Information Sciences 589 (2022) 849–877

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Error-aware Markov blanket learning for causal feature selection

Xianjie Guo[a,b], Kui Yu[a,b,*], Fuyuan Cao[c], Peipei Li[a,b], Hao Wang[a,b]

[a] Key Laboratory of Knowledge Engineering With Big Data of Ministry of Education (Hefei University of Technology), Hefei 230601, China
[b] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China
[c] School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China

ARTICLE INFO

Article history:
Received 16 September 2021
Received in revised form 21 November 2021
Accepted 27 December 2021
Available online 04 January 2022

2021 MSC:
00-01
99-00

Keywords:
Causal feature selection
Markov blanket
Bayesian network
Classification

ABSTRACT

Causal feature selection has attracted much attention in recent years, since it has better robustness than the traditional feature selection. Existing causal feature selection algorithms aim to identify a Markov blanket (MB) of the class variable. The MB of the class variable implies potential local causal relations around the class variable and has been proven to be the optimal feature subset for feature selection. Since almost all existing causal feature selection methods employ conditional independence (CI) tests to learn MBs, in practical settings, existing causal feature selection algorithms encounter the problem of CI test errors, which seriously deteriorates the performance of those existing methods. To solve this issue, in this paper, we propose an Error-Aware Markov Blanket learning (EAMB) algorithm with two novel subroutines to tackle the CI test error problem. Specifically, EAMB first identifies the MB of the class variable using one subroutine, and then utilizes the other subroutine to selectively recover the missed true MB features from the discarded features. The extensive experiments on 13 real-world datasets validate the effectiveness of EAMB against fourteen state-of-the-art causal feature selection algorithms and four well-established traditional feature selection methods.

1. Introduction

Causal feature selection aims to discover a Markov blanket (MB) of a class variable for building accurate and robust prediction models. The MB was first defined and discussed by Judea Pearl in the context of a Bayesian network (BN) [1]. Under the faithfulness assumption (see Definition 3.5 in Section 3), the MB of a variable in a BN consists of its parents (direct causes), children (direct effects), and spouses (the other parents of the children of the variable). As illustrated in Fig. 1, the MB of variable Y consists of A, G and F (parents), B, C and D (children), and E and O (spouses).

Given the MB of a variable in a BN, all other variables are independent of this variable [1]. In theory, the MB of the class variable is the optimal solution to the feature selection problem [2,3]. In addition, since the MB of a variable provides a complete picture of the local causal structure around the variable, the variables in the MB are potential causally informative features which can improve the explanatory capability of predictive models [2]. In recent years, many causal feature selection

[*] Corresponding author at: School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China.
E-mail addresses: xianjieguo@mail.hfut.edu.cn (X. Guo), yukui@hfut.edu.cn (K. Yu), cfy@sxu.edu.cn (F. Cao), peipeili@hfut.edu.cn (P. Li), jsjxwangh@hfut.edu.cn (H. Wang).

https://doi.org/10.1016/j.ins.2021.12.118
0020-0255/© 2021 Elsevier Inc. All rights reserved.

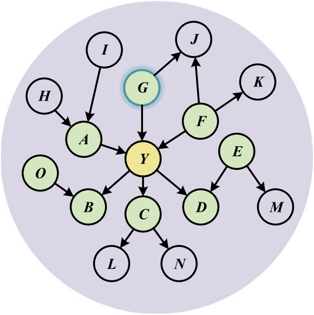


Fig. 1. Example of Bayesian network and Markov blanket. The class variable Y is in orange and its MB are in green.

methods have been proposed which are roughly divided into two different types: simultaneous MB learning and non-simultaneous MB learning.

For a class variable Y, the first type of methods does not distinguish PC (parents and children) of Y from its SP (spouses) during MB learning, such as the growth and shrink (GS) [4] algorithm and the incremental association MB (IAMB) [5]. At each iteration, this type of methods uses the entire set of features currently selected as the conditioning set for conducting conditional independence (CI) tests to calculate the dependence/independence relations between a variable and Y. For example, in Fig. 1, assuming that we aim to identify the MB of Y and the set $\boldsymbol{S} = \{A, B, C, D, E, F, O\}$ is the feature set currently selected, to calculate the dependence/independence relations between Y and G (or other features outside of $\boldsymbol{S}$), the first type of methods considers the entire set $\boldsymbol{S}$ as the conditioning set. This leads to the problem that the number of data samples required by those methods will be exponential to the size of $\boldsymbol{S}$ [2].

To mitigate the problem mentioned above, the second type of algorithms was proposed, such as the min–max MB (MMMB) [6] and the balanced MB (BAMB) [7] discoveries. This type of methods adopts a divide-and-conquer strategy to reduce data sample requirements. Specifically, those methods first find the PC of Y, then learn SP of Y. For learning PC of Y, instead of using the entire feature set $\boldsymbol{S}$ currently selected, the second type of algorithms explores all possible subsets of $\boldsymbol{S}$. Using the above example again, to determine the relation between Y and G, this type of methods searches for all possible subsets within $\boldsymbol{S}$ for finding a subset to make Y and G independent. In the worst case, all subsets within $\boldsymbol{S}$ need to be examined.

Based on the above discussions, it has been observed that both types of methods always encounter the following CI test errors in real-world applications.

- For the first type of methods, given a conditioning set, those methods only need to perform one CI test to determine the (in) dependence relations between variables, thus they are computationally efficient. However, when the size of the conditioning set increases, the CI tests will become unreliable, leading to wrong CI test results.
- For the second type of methods, those algorithms need to conduct many times of CI tests to determine the (in) dependence relations between variables, instead of one CI test. When the size of data samples is finite, more CI tests are performed, less combined statistical power (i.e., combination of individual powers of all tests) is [8]. Unreliable CI tests will greatly reduce the quality of CI tests, leading to incorrect results. Furthermore, when the size of a candidate MB becomes large, the computation costs will be expensive or prohibitive [3].

Both types of CI test problems seriously deteriorates the performance of existing causal feature selection methods. However, few studies have been proposed to tackle the two types of CI test problems simultaneously so far. In this paper, our contributions are as follows.

1. We propose an Error-Aware MB learning (EAMB) algorithm. The EAMB algorithm consists of two novel subroutines: the Efficiently Simultaneous MB (ESMB) and Selectively Recover MB (SRMB) algorithms for tackling the problems mentioned above simultaneously.
2. ESMB aims to tackle multiple CI test and expensive computational problems existing in the second type of methods. First, it adopts the idea of simultaneous MB learning to speed up the computational efficiency of EAMB. Second, it proposes a double-shrinking strategy to reduce the sizes of both conditioning set (the candidate MB features currently selected) and candidate feature set (the set outside of the candidate MB features currently selected) simultaneously for reducing unreliable CI tests as many as possible.
3. To tackle the unreliable CI test problem existing in the first type of methods due to the large size of the conditional set, we first present a relaxed AND (R-AND) rule. Then by the R-AND rule, SRMB proposes a selective strategy to find the MB of a feature within the high-dimensional discarded features to efficiently identify missed MB features due to unreliable CI tests.

4. By comparing EAMB with fourteen state-of-the-art causal feature selection algorithms and four well-established traditional feature selection algorithms, we conduct extensive experiments to validate the effectiveness of EAMB.

The rest of this paper is organized as follows. Section 2 reviews the related work and Section 3 introduces the basic notations and definitions. In Section 4, we propose the EAMB algorithm and its two subroutines (ESMB and SRMB). The experimental results and analysis are presented in Section 5. Finally, Section 6 concludes the paper.

2. Related work

Feature selection has been widely studied and used in the machine learning and pattern recognition community, since it can reduce the complexity of the problem while improving the prediction accuracy, robustness and interpretability of the learning algorithm. Given its importance, many excellent studies on feature selection methods have been proposed recently, such as [9] for a review of classic feature selection methods, [10,11] for a comprehensive survey on information-theoretic feature selection algorithms.

Traditional feature selection methods can be classified into three categories: filter, wrapper, and embedded methods. Filter methods try to find the subset of features that are most associated with the class variable. Yu et al. propose a fast correlation-based filter (FCBF) method [12], which exploits symmetrical uncertainty for feature selection. Other filter methods, for example, Rodríguez-Luján et al. propose a quadratic programming feature selection (QPFS) method [13] which takes into account simultaneously the mutual information between all pairs of features and the relevance of each feature to the class variable, and Lohrmann et al. propose a filter feature ranking method for feature selection based on fuzzy similarity and entropy measures (FSAE) [14]. Wrapper methods select the features according to classifier performance metrics. For instance, Maldonado et al. propose a wrapper method for feature selection problems using support vector machines (SVMs) [15]. But wrapper methods might suffer from high computational complexity especially for high-dimensional data [16]. Embedded methods combine the filter selection stage with the learning step and obtain the feature subsets by optimizing the objective function, such as regression shrinkage and selection via the lasso (LASSO) [17].

However, most of the traditional feature selection algorithms do not explicitly uncover cause relationships between features and the class variable, and thus they are lack of interpretability and robustness [18–21]. To address this problem, causal feature selection algorithms are presented. Causal feature selection algorithms can be applied not only to static environment, but also to dynamic environment. For example, recently, Mastakouri et al. proposed a novel and sound causal feature selection algorithm to deal with time series data with latent variables [22]. In this paper, we focus on learning causal features by finding the MB of the class variable [1] in a static environment. Koller and Sahami proposed the first MB discovery algorithm (KS) [23] and they were the first to introduce the MB to feature selection. [24] have theoretically proved that the MB of the class variable is the optimal set of features for supervised predictions.

Based on the work [23], in the past decade, numerous causality-based feature selection algorithms have been developed [25]. According to the search strategy, existing causality-based feature selection algorithms can be divided into two categories: simultaneous MB learning and non-simultaneous MB learning. Simultaneous MB learning algorithm adopts a forward–backward strategy to greedily find PC (parents and children) and SP (spouses) of the class variable simultaneously without distinguishing PC of the class variable from its SP during MB learning. The GSMB [4] is the first sound algorithm for the MB learning. IAMB [5] aims to improve the GSMB with a dynamic heuristic, which significantly improves the accuracy. Based on IAMB, many of its variants have been developed, such as Inter-IAMB [26], Fast-IAMB [27], LRH [28], FBED^K [29] and TLMB [30]. Inter-IAMB utilizes an interleaving strategy to keep the size of currently selected feature set as small as possible during the algorithm execution. To further improve the efficiency of IAMB, Fast-IAMB adopts an aggressively greedy strategy and FBED^K employs an early dropping strategy to speed up IAMB in the forward phase. Different from IAMB and its other variants discussed above, LRH is proposed to add as few false positives as possible to the candidate MB feature set. To further improve the effectiveness of simultaneous MB learning algorithms, Wu et al. propose a tolerant MB discovery (TLMB) algorithm [30], which maps the feature space and target space to a reproducing kernel Hilbert space through the conditional covariance operator, to measure the causal information carried by a feature. However, existing simultaneous MB learning algorithms are time efficient but require the number of samples to be exponential to the size of the MB, leading to CI test errors when data samples are insufficient.

To alleviate the data inefficiency problem, non-simultaneous MB learning methods are proposed which employ a divide-and-conquer strategy to learn PC and SP separately. The representative non-simultaneous MB learning algorithms include MMMB [6], HITON-MB [31], PCMB [32], IPCMB [33], MBOR [34], STMB [35], CCMB [36], BAMB [7] and EEMB [37]. MMMB is the first to adopt the divide-and-conquer strategy to search MBs, in which the subsets of PC are used as the conditioning set for conditional independence tests. The difference with MMMB is that HITON-MB tries to remove false positives from the PC set as early as possible by interleaving the shrinking phase and the growing phase. Although MMMB and HITON-MB proved to be theoretically unsound under the faithfulness assumption [32], they provide a novel way for accurate MB discovery. Compared to MMMB and HITON-MB, the Parents-Children-based MB (PCMB) algorithm and Iterative Parent–Child-based search of MB (IPCMB) algorithm, are proved to be correct under the faithfulness assumption. De et al. propose the MBOR algorithm [34], which first utilizes a fast but data inefficiency algorithm to obtain the initial MB and then corrects the MB through a divide-and-conquer search. However, for SP discovery, all of the above algorithms need to discover the

PC of each feature within the PC set of the class variable. To reduce the computational complexity, STMB [35] discovers spouses from all features excluding the current PC set instead of the expensive step of discovering the PC of PC of the class variable. Since existing MB learning algorithms rarely consider the true positives discarded during the MB search process, Wu et al. propose a cross-check and complement MB discovery (CCMB) algorithm [36] to repair this problem and further improve the accuracy of MB discovery. For achieving the trade-off between data efficiency and time efficiency, BAMB and EEMB implement the PC discovery phase and the SP identifying phase alternatively instead of discovering PC and identifying SP separately.

Although the second category of algorithms improves the data efficiency, the number of CI tests they need to conduct is required to be exponential to the size of currently selected feature set. Thus, when the size of the MB becomes large, the second category of algorithms not only significantly decreases the efficiency but also increases the probability of CI test errors. In this paper, we design a novel approach to alleviate the CI test problems that two categories of algorithms face on high-dimensional and small sample datasets while maintaining a reasonable time cost.

3. Notations and definitions

In this section, we introduce the key concepts, including Bayesian network, Markov blanket, and the relevant definitions and propositions. Table 1 provides a summary of the notations frequently used in this paper.

Definition 3.1 (*Bayesian Network, BN [1]*). Let $\mathbb{P}$ denote the joint probability distribution over feature set F of a directed acyclic graph (DAG) $\mathbb{G}$. The triplet $<\boldsymbol{F}, \mathbb{G}, \mathbb{P}>$ is called a Bayesian network if and only if $<\boldsymbol{F}, \mathbb{G}, \mathbb{P}>$ satisfies the Markov condition: every node of $\mathbb{G}$ is independent of any subset of its non-descendants conditioning on the parents of the node.

Definition 3.2 (*Conditional Independence*). Features F_i and F_j are conditionally independent given a feature set S if $P(F_i, F_j|\boldsymbol{S})$=$P(F_i|\boldsymbol{S})P(F_j|\boldsymbol{S})$, denoting as $F_i \perp\!\!\!\perp F_j|\boldsymbol{S}$. Similarly, $F_i \not\perp\!\!\!\perp F_j|\boldsymbol{S}$ represents that F_i and F_j are conditionally dependent given a feature set S.

Definition 3.3 (*Blocked Path [1]*). A path γ from feature F_i to F_j is blocked by a feature set S, if any of the following holds true: (1) γ contains a chain $F_i \rightarrow F_k \rightarrow F_j$ or a fork $F_i \leftarrow F_k \rightarrow F_j$ with the middle feature $F_k \in S$ and (2) γ contains an inverted fork (or collider) $F_i \rightarrow F_k \leftarrow F_j$ with $F_k \notin S$.

Definition 3.4 (*D-Separation [1]*). In a DAG $\mathbb{G}$, two features F_i and F_j are d-separated by a feature set $S \subset F$ iff S blocks every path from F_i to F_j, denoting as d-sep$(F_i, F_j|S)$.

Definition 3.5 (*Faithfulness [38]*). Given a BN $<\boldsymbol{F}, \mathbb{G}, \mathbb{P}>$, $\mathbb{P}$ is faithful to $\mathbb{G}$ when for any $F_i, F_j \in F$ and $S \subseteq F \setminus \{F_i, F_j\}$, $F_i \perp\!\!\!\perp F_j|\boldsymbol{S}$ in $\mathbb{P}$ iff d-sep$(F_i, F_j|S)$ in $\mathbb{G}$.

Definition 3.5 shows that conditional independence and d-separation are equivalent if the dataset and its underlying BN are faithful to each other.

In a BN, due to the symmetry relation of a node and its parents (or its children), the AND rule is defined as follows.

Definition 3.6 (*AND rule*). In a BN, if both $F_i \in \boldsymbol{PC}(F_j)$ and $F_j \in \boldsymbol{PC}(F_i)$ hold, F_i is a parent (or a child) of F_j, where $\boldsymbol{PC}(F_i)$ denotes the set of parents and children of F_i.

Definition 3.7 (*Markov Blanket, MB [1]*). Under the faithfulness assumption, the MB of any node in a Bayesian network is unique and it consists of the node's parents, children, and spouses (other parents of the node's children).

In Bayesian networks, the MB of a node renders the node statistically independent of all the remaining nodes conditioning on the MB [1], as shown in Proposition 3.1.

Proposition 3.1. In a BN, let $\mathbf{MB}(F_i)$ be the MB of node $F_i, \forall F_j \in \mathbf{F} \setminus (\mathbf{MB}(F_i) \cup Y)$, $F_i \perp\!\!\!\perp F_j|\mathbf{MB}(F_i)$ holds.

Based on Proposition 3.1, Proposition 3.2 bridges the gap between MB learning and feature selection and illustrates that learning the MB of the class variable is actually a procedure of optimal feature selection.

Proposition 3.2 (*[5,3]*). Under the faithfulness assumption, $\forall F_i \in \mathbf{F}$, F_i belongs to the MB of the class variable Y ($\mathbf{MB}(Y)$), if and only if F_i is a strongly relevant feature.

Proposition 3.3 (*[1,38]*). In a BN, if there is an edge between F_i and $Y, \forall \mathbf{S} \subseteq \mathbf{F} \setminus \{F_i\}, F_i \not\perp\!\!\!\perp Y|\mathbf{S}$ holds.

Table 1
Summary of Notation.

Symbol	Meaning
$\boldsymbol{F}$	feature set
F_i	i-th feature
$\mathbb{P}$	a joint probability distribution over $\boldsymbol{F}$
$\mathbb{G}$	a directed acyclic graph (DAG) over $\boldsymbol{F}$
Y	the class variable of dataset
$\boldsymbol{S}$	a feature set within $\boldsymbol{F}$
$F_i \perp\!\!\!\perp F_j \vert \boldsymbol{S}$	F_i is conditionally independent of F_j given $\boldsymbol{S}$
$F_i \not\perp\!\!\!\perp F_j \vert \boldsymbol{S}$	F_i is conditionally dependent of F_j given $\boldsymbol{S}$
$\boldsymbol{F} \backslash F_i$	all features in $\boldsymbol{F}$ excluding F_i
$\boldsymbol{MB}(Y)$	Markov blanket of Y
$\boldsymbol{PC}(Y)$	the set of parents and children of Y
$\boldsymbol{SP}(Y)$	the set of spouses of Y
$\boldsymbol{CurMB}(Y)$	the currently selected MB feature set of Y
$\boldsymbol{CanF}$	the candidate feature set
$\boldsymbol{Q}$	a feature queue within $\boldsymbol{F}$
$dep(.)$	a measure of the strength of the dependence
$\lfloor . \rfloor$	rounding down an integer
$.[i]$	the i-th feature of a queue
$\vert . \vert$	the size of a set
k	recall coefficient ($k \in [0,1]$)
α	the significance level of the statistical test

Proposition 3.3 states that if F_i is a parent or a child of Y, F_i and Y are not conditionally independent conditioning on any feature subsets. Proposition 3.3 is the rationale of learning the PC set of a variable of all existing causal feature selection algorithms.

