Information Sciences 509 (2020) 227–242

Contents lists available at ScienceDirect

Information Sciences

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

Towards efficient and effective discovery of Markov blankets for feature selection

Hao Wang[a,b], Zhaolong Ling[a,b], Kui Yu[a,b,*], Xindong Wu[a,c]

[a] Key Laboratory of Knowledge Engineering with Big Data of Ministry of Education, Hefei University of Technology, Hefei 230009, China
[b] School of Computer and Information, Hefei University of Technology, Hefei 230009, China
[c] Mininglamp Academy of Sciences, Mininglamp Technology, Beijing 100084, China

ARTICLE INFO

Article history:
Received 9 October 2018
Revised 2 September 2019
Accepted 9 September 2019
Available online 10 September 2019

Keywords:
Markov blanket
Bayesian network
Feature selection

ABSTRACT

The Markov blanket (MB), a key concept in a Bayesian network (BN), is essential for large-scale BN structure learning and optimal feature selection. Many MB discovery algorithms that are either efficient or effective have been proposed for addressing high-dimensional data. In this paper, we propose a new algorithm for Efficient and Effective MB discovery, called EEMB. Specifically, given a target feature, the EEMB algorithm discovers the PC (i.e., parents and children) and spouses of the target simultaneously and can distinguish PC from spouses during MB discovery. We compare EEMB with the state-of-the-art MB discovery algorithms using a series of benchmark BNs and real-world datasets. The experiments demonstrate that EEMB is competitive with the fastest MB discovery algorithm in terms of computational efficiency and achieves almost the same MB discovery accuracy as the most accurate of the compared algorithms.

1. Introduction

The Markov blanket (MB) in a Bayesian network (BN) was developed by Pearl [16]. Under the faithfulness assumption (see Definition 3 in Section 3), the MB of a feature (i.e., a node) in a BN consists of the feature's parents, children, and spouses (other parents of the feature's children), as shown in Fig. 1 [27]. MB discovery plays an essential role in scalable BN structure learning and optimal feature selection. For example, by finding the MB of each feature in a dataset, we can use the discovered MBs as constraints to develop efficient and effective local-to-global BN structure learning algorithms [2,23]. Meanwhile, conditioning on the MB of a class attribute in a dataset, all the remaining features are conditionally independent of the class attribute; thus, the MB of the class attribute is theoretically optimal for feature selection [6,25,26,28,29].

Many algorithms for MB discovery have been proposed and can be divided into two main types. Given a target feature, the first type of algorithm finds the parents and children (PC) and spouses of the target simultaneously; for example, GS [15] and IAMB [22]. These methods use the entire set of currently selected features in each iteration as a conditioning set to determine whether to add/remove a feature to/from the currently selected features. This type of method is computationally efficient but requires an exponential number of data samples with respect to the size of the MB of the target. In addition, such MB discovery algorithms cannot distinguish PC from spouses in the discovered MB set.

* Corresponding author at: School of Computer and Information, Hefei University of Technology, Hefei 230009, China.
E-mail addresses: jsjxwangh@hfut.edu.cn (H. Wang), z_dragonl@mail.hfut.edu.cn (Z. Ling), yukui@hfut.edu.cn (K. Yu), wuxindong@mininglamp.com (X. Wu).

https://doi.org/10.1016/j.ins.2019.09.010
0020-0255/

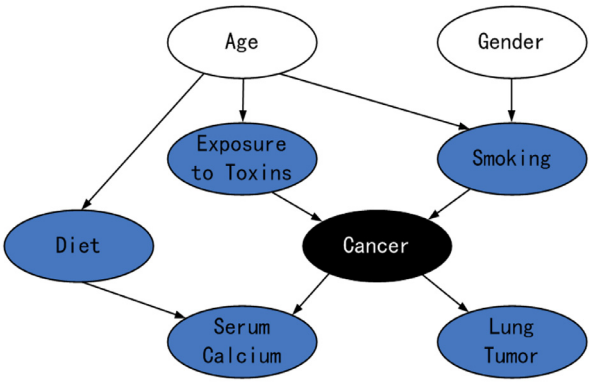

Fig. 1. The Markov blanket (in blue) of the node "Cancer" comprises "Exposure to Toxins" and "Smoking" (parents), "Serum Calcium" and "Lung Tumor" (children), and " Diet" (spouse). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The second type of algorithm first discovers the PC of a given target feature and then finds the spouses of the target; for example, MMMB [21], HITON-MB [3], PCMB [18] and IPCMB [11]. Instead of using the entire set of currently selected features as a conditioning set, these methods perform an exhaustive subset search within the currently selected features to find the PC of the target. Moreover, this type of method must find the PC of each feature in the found PC set of the target to find the spouses of the target. These approaches substantially reduce the required number of data samples and improve the MB discovery performance, especially with high-dimensional data with small-sized data samples. However, this type of algorithm is computationally expensive or even prohibitive when the size of the PC set of the features within the target's PC set is large.

Recently, Gao and Ji [12] proposed the Simultaneous Markov Blanket (STMB) algorithm. Given a target feature, STMB first discovers the PC and then finds the spouses of the target. However, STMB finds the spouses of the target from all features, excluding the found PC set of the target, instead of discovering spouses from the union of the PC set of each feature within the found PC set of the target. Thus, in this case, the time complexity of STMB is related only to the size of the PC set of the target, which makes STMB more efficient than the second type of method. However, STMB adopts the same strategy as the first type of method to remove the false positives in the found PC and spouse sets, which deteriorates the performance of STMB for MB discovery when a dataset is high-dimensional with small-sized data samples.

Accordingly, the question arises of whether we can propose a new method that is both as efficient as the first type of method and as effective as the second type of method. To address this task, in this paper, our main contributions are the following:

1. We propose the EEMB algorithm, a new algorithm for Efficient and Effective discovery of Markov Blankets. The EEMB algorithm integrates the PC discovery strategy of the second type of MB discovery method and the spouse discovery method of STMB. By means of this integration, EEMB discovers the PC and spouses of a given target simultaneously and can distinguish the PC from spouses during MB discovery.
2. On a series of benchmark Bayesian networks and real-world datasets, comparison with the state-of-the-art MB discovery algorithms shows that EEMB is approximately as fast as the first type of method and as accurate as the second type of method, especially with high-dimensional data with small-sized data samples.

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 provides notations and definitions. Section 4 presents the proposed algorithm. Section 5 discusses the experimental results, and Section 6 concludes the paper.

2. Related work

Margaritis and Thrun [15] proposed the first MB discovery algorithm, the Grow-Shrink algorithm (GS), which consists of two steps, growing and shrinking. Given a target feature, in the growing step, GS finds the candidate MBs of the target, and in the shrinking step, GS removes false positives from the found candidate MBs. The Incremental Association Markov Blanket (IAMB) was proposed based on the GS algorithm [22]. The difference between GS and IAMB is the growing step: in the growing step, in each iteration, GS (randomly) selects a feature that is not independent of the target conditioned on the currently selected features, whereas IAMB adopts a dynamic strategy that selects the feature with the highest association with the target conditioned on the currently selected features. Since then, many variants of IAMB, such as inter-IAMB, IAMB-nPC, inter-IAMBnPC, and Fast-IAMB [27], have been proposed. However, the number of data samples required by IAMB and its variants scales exponentially with the size of the MB of the target. In addition, these methods are not able to distinguish the PC from the spouses during MB discovery.

To reduce the number of data samples required for MB discovery, the Min-Max Markov Blanket (MMMB) algorithm [21] implements a divide-and-conquer approach to discover the PC and spouses separately. MMMB performs a subset search within the currently selected features to discover the PC set of the target. HITON-Markov Blanket (HITON-MB) [3] uses the same framework as MMMB but attempts to remove false positives from the PC set as early as possible by interleaving the shrinking phase into the growing phase. Although MMMB and HITON-MB proved to be incorrect under the faithfulness assumption [18], they provide a solid foundation for MB discovery.

The Parents-and-Children-based Markov Blanket algorithm (PCMB) [18] is an algorithm that is proved to be correct under the faithfulness assumption. PCMB uses a symmetry constraint to ensure the correctness of the discovered PC set, but the excessively cautious symmetry constraint checking leads to high computational cost. Accordingly, the Iterative Parent-Child-based search of Markov Blanket (IPCMB) algorithm [11] uses the same method as the PC algorithm [19] to find PC set, which greatly increases the efficiency without sacrificing accuracy. These algorithms greatly improve the effectiveness of MB discovery, especially with a small-sized sample dataset. However, for spouse discovery and symmetry constraint checking, these algorithms need to discover the PC of each feature within the PC set of the target, which is very computationally expensive [12].

The STMB algorithm [12] was proposed to address the problem of computational inefficiency. STMB first finds the PC of the target and then discovers the spouses of the target from the features, excluding the found PC set of the target. This spouse discovery strategy makes STMB efficient. However, when removing false positives from the found PC and spouse sets, instead of performing a subset search within the currently selected features, STMB considers the entire found PC or spouse set as a conditioning set, which may degrade the MB discovery performance of STMB.

3. Notations and definitions

In this section, we will introduce BNs and MBs. Table 1 defines the notation used in this paper.

Definition 1 (Conditional Independence [16])**.** Variable X is conditionally independent of variable Y given Z, if and only if $P(X = x, Y = y|Z = z) = P(X = x|Z = z)P(Y = y|Z = z)$.

Definition 2 (Bayesian Network [16])**.** Let G denote a directed acyclic graph (DAG) defined on U, and let P represent the conditional probability distribution of each feature $X \in U$ given its parents. We call the triplet $< U, G, P >$ a BN if $< U, G, P >$ satisfies the ***Markov condition***: all variables in U are independent of all its non-descendants given its parents.

Definition 3 (Faithfulness. [19])**.** A BN $< U, G, P >$ is faithful, if and only if all conditional independencies between features in G are captured by P.

Definition 3 states that in a faithful BN, if X and Y are d-separated by S in G, then they will be conditionally independent conditioned on S in P.

Definition 4 (V-Structure [16])**.** The triplet of nodes X, Y, and Z forms a v-structure if node Z has two incoming edges from X and Y but X and Y are non-adjacent, e.g., $X \rightarrow Z \leftarrow Y$.

