97	1.40
	CCMB	93.06±0.69	91.12±0.69	88.95±0.74	86.43±0.81	84.04±0.75	1.55
	SRMB	92.94±0.71	90.93±0.70	88.19±0.75	86.52±0.73	83.97±0.76	1.56
	SLL	92.98±1.02	90.15±1.04	87.03±1.15	84.97±1.12	81.69±1.25	5.22
	S^{2}TMB	91.72±1.14	90.06±1.19	87.99±1.21	85.42±1.16	82.04±1.27	4.93
	KMB−	93.12±2.06	93.14±2.57	92.85±2.31	92.67±2.65	92.18±3.01	1.36
	KMB	96.74±0.57	96.37±0.37	96.62±0.54	96.45±0.59	96.84±0.61	1.61
$p_n = 1.00$	IAMB	56.37±3.93	55.20±3.99	51.58±4.52	47.11±4.58	43.06±4.73	0.39
	HITON-MB	61.39±2.58	58.12±2.76	54.97±3.02	50.01±3.47	46.80±3.84	1.41
	CCMB	63.31±3.79	60.25±3.96	56.09±4.53	52.98±4.62	50.68±4.81	1.54
	SRMB	62.02±4.04	60.04±4.27	57.19±4.57	54.31±4.73	51.13±4.90	1.58
	SLL	73.10±5.02	71.77±5.31	68.25±5.54	65.43±5.60	62.77±5.73	5.24
	S^{2}TMB	69.98±5.31	67.62±5.98	65.14±5.84	62.03±5.92	60.15±6.25	4.99
	KMB−	82.73±3.05	82.56±3.58	82.09±3.29	81.25±3.56	80.59±3.93	1.38
	KMB	95.10±0.63	95.13±0.67	94.39±0.64	94.66±0.71	94.47±0.81	1.63

Table 2: Average F_1 score ($\times 10^{-2}$) of discovered MB with respect to the proportion of the continuous variables.

and the performance of these algorithms under nonlinear relationships is not stable enough. The scoring process can better account for multivariate relationships than CI tests. Nevertheless, since the MB algorithm considers each variable in sequence and most scoring techniques assume linear relationships, score-based methods still fail to identify most of the nonlinear relations even though they have advantages over constraint-based methods in identifying multivariate relations. In addition, KMB efficaciously reduces the search space by ranking the candidate variables with the assistance of the CCO, thus being more efficient than score-based methods. This advantage is also reflected in the comparison with KMB−. Although KMB− can handle nonlinear multivariate relations well through the CCO, its performance is not as good as KMB due to the estimation error of CCO. Because of this, the performance variance of KMB− is large and its performance is not stable.

Performance comparison with varied MB scale: We further validate the performance of KMB on fully mixed data. We observe changes in the performance of these algorithms by adjusting the average size of MBs and controlling the other experimental settings to be the same in the simulated data. It is observed from Figure 4 that KMB consistently outperforms other comparing algorithms on mixed data. With the expansion of the MB scale, the performance of all algorithms tends to decline due to the more dense DAG. The accuracy of the CI tests in the constraint-based method is limited by the size of the conditioning set, and the score-based methods are affected by the complexity of the model, leading to performance degradation. KMB uses a more general strategy to learn MB, which can maintain its superiority in mixed data with high complexity.

Influence of Parameter k in KMB: A hyper-parameter k is used to speed up KMB by only traversing the first k

