BAMB: A Balanced Markov Blanket Discovery Approach to Feature Selection

ZHAOLONG LING, KUI YU, and HAO WANG*, Hefei University of Technology, China
LIN LIU, University of South Australia, Australia
WEI DING, University of Massachusetts Boston, USA
XINDONG WU, Mininglamp Technology, China

The discovery of Markov blankets (MB) for feature selection has attracted much attention in recent years since the MB of the class attribute is the optimal feature subset for feature selection. However, almost all existing MB discovery algorithms focus on either improving computational efficiency or boosting learning accuracy, instead of both. In this paper, we propose a novel MB discovery algorithm for balancing efficiency and accuracy, called BAMB (BAlanced Markov Blanket discovery). To achieve this goal, given a class attribute of interest, BAMB finds candidate PC (parents and children) and spouses and removes false positives from the candidate MB set in one go. Specifically, once a feature is successfully added to the current PC set, BAMB finds the spouses with regard to this feature, then uses the updated PC and the spouse set to remove false positives from the current MB set. This makes the PC and spouses of the target as small as possible, and thus to achieve a trade-off between computational efficiency and learning accuracy. In the experiments, we first compare BAMB with 8 state-of-the-art MB discovery algorithms on 7 benchmark Bayesian networks, then we use 10 real-world datasets and compare BAMB with 12 feature selection algorithms, including 8 state-of-the-art MB discovery algorithms and 4 other well-established feature selection methods. On prediction accuracy, BAMB outperforms 12 feature selection algorithms compared. On computational efficiency, BAMB is close to the IAMB algorithm while is much faster than the remaining seven MB discovery algorithms.

CCS Concepts: • **Computing methodologies** → Feature selection.

Additional Key Words and Phrases: Markov blanket, Feature selection, Bayesian network, Classification

*This is the corresponding author.

Authors' addresses: Zhaolong Ling, z_dragonl@163.com; Kui Yu, yukui@hfut.edu.cn; Hao Wang, jsjxwangh@hfut.edu.cn, Key Laboratory of Knowledge Engineering with Big Data of Ministry of Education (Hefei University of Technology), and School of of Computer and Information, Hefei University of Technology, Hefei, Anhui, 230009, China; Lin Liu, lin.liu@unisa.edu.au, School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, SA, 5095, Australia; Wei Ding, ding@cs.umb.edu, Department of Computer Science, University of Massachusetts Boston, Boston, MA, 02125, USA; Xindong Wu, wuxindong@mininglamp.com, Key Laboratory of Knowledge Engineering with Big Data of Ministry of Education (Hefei University of Technology), and Mininglamp Academy of Sciences, Mininglamp Technology, Beijing, 100084, China.

2157-6904/2019/1-ART1 $15.00
https://doi.org/10.1145/3335676

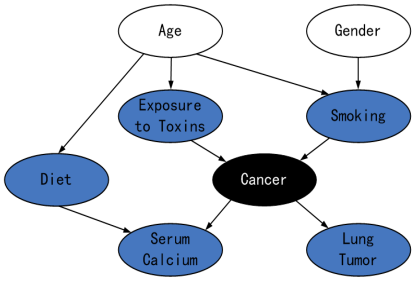


Fig. 1. The Markov blanket (in blue) of the node "Cancer" comprises "Exposure to Toxins" and "Smoking" (parents), "Serum Calcium" and "Lung Tumor" (children), and " Diet" (spouse)

ACM Reference Format:
Zhaolong Ling, Kui Yu, Hao Wang, Lin Liu, Wei Ding, and Xindong Wu. 2019. BAMB: A Balanced Markov Blanket Discovery Approach to Feature Selection. *ACM Trans. Intell. Syst. Technol.* 1, 1, Article 1 (January 2019), 25 pages. https://doi.org/10.1145/3335676

1 INTRODUCTION

Feature selection is commonly used to identify a subset of relevant features (aka variables or attributes) from a large number of features for building better prediction models [5, 6, 28, 31, 32]. For example, in bioinformatics, feature (gene) selection can identify a small number of informative genes for predicting diseases. In the era of big data, feature selection is more critical than ever, since high-dimensional data has become ubiquitous in various applications. For instance, in cancer genomics, a gene expression dataset may easily have more than 10,000 features.

In recent years, discovering Markov blanket (MB) for feature selection has attracted much attention [30]. Given a target attribute of interest, the MB of the target consists of the target's parents, children, and spouses (other parents of the children of the target) [18], as illustrated in Fig. 1 [29]. The MB of a target is a minimal set of features that renders all other features conditionally independent of the target [18]. Therefore, theoretically, the MB of the class attribute is an optimal set for feature selection [1].

Koller and Sahami [15] were the first to introduce the concept of MB to the field of feature selection. Based on their pioneer work, many MB discovery methods have been proposed, which can be divided into two categories: constraint-based and score-based methods.

Constraint-based methods use conditional independence tests for MB discovery. The representative algorithms include GS (Grow-Shrink) [16], IAMB (Iterative Associative MB), MMMB (Max-Min MB) [25], HITON-MB (HITON-MB) [2], PCMB (Parent-Children MB) [20], IPCMB (Iterative Parent-Child based search of MB) [11], and STMB (Simultaneous Markov Blanket) [12]. However, although GS, IAMB, and STMB are more efficient than the others, the number of samples required by them grows exponentially with the size of the MB of the target, since they use the entire set of features selected currently as the conditioning set in conditional independence tests. Therefore, when the sample size of a dataset is not big enough, these algorithms cannot find the MB accurately. MMMB, HITON-MB, PCMB, and IPCMB mitigate the problem of the large sample requirement by performing an exhaustive subset search within the features selected currently, but the

search is computationally expensive or even prohibitive when the size of the set of currently selected features becomes large.

Score-based algorithms employ Bayesian score functions to learn MBs, mainly including SLL (Score-based Local Learning) [17] and S^2TMB (Score-based Simultaneous MB) [13]. However, SLL and S^2TMB can be computationally expensive, because at each iteration they need to learn a Bayesian network structure involving all features selected currently, which is time consuming or infeasible when the Bayesian network is large.

In this paper, we aim to achieve both efficient and accurate MB discovery. The main contributions of the paper are summarized as follows.

(1) We propose a new constraint-based MB discovery algorithm, called BAMB (BAlanced Markov Blanket discovery). BAMB finds candidate PC (parents and children) and spouses and removes false positives from the candidate set in one go. In this way, the current PC and spouse set is kept to be as small as possible for the trade-off between computational efficiency and learning accuracy.
(2) We have conducted comprehensive experiments using seven benchmark Bayesian networks and ten real-world datasets, and have compared BAMB with twelve existing methods, including eight state-of-the-art MB discovery algorithms and four other well-established feature selection methods, to validate the efficiency and accuracy of the proposed BAMB.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces the BAMB algorithm. Section 4 presents and discusses the experimental results, and Section 5 concludes the paper.

2 RELATED WORK

Constraint-based MB discovery algorithms find MB using conditional independence tests. The first theoretically sound MB discovery algorithm, the GS (Grow-Shrink) algorithm [16] was proposed by Margaritis and Thrun. GS contains two sequential (and separated) phases, for growing and shrinking the candidate MB set respectively. In the growing phase, if a feature is dependent on the given target conditioning on the features selected currently, GS adds the feature to the candidate MB set, and the growing phase completes when all features are checked. In the shrinking phase, GS removes all false positives from the candidate MB set by testing the conditional independence between a candidate and the target, conditioning on all other features in the candidate set. The Incremental Association MB (IAMB) algorithm [26] is a modified version of GS. However, unlike GS, IAMB adds the feature having the highest association with the target into the candidate MB set at each iteration, thus achieves better discovery accuracy. Over the years, many variants of IAMB have been proposed, such as inter-IAMB, IAMBnPC, inter-IAMBnPC, and Fast-IAMB [29]. However, GS, IAMB, and IAMB's variants use the set of all currently selected features as the conditioning set, so the required number of data instances is exponential to the size of the MB to achieve reliable results. Moreover, they are not able to distinguish PC from spouses in a discovered MB.

To reduce the number of data samples required, Min-Max MB (MMMB) [25] applies a divide-and-conquer approach that breaks the problem of identifying MB into two subproblems, identifying PC and identifying spouses, and performs a subset search within the features currently selected for discovering the PC set of the target. HITON-MB [2] is a modified version of MMMB. It 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 [20], they provide a novel way for accurate MB discovery. Compared to MMMB and HITON-MB, two divide-and-conquer MB discovery approaches, the Parents and Children based MB algorithm (PCMB) [20] and Iterative Parent-Child based search of MB (IPCMB) algorithm [11], are proved to be correct under the faithfulness assumption.

The algorithms which employ the divide-and-conquer strategy have greatly improved the accuracy of MB discovery, especially with small-sized datasets. However, for spouse discovery, they need to apply the symmetry constraints to find spouses from the PC of each feature in the PC set of the target feature. To reduce the computational complexity, Gao and Ji [12] proposed the Simultaneous MB (STMB) algorithm. STMB also adopts the same strategy for PC discovery as IPCMB does. But for finding spouses, STMB avoids the expensive step of the symmetry checking by discovering spouses from all features excluding the current PC set. However, STMB employs the same strategy used by IAMB to remove false positives, which makes STMB inaccurate on small-sized datasets.

In addition to the constraint-based MB discovery algorithms, score-based algorithms discover MBs using score-based Bayesian network structure learning methods. The representative algorithms are the score-based local learning (SLL) [17] and the score-based simultaneous MB (S^2TMB) [13] algorithms. Based on the score metrics for Bayesian network structure learning, SLL finds the PC set of a given target first, then discovers spouses of the target. However, SLL also uses computationally expensive symmetry constraints checking to ensure the correctness of the approach. S^2TMB removes the symmetry constraints from both the PC and spouse search steps, and thus achieves better efficiency than SLL. However, SLL and S^2TMB need to learn a Bayesian network structure involving all features currently selected at each iteration, which is time consuming or infeasible when the Bayesian networks are large.

In summary, the existing MB discovery methods focus on improving either accuracy or efficiency. In this paper, we propose BAMB, a new algorithm for MB discovery, to achieve a trade-off between computational efficiency and learning accuracy.

Table 1. Summary of Notation

Symbol	Meaning
$\mathbf{U}$	a feature set
G	a directed acyclic graph over $\mathbf{U}$
P	a joint probability distribution over $\mathbf{U}$
X, Y	a feature
x, y	a discrete value that a feature may take
T	a given target feature in $\mathbf{U}$
$\mathbf{Z}, \mathbf{S}$	a conditioning set within $\mathbf{U}$
$X \perp\!\!\!\perp Y \mid \mathbf{Z}$	X is conditionally independent of Y given $\mathbf{Z}$
$X \not\perp\!\!\!\perp Y \mid \mathbf{Z}$	X is conditionally dependent on Y given $\mathbf{Z}$
$\mathbf{U} \setminus \{X\}$	all features in $\mathbf{U}$ excluding X
$\mathbf{MB}_T$	Markov blanket of T
$\mathbf{PC}_T$	parents and children of T
$\mathbf{CPC}_T$	a candidate set of $\mathbf{PC}_T$
$\mathbf{SP}_T$	spouses of T
$\mathbf{SP}_T\{X\}$	a subset of spouses of T with regard to T's child X
$\mathbf{CSP}_T\{X\}$	a candidate set of $\mathbf{SP}_T\{X\}$
$\mathbf{Tmp}$	a temporary set
$\mathbf{Sep}_T\{X\}$	a set that d-separates X from T
$dep(.)$	a measure of the strength of the dependence
$\|.\|$	the size of a set

3 NOTATIONS AND DEFINITIONS

Table 1 provides a summary of the notation used in this paper. In the following, we will introduce the key concepts, including Bayesian network, Markov blanket, and the relevant definitions and theorems.