Table 1
Summary of Notations.

Notation	Meaning
U	a feature set
G	a directed acyclic graph over U
P	a joint probability distribution over U
X, Y	a feature
x, y	a discrete value that a feature may take
T	a given target feature in U
Z, S	a conditioning set within U
$X \perp\!\!\!\perp Y \mid Z$	X is conditionally independent of Y given Z
$X \not\perp\!\!\!\perp Y \mid Z$	X is conditionally dependent on Y given Z
$U \backslash \{X\}$	all features in U excluding X
MB_T	a Markov blanket of T
PC_T	parents and children of T
PCS_T	a superset of PC_T
SP_T	spouses of T
$SP_T\{X\}$	a subset of spouses of T with regard to T's child X
$SPS_T\{X\}$	a superset of $SP_T\{X\}$
$CanPC_T$	a candidate PC set of T
$Sep_T\{X\}$	a set that d-separates X from T
IND_T	a set containing all the features independent of T in U
$dep(.)$	a measure of the strength of the dependence
$\|U\|$	the total number of features in U
$\|PC\|$	the largest size of the PC set among all features
$\|SP_T\{X\}\|$	the size of $SP_T\{X\}$

Definition 5 (D-Separation [16]). A collider on a path p is a node with two incoming edges that belong to p. A path between X and Y given a conditioning set $S \subseteq U \backslash \{X \cup Y\}$ is open, if and only if every collider in p is in S or has a descendant in S and no other nodes on p are in S. A path is blocked if it is not open. X and Y are d-separated by S if every path from X to Y is blocked by S.

Definition 6 (Markov Blanket [16]). In a faithful BN, given a target feature T, the Markov blanket of T, MB_T, is unique and consists of the parents, children, and spouses of T.

Property 1 [16]. *In a faithful BN, given MB_T of T, T is conditionally independent of all remaining features in $U \backslash MB_T$, that is, $T \perp\!\!\!\perp X | MB_T$, for $\forall X \in U \backslash \{T \cup MB_T\}$.*

In the following, Theorem 1 demonstrates the relationship between the parents and children in a BN, and Theorem 2 presents the property of a spouse in an MB.

Theorem 1 [17,19]. *In a BN, for any two distinct features $X \in U$ and $Y \in U$, if an edge exists between X and Y, then $\forall S \subseteq U \backslash \{X \cup Y\}$, $X \not\perp\!\!\!\perp Y | S$ holds.*

Since any feature X that has an edge with Y belongs to the PC_Y, Theorem 1 gives rise to an immediate algorithm for identifying PC_Y: for any feature $X \in U \backslash \{Y\}$ and all $S \subseteq U \backslash \{X \cup Y\}$, test whether $X \perp\!\!\!\perp Y | S$. If an S exists for which $X \perp\!\!\!\perp Y | S$, then $X \notin PC_Y$; otherwise, $X \in PC_Y$.

Theorem 2 [17,19]. *In a BN, X, Y, $Z \in U$, X and Y are not adjacent and Z is a collider (e.g., $X \rightarrow Z \leftarrow Y$). If for $\exists S \subseteq U \backslash \{X \cup Y \cup Z\}$ such that $X \perp\!\!\!\perp Y | S$ and $X \not\perp\!\!\!\perp Y | \{S \cup Z\}$, X is a spouse of Y.*

For Theorem 2, we distinguish two cases: (1) X is a spouse of Y and $X \in PC_Y$, for example, $X \rightarrow Z \leftarrow Y$ and $X \rightarrow Y$. In this case, we cannot use Theorem 2 to identify Z as a collider and X as a spouse. However, we do not have to because $X \in PC_Y$, so it will be identified by PC discovery. (2) $X \in MB_Y \backslash PC_Y$, in which case, we can use Theorem 2 to locally discover the subgraph $X \rightarrow Z \leftarrow Y$ and determine that X should be included in MB_Y.

Property 2 (Symmetry constraint *[12]*). *In a BN, if $X \in PC_T$ exists, then $T \in PC_X$ holds.*

4. Proposed algorithm

4.1. Overview of the EEMB algorithm

In this section, we propose the EEMB algorithm, as described in Algorithm 1. EEMB consists of two subroutines: *ADDTrue* (Algorithm 2) and *RMFalse* (Algorithm 3). In the growing phase, the *ADDTrue* subroutine simultaneously discovers a superset of the PC of the target feature T (PCS_T) and a superset of the spouses of T (SPS_T). Then, in the shrinking step, the *RMFalse* subroutine removes false positives from PCS_T and SPS_T.

Algorithm 1 *EEMB.*

Require: T: target; D: dataset
Ensure: [PC_T: Parents and children of T; SP_T: spouses of T]: Markov blanket of T
Phase I: *Growing phase*
1: $[PCS_T, SPS_T] \leftarrow ADDTrue(T, D)$
Phase II: *Shrinking phase*
2: $[PC_T, SP_T] \leftarrow RMFalse(T, PCS_T, SPS_T, D)$

4.2. The ADDTrue subroutine

Before describing the *ADDTrue* subroutine (Algorithm 2) step by step, we discuss the main idea of Algorithm 2.

Strategy. Given a target feature, EEMB proposes a new divide-and-conquer approach to find the PC and spouses of the target simultaneously.

1) *Strategy for finding spouses*: The divide-and-conquer approach is more effective than the first type of MB discovery algorithm [18] but is computationally expensive. The second type of algorithm finds spouses by finding the PC set of each feature in the found PC set of the target, so these algorithms spend much more time on MB discovery than do the first type of algorithms. Compared with these algorithms, STMB efficiently finds spouses from the non-PC set, but STMB still requires time to search the non-PC set to find spouses after finding the PC set. EEMB directly finds spouses from all features that are conditionally independent of T during PC discovery; therefore, EEMB finds the PC and spouses simultaneously. Thus, the time complexity of EEMB depends mainly on the computational cost of PC discovery and not spouse discovery.
2) *Strategy for finding PC*: As discussed above, an efficient strategy to find PC is crucial to improve the efficiency of EEMB. EEMB uses a forward strategy to find PC. Initially, EEMB assumes that the PC set of the target feature is empty and then adds features one by one by selecting the feature with the highest association with a target in each iteration.

Algorithm 2 *ADDTrue* .

Require: T: target; D: dataset
Ensure: PCS_T: superset of PC set of T; SPS_T: superset of spouses of T
1: $PCS_T \leftarrow \emptyset$
2: $CanPC_T \leftarrow U \backslash \{T\}$
3: $IND_T \leftarrow$ all X in $\{U \backslash \{T\}\}$ and $T \perp\!\!\!\perp X | \emptyset$
4: **repeat**
5: $Y \leftarrow argmax_{X \in CanPC_T} dep(T, X|\emptyset)$
6: $PCS_T \leftarrow PCS_T \cup \{Y\}$
7: $CanPC_T \leftarrow CanPC_T \backslash \{Y\}$
8: **for** each $X \in PCS_T$ **do**
9: **if** $T \perp\!\!\!\perp X|Z$ for some $Z \subseteq PCS_T \backslash \{X\}$ **then**
10: $PCS_T \leftarrow PCS_T \backslash \{X\}$
11: $SPS_T\{X\} \leftarrow \emptyset$
12: $Sep_T\{X\} \leftarrow Z$
13: $IND_T \leftarrow IND_T \cup \{X\}$
14: **for** each $A \in PCS_T$ **do**
15: **if** $T \not\perp\!\!\!\perp X|Sep_T\{X\} \cup \{A\}$
16: $SPS_T\{A\} \leftarrow SPS_T\{A\} \cup \{X\}$
17: **end if**
18: **end for**
19: **else if** X is the last feature added to PCS_T **then**
20: **for** each $B \in IND_T$
21: **if** $T \not\perp\!\!\!\perp B|Sep_T\{B\} \cup \{X\}$ **then**
22: $SPS_T\{X\} \leftarrow SPS_T\{X\} \cup \{B\}$
23: **end if**
24: **end for**
25: **end if**
26: **end for**
27: **until** $CanPC_T$ is empty

Meanwhile, when a new feature is added to the current set of selected features, the EEMB checks the currently selected features and removes false positives. The forward strategy can keep the number of currently selected features as small as possible in each iteration by removing false positive features when adding features. This strategy is beneficial because conditioning on larger sets of features would increase the risk of missing features that are weakly associated with the target [8]. Compared to the backward strategy used by STMB for PC discovery, the forward strategy is similar in terms of computational complexity (we will discuss the computational complexity of the forward strategy and backward strategy in IV-E), but the runtime of the forward strategy is shorter than that of the backward strategy when finding the PC set because most BNs have a large number of features but a small-sized PC set of each feature.

Description. For each feature in $\{U \backslash \{T\}\}$ (line 2), EEMB first checks whether X is conditionally dependent on T and then adds all features in U that are conditionally independent of T given the empty set into IND_T (line 3). The features with the strongest association with T (line 5) are added to the PCS_T set in each iteration (line 6). Subsequently, EEMB checks each feature X in PCS_T. The following two cases can occur.

- Case 1: If feature X is conditionally independent of T (line 9), then X is removed from PCS_T (line 10). There is no need to continue checking whether the features in the PCS_T set belong to the PCS_T (this idea is also applied to the removing phase). The spouses ($SPS_T\{X\}$) that have a corresponding child feature X with T must also be removed (line 11) since feature X has been removed. Meanwhile, EEMB keeps the set that d-separates X from T (line 12) and adds X into IND_T to update the set containing all the features independent of T in U (line 13).
  Moreover, EEMB considers whether the just removed feature X belongs to the spouses of T corresponding to each feature within PCS_T (lines 14–18): if $T \not\perp\!\!\!\perp X|Sep_T\{X\} \cup \{A\}$ and $A \in PCS_T$, feature X could be a spouse of T with regard to T's child A (line 16).
- Case 2: If feature X is conditionally dependent on T, then X belongs to PCS_T, and when checking the features in PCS_T, only the newly generated subsets by the addition of X must be tested (this optimization is also applied to the removing phase).
  Moreover, if X is the last feature added to PCS_T (line 19), EEMB directly finds the spouses from all features in U that are conditionally independent of T (lines 20–24): if $T \not\perp\!\!\!\perp B|Sep_T\{B\} \cup \{X\}$, feature B could be a spouse of T with regard to T's child X (line 22).