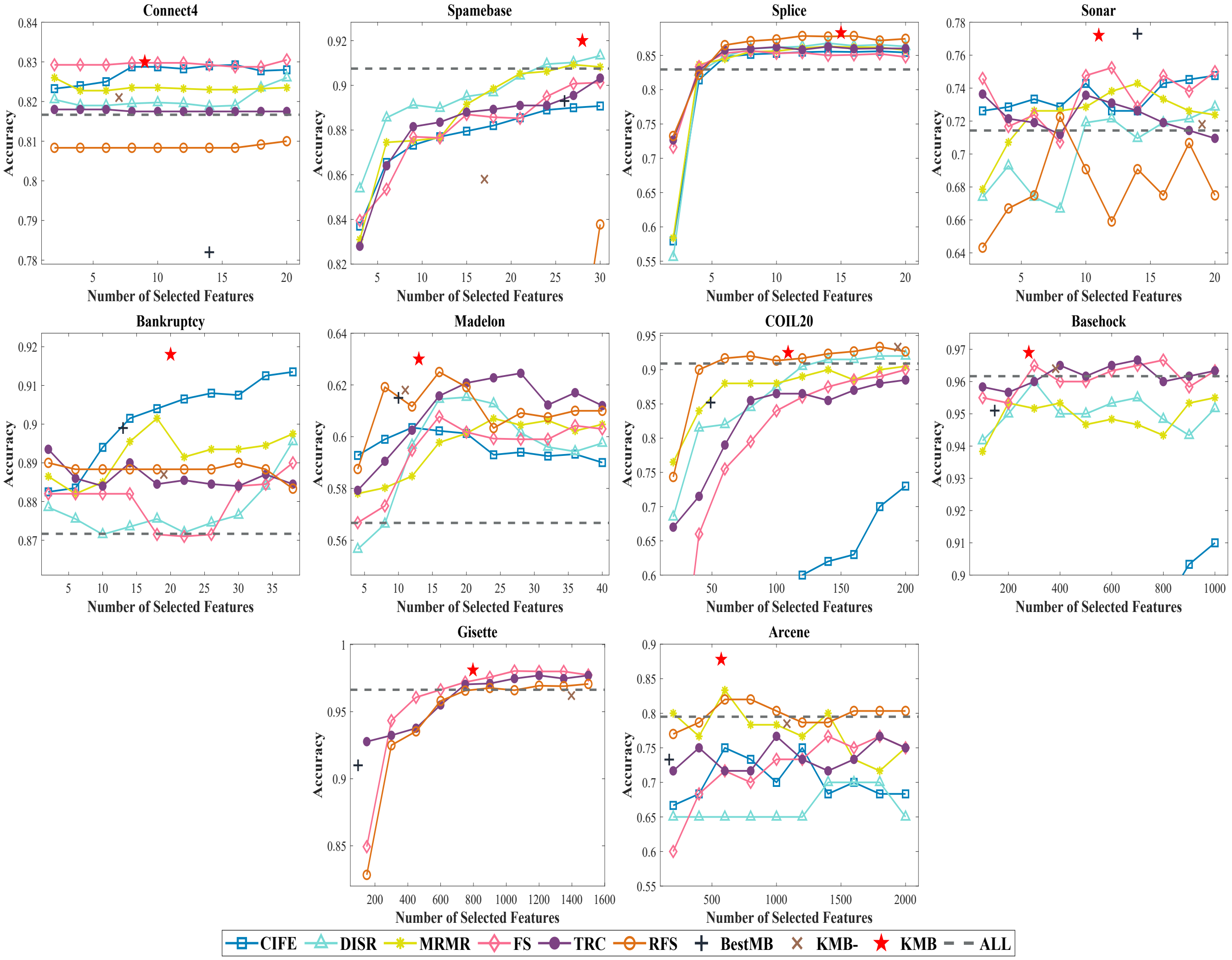


Figure 6: Classification accuracy of variable subsets selected by KMB, KMB−, and other state-of-the-art comparing algorithms on ten real-world datasets. The performance before feature selection is demonstrated via a black dotted line.

variables after a pre-sorting procedure. The choice of the value of k determines how KMB trades off efficiency and accuracy. We plot the performance of KMB for different k values on the aforementioned mixed data in Figure 5. The value of k is adjusted according to the MB scale since k should not be less than the actual size of MB to ensure that all MB variables are traversed to be possible. As can be seen from Figure 5, the accuracy of KMB gradually increases and tends to be stable as the k increases. When the sample is insufficient, the k value will also fluctuate around the optimal performance. According to the experimental results in Figure 5, when the k value is selected to be 1.5-2 times the size of MB, the ideal performance can be achieved, which can be used as a reference in practice.