Proposition 3.4 [38]. In a BN, assuming that F_i is adjacent to F_j, F_j is adjacent to F_k, and F_i is not adjacent to F_k (e.g., $F_i \rightarrow F_j \leftarrow F_k$), if $\exists \mathbf{S} \subseteq \mathbf{F} \setminus \{F_i, F_j, F_k\}$ such that $F_i \perp\!\!\!\perp F_k | \mathbf{S}$ and $F_i \not\perp\!\!\!\perp F_k | \{\mathbf{S}, F_j\}$ hold, F_i is a spouse of F_k.

Proposition 3.4 presents the relationship between a node and its spouses in a BN. It indicates that if F_i is a spouse of F_k and F_j is their common child, there exists a subset $\boldsymbol{S} \subseteq \boldsymbol{F} \setminus \{F_i, F_j, F_k\}$ such that F_i and F_k are independent given $\boldsymbol{S}$ but they are dependent given $\boldsymbol{S} \cup F_j$. For instance, E is the spouse of Y in Fig. 1. E and Y are independent ($\boldsymbol{S}$ is an empty set), but they are dependent conditioning on their common child D. Proposition 3.4 provides the idea of how to find a spouse of the class variable.

4. Our method

In this section, we propose an Error-Aware Markov Blanket learning (EAMB) algorithm, as described in Algorithm 1. EAMB consists of two subroutines: ESMB (Algorithm 2) and and SRMB (Algorithm 3).

Algorithm 1: EAMB

```
Input: $Y$: class variable, $\boldsymbol{F}$: feature set, $k$: recall coefficient ($k \in [0,1]$)
Output: $\boldsymbol{MB}$: the Markov blanket of $Y$
  Phase I: Learn the MB set of Y
$\boldsymbol{CurMB}$ = $ESMB(Y, \boldsymbol{F})$
  Phase II: Recover the MB features missed in Phase I
$\boldsymbol{MB}$ = $SRMB(Y, \boldsymbol{F}, \boldsymbol{CurMB}, k)$
return $\boldsymbol{MB}$
```

In Phase I, the ESMB subroutine learns the MB feature set of the class variable Y. To address multiple CI test and computational problems existing in the second type of MB learning methods, ESMB extends the idea of the simultaneous MB learning approach for speeding up computational efficiency and reducing the unreliable tests. In Phase II, to tackle the unreliable CI test problem due to a large size of conditioning sets, the SRMB subroutine recovers the MB features missed in Phase I from the discarded features using a selective strategy.

4.1. The ESMB subroutine

Given a class variable Y, Step 1 (Lines 1–19) of ESMB aims to discover $\boldsymbol{PC}(Y)$ (the set of parent–child features of Y) and $\boldsymbol{SP}(Y)$ (the set of spouse features of Y) simultaneously, and Step 2 (Lines 20–21) of ESMB is to recover missed spouses of Y and remove false positives from the ***CurMB***(Y) (the currently selected MB feature set of Y) learnt at Lines 1–19. Specifically, we discuss the two steps in detail as follows.

Step 1 (Lines 1–19). Assuming that ***CurMB***(Y) is empty in advance and ***CanF*** stores the candidate features outside of currently selected features (i.e., $\boldsymbol{F}\backslash$***CurMB(Y)***). Existing simultaneous MB learning methods need to repeatedly calculate the (in) dependency between Y and each feature within ***CanF***. Those repeated CI tests increase time costs and easily produce incorrect results as well. To tackle the problem, ESMB adopts a double-shrinking strategy. At Lines 9–13, ESMB dynamically shrinks the size of ***CanF*** as small as possible for avoiding repeated CI tests. At Lines 15–18, ESMB dynamically shrinks the size of ***CurMB***(Y) as small as possible for reducing the requirement of data samples.

Specifically, at Lines 4–7, *ESMB* first calculates the dependence between each feature F_i ($F_i \in$ ***CanF***) and Y conditioning on ***CurMB***(Y), then adds the feature F_{best} that has the maximum dependence with Y to ***CurMB***(Y) if F_{best} and Y are conditionally dependent conditioning on ***CurMB***(Y). Meanwhile, F_{best} is removed from ***CanF***. At Line 4, the function *dep*() can be instantiated by chi-squared test, mutual information and so on.

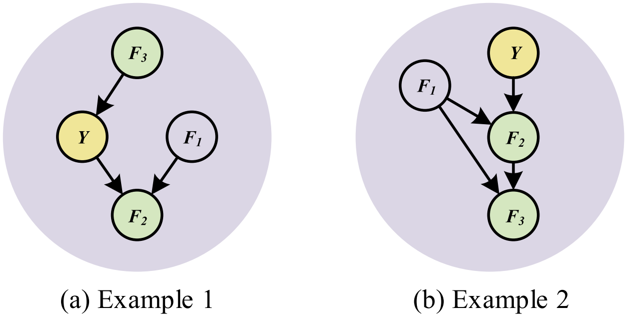
At Lines 9–13, if $F_i \in$ ***CanF*** is independent of Y conditioning on ***CurMB***(Y), F_i is removed from ***CanF*** and never considered as candidate features again. This strategy can make the size of ***CanF*** as small as possible to avoid repeated CI tests. At Lines 15–18, ESMB checks whether each feature F_i in ***CurMB***(Y) is independent of Y conditioning on ***CurMB***(Y) $\setminus F_i$. If so, F_i is removed from ***CurMB***(Y) to keep the size of ***CurMB***(Y) as small as possible.

The above two shrinking strategies (i.e. Lines 9–13 and Lines 15–18) are performed repeatedly, until ***CurMB***(Y) does not changes, and at this time, ***CanF*** must be empty. That is to say, each feature in $\boldsymbol{F}\backslash$***CurMB***(Y) and Y are conditionally independent given ***CurMB***(Y). But Lines 2 to 19 has the following drawbacks.

1. Some true spouses of Y may be discarded. Since ***CurMB***(Y) is initialized to an empty set at Line 1, if a spouse of Y and Y are conditionally independent conditioning on an empty set, such a spouse cannot be added to ***CurMB***(Y) at Line 6 and will be removed from ***CanF*** at Line 11. For example, as shown in Fig. 2(a), $F_1 \perp\!\!\!\perp Y|\varnothing$ holds. Thus, F_1 cannot be added to ***CurMB***(Y) at Lines 2–19. In addition, according to Proposition 3.4, if the common child of a spouse of Y and Y is not added to ***CurMB*** (Y) before the spouse, this spouse will be discarded at Lines 9 to 13. If those spouses of Y are not added to ***CurMB***(Y), this further leads to the following problem.
2. Some false MB features are added to ***CurMB***(Y) at Lines 2–19. For instance, as shown in Fig. 2(b), if F_1 is not added to ***CurMB***(Y), then the following holds: $F_3 \not\perp\!\!\!\perp Y|\varnothing, F_3 \not\perp\!\!\!\perp Y|F_2$ according to Definition 3.3. In this case, F_3 as a false MB feature of Y will be added to ***CurMB***(Y). Otherwise, if F_1 is added to ***CurMB***(Y), the set $\{F_2, F_1\}$ makes F_3 and Y conditionally independent of $Y : F_3 \perp\!\!\!\perp Y|\{F_2, F_1\}$ due to the Markov condition.

Algorithm 2: *ESMB*

```
Input: $Y$, $\boldsymbol{F}$
Output: CurMB: the currently selected MB feature set of $Y$
{Step 1: Discover the PC features of Y and the spouse features of Y simultaneously.}
1: Initialization: CurMB $\leftarrow \varnothing$, CanF $\leftarrow \boldsymbol{F}$
2: repeat
3:    // Select the best feature from CanF to CurMB
4:    $F_{best} \leftarrow \arg\max_{F_i \in CanF} dep(F_i, Y|CurMB)$
5:    if $F_{best} \not\perp\!\!\!\perp Y|CurMB$ then
6:       CurMB $\leftarrow$ CurMB $\cup F_{best}$, CanF $\leftarrow$ CanF $\setminus F_{best}$
7:    end if
8:    // Shrink the candidate feature space CanF
9:    for each $F_i \in$ CanF do
10:      if$F_i \perp\!\!\!\perp Y|CurMB$ then
11:         CanF $\leftarrow$ CanF $\setminus F_i$
12:      end if
13:   end for
14:   // Shrink the currently selected feature space CurMB
15:   $F_{worst} \leftarrow \arg\min_{F_i \in CurMB} dep(F_i, Y|CurMB \backslash F_i)$
16:   if$F_{worst} \perp\!\!\!\perp Y|CurMB \backslash F_{worst}$ then
17:      CurMB $\leftarrow$ CurMB $\setminus F_{worst}$
18:   end if
19: untilCurMB does not change
  {Step 2: Recover the missed spouse features and remove the false PC features.}
20: CanF $\leftarrow \boldsymbol{F} \setminus$ CurMB
21: Execute Lines 2–19 again.
22: returnCurMB
```



(a) Example 1

(b) Example 2

Fig. 2. Two Examples for illustrating the drawbacks of Lines 2–19.

Step 2 (Lines 20–21) is proposed to deal with the drawbacks mentioned above and is discussed as follows.

Step 2 (Lines 20–21). Step 2 first recovers the missed spouses from the discarded feature set $\boldsymbol{CanF}$ (i.e., $\boldsymbol{F} \backslash \boldsymbol{CurMB}(Y)$). After re-running Lines 4 to 7, ESMB can identify all missed spouses, since by Proposition 3.3, at Step 1, all PC features will be added to $\boldsymbol{CurMB}(Y)$. For example, in Fig. 2(b), at Step 1, assuming that F_1 is discarded before F_2 being added to $\boldsymbol{CurMB}(Y)$. Since F_2 as the common child of F_1 and Y is added to $\boldsymbol{CurMB}(Y)$ after Step 1, by Proposition 3.4, F_1 will be recovered from $\boldsymbol{CanF}$ when ESMB re-runs Lines 4 to 7 at Step 2.

Second, since F_1 is added to $\boldsymbol{CurMB}(Y)$ at Step 2. In this case, $\boldsymbol{CurMB}(Y)$ includes all parents of F_3, i.e., F_1 and F_2. By the Markov condition, at Lines 15 to 18, in Fig. 2(b), the false MB feature F_3 is removed from $\boldsymbol{CurMB}(Y)$.

In summery, Step 1 (Lines 1–19 of Algorithm 2) discovers all $\boldsymbol{PC}(Y)$ and part of $\boldsymbol{SP}(Y)$, and Step 2 re-runs Lines 2–19 to recover the missed spouses and remove false MB features. To illustrate the correctness of ESMB, Theorem 4.1 is proposed and proved as follows.

Theorem 4.1 (*Correctness of ESMB*). *Under the faithfulness assumption, excluding CI test errors, ESMB outputs all and only the MB of the given target variable.*

Proof. In Step I, ESMB (Algorithm 2) finds all true PC and several spouses of Y. All PC features of Y are conditionally dependent on Y given any $CurMB(Y)$ based on Theorem 3.3. Thus, ESMB eventually adds all the true PC of Y to $CurMB(Y)$ (Lines 4–7). Meanwhile, some false positives (such as F_i) are deleted from $CurMB(Y)$ if $F_i \perp\!\!\!\perp Y|\boldsymbol{CurMB}(Y) \setminus \{F_i\}$ holds (Lines 15–18). In addition, some spouses of Y may belong to $CurMB(Y)$ when the path between spouse and Y cannot be blocked given $CurMB(Y)$. For instance, a path $Y \leftarrow F_1 \rightarrow F_3 \rightarrow F_2 \leftarrow Y$ (F_3 is the spouse of Y and F_2 is the child of Y) makes $F_3 \not\perp\!\!\!\perp Y|\varnothing$ hold, and when $F_2 \in \boldsymbol{CurMB}(Y), F_3 \not\perp\!\!\!\perp Y|\boldsymbol{CurMB}(Y)$ holds (see Definition 3.3). So some spouses are added to $CurMB(Y)$.

In Step II, ESMB (Algorithm 2) retrieves all spouse of Y while removing false positives in the PC set of Y. After implementing the Step I, in theory, all true PC features of Y belong to $CurMB(Y)$, i.e., the true child of Y is added to $CurMB(Y)$, and the spouses independent of Y in Step I are added to $CanF$ (Line 20). According to Definition 3.3, all true spouses of Y are conditionally dependent on Y given the children of Y. Thus, ESMB can retrieve all spouse of Y. Finally, ESMB directly applies Proposition 3.1 to remove all false positives and obtains the exact MB of Y.

ESMB adopts a double-shrinking strategy to shrink the sizes of both $\boldsymbol{CurMB}(Y)$ and $\boldsymbol{CanF}$ to reduce unreliable tests as many as possible, however, it still adopts the idea of the simultaneous MB learning approach, that is, using the entire $\boldsymbol{CurMB}(Y)$ as conditioning set at each computation. To perform a reliable conditional independence (CI) test between features F_i and F_j given condition set $\boldsymbol{S}$, the average number of instances per cell of the contingency table of $\{F_i, F_j\} \cup \boldsymbol{CurMB}$ must be at least t [27], i.e.,

$$\frac{N}{d_{F_i} \times d_{F_j} \times d_{\boldsymbol{CurMB}}} \geqslant t. \tag{1}$$

where d_{F_i} and $d_{\boldsymbol{CurMB}}$ denote the number of values that feature F_i and the features in set $\boldsymbol{CurMB}$ (jointly) take, respectively. N denotes the number of instances in a dataset. t is a constant and its value is always set to 5 or 10.

From Eq. (1), we can see that the number of data samples required by ESMB is still exponential to the size of $\boldsymbol{CurMB}$. Therefore, when the size of $\boldsymbol{CurMB}(Y)$ is large enough and the number of data samples is insufficient, ESMB will miss some true MB features of Y. To solve this problem, we propose the SRMB subroutine next section.

4.2. The SRMB subroutine

The Selectively Recover MB (SRMB) algorithm is as shown in Algorithm 3. To solve the reliability of CI tests due to the large size of conditioning sets, we relax the AND rule (Definition 3.6) and propose the R-AND rule as follows.

Algorithm 3: *SRMB*

```
Input: $Y$, $\boldsymbol{F}$, $\boldsymbol{CurMB}$, $k$
Output: $\boldsymbol{MB}$
1: Descending sort $F_i \in \boldsymbol{F} \setminus \boldsymbol{CurMB}$, according to relevancy
2: Add the top $k\%$ features to queue $\boldsymbol{Q}$
3: $\boldsymbol{MB} = \boldsymbol{CurMB}$
  // Recover the missed MB features from queue $\mathbf{Q}$
4: for $i = 1$ to $|\boldsymbol{Q}|$ do
5:    $\boldsymbol{CurMB}(\boldsymbol{Q}[i]) = ESMB(\boldsymbol{Q}[i], \boldsymbol{F})$
6:    if $Y \in \boldsymbol{CurMB}(\boldsymbol{Q}[i])$ then
7:       $\boldsymbol{MB} \leftarrow \boldsymbol{MB} \cup \mathbf{Q}[i]$
8:    end if
9: end for
10: return $\boldsymbol{MB}$
```

Definition 4.1 (*R-AND Rule*). In a BN, (1) if $F_j \in \mathbf{PC}(F_i)$ or $F_i \in \mathbf{PC}(F_j), F_j$ is a parent (or a child) of F_i. (2) if $F_j \in \mathbf{SP}(F_i)$ or $F_i \in \mathbf{SP}(F_j), F_j$ is a spouse of F_i.

Clearly the R-AND rule is less strict than the AND rule. In real-world settings, in a BN, the size of the MB of the class variable Y is large, while the size of the MB of each feature within the MB of Y is always small, as shown in Fig. 3. In this case, by the R-AND rule, if the size of the MB of Y is large and data samples are finite, we are able to learn the MB of each variable within the MB of Y to get the MB of Y. Motivated by the idea, to recover the MB features missed by ESMB, SRMB learns those missed MB features from the discarded feature set $\boldsymbol{F}\setminus\boldsymbol{CurMB}(Y)$. Using this strategy, SRMB can mitigate the CI test problem existing in the first type of methods (i.e., simultaneous MB learning).

However, for a high-dimensional dataset, the size of $\boldsymbol{CurMB}(Y)$ is alway relatively small, while the size of $\boldsymbol{F}\setminus\boldsymbol{CurMB}(Y)$ is high dimensional. Then we propose a selective strategy to find the missed MB features from $\boldsymbol{F}\setminus\boldsymbol{CurMB}(Y)$. SRMB only examines the features that have high dependencies with Y. The rationale behind this strategy has two aspects. First, after implementing ESMB, the number of the missed MB features within $\boldsymbol{F}\setminus\boldsymbol{CurMB}(Y)$ is small, then it is not necessary to examine all features in $\boldsymbol{F}\setminus\boldsymbol{CurMB}(Y)$. Second, the features within $\boldsymbol{F}\setminus\boldsymbol{CurMB}(Y)$ that have high dependencies with Y have a high probability to be identified as the missed MB features.

SRMB is performed as follows. At Lines 1–2, SRMB first ranks the features within $\boldsymbol{F}\setminus\boldsymbol{CurMB}(Y)$ in descending according to their dependency with Y. Then it adds the top $k\%$ features to the queue $\boldsymbol{Q}$ (the analysis of parameter k please see Section 5.4). At Lines 4–9, with the R-AND rule, SRMB recovers the missed MB features by using the ESMB algorithm to find the MB of each feature in the queue $\boldsymbol{Q}$.