Algorithm 3 *RMFalse*.

Require: T: target; D: dataset; PCS_T: superset of PC set of T; SPS_T: superset of spouses of T
Ensure: PC_T: Parents and children of T; SP_T: spouses of T

Phase I: *Remove false positives from* SPS_T

for each Y in SPS_T **do**
  $SP_T\{Y\} \leftarrow \emptyset$
  repeat
    $A \leftarrow argmax_{X \in SPS_T\{Y\}} dep(T, X|Sep_T\{X\} \cup \{Y\})$
    $SP_T\{Y\} \leftarrow SP_T\{Y\} \cup \{A\}$
    $SPS_T\{Y\} \leftarrow SPS_T\{Y\} \backslash \{A\}$
    for each $X \in SP_T\{Y\}$ **do**
      if $T \perp\!\!\!\perp X|Z \cup Y$ for some $Z \in SP_T\{Y\} \backslash \{X\}$ **then**
        $SP_T\{Y\} \leftarrow SP_T\{Y\} \backslash \{X\}$
      end if
    end for
  until $SPS_T\{Y\}$ is empty
  $SPS_T\{Y\} \leftarrow SP_T\{Y\}$
  $SP_T\{Y\} \leftarrow \emptyset$
  repeat
    $B \leftarrow argmax_{X \in SPS_T\{Y\}} dep(T, X|Sep_T\{X\} \cup \{Y\})$
    $SP_T\{Y\} \leftarrow SP_T\{Y\} \cup \{B\}$
    $SPS_T\{Y\} \leftarrow SPS_T\{Y\} \backslash \{B\}$
    for each $X \in SP_T\{Y\}$ **do**
      if $T \perp\!\!\!\perp X|Z \cup Y$ for some $Z \subseteq PCS_T \cup SP_T\{Y\} \backslash \{X\}$ **then**
        $SP_T\{Y\} \leftarrow SP_T\{Y\} \backslash \{X\}$
      end if
    end for
  until $SPS_T\{Y\}$ is empty
end for

Phase II: *Remove false positives from* PCS_T

$PC_T \leftarrow \emptyset$
repeat
  $C \leftarrow argmax_{X \in PCS_T} dep(T, X|\emptyset)$
  $PC_T \leftarrow PC_T \cup \{C\}$
  $PCS_T \leftarrow PCS_T \backslash \{C\}$
  for each $X \in PC_T$ **do**
    if $T \perp\!\!\!\perp X|Z \cup_{Y \in Z} SP_T\{Y\}$ for some $Z \subseteq PC_T \backslash \{X\}$ **then**
      $PC_T \leftarrow PC_T \backslash \{X\}$
      $SP_T\{X\} \leftarrow \emptyset$
    end if
  end for
until PCS_T is empty

However, some false positives will be added to PCS_T and SPS_T after Algorithm 2. For example, as shown in Fig. 2, by Definition 5, the non-child descendant X is in PCS_T, and the spouses' parent Y remains in SPS_T. Algorithm 3 is proposed in the next section to remove these false positives.

4.3. The RMFalse subroutine

In the section, we first present the main idea of the *RMFalse* subroutine (Algorithm 3) and then describe the subroutine step by step.

Strategy. The second type of algorithm removes false positives using the symmetry constraint by finding the PC set of each feature in the found PC set of T. These algorithms are effective but inefficient. Compared with these algorithms, the first type of algorithm and STMB sacrifice accuracy for efficiency, especially in high-dimensional datasets with small-sized data samples, by using the entire set of currently selected features as a conditioning set to remove false positives. By contrast, EEMB removes false positives by performing a subset search within the found MB to improve the effectiveness. Moreover, EEMB uses a forward strategy in this phase to achieve far better efficiency and effectiveness.

Description. EEMB divides the phase of removing false positives into two steps: removing false spouses and removing false PC.

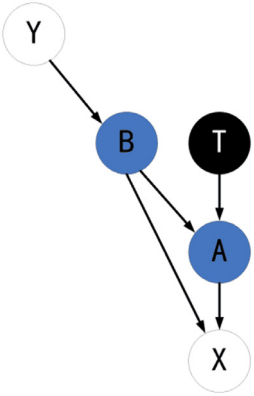

Fig. 2. $\mathbf{PC}_T = \{A\}, X \notin \mathbf{PC}_T$. Note that no subsets within the PC of T make X conditionally independent of T: $X \not\perp\!\!\!\perp T|\emptyset$, $X \not\perp\!\!\!\perp T|A$. In addition, $\mathbf{SP}_T = \{B\}, Y \notin \mathbf{SP}_T$. Since $Y \perp\!\!\!\perp T|\emptyset$ and $Y \not\perp\!\!\!\perp T|A, Y$ could be a spouse of T. However, the MB of T can make X and Y conditionally independent of T: $X \perp\!\!\!\perp T|\{A \cup B\}$, $Y \perp\!\!\!\perp T|\{A \cup B\}$.

- Remove false positive spouses from SPS_T: In lines 2–12, EEMB aims to filter false positive spouses and thus considers the union of Y and the subsets within $SP_T\{Y\}$ as the conditioning set (line 8). Then, in lines 13–24, EEMB removes all false spouses and considers the subset of the union of PCS_T and $SP_T\{Y\}$ as a conditioning set (line 20).
- Remove false positives from PCS_T: EEMB removes false PC by conditioning on the subsets within the currently selected features within PCS_T and SPS_T (line 32). $SP_T\{X\}$ must be removed since X is conditionally independent of T (line 34).

For example, as shown in Fig. 2, the MB of T can make X and Y conditionally independent of T. Consequently, the false positive Y will be removed at line 9 or 21, and the false positive X will be removed at line 33. Thus, no false positives exist in PC_T and SP_T.

4.4. Correctness of EEMB

Theorem 3. *Under the faithfulness assumption, EEMB finds the exact MB of a given target.*

Proof. In the growing phase, *ADDTrue* (Algorithm 2) finds all true MB of T. EEMB adds the features that are conditionally dependent on T given an empty set to the PCS_T (line 6) and removes some false positives of PC from PCS_T by conditioning on the subset within PCS_T based on Theorem 1 (line 10). According to Theorem 1, the feature $X \in PC_T$ is conditionally dependent on T given all $S \subseteq U \backslash \{X \cup T\}$; thus, PCS_T contains all the true PC of T. While finding PCS_T, EEMB finds the spouses of T simultaneously, but this process does not affect finding PCS_T. If feature Z is a collider that forms a V-structure, $Y \rightarrow Z \leftarrow T$, then feature Y is considered a candidate spouse of T via Z by Theorem 2 (lines 16 and 22). Due to the exhaustive search, EEMB would not miss any true positive spouses among all features in U that are conditionally independent of T given an empty set (line 3) or some false PC nodes (line 9); thus, we find all true positive spouses. Consequently, PCS_T and SPS_T together contain all the true PC and spouses of T.

In the shrinking phase, Algorithm 3 removes false positives in the PC and spouse set found in the growing phase. As shown in Fig. 2, two types of false positives exist in the found MB: (1) non-child descendants of T in PCS_T (e.g., node X); and (2) parents of the spouses of T in SPS_T (e.g., node Y). EEMB directly applies Property 1 to remove all false spouses from SPS_T (lines 8 and 20) because conditional independence relationships with T given some PC nodes and spouses (i.e., a candidate set of MB) would indicate false MB nodes. Then, EEMB contains only the true spouses in SP_T as the exhaustive test ensures that no false positive spouses remain. The subsets within PCS_T and the corresponding $SP_T\{Y\}$ together constitute the complete MB of T since PCS_T contains all true PC; thus, EEMB removes non-child descendant nodes of T owing to Property 1 (line 32). Consequently, PC_T and SP_T together contain all and only the true positive PC set and spouses after Algorithm 3. In other words, PC_T and SP_T together contain all and only the true positive MB nodes. □

4.5. Computational complexity

Since almost all state-of-the-art MB discovery algorithms employ conditional independence tests (CITs) to calculate the dependence/independence of features [1], in this paper, we use the number of CITs to represent the computational complexity of the EEMB algorithm and its rivals. Algorithm 2 first sorts the features using $|U|$ measures of association and then performs an exhaustive subset search in the currently selected PC set in each iteration. Thus, the computational complexity of Algorithm 2 is $O(|U|2^{|PC|})$ CITs. The computational complexity of phase I of Algorithm 3 is $O(|SP_T\{X\}|2^{(|PC|+|SP_T\{X\}|)})$ CITs, and the complexity of phase II of Algorithm 3 is $O(|PC|2^{|PC|})$ CITs. Overall, the computational complexity of EEMB is $O(|U|2^{|PC|} + |SP_T\{X\}|2^{(|PC|+|SP_T\{X\}|)} + |PC|2^{|PC|}) = O(|U|2^{|PC|})$ CITs.

Table 2
Computational Complexity of Each Markov Blanket Discovery Algorithm.

Algorithms	IAMB	MMMB	HITON-MB	PCMB	IPCMB	STMB	EEMB
Complexity	$O(\|U\|^2)$	$O(\|U\|\|PC\|2^{\|PC\|})$	$O(\|U\|\|PC\|2^{\|PC\|})$	$O(\|U\|\|PC\|^2 2^{\|PC\|})$	$O(\|U\|\|PC\|2^{\|U\|})$	$O(\|U\|2^{\|U\|})$	$O(\|U\|2^{\|PC\|})$

The algorithm used to enforce the symmetry constraint requires additional $|PC|$ CITs. Specifically, the forward strategy used by Algorithm 2 tests $T \perp\!\!\!\perp X|Z$ for all subsets Z of the largest PC set in each iteration, and the backward strategy used by STMB for PC discovery tests $T \perp\!\!\!\perp X|Z$ for all subsets Z of the entire set in each iteration. Consequently, Algorithm 2 requires $O(|U|2^{|PC|})$ CITs. The backward strategy used by STMB for PC discovery requires $O(|U|2^{|U|})$ CITs, and the forward strategy is much faster than the backward strategy due to $|PC| \ll |U|$.