Definition 1 (Conditional Independence). Two variables X and Y are conditionally independent given $\mathbf{Z}$, iff $P(X = x, Y = y|\mathbf{Z} = z) = P(X = x|\mathbf{Z} = z)P(Y = y|\mathbf{Z} = z)$.

Definition 2 (Bayesian Network) [18]. Let P be a discrete joint probability distribution of a set of random nodes (features) $\mathbf{U}$ via a directed acyclic graph G. We call the triplet $< \mathbf{U}, G, P >$ a Bayesian network if $< \mathbf{U}, G, P >$ satisfies the *Markov Condition*: every node in $\mathbf{U}$ is conditionally independent of its non-descendant nodes given its parents.

Definition 3 (Faithfulness) [23]. A Bayesian network is presented by a directed acyclic graph G and a joint probability distribution P over a feature set $\mathbf{U}$. G and P are faithful to each other iff all and only the conditional independencies between features in G are captured by P.

Definition 4 (V-Structure) [18]. The triplet of nodes X, Y, and Z form a V-structure with Z being the collider if node Z has two incoming edges from X and Y, respectively, but X and Y are non-adjacent, forming $X \rightarrow Z \leftarrow Y$.

Definition 5 (D-Separation) [18]. A path p between X and Y given $\mathbf{Z} \subseteq \mathbf{U} \setminus \{X \cup Y\}$ is open, iff (1) every collider on p is in $\mathbf{Z}$ or has a descendant in $\mathbf{Z}$, and (2) no other non-collider nodes on p are in $\mathbf{Z}$. Otherwise, the path p is blocked. Two nodes X and Y are d-separated given $\mathbf{Z}$, iff every path from X to Y is blocked by $\mathbf{Z}$.

Definition 6 (Markov Blanket) [18]. In a faithful Bayesian network, the $\mathbf{MB}_T$ is the set of parents, children, and spouses (parents of children) of node T. The $\mathbf{MB}_T$ of any node T is unique.

Theorem 1 [19, 23] Under the faithfulness assumption, in a Bayesian network, 1) a pair of nodes $X \in \mathbf{U}$ and $Y \in \mathbf{U}$ are adjacent, iff $X \not\perp\!\!\!\perp Y|\mathbf{Z}$, $\forall \mathbf{Z} \subseteq \mathbf{U} \setminus \{X, Y\}$; and 2) a triplet of nodes X, Y, and Z form a V-structure: $X \rightarrow Z \leftarrow Y$, iff $X \perp\!\!\!\perp Y|\mathbf{S}$ and $X \not\perp\!\!\!\perp Y|\{\mathbf{S} \cup Z\}$, $\exists \mathbf{S} \subseteq \mathbf{U} \setminus \{X, Y, Z\}$.

Theorem 2 [18] Given the Markov blanket of a target feature T, denoted as $\mathbf{MB}_T$, all other features are conditionally independent of T, that is, $T \perp\!\!\!\perp X|\mathbf{MB}_T$, for $\forall X \in \mathbf{U} \setminus \{T \cup \mathbf{MB}_T\}$.

4 THE PROPOSED BAMB ALGORITHM

In this section, we present the proposed BAMB algorithm. The main idea of the BAMB algorithm is shown in Fig. 2. Assuming the set, **Tmp** includes all features in $\mathbf{U}$ excluding T initially, in Fig. 2, in the beginning, BAMB firstly selects a feature A in **Tmp** with the highest association with T by conditioning on an empty set, and at the same time removes A from **Tmp**. Then if there exists a subset of $\mathbf{CPC}_T$ such that A and T are independent conditioning on this subset, BAMB directly considers the next feature in **Tmp**. Otherwise, BAMB adds A to $\mathbf{CPC}_T$ (the candidate set of PC of T). Due to the A's inclusion, it immediately triggers BAMB to remove false positives in $\mathbf{CPC}_T \setminus \{A\}$. After the removal step, BAMB discovers the candidate spouses of T with regard to A, called $\mathbf{CSP}_T\{A\}$, from the set $\mathbf{U} \setminus \{\mathbf{CPC}_T \cup \{T\}\}$. Finally, BAMB uses the union of $\mathbf{CSP}_T$ and $\mathbf{CPC}_T$ currently selected to update $\mathbf{CSP}_T\{A\}$ and $\mathbf{CPC}_T$, respectively. After all the above steps are completed, BAMB considers the next feature in **Tmp**. BAMB is terminated until the set **Tmp** is empty.

As described in Algorithm 1, the BAMB algorithm includes 3 steps. Step 1 (lines 5 to 23) gets a candidate set of PC and a candidate set of spouses. Step 2 (lines 24 to 34) removes

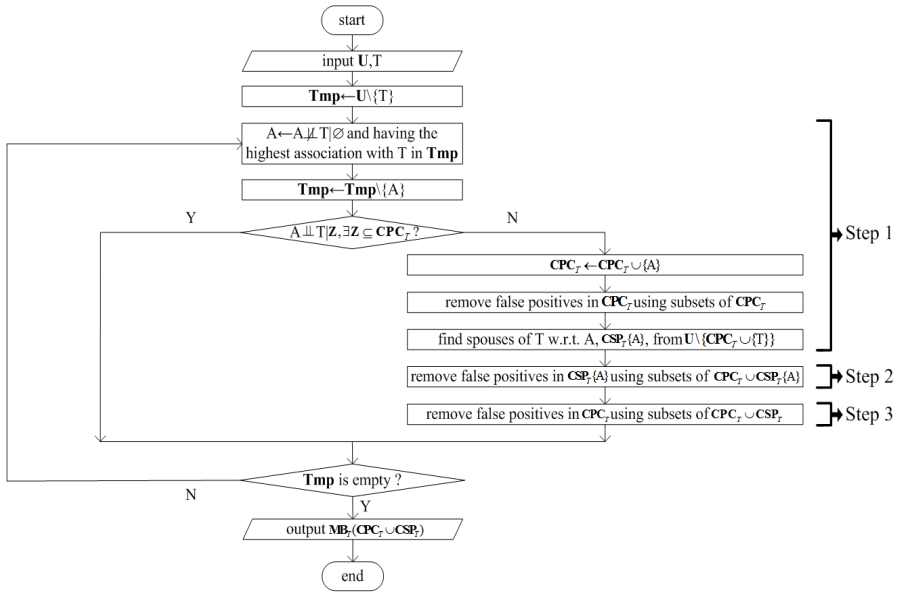


Fig. 2. The flow chart of BAMB

false positives from the candidate set of spouses, and Step 3 (lines 35 to 49) removes false positives from the candidate set of PC.

The rest of Section 4 is organized as follows. Section 4.1 presents a detailed description of BAMB. Section 4.2 gives a tracing example of BAMB. Sections 4.3 and 4.4 analyze the correctness and computational complexity of BAMB, respectively.

4.1 Algorithm Description

By the idea of BAMB shown in Fig. 2, Algorithm 1 gives the implementation detail of BAMB. In Algorithm 1, BAMB firstly calculates the association of each feature in $\mathbf{U}\backslash\{T\}$ with T given empty set and stores the separating set for the feature when it is independent of T. Then BAMB repeats Steps 1 to 3 (lines 4-50) as follows until **Tmp** is empty.

Step 1: Find the candidate set of PC and the candidate set of spouses of T. BAMB firstly selects from **Tmp** the feature, denoted as A with the highest association with T, and removes A from **Tmp**. If $\exists \mathbf{Z} \subseteq \mathbf{CPC}_T$ such that A and T are independent conditioning on $\mathbf{Z}$, BAMB directly considers the next feature in **Tmp** and does not implement the remaining steps (lines 9-49). Otherwise, BAMB adds A to $\mathbf{CPC}_T$. And since the new feature A is successfully added to $\mathbf{CPC}_T$, BAMB is triggered to check each feature in $\mathbf{CPC}_T \setminus \{A\}$ for removing false positives to keep the size of $\mathbf{CPC}_T$ as small as possible during the search procedure. In addition, in this step, BAMB removes the features independent of T from $\mathbf{CPC}_T \setminus \{A\}$ only conditioning on the subsets, including the newly added feature A to make the search process efficient.

Meanwhile, once the new feature A is added to $\mathbf{CPC}_T$, BAMB finds the candidate set of spouses of T with regard to A (lines 18-23). Instead of finding the spouses of T with regard to A from the parents and children of A, BAMB discovers the spouses of T in $\mathbf{U}\backslash\{T\}\backslash\mathbf{CPC}_T$. By Theorem 1, if a feature C in $\mathbf{U}\backslash\{T\}\backslash\mathbf{CPC}_T$ is dependent on T conditioning on the

ALGORITHM 1: The BAMB Algorithm

Input: D: dataset; T: target.
Output: [$\mathbf{PC}_T$, $\mathbf{SP}_T$]: Markov blanket of T.

1. $\mathbf{CPC}_T \leftarrow \emptyset$;
2. $\mathbf{Tmp} \leftarrow \mathbf{U} \setminus \{T\}$;
3. $\mathbf{Sep}_T\{X\} \leftarrow \emptyset$ for each $X \in \mathbf{U} \setminus \{T\}$ and $T \perp\!\!\!\perp X|\emptyset$;
4. **repeat**
   */*Step 1: Find the candidate set of PC and candidate set of spouses*/*
5. $A \leftarrow argmax_{X \in \mathbf{Tmp}} dep(T, X|\emptyset)$;
6. $\mathbf{Tmp} \leftarrow \mathbf{Tmp} \setminus \{A\}$;
7. **if** $T \perp\!\!\!\perp A|\mathbf{Z}$ for some $\mathbf{Z} \subseteq \mathbf{CPC}_T$ **then**
8. $\mathbf{Sep}_T\{A\} \leftarrow \mathbf{Z}$;
9. **else**
10. $\mathbf{CPC}_T \leftarrow \mathbf{CPC}_T \cup \{A\}$;
11. **for** each $B \in \mathbf{CPC}_T$ and $B \neq A$**do**
12. **if** $T \perp\!\!\!\perp B|\mathbf{Z}$ for some $\mathbf{Z} \subseteq \mathbf{CPC}_T \setminus \{B\}$ **then**
13. $\mathbf{CPC}_T \leftarrow \mathbf{CPC}_T \setminus \{B\}$;
14. $\mathbf{CSP}_T\{B\} \leftarrow \emptyset$;
15. $\mathbf{Sep}_T\{B\} \leftarrow \mathbf{Z}$;
16. **end if**
17. **end for**
18. $\mathbf{CSP}_T\{A\} \leftarrow \emptyset$;
19. **for** each C in $\{\mathbf{U} \setminus \{T\} \setminus \mathbf{CPC}_T\}$
20. **if** $T \not\perp\!\!\!\perp C|\mathbf{Sep}_T\{C\} \cup \{A\}$ **then**
21. $\mathbf{CSP}_T\{A\} \leftarrow \mathbf{CSP}_T\{A\} \cup \{C\}$;
22. **end if**
23. **end for**
    */*Step 2: Remove false positives from the candidate set of spouses*/*
24. $\mathbf{SP}_T\{A\} \leftarrow \emptyset$;
25. **repeat**
26. $E \leftarrow argmax_{X \in \mathbf{CSP}_T\{A\}} dep(T, X|\mathbf{Sep}_T\{X\} \cup \{A\})$;
27. $\mathbf{CSP}_T\{A\} \leftarrow \mathbf{CSP}_T\{A\} \setminus \{E\}$;
28. $\mathbf{SP}_T\{A\} \leftarrow \mathbf{SP}_T\{A\} \cup \{E\}$;
29. **for** each $X \in \mathbf{SP}_T\{A\}$ **do**
30. **if** $T \perp\!\!\!\perp X|\mathbf{Z} \cup A$ for some $\mathbf{Z} \subseteq \mathbf{CPC}_T \cup \mathbf{SP}_T\{A\} \setminus \{X\}$ **then**
31. $\mathbf{SP}_T\{A\} \leftarrow \mathbf{SP}_T\{A\} \setminus \{X\}$;
32. **end if**
33. **end for**
34. **until** $\mathbf{CSP}_T\{A\}$ is empty
    */*Step 3: Remove false positives from the candidate set of PC*/*
35. $\mathbf{PC}_T \leftarrow \emptyset$;
36. **repeat**
37. $F \leftarrow argmax_{X \in \mathbf{CPC}_T} dep(T, X|\emptyset)$;
38. $\mathbf{CPC}_T \leftarrow \mathbf{CPC}_T \setminus \{F\}$;
39. $\mathbf{PC}_T \leftarrow \mathbf{PC}_T \cup \{F\}$;
40. **for** each $X \in \mathbf{PC}_T$ **do**
41. **if** $T \perp\!\!\!\perp X|\mathbf{Z} \cup_{Y \in \mathbf{Z}} \mathbf{SP}_T\{Y\}$ for some $\mathbf{Z} \subseteq \mathbf{PC}_T \setminus \{X\}$ **then**
42. $\mathbf{PC}_T \leftarrow \mathbf{PC}_T \setminus \{X\}$;
43. $\mathbf{SP}_T\{X\} \leftarrow \emptyset$;
44. **end if**
45. **end for**
46. **until** $\mathbf{CPC}_T$ is empty
47. $\mathbf{CSP}_T\{A\} \leftarrow \mathbf{SP}_T\{A\}$
48. $\mathbf{CPC}_T \leftarrow \mathbf{PC}_T$
49. **end if**
50. **until Tmp** is empty