4.3. Computational complexity of EAMB

The computational complexity of the state-of-the-art causal feature selection methods depends on the number of conditional independence (CI) tests [2]. Since EAMB (Algorithm 1) needs to execute ESMB (Algorithm 2) and SRMB (Algorithm 3) sequentially and SRMB is to call ESMB multiple times, we first analyze computational complexity of ESMB.

The computational complexity of Algorithm 2: ESMB consists of Step 1 and Step 2. Step 1 first calculates the dependence between each feature and Y, then selects the feature F_{best} that has the maximum dependence with Y at each iteration. Whenever ESMB adds a feature F_{best} to the currently selected MB set at each iteration, we also remove the false MB features at the same time. In theory, this “interleave” approach will keep only the true MB set in $\boldsymbol{CurMB}(Y)$, so the computational complexity of ESMB is proportional to the size of the MB set. Therefore, Step 1 of ESMB takes $O(|\boldsymbol{F}||\boldsymbol{MB}|)$ CI tests, similarly, Step 2 also takes $O(|\boldsymbol{F}||\boldsymbol{MB}|)$ CI tests because of $|\boldsymbol{CurMB}| \ll |\boldsymbol{F}|$. Thus, ESMB takes $O(|\boldsymbol{F}||\boldsymbol{MB}| + |\boldsymbol{F}||\boldsymbol{MB}|)$=$O(|\boldsymbol{F}||\boldsymbol{MB}|)$ CI tests.

The computational complexity of Algorithm 3: Considering that SRMB treats each feature in feature queue $\boldsymbol{Q}$ as a target variable and implements the ESMB algorithm, it needs to execute ESMB $|\boldsymbol{Q}|$ times, i.e., SRMB takes $O(|\boldsymbol{Q}||\boldsymbol{F}||\boldsymbol{MB}|)$=$O(\lfloor k \cdot |\boldsymbol{F}|\rfloor|\boldsymbol{F}||\boldsymbol{MB}|)$ CI tests ($\lfloor . \rfloor$ denotes the rounding down of an integer).

Overall, EAMB takes $O(|\boldsymbol{F}||\boldsymbol{MB}| + \lfloor k \cdot |\boldsymbol{F}|\rfloor|\boldsymbol{F}||\boldsymbol{MB}|)$=$O(\lfloor k \cdot |\boldsymbol{F}|\rfloor|\boldsymbol{F}||\boldsymbol{MB}|)$ CI tests. Specifically, when $k = 0$, we think that EAMB takes $O(|\boldsymbol{F}||\boldsymbol{MB}|)$ CI tests. When $k = 1$, EAMB takes $O(|\boldsymbol{F}|^2|\boldsymbol{MB}|)$ CI tests. We summarize the computational complexity of the state-of-the-art causal feature selection algorithms in Table 2. From the table, Fast-IAMB is the fastest among all algorithms, and EAMB is as fast as Inter-IAMB in the best case. Moreover, EAMB is always faster than LRH, MMMB, HITON-MB, PCMB, IPCMB, MBOR, STMB, BAMB and EEMB algorithms.

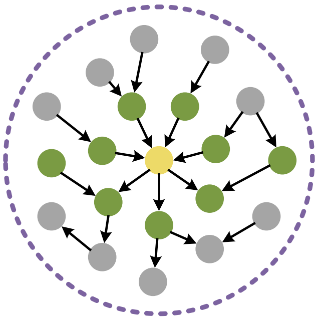


Fig. 3. An example of which the size of the MB of Y is large while the size of the MB of each feature in the MB of Y is small. The class variable Y is in yellow, the MB of Y is in green and other features are in gray.

4.4. Similarities and differences between our method and existing methods

In this section, we first briefly introduce the similarities and differences between EAMB and other causal feature selection algorithms, and then describe the shortcoming of existing methods and the advantages of our proposed method. EAMB consists of two subroutines: ESMB (Algorithm 2) and and SRMB (Algorithm 3).

- **Similarities.** Our proposed ESMB algorithm adopts the similar strategy to existing Inter-IAMB [26] and FBED^K [29] algorithms for shrinking the size of the candidate feature set and the size of the currently selected MB feature set. However, Inter-IAMB only shrinks the size of the currently selected MB feature set, while FBEDK shrinks the size of the candidate feature set.
- **Differences.** In the ESMB algorithm, first, we propose a dynamical double-shrinking strategy, i.e., shrinking both the size of the candidate feature set and the size of the currently selected MB feature set simultaneously to speeds up computational efficiency and reduces the unreliable (or redundant) CI tests. Second, we design a relaxed AND rule, R-AND rule. Based on this rule, we propose the SRMB algorithm and design a novel selective strategy to find the missed MB features from the currently discarded feature set with high dimensionality. Thus, SRMB can effectively and efficiently recover the missed MB features due to the CI test problem existing in the simultaneous MB learning methods.

In the following, we give the detailed descriptions about the disadvantages of existing causal feature selection algorithms and the advantages of EAMB.

- **The disadvantages of existing methods.** Existing simultaneous MB learning algorithms are time efficient but data inefficient. Specifically, they require the number of samples exponential to the size of the MB and thus when data samples are insufficient (e.g. small data samples), the CI tests will become unreliable, which seriously deteriorates the performance of those existing methods. In contrast, existing non-simultaneous MB learning methods are computationally expensive,

Table 2
Computational Complexity of Causal Feature Selection Algorithms.

Algorithms	Computational Complexity
GSMB	$O(\lvert\boldsymbol{F}\rvert^2)$
IAMB	$O(\lvert\boldsymbol{F}\rvert^2)$
Inter-IAMB	$O(\lvert\boldsymbol{F}\rvert\lvert\boldsymbol{MB}\rvert)$
Fast-IAMB	$O(\lvert\boldsymbol{F}\rvert)$
LRH	$O(\lvert\boldsymbol{F}\rvert^3)$
FBED^K	$O((K+1)\cdot\lvert\boldsymbol{F}\rvert^2)$
MMMB	$O(2^{\lvert\boldsymbol{PC}\rvert}\lvert\boldsymbol{F}\rvert\lvert\boldsymbol{PC}\rvert)$
HITON-MB	$O(2^{\lvert\boldsymbol{PC}\rvert}\lvert\boldsymbol{F}\rvert\lvert\boldsymbol{PC}\rvert)$
PCMB	$O(2^{\lvert\boldsymbol{PC}\rvert}\lvert\boldsymbol{F}\rvert\lvert\boldsymbol{PC}\rvert^2)$
IPCMB	$O(2^{\lvert\boldsymbol{F}\rvert}\lvert\boldsymbol{F}\rvert\lvert\boldsymbol{PC}\rvert)$
MBOR	$O(\lvert\boldsymbol{F}\rvert^2\lvert\boldsymbol{MB}\rvert)$
STMB	$O(2^{\lvert\boldsymbol{F}\rvert}\lvert\boldsymbol{F}\rvert)$
BAMB	$O(2^{\lvert\boldsymbol{PC}\rvert}\lvert\boldsymbol{F}\rvert)$
EEMB	$O(2^{\lvert\boldsymbol{PC}\rvert}\lvert\boldsymbol{F}\rvert)$
EAMB	$O(\lfloor k\cdot\lvert\boldsymbol{F}\rvert\rfloor\lvert\boldsymbol{F}\rvert\lvert\boldsymbol{MB}\rvert)$

Table 3
Description of datasets used in the experiments.

No.	Dataset	Number of instances	Number of classes	Number of features
1	colon	62	2	2,000
2	srbct	63	**4**	2,308
3	leuk	72	2	7,070
4	leukemia	72	2	7,129
5	arcene	100	2	10,000
6	prostate	102	2	6,033
7	dexter	300	2	20,000
8	madelon	2,000	2	500
9	splice	3,175	**3**	60
10	spambase	4,601	2	57
11	bankrupty	7,063	2	147
12	dnatest	1,186	**3**	180
13	semeion	1,593	**10**	256

even if they achieve better performance. Specifically, compared to the simultaneous MB learning algorithms, they need to conduct many times of CI tests to determine the dependence/independence relationships between variables. When the size of the currently selected MB feature set becomes large, the computational costs of non-simultaneous MB learning methods will be expensive or even prohibitive. In addition, when the size of data samples is finite, more CI tests are performed, less combined statistical power is, leading to incorrect results.

- **The advantages of our method.** (1) In terms of effectiveness, by calling the ESMB subroutine with the double-shrinking strategy, EAMB can reduce unreliable CI tests as many as possible; by calling the SRMB subroutine with the selective retrieval strategy, EAMB can retrieve the missed MB features due to the unreliable CI test problem, especially on the datasets with high dimensionality and small data samples. (2) In terms of efficiency, in Phase I of EAMB, EAMB adopts the efficient ESMB subroutine to tackle multiple CI test and expensive computational problems existing in the existing non-simultaneous MB learning methods. In Phase II of EAMB, although the SRMB subroutine will call the ESMB subroutine many times (Lines 4–5 of Algorithm 3), even in the worst case (i.e., the parameter k of EAMB takes the maximum value: 1), the time complexity of EAMB is still lower than that of most non-simultaneous MB learning algorithms (see Table 2 for details). In fact, on most real-world datasets, when the parameter k of EAMB is 0.05 to 0.25, the EAMB algorithm can reach the optimal solution (please see Section 5.3).

5. Experiments

In this section, we evaluate the effectiveness of our proposed EAMB by comparing with its rivals, including 6 simultaneous MB discovery algorithms, i.e., GSMB [4], IAMB [5], Inter-IAMB [26], Fast-IAMB [27], LRH [28] and FBED^K [29], 8 non-simultaneous MB discovery algorithms, i.e., MMMB [6], HITON-MB [31], PCMB [32], IPCMB [33], MBOR [34], STMB [35], BAMB [7] and EEMB [37], and 4 well-established feature selection algorithms, i.e., LASSO [17], FCBF [12], QPFS [13] and FSAE [14].

Section 5 is organized as follows. Section 5.1 describes the experimental settings. Section 5.2 presents experiments of EAMB with 18 state-of-the-art algorithms. Section 5.3 analyzes the impact of parameter k on EAMB and Section 5.4 verifies the rationality of selective strategy of SRMB.

5.1. Experiment settings

Datasets. We use 13 real-world datasets: *colon*, *srbct*, *leuk*, *leukemia*, *arcene*, *prostate*, *dexter*, *madelon*, *splice*, *spambase*, *bankrupty*, *dnatest* and *semeion* to evaluate EAMB against its rivals. These 13 real-world datasets are from the UCI Machine Learning Repository and NIPS2003 feature selection challenge datasets. Details of the datasets are summarized in Table 3, and we can see that most of datasets are high-dimensional small samples. In addition, *srbct*, *splice dnatest* and *semeion* are multi-class datasets, *madelon* includes a lot of artificial noise, and *bankrupty* is a class-imbalance dataset (the ratio of positive and negative classes is about 1:9).

Parameter setting. In the following, we illustrate the parameter settings of all algorithms.

- The conditional independence tests are G^2 tests with the statistical significance level of 0.01, and the information threshold of FCBF is set to 0.01. For the FBED^K algorithm, the value of K is set to 1, which is enough to make FBED^K converge.
- We apply 10-fold cross-validation for all datasets and adopt four classifiers, i.e., NB (Naive Bayes), KNN (K-Nearest Neighbors), DT (Decision Tree) and ANN (Artificial Neural Network) to compute their classification accuracies achieved by using the selected feature subsets. The value of k for the KNN classifier is set to 10 and KNN uses the linear kernel.

Evaluation metrics. We use the following metrics for the feature selection evaluation.

Table 4
Classification Accuracy (in %) of EAMB and Other Simultaneous MB learning Algorithms.

Classifier	Dataset	GSMB	IAMB	Inter-IAMB	Fast-IAMB	LRH	FBED	EAMB
NB	arcene	69.05	73.05	73.05	63.85	63.14	73.05	**84.16**
	dexter	70.33	79.00	79.00	74.33	77.00	79.00	**90.33**
	leuk	73.75	94.46	94.46	94.46	**98.57**	94.46	**98.57**
	leukemia	75.24	89.23	89.23	94.58	93.57	89.23	**100.00**
	prostate	66.64	91.00	91.00	93.00	92.00	91.00	**96.00**
	colon	51.43	71.90	71.90	71.90	75.48	71.90	**83.33**
	srbct	47.29	64.57	64.57	67.43	85.00	64.57	**100.00**
	madelon	57.45	60.70	60.70	58.75	60.10	60.35	**61.35**
	splice	51.91	79.62	79.62	89.16	77.48	79.62	**96.28**
	spambase	79.18	89.54	89.54	89.02	88.04	89.54	**91.15**
	bankrupty	88.56	89.35	89.35	79.34	85.16	89.32	**89.57**
	dnatest	49.66	89.12	89.12	88.96	89.71	89.12	**95.02**
	semeion	19.20	49.35	49.35	27.11	53.04	49.35	**78.42**
KNN	arcene	65.83	67.94	67.94	62.05	55.92	66.94	**82.16**
	dexter	69.33	75.33	75.33	74.00	76.33	75.33	**86.67**
	leuk	73.75	94.46	94.46	94.46	95.71	94.46	**98.57**
	leukemia	71.07	89.05	89.05	95.00	93.57	89.05	**98.75**
	prostate	50.09	90.00	90.00	94.00	92.00	90.00	**95.00**
	colon	56.43	66.19	66.19	66.19	73.81	66.19	**83.57**
	srbct	47.29	47.29	47.29	48.71	78.14	47.29	**100.00**
	madelon	55.45	60.95	60.95	58.70	**62.95**	60.50	61.15
	splice	45.04	79.72	79.72	**88.28**	76.91	79.72	87.97
	spambase	79.52	90.22	90.22	90.04	88.61	90.22	**91.76**
	bankrupty	88.36	90.09	90.09	86.04	86.63	**90.22**	**90.22**
	dnatest	41.56	88.45	88.45	87.95	88.79	88.45	**89.55**
	semeion	16.30	43.95	43.95	25.30	50.01	43.95	**82.62**
DT	arcene	68.05	71.05	71.05	64.25	63.14	70.05	**77.16**
	dexter	70.33	78.67	78.67	70.33	78.67	78.67	**86.67**
	leuk	73.75	94.46	94.46	94.46	**95.89**	94.46	**95.89**
	leukemia	72.74	90.48	90.48	93.15	91.90	90.48	**94.58**
	prostate	64.82	91.00	91.00	**94.09**	92.00	91.00	91.00
	colon	51.43	71.90	71.90	71.90	72.38	71.90	**73.57**
	srbct	47.29	64.57	64.57	68.86	83.57	64.57	**89.19**
	madelon	55.85	63.85	63.85	60.45	**66.30**	64.35	62.20
	splice	51.28	79.97	79.97	89.29	77.89	79.97	**93.61**
	spambase	80.61	90.98	90.98	90.59	89.00	90.98	**91.81**
	bankrupty	88.56	**90.61**	**90.61**	88.56	88.60	90.50	90.51
	dnatest	48.39	89.80	89.80	87.19	88.87	89.80	**91.15**
	semeion	19.20	49.47	49.47	29.69	52.03	49.47	**72.65**
ANN	arcene	58.41	69.05	63.23	58.16	64.14	60.05	**75.83**
	dexter	68.67	74.67	77.33	74.00	72.00	73.33	**88.67**
	leuk	72.32	93.04	91.61	93.04	90.18	93.21	**95.89**
	leukemia	65.18	86.37	87.80	90.48	93.57	89.05	**98.75**
	prostate	61.73	**93.00**	92.00	92.09	91.00	78.36	**93.00**
	colon	54.76	63.33	71.90	68.57	69.05	70.24	**82.38**
	srbct	44.43	64.57	64.57	67.19	79.57	64.57	**97.14**
	madelon	56.25	61.10	59.30	57.40	59.35	**61.45**	61.00
	splice	51.02	76.66	76.85	83.84	75.94	76.19	**86.08**
	spambase	78.85	90.09	90.57	90.28	89.15	90.63	**92.33**
	bankrupty	87.91	89.64	89.76	88.52	88.56	89.38	**89.89**
	dnatest	48.22	87.69	88.79	88.28	87.20	88.79	**92.58**
	semeion	18.57	47.64	48.97	27.05	49.71	48.85	**76.01**

- *Classification Accuracy*. Classification accuracy is the percentage of the correctly classified test instances that are previously unseen.
- *Precision*. The number of true positives divided by the number of positives in the prediction label.
- *Recall*. The number of true positives divided by the number of positives in the test label.
- $F1 = (2 * Precision * Recall)/(Precision + Recall)$. The $F1$ score is the harmonic average of the precision and recall, where F1 = 1 is the best case (perfect precision and recall) while F1 = 0 is the worst case.
- *Running Time*. We report running time (in seconds) as the efficiency measure of different algorithms.
- *Number of Selected Features*. The size of the feature subset selected by an algorithm.

Implementation details. (1) All algorithms are implemented in MATLAB, and all experiments are conducted on a computer with Inter Core i5-8400 2.80-GHz CPU and 16-GB memory. (2) Considering that the performance of LASSO, QPFS and FSAE depends on the number of selected features (ζ) and EAMB algorithm relies on the parameter k, we traverse the number

Table 5
Precision Metric (in %) of EAMB and Other Simultaneous MB learning Algorithms.