The computational complexities of the state-of-the-art MB discovery algorithms are summarized in Table 2. Table 2 shows that IAMB is the fastest, and EEMB is the second-fastest and is very close to IAMB in terms of computational efficiency.

5. Experiments

In this section, we compare the EEMB algorithm with the state-of-the-art MB discovery algorithms in terms of efficiency and effectiveness using six benchmark BNs and ten real-world datasets. The compared MB discovery algorithms are IAMB [22], MMMB [21], HITON-MB [3], PCMB [18], IPCMB [11] and STMB [12]. All algorithms are implemented in MATLAB. All experiments are conducted in Windows 7 with an Intel Core i5-5200U with 8 GB RAM. The conditional independence test is the G^2 test at the 0.01 significance level. The best results are highlighted in bold face in the tables.

To determine whether EEMB and its rivals have significantly different accuracy, we conduct the Friedman test at the 5% significance level [9]. The null hypothesis states that the performance of EEMB and that of its rivals is not significantly different. The average ranks are calculated using the Friedman test (for the calculation of the average ranks, please see [9]). When the null hypothesis of the Friedman test is rejected, we proceed with the Nemenyi test [9] as a post hoc test. In the Nemenyi test, the performance of two methods is significantly different if the corresponding average ranks differ by at least the critical difference (for calculating the critical difference, please see [9]).

5.1. Benchmark BN datasets

The six benchmark BN datasets are described in Table 3.[1] For each benchmark BN network, we use two groups of data, one group includes 10 datasets with 500 data instances to represent small-sized dataset samples, and the other group contains 10 datasets with 5000 data instances to represent large-sized dataset samples. The MB of each feature can be read from the benchmark BN networks. Accordingly, in the experiments, we evaluate the algorithms with respect to the following metrics.

- Effectiveness. $F1 = 2 * precision * recall/(precision + recall)$. Precision denotes the number of true positives in the output (i.e., the features in the output of an algorithm belonging to the true MB of a given target in a test DAG) divided by the total number of features in the output of the algorithm. Recall represents the number of true positives in the output divided by the number of true positives (the number of true MB of a given target) in a test DAG. The F1 score is the harmonic average of precision and recall, where $F1 = 1$ is the best case (perfect precision and recall) and $F1 = 0$ is the worst case.
- Efficiency. We measure the efficiency of an algorithm using both the number of CITs and runtime.

For each algorithm, we report the average F1, precision, recall, number of CITs, and runtime for ten datasets. In the following tables, the results are shown in the format of $A \pm B$, where A represents the average F1, precision, recall, number of CITs, or runtime, and B is the standard deviation.

Using small-sized BNs. We report the results of EEMB and its rivals on three small-sized networks, Child [7], Insurance [5] and Alarm [4]. We run each algorithm to discover the MBs for all nodes in each BN. Table 4 summarizes the F1, precision, recall, number of CITs, and runtime of each network for different sample sizes, and Table 5 summarizes the average results for different sample sizes.

On the Child network with small-sized data samples, EEMB is the most accurate in terms of the F1 metric and is competitive with IAMB in terms of computational efficiency (number of CITs corresponds to the runtime). On the Child network with large-sized data samples, EEMB has comparable accuracy to that of HITON-MB, but EEMB is much faster (more than 17 times) than HITON-MB in terms of the number of CITs. On the Insurance network, MMMB and EEMB are the most accurate algorithms with both small-sized and large-sized data samples. Meanwhile, EEMB has comparable running speed to that of IAMB and is much faster (over 12 times) than HITON-MB in terms of CITs on large-sized sample datasets. On the Alarm

[1] These datasets are publicly available at http://www.dsl-lab.org/supplements/mmhc_paper/mmhc_index.html.

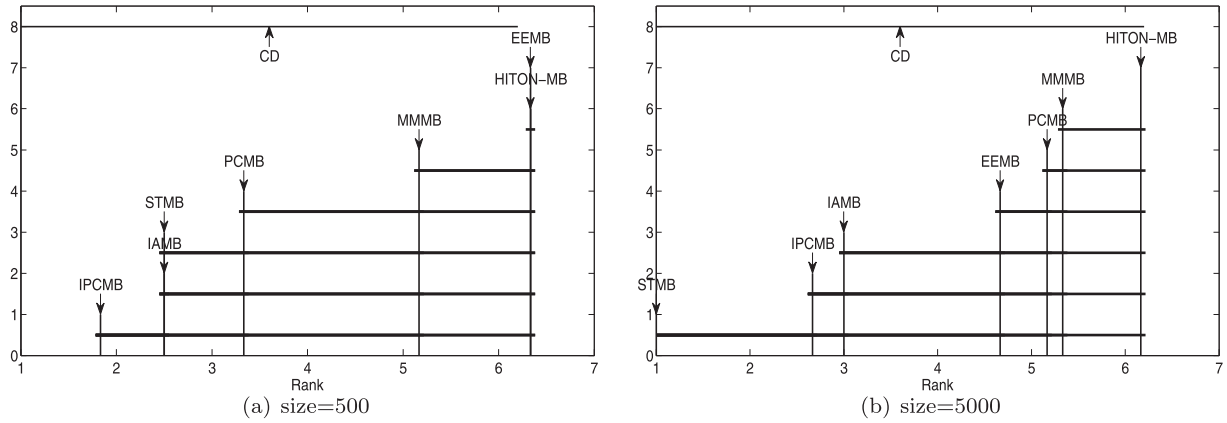

Fig. 3. Crucial difference diagram of the Nemenyi test of the F1 metric for small-sized BNs.

network, EEMB achieves competitive F1 values with HITON-MB and PCMB and is faster. Although IAMB is faster than EEMB, it is significantly inferior to EEMB in terms of the F1 metric in all cases.

To further compare the accuracy of EEMB with that of IAMB, MMMB, HITON-MB, PCMB, IPCMB, and STMB on small-sized networks, we conduct the Friedman test at the 5% significance level. The null hypothesis states that the performance of EEMB and that of its rivals is the same. The average rank of EEMB against its rivals in terms of F1 is shown in the last column of Table 5. With small-sized data samples, the null hypothesis is rejected, and the average ranks for IAMB, MMMB, HITON-MB, PCMB, IPCMB, STMB, and EEMB are 2.50, 5.17, 6.33, 3.33, 1.83, 2.50, and 6.33, respectively (the higher the average rank is, the better the prediction accuracy). Then, we proceed with the Nemenyi test as a post hoc test. The Nemenyi test indicates that the performance of the seven methods is significantly different if the corresponding average ranks differ by at least the critical difference, which is 5.20. Thus, EEMB is significantly better than IPCMB on small-sized sample datasets. For large-sized data samples, the null hypothesis is rejected, and the average ranks for IAMB, MMMB, HITON-MB, PCMB, IPCMB, STMB, and EEMB are 3.00, 5.33, 6.17, 5.17, 2.67, 1.0, and 4.67, respectively. Thus, HITON-MB is significantly better than STMB on large-sized sample datasets. We plot the crucial difference diagram of the Nemenyi test in Fig. 3.

Using large-sized BNs. We validate our proposed EEMB algorithm using three large-sized networks: Child10, Insurance10, and Alarm10. These three networks were generated by tiling 10 copies of the Child, Insurance, and Alarm networks, respectively [20]. We randomly select 10% of the features in each BN, find their MBs, and report the F1, precision, recall, number of CITs, and runtime. Table 6 summarizes the results of EEMB and its rivals using different data samples, and Table 7 summarizes the average results.

On large-sized networks, EEMB has comparable speed with that of IAMB but is much faster than the other algorithms. In terms of the F1 metric, EEMB illustrates comparable accuracy with HITON-MB. EEMB is the most accurate on small-sized sample data.

Meanwhile, we use the Friedman test at the 5% significance level to further evaluate the accuracy, and we summarize the average ranks of EEMB against its rivals in terms of F1 in the last column of Table 7. For small-sized data samples, the null hypothesis is rejected, and the average ranks for IAMB, MMMB, HITON-MB, PCMB, IPCMB, STMB, and EEMB are 2.67, 5.50, 5.83, 4.00, 2.33, 1.00, and 6.67, respectively. Then, we proceed with the Nemenyi test as a post hoc test. In the Nemenyi test, the critical difference is 5.20. Thus, EEMB has significantly better accuracy than STMB for small-sized data samples. For large-sized data samples, the null hypothesis is rejected, and the average ranks for IAMB, MMMB, HITON-MB, PCMB, IPCMB, STMB, and EEMB are 2.67, 5.50, 6.50, 4.33, 2.67, 1.00, and 5.33, respectively. Then, we proceed with the Nemenyi test as a post hoc test and conclude that HITON-MB is significantly more accurate than STMB for large-sized data samples. We plot the crucial difference diagram of the Nemenyi test in Fig. 4.

Summary. For the benchmark BN datasets, as discussed in Section 4.5 and according to the detailed information of each dataset provided in the Table 3, the experimental results are mostly consistent with the computational complexity analysis. The differences in efficiency and effectiveness for different numbers of data samples can be contributed to the fact that small-sized sample datasets could introduce more erroneous dependencies [12].

For the benchmark BNs, HITON-MB is the most accurate algorithm while IAMB is the fastest. For small-sized datasets, EEMB is the most accurate. In summary, EEMB is approximately as fast as IAMB and almost as accurate as HITON-MB.