union of the separating set of C and A, C is considered as a spouse of T and is added to $\mathbf{CSP}_T\{A\}$ (the candidate set of the spouse set of T with respect to A).

However, as shown in Fig. 3, false positives such as X and Y may be added to $\mathbf{CSP}_T\{A\}$ and $\mathbf{CPC}_T$, respectively. Thus after Step 1, BAMB immediately implements Steps 2 and 3 to remove these false positives from $\mathbf{CPC}_T$ and $\mathbf{CSP}_T\{A\}$, respectively. This strategy not only removes false positives but also keeps the size of $\mathbf{CPC}_T$ and $\mathbf{CSP}_T\{A\}$ as small as possible during the search procedure, and thus improving the efficiency and accuracy.

Step 2: Remove false spouses from $\boldsymbol{CSP}_T\{A\}$. In this step, BAMB removes false spouses from the $\mathbf{CSP}_T\{A\}$ using $\mathbf{CPC}_T$ currently selected. At the beginning of Step 2, $\mathbf{SP}_T\{A\}$ is empty, BAMB selects a feature with the highest association with A in $\mathbf{CSP}_T\{A\}$ and adds it to $\mathbf{SP}_T\{A\}$ (Line 28). Meanwhile, the feature is removed from $\mathbf{CSP}_T\{A\}$. Since the spouses in $\mathbf{CSP}_T\{A\}$ are the parents of A, Line 28 adds the most likely spouses in $\mathbf{CSP}_T\{A\}$ to $\mathbf{SP}_T\{A\}$ for removing false spouses as soon as possible. Then, for each feature X in $\mathbf{SP}_T\{A\}$, if there exists a subset of the union $\{\mathbf{SP}_T\{A\} \setminus \{X\} \cup \mathbf{CPC}_T\}$ such that X is independent of T, X is removed from $\mathbf{SP}_T\{A\}$. The process is repeated until $\mathbf{CSP}_T\{A\}$ is empty.

Step 3: Remove false positives from the candidate set of PC of T. Instead of directly removing false positives from $\mathbf{CPC}_T$, BAMB uses the same strategy as in Step 2. BAMB assumes that the set $\mathbf{PC}_T$ is empty initially, and adds the feature with the highest association with T in $\mathbf{CPC}_T$ to $\mathbf{PC}_T$ at each iteration. Then, for each feature X in $\mathbf{PC}_T$, if there exists a subset of the union $\{\mathbf{CPC}_T \cup \mathbf{CSP}_T\}$ such that X and T are independent conditioning on this subset, BAMB removes X from $\mathbf{PC}_T$. Meanwhile, BAMB sets $\mathbf{SP}_T\{X\}$ (the spouses of T with regard to X) empty (Line 43). Step 3 is repeated until $\mathbf{CPC}_T$ is empty.

In addition, at Step 2, even if A has no spouses, step 3 will still be executed. For example, as shown in Fig. 3, assuming that currently $\mathbf{CPC}_T = \{M, X\}$, $\mathbf{SP}_T\{M\} = \{N\}$, and $\mathbf{SP}_T\{X\} = \emptyset$. Then P is added to $\mathbf{CPC}_T$ at Step 1, although $\mathbf{SP}_T\{P\}$ is empty. At Step 3, due to the inclusion of P, BAMB can use the conditioning set $\{M, N, P\}$ to remove the non-child descendant of T from $\mathbf{CPC}_T$.

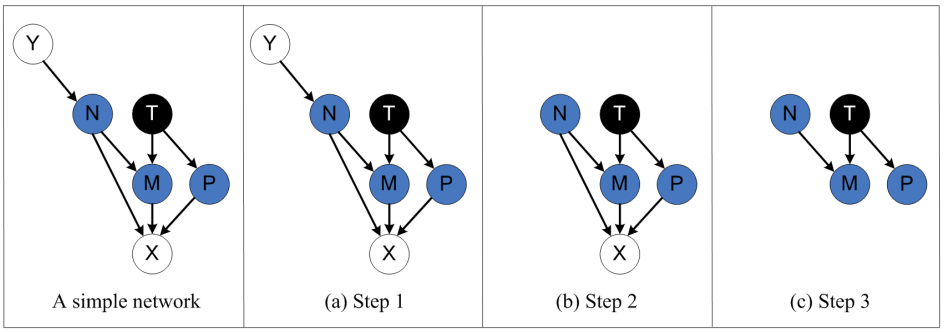


Fig. 3. An example of BAMB's execution process.

4.2 Tracing BAMB

In this section, we use the example in Fig. 3. to trace the execution of BAMB. In the following, T is the target feature and $\mathbf{MB}_T = \{M, N, P\}$.

(1) *Step 1:* Referring to the simple network, i.e., the first network in Fig. 3, since $Y \perp\!\!\!\perp T|\emptyset$ and $N \perp\!\!\!\perp T|\emptyset$, nodes Y and N are not added to $\mathbf{CPC}_T$. Notice that, the true PC

set, $\mathbf{PC}_T = \{M, P\}, X \notin \mathbf{PC}_T$, but no subsets within $\mathbf{PC}_T$ make X conditionally independent of T: $X \not\perp\!\!\!\perp T|\emptyset$, $X \not\perp\!\!\!\perp T|M$, $X \not\perp\!\!\!\perp T|P$, and $X \not\perp\!\!\!\perp T|\{M \cup P\}$. In addition, since $N \not\perp\!\!\!\perp T|M$, node N is added to $\mathbf{CSP}_T\{M\}$. Notice that, the true spouse set, $\mathbf{SP}_T\{M\} = \{N\}, Y \notin \mathbf{SP}_T\{M\}$, and Y is added to $\mathbf{CSP}_T\{M\}$ due to $Y \not\perp\!\!\!\perp T|M$. Thus, as shown in Fig. 3 (a), after Step 1, there are some non-child descendants remaining in the candidate set of PC of T: $\mathbf{CPC}_T = \{M, X, P\}$, and some spouses' parents remaining in the candidate set of spouses of T: $\mathbf{CSP}_T\{M\} = \{N, Y\}$, and we need Step 2 and Step 3 to remove these false positives.

(2) *Step 2:* As shown in Fig. 3 (b), for some $\mathbf{Z} \subseteq \mathbf{CPC}_T \cup \mathbf{SP}_T\{M\} \backslash \{Y\}$, the conditioning set $\{\mathbf{Z} \cup M\}$ makes node Y conditionally independent of T: $Y \perp\!\!\!\perp T|\{M \cup N\}$, thus Y is removed from $\mathbf{SP}_T\{M\}$, and BAMB only includes the true spouses of T after Step 2.

(3) *Step 3:* As shown in Fig. 3 (c), for some $\mathbf{Z} \subseteq \mathbf{PC}_T \backslash \{X\}$, the conditioning set $\{\mathbf{Z} \cup_{Y \in \mathbf{Z}} \mathbf{SP}_T\{Y\}\}$ can form the MB of T, and make node X conditionally independent of T: $X \perp\!\!\!\perp T|\{M \cup N \cup P\}$, thus X is removed from $\mathbf{CPC}_T$, and BAMB finds all and only the MB of T after step 3.

4.3 Correctness of BAMB

Theorem 3 (Correctness of BAMB) Under the faithfulness assumption, BAMB outputs all and only the MB of the given target attribute.

Proof: *1) Step 1 finds all the true positive MB nodes.* According to Theorem 1, BAMB adds the features which are conditionally dependent T into the candidate set of PC of T and removes features which are conditionally independent of T from the candidate set of PC of T. Since the true PC of T is dependent on T given any subsets in $\mathbf{U}$, the candidate set of PC of T contains all true PC of T. Meanwhile, BAMB finds spouses of T at the same time: if feature B is a collider that forms the V-structure: $C \rightarrow B \leftarrow T$, feature C is considered as a candidate spouse of T via B by Theorem 1. Due to exhaustive search, BAMB will not miss any spouse of T in the set of all features excluding the found PC set of T. Accordingly, the candidate set of PC of T contains all true PC of T and the candidate set of spouses of T contains all true spouses of T.

2) Step 2 and Step 3 only remove false positive MB nodes. After Step 1, the candidate PC set and spouses set found by BAMB have included all true PC and spouses of T. However, as shown in Fig. 3 (a), although the path of $X - M - T$ is blocked by node M, node X also has another path $X - N - M - T$ to reach T. Moreover, since BAMB finds spouses from non-PC set, node Y could also form the V-structure $Y \rightarrow M \leftarrow T$. Therefore, there are two types of false positives existing in the candidate sets: 1) non-child descendants of T in the candidate PC set of T; and 2) parents of spouses of T in the candidate spouse set of T.

BAMB uses Theorem 2 to remove these false positives. Since some nodes in the candidate PC set and some nodes in the candidate spouse set together from the MB of T, BAMB directly removes spouses' parent nodes from the candidate set of the subset of spouses of T. And true spouses will not be removed since the conditioning set always contains the common child of T and spouses. Then BAMB only includes the true spouses of T after Step 2. The candidate PC set contains all true PC of T, so the subset within the candidate set of PC of T and the corresponding subset of spouses of T together consists of the MB of T, then BAMB removes non-child descendant nodes from the candidate set of PC of T. And since the true PC is always dependent on T given any subsets in $\mathbf{U}$, only true PC of T can be reserved after Step 3. Hence, BAMB finds all and only the MB of T.