Classifier	Dataset	GSMB	IAMB	Inter-IAMB	Fast-IAMB	LRH	FBED	EAMB
NB	arcene	62.13	64.71	64.71	58.17	56.28	64.71	**80.67**
	dexter	90.44	93.84	93.84	67.48	74.09	93.84	**95.30**
	leuk	83.95	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	leukemia	58.33	90.17	90.17	96.67	97.50	90.17	**100.00**
	prostate	74.25	90.64	90.64	95.50	93.14	90.64	**98.33**
	colon	66.76	72.83	72.83	72.83	81.33	72.83	**89.00**
	srbct	29.83	39.24	39.24	42.15	70.00	39.24	**100.00**
	madelon	58.75	60.86	60.86	57.57	58.71	**61.81**	61.69
	splice	17.30	77.93	77.93	87.23	76.48	77.93	**95.59**
	spambase	77.21	88.32	88.32	88.45	87.36	88.32	**91.92**
	bankrupty	0.00	75.14	75.14	30.23	34.54	**77.40**	**77.40**
	dnatest	30.92	87.01	87.01	87.22	87.82	87.01	**94.36**
	semeion	12.67	49.37	49.37	21.27	55.38	49.37	**79.30**
KNN	arcene	49.08	62.05	62.05	53.67	24.17	61.33	**80.50**
	dexter	85.40	81.09	81.09	67.05	82.72	81.09	**95.90**
	leuk	83.95	**100.00**	**100.00**	**100.00**	96.67	**100.00**	**100.00**
	leukemia	41.67	92.50	92.50	**100.00**	97.50	92.50	**100.00**
	prostate	15.00	95.50	95.50	**98.00**	93.14	95.50	**98.00**
	colon	70.83	74.83	74.83	74.83	80.00	74.83	**87.33**
	srbct	29.83	31.25	31.25	30.29	62.08	31.25	**100.00**
	madelon	63.72	70.20	70.20	**76.83**	67.99	68.71	72.60
	splice	36.56	78.35	78.35	**86.34**	76.13	78.35	86.12
	spambase	80.45	90.92	90.92	91.83	89.02	90.92	**92.28**
	bankrupty	1.82	65.84	65.84	20.46	33.89	**67.79**	**67.79**
	dnatest	35.91	86.34	86.34	85.97	86.84	86.34	**87.90**
	semeion	15.43	48.25	48.25	24.05	52.42	48.25	**84.98**
DT	arcene	60.25	64.43	64.43	59.49	56.28	63.71	**74.83**
	dexter	89.68	**93.75**	**93.75**	83.94	80.77	**93.75**	**93.75**
	leuk	83.95	**100.00**	**100.00**	**100.00**	96.67	**100.00**	**100.00**
	leukemia	48.33	93.50	93.50	95.00	91.67	93.50	**100.00**
	prostate	72.58	90.64	90.64	**96.57**	93.14	90.64	90.64
	colon	66.76	72.83	72.83	72.83	**79.33**	72.83	78.00
	srbct	29.83	39.24	39.24	43.19	69.75	39.24	**82.50**
	madelon	56.56	65.75	65.75	66.54	**68.03**	66.69	64.26
	splice	34.76	78.03	78.03	87.36	77.00	78.03	**92.47**
	spambase	83.09	90.18	90.18	90.74	88.75	90.18	**91.42**
	bankrupty	0.00	70.54	70.54	5.00	48.00	71.43	**71.58**
	dnatest	33.81	87.96	87.96	85.04	87.27	87.96	**89.81**
	semeion	12.67	47.37	47.37	33.96	53.26	47.37	**73.15**
ANN	arcene	42.71	57.33	54.07	52.83	52.78	49.38	**70.98**
	dexter	90.19	86.23	86.45	76.28	73.33	85.14	**90.98**
	leuk	86.81	96.33	93.00	98.00	96.33	97.14	**100.00**
	leukemia	56.19	87.00	87.67	92.67	96.67	90.17	**100.00**
	prostate	59.96	92.64	92.64	94.07	90.64	78.31	**94.33**
	colon	59.38	73.17	76.17	74.83	78.00	74.83	**89.00**
	srbct	25.88	39.24	39.24	42.57	62.71	39.24	**98.33**
	madelon	57.63	62.14	59.20	60.84	60.94	63.66	**64.48**
	splice	18.99	72.85	74.85	81.90	74.72	71.52	**83.62**
	spambase	81.80	89.28	89.30	88.88	89.99	89.27	**91.92**
	bankrupty	0.00	56.71	57.90	0.00	0.00	69.72	**75.58**
	dnatest	25.03	85.88	86.89	86.34	85.07	86.97	**91.54**
	semeion	11.95	41.31	44.80	24.49	49.19	45.27	**78.21**

of selected features of LASSO, QPFS and FSAE from 5 to 50 with the interval 5 features and the value of k of EAMB from 0 to 1 with the interval 0.05, and then record the highest classification accuracy (as well as precision, recall and F1) as the final result of each algorithm on a dataset. And we only record the running time and the number of selected features of an algorithm when using a KNN classifier.

5.2. Experimental results on real-world dataset

In this section, we present the results obtained by EAMB in comparison with 14 causal feature selection algorithms (GSMB, IAMB, Inter-IAMB, Fast-IAMB, LRH, FBED^K, MMMB, HITON-MB, PCMB, IPCMB, MBOR, STMB, BAMB and EEMB)[1] and 4 non-causal feature selection algorithms (LASSO, FCBF, QPFS and FSAE).

[1] The source codes are available at https://github.com/kuiy

Table 6
Recall Metric (in %) of EAMB and Other Simultaneous MB learning Algorithms.

Classifier	Dataset	GSMB	IAMB	Inter-IAMB	Fast-IAMB	LRH	FBED	EAMB
NB	arcene	79.50	77.50	77.50	68.50	66.50	77.50	**84.50**
	dexter	46.00	62.67	62.67	**94.67**	90.00	62.67	86.67
	leuk	79.00	91.00	91.00	91.00	97.50	91.00	**100.00**
	leukemia	41.67	85.00	85.00	90.00	86.67	85.00	**100.00**
	prostate	61.67	92.00	92.00	90.00	92.00	92.00	**94.00**
	colon	67.50	**90.00**	**90.00**	**90.00**	82.50	**90.00**	**90.00**
	srbct	45.42	49.58	49.58	52.08	77.50	49.58	**100.00**
	madelon	51.90	61.60	61.60	67.20	**68.90**	55.20	61.30
	splice	33.33	79.06	79.06	89.79	75.29	79.06	**96.23**
	spambase	66.96	84.66	84.66	83.01	81.46	84.66	**85.66**
	bankrupty	0.00	10.16	10.16	**60.89**	27.95	9.41	31.30
	dnatest	34.08	88.62	88.62	88.70	89.50	88.62	**94.55**
	semeion	19.06	49.25	49.25	27.07	52.87	49.25	**78.42**
KNN	arcene	57.00	54.00	54.00	54.00	25.00	54.00	**87.00**
	dexter	46.67	80.00	80.00	**95.33**	70.67	80.00	80.00
	leuk	79.00	91.00	91.00	91.00	95.00	91.00	**100.00**
	leukemia	23.33	78.33	78.33	86.67	86.67	78.33	**96.67**
	prostate	4.00	84.00	84.00	90.00	92.00	84.00	**94.00**
	colon	65.00	77.50	77.50	77.50	82.50	77.50	**90.00**
	srbct	45.42	44.17	44.17	46.67	70.83	44.17	**100.00**
	madelon	27.90	39.10	39.10	25.50	**49.70**	40.70	42.00
	splice	35.51	80.06	80.06	89.36	74.25	80.06	**89.81**
	spambase	63.49	83.73	83.73	82.13	81.35	83.73	**87.42**
	bankrupty	0.49	31.33	31.33	23.29	18.06	30.71	**32.57**
	dnatest	35.66	87.81	87.81	87.43	88.62	87.81	**90.91**
	semeion	16.29	43.81	43.81	25.34	49.92	43.81	**82.50**
DT	arcene	**82.00**	71.50	71.50	64.00	66.50	71.50	77.00
	dexter	46.00	62.00	62.00	63.33	80.00	62.00	**84.67**
	leuk	79.00	91.00	91.00	91.00	**97.50**	91.00	**97.50**
	leukemia	35.00	85.00	85.00	**88.33**	**88.33**	85.00	85.00
	prostate	61.67	92.00	92.00	92.00	92.00	92.00	**94.00**
	colon	67.50	**90.00**	**90.00**	**90.00**	80.00	**90.00**	**90.00**
	srbct	45.42	49.58	49.58	53.33	76.25	49.58	**85.42**
	madelon	54.10	58.50	58.50	42.80	**61.70**	59.60	57.70
	splice	34.50	79.21	79.21	89.75	74.85	79.21	**93.22**
	spambase	63.93	86.54	86.54	84.77	82.51	86.54	**88.19**
	bankrupty	0.00	31.19	31.19	0.25	2.98	30.44	**41.08**
	dnatest	34.17	89.12	89.12	85.87	88.28	89.12	**90.44**
	semeion	19.06	49.38	49.38	29.81	51.98	49.38	**72.54**
ANN	arcene	57.00	65.50	69.50	52.00	64.00	54.00	**77.50**
	dexter	42.67	62.00	70.67	82.00	77.33	63.33	**88.67**
	leuk	73.00	93.00	93.00	91.00	89.50	93.00	**97.50**
	leukemia	46.67	85.00	81.67	85.00	86.67	85.00	**100.00**
	prostate	58.67	**94.00**	92.00	90.00	92.00	84.00	92.00
	colon	82.50	67.50	80.00	75.00	75.00	80.00	**90.00**
	srbct	41.67	49.58	49.58	53.33	69.58	49.58	**97.92**
	madelon	51.20	57.80	**60.80**	48.40	56.20	55.10	57.60
	splice	33.42	75.32	75.14	83.00	72.64	74.25	**85.70**
	spambase	59.96	85.05	86.43	86.15	81.63	86.65	**89.02**
	bankrupty	0.00	13.64	16.97	0.00	0.00	10.25	**20.93**
	dnatest	32.94	86.62	88.27	87.74	85.90	87.59	**92.86**
	semeion	18.44	47.49	48.85	27.07	49.58	48.75	**75.97**

In Tables 4–21 "-" denotes that a method fails to generate any output with the corresponding dataset after running more than one day or no feature selected and the best results are highlighted in bold face.

5.2.1. Comparison of EAMB with simultaneous MB learning methods

In this section, we report the results obtained by EAMB and the state-of-the-art simultaneous MB learning algorithms, including GSMB, IAMB, Inter-IAMB, Fast-IAMB, LRH and FBED^K.

- Classification accuracy: Table 4 summarizes the classification accuracy of EAMB against GSMB, IAMB, Inter-IAMB, Fast-IAMB, LRH and FBED^K using NB, KNN, DT and ANN classifiers respectively. We observe that using NB classifier, EAMB is never worse than its six rivals in classification accuracy. Since the simultaneous MB learning algorithms suffer from the data efficiency problem, they are much lower than EAMB on accuracy, especially on the high-dimensional datasets with smal data samples. In particular, on the *srbct* dataset, no matter which classifier is used, GSMB, IAMB, Inter-

Table 7
F1 Metric (in %) of EAMB and Other Simultaneous MB learning Algorithms.

Classifier	Dataset	GSMB	IAMB	Inter-IAMB	Fast-IAMB	LRH	FBED	EAMB
NB	arcene	68.56	68.97	68.97	61.48	58.72	68.97	**81.64**
	dexter	60.10	74.72	74.72	78.71	79.23	74.72	**89.78**
	leuk	78.90	94.44	94.44	94.44	98.57	94.44	**98.89**
	leukemia	45.71	85.40	85.40	91.00	89.57	85.40	**100.00**
	prostate	62.22	91.00	91.00	92.44	91.83	91.00	**95.48**
	colon	62.26	79.91	79.91	79.91	80.32	79.91	**87.06**
	srbct	35.72	43.75	43.75	46.47	73.41	43.75	**100.00**
	madelon	54.89	60.76	60.76	61.93	**63.31**	58.07	61.14
	splice	22.78	78.47	78.47	88.49	75.87	78.47	**95.91**
	spambase	71.62	86.39	86.39	85.60	84.26	86.39	**88.28**
	bankrupty	0.00	17.73	17.73	**40.28**	28.70	16.63	39.84
	dnatest	31.89	87.80	87.80	87.95	88.65	87.80	**94.41**
	semeion	14.99	49.20	49.20	23.53	54.08	49.20	**78.85**
KNN	arcene	51.57	53.58	53.58	52.40	23.81	52.97	**80.87**
	dexter	59.77	76.24	76.24	78.63	73.16	76.24	**85.25**
	leuk	78.90	94.44	94.44	94.44	95.71	94.44	**98.89**
	leukemia	28.33	83.24	83.24	91.00	89.57	83.24	**98.00**
	prostate	6.19	87.88	87.88	93.28	91.83	87.88	**94.39**
	colon	64.10	71.69	71.69	71.69	79.43	71.69	**86.90**
	srbct	35.72	36.22	36.22	36.30	66.08	36.22	**100.00**
	madelon	38.34	49.60	49.60	38.03	**57.11**	50.27	51.25
	splice	36.02	79.17	79.17	87.83	75.16	79.17	**87.92**
	spambase	70.85	87.04	87.04	86.64	84.92	87.04	**89.28**
	bankrupty	0.78	41.61	41.61	17.63	21.40	41.33	**42.48**
	dnatest	35.70	87.06	87.06	86.69	87.72	87.06	**89.26**
	semeion	15.63	45.86	45.86	24.42	51.12	45.86	**83.70**
DT	arcene	68.97	64.85	64.85	59.39	58.72	64.24	**73.66**
	dexter	60.17	74.23	74.23	64.49	78.05	74.23	**86.10**
	leuk	78.90	94.44	94.44	94.44	**96.75**	94.44	**96.75**
	leukemia	37.71	86.74	86.74	89.81	88.48	86.74	**90.67**
	prostate	61.27	91.00	91.00	**93.62**	91.83	91.00	91.00
	colon	62.26	79.91	79.91	79.91	77.30	79.91	**80.96**
	srbct	35.72	43.75	43.75	47.61	72.61	43.75	**83.78**
	madelon	54.66	61.63	61.63	51.80	**64.57**	62.44	60.24
	splice	33.97	78.59	78.59	88.54	75.90	78.59	**92.76**
	spambase	72.09	88.28	88.28	87.61	85.50	88.28	**89.38**
	bankrupty	0.00	43.07	43.07	0.47	5.49	41.90	**47.67**
	dnatest	33.62	88.53	88.53	85.45	87.77	88.53	**90.12**
	semeion	14.99	48.33	48.33	31.57	52.60	48.33	**72.84**
ANN	arcene	48.25	59.22	59.64	48.62	55.56	48.49	**66.30**
	dexter	55.86	71.53	75.53	74.60	72.74	70.37	**88.51**
	leuk	76.28	93.54	92.58	93.33	92.11	93.89	**96.67**
	leukemia	44.11	82.33	82.83	86.83	89.00	84.74	**98.00**
	prostate	54.15	**93.00**	91.89	91.68	91.00	80.32	92.67
	colon	68.46	67.45	77.34	74.17	74.41	76.45	**86.27**
	srbct	31.54	43.75	43.75	47.18	65.85	43.75	**98.12**
	madelon	53.61	59.61	**59.73**	52.11	58.21	58.27	58.90
	splice	23.92	73.93	74.98	82.44	73.66	72.75	**84.64**
	spambase	69.02	87.09	87.79	87.43	85.50	87.90	**90.05**
	bankrupty	0.00	21.37	25.52	0.00	0.00	17.05	**30.87**
	dnatest	28.15	86.24	87.57	87.03	85.47	87.27	**92.06**
	semeion	14.06	44.08	46.72	25.35	49.32	46.85	**77.06**

IAMB, Fast-IAMB and FBED^K significantly less accurate than EAMB. This is because the *srbct* dataset has many classes but few instances, which makes the simultaneous MB learning methods difficult to identify the true MB of Y due to the data inefficiency. For the *madelon* dataset, owing to the inclusion of a lot of artificial noise, the classification accuracy of all the algorithms is not ideal. Since GSMB does not rank features based on dependency, it works very poorly. Furthermore, we note that the classification accuracy of IAMB, Inter-IAMB and FBED^K algorithms is almost the same on most datasets. Through further research, we find that due to data inefficiency, the simultaneous MB learning methods only select few features with high dependency to Y, and IAMB, Inter-IAMB and FBED^K almost select the same feature subset on the datasets with small-sized data samples.

- Precision, Recall and F1 metrics: From Table 5–7 we can see that on most datasets (such as *leuk*, *colon*, *srbct*, *splice*, *spambase*, *dnatest* and *semeion*), no matter which classifier (i.e., NB, KNN, DT and ANN classifiers) is used, our method achieves the highest values of precision, recall and F1. Specifically, on the *leuk*, *leukemia* and *srbct* datasets, EAMB achieves 100% precision metric using both NB and KNN classifiers, and 100% recall metric using NB classifier; on the *srbct* and *semeion*

Table 8
Running Time (in Seconds) of EAMB and Other Simultaneous MB learning Algorithms based on KNN.

Dataset	GSMB	IAMB	Inter-IAMB	Fast-IAMB	LRH	FBED	EAMB
arcene	0.08	0.87	0.87	**0.27**	0.30	0.60	27.64
dexter	1.15	2.71	2.84	**0.50**	1.53	1.62	17.24
leuk	0.03	0.20	0.20	**0.18**	0.21	0.21	14.97
leukemia	0.05	0.38	0.40	**0.19**	0.21	0.42	2.30
prostate	0.04	0.34	0.36	**0.16**	0.19	0.37	93.25
colon	0.01	**0.06**	**0.06**	**0.06**	0.07	**0.06**	0.68
srbct	0.01	**0.03**	**0.03**	**0.03**	0.04	0.04	95.65
madelon	0.02	0.20	0.20	**0.03**	0.08	0.06	0.19
splice	**0.00**	0.01	0.02	**0.00**	0.01	0.01	0.03
spambase	**0.01**	0.08	0.08	**0.01**	0.13	0.06	0.57
bankrupty	**0.01**	0.21	0.21	**0.01**	0.14	0.12	0.12
dnatest	0.00	0.05	0.05	**0.01**	0.02	0.03	0.21
semeion	0.00	0.04	0.04	**0.01**	0.20	0.04	5.04

Table 9
Number of Selected Features of EAMB and Other Simultaneous MB learning Algorithms based on KNN.

Dataset	GSMB	IAMB	Inter-IAMB	Fast-IAMB	LRH	FBED	EAMB
arcene	2.50	3.00	3.00	5.00	**1.60**	3.00	4.40
dexter	3.90	**4.00**	**4.00**	6.00	7.00	**4.00**	9.90
leuk	**1.00**	**1.00**	**1.00**	**1.00**	3.00	**1.00**	355.40
leukemia	**2.00**	**2.00**	**2.00**	4.00	3.00	**2.00**	6.00
prostate	1.80	2.00	2.00	4.20	**1.90**	2.00	7.40
colon	**1.00**	**1.00**	**1.00**	**1.00**	3.00	**1.00**	53.20
srbct	**1.00**	**1.00**	**1.00**	2.00	3.00	**1.00**	66.10
madelon	5.50	**5.90**	**5.90**	8.00	9.00	6.00	6.60
splice	**3.00**	**3.00**	**3.00**	4.00	**3.00**	**3.00**	6.00
spambase	7.90	**8.00**	**8.00**	10.00	11.00	**8.00**	16.20
bankrupty	5.40	**9.00**	**9.00**	11.00	11.30	**9.00**	**9.00**
dnatest	5.00	**6.00**	**6.00**	7.00	9.00	**6.00**	17.60
semeion	2.00	**4.00**	**4.00**	5.00	10.00	**4.00**	64.00

datasets, regardless of which classifier is used, the F1 score of EAMB is significantly higher than that of its rivals. Even on the dataset with class-imbalance (such as *bankrupty*), EAMB is also able to achieve the highest F1 score when using KNN, DT and ANN classifiers.