5.2. Real-World datasets

In addition to the benchmark BNs, in this section, we use ten real-world datasets with low to high dimensionality, as shown in Table 8. Since the MBs in the real-world datasets are unknown, we use the MBs discovered by the EEMB algorithm and its rivals for feature selection for classification [14]. In Table 8, the first five datasets are from the UCI machine learning

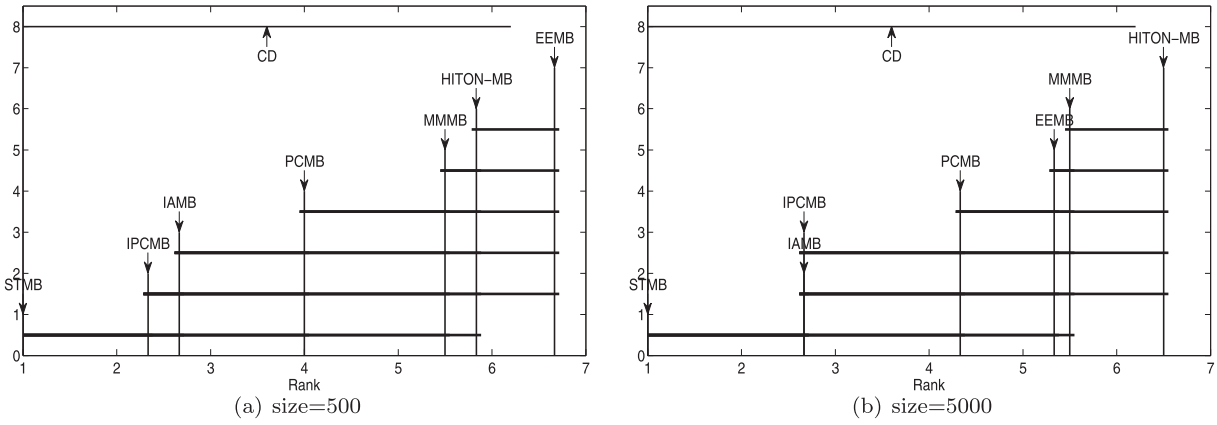

Fig. 4. Crucial difference diagram of the Nemenyi test for the F1 metric on large-sized BNs.

Table 3
Summary of Benchmark BNs.

Network	Num. Vars	Num. Edges	Max In-/out-Degree	Min/Max |PCset|	Domain Range
Child	20	25	2/7	1/8	2–6
Insurance	27	52	3/7	1/9	2–5
Alarm	37	46	4/5	1/6	2–4
Child10	200	257	2/7	1/8	2–6
Insurance10	270	556	5/8	1/11	2–5
Alarm10	370	570	4/7	1/9	2–4

repository [10], the *hiva, ovariancancer*, and *breastcancer* datasets are three biomedical datasets [13,24], and the *madelon* and *dexter* datasets are from the NIPS 2003 feature selection challenge. To avoid bias in the error estimation we apply 10-fold cross-validation for all datasets. We use the following metrics for the feature selection evaluation.

- Effectiveness. Compactness is the number of selected features. Prediction accuracy is the percentage of the correctly classified test instances that were previously unseen. We report both compactness and the prediction accuracy of the KNN classifier and SVM classifier[2] as the effectiveness measures of the different algorithms.
- Efficiency. We report both the number of CITs and runtime as the efficiency measures of the different algorithms.

Tables 9 and 10 summarize the results of the feature selection by the different algorithms. The results are shown in the format of $A \pm B$, where A represents the accuracy, compactness, number of CITs, or time, and B is the corresponding standard deviation. Table 11 reports the win/tie/loss counts of EEMB against other algorithms. The results are shown in the format of $A/B/C$: A denotes the number of times EEMB outperforms the other algorithms on ten datasets while B and C represent the numbers of ties and losses, respectively. Table 12 shows the average results on 10 real-world datasets.

Performance. In terms of prediction accuracy, Table 11 shows that EEMB is superior to IAMB, MMMB, HITON-MB, PCMB, IPCMB, and STMB on most datasets when using SVM. Furthermore, the prediction accuracy of EEMB is never worse than that of the other algorithms when using KNN. Tables 9 and 10 show that for the first five datasets, the accuracy advantage of EEMB is not obvious due to the small number of features. However, for the latter five feature sets with more than 500 features, the accuracy of EEMB is much higher than that of the other algorithms. In Table 12, on average, EEMB is the most accurate algorithm, and EEMB is 9% and 5% more accurate than the least accurate IAMB when using KNN and SVM, respectively. Moreover, IAMB is much worse than MMMB, HITON-MB, and EEMB in terms of average prediction accuracy. For the four datasets *heart, spectf, sonar*, and *dexter*, IAMB is 10% less accurate than EEMB when using KNN.

To further evaluate the prediction accuracy of EEMB against its rivals, we conduct the Friedman test at the 5% significance level. The null hypothesis states that the prediction accuracy of EEMB and that of its rivals is the same. The average ranks of EEMB against its rivals when using KNN and SVM are summarized in the last two columns of Table 12. For KNN, the null hypothesis is rejected, and the average ranks for IAMB, MMMB, HITON-MB, and EEMB are 1.55, 2.10, 2.70, and 3.65, respectively. Then, we proceed with the Nemenyi test as a post hoc test. In the Nemenyi test, the performance of the four methods is significantly different if the corresponding average ranks differ by at least the critical difference, which is 1.48. Thus, EEMB is significantly better than IAMB and MMMB when using the KNN classifier. We plot the crucial difference diagram of the Nemenyi test in Fig. 5. For SVM, the null hypothesis is accepted, and the average ranks for IAMB, MMMB,

[2] The LIBSVM library is available at www.csie.ntu.edu.tw/~cjlin/libsvm.

Table 4
F1, Precision, Recall, Number of CITs, and Runtime (in Seconds) on Small-Sized BNs Using Different Data Sizes.

Size	Dataset	Algorithm	F1	Precision	Recall	CITs	Time
		IAMB	0.75 ± 0.03	**0.95 ± 0.03**	0.67 ± 0.03	**49 ± 1**	**0.01 ± 0.00**
		MMMB	0.77 ± 0.04	0.78 ± 0.05	0.85 ± 0.05	1120 ± 254	0.16 ± 0.03
		HITON-MB	0.79 ± 0.04	0.79 ± 0.04	**0.87 ± 0.05**	3852 ± 1241	0.39 ± 0.10
	Child	PCMB	0.73 ± 0.03	0.79 ± 0.04	0.77 ± 0.04	7405 ± 2257	0.95 ± 0.22
		IPCMB	0.69 ± 0.04	0.68 ± 0.06	0.80 ± 0.04	1184 ± 92	0.20 ± 0.01
		STMB	0.77 ± 0.04	0.81 ± 0.05	0.79 ± 0.03	215 ± 7	0.05 ± 0.00
		EEMB	**0.81 ± 0.04**	0.84 ± 0.04	0.83 ± 0.04	176 ± 22	0.04 ± 0.00
		IAMB	0.56 ± 0.03	**0.89 ± 0.04**	0.45 ± 0.02	**75 ± 2**	**0.02 ± 0.00**
		MMMB	**0.62 ± 0.03**	0.68 ± 0.04	0.61 ± 0.03	513 ± 105	0.13 ± 0.02
		HITON-MB	**0.62 ± 0.03**	0.71 ± 0.04	0.61 ± 0.03	1069 ± 238	0.24 ± 0.04
500	Insurance	PCMB	0.57 ± 0.02	0.68 ± 0.02	0.54 ± 0.03	2663 ± 842	0.65 ± 0.15
		IPCMB	0.56 ± 0.02	0.54 ± 0.04	**0.66 ± 0.07**	41,819 ± 35,409	4.68 ± 3.86
		STMB	0.56 ± 0.03	0.59 ± 0.08	0.61 ± 0.05	1811 ± 1338	0.24 ± 0.15
		EEMB	**0.62 ± 0.02**	0.75 ± 0.03	0.58 ± 0.02	174 ± 19	0.05 ± 0.01
		IAMB	0.73 ± 0.04	**0.90 ± 0.05**	0.66 ± 0.04	**109 ± 2**	**0.03 ± 0.00**
		MMMB	0.78 ± 0.04	0.81 ± 0.04	0.80 ± 0.04	480 ± 74	0.12 ± 0.01
		HITON-MB	**0.80 ± 0.03**	0.84 ± 0.03	**0.82 ± 0.03**	1631 ± 293	0.33 ± 0.04
	Alarm	PCMB	0.74 ± 0.03	0.80 ± 0.04	0.74 ± 0.03	2397 ± 469	0.61 ± 0.09
		IPCMB	0.73 ± 0.03	0.74 ± 0.03	0.77 ± 0.04	879 ± 37	0.25 ± 0.01
		STMB	0.65 ± 0.04	0.69 ± 0.05	0.72 ± 0.04	324 ± 10	0.09 ± 0.00
		EEMB	0.79 ± 0.03	0.84 ± 0.04	0.79 ± 0.03	178 ± 8	0.05 ± 0.00
		IAMB	0.90 ± 0.02	0.95 ± 0.03	0.88 ± 0.01	**63 ± 1**	**0.06 ± 0.00**
		MMMB	0.97 ± 0.01	0.96 ± 0.02	**0.99 ± 0.01**	897 ± 25	0.96 ± 0.03
		HITON-MB	**0.98 ± 0.02**	0.97 ± 0.03	**0.99 ± 0.01**	2771 ± 112	3.08 ± 0.12
	Child	PCMB	**0.98 ± 0.01**	**0.98 ± 0.01**	**0.99 ± 0.01**	5031 ± 106	5.49 ± 0.13
		IPCMB	0.96 ± 0.02	0.95 ± 0.03	**0.99 ± 0.01**	1877 ± 155	1.94 ± 0.17
		STMB	0.89 ± 0.03	0.84 ± 0.04	0.98 ± 0.02	374 ± 35	0.39 ± 0.04
		EEMB	0.95 ± 0.02	0.95 ± 0.03	0.98 ± 0.02	161 ± 4	0.17 ± 0.00
		IAMB	0.76 ± 0.01	**0.94 ± 0.02**	0.67 ± 0.01	**104 ± 2**	**0.13 ± 0.00**
		MMMB	**0.79 ± 0.02**	0.88 ± 0.03	0.76 ± 0.02	1186 ± 124	1.63 ± 0.18
		HITON-MB	0.78 ± 0.02	0.89 ± 0.03	0.74 ± 0.02	3175 ± 414	4.47 ± 0.62
5000	Insurance	PCMB	0.74 ± 0.01	0.86 ± 0.02	0.68 ± 0.02	7206 ± 1145	10.00 ± 1.68
		IPCMB	0.66 ± 0.03	0.64 ± 0.03	0.74 ± 0.03	3509 ± 449	4.67 ± 0.61
		STMB	0.65 ± 0.02	0.64 ± 0.04	**0.77 ± 0.03**	703 ± 47	0.96 ± 0.07
		EEMB	**0.79 ± 0.01**	0.89 ± 0.02	0.75 ± 0.02	267 ± 12	0.37 ± 0.02
		IAMB	0.90 ± 0.02	0.94 ± 0.02	0.89 ± 0.01	**142 ± 2**	**0.19 ± 0.00**
		MMMB	0.94 ± 0.02	0.92 ± 0.02	**0.97 ± 0.01**	604 ± 26	0.83 ± 0.04
		HITON-MB	**0.96 ± 0.01**	**0.97 ± 0.02**	**0.97 ± 0.01**	1543 ± 38	2.15 ± 0.06
	Alarm	PCMB	0.95 ± 0.02	0.95 ± 0.01	0.96 ± 0.02	2870 ± 215	3.97 ± 0.32
		IPCMB	0.86 ± 0.02	0.81 ± 0.02	**0.97 ± 0.01**	1656 ± 54	2.18 ± 0.07
		STMB	0.78 ± 0.02	0.73 ± 0.02	0.96 ± 0.01	531 ± 15	0.73 ± 0.03
		EEMB	0.94 ± 0.02	**0.97 ± 0.03**	0.94 ± 0.01	212 ± 4	0.29 ± 0.01

Table 5
Average Results on Small-Sized BNs Using Different Data Sizes.