Table 2. Computational complexity of constraint-based MB discovery algorithms

Algorithms	Computational Complexity
IAMB	$O(\lvert\mathbf{U}\rvert^2)$
MMMB	$O(2^{\lvert\mathbf{PC}\rvert}\lvert\mathbf{U}\rvert\lvert\mathbf{PC}\rvert)$
HITON-MB	$O(2^{\lvert\mathbf{PC}\rvert}\lvert\mathbf{U}\rvert\lvert\mathbf{PC}\rvert)$
PCMB	$O(2^{\lvert\mathbf{PC}\rvert}\lvert\mathbf{U}\rvert\lvert\mathbf{PC}\rvert^2)$
IPCMB	$O(2^{\lvert\mathbf{U}\rvert}\lvert\mathbf{U}\rvert\lvert\mathbf{PC}\rvert)$
STMB	$O(2^{\lvert\mathbf{U}\rvert}\lvert\mathbf{U}\rvert)$
BAMB	$O(2^{\lvert\mathbf{PC}\rvert}\lvert\mathbf{U}\rvert)$

4.4 Computational Complexity

The computational complexity of the state-of-the-art constraint-based MB discovery algorithms depends on the number of conditional independence (CI) tests [1]. BAMB (Algorithm 1) firstly sorts the features based on their associations with T, then performs an exhaustive subset search in the currently selected PC set at each iteration. When we find PC at each iteration, we also remove the false PC at the same time. In theory, this "interleave" approach will keep only the true PC set in $\mathbf{CPC}_T$, so the computational complexity of BAMB is proportional to the size of the PC set. Therefore, the computational complexity of Step 1 of BAMB takes $O(|\mathbf{U}|2^{|\mathbf{PC}|})$ CI tests, Step 2 takes $O(|\mathbf{SP}_T\{X\}|2^{(|\mathbf{PC}|+|\mathbf{SP}_T\{X\}|)})$ CI tests, and Step 3 takes $O(|\mathbf{PC}|2^{|\mathbf{PC}|})$ CI tests. Overall, BAMB takes $O(|\mathbf{U}|2^{|\mathbf{PC}|} + |\mathbf{SP}_T\{X\}|2^{(|\mathbf{PC}|+|\mathbf{SP}_T\{X\}|)} + |\mathbf{PC}|2^{|\mathbf{PC}|}) = O(|\mathbf{U}|2^{|\mathbf{PC}|})$ CI tests.

Specifically, the PC discovery of BAMB in Step 1 finds separating set from all subsets of the PC set at any iteration, and the PC discovery of STMB finds separating set from all subsets of $\mathbf{U}$ at any iteration. Consequently, the computational complexity of Step 1 of BAMB takes $O(|\mathbf{U}|2^{|\mathbf{PC}|})$ CI tests, and STMB for PC discovery takes $O(|\mathbf{U}|2^{|\mathbf{U}|})$ CI tests. In the worst case, when all features are the PC of the target feature, that is, there is no separating set, the time complexity of PC discovery of BAMB is same with STMB. However, most of the BNs have a large number of features but a small-sized PC set of each feature, so that BAMB will perform fewer tests. Thus, BAMB is much faster than STMB in PC discovery because of $|\mathbf{PC}| \ll |\mathbf{U}|$.

In addition, since symmetry constraint will check for each feature in the PC set uses the same algorithm as that for finding the PC set, an algorithm enforcing symmetry constraint check would be $|\mathbf{PC}|$ times more costly. Thus, PCMB would cost $|\mathbf{PC}|$ times more than MMMB and HITON-MB, IPCMB would cost $|\mathbf{PC}|$ times more than STMB, MMMB, and HITON-MB would cost $|\mathbf{PC}|$ times more than BAMB. Since $|\mathbf{PC}|$ is always far less than $|\mathbf{U}|$, MMMB and HITON-MB are faster than IPCMB, and BAMB is faster than STMB.

We summarize the computational complexity of the state-of-the-art constraint-based MB discovery algorithms in Table 2. From the table, IAMB is the fastest among all algorithms, while BAMB is the second-fastest algorithm, closely following to IAMB.

5 EXPERIMENTS

In this section, we firstly evaluate the efficiency and accuracy of BAMB with five sparse benchmark BN datasets and two local dense benchmark BN datasets, and compare BAMB with eight existing MB discovery algorithms, including six state-of-the-art constraint-based MB discovery algorithms, IAMB [26], MMMB [25], HITON-MB [2], PCMB [20], IPCMB [11], and STMB [12], and two state-of-the-art score-based MB discovery algorithms, SLL [17] and S^{2}TMB [13].

Then we compare BAMB with 12 other algorithms on 10 real-world datasets, including 8 MB discovery algorithms, IAMB, MMMB, HITON-MB, PCMB, IPCMB, STMB, SLL, and S^{2}TMB, and four well-established feature selection algorithms, FCBF [33], mRMR [21], SPFS-LAR [34], and MRF [8].

The implementation details and parameter settings of all the algorithms are as follows:

(1) IAMB, MMMB, HITON-MB, PCMB, IPCMB, STMB, and BAMB are implemented in MATLAB, and the conditional independence tests are G^2 tests with the statistical significance level of 0.01.
(2) SLL and S^{2}TMB are implemented in C++.
(3) FCBF, mRMR, SPFS-LAR, and MRF are implemented in MATLAB and C language. The information threshold of FCBF is set to 0. As for mRMR, SPFS-LAR, and MRF, we choose the top N features where N is the size of the MB obtained by BAMB on each dataset.

All experiments are conducted on a 2.20 GHz Intel Core i5-5200U with 4GB RAM.

Table 3. Summary of sparse 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
Insurance10	270	556	5/8	1/11	2-5
Alarm10	370	570	4/7	1/9	2-4

5.1 Sparse Benchmark BN Datasets

We use five sparse benchmark BN datasets with a range of dimensionality in our experiments, and the five sparse benchmark BN datasets are described in Table 3[1]. For each benchmark BN network, we use two groups of data, one group including 10 datasets with 1,000 data instances to represent small-sized datasets samples, and the other group containing 10 datasets with 5,000 data instances to represent large-sized datasets samples. For benchmark BN networks, the MB of each feature can be read from those networks. Accordingly, in the experiments, we evaluate the algorithms using the following metrics.

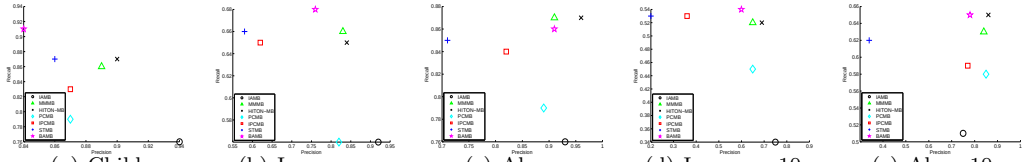
- Accuracy. $F1 = 2 * precision * recall/(precision + recall)$. The precision metric denotes the number of true positives in the output (i.e., the features in the output of an algorithm belonging to the true MB of a given target in a test DAG) divided by the number of features in the output of the algorithm, while the recall metric represents the number of true positives in the output divided by the number of true positives (the number of the true MB of a given target) in a test DAG. 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.
- Efficiency. We measure the efficiency of an algorithm using both the number of CI tests and runtime.

We report the results of BAMB and its rivals using three small-sized networks, Child [9], Insurance [4] and Alarm [3]. We run each algorithm to discover the MBs for all features in

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

Table 4. Comparison of BAMB with Constraint-Based MB Methods on sparse benchmark BN Datasets (size=1,000)

Dataset	Algorithm	F1	Precision	Recall	CI tests	Time
Child	IAMB	0.82±0.02	**0.94±0.03**	0.76±0.02	**54±1**	**0.02±0.00**
	MMMB	0.85±0.02	0.89±0.04	0.86±0.02	823±85	0.20±0.02
	HITON-MB	**0.87±0.02**	0.90±0.03	0.87±0.02	3.0e3±561	0.61±0.10
	PCMB	0.80±0.02	0.87±0.03	0.79±0.02	4.7e3±749	1.13±0.14
	IPCMB	0.83±0.02	0.87±0.03	0.83±0.03	1.2e3±57	0.39±0.02
	STMB	0.85±0.05	0.86±0.05	0.87±0.04	221±7	0.07±0.00
	BAMB	0.85±0.03	0.84±0.04	**0.91±0.01**	441±58	0.11±0.02
Insurance	IAMB	0.66±0.01	**0.92±0.03**	0.56±0.01	**86±2**	**0.02±0.00**
	MMMB	**0.71±0.02**	0.83±0.03	0.66±0.02	511±47	0.18±0.01
	HITON-MB	0.70±0.02	0.84±0.04	0.65±0.01	1.2e3±235	0.41±0.07
	PCMB	0.64±0.02	0.82±0.02	0.56±0.02	2.3e3±302	0.82±0.10
	IPCMB	0.60±0.03	0.62±0.05	0.65±0.04	2.2e4±3.3e4	5.37±8.15
	STMB	0.58±0.03	0.58±0.06	0.66±0.04	1.1e3±1.3e3	0.31±0.32
	BAMB	0.69±0.02	0.76±0.04	**0.68±0.01**	404±51	0.12±0.01
Alarm	IAMB	0.81±0.01	0.93±0.01	0.76±0.01	**120±2**	**0.04±0.00**
	MMMB	0.87±0.01	0.91±0.02	**0.87±0.01**	437±33	0.17±0.01
	HITON-MB	**0.90±0.01**	**0.96±0.02**	**0.87±0.01**	1.1e3±109	0.41±0.04
	PCMB	0.81±0.02	0.89±0.02	0.79±0.03	1.8e3±184	0.71±0.06
	IPCMB	0.81±0.03	0.82±0.02	0.84±0.04	1.1e3±46	0.42±0.02
	STMB	0.72±0.02	0.71±0.02	0.85±0.02	392±12	0.16±0.00
	BAMB	0.86±0.01	0.91±0.02	0.86±0.01	280±16	0.11±0.01
Insurance10	IAMB	0.44±0.22	**0.75±0.32**	0.34±0.21	**933±265**	**1.25±0.36**
	MMMB	0.53±0.28	0.65±0.30	0.52±0.34	2.1e3±1.2e3	2.76±1.50
	HITON-MB	**0.55±0.28**	0.69±0.30	0.52±0.34	5.3e3±3.7e3	6.19±4.27
	PCMB	0.50±0.28	0.65±0.33	0.45±0.31	9.8e3±6.7e3	9.44±6.38
	IPCMB	0.41±0.21	0.36±0.17	0.53±0.33	1.6e4±1.0e4	14.41±9.13
	STMB	0.27±0.15	0.20±0.11	0.53±0.32	6.4e3±4.6e3	5.47±3.70
	BAMB	0.53±0.25	0.60±0.28	**0.54±0.32**	1.9e3±1.5e3	1.64±1.26
Alarm10	IAMB	0.55±0.23	0.75±0.28	0.51±0.29	**1.4e3±351**	**1.17±0.52**
	MMMB	0.67±0.26	0.84±0.27	0.63±0.31	1.7e3±688	2.41±1.08
	HITON-MB	**0.70±0.26**	**0.86±0.27**	**0.65±0.31**	2.5e3±1.6e3	2.61±1.86
	PCMB	0.64±0.27	0.85±0.29	0.58±0.31	6.3e3±4.0e3	6.53±4.23
	IPCMB	0.61±0.27	0.77±0.29	0.59±0.32	8.6e3±6.1e3	10.49±7.16
	STMB	0.39±0.20	0.34±0.23	0.62±0.33	3.0e3±1.8e3	3.55±1.87
	BAMB	0.66±0.25	0.78±0.28	**0.65±0.32**	1.6e3±755	1.64±0.80
MEAN	IAMB	0.66	**0.86**	0.59	**515**	**0.50**
	MMMB	0.73	0.82	0.71	1.1e3	1.14
	HITON-MB	**0.74**	0.85	0.71	2.6e3	2.05
	PCMB	0.68	0.82	0.63	5.0e3	3.73
	IPCMB	0.65	0.69	0.69	9.6e3	6.22
	STMB	0.56	0.54	0.71	2.2e3	1.91
	BAMB	0.72	0.78	**0.73**	933	0.72

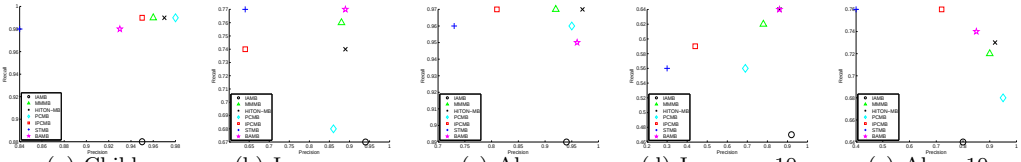


(a) Child (b) Insurance (c) Alarm (d) Insurance10 (e) Alarm10

Fig. 4. Precision-recall of BAMB and Constraint-Based MB Methods on sparse benchmark BN Datasets (Size=1,000).