- Running time: Table 8 reports the running time of EAMB, GSMB, IAMB, Inter-IAMB, Fast-IAMB, LRH and FBED^{K}. From the results, we can see that GSMB, IAMB, Inter-IAMB, Fast-IAMB, LRH and FBED^{K} are significantly faster than EAMB. Although EAMB uses the efficient simultaneous MB learning algorithm (ESMB) to identify MB of Y, Phase II of EAMB repeatedly calls ESMB algorithm to recover the missed MB features. Bigger is the parameter k of EAMB, slower is EAMB. It is worth noting that although the time complexity of GSMB is higher than that of Fast-IAMB, the running time of GSMB is much lower than that of Fast-IAMB. This is because GSMB does not rank candidate features according to the dependency, which renders the forward and backward phases of GSMB to quickly converge and terminate. On the *bankrupty* dataset, EAMB not only has higher accuracy than IAMB, Inter-IAMB and LRH, but also outperforms them in terms of efficiency.
- Number of Selected Features: Table 9 shows the numbers of selected features by EAMB, GSMB, IAMB, Inter-IAMB, Fast-IAMB, LRH and FBED^{K}. From the results, we can see that the simultaneous MB learning algorithms selects fewer features than EAMB since they suffer from the data efficiency problem (i.e., many key features are independent of Y conditioning on $\boldsymbol{CurMB}(Y)$). On the *bankrupty* dataset, although EAMB, IAMB, Inter-IAMB and FBED select the same number of MB features, EAMB achieves higher accuracy, which indicates that the quality of features selected by EAMB is better than that selected by IAMB, Inter-IAMB and FBED.

5.2.2. Comparison of EAMB with non-simultaneous MB learning methods

In this section, we compare the EAMB algorithm with the state-of-the-art non-simultaneous MB learning methods, including MMMB, HITON-MB, PCMB, IPCMB, MBOR, STMB, BAMB and EEMB, and the results are discussed as follows.

- Classification accuracy: From Table 10 we can see that EAMB is superior to the other algorithms on most datasets using KNN, DT and ANN classifiers. As for the *arcene* and *splice* datasets, when using KNN classifier, the classification accuracy of EAMB is 10% to 20% or more higher than the other algorithms. Furthermore, using NB classifier, EAMB is never worse than other algorithms except on the *semeion* dataset. In particular, on the *arcene* dataset, EAMB is more than 10% higher than MMMB, HITON-MB, BAMB and EEMB, and more than 20% higher than PCMB, IPCMB, MBOR and STMB on classification accuracy. On the dataset with artificial noise (such as *madelon*), as the quality of MB features selected by EAMB is higher

Table 10
Classification Accuracy (in %) of EAMB and Other Non-simultaneous MB learning Algorithms.

Classifier	Dataset	MMMB	HITON-MB	PCMB	IPCMB	MBOR	STMB	BAMB	EEMB	EAMB
NB	arcene	73.05	73.36	61.21	62.21	63.21	64.14	71.36	75.36	**84.16**
	dexter	84.67	85.33	83.33	82.67	89.67	**90.33**	88.00	85.67	**90.33**
	leuk	-	-	-	-	-	-	-	-	**98.57**
	leukemia	97.50	97.50	98.75	98.75	-	98.75	95.65	97.50	**100.00**
	prostate	-	-	-	88.00	-	64.82	-	-	**96.00**
	colon	-	-	-	-	0.00	74.05	-	-	**83.33**
	srbct	-	-	-	-	0.00	-	-	-	**100.00**
	madelon	59.00	58.65	56.00	57.50	59.85	59.55	60.30	60.00	**61.35**
	splice	95.72	95.78	95.97	95.97	95.71	96.09	95.81	96.19	**96.28**
	spambase	88.09	88.13	88.22	88.26	88.70	88.61	89.00	90.09	**91.15**
	bankrupty	83.55	84.02	85.52	85.22	84.94	80.48	84.71	87.78	**89.57**
	dnatest	94.35	94.52	94.52	94.35	94.77	93.51	94.27	93.76	**95.02**
	semeion	**85.50**	84.81	-	76.15	**85.50**	84.62	84.81	84.37	78.42
KNN	arcene	72.36	69.36	55.10	54.10	56.10	70.85	68.05	70.05	**82.16**
	dexter	85.00	85.67	83.33	80.33	86.00	81.67	85.67	85.00	**86.67**
	leuk	-	-	-	-	-	-	-	-	**98.57**
	leukemia	97.50	97.50	**98.75**	97.50	-	95.65	97.08	96.25	**98.75**
	prostate	-	-	-	85.00	-	66.64	-	-	**95.00**
	colon	-	-	-	-	0.00	79.05	-	-	**83.57**
	srbct	-	-	-	-	0.00	-	-	-	**100.00**
	madelon	61.80	59.15	52.55	55.75	61.20	59.05	**63.40**	62.30	61.15
	splice	69.95	69.70	68.79	69.48	71.81	69.98	69.92	69.61	**87.97**
	spambase	**92.39**	**92.39**	92.24	92.13	92.04	91.91	91.74	91.94	91.76
	bankrupty	89.35	89.54	89.28	89.27	89.03	89.00	89.96	90.13	**90.22**
	dnatest	87.60	87.18	88.11	87.60	83.14	82.54	89.21	89.38	**89.55**
	semeion	89.28	90.09	-	83.17	**90.59**	89.21	90.21	88.58	82.62
DT	arcene	76.36	72.36	61.21	62.21	64.21	64.34	75.27	76.36	**77.16**
	dexter	84.00	85.00	83.67	81.67	**86.67**	82.67	**86.67**	85.33	**86.67**
	leuk	-	-	-	-	-	-	-	-	**95.89**
	leukemia	89.40	90.65	89.23	87.56	-	86.55	91.73	90.65	**94.58**
	prostate	-	-	-	89.00	-	73.64	-	-	**91.00**
	colon	-	-	-	-	0.00	70.48	-	-	**73.57**
	srbct	-	-	-	-	0.00	-	-	-	**89.19**
	madelon	62.55	60.50	56.05	58.40	61.80	60.40	**64.25**	62.25	62.20
	splice	92.06	91.94	92.28	91.87	92.03	91.68	92.28	92.06	**93.61**
	spambase	91.61	91.72	91.68	91.70	91.65	91.81	91.65	**91.83**	91.81
	bankrupty	88.74	88.70	88.66	88.63	88.33	88.46	89.06	90.09	**90.51**
	dnatest	90.39	90.39	90.56	90.48	88.54	88.54	90.14	90.39	**91.15**
	semeion	**76.46**	75.34	-	69.93	75.58	74.20	75.08	73.76	72.65
ANN	arcene	72.36	69.36	62.21	60.21	61.12	62.14	61.45	69.64	**75.83**
	dexter	76.67	80.67	78.33	77.00	80.00	71.33	85.67	74.00	**88.67**
	leuk	-	-	-	-	-	-	-	-	**95.89**
	leukemia	94.82	97.50	95.42	93.39	-	95.71	91.90	**98.75**	**98.75**
	prostate	-	-	-	83.45	-	86.18	-	-	**93.00**
	colon	-	-	-	-	0.00	78.81	-	-	**82.38**
	srbct	-	-	-	-	0.00	-	-	-	**97.14**
	madelon	58.60	57.15	55.80	57.15	60.60	56.05	58.60	58.25	**61.00**
	splice	84.91	79.72	77.06	84.41	81.31	81.20	84.16	80.70	**86.08**
	spambase	**92.98**	92.61	92.91	92.85	92.87	92.57	89.68	90.65	92.33
	bankrupty	**89.89**	89.25	89.38	89.57	89.85	**89.89**	89.61	89.75	**89.89**
	dnatest	**92.75**	92.67	92.33	92.67	88.77	90.81	92.58	91.91	92.58
	semeion	81.05	82.04	-	75.33	81.37	81.61	**84.44**	82.37	76.01

than that selected by its rivals, EAMB achieves higher classification accuracy using both NB and ANN classifiers. Although non-simultaneous MB learning algorithms alleviate the data inefficiency problem, more CI tests are performed, which greatly reduces the quality of CI tests when the size of data samples is finite. Thus, on real-world datasets, the performance of non-simultaneous MB learning algorithms are still not satisfactory. In addition, when the MB of class variable Y is larger, the time and space cost of non-simultaneous MB learning methods will increase significantly, which renders MMMB, HITON-MB, PCMB, IPCMB, MBOR, STMB, BAMB and EEMB unsuccessfully to generate any output with some datasets (such as *leuk*, *leukemia*, *prostate*, *colon*, *srbct* and *semeion*).

- Precision, Recall and F1 metrics: From Table 11–13 we can observe that regardless of which classifier is used, EAMB achieves higher precision and higher recall on most datasets, especially with high dimensionality and small size samples (such as *arcene*, *leuk*, *leukemia*, *colon*, *srbct* and *prostate*). Thus, on most datasets, EAMB also get higher F1 score than its

Table 11
Precision Metric (in %) of EAMB and Other Non-simultaneous MB learning Algorithms.

Classifier	Dataset	MMMB	HITON-MB	PCMB	IPCMB	MBOR	STMB	BAMB	EEMB	EAMB
NB	arcene	71.00	72.67	45.43	51.14	52.81	55.95	70.67	76.50	**80.67**
	dexter	93.43	93.97	91.10	92.07	91.27	89.52	92.95	92.49	**95.30**
	leuk	-	-	-	-	-	-	-	-	**100.00**
	leukemia	**100.00**	**100.00**	**100.00**	**100.00**	-	**100.00**	94.17	**100.00**	**100.00**
	prostate	-	-	-	90.50	-	61.76	-	-	**98.33**
	colon	-	-	-	-	0.00	83.00	-	-	**89.00**
	srbct	-	-	-	-	0.00	-	-	-	**100.00**
	madelon	59.31	59.37	**75.96**	68.78	59.09	58.83	60.63	60.12	61.69
	splice	95.10	95.17	95.34	95.37	95.15	95.56	95.15	**95.60**	95.59
	spambase	87.92	87.98	88.36	88.43	88.95	88.67	89.45	90.37	**91.92**
	bankrupty	38.37	39.26	41.33	41.12	41.20	34.01	40.77	47.11	**77.40**
	dnatest	93.57	93.71	93.74	93.54	94.18	92.94	93.44	92.84	**94.36**
	semeion	86.53	86.19	-	77.37	**86.55**	85.63	85.91	85.41	79.30
KNN	arcene	67.71	65.33	15.00	14.29	21.67	67.64	60.33	63.98	**80.50**
	dexter	92.72	93.39	90.62	82.88	93.12	78.62	93.96	91.87	**95.90**
	leuk	-	-	-	-	-	-	-	-	**100.00**
	leukemia	**100.00**	**100.00**	**100.00**	**100.00**	-	**100.00**	96.67	**100.00**	**100.00**
	prostate	-	-	-	88.67	-	62.49	-	-	**98.00**
	colon	-	-	-	-	0.00	85.50	-	-	**87.33**
	srbct	-	-	-	-	0.00	-	-	-	**100.00**
	madelon	72.56	72.93	44.63	52.09	66.92	62.30	71.68	**73.01**	72.60
	splice	75.17	74.98	74.80	74.95	75.60	75.48	74.97	75.20	**86.12**
	spambase	92.57	92.62	**92.86**	92.65	92.11	92.10	91.57	92.21	92.28
	bankrupty	60.42	63.92	60.27	61.50	59.33	58.57	62.85	65.33	**67.79**
	dnatest	86.10	85.74	86.70	86.22	83.00	82.54	87.69	**87.94**	87.90
	semeion	90.85	91.52	-	85.33	**91.88**	90.46	91.48	89.99	84.98
DT	arcene	77.50	77.17	45.43	51.38	53.76	60.33	77.17	**81.33**	74.83
	dexter	92.12	91.50	92.60	90.63	89.31	83.93	90.34	90.49	**93.75**
	leuk	-	-	-	-	-	-	-	-	**100.00**
	leukemia	87.50	90.00	89.17	85.83	-	82.83	95.00	90.00	**100.00**
	prostate	-	-	-	89.67	-	70.32	-	-	**90.64**
	colon	-	-	-	-	0.00	75.83	-	-	**78.00**
	srbct	-	-	-	-	0.00	-	-	-	**82.50**
	madelon	62.67	62.28	**75.10**	66.75	64.22	60.57	65.80	64.28	64.26
	splice	90.99	90.85	91.21	90.76	90.97	90.57	91.24	90.99	**92.47**
	spambase	89.49	89.82	89.93	90.23	89.68	89.62	90.28	90.55	**91.42**
	bankrupty	50.93	50.65	50.59	50.54	49.10	49.38	52.77	59.43	**71.58**
	dnatest	89.46	89.46	89.70	89.72	87.79	87.53	89.28	89.45	**89.81**
	semeion	**77.69**	76.11	-	71.29	76.51	74.97	76.09	74.25	73.15
ANN	arcene	69.33	65.62	36.96	39.71	51.81	53.81	58.56	68.93	**70.98**
	dexter	76.58	88.06	87.65	81.68	80.83	69.34	90.30	80.17	**90.98**
	leuk	-	-	-	-	-	-	-	-	**100.00**
	leukemia	**100.00**	**100.00**	93.33	91.67	-	94.17	94.17	97.50	**100.00**
	prostate	-	-	-	85.79	-	85.05	-	-	**94.33**
	colon	-	-	-	-	0.00	85.67	-	-	**89.00**
	srbct	-	-	-	-	0.00	-	-	-	**98.33**
	madelon	59.43	58.90	**75.27**	68.03	62.89	56.35	63.70	58.87	64.48
	splice	82.51	76.10	71.58	82.09	77.93	78.26	81.74	77.48	**83.62**
	spambase	**92.54**	91.84	92.19	92.12	91.63	91.26	87.84	91.40	91.92
	bankrupty	56.91	50.03	67.58	56.68	69.95	48.89	53.71	74.55	**75.58**
	dnatest	91.48	**91.64**	91.15	91.62	86.93	89.54	91.45	90.58	91.54
	semeion	81.51	82.57	-	76.63	81.33	82.15	**85.36**	83.35	78.21

rivals. Specifically, on the *arcene* dataset, the F1 score of EAMB is 10% or more higher than its rivals using both NB and KNN classifiers; on the *prostate* dataset, EAMB is more than 24% higher than STMB using NB classifier, and more than 9% higher than IPCMB on F1 score.

- Running time: Since non-simultaneous MB learning methods need to learn the PC of the features within $\boldsymbol{PC}(Y)$ for finding $\boldsymbol{SP}(Y)$, their computational time will significantly increase when the size of $\boldsymbol{MB}(Y)$ becomes large. In Table 14 we see that EAMB is much faster than MMMB, HITON-MB, PCMB, IPCMB, MBOR, STMB, BAMB and EEMB on the *leuk*, *leukemia*, *colon*, *srbct*, *splice*, *spambase*, *bankrupty*, *dnatest* and *semeion* datasets. In particular, on *colon* dateset, EAMB only runs for 0.68 s while MMMB, HITON-MB, PCMB, IPCMB, BAMB and EEMB run for more than one day. For the *splice* dataset, EAMB is more than 244 times faster than other algorithms. On the *spambase*, *bankrupty* and *semeion* datasets, EAMB is also significantly faster than its rivals. Since PCMB uses symmetry checking (i.e., it needs to discover the PC of PC of Y) for removing false positives, its time efficiency is low.

Table 12
Recall Metric (in %) of EAMB and Other Non-simultaneous MB learning Algorithms.