Size	Algorithm	F1	Precision	Recall	CITs	Time	Rank-F1
	IAMB	0.68	**0.91**	0.59	**78**	**0.02**	2.50
	MMMB	0.72	0.76	0.75	704	0.14	5.17
	HITON-MB	0.73	0.78	**0.77**	2184	0.32	**6.33**
500	PCMB	0.68	0.76	0.68	4155	0.74	3.33
	IPCMB	0.66	0.65	0.74	14,627	1.71	1.83
	STMB	0.66	0.70	0.71	783	0.13	2.50
	EEMB	**0.74**	0.81	0.73	176	0.05	**6.33**
	IAMB	0.85	**0.94**	0.81	**103**	**0.13**	3.00
	MMMB	0.90	0.92	**0.91**	896	1.14	5.33
	HITON-MB	**0.91**	**0.94**	0.90	2496	3.23	**6.17**
5000	PCMB	0.89	0.93	0.88	5036	6.49	5.17
	IPCMB	0.83	0.80	0.90	2347	2.93	2.67
	STMB	0.77	0.74	0.90	536	0.69	1.00
	EEMB	0.89	**0.94**	0.89	213	0.28	4.67

Table 6
F1, Precision, Recall, Number of CITs, and Runtime (in Seconds) on Large-Sized BNs Using Different Data Sizes.

Size	Dataset	Algorithm	F1	Precision	Recall	CITs	Time
		IAMB	0.52 ± 0.24	**0.81 ± 0.32**	0.43 ± 0.27	**557 ± 174**	**0.21 ± 0.06**
		MMMB	0.60 ± 0.26	0.70 ± 0.33	0.61 ± 0.30	1850 ± 1163	0.71 ± 0.47
		HITON-MB	0.60 ± 0.26	0.69 ± 0.34	0.61 ± 0.30	5316 ± 3824	1.32 ± 0.91
	Child10	PCMB	0.57 ± 0.29	0.67 ± 0.37	0.58 ± 0.33	11,291 ± 10,334	3.75 ± 3.19
		IPCMB	0.55 ± 0.27	0.59 ± 0.33	0.60 ± 0.33	5122 ± 4051	2.68 ± 2.05
		STMB	0.44 ± 0.27	0.39 ± 0.31	**0.65 ± 0.36**	2015 ± 1540	0.90 ± 0.50
		EEMB	**0.61 ± 0.30**	0.65 ± 0.34	0.64 ± 0.32	1503 ± 1648	0.56 ± 0.54
		IAMB	0.39 ± 0.22	**0.72 ± 0.35**	0.29 ± 0.19	**799 ± 240**	**0.52 ± 0.16**
		MMMB	**0.47 ± 0.26**	0.61 ± 0.34	0.46 ± 0.32	1948 ± 1176	1.40 ± 0.82
		HITON-MB	**0.47 ± 0.26**	0.60 ± 0.33	0.46 ± 0.32	4344 ± 3288	2.73 ± 1.99
500	Insurance10	PCMB	0.46 ± 0.26	0.64 ± 0.36	0.42 ± 0.30	10,657 ± 7408	7.53 ± 5.17
		IPCMB	0.33 ± 0.18	0.27 ± 0.15	0.55 ± 0.30	82,483 ± 237,186	63.44 ± 181.53
		STMB	0.24 ± 0.13	0.17 ± 0.09	**0.56 ± 0.32**	7606 ± 4419	5.53 ± 2.85
		EEMB	**0.47 ± 0.26**	0.54 ± 0.31	0.46 ± 0.29	1439 ± 940	1.52 ± 0.95
		IAMB	0.51 ± 0.26	0.77 ± 0.32	0.43 ± 0.29	**1190 ± 342**	**0.97 ± 0.26**
		MMMB	0.54 ± 0.27	0.73 ± 0.32	0.50 ± 0.32	1540 ± 855	1.42 ± 0.76
		HITON-MB	0.56 ± 0.28	0.78 ± 0.33	0.50 ± 0.31	2339 ± 1887	1.97 ± 1.49
	Alarm10	PCMB	0.52 ± 0.29	**0.80 ± 0.33**	0.45 ± 0.32	6053 ± 6052	5.44 ± 5.06
		IPCMB	0.42 ± 0.28	0.45 ± 0.29	**0.53 ± 0.33**	34,323 ± 158,384	36.84 ± 171.12
		STMB	0.36 ± 0.21	0.34 ± 0.20	0.52 ± 0.33	3269 ± 1545	3.24 ± 1.33
		EEMB	**0.57 ± 0.29**	0.74 ± 0.32	0.51 ± 0.32	1332 ± 698	1.71 ± 0.88
		IAMB	0.87 ± 0.14	**0.93 ± 0.15**	0.85 ± 0.18	**888 ± 236**	**3.42 ± 0.92**
		MMMB	**0.95 ± 0.07**	0.91 ± 0.12	0.99 ± 0.03	2125 ± 903	8.26 ± 3.52
		HITON-MB	**0.95 ± 0.07**	0.92 ± 0.12	0.99 ± 0.03	4938 ± 2406	21.30 ± 10.50
	Child10	PCMB	**0.95 ± 0.07**	0.91 ± 0.12	0.99 ± 0.03	12,439 ± 7319	55.26 ± 32.71
		IPCMB	0.90 ± 0.10	0.83 ± 0.17	**1.00 ± 0.00**	9288 ± 5792	40.55 ± 25.38
		STMB	0.52 ± 0.23	0.40 ± 0.27	0.98 ± 0.08	4202 ± 4499	18.90 ± 20.67
		EEMB	**0.95 ± 0.07**	0.95 ± 0.10	0.96 ± 0.08	1298 ± 908	5.64 ± 4.07
		IAMB	0.59 ± 0.19	**0.92 ± 0.21**	0.47 ± 0.23	**1164 ± 367**	**6.24 ± 2.12**
		MMMB	0.66 ± 0.23	0.78 ± 0.24	0.62 ± 0.28	3496 ± 1672	19.29 ± 9.45
		HITON-MB	**0.71 ± 0.24**	0.86 ± 0.23	**0.64 ± 0.28**	12,678 ± 8543	71.85 ± 49.08
5000	Insurance10	PCMB	0.58 ± 0.30	0.69 ± 0.33	0.56 ± 0.35	20,977 ± 13088	118.66 ± 74.57
		IPCMB	0.46 ± 0.24	0.44 ± 0.27	0.59 ± 0.34	20,267 ± 13,421	111.52 ± 73.72
		STMB	0.35 ± 0.18	0.30 ± 0.24	0.56 ± 0.33	5344 ± 6547	30.20 ± 38.15
		EEMB	0.70 ± 0.22	0.88 ± 0.22	0.62 ± 0.27	1742 ± 856	9.49 ± 4.77
		IAMB	0.66 ± 0.24	0.80 ± 0.27	0.64 ± 0.30	**1707 ± 548**	**12.23 ± 4.05**
		MMMB	0.77 ± 0.24	0.90 ± 0.21	0.72 ± 0.30	2126 ± 1064	15.06 ± 7.61
		HITON-MB	**0.78 ± 0.23**	0.92 ± 0.21	0.73 ± 0.29	4525 ± 3869	32.30 ± 27.98
	Alarm10	PCMB	0.76 ± 0.23	**0.95 ± 0.18**	0.68 ± 0.27	8,936 ± 6039	63.42 ± 43.11
		IPCMB	0.68 ± 0.20	0.72 ± 0.27	**0.76 ± 0.28**	14,184 ± 10,102	101.78 ± 72.36
		STMB	0.45 ± 0.19	0.40 ± 0.27	**0.76 ± 0.28**	4,207 ± 2721	30.47 ± 19.59
		EEMB	0.76 ± 0.24	0.86 ± 0.25	0.73 ± 0.29	1,651 ± 744	11.69 ± 5.33

Table 7
Average Results on Large-Sized BNs Using Different Data Sizes.

Size	Algorithm	F1	Precision	Recall	CITs	Time	Rank-F1
	IAMB	0.47	**0.77**	0.38	**849**	**0.57**	2.67
	MMMB	0.54	0.68	0.52	1779	1.18	5.50
	HITON-MB	0.54	0.69	0.52	4000	2.01	5.83
500	PCMB	0.52	0.70	0.48	9334	5.57	4.00
	IPCMB	0.43	0.44	0.56	40,643	34.32	2.33
	STMB	0.35	0.30	**0.58**	4297	3.22	1.00
	EEMB	**0.55**	0.64	0.54	1425	1.26	**6.67**
	IAMB	0.71	0.88	0.65	**1253**	**7.30**	2.67
	MMMB	0.79	0.86	0.78	2582	14.20	5.50
	HITON-MB	**0.81**	**0.90**	**0.79**	7380	41.82	**6.50**
5000	PCMB	0.76	0.85	0.74	14,117	79.11	4.33
	IPCMB	0.68	0.66	0.78	14,580	84.62	2.67
	STMB	0.44	0.37	0.77	4584	26.52	1.00
	EEMB	0.80	**0.90**	0.77	1564	8.94	5.33

Table 8
Summary of the Benchmark Real-World Datasets.