Feature Selection on Real-world Data

To demonstrate the performance in the feature selection task, six traditional feature selection algorithms (CIFE (Lin and Tang 2006), DISR (Meyer, Schretter, and Bontempi 2008), MRMR (Peng, Long, and Ding 2005), FS (Duda, Hart, and Stork 2012), TRC (Nie et al. 2008), RFS (Nie et al. 2010)) and six causal (MB discovery-based) feature selection algorithms (same as previous experiments) are compared on ten datasets from diverse application domains with different scales. Table 3 provides their domains and standard statistics, including the number of features, training samples, and test samples. We employ KMB as well as other algorithms to select features first and then train the classifier SVM with these selected features, where SVM is simple and effective with few parameters, and can demonstrate the strengths of these feature subsets more clearly. Figure 6 shows the variation curves of average accuracy with respect to the percentage of selected features, where “BestMB” denotes the best performance among all the MB discovery algorithms involved in the comparison and “ALL” denotes the performance achieved with all features (without feature selec-

Dataset	Domain	#Features	#Training	#Test
Connect4	Game	42	1600	400
Spamebase	Spame Email	57	1600	400
Splice	Bioinformatics	60	1600	400
Sonar	Physical	60	160	40
Bankruptcy	Economic	147	1500	300
Madelon	Artificial	500	1600	400
Coil20	Face Image	1024	500	100
Basehock	Text	4862	1000	200
Gisette	Digit Recognition	5000	5000	1000
Arcene	Mass Spectrometry	10000	100	100

Table 3: Details of the real-world datasets.

tion). Figure 6 in the main text shows the variation curves of average classification accuracy with respect to the percentage of selected features, where “BestMB” denotes the best performance among all the MB discovery algorithms involved in the comparison and “ALL” denotes the performance achieved with all features (without feature selection). From Figure 6 in the main text, it can be seen that KMB consistently outperforms other algorithms under the same number of selected features. Moreover, the best performance achieved by the classifier after feature selection is better than the performance without feature selection in all datasets, although the advantage brought by feature selection is not significant enough on datasets with sufficient instances (e.g., Spamebase and Gisette datasets).

We can also conclude that KMB shows significant superiority in datasets of various scales, such as small-scale Spamebase, medium-scale Bankruptcy, and large-scale Arcene. In these three datasets, the features selected by KMB achieve better accuracy even compared to the best performance achieved by each algorithm. On the small-scale datasets, KMB could achieve the best or very competitive results, which demonstrates the effectiveness of KMB in the feature selection task. Note that the accuracy of “BestMB” on the Splice dataset is not plotted, this is because the best performing algorithm, SRMB, chose 38 features, accounting for $60 - 70\%$ of the total feature size, which does not meet the goal of feature selection algorithm that selecting fewer features. We also note that the predictability of features selected by KMB is slightly worse than the features selected by FS (with 20 features) on the Gisette dataset, KMB$-$ on the Sonar dataset, and RFS (160, 180, 200 features) on the Coil20 dataset, respectively. Nonetheless, KMB has chosen far fewer features than these algorithms.