Table 5. Comparison of BAMB with Constraint-Based MB Methods on sparse benchmark BN Datasets (size=5,000)

Dataset	Algorithm	F1	Precision	Recall	CI tests	Time
Child	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**	2.8e3±112	3.08±0.12
	PCMB	**0.98±0.01**	**0.98±0.01**	**0.99±0.01**	5.0e3±106	5.49±0.13
	IPCMB	0.96±0.02	0.95±0.03	**0.99±0.01**	1.9e3±155	1.94±0.17
	STMB	0.89±0.03	0.84±0.04	0.98±0.02	374±35	0.39±0.04
	BAMB	0.95±0.02	0.93±0.02	0.98±0.02	376±11	0.40±0.03
Insurance	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	1.2e3±124	1.63±0.18
	HITON-MB	0.78±0.02	0.89±0.03	0.74±0.02	3.2e3±414	4.47±0.62
	PCMB	0.74±0.01	0.86±0.02	0.68±0.02	7.2e3±1.1e3	10.00±1.68
	IPCMB	0.66±0.03	0.64±0.03	0.74±0.03	3.5e3±449	4.67±0.61
	STMB	0.65±0.02	0.64±0.04	**0.77±0.03**	703±47	0.96±0.07
	BAMB	0.80±0.01	0.89±0.03	**0.77±0.02**	619±39	0.92±0.06
Alarm	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**	1.5e3±38	2.15±0.06
	PCMB	0.95±0.02	0.95±0.01	0.96±0.02	2.9e3±215	3.97±0.32
	IPCMB	0.86±0.02	0.81±0.02	**0.97±0.01**	1.7e3±54	2.18±0.07
	STMB	0.78±0.02	0.73±0.02	0.96±0.01	531±15	0.73±0.03
	BAMB	0.94±0.02	0.96±0.03	0.95±0.01	351±11	0.51±0.03
Insurance10	IAMB	0.59±0.19	**0.92±0.21**	0.47±0.23	**1.2e3±367**	**5.52±2.38**
	MMMB	0.66±0.23	0.78±0.24	0.62±0.28	3.5e3±1.7e3	19.93±11.03
	HITON-MB	**0.71±0.24**	0.86±0.23	**0.64±0.28**	1.3e4±8.5e3	85.18±84.62
	PCMB	0.58±0.30	0.69±0.33	0.56±0.35	2.1e4±1.3e4	111.70±76.83
	IPCMB	0.46±0.24	0.44±0.27	0.59±0.34	2.0e4±1.3e4	115.74±77.76
	STMB	0.35±0.18	0.30±0.24	0.56±0.33	5.3e3±6.5e3	38.20±58.83
	BAMB	0.70±0.22	0.86±0.23	**0.64±0.27**	2.7e3±2.3e3	15.20±13.22
Alarm10	IAMB	0.66±0.24	0.80±0.27	0.64±0.30	**1.7e3±548**	**12.56±5.44**
	MMMB	0.77±0.24	0.90±0.21	0.72±0.30	2.1e3±1.1e3	13.24±7.27
	HITON-MB	**0.78±0.23**	0.92±0.21	0.73±0.29	4.5e3±3.9e3	35.62±41.13
	PCMB	0.76±0.23	**0.95±0.18**	0.68±0.27	8.9e3±6.0e3	59.01±48.08
	IPCMB	0.68±0.20	0.72±0.27	**0.76±0.28**	1.4e4±1.0e4	7.93±74.82
	STMB	0.45±0.19	0.40±0.27	**0.76±0.28**	4.2e3±2.7e3	28.44±20.86
	BAMB	0.76±0.24	0.85±0.24	0.74±0.29	1.9e3±1.1e3	12.87±8.81
MEAN	IAMB	0.76	**0.91**	0.71	**636**	**3.69**
	MMMB	0.83	0.89	0.81	1.7e3	7.32
	HITON-MB	**0.84**	0.92	0.81	4.9e3	26.10
	PCMB	0.80	0.89	0.77	9.0e3	38.03
	IPCMB	0.72	0.71	0.81	8.3e3	26.49
	STMB	0.62	0.58	0.81	2.2e3	13.74
	BAMB	0.83	0.90	**0.82**	1.2e3	5.98



(a) Child (b) Insurance (c) Alarm (d) Insurance10 (e) Alarm10

Fig. 5. Precision-recall of BAMB and Constraint-Based MB Methods on sparse benchmark BN Datasets (Size=5,000).

each BN. We also validate our proposed BAMB algorithm using two large-sized networks, Insurence10 and Alarm10. These two networks were generated by tiling 10 copies of the Insurance, and Alarm networks, respectively [24]. We randomly select 10% features in each BN and find their MBs.

For an algorithm, we report average results of F1, precision, recall, number of CI tests, and runtime over 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 CI tests, or runtime, and B is the standard deviation. "-" denotes that a method fails to generate any output with the corresponding dataset after running out of memory, and the best results are highlighted in bold face. Moreover, we show the precision-recall in Fig. 4 to Fig. 7 to make our experimental results clearer. The more the point of an algorithm at the upper right, the better the result.

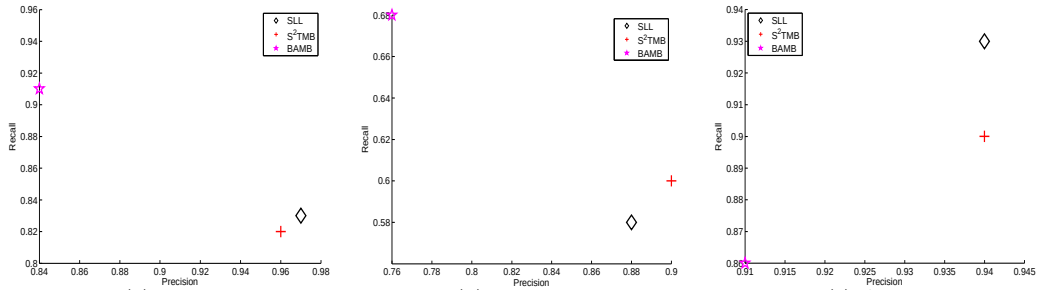
5.1.1 Comparison of BAMB with Constraint-Based MB Methods on sparse benchmark BN Datasets. According to Tables 4 and 5, on average, HITON-MB is the most accurate algorithm and IAMB is the fastest algorithm among all constraint-based algorithms under comparison. Meanwhile, BAMB illustrates comparable accuracy with HITON-MB in each BN with all cases, and BAMB is much faster (over 3 times) than HITON-MB on average in terms of the number of CI tests. Although IAMB is faster than BAMB, it is significantly inferior to BAMB on the F1 metric (over 6%) on average. Compared with other MB discovery algorithms, BAMB is more efficient and more accurate than STMB. On average, with small-sized data samples, BAMB is 36.8%, 64.9%, and 80.7% lower in time than MMMB, HITON-MB, and PCMB, respectively. And on average with large-sized data samples, BAMB is 18.3%, 77.1%, and 84.3% lower in time than MMMB, HITON-MB, and PCMB respectively. Specifically, BAMB is more efficient than MMMB, HITON-MB, and PCMB on each benchmark BN dataset with both small-sized and large-sized data samples. In addition, BAMB has the highest average Recall among all constraint-based algorithms under comparison, which means that BAMB can choose much more true features.

MMMB and HITON-MB have the same computational complexity in theory, as shown in Table 2. However, in practice, when actually finding PC, MMMB first adds all features that are dependent of T to the candidate PC set, then removes false positives from the candidate PC set, whereas after HITON-MB adds a feature dependent of T to the candidate PC, it immediately starts to remove false positives from the candidate PC set. Although this strategy does not change the result of the complexity analysis of HITON-MB and it makes HITON-MB remove false positives as soon as possible to improve accuracy, results in lower efficiency for HITON-MB in practices for the following reason: When a newly arrived feature can be removed by a subset of the candidate PC set, HITON-MB still checks each feature in the candidate PC set whether it is a false parent of child. It is unnecessary due to conditioned on all subsets of all other variables in the current set (including the newly arrived feature and all features already in the candidate set). However, for features already in the candidate set, it is unnecessary to repeat the conditional independence test for any of them conditioned on any subset of features already in the candidate PC set.

5.1.2 Comparison of BAMB with Score-Based MB Methods on sparse benchmark BNs. We also validate our proposed algorithm by comparing BAMB with score-based algorithms, and Tables 6 and 7 summarize the results. Score-based algorithms do not use conditional independence tests, so we do not report the comparison of BAMB with score-based MB algorithms on CI tests. On small-size networks, score-based algorithms achieve better accuracy than BAMB, but BAMB is much faster than them. However, on large-sized networks, since

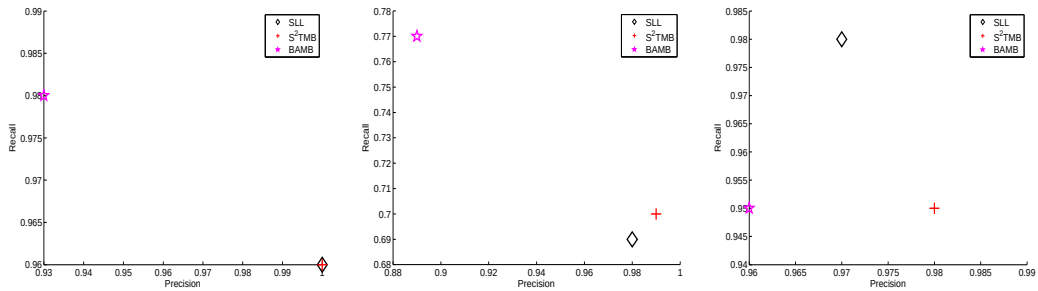
Table 6. Comparison of BAMB with Score-Based MB Methods on sparse benchmark BNs Datasets (size=1,000)

Dataset	Algorithm	F1	Precision	Recall	Time
Child	SLL	**0.88±0.04**	**0.97±0.04**	0.83±0.04	1.22±0.96
	S^2TMB	0.87±0.04	0.96±0.04	0.82±0.05	0.21±0.79
	BAMB	0.85±0.03	0.84±0.04	**0.91±0.01**	**0.11±0.02**
Insurance	SLL	0.68±0.03	0.88±0.03	0.58±0.03	0.64±0.28
	S^2TMB	**0.69±0.02**	**0.90±0.03**	0.60±0.02	0.26±0.45
	BAMB	**0.69±0.02**	0.76±0.04	**0.68±0.01**	**0.12±0.01**
Alarm	SLL	**0.92±0.02**	**0.94±0.03**	**0.93±0.01**	1.12±0.77
	S^2TMB	0.90±0.03	**0.94±0.03**	0.90±0.02	0.44±0.93
	BAMB	0.86±0.01	0.91±0.02	0.86±0.01	**0.11±0.01**
Insurance10	SLL	-	-	-	-
	S^2TMB	-	-	-	-
	BAMB	**0.53±0.25**	**0.60±0.28**	**0.54±0.32**	**1.64±1.26**
Alarm10	SLL	-	-	-	-
	S^2TMB	-	-	-	-
	BAMB	**0.66±0.25**	**0.78±0.28**	**0.65±0.32**	**1.64±0.80**



(a) Child (b) Insurance (c) Alarm

Fig. 6. Precision-recall of BAMB and Score-based MB Methods on sparse benchmark BNs Datasets (size=1,000).



