Causality-based Feature Selection: Methods and Evaluations

Kui Yu   YUKUI@HFUT.EDU.CN
Xianjie Guo   XIANJIE_REASONLESS@163.COM
School of Computer Science and Information Engineering
Hefei University of Technology, Hefei, 230601, China

Lin Liu   LIN.LIU@UNISA.EDU.AU
Jiuyong Li   JIUYONG.LI@UNISA.EDU.AU
School of Information Technology and Mathematical Sciences
University of South Australia, Adelaide, 5095, SA, Australia

Hao Wang   JSJXWANGH@HFUT.EDU.CN
Zhaolong Ling   Z_DRAGONL@163.COM
School of Computer Science and Information Engineering
Hefei University of Technology, Hefei, 230601, China

Xindong Wu   WUXINDONG@MININGLAMP.COM
Mininglamp Academy of Sciences, Mininglamp Technology, Bejing, 100084, China

Editor:

Abstract

Feature selection is a crucial preprocessing step in data analytics and machine learning. Classical feature selection algorithms select features based on the correlations between predictive features and the class variable and do not attempt to capture causal relationships between them. It has been shown that the knowledge about the causal relationships between features and the class variable has potential benefits for building interpretable and robust prediction models, since causal relationships imply the underlying mechanism of a system. Consequently, causality-based feature selection has gradually attracted greater attentions and many algorithms have been proposed. In this paper, we present a comprehensive review of recent advances in causality-based feature selection. To facilitate the development of new algorithms in the research area and make it easy for the comparisons between new methods and existing ones, we develop the first open-source package, called CausalFS, which consists of most of the representative causality-based feature selection algorithms (available at https://github.com/kuiy/CausalFS). Using CausalFS, we conduct extensive experiments to compare the representative algorithms with both synthetic and real-world data sets. Finally, we discuss some challenging problems to be tackled in future causality-based feature selection research.

Keywords: Feature selection, Causality-based feature selection, Bayesian network, Markov boundary

1. Introduction

Feature selection plays an essential role in high-dimensional data analytics (Guyon and Elisseeff, 2003; Aliferis et al., 2010a; Li et al., 2017; Brown et al., 2012; Wu et al., 2013) and it is widely employed in all kinds of machine learning solutions. Feature selection is to find a

subset of features from a large number of predictive features for building predictive models for a target or class variable of interest. For example, gene (i.e., feature) selection can identify a small number of informative genes from a high-dimensional gene data set for predicting a disease or directing experimental studies to validate the identified genes (as genetic factors of a disease) in laboratories. Now feature selection is more critical than ever, since a data set with high-dimensionality has become ubiquitous in various applications (Zhai et al., 2014; Li and Liu, 2017; Yu et al., 2016). In the previous example, a gene expression data set may easily have more than 10,000 predictive features (Saeys et al., 2007). For another example, the Web Spam Corpus 2011 collected approximately 16 million predictive features for malicious web detection (Wang et al., 2012). Almost all machine learning methods may not directly work on data sets of such high dimensionality without feature selection. As a result, in the last two decades, feature selection has been well studied and has achieved great success in reducing computational costs of learning and improving the generalization ability of predictive models (Li et al., 2017).

Existing feature selection methods can be broadly categorized into filter, wrapper, and embedded methods. A filter method is independent of a predictive model, whereas the other two types of methods are predictive model dependent. Due to their independence of predictive models, filter methods are able to achieve fast processing speed and have no bias on specific predictive models. With the rapid increase of high dimensional data, filter methods have been attracting more attentions than ever. In this paper, we focus on causality-based feature selection, an emerging successful type of filter methods. In feature selection, a feature is considered as a strongly relevant feature, or a weakly relevant feature, or an irrelevant feature with respect to a class variable of interest (Kohavi and John, 1997). A classical feature selection method aims to find a subset of relevant features based on the correlations between (predictive) features and the class variable (Guyon and Elisseeff, 2003). In general, correlations do not capture the causal relationships between features and the class variable, but only their co-occurrences. Recent studies have shown that causal features may provide the following potential benefits in feature selection for classification (Guyon et al., 2007; Aliferis et al., 2010a).

- Causal features can improve the interpretability of predictive models (Hofman et al., 2017; Ribeiro et al., 2016). Correlations capture only the co-occurrence of features and the class variable, hence the selected features often do not provide a convincing interpretation for predictions. For example, a strong correlation between *yellow fingers* (of a smoker) and *lung cancer* may be found in patient records, making *yellow fingers* a good predictive feature of *lung cancer*. However, clearly *yellow fingers* is not a reasonable interpretation at all for *lung cancer*. In fact, the causes of *lung cancer*, such as *smoking*, can provide a reasonable interpretation for the prediction of lung cancer.

- Causal features can improve the robustness of predictive models (Athey, 2017; Peters et al., 2016; Zhang et al., 2015; Magliacane et al., 2018). Causal relationships imply the underlying mechanism about the class variable and thus they are persistent across different settings or environments. For example, we want to build a predictive model to diagnose *lung cancer* using historical patient data. Based on the historical data, a predictive model built using non-causal features such as yellow fingers may not
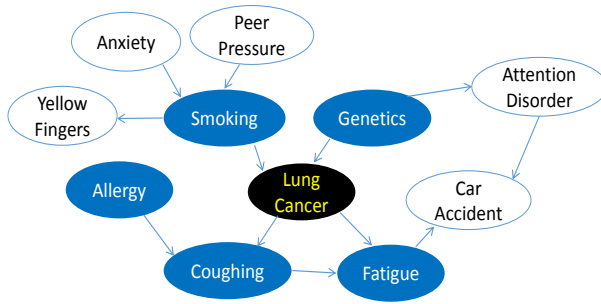
Figure 1: An example of a MB in a lung-cancer Bayesian network

produce good predictions. This is because nowdays smokers are very careful to hide their smoking habit and do not leave yellow stain on their fingers, and hence the distribution of symptoms presented by patients may become different from that of the historical data. In contrast, if the causes of *lung cancer* of patients (such as *smoking*) were selected as the predictive features, a model built on the historical data will be robust.

In recent years, causality-based feature selection has gradually attracted more and more attentions from both machine learning and causal discovery domains (Aliferis et al., 2010a; Guyon et al., 2007; Borboudakis and Tsamardinos, 2019; Yu et al., 2018b). Causality-based feature selection methods identify a Markov boundary (MB) or a subset of the MB such as parents and children (PC) (i.e., direct causes and direct effects) of the class variable from a data set. The notion of MB was proposed in the context of a Bayesian network (BN) (Pearl, 2014). If the model generating the data set can be faithfully represented by a BN, then the MB of the class variable is unique and consists of the parents, children, and spouses (i.e., other parents of the class variable's children) of the class variable. Figure 1 gives an example of an MB in the BN of lung cancer (Guyon et al., 2007). The MB of *Lung cancer* includes *Smoking* and *Gentics* (parents), *Coughing* and *Fatigue* (children), and *Allergy* (spouse). As can be seen in Figure 1, the MB of a class variable implies the local causal relationships between the class variable and the features (variables) in its MB. Most importantly, under certain assumptions (to be discussed in Section 2), the MB of the class variable is the minimal feature subset with maximum predictivity for classification since all other features are probabilistically independent of the class variable conditioning on its MB (Pearl, 2014; Koller and Sahami, 1996; Tsamardinos and Aliferis, 2003).

Since causality-based feature selection explicitly induces the local causal relationships around the class variable and has theoretical guarantees, it becomes a promising direction (Guyon et al., 2007; Aliferis et al., 2010a). In the past decade, many efforts have been made on causality-based feature selection and many algorithms have been proposed (without learning an entire BN structure involving all features in a data set) (Aliferis et al., 2010a; Borboudakis and Tsamardinos, 2019; Yu et al., 2018b). The developed causality-based feature selection algorithms provide a new and complementary algorithmic methodology to enrich feature selection, especially for achieving explainable and robust machine learning methods.

3

To advance the research in causality-based feature selection, a comprehensive review of the state-of-the-art techniques in this area is in need. However, so far, there has not been such a review available. Aliferis et al. (Guyon et al., 2007; Aliferis et al., 2010a,b) proposed a general local learning framework for causality-based feature selection, which are focused on three specific causality-based feature selection algorithms (e.g., MMMB, HITON-MB, and semi-HITON-MB) and their extensions to the BN structure learning, but the work is not a survey paper. Guyon et al. (Guyon et al., 2007) presented a comparison of the motivations and pros/cons of causality-based and classical feature selection approaches at the conceptual level, but again they did not provide a survey of causality-based feature selection algorithms. Recently, Yu et al. (Yu et al., 2018b) discussed some representative causality-based and classical feature selection algorithms, but the work was focused on developing a unified view to link causality-based feature selection with classical feature selection instead of presenting an extensive review of causality-based feature selection algorithms. There have been some recent reviews on causal inference, such as (Guo et al., 2018; Glymour et al., 2019; Zhang et al., 2017), but they mainly focused on the advances on learning causal relations between features. Meanwhile almost all reviews regarding feature selection focused on classical feature selection methods in the past decades (Guyon and Elisseeff, 2003; Brown et al., 2012; Li and Liu, 2017; Li et al., 2017). In summary, so far there is little work on a comprehensive review of causality-based feature selection algorithms.

In addition, there is no any open-source toolbox/package which implements existing causality-based feature selection algorithms. An open-source toolbox plays a crucial role for facilitating the development of new algorithms and making comparisons between the new methods and existing ones easy, and it may further promote both scientific and practical studies in machine learning and causal discovery.

In this paper, we make the following contributions to fill the gaps discussed above.

- We extensively review the causality-based feature selection methods, including the two types of causality-based feature selection methods, constraint-based and score-based methods, and the algorithms of distinguishing causes and effects. To the best of our knowledge, this is the first attempt on presenting an extensive survey of causality-based feature selection and its recent advances.

- We develop the first comprehensive open-source package that implements the representative and state-of-the-art causality-based feature selection algorithms. The package is written in C language, easy to use, and completely open source.

- We conduct a comprehensive empirical evaluation on representative causality-based feature selection algorithms with both synthetic and real-world data sets.

The rest of the paper is organized as follows. Section 2 gives basic background knowledge. Section 3 reviews constraint-based methods. Section 4 reviews score-based methods. Section 5 discusses the algorithms for distinguishing causes from effects using the outputs of causality-based feature selection algorithms. Section 6 presents the open-source package of causality-based feature selection. Section 7 reports the evaluation results. Section 8 concludes the paper and discusses some open problems.

2. Markov boundary and Causality-based feature selection

In this section, we first briefly introduce the background knowledge of MB and BN, then we link MB with feature selection.

2.1 Bayesian network, Markov boundary, and causality-based feature selection

Let C be a class variable and $F = \{F_1, F_2, \cdots, F_M\}$ be a feature set including M distinct features. We use $F_i \perp\!\!\!\perp F_j | S$, where $i \neq j$ and $S \subseteq F \setminus \{F_i, F_j\}$, to denote that F_i is conditionally independent of F_j given feature set S, and $F_i \not\perp\!\!\!\perp F_j | S$ to represent that F_i is conditionally dependent on F_j given S.

Let S be any set of variables within V, we use $S \setminus V_i$ as the shorthand of $S \setminus \{V_i\}$ and $S \cup V_i$ as the shorthand of $S \cup \{V_i\}$. Let $V = F \cup C = \{V_1, V_2, \cdots, V_{M+1}\}$, $V_i = F_i$ ($1 \leq i \leq M$), and $V_{M+1} = C$. Let $P(V)$ be the joint probability distribution over V and $G = (V, E)$ represent a directed acyclic graph (DAG) with nodes V and edges E, where an edge $V_i \rightarrow V_j$ denotes that V_i is a parent (direct cause) of V_j while V_j is a child (direct effect) of V_i. The triplet $\langle V, G, P(V) \rangle$ is called a BN if and only if $\langle V, G, P(V) \rangle$ satisfies the Markov condition: every node of G is independent of any subset of its non-descendants conditioning on the parents of the node (Pearl, 2014). A BN encodes the joint probability $P(V)$ over V and decomposes $P(V)$ into a product of the conditional probability distributions of the variables given their parents in G. Let $Pa(V_i)$ be the set of parents of V_i in G. Then, we have

$$P(V_1, V_2, \cdots, V_{M+1}) = \prod_{i=1}^{M+1} P(V_i | Pa(V_i)) \tag{1}$$

In the following, we introduce the key concepts and assumptions related to BN, Markov boundary, and causality-based feature selection.