Classifier	Dataset	MMMB	HITON-MB	PCMB	IPCMB	MBOR	STMB	BAMB	EEMB	EAMB
NB	arcene	68.50	66.50	67.50	75.50	70.50	75.00	62.50	64.00	**84.50**
	dexter	75.33	76.00	76.00	72.67	88.00	**92.00**	82.67	78.00	86.67
	leuk	-	-	-	-	-	-	-	-	**100.00**
	leukemia	93.33	93.33	96.67	96.67	-	96.67	96.67	93.33	**100.00**
	prostate	-	-	-	86.00	-	84.33	-	-	**94.00**
	colon	-	-	-	-	0.00	75.00	-	-	**90.00**
	srbct	-	-	-	-	0.00	-	-	-	**100.00**
	madelon	59.40	57.10	21.20	31.90	**65.30**	63.70	60.80	60.00	61.30
	splice	95.44	95.53	95.75	95.70	95.35	95.85	95.50	95.94	**96.23**
	spambase	80.97	81.02	80.80	80.85	81.52	81.63	81.80	83.84	**85.66**
	bankrupty	70.79	70.54	62.12	64.84	62.77	**73.88**	70.67	47.52	31.30
	dnatest	93.86	94.03	94.06	93.77	94.47	93.15	93.74	93.32	**94.55**
	semeion	**85.54**	84.84	-	76.14	85.53	84.64	84.85	84.40	78.42
KNN	arcene	70.50	56.50	23.50	21.00	28.50	62.50	61.00	62.50	**87.00**
	dexter	76.67	77.33	77.33	82.67	78.00	**90.67**	76.67	77.33	80.00
	leuk	-	-	-	-	-	-	-	-	**100.00**
	leukemia	93.33	93.33	**96.67**	93.33	-	86.67	**96.67**	90.00	**96.67**
	prostate	-	-	-	82.00	-	88.00	-	-	**94.00**
	colon	-	-	-	-	0.00	82.50	-	-	**90.00**
	srbct	-	-	-	-	0.00	-	-	-	**100.00**
	madelon	40.80	33.30	10.10	24.20	45.40	**45.60**	45.40	41.00	42.00
	splice	72.86	72.60	71.97	72.41	74.23	72.24	72.97	72.33	**89.81**
	spambase	**87.70**	87.64	86.98	86.93	87.26	86.93	87.04	86.87	87.42
	bankrupty	20.05	21.03	21.28	18.21	14.37	13.25	30.44	29.34	**32.57**
	dnatest	89.71	89.27	90.18	89.49	86.19	85.22	90.52	90.70	**90.91**
	semeion	89.20	90.01	-	83.08	**90.52**	89.13	90.14	88.49	82.50
DT	arcene	64.50	56.50	67.50	73.00	70.50	67.00	64.50	68.50	**77.00**
	dexter	75.33	78.00	74.00	72.00	80.67	82.67	82.67	79.33	**84.67**
	leuk	-	-	-	-	-	-	-	-	**97.50**
	leukemia	85.00	85.00	81.67	81.67	-	**88.33**	83.33	85.00	85.00
	prostate	-	-	-	88.00	-	85.00	-	-	**94.00**
	colon	-	-	-	-	0.00	85.00	-	-	**90.00**
	srbct	-	-	-	-	0.00	-	-	-	**85.42**
	madelon	**62.90**	58.00	23.30	40.10	53.50	59.30	60.30	56.00	57.70
	splice	91.20	91.10	91.37	90.89	91.23	90.75	91.41	91.11	**93.22**
	spambase	89.30	89.19	88.91	88.64	89.19	**89.68**	88.42	88.58	88.19
	bankrupty	**48.77**	46.29	45.19	44.69	44.31	45.67	46.77	45.43	41.08
	dnatest	89.21	89.21	89.38	89.20	87.09	87.57	89.06	89.37	**90.44**
	semeion	**76.32**	75.23	-	69.85	75.45	74.08	74.97	73.67	72.54
ANN	arcene	69.50	67.00	56.00	55.00	72.50	67.00	66.50	67.00	**77.50**
	dexter	82.67	75.33	70.67	76.00	74.67	58.67	81.33	67.33	**88.67**
	leuk	-	-	-	-	-	-	-	-	**97.50**
	leukemia	86.67	93.33	96.67	93.33	-	95.00	86.67	**100.00**	**100.00**
	prostate	-	-	-	86.00	-	90.67	-	-	**92.00**
	colon	-	-	-	-	0.00	82.50	-	-	**90.00**
	srbct	-	-	-	-	0.00	-	-	-	**97.92**
	madelon	56.60	56.20	22.10	31.80	52.90	56.10	53.80	56.60	**57.60**
	splice	84.29	78.19	74.15	83.92	79.59	79.80	83.35	79.11	**85.70**
	spambase	89.41	89.19	89.63	89.52	**90.29**	89.85	82.10	83.55	89.02
	bankrupty	28.06	**29.73**	18.57	23.48	26.39	29.06	23.68	19.15	20.93
	dnatest	**93.05**	92.91	92.63	92.59	87.85	90.86	92.64	91.89	92.86
	semeion	81.01	82.02	-	75.26	81.27	81.49	**84.41**	82.27	75.97

- Number of Selected Features: Table 15 reports the numbers of selected features of EAMB, MMMB, HITON-MB, PCMB, IPCMB, MBOR, STMB, BAMB and EEMB. From the result, EAMB is very competitive with other non-simultaneous MB learning algorithms. On the *leukemia* and *prostate* datasets, EAMB selects fewer features and achieves higher accuracy than its rivals. In contrast, STMB selects more features on all datasets and achieves lower accuracy than the other methods. Particularly, on the *colon* and *srbct* datasets, MBOR does not produce any features.

5.2.3. Comparison of EAMB with non-causal feature selection methods

In this section, we report and discuss the experimental results of the EAMB algorithm agaist four well-established feature selection methods, LASSO, FCBF, QPFS and FSAE.

Table 13
F1 Metric (in %) of EAMB and Other Non-simultaneous MB learning Algorithms.

Classifier	Dataset	MMMB	HITON-MB	PCMB	IPCMB	MBOR	STMB	BAMB	EEMB	EAMB
NB	arcene	67.97	67.57	53.48	60.15	59.20	62.57	64.39	67.19	**81.64**
	dexter	82.78	83.50	81.45	80.11	89.31	**90.49**	86.96	84.15	89.78
	leuk	-	-	-	-	-	-	-	-	**98.89**
	leukemia	95.00	95.00	98.00	98.00	-	98.00	94.57	96.00	**100.00**
	prostate	-	-	-	87.42	-	70.93	-	-	**95.48**
	colon	-	-	-	-	0.00	77.70	-	-	**87.06**
	srbct	-	-	-	-	0.00	-	-	-	**100.00**
	madelon	59.09	57.87	29.09	40.06	**61.77**	61.05	60.35	59.77	61.14
	splice	95.27	95.35	95.54	95.53	95.25	95.70	95.33	95.77	**95.91**
	spambase	84.24	84.30	84.36	84.40	85.02	84.94	85.40	86.93	**88.28**
	bankrupty	49.70	50.39	49.57	50.21	48.66	46.53	**51.60**	47.10	39.84
	dnatest	93.71	93.87	93.89	93.65	94.32	93.04	93.58	93.07	**94.41**
	semeion	86.03	85.51	-	76.75	**86.04**	85.13	85.38	84.90	78.85
KNN	arcene	68.35	57.08	18.27	17.00	23.99	62.44	57.13	58.63	**80.87**
	dexter	83.16	83.80	81.75	80.99	84.74	83.53	83.57	83.22	**85.25**
	leuk	-	-	-	-	-	-	-	-	**98.89**
	leukemia	95.00	95.00	**98.00**	95.00	-	91.33	96.00	93.00	**98.00**
	prostate	-	-	-	84.55	-	72.89	-	-	**94.39**
	colon	-	-	-	-	0.00	82.96	-	-	**86.90**
	srbct	-	-	-	-	0.00	-	-	-	**100.00**
	madelon	51.32	43.95	14.27	31.14	53.44	52.55	**54.94**	51.09	51.25
	splice	73.99	73.77	73.35	73.66	74.91	73.81	73.95	73.73	**87.92**
	spambase	**90.03**	**90.03**	89.78	89.64	89.58	89.40	89.21	89.41	89.28
	bankrupty	30.02	31.29	31.21	27.66	22.86	21.41	40.82	40.36	**42.48**
	dnatest	87.87	87.47	88.41	87.82	84.56	83.85	**89.30**	**89.30**	89.26
	semeion	90.02	90.76	-	84.19	**91.19**	89.79	90.81	89.23	83.70
DT	arcene	68.71	61.61	53.48	59.48	59.81	62.23	68.51	69.89	**73.66**
	dexter	82.21	83.54	81.33	79.09	85.62	82.75	85.98	84.04	**86.10**
	leuk	-	-	-	-	-	-	-	-	**96.75**
	leukemia	84.38	85.81	83.90	81.90	-	82.90	86.48	85.81	**90.67**
	prostate	-	-	-	88.61	-	76.26	-	-	**91.00**
	colon	-	-	-	-	0.00	78.50	-	-	**80.96**
	srbct	-	-	-	-	0.00	-	-	-	**83.78**
	madelon	62.62	58.88	30.39	45.13	58.12	59.86	**62.64**	59.53	60.24
	splice	91.09	90.97	91.29	90.82	91.10	90.66	91.32	91.05	**92.76**
	spambase	89.31	89.42	89.34	89.34	89.37	**89.58**	89.27	89.49	89.38
	bankrupty	49.73	48.21	47.68	47.36	46.43	47.37	49.36	**51.37**	47.67
	dnatest	89.33	89.33	89.54	89.45	87.44	87.54	89.16	89.41	**90.12**
	semeion	**76.99**	75.67	-	70.56	75.97	74.53	75.52	73.96	72.84
ANN	arcene	65.16	64.94	43.90	45.55	58.79	57.88	59.16	64.95	**66.30**
	dexter	78.20	79.68	76.96	77.00	76.72	59.62	84.91	70.73	**88.51**
	leuk	-	-	-	-	-	-	-	-	**96.67**
	leukemia	91.00	95.00	94.00	91.33	-	93.24	87.57	**98.57**	98.00
	prostate	-	-	-	84.54	-	87.03	-	-	**92.67**
	colon	-	-	-	-	0.00	82.62	-	-	**86.27**
	srbct	-	-	-	-	0.00	-	-	-	**98.12**
	madelon	57.30	56.03	29.38	39.84	56.59	56.11	57.86	58.19	**58.90**
	splice	83.39	77.13	72.81	82.99	78.75	79.02	82.54	78.28	**84.64**
	spambase	**90.89**	90.45	90.86	90.76	90.87	90.47	83.88	86.41	90.05
	bankrupty	35.07	**36.21**	27.07	31.98	34.78	35.07	31.12	28.51	30.87
	dnatest	92.25	**92.27**	91.88	92.10	87.38	90.19	92.04	91.23	92.06
	semeion	81.25	82.28	-	75.94	81.24	81.80	**84.88**	82.80	77.06

- Classification accuracy: In Table 16 no matter which classifier is used, EAMB is much more accurate than the other four algorithms on most of datasets. Specifically, regardless of which classifier is used, EAMB is never worse than LASSO in classification accuracy on each dataset. And on the datasets with a large number of features and a small number of samples: such as *leuk*, *leukemia*, *arcene*, *prostate* and *dexter*, the advantage of EAMB in classification accuracy is more obvious.
- Precision, Recall and F1 metrics: Tables 17–19 show that no matter which classifier is used, our method achieves the highest values of precision, recall and F1 on most datasets. Specifically, on the *leukemia* dataset, the F1 score of EAMB is more than 28% higher than that of QPFS using NB classifier; on the *srbct* dataset, the F1 score of EAMB is more than 32% higher than that of LASSO using ANN classifier; on the *arcene* dataset, the F1 score of EAMB is more than 11% higher than that of FCBF using DT classifier. On the *bankrupty* dataset with class-imbalance, EAMB is significantly superior to FCBF and FSAE.

Table 14
Running Time (in Seconds) of EAMB and Other Non-simultaneous MB learning Algorithms based on KNN.

Dataset	MMMB	HITON-MB	PCMB	IPCMB	MBOR	STMB	BAMB	EEMB	EAMB
arcene	5.93	1.91	14.03	9.94	2.04	4.78	**1.67**	1.79	27.64
dexter	40.30	**4.56**	190.99	40.66	288.28	13.68	10.21	6.56	17.24
leuk	-	-	-	-	-	-	-	-	**14.97**
leukemia	5.29	2.76	28.01	19.41	-	8.50	635.84	2.81	**2.30**
prostate	-	-	-	24.88	-	**8.10**	-	-	93.25
colon	-	-	-	-	2856.23	976.04	-	-	**0.68**
srbct	-	-	-	-	127.22	-	-	-	**95.65**
madelon	0.21	0.15	0.47	0.27	0.80	**0.14**	0.54	0.22	0.19
splice	9.22	8.66	357.17	7.32	9.70	34.12	86.22	40.85	**0.03**
spambase	54.96	34.06	1199.69	107.32	124.34	72.88	67.92	30.70	**0.57**
bankrupty	48.31	40.41	613.16	71.70	1136.54	89.63	135.82	52.99	**0.12**
dnatest	1.52	1.51	31.23	1.79	319.25	15.36	3.63	3.08	**0.21**
semeion	5202.98	5142.13	-	2707.91	1299.27	2447.44	21177.23	14457.03	**5.04**

Table 15
Number of Selected Features of EAMB and Other Non-simultaneous MB learning Algorithms based on KNN.

Dataset	MMMB	HITON-MB	PCMB	IPCMB	MBOR	STMB	BAMB	EEMB	EAMB
arcene	4.70	4.00	**1.60**	1.80	1.70	643.50	4.90	4.70	4.40
dexter	10.00	10.10	7.40	**6.80**	30.90	257.10	13.40	9.60	9.90
leuk	-	-	-	-	-	-	-	-	**355.40**
leukemia	31.60	34.00	16.40	21.10	-	141.70	21.00	8.80	**6.00**
prostate	-	-	-	25.20	-	478.80	-	-	**7.40**
colon	-	-	-	-	**0.00**	149.90	-	-	53.20
srbct	-	-	-	-	**0.00**	-	-	-	66.10
madelon	5.90	5.00	**1.50**	2.80	7.20	24.30	7.30	6.80	6.60
splice	50.40	50.90	46.10	47.10	51.50	49.20	47.80	43.30	**6.00**
spambase	43.60	43.50	41.30	41.40	48.20	54.10	34.90	30.00	**16.20**
bankrupty	60.70	57.90	42.40	45.00	46.50	89.70	44.80	20.40	**9.00**
dnatest	24.60	24.70	23.30	23.30	48.60	114.90	23.70	18.90	**17.60**
semeion	219.00	220.20	-	84.90	255.10	187.40	237.80	155.10	**64.00**

- Running time: From Table 20 we can see that EAMB is much faster than QPFS on most datasets. Specially, EAMB is 46.5 times faster than QPFS on *leukemia*, and 35.0 times faster than QPFS on *colon*. Since FCBF adopts pairwise mutual information test with lower time complexity to calculate relevancy between features and the class variable, the efficiency of EAMB is significantly slower than FCBF.
- Number of Selected Features: As shown in Table 21 we can see that compared with the other four algorithms, EAMB chooses fewer features but achieves higher classification accuracy on most of datasets. EAMB is also very competitive with FCBF, especially on the datasets with a large number of features: *arcene*, *dexter*, *leukemia*, *prostate* and *srbct*.

To further compare the performance (for classification accuracy) of EAMB with that of its rivals, the Friedman test and Nemenyi test [39] are employed. We first perform the Friedman test at the 0.05 significance level under the null-hypothesis, which states that all algorithms are performing equivalently (i.e., the average ranks of all algorithms are equivalent). The average ranks of EAMB and its rivals are summarized in Table 22. Since the MMMB, HITON-MB, PCMB, IPCMB, MBOR, STMB, BAMB and EEMB algorithms can not produce any output on some datasets, we does not record their average ranks in this table. From Table 22, we can see that the null hypothesis is rejected on each classifier. We also note that EAMB performs better than its rivals (the lower rank value is better).

To further analyze the significant difference between EAMB and its rivals, we perform the Nemenyi test, which states that the performance of two algorithm is significantly different if the corresponding average ranks differ by at least one critical difference (CD). The CD for the Nemenyi test is calculated as follows (i.e., Eq. (2)).

$$\mathrm{CD} = q_{\alpha,m}\sqrt{\frac{m(m+1)}{6|\mathscr{D}|}} \tag{2}$$

where α is the significance level and $|m|$ is the number of comparison algorithms, and $|\mathscr{D}|$ denotes the number of real-world datasets. In our experiments, $m = 11, q_{\alpha=0.05,m=11} = 3.219$ at significance level $\alpha = 0.05$. Whether using NB or KNN classifiers, $|\mathscr{D}| = 13$, and thus CD = 4.19.

Figs. 4(a), Figs. 4(b), Figs. 4(c) and Figs. 4(d) provide the CD diagrams, where the average rank of each algorithm is marked along the axis (lower ranks to the right). Using NB classifier, we observe that EAMB achieves a comparable performance against FCBF, QPFS and FSAE, and EAMB significantly performs better than the other algorithms. Using KNN classifier, we

Table 16
Classification Accuracy (in %) of the Well-established Feature Selection Methods and EAMB.

Classifier	Dataset	LASSO	FCBF	QPFS	FSAE	EAMB
NB	arcene	72.07	71.43	63.23	56.01	**84.16**
	dexter	85.67	89.33	**90.33**	50.00	**90.33**
	leuk	93.04	97.14	94.46	87.68	**98.57**
	leukemia	98.75	97.08	83.33	91.37	**100.00**
	prostate	92.00	95.00	63.64	94.00	**96.00**
	colon	68.81	**85.24**	**85.24**	78.81	83.33
	srbct	85.05	98.33	**100.00**	89.19	**100.00**
	madelon	61.15	57.00	61.20	**62.20**	61.35
	splice	90.49	95.37	**96.28**	95.46	**96.28**
	spambase	88.07	90.04	91.37	**91.65**	91.15
	bankrupty	88.55	0.00	88.67	88.59	**89.57**
	dnatest	77.66	89.97	94.43	94.52	**95.02**
	semeion	74.65	80.29	**80.91**	69.75	78.42
KNN	arcene	69.05	68.34	71.12	56.01	**82.16**
	dexter	78.00	82.33	86.00	50.00	**86.67**
	leuk	90.36	95.71	95.71	95.71	**98.57**
	leukemia	**98.75**	97.08	74.35	**98.75**	**98.75**
	prostate	91.09	**95.00**	76.36	**95.00**	**95.00**
	colon	70.48	84.05	**86.90**	80.24	83.57
	srbct	77.10	95.71	**100.00**	98.57	**100.00**
	madelon	57.35	51.25	60.50	57.50	**61.15**
	splice	81.13	78.40	87.37	87.37	**87.97**
	spambase	90.91	90.50	**92.13**	92.20	91.76
	bankrupty	88.62	0.00	88.46	88.56	**90.22**
	dnatest	61.80	88.54	88.97	87.52	**89.55**
	semeion	82.49	81.49	81.92	72.14	**82.62**
DT	arcene	69.14	69.36	69.03	56.01	**77.16**
	dexter	82.00	80.33	**88.33**	50.00	86.67
	leuk	90.36	**95.89**	94.46	**95.89**	**95.89**
	leukemia	94.40	93.15	77.08	93.15	**94.58**
	prostate	88.18	88.09	69.73	**94.00**	91.00
	colon	67.38	**78.57**	74.05	74.05	73.57
	srbct	71.05	84.10	83.57	**89.19**	**89.19**
	madelon	59.80	57.10	60.15	59.25	**62.20**
	splice	86.55	92.88	93.32	92.60	**93.61**
	spambase	90.48	91.05	**92.17**	92.13	91.81
	bankrupty	88.83	0.00	88.70	88.56	**90.51**
	dnatest	70.91	87.95	91.99	**92.50**	91.15
	semeion	68.55	**74.69**	73.63	68.30	72.65
ANN	arcene	72.16	65.25	66.87	56.01	**75.83**
	dexter	70.00	79.67	88.00	50.00	**88.67**
	leuk	90.36	90.18	93.04	73.93	**95.89**
	leukemia	**98.75**	96.90	82.80	98.33	**98.75**
	prostate	93.00	**94.00**	67.64	92.18	93.00
	colon	72.62	77.14	82.14	76.19	**82.38**
	srbct	65.43	84.00	**97.14**	**97.14**	**97.14**
	madelon	60.05	57.10	60.50	59.85	**61.00**
	splice	72.00	85.29	85.83	85.29	**86.08**
	spambase	91.39	91.37	**93.00**	92.72	92.33
	bankrupty	88.69	0.00	88.86	88.56	**89.89**
	dnatest	72.85	88.45	91.99	**92.58**	**92.58**
	semeion	73.96	72.20	74.51	65.05	**76.01**

note that EAMB significantly outperforms GSMB, IAMB, Inter-IAMB, Fast-IAMB, LRH and FBED^K, and EAMB achieves a comparable performance against the other algorithms. Using DT classifier, we see that EAMB significantly outperforms GSMB and Fast-IAMB, and EAMB achieves a comparable performance against the other algorithms. Figs. 4(d) show that EAMB achieves a comparable performance against FCBF, QPFS, LASSO and FSAE, and EAMB significantly performs better than the other algorithms when using ANN classifier. No matter which classifier is used, EAMB is the only algorithm that achieves the lowest rank value (the lower rank value is better).