(a) Child (b) Insurance (c) Alarm

Fig. 7. Precision-recall of BAMB and Score-based MB Methods on sparse benchmark BNs Datasets(size=5,000).

Table 7. Comparison of BAMB with Score-Based MB Methods on sparse benchmark BNs Datasets (size=5,000)

Dataset	Algorithm	F1	Precision	Recall	Time
Child	SLL	**0.98±0.00**	**1.00±0.00**	0.96±0.01	5.50±3.13
	S^2TMB	**0.98±0.01**	**1.00±0.00**	0.96±0.01	0.81±2.58
	BAMB	0.95±0.02	0.93±0.02	**0.98±0.02**	**0.40±0.03**
Insurance	SLL	0.79±0.02	0.98±0.02	0.69±0.03	3.39±1.48
	S^2TMB	0.79±0.01	**0.99±0.02**	0.70±0.02	1.31±2.53
	BAMB	**0.80±0.01**	0.89±0.03	**0.77±0.02**	**0.92±0.06**
Alarm	SLL	**0.97±0.01**	0.97±0.02	**0.98±0.00**	4.11±2.57
	S^2TMB	0.95±0.01	**0.98±0.02**	0.95±0.01	1.31±1.82
	BAMB	0.94±0.02	0.96±0.03	0.95±0.01	**0.51±0.03**
Insurance10	SLL	-	-	-	-
	S^2TMB	-	-	-	-
	BAMB	**0.70±0.22**	**0.86±0.23**	**0.64±0.27**	**15.20±13.22**
Alarm10	SLL	-	-	-	-
	S^2TMB	-	-	-	-
	BAMB	**0.76±0.24**	**0.85±0.24**	**0.74±0.29**	**12.87±8.81**

score-based algorithms use the dynamic programming algorithm [22], they can fail to finish the MB discovery due to memory limitation. Specifically, BAMB is the most accurate algorithm on the $Insurance$ network in all cases.

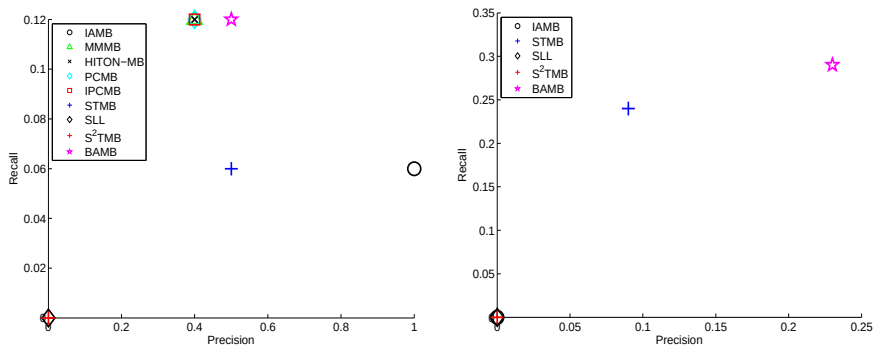
Table 8. Summary of local dense benchmark BNs

Network	Num. Vars	Num. Edges	Max In/out-Degree	Min/Max |PCset|	Domain Range
Hailfinder	56	66	4/16	1/17	2-11
Munin	189	282	3/15	1/15	1-21

5.2 Local Dense Benchmark BN Datasets

We also validate the efficiency and accuracy of our algorithm on local dense networks, because these networks have the nodes with large size of PC, such as the 3rd node of $Hailfinder$, including 17 PC nodes, and the 95th node of $Munin$, including 15 PC nodes. $Hailfinder$ and $Munin$ are described in Table 8. The experimental datasets are also publicly available from [27], and we use the first dataset in 10 datasets with data samples size of 1000. We select the 3rd node of $Hailfinder$ and the 95th node of $Munin$, and find their MBs respectively. In the experiments, we use the same evaluation metrics as in Section 5.1. And we also show the precision-recall in Fig. 8.

From the results in Table 9, in efficiency, BAMB is slower than IAMB but much faster than other constraint-based MB discovery algorithms. And MMMB, HITON-MB, PCMB, and IPCMB fails to generate any output with $Munin$ since the running time exceeds more than one day. In accuracy, BAMB is the most accurate algorithm among all algorithms on each dataset. The dense network means that the size of the PC of each node will be larger than the sparse network. Since BAMB does not need to find PC of each feature in the PC set of the target for identifying spouses, accordingly, given a dense network, BAMB will be more efficient than the divide-and-conquer MB discovery algorithms which need to find PC of each feature in the PC set of the target. In addition, compared with score-based MB



(a) Hailfinder (b) Munin

Fig. 8. Precision-recall of BAMB and other MB Methods on local dense benchmark BNs Datasets.

Table 9. Comparison of BAMB with MB Methods on local dense benchmark BN Datasets

Dataset	Algorithm	F1	Precision	Recall	CI tests	Time
Hailfinder	IAMB	0.11	**1.00**	0.06	**110**	**0.03**
	MMMB	0.18	0.40	**0.12**	1,326	0.32
	HITON-MB	0.18	0.40	**0.12**	810	0.20
	PCMB	0.18	0.40	**0.12**	10,027	2.17
	IPCMB	0.18	0.40	**0.12**	1,295	0.41
	STMB	0.11	0.50	0.06	324	0.10
	SLL	0	0	0	/	0.06
	S^2TMB	0	0	0	/	0.05
	BAMB	**0.19**	0.50	**0.12**	185	0.08
Munin	IAMB	0	0	0	**339**	**0.08**
	MMMB	-	-	-	-	-
	HITON-MB	-	-	-	-	-
	PCMB	-	-	-	-	-
	IPCMB	-	-	-	-	-
	STMB	0.13	0.09	0.24	17,988	4.07
	SLL	0	0	0	/	27.41
	S^2TMB	0	0	0	/	0.14
	BAMB	**0.26**	**0.23**	**0.29**	14,498	2.21

discovery algorithms, BAMB is slower than SLL and S^2TMB, but both SLL and S^2TMB do not learn any results on each dataset. In summary, BAMB and the existing MB algorithms have low accuracy on these nodes in a BN with large size of PC.

5.3 Real-world Datasets

We use ten real-world datasets with various dimensionalities in our experiments. A summary of the datasets is shown in Table 10. The first six datasets are from the UCI machine learning repository [10], the ovarian-cancer is sourced from [14], the madelon, arcene, and dexter datasets are from the NIPS 2003 feature selection challenge. In the experiments, we apply 10-fold cross-validation for all datasets and use the following evaluation metrics:

- Accuracy. We report both the number of selected features and the prediction accuracy of KNN classifier and SVM classifier [7] for BAMB and all the algorithms compared. Prediction accuracy is the percentage of the correctly classified test instances in all test instances.

- Efficiency. We also report running time (in seconds) as the efficiency measure of different algorithms.

Table 10. Summary of real-world datasets

Dataset	Number of features	Number of instances
congress	16	435
wdbc	30	569
unblanced	32	856
sepctf	44	267
sonar	60	208
bankruptcy	147	7,063
ovarian-cancer	2,190	216
madelon	500	2,000
arcene	10,000	100
dexter	20,000	300

In the following tables, we also report the mean result and summarize the win/tie/lose counts of BAMB against other methods in the last rows of each table. "-" denotes that a method fails to generate any output with the corresponding dataset after running more than three days or no features selected, and the best results are highlighted in bold face.

Table 11. Prediction accuracy of BAMB and other constraint-based algorithms using KNN

Dataset	IAMB	MMMB	HITON-MB	PCMB	IPCMB	STMB	BAMB
congress	0.95±0.03	0.95±0.03	0.94±0.03	0.94±0.03	0.95±0.03	**0.96±0.03**	**0.96±0.03**
wdbc	0.77±0.05	0.77±0.05	0.77±0.05	0.77±0.05	**0.79±0.05**	0.78±0.04	**0.79±0.05**
unblanced	**0.81±0.07**	**0.81±0.07**	**0.81±0.07**	-	-	0.57±0.40	**0.81±0.07**
spectf	0.50±0.00	**0.70±0.20**	**0.70±0.20**	**0.70±0.20**	**0.70±0.20**	**0.70±0.20**	**0.70±0.20**
sonar	0.49±0.06	0.83±0.11	0.83±0.11	0.83±0.11	0.83±0.11	0.83±0.07	**0.84±0.06**
bankruptcy	0.88±0.02	0.88±0.01	0.88±0.01	-	-	0.88±0.01	**0.89±0.01**
ovarian-cancer	0.82±0.08	0.86±0.08	0.88±0.04	-	-	0.79±0.11	**0.90±0.06**
madelon	0.58±0.04	0.56±0.04	0.58±0.03	0.50±0.04	0.55±0.02	0.55±0.03	**0.62±0.04**
arcene	0.66±0.17	0.67±0.08	0.68±0.20	0.62±0.16	-	-	**0.72±0.12**
dexter	0.73±0.09	0.81±0.09	0.82±0.09	-	-	-	**0.88±0.07**
mean	0.72	0.78	0.79	-	-	-	**0.81**
win/tie/loss	9/1/0	8/2/0	8/2/0	9/1/0	8/2/0	8/2/0	/

Table 12. Prediction accuracy of BAMB and other constraint-based algorithms using SVM

Dataset	IAMB	MMMB	HITON-MB	PCMB	IPCMB	STMB	BAMB
congress	**0.96±0.02**	**0.96±0.02**	**0.96±0.02**	**0.96±0.02**	**0.96±0.02**	**0.96±0.04**	**0.96±0.03**
wdbc	0.77±0.05	0.77±0.05	0.77±0.05	0.77±0.05	**0.79±0.05**	0.78±0.04	**0.79±0.05**
unblanced	**0.99±0.00**	**0.99±0.00**	**0.99±0.00**	-	-	0.69±0.48	**0.99±0.00**
spectf	0.74±0.16	0.74±0.15	0.74±0.15	0.74±0.15	0.74±0.15	**0.81±0.14**	**0.81±0.14**
sonar	0.72±0.08	**0.86±0.07**	**0.86±0.07**	**0.86±0.07**	**0.86±0.07**	0.82±0.07	0.83±0.06
bankryptcy	**0.90±0.01**	**0.90±0.00**	**0.90±0.00**	-	-	0.89±0.00	**0.90±0.01**
ovarian-cancer	0.88±0.06	0.91±0.04	0.91±0.05	-	-	0.81±0.09	**0.94±0.03**
madelon	**0.63±0.03**	0.60±0.02	0.61±0.03	0.56±0.04	0.61±0.03	0.61±0.04	**0.63±0.04**
arcene	0.68±0.16	0.68±0.16	0.70±0.17	0.62±0.16	-	-	**0.72±0.13**
dexter	0.81±0.07	0.85±0.09	0.85±0.07	-	-	-	**0.91±0.07**
mean	0.81	0.83	0.83	-	-	-	**0.85**
win/tie/loss	6/4/0	6/3/1	6/3/1	8/1/1	7/2/1	8/2/0	/

Table 13. Number of selected features of BAMB and other constraint-based algorithms