Definition 1 (Faithfulness) *(Pearl, 2014) Given a BN $< V, G, P(V) >$, G is faithful to $P(V)$ if and only if every conditional independence present in P is entailed by G and the Markov condition. $P(V)$ is faithful to G if and only if there exists a DAG G such that G is faithful to $P(V)$.*

Definition 2 (Causal sufficiency) *(Pearl, 2014; Spirtes et al., 2000) Causal sufficiency assumes that any common cause of two or more variables in V is also in V.*

Before introducing Markov boundary, we first present the concept of Markov blanket.

Definition 3 (Markov blanket, Mb) *(Pearl, 2014) A Markov blanket of the class variable C ($Mb(C)$) in V is a set of variables conditioned on which all other variables are independent of C, that is, for every $V_i \in V \setminus (Mb(C) \cup C)$, $C \perp\!\!\!\perp V_i | Mb(C)$.*

Clearly, a variable may have multiple Markov blankets. For example, the set of all variables V excluding C is also a Markov blanket of C. In practice, we are often interested in minimal Markov blankets.

Definition 4 (Markov boundary, MB) *(Pearl, 2014) If no proper subset of $Mb(C)$ satisfies the definition of Markov blanket of C, then $Mb(C)$ is called the Markov boundary of C, denoted as $MB(C)$.*

Theorem 1 states the uniqueness of MBs and what the MB of a node is in a BN.

Theorem 1 *(Pearl, 2014) Under the faithfulness assumption, the MB of a node in a BN is unique and it consists of the node's parents (direct causes), children (direct effects), and spouses (other parents of the node's children).*

In a BN, the MB of a node renders the node statistically independent of all the remaining nodes conditioning on the MB (Pearl, 2014), as shown in Proposition 1 below.

Proposition 1 *(Pearl, 2014) In a BN, let $MB(X)$ be the MB of node X, $\forall Y \in V \setminus (MB(X) \cup X)$, $X \perp\!\!\!\perp Y | MB(X)$ holds.*

For example, given the MB of *Lung cancer* in Figure 1, i.e., *Smoking*, *Gentics*, *Coughing*, *Fatigue*, and *Allergy*, *Lung cancer* is independent of the remaining nodes. Proposition 1 illustrates that learning the MB of the class variable is actually a procedure of feature selection (Tsamardinos and Aliferis, 2003; Aliferis et al., 2010a). Koller and Sahami (Koller and Sahami, 1996) were the first to introduce the concept of MBs to feature selection. The work in (Tsamardinos and Aliferis, 2003; Yu et al., 2018b) stated that under the faithfulness assumption, (1) the strongly relevant features belong to the MB of the class variable, and (2) the MB is the minimal feature subset with maximum predictivity for classification. Existing causality-based feature selection algorithms aim to learn the MB of the class variable or a subset of the MB (Guyon et al., 2007; Aliferis et al., 2010a).

2.2 The general strategy of causality-based feature selection

Forward-backward feature selection is one of the most basic and commonly-used feature selection frameworks. The forward phase of forward-backward selection starts with a (usually empty) set of features and adds features to it, until a given stopping criterion is met, a the backward phase of forward-backward selection starts with a set of features (usually obtained from the forward phase) and then removes features from that set until a stopping criterion is met. Under the forward-backward framework, there are two general strategies for feature selection. The standard forward-backward feature selection (SFBS) strategy (Algorithm 1) starts with a forward phase for selecting a subset of candidate features S and then a backward phase for removing false positives from S. The interleaving forward-backward feature selection (IFBS) strategy (Algorithm 2) performs the froward phase and backward phase alternatively. Specifically, if there are new features added to S at the forward phase, IFBS immediately triggers the backward phase and implements both phases alternatively. Existing causality-based feature selection algorithms adopt either the SFBS or the IFBS strategy, and they employ a selection criterion Φ, such as information gain, independence tests, and score criteria, to add/remove features to/from S.

Algorithm 1: Standard forward-backward selection (SFBS)

1: **Input**: Feature Set F and the class variable C
    Output: The set of selected S
2: $S = \emptyset$;
3: //Forward phase: Adding features to S
4: **repeat**
5:    Identify the most informative feature $X \in F$ by selection criterion Φ;
6:    **if** $\Phi(S \cup X) > \Phi(S)$ **then**
7:      $S = S \cup X$ and $F = F \setminus X$;
8:    **end if**
9: **until** no features in F are added to S;
10: //Backward phase: Removing features from S
11: **repeat**
12:    Find the least informative feature $Y \in S$ by selection criterion Φ;
13:    **if** $\Phi(S \setminus Y) \geq \Phi(S)$ **then**
14:      $S = S \setminus Y$;
15:    **end if**
16: **until** no features in S are removed
17: Output S

Algorithm 2: Interleaving forward-backward selection (IFBS)

1: **Input**: Feature Set F and the class variable C
    Output: The set of selected S
2: $S = \emptyset$;
3: **repeat**
4:    //Forward phase: Adding features to S
5:    Identify the most informative feature $X \in F$ by selection criterion Φ;
6:    **if** $\Phi(S \cup X) > \Phi(S)$ **then**
7:      $S = S \cup X$ and $F = F \setminus X$;
8:      //Backward phase: Removing features from S
9:      **repeat**
10:        Find the least informative feature $Y \in S$ by selection criterion Φ;
11:        **if** $\Phi(S \setminus Y) \geq \Phi(S)$ **then**
12:          $S = S \setminus Y$;
13:        **end if**
14:      **until** no features in S are not removed
15:    **end if**
16: **until** no features in F are added to S;
17: Output S

Table 1: Representative constraint-based algorithms using conditional independence tests

Category	Representative algorithm
Simultaneous MB learning (learning PC and spouses simultaneously and do not distinguish PC from spouses)	GSMB (Margaritis and Thrun, 2000)
	IAMB (Tsamardinos and Aliferis, 2003)
	IAMBnPC (Tsamardinos et al., 2003b)
	IAMB-IP (Pocock et al., 2012)
	Fast-IAMB (Yaramakala and Margaritis, 2005)
	Inter-IAMB (Tsamardinos et al., 2003b)
	Inter-IAMBnPC (Tsamardinos et al., 2003b)
	FBED^K (Borboudakis and Tsamardinos, 2019)
	PFBP (Tsamardinos et al., 2019)
Dvide-and-conquer MB learning (learning PC and spouses separately)	MMMB (Tsamardinos et al., 2003a)
	HITON-MB (Aliferis et al., 2003)
	Semi-HITON-MB (Aliferis et al., 2010a)
	PCMB (Pena et al., 2007)
	IPCMB (Fu and Desmarais, 2008)
	MBOR (De Morais and Aussem, 2008)
	STMB (Gao and Ji, 2017a)
	CCMB (Wu et al., 2019)
MB learning with interleaving PC and spouse learning	BAMB (Ling et al., 2019)
	EEMB (Wang et al., 2020)
MB learning with relaxed assumptions (e.g. the faithfulness assumption or causal sufficiency assumption)	KIAMB (Pena et al., 2007)
	TIE* (Statnikov et al., 2013)
	SGAI (Yu et al., 2017)
	LCMB (Liu and Liu, 2016)
	WLCMB (Liu and Liu, 2016)
	M3B (Yu et al., 2018c)
MB learning with special purpose (e.g. multiple data sets, distribution shift, and weak supervision)	MIMB (Yu et al., 2018a)
	MCFS (Yu et al., 2019)
	MIAMB and MKIAMB (Liu and Liu, 2018)
	BASSUM (Cai et al., 2011)
	Semi-IAMB (Sechidis and Brown, 2018)

3. Constraint-based methods

In this section, we will discuss the constraint-based MB learning methods to learn the MB or PC of the class variable, i.e., the methods using conditional independence tests. Constraint-based methods can be categorized into five types: simultaneous MB learning, divide-and-conquer MB learning, MB learning with interleaving PC and spouse learning, MB learning with relaxed assumptions, and MB learning with special purpose. A summary of the representative algorithms of the five types is given in Table 1.

In the following, Section 3.1 presents the basis of constraint-based methods. Section 3.2 gives the brief discussions of the five types of constraint-based methods. Section 3.3 extensively reviews existing constraint-based methods of each type.

3.1 Basis of the constraint-based methods

The constraint-based methods are mainly based on Propositions 2 and 3 below. Proposition 2 illustrates the dependent relations between a node and its parents (or children). It states that if V_i is a parent or a child of V_j in a BN, V_i and V_j are not independent conditioning on any subsets of $V \setminus \{V_i, V_j\}$. For example, in Figure 1, there is an edge between *Smoking* and *Lung cancer*, and thus *Smoking* and *Lung cancer* are not independent given any subsets of the remaining features.

Proposition 2 *(Spirtes et al., 2000) In a BN, if node V_i is a parent (or a child) of V_j, then $\forall S \subseteq V \setminus \{V_i, V_j\}$, $V_i \not\perp V_j | S$.*

Proposition 3 presents the relation between a node and its spouses in a BN. It indicates that if V_i is a spouse of V_k and V_j is their common child, there exists a subset $S \subseteq V \setminus \{V_i, V_j, V_k\}$ such that V_i and V_k are independent given S but they are dependent given $S \cup V_j$. For instance, *Allergy* is the spouse of *Lung cancer* in Figure 1. *Allergy* and *Lung cancer* are independent (S is an empty set), but they are dependent conditioning on their common child *Coughing*. Proposition 3 shows that *Allergy* (spouse) and *Coughing* together carry more predictive information about *Lung cancer* than *Coughing* only. Proposition 3 also states that spouses of V_k consist of all parents of the children of V_k (excluding V_k).

Proposition 3 *(Spirtes et al., 2000) In a BN, assuming that V_i is adjacent to V_j, V_j is adjacent to V_k, and V_i is not adjacent to V_k (e.g., $V_i \to V_j \leftarrow V_k$), if $\exists S \subseteq V \setminus \{V_i, V_j, V_k\}$ such that $V_i \perp\!\!\!\perp V_k | S$ and $V_i \not\perp V_k | \{S, V_j\}$ hold, V_i is a spouse of V_k.*

Existing constraint-based methods adopt either the SFBS strategy or the IFBS strategy, and they employ the statistical independence tests as the selection criteria, denoted as Φ, to add/remove features to/from S as follows.

Given the class variable C, in SFBS (or IFBS), at each iteration, let S be a set of features currently selected, if $X \in V \setminus \{C \cup S\}$ and C are conditionally independent conditioning on S (or a subset $S' \subset S$), that is, $X \perp\!\!\!\perp C | S$ (or $X \perp\!\!\!\perp C | S'$), X does not provide any predictive information to C conditioning on S (i.e., $\Phi(S \cup X) \leq \Phi(S)$ in SFBS or IFBS). In this case, X is not to be added to S at the forward phase or removed from S at the backward phase.

Five types of conditional independence tests are used by current constraint-based methods, λ^2 test, G^2 test, mutual information for discrete features (McDonald, 2009), Fishers Z tests for continuous features with linear relations with additive Gaussian errors (Peña, 2008), and kernel-based tests for continuous features with nonlinearity and non-Gaussian noise (Zhang et al., 2012).

3.2 Overview of constraint-based methods

In the section, we will give a brief overview of the five types of constraint-based methods. The detailed review of the representative algorithms of each type will be presented in Section 3.3.