5.3. Parameter sensitivity analysis

In this section, based on KNN classifier and the metric of classification accuracy, we will study the influence of the parameter on the proposed methods. Figs. 5(b), 5(c) and 5(d) show the variation curve of classification accuracy of LASSO, QPFS and

Table 17
Precision Metric (in %) of the Well-established Feature Selection Methods and EAMB.

Classifier	Dataset	LASSO	FCBF	QPFS	FSAE	EAMB
NB	arcene	72.33	66.00	55.01	0.00	**80.67**
	dexter	93.68	90.93	88.39	0.00	**95.30**
	leuk	98.00	97.50	95.50	89.33	**100.00**
	leukemia	**100.00**	96.67	74.17	90.00	**100.00**
	prostate	96.67	98.00	61.13	96.33	**98.33**
	colon	81.33	**93.50**	92.67	86.00	89.00
	srbct	87.08	99.17	**100.00**	89.68	**100.00**
	madelon	61.68	57.03	64.76	**72.06**	61.69
	splice	89.66	94.48	95.58	94.98	**95.59**
	spambase	87.03	88.30	91.08	90.82	**91.92**
	bankrupty	50.00	0.00	57.06	36.16	**77.40**
	dnatest	76.53	88.02	93.52	93.77	**94.36**
	semeion	75.56	**81.51**	81.49	69.83	79.30
KNN	arcene	78.67	62.33	72.50	0.00	**80.50**
	dexter	94.89	89.71	94.60	50.00	**95.90**
	leuk	98.00	97.50	95.83	**100.00**	**100.00**
	leukemia	**100.00**	96.67	45.00	**100.00**	**100.00**
	prostate	**100.00**	98.33	72.64	98.00	98.00
	colon	75.50	**92.17**	91.00	84.00	87.33
	srbct	70.39	97.92	**100.00**	99.17	**100.00**
	madelon	68.44	69.49	66.58	67.54	**72.60**
	splice	79.29	79.58	85.24	85.24	**86.12**
	spambase	92.52	90.00	**92.93**	92.67	92.28
	bankrupty	51.97	0.00	49.97	1.19	**67.79**
	dnatest	60.28	86.53	87.22	85.97	**87.90**
	semeion	84.22	82.93	83.79	74.01	**84.98**
DT	arcene	64.67	69.67	72.00	0.00	**74.83**
	dexter	93.68	83.76	92.11	0.00	**93.75**
	leuk	91.67	96.67	98.33	96.67	**100.00**
	leukemia	95.00	97.50	75.00	95.00	**100.00**
	prostate	91.81	86.74	70.98	**96.33**	90.64
	colon	74.33	**87.33**	82.67	77.50	78.00
	srbct	71.25	78.33	75.42	80.79	**82.50**
	madelon	60.07	57.10	67.43	**75.74**	64.26
	splice	85.19	91.86	91.51	91.64	**92.47**
	spambase	90.10	89.92	**91.42**	91.40	**91.42**
	bankrupty	52.07	0.00	57.06	26.86	**71.58**
	dnatest	69.47	86.19	91.33	**91.46**	89.81
	semeion	69.20	**75.34**	74.48	69.11	73.15
ANN	arcene	70.17	57.33	59.45	8.55	**70.98**
	dexter	80.34	85.11	**90.98**	40.00	**90.98**
	leuk	96.67	97.50	97.14	74.29	**100.00**
	leukemia	97.50	96.67	83.33	**100.00**	**100.00**
	prostate	96.67	**98.33**	71.75	94.90	94.33
	colon	77.88	84.83	**89.00**	87.00	**89.00**
	srbct	66.18	83.75	96.04	**98.33**	**98.33**
	madelon	59.69	57.10	65.21	**71.83**	64.48
	splice	69.13	83.09	82.77	83.00	**83.62**
	spambase	90.81	89.67	92.16	**92.23**	91.92
	bankrupty	32.65	0.00	47.67	10.00	**75.58**
	dnatest	71.71	86.57	91.27	91.53	**91.54**
	semeion	74.65	73.86	74.47	65.04	**78.21**

FSAE by varying the parameter ζ (i.e. the number of selected features), respectively. The impact of k on the classification accuracy of EAMB is shown in Fig. 5(a).

In Figs. 5(b), 5(c) and 5(d), we note that there is no obvious rule for classification accuracy and the number of selected features. For selecting different number of features, the classification accuracy fluctuates greatly. But, in Fig. 5(a), we observe that when $0.05 \leqslant k \leqslant 0.25$, EAMB achieves the highest classification accuracy on all datasets except *srbct* and *semeion* datasets. For the *srbct* dataset, more classes and fewer instances render Eq. (1) of Section 4.2 difficult to hold, which makes ESMB unable to fully identify the MB of dense features (the size of the MB of such feature is large). Nevertheless, on the *srbct* dataset, most of the MB features of class variable are the dense features, i.e., SRMB cannot immediately recover the missed MB features through R-AND rule. Thus, when $k = 0.7$, classification accuracy of EAMB achieves the highest.

Based on the analysis of k of EAMB above, we can draw the following two conclusions.

Table 18
Recall Metric (in %) of the Well-established Feature Selection Methods and EAMB.

Classifier	Dataset	LASSO	FCBF	QPFS	FSAE	EAMB
NB	arcene	65.50	70.50	77.50	0.00	**84.50**
	dexter	78.00	88.00	**96.67**	0.00	86.67
	leuk	92.00	97.50	97.50	95.50	**100.00**
	leukemia	96.67	96.67	75.00	76.67	**100.00**
	prostate	90.00	92.00	80.33	92.00	**94.00**
	colon	72.50	85.00	87.50	85.00	**90.00**
	srbct	88.33	98.75	**100.00**	90.00	**100.00**
	madelon	65.00	**72.00**	65.70	64.10	61.30
	splice	90.19	95.22	96.17	95.07	**96.23**
	spambase	82.79	86.21	85.49	**94.54**	85.66
	bankrupty	60.77	0.00	**73.02**	60.58	31.30
	dnatest	76.25	89.64	94.35	94.32	**94.55**
	semeion	74.64	80.26	**80.91**	69.71	78.42
KNN	arcene	36.50	70.50	66.50	0.00	**87.00**
	dexter	64.67	74.67	94.00	**100.00**	80.00
	leuk	90.00	95.50	97.50	95.00	**100.00**
	leukemia	96.67	96.67	30.00	**100.00**	96.67
	prostate	92.00	92.00	86.00	92.00	**94.00**
	colon	87.50	85.00	90.00	**95.00**	90.00
	srbct	74.58	96.67	**100.00**	98.75	**100.00**
	madelon	**55.90**	17.80	44.00	39.40	42.00
	splice	83.10	81.59	88.62	88.62	**89.81**
	spambase	83.95	85.44	**87.42**	87.20	**87.42**
	bankrupty	23.39	0.00	**66.19**	8.77	32.57
	dnatest	63.35	88.20	90.87	89.12	**90.91**
	semeion	82.37	81.35	81.81	72.04	**82.50**
DT	arcene	59.50	61.50	71.00	0.00	**77.00**
	dexter	78.00	79.33	**92.00**	0.00	84.67
	leuk	96.00	**97.50**	95.50	**97.50**	**97.50**
	leukemia	**91.67**	85.00	68.33	88.33	85.00
	prostate	88.33	90.00	76.67	92.00	**94.00**
	colon	80.00	82.50	87.50	85.00	**90.00**
	srbct	72.92	81.25	79.17	**85.42**	**85.42**
	madelon	58.90	**72.00**	57.70	57.50	57.70
	splice	85.79	91.86	92.67	91.79	**93.22**
	spambase	86.76	87.09	89.57	**96.41**	88.19
	bankrupty	33.66	0.00	40.46	2.35	**41.08**
	dnatest	69.01	86.49	91.28	**91.99**	90.44
	semeion	68.43	**74.63**	73.55	68.19	72.54
ANN	arcene	70.50	64.00	57.00	20.00	**77.50**
	dexter	70.00	77.33	**94.67**	80.00	88.67
	leuk	92.00	87.00	95.50	93.00	**97.50**
	leukemia	**100.00**	96.67	71.67	**100.00**	**100.00**
	prostate	**92.00**	90.00	79.00	90.33	**92.00**
	colon	82.50	80.00	87.50	**90.00**	**90.00**
	srbct	65.83	82.92	96.25	97.50	**97.92**
	madelon	62.50	**72.00**	61.00	57.30	57.60
	splice	69.87	84.95	85.66	84.76	**85.70**
	spambase	86.98	88.36	**90.02**	89.46	89.02
	bankrupty	5.57	0.00	9.76	0.62	**20.93**
	dnatest	71.75	87.65	92.21	91.86	**92.86**
	semeion	73.88	72.13	74.46	64.93	**75.97**

1. When k = 0.05, EAMB achieves the approximate optimal classification accuracy on the most of datasets. For datasets with smaller samples and a larger number of classes, k can be increased appropriately (the upper limit is generally 0.25).
2. On the dataset with many dense features (such as *srbct* and *semeion*), when k takes a larger value (such as 0.7), EAMB can approximate the highest classification accuracy.

5.4. Rationale of the selective strategy of SRMB

In this section, we use the experimental results on benchmark and real-world datasets to demonstrate the rationality of selective strategy of SRMB.

Table 19
F1 Metric (in %) of the Well-established Feature Selection Methods and EAMB.

Classifier	Dataset	LASSO	FCBF	QPFS	FSAE	EAMB
NB	arcene	65.39	67.59	62.93	0.00	**81.64**
	dexter	84.15	89.14	**89.78**	0.00	**89.78**
	leuk	94.67	97.50	95.68	91.43	**98.89**
	leukemia	98.00	96.00	71.95	81.33	**100.00**
	prostate	90.81	94.39	69.25	93.48	**95.48**
	colon	74.25	86.42	86.30	81.63	**87.06**
	srbct	87.65	98.95	**100.00**	89.60	**100.00**
	madelon	61.62	59.49	**62.48**	61.85	61.14
	splice	89.92	94.85	95.86	95.02	**95.91**
	spambase	84.51	87.16	88.61	**89.37**	88.28
	bankrupty	46.33	0.00	**50.68**	31.88	39.84
	dnatest	76.38	88.82	93.93	94.04	**94.41**
	semeion	75.10	80.88	**81.20**	69.77	78.85
KNN	arcene	45.63	65.55	64.06	0.00	**80.87**
	dexter	73.89	80.66	**85.25**	66.67	**85.25**
	leuk	92.37	96.39	96.59	95.56	**98.89**
	leukemia	**98.00**	96.00	34.67	**98.00**	**98.00**
	prostate	89.88	93.69	78.17	**94.39**	**94.39**
	colon	79.56	86.77	**89.92**	85.24	86.90
	srbct	72.21	97.24	**100.00**	98.95	**100.00**
	madelon	**54.50**	25.37	51.85	48.02	51.25
	splice	81.15	80.57	86.90	86.90	**87.92**
	spambase	87.88	87.62	89.62	**89.78**	89.28
	bankrupty	31.34	0.00	**48.58**	2.10	42.48
	dnatest	61.76	87.35	88.53	87.52	**89.26**
	semeion	83.28	82.13	82.78	73.01	**83.70**
DT	arcene	60.52	61.75	65.32	0.00	**73.66**
	dexter	81.03	80.36	**88.13**	0.00	86.10
	leuk	93.43	**96.75**	95.66	**96.75**	**96.75**
	leukemia	**90.67**	89.24	65.71	89.81	**90.67**
	prostate	87.47	88.08	70.85	88.08	**91.00**
	colon	75.56	**80.96**	79.65	79.51	**80.96**
	srbct	71.80	79.43	77.09	**88.63**	83.78
	madelon	58.87	59.54	57.46	56.28	**60.24**
	splice	85.49	91.86	92.56	91.67	**92.76**
	spambase	87.69	88.40	89.80	**89.86**	89.38
	bankrupty	38.34	0.00	43.94	4.11	**47.67**
	dnatest	69.23	86.33	91.27	**91.72**	90.12
	semeion	68.81	**74.99**	74.01	68.65	72.84
ANN	arcene	64.51	57.88	52.55	11.96	**66.30**
	dexter	69.61	79.19	88.10	53.33	**88.51**
	leuk	91.99	91.35	94.73	81.66	**96.67**
	leukemia	**98.57**	96.00	73.67	98.00	98.00
	prostate	91.67	92.58	70.45	91.82	**92.67**
	colon	78.80	81.63	83.85	82.89	**86.27**
	srbct	65.54	83.11	96.13	97.91	**98.12**
	madelon	60.95	59.54	**61.09**	58.22	58.90
	splice	69.50	84.01	83.70	83.87	**84.64**
	spambase	88.83	88.94	**90.99**	90.61	90.05
	bankrupty	9.00	0.00	15.19	1.09	**30.87**
	dnatest	71.70	87.10	91.64	91.17	**92.06**
	semeion	74.26	72.97	74.44	64.95	**77.06**

5.4.1. Rationale on benchmark dataset

As shown in Fig. 6, using ESMB algorithm, we conducted experiments on the well-known benchmark Alarm BN with 37 variables. Since the MB of each variable can be read off from the benchmark BN, we generated 500 samples from the BN and select 17 dense variables (denoted as ordinate) in this BN for showing the rationale of the selective strategy of SRMB.

For a CI test of F_i and Y conditioning on $\boldsymbol{S}$, the corresponding p-value p_{F_i} is smaller, the relevancy between F_i and Y is higher. If and only if $p_{F_i} > \alpha$ (α is the significance level of the statistical test), we accept the null hypothesis "$H_i : F_i \perp\!\!\!\perp Y | \boldsymbol{S}$ ". In Fig. 6, assuming V_1 is the class variable, we use hollow circle to record p_{V_i} ($V_i \in$ all variables except $\boldsymbol{MB}(V_1)$), that is, $p_{V_i} > \alpha$. Among them, the red hollow circle denotes the missed true MB variables of a class variable. We can see that the p-values of missed true MB variables are closer to α than the p-value of other variables. And the red hollow circles farther

Table 20
Running Time (in Seconds) of the Well-established Feature Selection Methods and EAMB.

Dataset	LASSO	FCBF	QPFS	FSAE	EAMB
arcene	1.37	**0.26**	53.55	1.31	27.64
dexter	4.06	**0.35**	347.34	3.37	17.24
leuk	0.85	**0.25**	76.87	0.81	14.97
leukemia	0.78	**0.30**	106.91	0.83	2.30
prostate	0.90	**0.23**	112.69	0.74	93.25
colon	0.27	**0.04**	23.77	0.22	0.68
srbct	0.27	**0.19**	37.94	0.52	95.65
madelon	2.08	**0.03**	2.56	0.31	0.19
splice	0.26	**0.03**	0.06	0.08	**0.03**
spambase	0.53	**0.04**	0.07	0.07	0.57
bankrupty	2.54	**0.02**	0.13	0.21	0.12
dnatest	0.34	**0.01**	0.10	0.10	0.21
semeion	1.57	**0.08**	0.19	0.51	5.04

Table 21
Number of Selected Features of the Well-established Feature Selection Methods and EAMB.

Dataset	LASSO	FCBF	QPFS	FSAE	EAMB
arcene	35.00	33.50	35.00	5.00	**4.40**
dexter	15.00	41.30	50.00	**5.00**	9.90
leuk	40.00	52.30	**5.00**	45.00	355.40
leukemia	**5.00**	51.10	50.00	15.00	6.00
prostate	20.00	41.90	50.00	35.00	**7.40**
colon	45.00	**7.40**	35.00	35.00	53.20
srbct	50.00	96.80	25.00	**20.00**	66.10
madelon	10.00	**2.00**	15.00	20.00	6.60
splice	**5.00**	19.70	10.00	**5.00**	6.00
spambase	50.00	**10.60**	30.00	45.00	16.20
bankrupty	35.00	**0.00**	45.00	5.00	9.00
dnatest	10.00	**8.10**	15.00	15.00	17.60
semeion	50.00	**25.60**	50.00	50.00	64.00

Table 22
The average ranks of EAMB and its rivals using NB, KNN, DT and ANN classifiers.