Dataset	IAMB	MMMB	HITON-MB	PCMB	IPCMB	STMB	BAMB
congress	**3±0**	**3±1**	**3±1**	**3±1**	**3±1**	5±1	4±1
wdbc	**3±0**	5±1	5±1	4±1	6±1	5±1	6±1
unblanced	2±0	2±0	2±0	-	-	**1±1**	2±0
spectf	**1±0**	29±3	29±3	29±3	29±3	9±2	9±2
sonar	**1±0**	59±1	59±1	59±1	59±1	20±1	20±2
bankruptcy	**9±0**	62±3	58±2	-	-	80±4	52±3
ovarian-cancer	**3±0**	10±2	7±1	-	-	377±103	29±4
madelon	6±0	6±1	6±1	**2±1**	7±2	26±6	9±1
arcene	3±0	7±4	4±1	**2±0**	-	-	15±4
dexter	**4±0**	11±4	12±1	-	-	-	39±3
mean	**4**	19	19	-	-	-	19
win/tie/loss	0/1/9	3/1/6	3/1/6	6/0/4	7/1/2	6/2/2	/

Table 14. Number of CI tests of BAMB and other constraint-based algorithms

Dataset	IAMB	MMMB	HITON-MB	PCMB	IPCMB	STMB	BAMB
congress	**58±6**	1.6e3±1.9e4	930±256	2.3e4±1.3e4	2.5e3±491	211±112	167±41
wdbc	**117±0**	760±110	1.9e3±428	3.2e3±536	1.7e3±170	555±57	444±146
unblanced	92±10	4.9e4±7.0e3	1.7e5±3.0e4	-	-	**69±4**	72±22
spectf	**88±0**	8.2e3±3.7e3	1.1e4±5.5e3	1.1e5±5.6e4	5.1e6±8.3e5	5.8e5±45	1.5e3±851
sonar	**120±0**	2.9e6±5.1e5	6.7e6±1.3e6	2.0e9±4.0e7	4.2e7±2.6e6	1.9e6±6.5e5	2.7e5±7.6e3
bankruptcy	**1.4e3±0**	1.1e6±2.6e5	1.3e7±3.3e6	-	-	1.4e6±1.6e5	9.7e5±2.0e5
ovarian-cancer	**9.0e3±669**	4.8e4±6.3e3	5.2e5±2.9e5	-	-	9.4e5±3.8e5	5.7e4±1.3e4
madelon	**3.0e3±0**	3.7e3±581	6.2e3±2.9e3	9.1e3±4.6e3	2.8e4±5.4e3	8.1e3±838	5.3e3±1.2e3
arcene	**4.0e4±44**	1.6e5±8.2e4	1.7e7±2.7e7	3.2e5±1.4e5	-	-	6.7e4±2.0e4
dexter	**4.7e4±171**	7.9e5±8.4e5	3.6e5±7.0e4	-	-	-	2.8e5±2.5e4
mean	**1.0e4**	5.1e5	1.6e6	-	-	-	1.4e5
win/tie/loss	1/0/9	8/0/2	10/0/0	10/0/0	10/0/0	9/0/1	/

Table 15. Running time (in seconds) of BAMB and other constraint-based algorithms

Dataset	IAMB	MMMB	HITON-MB	PCMB	IPCMB	STMB	BAMB
congress	**0.11**	2.15	2.36	21.42	3.44	0.59	0.51
wdbc	**0.54**	3.57	8.87	17.46	8.01	1.43	1.22
unblanced	0.36	87.94	269.39	-	-	**0.27**	0.28
spectf	**0.29**	8.65	9.85	74.38	2,039.16	308.19	7.37
sonar	**0.17**	1,345.32	4,385.57	92,628.46	22,371.28	1,189.69	215.89
bankruptcy	**65.43**	43,358.15	173,522.83	-	-	80,087.51	30,150.02
ovarian-cancer	**229.64**	1,081.96	13,331.72	-	-	29,424.55	1,269.48
madelon	**109.88**	157.81	289.84	388.79	1,227.08	348.42	184.14
arcene	**1,636.45**	4,752.38	130,688.32	14,187.03	-	-	2,042.33
dexter	**16,755.21**	168,136.09	76,008.21	-	-	-	49,850.37
mean	**1,879.81**	21,893.40	39,851.70	-	-	-	8,391.16
win/tie/loss	1/0/9	8/0/2	10/0/0	10/0/0	10/0/0	9/1/0	/

5.3.1 Comparison of BAMB with Constraint-Based MB Methods on Real-world Datasets. In this section, we present the results obtained by BAMB in comparison with the state-of-the-art constraint-based MB discovery algorithms, IAMB, MMMB, HITON-MB, PCMB, IPCMB, and STMB.

1) Prediction accuracy: Tables 11 and 12 summarize the prediction accuracy of BAMB against IAMB, MMMB, HITON-MB, PCMB, IPCMB, and STMB using KNN and SVM, respectively. With the counts of win/tie/lose, we can see that BAMB is superior to other algorithms on most datasets using SVM. Furthermore, using KNN, BAMB is never worse

than other algorithms in prediction accuracy. Since there are no overlapping PC sets during the symmetry check, PCMB and IPCMB fail on the *unblanced* dataset. On average, IAMB is worse than the other algorithms in prediction accuracy. In particular, on the *spectf*, *sonar*, *arcene*, and *dexter* datasets, IAMB is 6% less accurate than BAMB using KNN. In theory, since both BAMB and this type of algorithms performing exhaustive subset search within the features selected currently, BAMB has comparable accuracy with them. According to the experimental results on the benchmark BN datasets, BAMB is also comparable with this type of algorithm in accuracy. However, the experimental results on real-world datasets show that BAMB is more accurate (recall value) than this type of algorithms. This means that BAMB can find more true positives (i.e., features within true MBs) than the other algorithms according to the experiments on the benchmark BN datasets. Because MMMB, HITON-MB, PCMB, and IPCMB find spouses from the PC set of each feature in the found PC set of the target, if they mistakenly lose some nodes during the actual calculation of PC discovery, then it will lead to the loss of some true spouses of the target. The strategy of BAMB of finding spouses from non-PC set can avoid losing the candidate of spouses of the target, which improve the prediction accuracy. Meanwhile, BAMB also gets much more false positives in its output, then BAMB has lower accuracy (precision value) on BNs.

2) Number of selected features: Table 13 shows the numbers of selected features by BAMB, IAMB, MMMB, HITON-MB, PCMB, IPCMB, and STMB. According to the counts of win/tie/lose, BAMB is also very competitive with its rivals. This means that BAMB selects a few features. STMB selects more features in *ovarian-cancer* and *madelon* and achieves lower accuracy than the other methods.

3) Efficiency: The number of CI tests is corresponding to runtime, and the experimental results in Table 14 and 15 are consistent with the discussion of the computational complexity in Section 4.4. Due to the high computational costs, PCMB, IPCMB, and STMB fail on the *bankruptcy*, *ovarian-cancer*, *arcene* and *dexter* datasets (the running time exceeding three days). Although IAMB is the fastest algorithm under comparison, IAMB almost gets the worst prediction accuracy. BAMB shows a comparable efficiency with IAMB and much faster than the remaining algorithms on most datasets. Especially on *unblanced*, BAMB is faster than IAMB. On average in terms of the number of CI tests, BAMB is 3.6 times faster than MMMB, and 11.2 times faster than HITON-MB. On average, BAMB is 2.6 times faster than MMMB and 4.8 times faster than HITON-MB. Especially with the high-dimensional datasets *arcene* and *dexter* with more than 10,000 features, BAMB is at least 2 times faster than MMMB and HITON-MB.

Table 16. Prediction accuracy of score-based MB discovery algorithms and BAMB using KNN

Dataset	SLL	S^2TMB	BAMB
congress	0.95±0.04	0.94±0.04	**0.96±0.03**
wdbc	0.77±0.05	0.78±0.05	**0.79±0.05**
unblanced	0.24±0.32	0.24±0.32	**0.81±0.07**
spectf	0.61±0.09	0.61±0.09	**0.70±0.20**
sonar	0.53±0.08	0.54±0.09	**0.84±0.06**
bankruptcy	-	0.83±0.03	**0.89±0.01**
ovarian-cancer	-	0.86±0.07	**0.90±0.06**
madelon	0.52±0.04	0.52±0.04	**0.62±0.04**
arcene	-	0.69±0.15	**0.72±0.12**
dexter	-	-	**0.88±0.07**
win/tie/loss	10/0/0	10/0/0	/

Table 17. Prediction accuracy of score-based MB discovery algorithms and BAMB using SVM

Dataset	SLL	S^2TMB	BAMB
congress	0.95±0.03	0.95±0.03	**0.96±0.03**
wdbc	0.77±0.05	0.78±0.05	**0.79±0.05**
unblanced	0.40±0.51	0.40±0.51	**0.99±0.00**
spectf	0.72±0.15	0.72±0.15	**0.81±0.14**
sonar	0.69±0.09	0.70±0.07	**0.83±0.06**
bankruptcy	-	0.89±0.00	**0.90±0.01**
ovarian-cancer	-	0.89±0.06	**0.94±0.03**
madelon	0.57±0.03	0.57±0.03	**0.63±0.04**
arcene	-	**0.73±0.20**	0.72±0.13
dexter	-	-	**0.91±0.07**
win/tie/loss	10/0/0	9/0/1	/

Table 18. Number of selected features of score-based MB discovery algorithms and BAMB

Dataset	SLL	S^2TMB	BAMB
congress	**4±1**	**4±1**	**4±1**
wdbc	**5±1**	7±1	6±1
unblanced	**0±1**	**0±1**	2±0
spectf	**3±1**	**3±1**	9±2
sonar	**2±1**	**2±1**	20±2
bankruptcy	-	**9±1**	52±3
ovarian-cancer	-	**7±2**	29±4
madelon	6±1	**5±1**	9±1
arcene	-	**6±1**	15±4
dexter	-	-	**39±3**
win/tie/loss	4/1/5	2/1/7	/

Table 19. Running time (in seconds) of score-based MB discovery algorithms and BAMB

Dataset	SLL	S^2TMB	BAMB
congress	6.71	1.03	**0.51**
wdbc	42.75	74.01	**1.22**
unblanced	2.73	**0.13**	0.24
spectf	**0.21**	0.95	7.37
sonar	4.37	**0.92**	215.89
bankruptcy	-	**10,232.59**	30,150.02
ovarian-cancer	-	41,695.87	**1,269.48**
madelon	412.51	570.34	**184.14**
arcene	-	26,337.03	**2,042.33**
dexter	-	-	**49,850.37**
win/tie/loss	8/0/2	6/0/4	/

5.3.2 Comparison of BAMB with Score-Based MB Methods on Real-world Datasets. In this section, we compare the BAMB algorithm with the state-of-the-art score-based methods, SLL and S^2TMB, and the results are discussed as follows.

1) Prediction accuracy: From Tables 16 and 17, we see that using either KNN or SVM, BAMB is more accurate than SLL and S^2TMB. Even SLL and S^2TMB fail on high-dimensional datasets due to memory limitation. Especially on the four datasets *unblanced*, *spectf*, *sonar*, and *madelon*, BAMB is 9% more accurate than SLL and S^2TMB using KNN.

2) Number of selected features: Table 18 reports the numbers of selected features of BAMB, SLL, and S^2TMB. From the result, BAMB is very competitive with SLL and S^2TMB.

3) Running time: SLL and S^{2}TMB are implemented in C++, while BAMB is implemented in MATLAB. In general, programs written in C++ may be more efficient than the one written by MATLAB. However, from Table 19, we can see that BAMB is much faster than SLL and S^{2}TMB on the high-dimensional datasets. Especially, BAMB is 32.8 times faster than S^{2}TMB on *ovarian-cancer*, and 20.7 times faster than S^{2}TMB on *arcene*.