1. Simultaneous MB learning. Given the class variable C, a simultaneous MB learning algorithm aims to find parents, children, and spouses of C simultaneously, and do not distinguish PC (parents and children) of C from its spouses during the MB learning. As shown
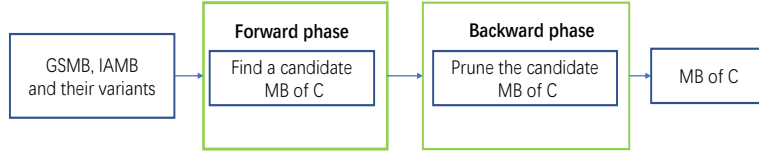
Figure 2: Simultaneous MB learning

in Figure 2, the simultaneous MB learning approach adopts a forward-backward strategy to greedily learn a MB of C by conditioning on the entire candidate MB of C ($CMB(C)$) currently selected at each iteration. The representative simultaneous MB learning algorithms include GSMB (Margaritis and Thrun, 2000), IAMB (Tsamardinos and Aliferis, 2003), IAMBnPC (Tsamardinos et al., 2003b), Fast-IAMB (Yaramakala and Margaritis, 2005), Inter-IAMB (Tsamardinos et al., 2003b), Inter-IAMBnPC (Tsamardinos et al., 2003b), IAMB-IP (Pocock et al., 2012), FBED^K (Borboudakis and Tsamardinos, 2019), and PFBP (Tsamardinos et al., 2019). The GSMB algorithm was the first algorithm for learning a MB of the class variable without learning an entire Bayesian network. IAMB and its variants are all the improved versions of GSMB. Inter-IAMB interleaves the forward phase and the backward phase of IAMB. Both FBED^K and PFBP are the state-of-the-art variants of IAMB.

Due to the use of the entire $CMB(C)$ currently selected for conditional independence tests at each computation, existing simultaneous MB learning algorithms reduce the number of independence tests, but require more data samples for each test since the number of data samples required is exponential to the size of the conditioning set. Thus the simultaneous MB learning algorithms are time efficient but not data efficient. When the sample size of a data set is not big enough, these algorithms cannot find the MB accurately. The quality of learnt MBs of these algorithms degrades greatly in practical settings due to the limited number of samples. They are expected to perform the best in problems where $MB(C)$ is small.

2. Divide-and-conquer MB learning. The divide-and-conquer MB learning approach aims to reduce the data requirements of the simultaneous MB learning approach. This approach breaks the problem of learning $MB(C)$ into two subproblems: first, learning parents and children of C (i.e., $PC(C)$), and second, learning the spouses of C (i.e., $SP(C)$). As for learning $PC(C)$, the divide-and-conquer approach does not use the entire $PC(C)$ as the conditioning set for conditional independence tests when determining whether feature X is a candidate member of $PC(C)$. Instead, it makes use of the subsets of $PC(C)$ which is much smaller than $CMB(C)$ used by the simultaneous MB learning approach when making decisions. Therefore, the divide-and-conquer MB learning approach needs significantly smaller number of samples than the simultaneous MB learning approach. For instance, to determine whether feature X is a candidate member of $PC(C)$, the divide-and-conquer approach explores possible subsets of $PC(C)$. If there exists a subset $S \subseteq PC(C)$ such that X and C are conditional independent given S, this subset exploring process will terminate and X will be discarded and never considered again. However, for the simultaneous MB learning approach, such as GSMB, IAMB, and their variants, the discarded features will be reconsidered many times (for identifying spouses).
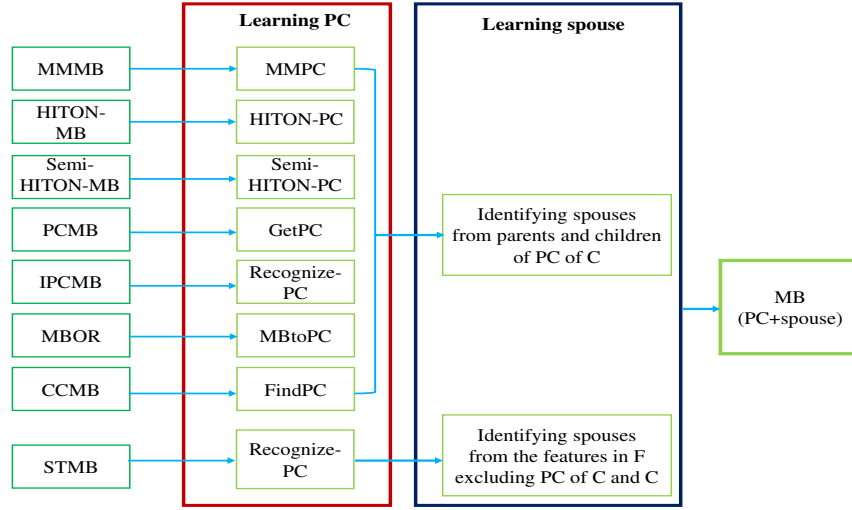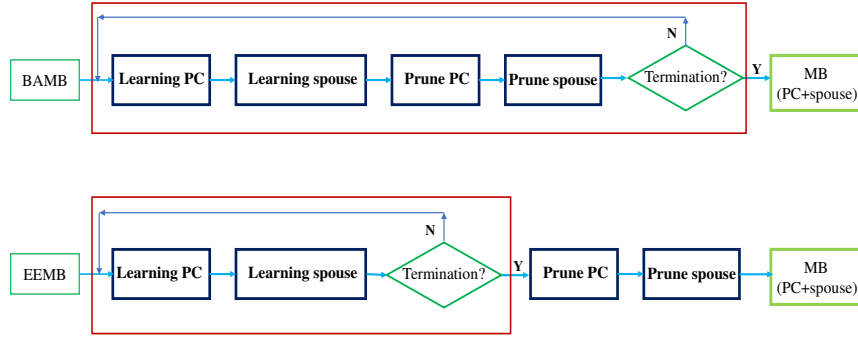
Figure 3: Divide-and-conquer MB learning

Figure 4: MB learning with interleaving PC and spouse learning

The representative divide-and-conquer algorithms include MMMB (Tsamardinos et al., 2003a), HITON-MB (Aliferis et al., 2003), semi-HITON-MB (Aliferis et al., 2003), PCMB (Peña et al., 2005), IPCMB (Fu and Desmarais, 2008), MBOR (De Morais and Aussem, 2008), STMB (Gao and Ji, 2017a), and CCMB (Wu et al., 2019). The main differences between those algorithms lie in the strategies of identifying parents and children of C and the strategies of finding spouses of C, as shown in Figure 3. This figure also presents the general steps of existing divide-and-conquer MB learning algorithms (i.e., learning PC and identifying spouses separately), and the PC learning used by the eight representative MB algorithms respectively.

The divide-and-conquer MB learning methods are data efficient but not time efficient. Although they mitigate the problem of the large sample requirement, existing divide-and-conquer MB learning algorithms will be computationally expensive when the size of currently selected features becomes large.

3. MB learning with interleaving PC and spouse learning. This approach is an extension of the divide-and-conquer approach. Instead of learning PC and identifying spouses separately, this approach implements the PC leaning phase and the spouse iden-

tifying phase alternatively. Specifically, once a candidate member of PC of C is added to the candidate $PC(C)$ at the PC learning phase, this approach triggers the spouse learning phase immediately. The representative algorithms include BAMB (Ling et al., 2019) and EEMB (Wang et al., 2020). The difference between BAMB and EEMB is shown in Figure 4, where we can see that BAMB learns the candidate PC and spouse sets of C and removes false positives from the two candidate sets in one go, while EEMB breaks BAMB into two independent subroutines: learning and pruning.

By interleaving PC and spouse learning, BAMB and EEMB attempt to keep both candidate PC and spouse sets as small as possible for achieving the trade-off between data efficiency and time efficiency. However, due to false PC inclusions, many false spouses may enter the candidate spouse set, leading to a large size of the candidate spouse set, which will degrade the performance of BAMB and EEMB.

4. MB learning with relaxed assumptions. The above algorithms are designed to learn the MB of the class variable under the faithfulness and causal sufficiency assumptions. In fact, both assumptions are often violated in practice.

The theoretical result has stated that if a data distribution satisfies the faithfulness assumption, the MB of the class variable is unique (Pearl, 2014). When the faithfulness assumption is violated, the MB of learnt from the data may not be unique (Pena et al., 2007; Statnikov et al., 2013). To deal with the violation of the faithfulness assumption, some research work has been done for identifying multiple MBs without the assumption, such as KIAMB (Pena et al., 2007), TIE* (Statnikov et al., 2013), SGAI (Yu et al., 2017), LCMB (Liu and Liu, 2016), and WLCMB (Liu and Liu, 2016). KIAMB was the first attempt to learn multiple MBs, but it needs to run multiple times and cannot guarantee finding all possible MBs of the class variable. TIE* can find all MBs of the class variable in a data set, but it is computationally expensive, especially when the number of MBs is large. SGAI may be more efficient than TIE* but it is not guaranteed to find all possible MBs of the class variable. WLCMB is motivated by KIAMB and thus it still suffers from the drawbacks of KIAMB. Thus it is still a challenging problem to tackle multiple MB learning when the faithfulness assumption is violated.

When the causal sufficiency assumption does not hold in a data set, if we still use a MB learning algorithm that assumes causal sufficiency, the MB learnt from the data set may not properly indicate the true causal relations. Yu et al. (Yu et al., 2018c) proposed the M3B algorithm to deal with the situation when the assumption of causal sufficiency is violated. However, M3B uses a backward strategy to learn PC of the class variable and needs to perform an exhaustive search over the currently selected PC set. Thus it also suffers from time efficiency and incorrect test problems of the constraint-based MB learning in general.

5. MB learning with special purpose. Beyond the algorithms discussed above, several MB learning algorithms have been proposed for special purposes, including the MIMB algorithm for identifying a MB of a class variable from multiple data sets (Yu et al., 2018a), the MCFS algorithm for stable prediction with distribution shift (Yu et al., 2019), the MI-AMB and MKIAMB algorithms for learning a MB of multiple class variables (Liu and Liu, 2018), and the BASSUM and Semi-IAMB algorithms for MB learning with weak supervision (Cai et al., 2011; Sechidis and Brown, 2018). These studies have shown that causal properties of features can facilitate semi-supervised learning and feature selection with distribution shifts. Moreover the intersection of machine learning and causal discovery has

---

Algorithm 3: The instantiation of SFBS for simultaneous MB learning

---

1: **Input**: Feature Set F and the class variable C
   Output: $CMB(C)$
2: $CMB(C) = \emptyset$;
3: //Forward phase: Adding candidate MB (relevant) features to $CMB(C)$
4: **repeat**
5:    Select a feature $X \in F$;
6:    **if** $X \not\perp\!\!\!\perp C|CMB(C)$ **then**
7:       $CMB(C) = CMB(C) \cup X$ and $F = F \setminus X$;
8:    **end if**
9: **until** no features in F are added to $CMB(C)$;
10: //Backward phase: Removing false positives from $CMB(C)$
11: **repeat**
12:    Select a feature $Y \in CMB(C)$;
13:    **if** $Y \perp\!\!\!\perp C|CMB(C) \setminus Y$ **then**
14:       $CMB(C) = CMB(C) \setminus Y$;
15:    **end if**
16: **until** no features in $CMB(C)$ are removed
17: Output $CMB(C)$

---

attracted increasing attention in areas beyond feature selection. For example, causal knowledge has inspired efficient transfer learning and domain adaptation methods for accurate prediction across different domains (Rojas-Carulla et al., 2018; Magliacane et al., 2018). It is a promising research area to link machine learning research with causality to develop explainable and robust machine learning methods and solutions to causal discovery for data analytics.

3.3 Detailed review of constraint-based methods

3.3.1 METHODS OF SIMULTANEOUS MB LEARNING

In this subsection, we firstly introduce the methods using SFBS, including GSMB, IAMB and the two extensions of IAMB, which are IAMBnPC and IAMP-IP. Then we introduce the methods employing IFBS, which are Fast-IAMB, Inter-IAMB and Inter-IAMBnPC. Since FBEDk and PFBP are the state-of-the-art simultaneous MB learning algorithms and the extensions of IAMB, they will be introduced at the end.