On large-scale datasets, we noticed that the existing MB discovery algorithms always select a small number of features, resulting in performance degradation. This is mainly because CI tests cannot handle situations with the large-scale conditioning set. While KMB overcomes this shortcoming and does not need to use a CI test, thus can achieve practical performance on large-scale real-world data. Due to the huge solution space of KMB$-$, the performance of KMB$-$ is not stable enough, and it tends to select as many features as possible on large-scale datasets. Compared with KMB$-$, KMB avoids this risk well and still maintains good performance on large-scale data. Furthermore, on the Gisette dataset, most of the traditional feature selection methods are not illustrated because they did not complete the experiments within three days in the experimental environment. This also exposes the disadvantage of traditional feature selection algorithms on large-scale data, that is, they cannot determine the number of selected features by themselves, which leads to the time-consuming try for the optimal feature size. MB discovery algorithms, represented by KMB, can automatically determine the number of selected features, although they may be slightly slower than traditional feature selection algorithms in a single run, they can save trial time for parameter tuning in real-world applications.

Conclusion

Although MB discovery algorithms possess a theoretical guarantee for optimal feature selection, they often fail to identify some critical features in real-world data due to the strict assumptions about data distribution, variable types, or correctness of criteria. To facilitate and promote the real-world applications of MB, this paper theoretically proves the equivalence between MB and the minimal CCO, based on which we propose a more practical MB learning strategy KMB. KMB evaluates the predictability of the mapping MB variables in the RKHS, without extra assumptions. KMB could consider nonlinear multivariate dependence in the RKHS, and can maintain feasibility and effectiveness in real-world data where variables can be numerical or categorical with linear or nonlinear, pairwise or multivariate relationships. Extensive experiments demonstrate the efficacy of these contributions. We believe that some research could benefit from this work, which is presented below to prompt possible future work: (1) The application of the KMB in both real-world feature selection and causal learning tasks; (2) Extension of the KMB for the big data analytic; (3) Improvement of the KMB to address the MB learning in low-quality data such as data with noise or missing value.

Acknowledgments

This work is sponsored by the National Key R&D Program of China 2021ZD0111700, the NSFC grants 62176245, 62137002, and 62006065, the Anhui Province grants 202104a05020011 and 202103a07020002, and the Fundamental Research Funds for the Central Universities. We also appreciate the comments from anonymous reviewers, which helped to improve the paper.

References

Akaike, H. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6): 716–723.

Aliferis, C. F.; Statnikov, A.; Tsamardinos, I.; Mani, S.; and Koutsoukos, X. D. 2010a. Local causal and Markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11(1): 171–234.

Aliferis, C. F.; Statnikov, A.; Tsamardinos, I.; Mani, S.; and Koutsoukos, X. D. 2010b. Local causal and Markov blanket induction for causal discovery and feature selection for classification part ii: Analysis and extensions. *Journal of Machine Learning Research*, 11(1): 235–284.

Aliferis, C. F.; Tsamardinos, I.; and Statnikov, A. 2003. HITON: a novel Markov blanket algorithm for optimal variable selection. In *Proceedings of the American Medical Informatics Association Annual Symposium*, 21–25.

Borboudakis, G.; and Tsamardinos, I. 2019. Forward-backward selection with early dropping. *Journal of Machine Learning Research*, 20(1): 276–314.

Buntine, W. 1991. Theory refinement on Bayesian networks. In *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, 52–60. Elsevier.

Duda, R. O.; Hart, P. E.; and Stork, D. G. 2012. *Pattern classification*. John Wiley and Sons.

Fukumizu, K.; Bach, F. R.; and Jordan, M. I. 2004. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 5(1): 73–99.

Gao, T.; and Ji, Q. 2017. Efficient score-based Markov Blanket discovery. *International Journal of Approximate Reasoning*, 80: 277–293.

Greub, W. H. 2012. *Linear algebra*, volume 23. Springer Science & Business Media.

Guo, X.; Yu, K.; Cao, F.; Li, P.; and Wang, H. 2022a. Error-Aware Markov Blanket Learning for Causal Feature Selection. *Information Sciences*, 589(4): 849–877.

Guo, X.; Yu, K.; Liu, L.; Cao, F.; and Li, J. 2022b. Causal Feature Selection With Dual Correction. *IEEE Transactions on Neural Networks and Learning Systems*. Doi: 10.1109/TNNLS.2022.3178075.