Dataset	Features	Instances
Heart	13	270
Spect	22	267
Unblanced	32	856
Sepctf	44	267
Sonar	60	208
Bankruptcy	147	7063
ovariancancer	2190	216
Breastcancer	12,533	181
Madelon	500	2000
Dexter	20,000	300

Table 9
Accuracy, Number of Selected Features, Number of CITS, and Runtime (in Seconds) on Small-Sized Real-World Datasets.

Dataset	Algorithm	KNN	SVM	Compactness	CITs	Time
	IAMB	0.68 ± 0.13	0.76 ± 0.07	**3 ± 1**	**40 ± 2**	**0.01 ± 0.00**
	MMMB	**0.79 ± 0.10**	0.81 ± 0.09	7 ± 1	444 ± 153	0.08 ± 0.02
	HITON-MB	**0.79 ± 0.11**	0.81 ± 0.09	7 ± 0	724 ± 226	0.12 ± 0.02
heart	PCMB	**0.79 ± 0.10**	0.81 ± 0.09	7 ± 1	2733 ± 1087	0.44 ± 0.13
	IPCMB	0.77 ± 0.09	**0.84** ± 0.07	7 ± 1	684 ± 233	0.11 ± 0.01
	STMB	0.78 ± 0.06	0.83 ± 0.06	5 ± 1	323 ± 259	0.05 ± 0.02
	EEMB	**0.79 ± 0.09**	0.81 ± 0.08	5 ± 1	226 ± 86	0.04 ± 0.01
	IAMB	0.77 ± 0.06	**0.79 ± 0.02**	3 ± 0	**85 ± 16**	**0.03 ± 0.00**
	MMMB	0.75 ± 0.06	**0.79 ± 0.02**	3 ± 1	325 ± 53	0.11 ± 0.02
	HITON-MB	0.77 ± 0.06	**0.79 ± 0.02**	3 ± 0	667 ± 188	0.23 ± 0.06
spect	PCMB	0.60 ± 0.33	0.63 ± 0.33	**1 ± 1**	779 ± 374	0.27 ± 0.12
	IPCMB	0.59 ± 0.32	0.64 ± 0.34	2 ± 2	707 ± 224	0.27 ± 0.08
	STMB	0.61 ± 0.33	0.64 ± 0.34	2 ± 2	337 ± 120	0.14 ± 0.07
	EEMB	**0.78 ± 0.06**	**0.79 ± 0.02**	3 ± 0	133 ± 23	0.05 ± 0.01
	IAMB	**0.81 ± 0.07**	**0.99 ± 0.00**	2 ± 0	92 ± 10	0.04 ± 0.01
	MMMB	**0.81 ± 0.07**	**0.99 ± 0.00**	2 ± 0	48,788 ± 7023	8.79 ± 0.97
	HITON-MB	**0.81 ± 0.07**	**0.99 ± 0.00**	2 ± 0	170,130 ± 29,508	26.94 ± 3.87
unblanced	PCMB	–	–	FAIL	–	–
	IPCMB	–	–	FAIL	–	–
	STMB	0.57 ± 0.40	0.69 ± 0.48	**1 ± 1**	69 ± 4	**0.03 ± 0.01**
	EEMB	**0.81 ± 0.07**	**0.99 ± 0.00**	2 ± 0	**68 ± 12**	**0.03 ± 0.00**
	IAMB	0.50 ± 0.00	0.74 ± 0.16	**1 ± 0**	**88 ± 0**	**0.03 ± 0.01**
	MMMB	**0.70 ± 0.20**	0.74 ± 0.15	29 ± 3	8214 ± 3686	0.87 ± 0.30
	HITON-MB	**0.70 ± 0.20**	0.74 ± 0.15	29 ± 3	10,879 ± 5482	0.99 ± 0.38
spectf	PCMB	**0.70 ± 0.20**	0.74 ± 0.15	29 ± 3	106,510 ± 56,074	7.44 ± 3.42
	IPCMB	**0.70 ± 0.20**	0.74 ± 0.15	29 ± 3	5,105,274 ± 833,386	203.92 ± 56.05
	STMB	**0.70 ± 0.20**	**0.81 ± 0.14**	9 ± 2	584,999 ± 45	30.82 ± 0.98
	EEMB	**0.70 ± 0.20**	**0.81 ± 0.14**	9 ± 2	2557 ± 1631	0.41 ± 0.28
	IAMB	0.49 ± 0.06	0.72 ± 0.08	**1 ± 0**	**120 ± 0**	**0.02 ± 0.00**
	MMMB	0.83 ± 0.11	**0.86 ± 0.07**	59 ± 1	2,911,123 ± 506,561	134.53 ± 19.28
	HITON-MB	0.83 ± 0.11	**0.86 ± 0.07**	59 ± 1	6,749,894 ± 1,284,965	438.56 ± 103.26
sonar	PCMB	0.83 ± 0.11	**0.86 ± 0.07**	59 ± 1	201,897,470 ± 39,974,255	9,262.85 ± 1,846.32
	IPCMB	0.83 ± 0.11	**0.86 ± 0.07**	59 ± 1	42,370,453 ± 2,600,838	2,237.13 ± 286.48
	STMB	0.83 ± 0.07	0.82 ± 0.07	20 ± 1	1,852,266 ± 649,304	118.97 ± 41.72
	EEMB	**0.84 ± 0.06**	0.83 ± 0.06	20 ± 2	51,622 ± 14,671	7.19 ± 2.31

HITON-MB, and EEMB are 1.80, 2.50, 2.50 and 3.20, respectively. Since PCMB, IPCMB, and STMB fail on some datasets, we do not use the Friedman test to compare EEMB with PCMB, IPCMB, and STMB.

With respect to the number of selected features, according to Tables 11 and 12, EEMB is very competitive with its rivals. Thus, EEMB does not choose an excessive number of features. STMB chooses too many features in *ovariancancer, hiva*, and *madelon* but still has poor accuracy.

The CIT and running time results in Table 11 show that the speed of EEMB is comparable with that of IAMB and is much faster than that of the other algorithms on each dataset. On *unblanced*, EEMB is faster than IAMB. According to the average CITs shown in Table 12, MMMB is 4.6 times slower than EEMB, and HitonMB is 19.6 times slower than EEMB.

Summary. For real-world datasets. Since no PC sets overlap during the symmetry check, PCMB and IPCMB fail on *unblanced*. Furthermore, due to the high computational complexity, PCMB, IPCMB, and STMB fail on datasets with more than 500 features (since the running time exceeds three days).

Table 10
Accuracy, Number of Selected Features, Number of CITS, and Runtime (in Seconds) on Large-Sized Real-World Datasets.

Dataset	Algorithm	KNN	SVM	Compactness	CITs	Time
	IAMB	0.88 ± 0.02	**0.90 ± 0.01**	**9 ± 0**	**1,373 ± 0**	**6.54 ± 0.10**
	MMMB	0.88 ± 0.01	**0.90 ± 0.00**	62 ± 3	1,145,668 ± 260,384	4,335.81 ± 926.50
	HITON-MB	0.88 ± 0.01	**0.90 ± 0.00**	58 ± 2	12,865,161 ± 3,272,809	17,352.28 ± 4,321.48
bankruptcy	PCMB	–	–	FAIL	–	–
	IPCMB	–	–	FAIL	–	–
	STMB	0.88 ± 0.01	0.89 ± 0.00	80 ± 4	1,354,802 ± 158,508	8,008.75 ± 1,115.70
	EEMB	**0.89 ± 0.01**	**0.90 ± 0.01**	26 ± 3	506,054 ± 50,403	2,196.56 ± 246.20
	IAMB	0.82 ± 0.08	0.88 ± 0.06	**3 ± 0**	**9,074 ± 669**	**22.96 ± 2.75**
	MMMB	0.86 ± 0.08	0.91 ± 0.04	10 ± 2	47,792 ± 6285	108.20 ± 20.06
	HITON-MB	0.88 ± 0.04	0.91 ± 0.05	7 ± 1	522,089 ± 288,407	1,333.17 ± 676.11
ovariancancer	PCMB	–	–	FAIL	–	–
	IPCMB	–	–	FAIL	–	–
	STMB	0.79 ± 0.11	0.81 ± 0.09	377 ± 103	940,233 ± 383,711	2,942.46 ± 1,202.45
	EEMB	**0.89 ± 0.05**	**0.94 ± 0.04**	20 ± 3	26,157 ± 2,817	63.11 ± 7.74
	IAMB	0.81 ± 0.09	0.82 ± 0.08	**4 ± 0**	**79,710 ± 4935**	**1,515.98 ± 206.95**
	MMMB	0.83 ± 0.06	0.85 ± 0.04	14 ± 6	367,847 ± 174,313	6,938.67 ± 3,105.52
	HITON-MB	0.85+0.09	0.84 ± 0.05	11 ± 2	1,980,752 ± 9,709,183	21,472.25 ± 6,302.57
breastcancer	PCMB	–	–	FAIL	–	–
	IPCMB	–	–	FAIL	–	–
	STMB	–	–	FAIL	–	–
	EEMB	**0.86 ± 0.05**	**0.86 ± 0.06**	30 ± 8	321,842 ± 48,599	5,683.22 ± 1,040.81
	IAMB	0.58 ± 0.04	**0.63 ± 0.03**	6 ± 0	**3,020 ± 0**	**10.99 ± 1.08**
	MMMB	0.56 ± 0.04	0.60 ± 0.02	6 ± 1	3717 ± 581	15.78 ± 2.55
	HITON-MB	0.58 ± 0.03	0.61 ± 0.03	6 ± 1	6173 ± 2865	28.98 ± 14.20
madelon	PCMB	0.50 ± 0.04	0.56 ± 0.04	**2 ± 1**	9067 ± 4561	38.88 ± 19.25
	IPCMB	0.55 ± 0.02	0.61 ± 0.03	7 ± 2	28,191 ± 5390	122.71 ± 24.14
	STMB	0.55 ± 0.03	0.61 ± 0.04	26 ± 6	8050 ± 838	34.84 ± 3.70
	EEMB	**0.62 ± 0.06**	**0.63 ± 0.04**	8 ± 1	3935 ± 488	16.85 ± 2.12
	IAMB	0.73 ± 0.09	0.81 ± 0.07	**4 ± 0**	**46,881 ± 171**	**1,675.52 ± 59.32**
	MMMB	0.81 ± 0.09	0.85 ± 0.09	11 ± 4	790,907 ± 840,530	16,813.61 ± 18,185.93
	HITON-MB	0.82 ± 0.09	0.85 ± 0.07	12 ± 1	358,990 ± 70,608	7,600.82 ± 1,487.94
dexter	PCMB	–	–	FAIL	–	–
	IPCMB	-	-	FAIL	-	-
	STMB	-	-	FAIL	-	-
	EEMB	**0.84 ± 0.09**	**0.89 ± 0.06**	21 ± 3	246,040 ± 21,707	5,708.22 ± 475.29

Table 11
Comparison Results of EEMB on 10 Real-World Datasets.