GSMB. The Growing-Shrinking MB (GSMB) learning algorithm (Margaritis and Thrun, 2000; Margaritis, 2009) instantiates the SFBS framework for simultaneous MB learning, as shown in Algorithm 3. Let $CMB(C)$ be the candidate MB of C currently selected, in the forward (growing) phase (Steps 5 to 10 in Algorithm 3), at each iteration, if $\exists X \in F \setminus CMB(C)$ such that $X \not\perp\!\!\!\perp C|CMB(C)$ holds, GSMB adds X to $CMB(C)$, until no features within $F \setminus CMB(C)$ are added to $CMB(C)$. In the backward (shrinking) phase (Steps 12 to 17 in Algorithm 3), GSMB sequentially removes from $CMB(C)$ the false positive $Y \in CMB(C)$ satisfying $Y \perp\!\!\!\perp C|CMB(C) \setminus Y$. However, at Step 5 of the forward phase, GSMB uses a static heuristic that at each time GSMB randomly selects a feature $X \in F$ satisfying

$X \not\perp C|CMB(C)$ and adds it to $CMB(C)$. The static heuristic may make many false positives enter $CMB(C)$ in the forward phase, leading to the growing of the size of $CMB(C)$. Given a fixed size of data samples, the larger size of $CMB(C)$, the more unreliable the independence tests. Thus the heuristic makes GSMB ineffective in coping with a data set of small sample size but high dimensionality.

IAMB, IAMBnPC, and IAMB-IP. To tackle the problem with GSMB, the incremental association Markov boundary (IAMB) algorithm (Tsamardinos and Aliferis, 2003) uses a dynamic heuristic at Step 5 in Algorithm 3 of the forward phase. At each iteration, IAMB adds to $CMB(C)$ the feature $X \in F \setminus CMB(C)$ with the highest association with C conditioning on the current $CMB(C)$ if $X \not\perp C|CMB(C)$ holds. This dynamic heuristic makes the features that belong to $MB(C)$ enter $CMB(C)$ as early as possible and reduces as much as possible the chance of false positives to enter $CMB(C)$ during the forward phase. Accordingly, IAMB performs better (with lower time complexity and lower data sample requirement) than GSMB since fewer false positives will be added to $CMB(C)$ in the forward phase. However, the number of required data samples of IAMB is still exponential with the size of $CMB(C)$ since the size of $CMB(C)$ may become large in the forward phase. To mitigate this problem, several variants of IAMB were proposed, such as IAMBnPC (Tsamardinos et al., 2003b), Inter-IAMB (Tsamardinos et al., 2003b), inter-IAMBnPC (Tsamardinos et al., 2003b), and Fast-IAMB (Yaramakala and Margaritis, 2005). Compared to IAMB, IAMBnPC only substitutes the backward phase (Steps 11 to 16 in Algorithm 3) as implemented in IAMB with the PC algorithm (Spirtes et al., 2000). IAMBnPC is more data efficient (with lower data sample requirement) than IAMB since the PC algorithm runs only on the subsets of the current $CMB(C)$ instead of conditioning on the entire $CMB(C)$. To leverage prior knowledge, the IAMB-IP (IAMB-Informative Prior) algorithm was proposed in (Pocock et al., 2012). It can incorporate domain knowledge priors and structure sparsity priors to improve the performance of MB learning when the data set is of small sample size but high dimensionality.

Inter-IAMB and Inter-IAMBnPC. These two algorithms adopt the IFBS framework, which is the key difference between them and IAMB. Algorithm 4 shows how they instantiate IFBS for simultaneous MB learning. The goal of the interleaving is to keep the size of $CMB(C)$ as small as possible during all steps of the algorithms' execution. Comparing to Inter-IAMB, the Inter-IAMBnPC algorithm substitutes the backward phase as implemented in inter-IAMB with the PC algorithm (Steps 9 to 14 in Algorithm 4).

Fast-IAMB. Similar to Inter-IAMB and Inter-IAMBnPC, Fast-IAMB instantiates IFBS as shown in Algorithm 4. However, different from IAMB and its other variants discussed above, Fast-IAMB adopts an aggressively greedy strategy in the forward phase to make it more efficient. Specifically, at Steps 5 to 6 in Algorithm 4, Fast-IAMB does not add one feature to $CMB(C)$ then immediately triggers the backward phase. Instead, Fast-IAMB greedily adds as many features conditionally dependent on C given the current $CMB(C)$ as possible in the forward phase until a conditional independence test is not reliable (i.e., we do not have enough data for conducting the test). When a test is not reliable in the forward phase, the backward phase is triggered. A reliable independence test for C and $X \in F \setminus CMB(C)$ given $CMB(C)$ should satisfy the rule that the average number of instances per cell of the contingency table of $X \cup C \cup CMB(C)$ must be at least k, i.e., $N/\{r_X * r_C * r_{CMB(C)}\} \geq k$ where the minimum value of k is set to 5 for reliable tests as

---
Algorithm 4: The instantiation of IFBS for simultaneous MB Learning
---
1: **Input**: Feature Set F and the class variable C
    Output: $CMB(C)$
2: $CMB(C) = \emptyset$;
3: **repeat**
4:    //Forward phase: Adding candidate MB (relevant) features to $CMB(C)$
5:    Select a feature $X \in F$ with the highest association with C;
6:    **if** $X \not\perp\!\!\!\perp C | CMB(C)$ **then**
7:      $CMB(C) = CMB(C) \cup X$ and $F = F \setminus X$;
8:      //Backward phase: Removing false positives from $CMB(C)$
9:      **repeat**
10:        Select a feature $Y \in CMB(C)$;
11:        **if** $Y \perp\!\!\!\perp C | CMB(C) \setminus Y$ **then**
12:          $CMB(C) = CMB(C) \setminus Y$;
13:        **end if**
14:      **until** no features in $CMB(C)$ are removed
15:    **end if**
16: **until** no features in F are added to $CMB(C)$;
17: Output $CMB(C)$
---

suggested by Agresti (Agresti and Kateri, 2011), N is the total number of data samples, and r_C denotes the number of discrete values that C takes. By the rule, at Steps 5 to 6 in Algorithm 4, Fast-IAMB will not perform a test when it is not reliable. This checking not only speeds up Fast-IAMB, but also reduces the risk of unreliable independence tests.

FBED^K. Borboudakis and Tsamardinos (Borboudakis and Tsamardinos, 2019) generalized the IAMB framework for feature selection and proposed the FBED^K (Forward-Backward selection with Early Dropping) algorithm to speed up IAMB. For IAMB, in the forward phase, at each iteration, it should reconsider all remaining features (including all discarded features at each iteration) to find the next best candidate. To tackle the issues, $FBED^K$ adopts an early dropping strategy in the forward phase. The main idea is that at each forward iteration, $FBED^K$ removes the features that are conditionally independent of C given the current $CMB(C)$ from the remaining features in F instead of keeping them in F. This leads to quickly reduce the number of candidate features in F, while keeping relevant features in it. A run of the forward phase with the early dropping terminates until F is empty. Then the forward phase is allowed to run up to K additional times to reconsider features dropped previously until no features can be dropped. Finally, the backward phase is applied to $CMB(C)$ obtained at the forward phase, and this is the same as the backward phase of IAMB. $FBED^K$ significantly improves computational efficiency, while retaining competitive accuracy.

PFBP. Motivated by FBED^K, the Parallel Forward-Backward with Pruning (PFBP) algorithm was proposed for improving IAMB to tackle big data with high dimensionality (Tsamardinos et al., 2019). PFBP enables computations to be performed in a parallel way by partitioning data both in terms of rows (samples) as well as columns (features) and using meta-analysis techniques to combine results of local computations.

Algorithm 5: The instantiation of SFBS for PC Learning

  1: **Input**: Feature set F and the class variable C
     Output: $CPC(C)$
  2: $CPC(C) = \emptyset$;
  3: // Filtering out irrelevant features by Proposition 2 (if $X \perp\!\!\!\perp C$ holds, $X \notin PC(C)$)
  4: $R = F \setminus S'$ $(\forall X \in S', \ X \perp\!\!\!\perp C | \emptyset)$;
  5: //Forward phase: Adding candidate PC (or relevant) features to $CPC(C)$
  6: **repeat**
  7:    Select the best feature $X \in R$ with a greedy strategy;
  8:    $CPC(C) = CPC(C) \cup \{X\}$; $R = R \setminus X$;
  9: **until** no features in R are added to $CPC(C)$;
10: //Backward phase: Removing false positives from $CPC(C)$;
11: **repeat**
12:    Consider each feature $Y \in CPC(C)$;
13:    **if** $\exists S \subseteq CPC(C) \setminus Y$ s.t. $Y \perp\!\!\!\perp C | S$ **then**
14:      $CPC(C) = CPC(C) \setminus Y$;
15:    **end if**
16: **until** no features in $CPC(C)$ are removed;
17: Output $CPC(C)$

In addition to the early dropping strategy proposed in (Borboudakis and Tsamardinos, 2019), PFBP also proposed two new heuristics of early stopping with the consideration of features within the same iteration and early returning the current best feature for addition or removal. It has been shown that PFBP can scale to millions of features and millions of training samples, and achieves a super-linear speedup with increasing sample size and linear scalability with respect to the number of features and processing cores.

3.3.2 Methods of divide-and-conquer MB learning

In this subsection, we will discuss eight representative divide-and-conquer algorithms, i.e., MMMB (Tsamardinos et al., 2003a), HITON-MB (Aliferis et al., 2003), semi-HITON-MB (Aliferis et al., 2003), PCMB (Peña et al., 2005), IPCMB (Fu and Desmarais, 2008), MBOR (De Morais and Aussem, 2008), STMB (Gao and Ji, 2017a), and CCMB (Wu et al., 2019). As illustrated in Figure 3, given the class variable C, how to learn its parents and children and identify its spouses is the main difference between those algorithms. Generally speaking, there are three strategies for learning parents and children of C: SFBS, IFBS and the backward framework. SFBS and IFBS for PC learning are very similar to those for MB learning. The instantiations of SFBS and IFBS for PC learning are present in Algorithms 5 and 6 respectively, while the backward framework for PC learning is shown in Algorithm 7.

    MMMB. The MMMB (Max-Min MB) algorithm (Tsamardinos et al., 2003a) first employs the MMPC (Max-Min Parents and Children) algorithm to find a candidate set of parents and children of C. MMPC (Tsamardinos et al., 2003a) utilizes the SFBS framework to search for candidate parents and children of C first, called $CPC(C)$, then prunes $CPC(C)$ at the backward phase, as shown in Algorithm 5. The novelty of MMPC lies

Algorithm 6: The instantiation of IFBS for PC Leaning

1: **Input**: Feature set F and the class variable C
   Output: $CPC(C)$
2: $CPC(C) = \emptyset$;
3: $R = F \setminus S'$ ($\forall X \in S', \ X \perp\!\!\!\perp C | \emptyset$);
4: **repeat**
5:   //Forward phase: Adding candidate PC (relevant) features to $CPC(C)$
6:   Select the best feature $X \in R$ with a greedy strategy;
7:   $CPC(C) = CPC(C) \cup \{X\}$; $R = R \setminus X$;
8:   //Backward phase: Removing false positives from $CPC(C)$
9:   **repeat**
10:    Consider each feature $Y \in CPC(C)$
11:    **if** $\exists S \subseteq CPC(C) \setminus Y$ s.t. $Y \perp\!\!\!\perp C | S$ **then**
12:      $CPC(C) = CPC(C) \setminus Y$;
13:    **end if**
14:   **until** no features in $CPC(C)$ are removed;
15: **until** no features in R are added to $CPC(C)$;
16: Output $CPC(C)$

Algorithm 7: The Backward Framework for PC Learning

1: **Input**: Feature Set F and the class variable C
   Output: $CPC(C)$
2: $CPC(C) = \{F\}$;
3: i=0;
4: **repeat**
5:   **for** each feature X in $CPC(C)$ **do**
6:    **if** $\exists S \subseteq CPC(C) \setminus X$ and $|S| = i$ such that $X \perp\!\!\!\perp C | S$ **then**
7:      $CPC(C) = CPC(C) \setminus X$;
8:    **end if**
9:   **end for**
10: i=i+1;
11: **until** $i > |CPC(C)|$;
12: Output $CPC(C)$

---

the fact that at Step 7 of the forward phase in Algorithm 5, MMPC proposes a Max-Min Parents and Children (MMPC) greedy search strategy to identify the best feature from $F \setminus CPC(C)$ at each iteration.