Guyon, I.; Aliferis, C.; et al. 2007. Causal feature selection. In *Computational methods of feature selection*, 79–102. Chapman and Hall/CRC.

Heckerman, D.; Geiger, D.; and Chickering, D. M. 1995. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3): 197–243.

Jiang, B.; Wu, X.; Zhou, X.; Liu, Y.; Cohn, A. G.; Sheng, W.; and Chen, H. 2022a. Semi-Supervised Multiview Feature Selection With Adaptive Graph Learning. *IEEE Transactions on Neural Networks and Learning Systems*. Doi: 10.1109/TNNLS.2022.3194957.

Jiang, B.; Xiang, J.; Wu, X.; Wang, Y.; Chen, H.; Cao, W.; and Sheng, W. 2022b. Robust multi-view learning via adaptive regression. *Information Sciences*, 610: 916–937.

Lam, W.; and Bacchus, F. 1994. Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence*, 10(3): 269–293.

Li, Y.; Korb, K. B.; and Allison, L. 2022. Markov Blanket Discovery using Minimum Message Length. *arXiv preprint, arXiv:2107.08140*.

Lin, D.; and Tang, X. 2006. Conditional infomax learning: an integrated framework for feature extraction and fusion. In *Proceedings of the 9th European Conference on Computer Vision*, 68–82. Springer.

Lin, H.; and Zhang, J. 2020. On learning causal structures from non-experimental data without any faithfulness assumption. In *Proceedings of the 31st International Conference on Algorithmic Learning Theory*, 554–582. PMLR.

Liu, X.-Q.; and Liu, X.-S. 2018. Markov blanket and Markov boundary of multiple variables. *Journal of Machine Learning Research*, 19(1): 1658–1707.

McDonald, J. H. 2009. *Handbook of biological statistics*, volume 2. sparky house publishing Baltimore, MD.

Meyer, P. E.; Schretter, C.; and Bontempi, G. 2008. Information-theoretic feature selection in microarray data using variable complementarity. *IEEE Journal of Selected Topics in Signal Processing*, 2(3): 261–274.

Nie, F.; Huang, H.; Cai, X.; and Ding, C. 2010. Efficient and Robust Feature Selection via Joint l2,1-Norms Minimization. In *Proceedings of the 23rd Advances in Neural Information Processing Systems*, 1813–1821.

Nie, F.; Xiang, S.; Jia, Y.; Zhang, C.; and Yan, S. 2008. Trace ratio criterion for feature selection. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, 671–676.

Niinimaki, T.; and Parviainen, P. 2012. Local structure discovery in Bayesian networks. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, 634–643.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann Publishers.

Pellet, J.-P.; and Elisseeff, A. 2008. Using Markov blankets for causal structure learning. *Journal of Machine Learning Research*, 9(7): 1295–1342.

Pena, J. M. 2008. Learning gaussian graphical models of gene networks with false discovery rate control. In *Proceedings of the European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, 165–176. Springer.

Peng, H.; Long, F.; and Ding, C. 2005. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (8): 1226–1238.

Schölkopf, B.; Herbrich, R.; and Smola, A. J. 2001. A generalized representer theorem. In *Proceedings of the 14th International Conference on Computational Learning Theory*, 416–426. Springer.

Schölkopf, B.; Smola, A. J.; Bach, F.; et al. 2002. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press.

Sokolova, E.; Groot, P.; Claassen, T.; and Heskes, T. 2014. Causal discovery from databases with discrete and continuous variables. In *Proceedings of the European Workshop on Probabilistic Graphical Models*, 442–457. Springer.

Spirtes, P.; Glymour, C. N.; and Scheines, R. 2000. *Causation, prediction, and search.* MIT press.

Tsamardinos, I.; and Aliferis, C. F. 2003. Towards principled feature selection: relevancy, filters and wrappers. In *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics*, 809–812.