Table 20. Prediction accuracy of the well-established feature selection methods and BAMB using KNN

Dataset	FCBF	mRMR	SPFS-LAR	MRF	BAMB
congress	**0.96±0.03**	**0.96±0.03**	**0.96±0.02**	0.95±0.03	**0.96±0.03**
wdbc	0.78±0.05	**0.80±0.04**	0.49±0.05	0.75±0.05	0.79±0.05
unblanced	0.68±0.06	0.72±0.06	0.70±0.09	0.03±0.02	**0.81±0.07**
spectf	0.64±0.22	0.71±0.18	0.60±0.16	**0.74±0.16**	0.70±0.20
sonar	0.60±0.09	0.82±0.09	**0.84±0.06**	0.77±0.07	**0.84±0.06**
bankruptcy	0.85±0.02	0.87±0.01	0.88±0.01	0.87±0.01	**0.89±0.01**
ovarian-cancer	0.88±0.08	**0.92±0.05**	0.78±0.11	0.51±0.10	0.90±0.06
madelon	0.51±0.05	0.54±0.03	0.49±0.04	0.51±0.03	**0.62±0.04**
arcene	0.65±0.07	0.71±0.12	0.63±0.10	0.65±0.15	**0.72±0.12**
dexter	0.85±0.08	0.87±0.07	0.76±0.06	0.77±0.07	**0.88±0.07**
mean	0.74	0.79	0.71	0.65	**0.81**
win/tie/loss	9/1/0	6/1/3	8/2/0	9/0/1	/

Table 21. Prediction accuracy of the well-established feature selection methods and BAMB using SVM

Dataset	FCBF	mRMR	SPFS-LAR	MRF	BAMB
congress	**0.96±0.03**	0.95±0.03	0.94±0.03	0.94±0.04	**0.96±0.03**
wdbc	0.78±0.04	**0.80±0.05**	0.63±0.00	0.75±0.06	0.79±0.05
unblanced	**0.99±0.00**	**0.99±0.00**	**0.99±0.00**	**0.99±0.00**	**0.99±0.00**
spectf	0.74±0.16	0.80±0.15	0.63±0.10	**0.81±0.14**	**0.81±0.14**
sonar	0.73±0.07	0.79±0.08	0.82±0.08	0.78±0.11	**0.83±0.06**
bankruptcy	0.89±0.00	0.89±0.00	**0.90±0.00**	0.89±0.00	**0.90±0.01**
ovarian-cancer	**0.94±0.05**	0.93±0.02	0.85±0.09	0.55±0.08	**0.94±0.04**
madelon	0.58±0.03	0.58±0.04	0.50±0.03	0.51±0.02	**0.63±0.04**
arcene	0.62±0.14	**0.72±0.08**	0.63±0.12	0.67±0.11	**0.72±0.13**
dexter	0.86±0.07	0.90±0.07	0.87±0.06	0.83±0.06	**0.91±0.07**
mean	0.81	0.83	0.78	0.77	**0.85**
win/tie/loss	7/3/0	7/2/1	8/2/0	8/2/0	/

Table 22. Number of Selected Features of FCBF and BAMB

Dataset	FCBF	BAMB
congress	**3±1**	4±1
wdbc	**5±0**	6±1
unblanced	**1±0**	2±0
spectf	**7±1**	9±2
sonar	**2±0**	20±2
bankruptcy	**10±1**	52±3
ovarian-cancer	**13±1**	29±4
madelon	20±1	**9±1**
arcene	33±2	**15±4**
dexter	49±2	**39±3**
mean	**14**	19
win/tie/loss	3/0/7	/

5.3.3 Comparison of BAMB with Other Feature Selection Methods on Real-world Datasets. In this section, we compare the BAMB algorithm with two well-established feature selection methods, FCBF and mRMR, and two state-of-the-art feature selection algorithms, SPFS-LAR and MRF. The experimental results are discussed as follows.

1) Prediction accuracy: In Tables 20 and 21, with the counts of win/tie/lose and average results, BAMB is much more accurate than other methods using both KNN and SVM. Specifically, BAMB is never worse than FCBF and SPFS-LAR in prediction accuracy using both KNN and SVM on each dataset. And on the datasets with a large number of features: *madelon*, *arcene*, and *dexter*, BAMB's advantage in prediction accuracy is more obvious.

2) Number of selected features: Since mRMR, SPFS-LAR, and MRF choose the same number of selected features with BAMB in each dataset, we only give the numbers of selected features of BAMB and FCBF in Table 22. However, in any case, the prediction accuracy of BAMB is better than mRMR, SPFS-LAR, and MRF. This means that BAMB chooses more correct MB features. With the counts of win/tie/lose, BAMB is also very competitive with FCBF, especially on the datasets with a large number of features: *madelon*, *arcene*, and *dexter*.

3) Running time: FCBF, mRMR, SPFS-LAR, and MRF are implemented in MATLAB and C language, but the core code of FCBF, mRMR, SPFS-LAR, and MRF are written in C language, so we do not give their running time here.

6 CONCLUSIONS

We have presented BAMB, a novel constraint-based MB discovery algorithm for balancing the efficiency and accuracy of MB discovery for feature selection, by finding candidate PC and spouses and removing false positives from the candidate set in one go. The experimental results have shown that the BAMB algorithm outperforms the state-of-the-art constraint-based and score-based Markov blanket feature selection methods and other well-established feature selection methods. Future research could focus on how to efficiently and accurately find MB on nodes in a BN with large size of PC, because the existing MB discovery algorithms are not suitable for these nodes according to our experimental results in Section 5.2.

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, 91746209, and 61673152), and the Anhui Province Key Research and Development Plan (No. 201904a05020073).

REFERENCES

[1] Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. 2010. 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, Jan (2010), 171–234.

[2] Constantin F Aliferis, Ioannis Tsamardinos, and Alexander Statnikov. 2003. HITON: a novel Markov Blanket algorithm for optimal variable selection. In *AMIA Annual Symposium Proceedings*, Vol. 2003. American Medical Informatics Association, 21.

[3] Ingo A Beinlich, Henri Jacques Suermondt, R Martin Chavez, and Gregory F Cooper. 1989. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *AIME 89*. Springer, 247–256.

[4] John Binder, Daphne Koller, Stuart Russell, and Keiji Kanazawa. 1997. Adaptive probabilistic networks with hidden variables. *Machine Learning* 29, 2-3 (1997), 213–244.

[5] Verónica Bolón-Canedo, D Rego-Fernández, Diego Peteiro-Barral, Amparo Alonso-Betanzos, Bertha Guijarro-Berdiñas, and Noelia Sánchez-Maroño. 2018. On the scalability of feature selection methods on high-dimensional data. *Knowledge and Information Systems* (2018), 1–48.
[6] Ruichu Cai, Zhenjie Zhang, and Zhifeng Hao. 2011. BASSUM: A Bayesian semi-supervised method for classification feature selection. *Pattern Recognition* 44, 4 (2011), 811–820.
[7] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* 2, 3 (2011), 27.
[8] Qiang Cheng, Hongbo Zhou, and Jie Cheng. 2011. The fisher-markov selector: fast selecting maximally separable feature subset for multiclass classification with applications to high-dimensional data. *IEEE transactions on pattern analysis and machine intelligence* 33, 6 (2011), 1217–1233.
[9] AP Dawid, RG Cowell, SL Lauritzen, and DJ Spiegelhalter. 1999. Probabilistic networks and expert systems. Springer-Verlag.
[10] Dua Dheeru and Efi Karra Taniskidou. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml
[11] Shunkai Fu and Michel C Desmarais. 2008. Fast Markov blanket discovery algorithm via local learning within single pass. In *Conference of the Canadian Society for Computational Studies of Intelligence*. Springer, 96–107.
[12] Tian Gao and Qiang Ji. 2017. Efficient Markov blanket discovery and its application. *IEEE transactions on cybernetics* 47, 5 (2017), 1169–1179.
[13] Tian Gao and Qiang Ji. 2017. Efficient score-based Markov Blanket discovery. *International Journal of Approximate Reasoning* 80 (2017), 277–293.
[14] Ben Hitt and Peter Levine. 2006. Multiple high-resolution serum proteomic features for ovarian cancer detection. US Patent App. 11/093,018.
[15] Daphne Koller and Mehran Sahami. 1996. *Toward optimal feature selection*. Technical Report. Stanford InfoLab.
[16] Dimitris Margaritis and Sebastian Thrun. 2000. Bayesian network induction via local neighborhoods. In *Advances in neural information processing systems*. 505–511.
[17] T Niinimki and Pekka Parviainen. 2012. Local structure discovery in bayesian networks. In *28th Conference on Uncertainty in Artificial Intelligence, UAI 2012; Catalina Island, CA; United States; 15 August 2012 through 17 August 2012*. 634–643.
[18] Judea Pearl. 1988. Morgan Kaufmann series in representation and reasoning. Probabilistic reasoning in intelligent systems: Networks of plausible inference.
[19] Judea Pearl. 2014. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.
[20] Jose M Pena, Roland Nilsson, Johan Björkegren, and Jesper Tegnér. 2007. Towards scalable and data efficient learning of Markov boundaries. *International Journal of Approximate Reasoning* 45, 2 (2007), 211–232.
[21] Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence* 27, 8 (2005), 1226–1238.
[22] Tomi Silander and Petri Myllymaki. 2006. A simple approach for finding the globally optimal bayesian network structure. In *Proceedings of the Twenty-Second Annual Conference on Uncertainty in Artificial Intelligence (UAI)*. 445–452.
[23] Peter Spirtes, Clark N Glymour, and Richard Scheines. 2000. *Causation, prediction, and search*. MIT press.
[24] A Statnikov, I Tsamardinos, and CF Aliferis. 2003. An algorithm for generation of large Bayesian networks. *Department of Biomedical Informatics, Discovery Systems Laboratory, Vanderbilt University, Tech. Rep. Technical Report DSL-03-01* (2003).
[25] Ioannis Tsamardinos, Constantin F Aliferis, and Alexander Statnikov. 2003. 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, 673–678.
[26] Ioannis Tsamardinos, Constantin F Aliferis, Alexander R Statnikov, and Er Statnikov. 2003. Algorithms for Large Scale Markov Blanket Discovery.. In *FLAIRS conference*, Vol. 2. 376–380.
[27] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. 2006. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine learning* 65, 1 (2006), 31–78.
[28] Xiaowei Xue, Min Yao, and Zhaohui Wu. 2018. A novel ensemble-based wrapper method for feature selection using extreme learning machine and genetic algorithm. *Knowledge and Information Systems*

57, 2 (2018), 389–412.
[29] Sandeep Yaramakala and Dimitris Margaritis. 2005. Speculative Markov blanket discovery for optimal feature selection. In *Data mining, fifth IEEE international conference on*. IEEE, 4–pp.
[30] Kui Yu, Lin Liu, and Jiuyong Li. 2018. A Unified View of Causal and Non-causal Feature Selection. *arXiv preprint arXiv:1802.05844* (2018).
[31] Kui Yu, Lin Liu, Jiuyong Li, Wei Ding, and Thuc Le. 2019. Multi-Source Causal Feature Selection. *IEEE transactions on pattern analysis and machine intelligence* DOI: 10.1109/TPAMI.2019.2908373 (2019).
[32] Kui Yu, Xindong Wu, Wei Ding, and Jian Pei. 2016. Scalable and accurate online feature selection for big data. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11, 2 (2016), 16.
[33] Lei Yu and Huan Liu. 2004. Efficient feature selection via analysis of relevance and redundancy. *Journal of machine learning research* 5, Oct (2004), 1205–1224.
[34] Zheng Zhao, Lei Wang, Huan Liu, and Jieping Ye. 2013. On similarity preserving feature selection. *IEEE Transactions on Knowledge and Data Engineering* 25, 3 (2013), 619–632.