Specifically, in the forward phase, at each iteration, given the current $CPC(C)$ (initially $CPC(C)$ is empty), for each feature X in the remaining candidate features (i.e., $X \in F \setminus CPC(C)$), MMPC first calculates the associations of X and C conditioning on all possible subsets of $CPC(C)$ respectively, and chooses the minimum association as the association of X and C. Then MMPC chooses the next feature to be included in $CPC(C)$ as the one that exhibits the maximum association among the features in $F \setminus CPC(C)$ and is

dependent on C, while the features independent of C are discarded and never considered as candidate PC again. The forward phase terminates until each feature in $F \setminus CPC(C)$ and C are independent given any subsets of $CPC(C)$. At the backward phase, MMPC examines whether each feature Y in $CPC(C)$ obtained in the forward phase is independent of C conditioning on all possible subsets of $CPC(C) \setminus Y$. If so, Y is removed from $CPC(C)$; otherwise it is retained.

Now we discuss how to learn spouses of C after $CPC(C)$ is obtained. The spouses of C are the parents of the children of C excluding $C \cup PC(C)$, i.e., $SP(C) = \bigcup_{Y' \in ch(C)} pa(Y') \setminus (C \cup PC(C))$. However, MMPC cannot distinguish parents from children of C during the procedure of identifying $PC(C)$. Thus, MMMB considers the union of parents and children of the features in $CPC(C)$ excluding $C \cup CPC(C)$ as the the candidate spouses of C, i.e., $SPC(C) = \bigcup_{Y' \in CPC(C)} PC(Y') \setminus (C \cup CPC(C))$. Then by Proposition 3, for each feature Y in the $SPC(C)$ set and each feature X in $CPC(C)$, if there exists a subset $S \subseteq F \setminus \{C, X, Y\}$ (S was identified and stored in the MMPC subroutine) such that both $C \perp\!\!\!\perp Y | S$ and $C \not\!\perp\!\!\!\perp Y | X \cup S$ hold, MMMB considers Y as a spouse of C.

HITON-MB and Semi-HITON-MB. HITON-MB uses the HITON-PC algorithm to discover $PC(C)$ (Aliferis et al., 2003). Different from MMPC, HITON-PC employs the IFBS framework as presented in Algorithm 6. HITON-PC interleaves the forward phase and the backward phase to make PC learning and false positive removal alternatively. In addition, at Step 6 in Algorithm 6, HITON-PC adopts a simpler search strategy than MMPC for learning candidate parents and children of C. Specifically, at Step 6, at each iteration, HITON-PC removes a feature, called X, with the highest association with C conditioning on an empty set from the candidate feature set R and adds it to $CPC(C)$, then triggers the backward phase for removing false positives from the current $CPC(C)$ due to the X's inclusion.

For identifying the spouses of C, in the original version of the HITON-MB algorithm (Aliferis et al., 2003), the idea of HITON-MB is the same as that of MMMB.

However, Pena et al. (Peña et al., 2005; Pena et al., 2007) pointed out that MMMB and HITON-MB cannot return the correct MB even under the faithfulness assumption. They found that (1) both MMPC and HITON-PC may return a superset of the true PC of C, and (2) the spouse discovery procedures of both MMMB and HITON-MB cannot find the correct spouses of C. Tsamardinos et al. (Tsamardinos et al., 2006) also identified the flaw of MMPC in point (1) above independently and proposed a corrected MMPC using the symmetric relation between parents and children in a BN (i.e., symmetric check). That is, if X is a parent or a child of C, C should be a child or a parent of X. Following this, Aliferis et al. (Aliferis et al., 2010a) proposed a general local learning (GLL) framework and corrected the two flaws discussed above. In addition, in (Aliferis et al., 2010a), a new Semi-interleaved HITON-PC (Semi-HITON-PC for brevity) algorithm was proposed to speed up HITON-PC. The difference between Semi-HITON-PC and HITON-PC is that at Step 10 in Algorithm 6, Semi-HITON-PC only considers the elimination of the newly added feature at Step 7 before the candidate feature set R becomes empty and a full feature elimination in $CPC(C)$ will be performed after R is empty. Employing Semi-HITON-PC, the semi-HITON-MB algorithm has been proposed accordingly (Aliferis et al., 2010a).

PCMB. The parents and children based MB (PCMB) algorithm (Peña et al., 2005; Pena et al., 2007) was the first correct divide-and-conquer MB learning algorithm. PCMB

Algorithm 8: The Framework of Spouse Learning

1: **Input**: C, $CPC(C)$, and $Sepset(X)$ for each feature X in F
   Output: Spouses of C ($SP(C)$)
2: $SP(C) = \emptyset$;
3: **for** each feature X in $CPC(C)$ **do**
4:     Find $CPC(X)$ using a PC learning algorithm (e.g., MMPC)
5:     **for** each feature Y in $CPC(X) \setminus \{C \cup CPC(C)\}$ **do**
6:         **if** $Y \not\perp\!\!\!\perp C | X \cup Sepset(Y)$ **then**
7:             $SP(C) = SP(C) \cup Y$;
8:         **end if**
9:     **end for**
10: **end for**
11: Output $SP(C)$

---

uses the two subroutines, called GetPCD and GetPC, to identify $PC(C)$. The GetPCD subroutine is to find $CPC(C)$, and the GetPC subroutine removes false positives in $CPC(C)$ using the symmetric check, i.e., for each feature X in $CPC(C)$, if the set of parents and children of X does not include C, X will be removed from $CPC(C)$.

GetPCD adopts the similar idea of MMPC, but the two algorithms have two differences. First, GetPCD adopts the IFBS framework and interleaves the forward and backward phases of MMPC. Second, in the backward phase, for each feature X in the current $CPC(C)$, GetPCD calculates the associations of X and C conditioned on all possible subsets of $CPC(C)$ and chooses the minimum association as the association of X and C. If X and C are assessed to be independent given the minimum association, X will be removed from $CPC(C)$. Pena et al. (Peña et al., 2005; Pena et al., 2007) stated that $CPC(C)$ learnt by GetPCD may be a superset of the true parents and children of C since some non-child descendants of C are added to $CPC(C)$. Thus GetPC was proposed to remove these non-child descendants using the symmetry check.

As for finding the spouses of C, for each feature $X \in CPC(C)$ obtained by GetPC, first, PCMB uses GetPC to find the parents and children of X (i.e., $PC(X)$), then for each feature Y in $PC(X)$, if there exists a subset S within $F \setminus \{C, X, Y\}$ (S was identified and stored in the procedure of GetPCD) such that both $C \perp\!\!\!\perp Y | S$ and $C \not\perp\!\!\!\perp Y | S \cup \{X\}$ hold, Y is a spouse of C with regard to X. The above procedure of finding the spouses of C is summarized in Algorithm 8 (Pena et al., 2007; Aliferis et al., 2010a). The study in (Pena et al., 2007; Aliferis et al., 2010a) has shown that if the input $CPC(C)$ and the PC learning algorithm used by Algorithm 8 are correct, Algorithm 8 is complete and sound (Aliferis et al., 2010a).

IPCMB. The Iterative Parent-Child based search of MB (IPCMB) algorithm (Fu and Desmarais, 2008) is quite similar to PCMB. The key difference between them is that IPCMB employs the RecognizePC algorithm (Li et al., 2015) to find the PC set of C. RecognizePC uses a backward strategy as shown in Algorithm 7. Initially, RecognizePC assumes that all features in F are the candidate PC of C, that is, $CPC(C) = F$. To remove false positives from $CPC(C)$, RecognizePC uses conditional independence tests to check each feature in $CPC(C)$ level by level of the cardinality of the conditioning sets, starting with an empty set.

For spouse discovery, IPCMB adopts the framework in Algorithm 8. Compared to the divide-and-conquer MB learning algorithms discussed above, an additional improvement is that IPCMB embeds the symmetry check before Step 5 in Algorithm 8. That is, for each feature X in $CPC(C)$, if $CPC(X)$ obtained at Step 4 does not include C, IPCMB does not implement Steps 6 to 8 and moves to the next feature in $CPC(C)$.

MBOR. The larger the size of a conditioning set in a conditional independence test, the less reliable is the independence test. the MB learning algorithms discussed above, such as IAMB (and its variants), MMMB, HITON-MB, and PCMB, may miss true positives due to the unreliability of the conditional independence tests if the conditioning set is large. In order to increase the data-efficiency and the robustness of MB learning, MBOR (Markov Boundary search using the OR condition) (De Morais and Aussem, 2008) was designed to keep the sizes of conditional sets as small as possible during the search. MBOR consists of the following three steps. At Step 1, MBOR discovers a superset of the parents and children of C ($PCS(C)$) and a superset of the spouses of C ($SPS(C)$) by severely restricting the size of a conditioning set S in the tests to $|S| \leq 1$ and $|S| \leq 2$ respectively. Thus this reduces the risk of missing features that are weakly associated to C and enhances the reliability of the independence tests. Then at Step 2, MBOR first uses the MBtoPC algorithm (De Morais and Aussem, 2008) to find $CPC(C)$, then learn parents and children of each feature in $PCS(C) \setminus CPC(C)$, and finally applies the OR condition to retrieve a parent or a child of C (called X) that $X \notin CPC(C)$, but $C \in PC(X)$. Finally, at Step 3, MBOR identifies the spouses of C using the framework in Algorithm 8.

The first difference between MBOR and the existing MB algorithms is that MBOR applies the "OR condition" to consider two features X and Y as neighbors if $Y \in PC(X)$ OR $X \in PC(Y)$. In contrast, MMMB, HITON-MB, and PCMB employ the "AND condition", which means that two features X and Y are considered as neighbors if $Y \in PC(X)$ AND $X \in PC(Y)$. The OR condition is less strict than the AND condition and makes it easier for true positives to enter the MB. The second difference is that MBOR finds a superset of the spouses of C from $F \setminus PC(C)$ at Step 1 instead of the union of parents and children of each feature in $PC(C)$. At Step 2, since MBtoPC employs the simultaneous MB discovery approach to find $PC(C)$, it still suffers from the problem of data inefficiency.

STMB. For the divide-and-conquer approach, in the spouse discovery step, identifying parents and children of each feature in $CPC(C)$ is the most computationally expensive due to the exhaustive search for conditioning sets. To mitigate the computational efficiency problem of identifying spouses, different from the algorithms described above, the simultaneous MB (STMB) algorithm (Gao and Ji, 2017a) presents two new strategies. First, STMB (Gao and Ji, 2017a) identifies the spouses of C from $F \setminus CPC(C)$ instead of the union of parents and children of each feature in $CPC(C)$. Second, STMB removes false positives from $CPC(C)$ using the candidate spouses selected currently instead of using the symmetric check. These two strategies may make STMB more efficient than MMMB, HITON-MB, and IPCMB in the spouse discovery phase, since it will be computationally expensive or prohibit to learn the union of parents and children of each feature in $CPC(C)$, especially when the size of $CPC(C)$ is large.

Specifically, STMB includes the following four steps. At Step 1, STMB finds $CPC(C)$ by using the RecognizePC algorithm. At Step 2, for each feature $X \in CPC(C)$, STMB identifies the spouses of C with regard to X from $F \setminus CPC(C)$ and removes false positives

from $CPC(C)$ using the candidate spouses selected at this step 2 alternatively. At Step 3, STMB removes false positives in $SP(C)$ by using the $CPC(C) \cup SP(C)$ obtained at Step 2. At Step 4, STMB removes false positives from $CPC(C)$ by using $SP(C)$ obtained at Step 3. After the four steps, STMB obtains the MB of C, i.e., $CPC(C) \cup SP(C)$.

Although STMB improves the computational efficiency of identifying spouses, STMB suffers from the problem of data inefficiency at Steps 3 and 4, since at the two steps, STMB uses an entire set as a conditional set instead of a subset exhaustive search.