Tsamardinos, I.; Aliferis, C. F.; Statnikov, A. R.; and Statnikov, E. 2003. Algorithms for large scale Markov blanket discovery. In *Proceedings of the Florida Artificial Intelligence Research Society Conference*, 376–380.

Tsamardinos, I.; Borboudakis, G.; Katsogridakis, P.; Pratikakis, P.; and Christophides, V. 2019. A greedy feature selection algorithm for Big Data of high dimensionality. *Machine Learning*, 108(2): 149–202.

Wahba, G. 1990. *Spline models for observational data.* SIAM.

Wu, L.; Li, Z.; Zhao, H.; Liu, Q.; and Chen, E. 2022a. Estimating fund-raising performance for start-up projects from a market graph perspective. *Pattern Recognition*, 121: 108204.

Wu, L.; Wang, H.; Chen, E.; Li, Z.; Zhao, H.; and Ma, J. 2022b. Preference Enhanced Social Influence Modeling for Network-Aware Cascade Prediction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2704–2708.

Wu, T.; Wu, X.; Wang, X.; Liu, S.; and Chen, H. 2022c. Nonlinear Causal Discovery in Time Series. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 4575–4579.

Wu, X.; Jiang, B.; Lyu, S.; Wang, X.; Chen, Q.; and Chen, H. 2022d. A Survey on Causal Feature Selection Based on Markov Boundary Discovery. *Pattern Recognition and Artificial Intelligence*, 35(5): 1–17.

Wu, X.; Jiang, B.; Wang, X.; Ban, T.; and Chen, H. 2023. Feature Selection in the Data Stream Based on Incremental Markov Boundary Learning. *IEEE Transactions on Neural Networks and Learning Systems*. Doi: 10.1109/TNNLS.2023.3249767.

Wu, X.; Jiang, B.; Yu, K.; and Chen, H. 2021. Separation and recovery Markov boundary discovery and its application in EEG-based emotion recognition. *Information Sciences*, 571: 262–278.

Wu, X.; Jiang, B.; Yu, K.; Chen, H.; and Miao, C. 2020a. Multi-label causal feature selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6430–6437.

Wu, X.; Jiang, B.; Yu, K.; Miao, C.; and Chen, H. 2020b. Accurate markov boundary discovery for causal feature selection. *IEEE Transactions on Cybernetics*, 50(12): 4983–4996.

Wu, X.; Jiang, B.; Zhong, Y.; and Chen, H. 2020c. Tolerant Markov Boundary Discovery for Feature Selection. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 2261–2264.

Wu, X.; Jiang, B.; Zhong, Y.; and Chen, H. 2022e. Multi-Target Markov Boundary Discovery: Theory, Algorithm, and Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Doi: 10.1109/TPAMI.2022.3199784.

Wu, X.; Tao, Z.; Jiang, B.; Wu, T.; Wang, X.; and Chen, H. 2022f. Domain Knowledge-enhanced Variable Selection for Biomedical Data Analysis. *Information Sciences*, 606: 469–488.

Yu, K.; Guo, X.; Liu, L.; Li, J.; Wang, H.; Ling, Z.; and Wu, X. 2020a. Causality-based feature selection: Methods and evaluations. *ACM Computing Surveys*, 53(5): 1–36.

Yu, K.; Li, J.; Ding, W.; and Le, T. D. 2020b. Multi-Source Causal Feature Selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(9): 2240–2256.

Yu, K.; Liu, L.; and Li, J. 2021. A unified view of causal and non-causal feature selection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(4): 1–46.

Yu, K.; Wu, X.; Ding, W.; Mu, Y.; and Wang, H. 2018. Markov blanket feature selection using representative sets. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11): 2775–2788.

Zhong, Y.; Wu, X.; Jiang, B.; and Chen, H. 2021. Multi-label local-to-global feature selection. In *Proceedings of the 2021 International Joint Conference on Neural Networks*, 1–8. IEEE.