Algorithm	KNN	SVM	Compactness	CITs	Time
IAMB	9/1/0	6/4/0	0/2/8	1/0/9	1/0/9
MMMB	7/3/0	5/4/1	4/2/4	9/0/1	9/0/1
HITON-MB	7/3/0	5/4/1	4/2/4	10/0/0	10/0/0
PCMB	8/2/0	8/1/1	8/0/2	10/0/0	10/0/0
IPCMB	9/1/0	8/0/2	8/0/2	10/0/0	10/0/0
STMB	9/1/0	8/1/1	5/3/2	10/0/0	8/2/0

Table 12
Average Results on 10 Real-World Datasets.

Algorithm	KNN	SVM	Compactness	CITs	Time	Rank-KNN	Rank-SVM
IAMB	0.71	0.80	**4**	**14,048**	**323.21**	1.55	1.80
MMMB	0.78	0.83	20	532,483	2835.65	2.10	2.50
HITON-MB	0.79	0.83	19	2,266,546	4825.43	2.70	2.50
PCMB	–	–	–	–	–	–	–
IPCMB	–	–	–	–	–	–	–
STMB	–	–	–	–	–	–	–
EEMB	**0.80**	**0.85**	14	115,863	1367.57	**3.65**	**3.20**

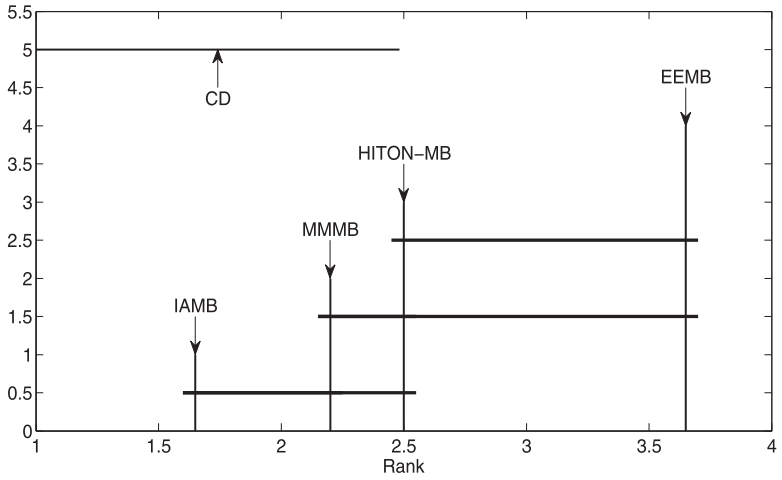

Fig. 5. Crucial difference diagram of the Nemenyi test for KNN on 10 real-world datasets.

On real-world datasets, IAMB is the fastest but the least accurate. EEMB is not only the most accurate algorithm but also the second-fastest.

6. Conclusion

In this paper, we proposed the EEMB algorithm, which is able to discover MBs efficiently and effectively with high-dimensional data with small-sized data samples. The extensive experiments show that EEMB is not only as fast as IAMB but also as accurate as HITON-MB for both MB discovery and feature selection. The advantages of EEMB are more obvious with small-sized sample data. Almost all state-of-the-art MB discovery algorithms depend on conditional independence tests, which leads to ineffectiveness on large-sized MBs. Thus, future research directions may include (1) developing new algorithms to address large-sized MB discovery and (2) proposing new methods for MB discovery rather than conditional independence testing.

Declaration of Competing Interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled, “Towards Efficient and Effective Discovery of Markov Blankets for Feature Selection”.

Acknowledgments

This work is partly supported by the National Key Research and Development Program of China (under grant 2016YFB1000901), the National Science Foundation of China (under grants 61876206 and 61673152), and the Anhui Province Key Research and Development Plan (No. 201904a05020073).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ins.2019.09.010.

References

[1] C.F. Aliferis, A. 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, J. Mach. Learn. Res. 11 (Jan) (2010) 171–234.
[2] C.F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, X.D. Koutsoukos, Local causal and Markov blanket induction for causal discovery and feature selection for classification part ii: analysis and extensions, J. Mach. Learn. Res. 11 (Jan) (2010) 235–284.
[3] C.F. Aliferis, I. Tsamardinos, A. Statnikov, Hiton: a novel Markov blanket algorithm for optimal variable selection, in: Proceedings of the AMIA Annual Symposium Proceedings, 2003, American Medical Informatics Association, 2003, p. 21.
[4] I.A. Beinlich, H.J. Suermondt, R.M. Chavez, G.F. Cooper, The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks, in: Proceedings of the AIME, Springer, 1989, pp. 247–256.
[5] J. Binder, D. Koller, S. Russell, K. Kanazawa, Adaptive probabilistic networks with hidden variables, Mach. Learn. 29 (2–3) (1997) 213–244.
[6] V. Bolón-Canedo, D. Rego-Fernández, D. Peteiro-Barral, A. Alonso-Betanzos, B. Guijarro-Berdiñas, N. Sánchez-Maroño, On the scalability of feature selection methods on high-dimensional data, Knowl. Inf. Syst. (2018) 1–48.
[7] A. Dawid, R. Cowell, S. Lauritzen, D. Spiegelhalter, Probabilistic Networks and Expert Systems, Springer-Verlag, 1999.

[8] S.R. De Morais, A. Aussem, A novel scalable and data efficient feature subset selection algorithm, in: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2008, pp. 298–312.
[9] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (Jan) (2006) 1–30.
[10] D. Dua, K.T. Efi, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2017. http://archive.ics.uci.edu/ml.
[11] S. Fu, M.C. Desmarais, Fast Markov blanket discovery algorithm via local learning within single pass, in: Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence, Springer, 2008, pp. 96–107.
[12] T. Gao, Q. Ji, Efficient Markov blanket discovery and its application, IEEE Trans. Cybern. 47 (5) (2017) 1169–1179.
[13] B. Hitt, P. Levine, Multiple high-resolution serum proteomic features for ovarian cancer detection, 2006, US Patent App. 11/093,018.
[14] D. Koller, M. Sahami, Toward optimal feature selection, Technical Report, Stanford InfoLab, 1996.
[15] D. Margaritis, S. Thrun, Bayesian network induction via local neighborhoods, in: Proceedings of the Advances in Neural Information Processing Systems, 2000, pp. 505–511.
[16] J. Pearl, Morgan Kaufmann series in representation and reasoning, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, CA, US, 1988.
[17] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Elsevier, 2014.
[18] J.M. Pena, R. Nilsson, J. Björkegren, J. Tegnér, Towards scalable and data efficient learning of Markov boundaries, Int. J. Approx. Reason. 45 (2) (2007) 211–232.
[19] P. Spirtes, C.N. Glymour, R. Scheines, Causation, Prediction, and Search, MIT press, 2000.
[20] A. Statnikov, I. Tsamardinos, C. Aliferis, An algorithm for generation of large Bayesian networks, Technical Report DSL-03-01, Department of Biomedical Informatics, Discovery Systems Laboratory, Vanderbilt University, 2003.
[21] 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, ACM, 2003, pp. 673–678.
[22] I. Tsamardinos, C.F. Aliferis, A.R. Statnikov, E. Statnikov, Algorithms for large scale Markov blanket discovery., in: Proceedings of the FLAIRS Conference, 2, 2003, pp. 376–380.
[23] I. Tsamardinos, L.E. Brown, C.F. Aliferis, The max-min hill-climbing Bayesian network structure learning algorithm, Mach. Learn. 65 (1) (2006) 31–78.
[24] Y. Wang, J.G. Klijn, Y. Zhang, A.M. Sieuwerts, M.P. Look, F. Yang, D. Talantov, M. Timmermans, M.E. Meijer-van Gelder, J. Yu, et al., Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer, The Lancet 365 (9460) (2005) 671–679.
[25] X. Wu, K. Yu, W. Ding, H. Wang, X. Zhu, Online feature selection with streaming features, IEEE Trans. Pattern Anal. Mach. Intell. 35 (5) (2013) 1178–1192.
[26] X. Xue, M. Yao, Z. Wu, A novel ensemble-based wrapper method for feature selection using extreme learning machine and genetic algorithm, Knowl. Inf. Syst. 57 (2) (2018) 389–412.
[27] S. Yaramakala, D. Margaritis, Speculative markov blanket discovery for optimal feature selection, in: Proceedings of the Fifth IEEE International Conference on Data Mining, IEEE, 2005, p. 4.
[28] K. Yu, L. Liu, J. Li, A unified view of causal and non-causal feature selection, (2018). arXiv preprint arXiv:1802.05844.
[29] K. Yu, L. Liu, J. Li, W. Ding, T. Le, Multi-source causal feature selection, IEEE Trans. Pattern Anal. Mach. Intell. (2019), doi:10.1109/TPAMI.2019.2908373.