CCMB. The existing MB learning algorithms mainly focus on how to remove the false positives during the MB search process and then make the false positive rate as low as possible. However, they rarely consider the true positives discarded due to incorrect conditional independence tests, leading to a high false negative rate, especially in the presence of insufficient or noise data samples.

To tackle this issue, Wu et al. (Wu et al., 2019) presented a new concept of *PCMasking* to describe a type of incorrect conditional independence tests in the MB learning process and theoretically analyzed the mechanism behind this type of tests. In the work, *PC-Masking* denotes that the class variable and its children may be independent of each other conditioning on its parents and vice versa due to incorrect independence tests. Based on the theoretical analysis, the cross-check and complement MB (CCMB) learning algorithm was proposed to repair this type of incorrect CI independence tests for accurate MB learning. Specifically, CCMB first learns the PC set of C using a subroutine called FindPC. FindPC is an improved version of the GetPCD algorithm and aims to effectively identify all possible true parents and children of C except for the PC features discarded by FindPC due to the *PCMasking* phenomenon. Then CCMB recovers the discarded PC features using the OR rule based on FindPC. The spouse learning phase of CCMB is the same as that of PCMB. The drawback of CCMB is that although it significantly reduces the false negative rate, CCMB achieves a little higher false positive rate than the divide-and-conquer algorithms discussed above due to the OR rule.

3.3.3 Methods of MB learning with interleaving PC and spouse learning

Different from the algorithms described above that learn PC and spouses separately, BAMB (Ling et al., 2019) and EEMB (Wang et al., 2020) implement the PC learning phase and the spouse identifying phase alternatively for the trade-off between data efficiency and time efficiency.

BAMB. The balanced MB learning (BAMB) algorithm (Ling et al., 2019) does not separate PC learning and spouse identifying into two independent phases. It finds the candidate PC and spouse set of C and removes false positives from the candidate set in one go. Specifically, using the IFBS framework, BAMB integrates PC learning and spouse identifying into one procedure. At each iteration, once a new feature is added to the current $CPC(C)$, BAMB is triggered to find the spouses of C ($SP(C)$) with regard to this feature. Then BAMB first uses the found $SP(C)$ to remove false positives from $CPC(C)$, then employs the updated $CPC(C)$ to prune $SP(C)$ in turn. In this way, during the MB search BAMB can keep both $CPC(C)$ and $SP(C)$ as small as possible for achieving a trade-off between data efficiency and time efficiency. However, in the PC learning and spouse identifying phase, due to false PC's inclusion, many false spouses may enter $SP(C)$, leading to a large size of $SP(C)$. BAMB will perform an subset search in the union of current

$SP(C)$ and $CPC(C)$ to remove false PC and spouses respectively, and thus the large size of $\{SP(C) \cup CPC(C)\}$ will make BAMB both time and data inefficient.

EEMB. To tackle the drawback of BAMB, the EEMB (efficient and effective MB) algorithm (Wang et al., 2020) breaks BAMB into two independent subroutines: ADDTrue and RMFalse. EEMB first uses the ADDTrue subroutine to learn the candidate PC set and the spouse set, then employs the RMFalse subroutine for pruning the two sets. In the ADDTrue subroutine, before a candidate PC feature X is added to the current $CPC(C)$, EEMB will test whether X is independent of C using the current $CPC(C)$. If so, X will be discarded and consider the next candidate PC feature. If not, EEMB is triggered to identify the spouses of C with regard to X without performing an subset search in the current $SP(C)$. After this pruning, EEMB will greatly prune the false PC features before the spouse learning phase is triggered and make both $CPC(C)$ and $SP(C)$ keep as small as possible before the RMFalse subroutine runs. In the RMFalse subroutine, EEMB first uses the union of $SP(C)$ and current $CPC(C)$ to prune $CPC(C)$, then removes false positives from $SP(C)$ using the union of the updated $CPC(C)$ and current $SP(C)$.

3.3.4 Methods of MB learning with relaxed assumptions

In this subsection, we will discuss six representative MB learning algorithms for tackling the situation where the faithfulness or causal sufficiency assumption is violated, i.e., KI-AMB (Pena et al., 2007), TIE* (Statnikov et al., 2013), SGAI (Yu et al., 2017), LCMB (Liu and Liu, 2016), WLCMB (Liu and Liu, 2016), and M3B (Yu et al., 2018c).

KIAMB. Let S_1, S_2, Z and W denote four mutually disjoint feature subsets, the composition property assumes that if $S_1 \perp\!\!\!\perp S_2|Z$ and $S_1 \perp\!\!\!\perp W|Z$ hold, then $X \perp\!\!\!\perp (S_2 \cup W)|Z$ holds (Pearl, 2014). The composition property assumption is much weaker than the faithfulness assumption. The KIAMB algorithm (Pena et al., 2007) aims to tackle MB learning when the faithfulness assumption is violated. The difference between KIAMB and IAMB is that KIAMB allows the user to specify the trade-off between greediness and randomness in the MB search through a randomization parameter $K \in [0,1]$. IAMB greedily adds to $CMB(C)$ the feature with the highest association with C among all features excluding features currently in $CMB(C)$, while KIAMB adds to $CMB(C)$ the features with the highest associations with C in the CanMB set which is a random subset of $CMB(C)$ with size $max(1, \llcorner(|CMB(C)| \cdot K)\lrcorner)$. K specifies the trade-off between greediness and randomness in the MB search: if setting $K = 1$, KIAMB is reduced to IAMB, while if taking $K = 0$, KIAMB is a completely random approach which is expected to identify all the MBs of C with a nonzero probability if running repeatedly for enough number of times. IAMB and KIAMB are both correct under the composition assumption (Pena et al., 2007). However, KIAMB does not guarantee finding all MBs of the class variable under the composition assumption and is computationally more expensive than IAMB because it has to be run multiple times.

TIE*. Statnikov et al. (Statnikov et al., 2013) relaxed the composition assumption to the local composition assumption and proposed a family of the TIE* (Target Information Equivalence) algorithm for multiple MB learning. Specifically, the TIE* algorithm mainly includes three steps. In Step 1, TIE* uses an existing single MB learning algorithm to learn a $MB(C)$ from a data set D defined on F (i.e., the original distribution) and outputs

$MB(C)$. In Step 2, TIE* uses a procedure to generate a new data set D_{new} (i.e., the embedded distribution that is obtained by removing subsets of features of $MB(C)$ from the original distribution D). The motivation is that D_{new} may lead to identifying of a new $MB(C)$ that was previously "invisible" to a single MB learning algorithm since it was "masked" by another MB of C. Next, in Step 3 the MB learning algorithm employed in Step 1 is applied to D_{new}, resulting in a new candidate MB of C, called $CMB_{new}(C)$ in the embedded distribution. If $CMB_{new}(C)$ is also a MB of C in the original distribution according to a criterion (independence tests or classification accuracy), then $CMB_{new}(C)$ is considered as a new MB of C. Steps 1-3 are repeated until all possible data sets D_{new} generated by the procedure used in Step 2 have been considered. It has been proved that the TIE* algorithm can output all possible MBs of the class variable in data set when the faithfulness assumption is violated.

SGAI. When the faithfulness assumption is violated, it may not be tractable for TIE* to learn all possible MBs for feature selection due to computational complexities. To deal with this problem, the SGAI (Selection via Group Alpha-Investing) algorithm was proposed (Yu et al., 2017). Compared to the standard constraint-based MB learning algorithms discussed above, SGAI combines the MB theory with the idea of classical feature selection. Instead of an exhaustive search over a large number of MBs in a data set, SGAI presents the concept of a representative set which consists of the features of all possible MBs. Each member in the representative set is not a single feature, but a feature set (i.e., a group of features). SGAI first uses the existing MB learning algorithms (e.g., HITIOM-MB) to learn the representative sets. Then SGAI presents a group Alpha-investing procedure to select a best subset from representative sets. The group Alpha-investing procedure is motivated by the Alpha-investing feature selection method (Zhou et al., 2006) and can simultaneously optimize selections within each representative set as well as between those sets to achieve a feature subset that maximizes the predictive power for classification.

Compared to TIE*, SGAI does not learn all possible MBs from a data set, but chooses a feature subset that maximizes the prediction power for classification instead. However, when both the numbers of groups in the representative set and features in each group become large, SGAI may not be efficient and effective. Furthermore, since the number of MBs in a data set is not known, the representative set cannot guarantee to include the features of all possible MBs in the data set. In this case, the final output of SGAI is not optimal for feature selection.

LCMB and WLCMB. To tackle incorrect independent tests, in (Liu and Liu, 2016), the problem of incorrect independent tests is described as swamping and masking. Swamping means a true positive becomes a false negative, while masking means a true negative becomes a false positive. Based on the KIAMB algorithm, the LRH algorithm (Liu and Liu, 2016) was proposed to tackle the problem of swamping and masking and it is correct under the local composition assumption.

Compared to KIAMB, the innovation of LRH is that a selection-exclusion-inclusion (SEI) procedure was proposed to search for a candidate MB set of C which contains as few false positives as possible. Specifically, in the SEI procedure, the selection phase selects the candidate MB features of C conditioning on the MB currently selected, then for each feature in this MB set, the exclusion phase removes this feature if it is independent of C conditioning on its neighbors in the MB set; finally, the inclusion phase chooses the K features in the

current MB with the high associations with C as the output of the SEI procedure at each iteration. Since IAMB and KIAMB remain correct under the local composition assumption, in (Liu and Liu, 2016), IAMB, KIAMB and LRH were integrated into a framework called LCMB (Local Composition MB). Furthermore, to tackle the violation of the faithfulness assumption, based on the LCMB framework, WLCMB (Weak Local Composition MB) was proposed (Liu and Liu, 2016). WLCMB interleaves LCMB with a search-resuming procedure and has a higher computational complexity than LCMB.

M3B. The M3B (Mining Maximal ancestral graph MB) algorithm was proposed to tackle MB learning using independence tests when the causal sufficiency assumption is violated. A Maximal ancestral graph (MAG) model has been developed to deal with latent common causes without pre-determining the number of latent common causes and their exact locations with respect to other features (Richardson et al., 2002; Silva and Ghahramani, 2009; Borboudakis and Tsamardinos, 2016). Thus, instead of using DAGs, in (Yu et al., 2018c), authors adopted the MAG model to represent latent common causes and the concept of MBs. Specifically, the work in (Yu et al., 2018c) first defines the concept of MB of the class variable in a MAG, i.e., MAG MB (MMB), and presents a theoretical analysis of its properties. Then the M3B algorithm was proposed to learn the MMB of the class variable and it was the first constraint-based algorithm that was specially designed for MB learning when the causal sufficiency assumption is violated. The M3B algorithm mainly includes two novel methods to find the MMB of the class variable, the AdjV (Adjacent feature) algorithm using a backward strategy as shown in Algorithm 7 to find the parents and children of the class variable and the RecSearch (Recursive Search) algorithm to discover the remaining features of the MMB of the class variable.

3.3.5 Methods of MB learning with special purpose

In the section, we will discuss the six representative MB learning algorithms for some special purposes, i.e., MIMB for identifying a MB of a class variable from multiple data sets (Yu et al., 2018a), MCFS for stable predictions with distribution shift (Yu et al., 2019), MIAMB and MKIAMB for learning a MB of multiple class variables (Liu and Liu, 2018), and BASSUM and Semi-IAMB for weak supervision learning (Cai et al., 2011; Sechidis and Brown, 2018).

MIMB. The MB learning algorithms discussed above all learn MBs from a single observational data set. There has been an increasing availability of interventional data collected from various sources, such as gene knockdown experiments by different labs for studying the same diseases. Recently, Yu et al. (Yu et al., 2018a) studied the problems of MB learning in multiple interventional (experimental) data sets. This is the first work systematically studying the conditions for finding the correct MB of a class variable and the conditions for identifying the parents of class variable through MB learning. Based on the theoretical analysis, authors designed the MIMB (Multiple Interventional MB) algorithm to learn MB in multiple Interventional data sets. MIMB also adopts a divide-and-conquer approach which consists of two new subroutines. One subroutine, called MIPC, was designed for discovering $PC(T)$ from multiple interventional data sets using the IFBS framework as presented in Algorithm 6, and the other was proposed to identify spouses of C from multiple interventional data sets based on the framework as shown in Algorithm 7 .