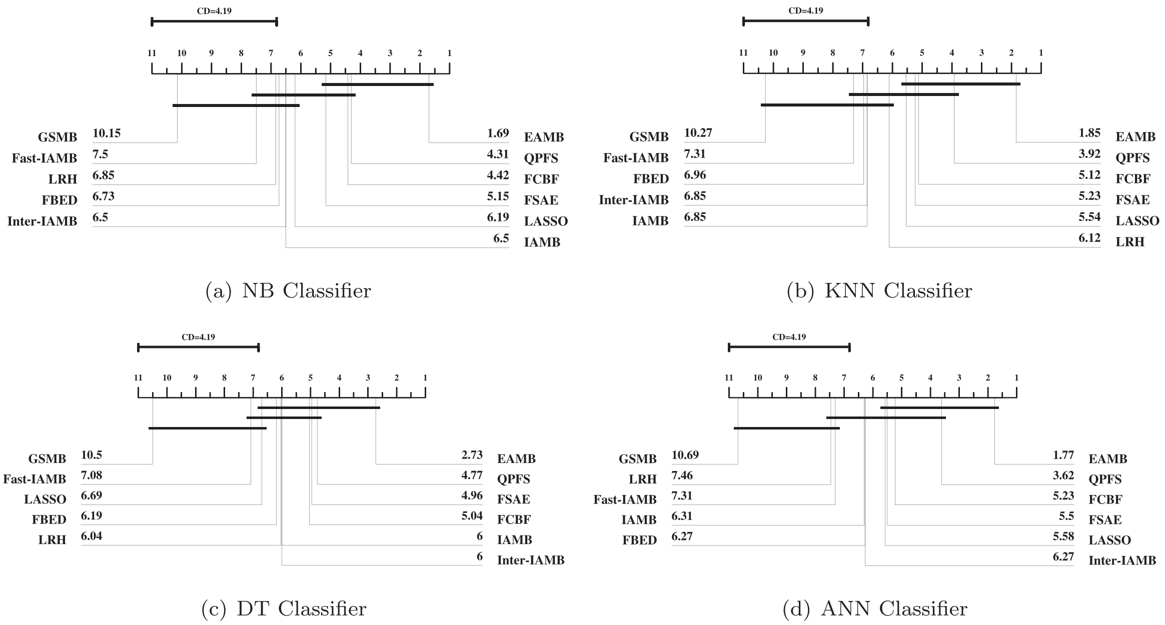
Algorithm		GSMB	IAMB	Inter-IAMB	Fast-IAMB	LRH	FBED	LASSO	FCBF	QPFS	FSAE	EAMB
	NB	10.15	6.50	6.50	7.50	6.85	6.73	6.19	4.42	4.31	5.15	**1.69**
	KNN	10.27	6.85	6.85	7.31	6.12	6.96	5.54	5.12	3.92	5.23	**1.85**
	DT	10.50	6.00	6.00	7.08	6.04	6.19	6.69	5.04	4.77	4.96	**2.73**
Avg rank	ANN	10.69	6.31	6.27	7.31	7.46	6.27	5.58	5.23	3.62	5.5	**1.77**

from α are generally the SP of a class variable, that is because the relevancy between the class variable and its SP is less than that between the class variable and PC.

5.4.2. Rationale on real-world dataset

In this section, we further validate the rationale of the selective strategy of SRMB by observing what effects the parameter k has on the number of selected features of EAMB. As shown in Fig. 7, we note that the number of selected features increases significantly when $k \leqslant 0.3$ on most datasets, and then tends to be stable. In particular, since *leuk* and *colon* are high-dimensional small samples and noisy datasets, a large number of true positives are wrongly discarded. Thus, the numbers of selected features have a linear relationship with k. On *srbct* dataset, when $k \geqslant 0.6$, the number of selected features starts to increase significantly, which has been explained in Section 5.3.

The above results indicate that false negatives with greater relevance to Y are more likely to be recalled (see Lines 1–2 of Algotithm 3). Combined with the analysis in Section 5.4.1 we further confirm that the selective strategy of SRMB is reasonable.



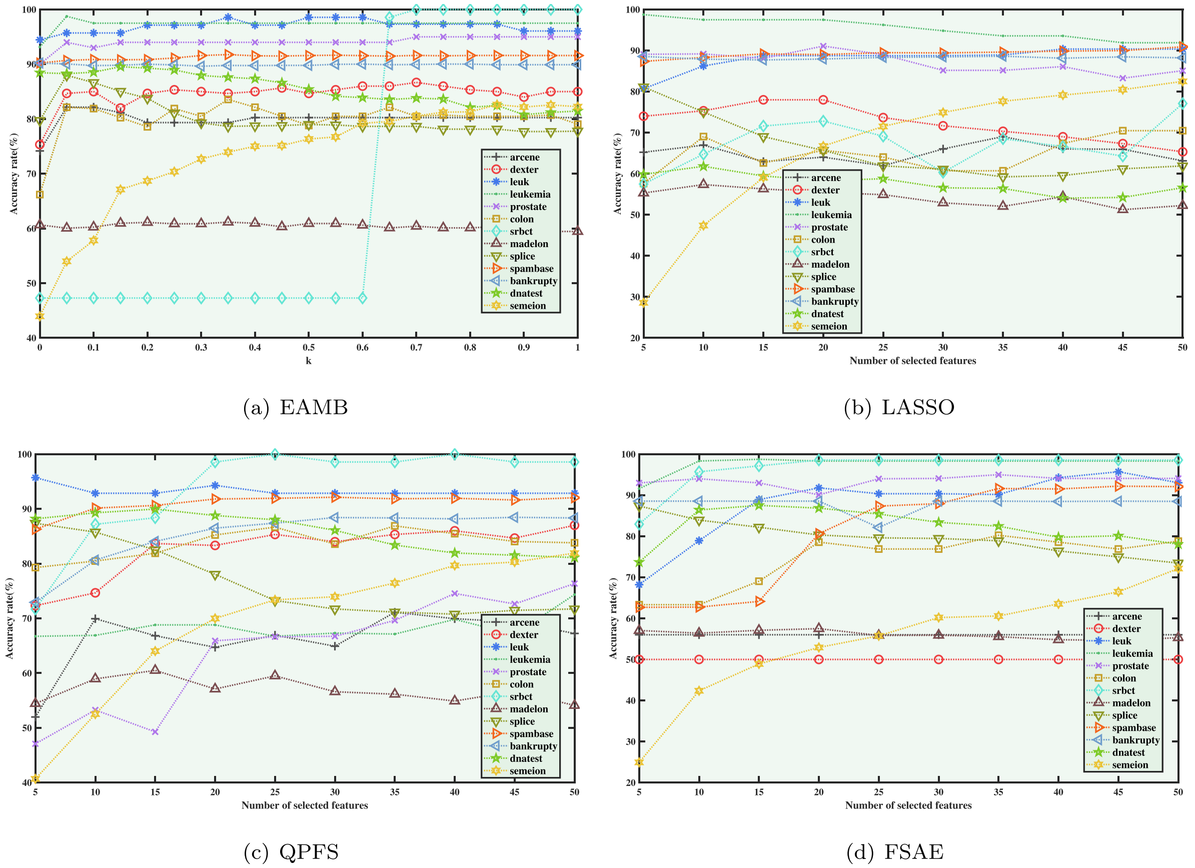
(a) NB Classifier

(b) KNN Classifier

(c) DT Classifier

(d) ANN Classifier

Fig. 4. Crucial difference diagram of the Nemenyi test on 13 real-world datasets. (Since MMMB, HITON-MB, PCMB, IPCMB, MBOR, STMB, BAMB and EEMB fail to generate any output on some datasets, their results are not shown in the crucial difference diagram.).



(a) EAMB

(b) LASSO

(c) QPFS

(d) FSAE

Fig. 5. Comparison of parameter sensitivity of EAMB, LASSO, QPFS and FSAE on 13 real-world datasets..

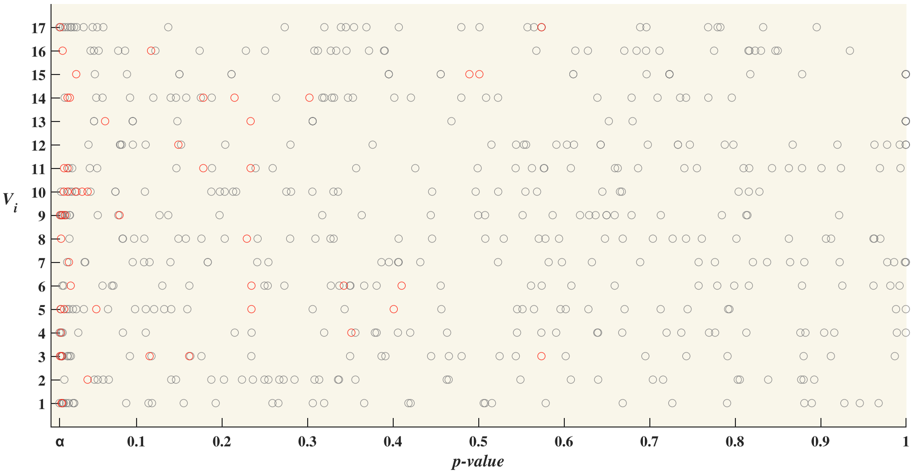


Fig. 6. An example of p-value distribution when conducting experiments on the benchmark BN dataset.

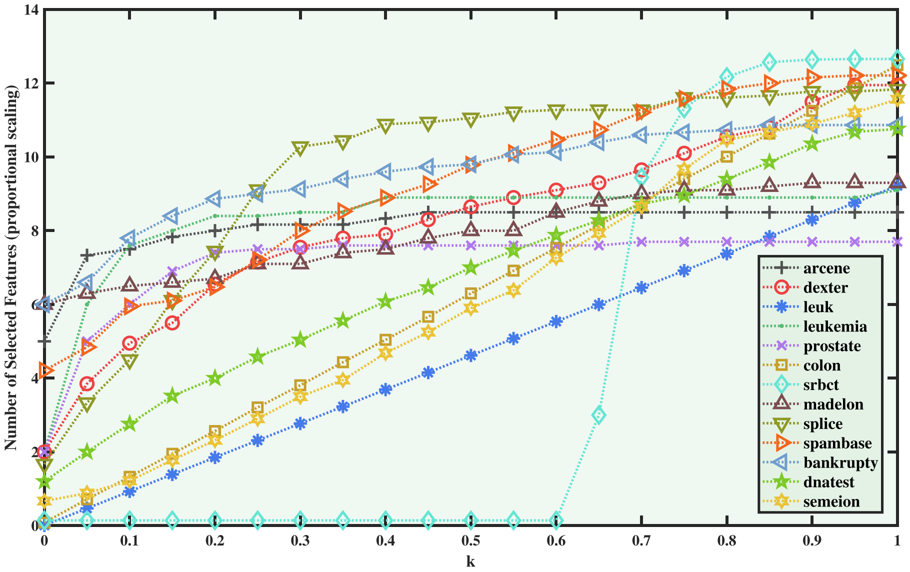


Fig. 7. Number of selected features of EAMB by varying the values of k ($k \in [0.1]$). (For the convenience of observation, the number of features selected by an algorithm on some datasets are compressed in proportion.).

6. Conclusion

This paper focuses on the problem that existing causal feature selection algorithms encounter CI test errors, which seriously degrades the performance of those existing methods. To address this problem, we present an Error-Aware Markov Blanket learning algorithm, EAMB, which contains two novel subroutines: the Efficiently Simultaneous MB (ESMB) and Selectively Recover MB (SRMB) algorithms. ESMB is used to speed up the computational efficiency of EAMB while reducing unreliable CI tests as many as possible, and SRMB utilizes a selective strategy to tackle the unreliable CI test problem caused by the low data efficiency. Finally, EAMB is extensively evaluated and compared with the state-of-the-art causal feature selection algorithms and well-established traditional feature selection methods on real-world datasets. And the results validate the effectiveness and superiority of EAMB in terms of feature selection. Furthermore, we verify the rationality of the selection strategy of SRMB through conducting experiments on benchmark and real-world datasets. In future, we will extend EAMB for learning the local or global causal structures, and selecting causal features from multiple datasets with different distributions [40].

CRediT authorship contribution statement

Xianjie Guo: Investigation, Methodology, Software, Writing - original draft. **Kui Yu:** Writing - original draft, Writing - review & editing, Supervision. **Fuyuan Cao:** Resources, Writing - review & editing. **Peipei Li:** Resources, Writing - review & editing. **Hao Wang:** Resources, Writing - review & editing.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Organizations: Hefei University of Technology, Hefei, 230601, China. Shanxi University, Taiyuan, 030006, China.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (under grant 2020AAA0106100), and National Natural Science Foundation of China (under Grants 61876206 and 62176082).

References

[1] J. Pearl, Probabilistic reasoning in intelligent systems: networks of plausible inference, Elsevier, 2014.
[2] C.F. Aliferis, A.R. Statnikov, I. Tsamardinos, S. Mani, X.D. Koutsoukos, 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 (2010) 171–234.
[3] K. Yu, L. Liu, J. Li, A unified view of causal and non-causal feature selection, ACM Transactions on Knowledge Discovery from Data (TKDD) 15 (4) (2021) 1–46.
[4] D. Margaritis, S. Thrun, Bayesian network induction via local neighborhoods, in: Advances in neural information processing systems, 2000, pp. 505–511..
[5] I. Tsamardinos, C.F. Aliferis, Towards principled feature selection: relevancy, filters and wrappers., in: AISTATS, 2003..
[6] I. Tsamardinos, C.F. Aliferis, A. Statnikov, Time and sample efficient discovery of markov blankets and direct causal relations, in: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 2003, pp. 673–678.
[7] Z. Ling, K. Yu, H. Wang, L. Liu, W. Ding, X. Wu, Bamb: A balanced markov blanket discovery approach to feature selection, ACM Transactions on Intelligent Systems and Technology (TIST) 10 (5) (2019) 1–25.
[8] I. Tsamardinos, L.E. Brown, Bounding the false discovery rate in local bayesian network learning., in: AAAI, 2008, pp. 1100–1105..
[9] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, ACM Computing Surveys (CSUR) 50 (6) (2017) 1–45.
[10] J.R. Vergara, P.A. Estévez, A review of feature selection methods based on mutual information, Neural Computing and Applications 24 (1) (2014) 175–186.
[11] G. Brown, A. Pocock, M.-J. Zhao, M. Luján, Conditional likelihood maximisation: a unifying framework for information theoretic feature selection, The Journal of Machine Learning Research 13 (1) (2012) 27–66.
[12] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, Journal of Machine Learning Research 5 (Oct) (2004) 1205–1224.
[13] I. Rodríguez-Luján, R. Huerta, C. Elkan, C.S. Cruz, Quadratic programming feature selection, Journal of Machine Learning Research 11 (2010) 1491–1516.
[14] C. Lohrmann, P. Luukka, M. Jablonska-Sabuka, T. Kauranne, A combination of fuzzy similarity measures and fuzzy entropy measures for supervised feature selection, Expert Systems with Applications 110 (2018) 216–236.
[15] S. Maldonado, R. Weber, A wrapper method for feature selection using support vector machines, Information Sciences 179 (13) (2009) 2208–2217.
[16] Y. Mohsenzadeh, H. Sheikhzadeh, S. Nazari, Incremental relevance sample-feature machine: A fast marginal likelihood maximization approach for joint feature selection and classification, Pattern Recognition 60 (2016) 835–848.
[17] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society: Series B (Methodological) 58 (1) (1996) 267–288.
[18] I. Guyon, C. Aliferis, A. Elisseeff, Causal feature selection, Computational methods of feature selection (2007) 63–82.
[19] X. Qi, X. Fan, Y. Gao, Y. Liu, Learning bayesian network structures using weakest mutual-information-first strategy, International Journal of Approximate Reasoning 114 (2019) 84–98.
[20] X. Qi, X. Fan, H. Wang, L. Lin, Y. Gao, Mutual-information-inspired heuristics for constraint-based causal structure learning, Information Sciences 560 (2021) 152–167.
[21] X. Wu, B. Jiang, K. Yu, H. Chen, Separation and recovery markov boundary discovery and its application in eeg-based emotion recognition, Information Sciences 571 (2021) 262–278.
[22] A.A. Mastakouri, B. Schölkopf, D. Janzing, Necessary and sufficient conditions for causal feature selection in time series with latent common causes, in: International Conference on Machine Learning, PMLR, 2021, pp. 7502–7511.
[23] D. Koller, M. Sahami, Toward optimal feature selection, Tech. rep., Stanford InfoLab (1996)..
[24] J.-P. Pellet, A. Elisseeff, Using markov blankets for causal structure learning, Journal of Machine Learning Research 9 (Jul) (2008) 1295–1342.
[25] K. Yu, X. Guo, L. Liu, J. Li, H. Wang, Z. Ling, X. Wu, Causality-based feature selection: Methods and evaluations, ACM Computing Surveys (CSUR) 53 (5) (2020) 1–36.
[26] I. Tsamardinos, C.F. Aliferis, A.R. Statnikov, E. Statnikov, Algorithms for large scale markov blanket discovery., in: FLAIRS conference, Vol. 2, 2003, pp. 376–380..
[27] S. Yaramakala, D. Margaritis, Speculative markov blanket discovery for optimal feature selection, in: Fifth IEEE International Conference on Data Mining (ICDM'05), IEEE, 2005, pp. 4–pp..
[28] X. Liu, X. Liu, Swamping and masking in markov boundary discovery, Machine Learning 104 (1) (2016) 25–54.
[29] G. Borboudakis, I. Tsamardinos, Forward-backward selection with early dropping, The Journal of Machine Learning Research 20 (1) (2019) 276–314.
[30] X. Wu, B. Jiang, Y. Zhong, H. Chen, Tolerant markov boundary discovery for feature selection, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 2261–2264.
[31] C.F. Aliferis, I. Tsamardinos, A. Statnikov, Hiton: a novel markov blanket algorithm for optimal variable selection, in: AMIA annual symposium proceedings, Vol. 2003, American Medical Informatics Association, 2003, p. 21..
[32] J.M. Pena, R. Nilsson, J. Björkegren, J. Tegnér, Towards scalable and data efficient learning of markov boundaries, International Journal of Approximate Reasoning 45 (2) (2007) 211–232.
[33] S. Fu, M.C. Desmarais, Fast markov blanket discovery algorithm via local learning within single pass, in: Conference of the Canadian Society for Computational Studies of Intelligence, Springer, 2008, pp. 96–107.

[34] S.R. De Morais, A. Aussem, A novel scalable and data efficient feature subset selection algorithm, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2008, pp. 298–312..
[35] T. Gao, Q. Ji, Efficient markov blanket discovery and its application, IEEE transactions on Cybernetics 47 (5) (2016) 1169–1179.
[36] X. Wu, B. Jiang, K. Yu, H. Chen, et al, Accurate markov boundary discovery for causal feature selection, IEEE Transactions on Cybernetics 50 (12) (2019) 4983–4996.
[37] H. Wang, Z. Ling, K. Yu, X. Wu, Towards efficient and effective discovery of markov blankets for feature selection, Information Sciences 509 (2020) 227–242.
[38] P. Spirtes, C.N. Glymour, R. Scheines, D. Heckerman, Causation, prediction, and search, MIT press, 2000..
[39] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine learning research 7 (Jan) (2006) 1–30.
[40] K. Yu, L. Liu, J. Li, W. Ding, T.D. Le, Multi-source causal feature selection, IEEE transactions on pattern analysis and machine intelligence 42 (9) (2019) 2240–2256, https://doi.org/10.1109/TPAMI.2019.2908373.