MCFS. To achieve stable predictions for multiple data sets with different distributions, based on the theoretical results in (Yu et al., 2018a), the MCFS (multi-source causal feature selection) algorithm was proposed (Yu et al., 2019). By utilizing the concept of causal invariance (Pearl, 2009; Peters et al., 2016) and mutual information, MCFS formulates the problem of stable predictions in multiple data sets as a search for an invariant set across different data sets. To speed up the search, this work analyzed the upper and lower bounds of the invariant set and made MCFS learn the best invariant set within the bounds for stable predictions. MCFS outperforms some well-known existing feature selection algorithms designed for a single data set. Furthermore, this work demonstrated that for multiple data sets with different distributions, the set of parents of a class variable is the minimal and promising invariant set for stable predictions, while the MB or PC of the class feature may not.

MIAMB and MKIAMB. The algorithms described above all focus on learning a MB of a single class variable, e.g., $MB(C)$, the MB of C. Recently, the work in (Liu and Liu, 2018) explored the problem of learning a MB of multiple class variables, e.g., one MB, $MB(C_1, C_2)$ for both class variables C_1 and C_2. This work first proved that under the local intersection assumption a MB of multiple class variables can be constructed by simply taking the union of the MBs of the individual class variable excluding the class variables from the union (if they are included in the union). Then the MB learning problem for multiple class variables was transformed to a number of MB learning problems of a single class variable. By considering the violation of faithfulness assumption, MIAMB and MKIAMB were proposed in the work (Liu and Liu, 2018). For a set of class variables of interest, given an ordering which determines which class variable's MB needs to be learned in the current step, MIAMB and MKIAMB first find a MB of two class variables, and then learns an MB of three class variables and so on until all the class variables are considered.

BASSUM. In many real-world applications, labelled examples are often expensive to acquire while it is easy to collected unlabelled data examples. To leverage both unlabelled and labelled data to help MB learning (i.e., weak-supervision MB learning), Cai et al. (Cai et al., 2011) proposed a novel BAyesian Semi-SUpervised Method (BASSUM). To our best knowledge, BASSUM was the first weak-supervision MB learning algorithm. In the first phase, BASSUM learns the parents and children and then the spouses of C by taking into account both labelled and unlabelled data examples using a modified version of the G^2 test. The modified version of the G^2 test can use unlabelled data examples to enhance the reliability of the conditional independence tests. In the second phase, to prune the MB obtained in the first phase using unlabelled data examples, a concept of *effective feature sets* was proposed. It is a subset of the PC set of C obtained in the first phase. Using the effective feature sets, BASSUM prunes the PC set of C without accessing the information of C in labeled data examples. However, one weakness of BASSUM is that there are no guarantees that the modified G^2 test will follow a chi-squared distribution. BASSUM also cannot deal with the situation of partially labelled samples, i.e., we have a small number of binary labelled data and a vast amount of unlabelled examples.

Semi-IAMB. In order to deal with partially labelled data samples, the work in (Sechidis and Brown, 2018, 2015) first proposed a generalization of the conditional independence tests for partially labelled samples and then extended the work to semi-supervised data which contains a small number of binary labelled data and a large number of unlabelled examples. In the

Table 2: Representative score-based MB learning algorithms using score criteria

Category	Algorithm
Divide-and conquer MB learning (learning PC and spouses separately using a BN structure learning algorithm)	SLL (Niinimki and Parviainen, 2012)
	S^2TMB (Gao and Ji, 2017b)
	S^2TMB^+ (Gao and Ji, 2017b)
Simultaneous MB learning (learning PC and spouses simultaneously)	DMB (Acid et al., 2013)
	RPDMB (Acid et al., 2013)
MB learning with relaxed assumptions	BSS-MB (Masegosa and Moral, 2012)
	LMB-CSEM (Gao and Ji, 2016)

work, authors proposed some theoretical results of hypothesis testing (e.g., the G^2 test) and feature ranking in the partially labelled data environment.

Specially, by assuming that all missing labels are negative,or assuming that they are positive, authors present a surrogate class variable for semi-supervised hypothesis testing. That is, let C_0 represent assigning 0 to all missing class labels and C_1 represent assigning 1 to all missing class labels, authors propose to use surrogate test $X \perp\!\!\!\perp C_0$ or $X \perp\!\!\!\perp C_1$ to replace the true unlabelled class variable test $X \perp\!\!\!\perp C$). And they have proved that (1) both surrogate tests (i.e., $X \perp\!\!\!\perp C_0$ or $X \perp\!\!\!\perp C_1$) have exactly the same false positive rate as the ideal test (i.e., $X \perp\!\!\!\perp C$); (2) both surrogate tests will have a higher false negative rate than the ideal test. To reduce false negative rate, authors suggested use more data samples (if possible) or prior knowledge of the class probability to determine which one of the two surrogates will have the lower false negative rate. Moreover, in the work, it has been proved that both surrogate tests produce exactly the same feature ranking as $X \perp\!\!\!\perp C$. Then based on these theoretical results authors developed the Semi-IAMB algorithm (Sechidis and Brown, 2018) which uses the surrogate tests. However, the theoretical results in the work now only can deal with binary class variables and consequently Semi-IAMB cannot learn the MB of a class variable with more than two classes. In addition, to reduce false negative rate and improve feature ranking quality, Semi-IAMB requires more data samples and prior knowledge for surrogate tests.

4. Score-based methods

This type of methods employs score-based BN structure learning algorithms to learn the MB or PC of the class variable instead of using independence tests. Table 2 summarizes the representative score-based MB learning algorithms. Score-based MB learning algorithms are not the focus in MB learning research, thus the number of algorithms is much smaller than constraint-based algorithms.

In the following, Section 4.1 presents the basis of score-based methods. Section 4.2 gives the brief discussions of score-based methods. Section 4.3 extensively reviews the representative score-based methods.

4.1 Basis of score-based methods

Given a data set D, score-based BN learning algorithms aim to find the structure of the BN, i.e. the DAG, that maximizes a scoring function, which is usually defined as a measure of fitness between the DAG and D. They use the scoring function in combination with a greedy search method in order to measure the goodness of each explored structure from the space of feasible solutions.

The representative scoring functions designed based on different principles include K2 (Cooper and Herskovi 1992), BDeu (Buntine, 1991), BDe (Heckerman et al., 1995), BIC/MDL (Lam and Bacchus, 1994; Schwarz et al., 1978), AIC (Akaike, 1974), and MIT (Campos, 2006). The score-based BN learning problem can be formulated as: given D, learning a DAG $G*$ such that $G* = \arg\max_{G \subseteq O} f(G : D)$ where $f(G : D)$ is the scoring function and O is the family of all possible DAGs defined on D. A desirable property for a scoring function is the decomposability that enables to compute the global score of a DAG by aggregating local scores. $f(G : D)$ is decomposable if the score assigned to a structure can be expressed as a combination of local scores of each node and its parents in G: $f(G : D) = \sum_{V_i \in V} f(V_i; pa_G(V_i) : D_{V_i, pa_G(V_i)})$.

Since scoring functions are decomposable, the main idea of score-based MB learning algorithms is to learn a DAG of the features currently selected, C, and a new feature, then reads the MB (or PC) from the DAG at each iteration. Thus the score-based algorithms can distinguish parents from children of the class variable during MB learning, while the constraint-based algorithms cannot.

4.2 Overview of score-based methods

Existing score-based MB learning algorithms are mainly the score-based variants of the constraint-based MB learning algorithms. Through learning a DAG around a class variable, these algorithms read the MB of the class variable from the DAG. Since existing score-based MB learning algorithms are motivated from constraint-based methods, in Table 2, we categorize these algorithms into three types: divide-and-conquer MB learning, simultaneous MB learning, and MB learning with relaxed assumptions.

The SLL algorithm (Niinimki and Parviainen, 2012) is a score-based variant of the divide-and-conquer MB learning algorithms. In the PC learning and spouse identifying phases, SLL employs a BN structure learning algorithm to learn PC and spouses separately. To removing false positives, SLL implements the symmetric check using the AND rule to remove false positives in the found PC set, while the symmetric check using the OR rule to remove false positives in the found spouse set. The symmetric check makes SLL computationally expensive as the size of the MB of the class variable becomes large.

To improve the search efficiency of SLL, the S^2TMB algorithm (Gao and Ji, 2017b) was proposed which is a score-based variant of STMB. S^2TMB learns the spouses of C from $F \setminus CPC(C)$ instead of the union of parents and children of each feature in $PC(C)$, and employs the found spouses and PC to remove false positives instead of the symmetric check. S^2TMB$^+$ is an improved version of S^2TMB for further improving the computationally efficiency of S^2TMB.

Different from SLL and S^2TMB, DMB and RPDMB (Acid et al., 2013) do not divide MB learning into the PC learning step and the spouse identifying step. Instead, DMB and RPDMB learn PC and spouses of C simultaneously. These two algorithms only need to

---

Algorithm 9: Candidate PC leaning by SLL

  1: **Input**: Feature set F and the class variable C
     Output: $CPC(C)$
  2: $CPC(C) = \emptyset$;
  3: **repeat**
  4:    Select a feature $X \in F$;
  5:    $F = F \setminus X$;
  6:    //using an existing score-based BN learning algorithm
  7:    Learning a DAG on the set $CPC(C) \cup \{X\} \cup \{C\}$;
  8:    Obtain $CPC(C)$ from the learnt DAG;
  9: **until** F is empty;
10: Output $CPC(C)$.

---

learn a DAG around the class variable to obtain a MB of class feature instead of learning many local DAGs.

When the the faithfulness assumption is violated, the BSS-MB algorithm (Masegosa and Moral, 2012) was proposed to learn multiple MBs using a score criterion. It is a score-based variant of KIAMB for learning multiple MBs When the the causal sufficiency assumption is violated, the LMB-CSEM algorithm (Gao and Ji, 2016) was the first score-based algorithm to learn the MB of C with latent features in a DAG. Using score-based methods, BSS-MB does not guarantee finding all possible MBs, and it does not show significantly advantages over KIAMB or TIE* in terms of time efficiency and learning accuracy. LMB-CSEM needs to use the EM algorithm to tackle the missing values of latent variables, and thus it will be computationally expensive when the size of data samples is large.

In summary, so far it is not easy to use score criteria for MB learning when the faithfulness or causal sufficiency are violated, and these algorithms may suffer the computational problem of BN structure learning and they are still based on the framework of the constraint-based MB learning. And they algorithms do not show significant advantages over the constraint-based MB learning algorithms, and they have not attached as much attention as constraint-based methods in the MB learning research.

4.3 Detailed review of score-based methods

4.3.1 Divide-and conquer methods

In this subsection, we discuss the three representative score-based methods with the divide-and conquer strategy as follows.

SLL. The SLL (Score-based Local Learning) algorithm (Niinimki and Parviainen, 2012) first learns the PC set of a class variable as shown in Algorithms 9 and 10, and second identifies the spouses of the class variable as shown in Algorithm 11. Specifically, SLL includes the following four steps.

- (Step 1) Finding candidate PC of C. In Algorithm 9, initially $CPC(C) = \emptyset$. At each iteration, SLL randomly selects a feature $X \in F$ and removes X from F, then uses a score-based BN learning algorithm, such as those in (Chickering, 2002;

Algorithm 10: PC leaning with symmetric checks by SLL
___

1: **Input**: Feature set F and the class variable C
   Output: $PC(C)$
2: Find $CPC(C)$ using Algorithm 9;
3: //Symmetric check whether $C \in PC(X)(X \in CPC(C))$
4: **repeat**
5:     Select a feature $X \in CPC(C)$;
6:     $CPC(C) = CPC(C) \setminus X$;
7:     Obtain $CPC(X)$ using Algorithm 9;
8:     **if** $C \notin CPC(X)$ **then**
9:         $CPC(C) = CPC(C) \setminus X$;
10:    **end if**
11: **until** $CPC(C)$ is empty;
12: $PC(C) = CPC(C)$;
13: Output $PC(C)$.
___

Algorithm 11: Spouse learning by SLL
___

1: **Input**: Feature set F and $PC(C)$
   Output: $SP(C)$
2: Find candidate spouses $CSP(C)$, i.e., PC of each feature in $PC(C)$ using Algorithm 10;
3: $CSP(C) = CSP(C) \setminus PC(C) \cup C$; $SP(C) = \emptyset$;
4: **repeat**
5:     Select a feature $X \in CSP(C)$;
6:     $CSP(C) = CSP(C) \setminus X$;
7:     Learning a DAG on the set $PC(C) \cup \{X\}cup\{C\} \cup SP(C)$;
8:     Obtain SP(C) from the learnt DAG;
9: **until** $CSP(C)$ is empty;
10: Output $SP(C)$.
___

Koivisto and Sood, 2004; Rohekar et al., 2018), to learn a DAG of the set $(C \cup CPC(C) \cup X)$. SLL obtains a new $CPC(C)$ from the learnt DAG. The final $CPC(C)$ will be obtained until the set F is empty.

- (Step 2) Symmetry checks for pruning $CPC(C)$. SLL uses a score-based variant of symmetric checks as shown in Algorithm 10. SLL learns the PC of each feature X in $CPC(C)$ using Algorithm 9. If $C \notin CPC(X)$, SLL removes X from the $CPC(C)$.

- (Step 3) Identifying the spouses of C as shown in Algorithm 11. Let $SP(C) = \emptyset$. SLL first uses Algorithm 9 to find the union of PC of each feature in $PC(C)$ obtained in Step 2 as the candidate spouses of C, called $CSP(C) \setminus PC(C) \cup C$. Then for each feature X in this union, SLL learns a DAG of $(C \cup PC(C) \cup X \cup SP(C))$ and obtains a new $SP(C)$ from the learnt DAG until the union is empty.

- (Step 4) Finalizing spouses of C by the OR-rule symmetry constraint. In this step, SLL performs symmetric checks for finalizing spouses. That is, if $C \in SP(X)$ but $X \notin SP(C)$, using the OR rule, X should be added to $SP(C)$. SLL first uses Algorithm 11 to find $SP(C)$. Then SLL learns the spouses of all features in $F \setminus PC(C)$ using Algorithm 11. If the spouse set of a feature includes C, the feature will be added to $SP(C)$. The symmetric check will be computational expensive when the size of $F \setminus PC(C)$ is large.

SLL is computationally expensive to lean DAGs for symmetric checks in Steps 2 and 4, especially with a large size of the MB of C.

S^2**TMB.** The S^2TMB (Score-based Simultaneous MB) algorithm aims to improve the search efficiency of SLL by removing the symmetry checks in both PC and spouse search steps (i.e., Steps 2 and 4 of SLL). S^2TMB mainly consists of the following two steps.

- (Step 1) S^2TMB shares the same Step 1 as SLL for learning $CPC(C)$.

- (Step 2) Pruning $CPC(C)$ and identifying $SP(C)$. Let $SP(C) = \emptyset$ and $R = F \setminus CPC(C)$. S^2TMB learns the spouses of C (i.e., $SP(C)$) from R instead of the union of parents and children of each feature in $PC(C)$. It prunes $CPC(C)$ and identifies $SP(C)$ simultaneously at Step 2. For each feature $X \in R$, S^2TMB learns iteratively a DAG of the subset of $C \cup CPC(C) \cup SP(C) \cup X$, and prunes $CPC(C)$ and obtains $SP(C)$ using the learnt DAG, until R is empty.

S^2**TMB$^+$.** However, in Step 2, the size of $CPC(C) \cup SP(C)$ may grow uncontrollably large, leading to the same expensive computational cost as BN structure learning. To make the size of BN structures learnt at each iteration as small as possible, S^2TMB$^+$ decomposes Step 2 of S^2TMB into two steps as follows. At Step 2(a), S^2TMB$^+$ only learns a DAG of $C \cup CPC(C) \cup X$ to prune $CPC(C)$ and obtain $SP(C)$ instead of $C \cup CPC(C) \cup SP(C) \cup X$. And Step 2(b) uses the features in $SP(C)$ one by one to prune both $CPC(C)$ and $SP(C)$.

- (Step 1) S^2TMB$^+$ uses the same method as S^2TMB for learning $CPC(C)$.

- (Step 2a) Pruning $CPC(C)$ and learning $SP(C)$. Let $SP(C) = \emptyset$ and $R = F \setminus CPC(C)$ initially. For each feature $X \in R$, S^2TMB$^+$ learns iteratively a DAG of the subset of $C \cup CPC(C) \cup X$ instead of $C \cup CPC(C) \cup SP(C) \cup X$, and prunes $CPC(C)$ and obtains $SP(C)$ using the learnt DAG, until R is empty.

- (Step 2b) Pruning spouses and $CPC(C)$. In this step, let $R = SP(C)$ and $SP(C) = \emptyset$. For each feature X in R, S^2TMB$^+$ learns iteratively a DAG of the subset $C \cup CPC(C) \cup X \cup SP(C)$, then obtain $CPC(C)$ and $SP(C)$ from the learnt DAG, until R is empty.

4.3.2 Simultaneous MB learning methods

DMB and RPDMB (Acid et al., 2013) are different from SLL and S^2TMB. They do not divide MB learning into the PC learning step and the spouse identifying step. Instead, DMB and RPDMB learn PC and spouses of C simultaneously. These two algorithms only need to learn a local DAG around the class variable to obtain a MB of class feature instead of

learning many local DAGs. Specially, they first define two restricted search spaces, that is, CDAGs (Class-focused DAGs, see Definition 1 in (Acid et al., 2013)) and CRPDAGs (Class-focused Restricted Partially Directed Acyclic Graphs, see Proposition 1 in (Acid et al., 2013)). Then starting from an empty graph, using the hill-climbing-based search operators proposed in (Acid et al., 2005), DMB carries out a local search in the space of CDAGs while RPDMB implements the search in the space of CRPDAG. Both algorithms terminate until the scoring function does not improve. Finally, the two algorithms read $MB(C)$ from the obtained graphs respectively. Compared to SLL and S^2TMB, DMB and RPDMB do not need to learn DAGs many times. But the problem is that how to obtain the two restricted search spaces is not clear in the paper (Acid et al., 2005). If the size of the restricted search space is large, the computational cost of learning DAGs will be expensive.

4.3.3 MB LEARNING METHODS WITH RELAXED ASSUMPTIONS

In this section, we will discuss two representative score-based MB learning algorithms, BSS-MB and LMB-CSEM, for tackling the situations when the assumptions of faithfulness or causal sufficiency is violated.

BSS-MB. The BSS-MB (Bayesian stochastic search of MBs) algorithm (Masegosa and Moral, 2012) is a score-based variant of KIAMB for learning multiple MBs when the the faithfulness assumption is violated. BSS-MB adopts a strategy similar to that used by KIAMB with $K = 0$, but it uses a Bayesian score framework instead of conditional independence tests. In the growing phase, BSS-MB incrementally adds new features to the candidate MB sets by computing the posterior probability of a conditional independence statement. In the shrinking phase, BSS-MB removes from the MB sets the false positives identified using a Bayesian score. In addition, compared to KIAMB, each MB set found by BSS-MB has an associated score which measures how well this feature subset acts as a MB.

LMB-CSEM. LMB-CSEM (Gao and Ji, 2016) treats identifying the latent features included in the MB as a missing value problem. It first assumes the existences of latent features in the MB of C, then assigns these latent features into different non-overlapping latent subspaces. Within each subspace, LMB-CSEM employs a constrained structure expectation-maximization (CSEM) algorithm to greedily learn the MB with latent features. Then the final optimal MB is obtained from the optimal MBs within each subspace. Specifically, LMB-CSEM has three major steps. At Step 1, LMB-CSEM uses a standard MB discovery algorithm to find a MB of C from observed features as the baseline. At Step 2, using the baseline MB set, LMB-CSEM employs the CSEM algorithm to learn a MB with one latent feature within each subspace. At Step 3, if the score of the learned MB with one latent feature in one subspace is higher than that of the baseline MB, the learned MB will be considered as a new baseline MB, and Steps 2 and 3 are repeated to learn another latent feature until adding more latent features into the learned MB no longer improves the MB score or violates the size constraint.

5. Methods for distinguishing parents from children

Distinguishing direct causes and direct effects of a class variable is critical to the prediction of the consequence of the actions/interventions or the construction of a robust prediction model in various big data analytics and machine learning tasks. Existing studies have illustrated

that the direct causes of a class variable is the minimal and stable subset for classification when training data and testing data have different distributions (Yin et al., 2008; Yu et al., 2019). However, existing constraint-based MB or PC learning methods cannot distinguish parents from children, as they are based on conditional independence tests only, which does not indicate directions of relationships. Table 3 summaries three approaches for addressing this problem, global BN structure learning, local structure learning, and local structure learning with experimental data. Note that in Table 3 we only summarize the representative local-to-global BN structure learning algorithms since global BN structure learning often is computationally prohibitive (Ghoshal and Honorio, 2017).

Global structure learning. A global structure learning algorithm first learns a global BN structure, then reads parents and children of C from the learnt BN (Nie et al., 2017; Benjumeda et al., 2019; Campos and Ji, 2011; Yehezkel and Lerner, 2009). However, global BN structure learning is a NP-hard problem due to the huge candidate structure in the search space (Chickering et al., 2004; Ghoshal and Honorio, 2017).

Accordingly, local-to-global structure leaning methods were proposed. This type of algorithms first identifies each feature's MB (or PC) using the existing causality-based feature selection methods, then constructs a DAG skeleton (i.e., a undirected graph) using the found MBs (or PCs), and finally orients the edges of the skeleton using independence tests or score criteria. Several local-to-global structure learning approaches have been proposed to reduce the complexity of global BN structure learning, such as GSBN (Growing-shrinking BN) (Margaritis and Thrun, 2000), MMHC (Max-Min Hill Climbing) (Tsamardinos et al., 2006), and SLL+C/G (Niinimki and Parviainen, 2012). In order to improve the MB learning efficiency, Pellet et al. (Pellet and Elisseeff, 2008) employed the Relief feature selection algorithm and proposed the TC (Total Conditioning) algorithm. This algorithm uses the Relief feature selection algorithm to identify an approximate MB, then employs conditional independence tests to orient edges. Gao et al. (Gao et al., 2017) recently proposed a novel GGSL (Graph Growing Structure Learning) algorithm for local-to-global structure learning. Instead of finding the MBs of all features first, GGSL starts with a randomly selected feature, and then gradually expands the learned structure through a series of local structure learning steps using score-based MB learning algorithms. Later, Gao et al. (Gao and Wei, 2018) proposed a parallel Bayesian network structure learning algorithm, call PSL, using multiple local structure learning agents at the same time. These studies have shown that the local-to-global BN learning approach can deal with a data set with thousands of features.

Local structure learning. Since in many real-world applications we are only interested in the causal relationships around a class variable (e.g., causal genes of lung cancer in a gene data set), it is not necessary to waste time and memory to learn a global BN structure. Then several local learning algorithms have been designed for distinguishing direct causes and direct effects of a class variable, such as the CMB (Causal Markov Blanket) (Gao and Ji, 2015), PCD-by-PCD (Parents, Children and some Descendants) (Yin et al., 2008), and MB-by-MB algorithms (Wang et al., 2014). Given a class variable, these algorithms first find a MB or PC of the class variable and constructs a local structure among the class variable and the features in the MB or PC, then sequentially finds the MB or PC of the features connected to the class variable and simultaneously constructs local structures along the paths starting from the class variable until the parents and children of the class variable have been distinguished or it is clear that the parents and children cannot be distinguished

Table 3: Representative methods for distinguishing parents from children

Category	Algorithm
Global structure learning	GSBN (Margaritis and Thrun, 2000)
	MMHC (Tsamardinos et al., 2006)
	TC (Pellet and Elisseeff, 2008)
	SLL+C (Niinimki and Parviainen, 2012)
	SLL+G (Niinimki and Parviainen, 2012)
	GGSL (Gao et al., 2017)
	PSL (Gao and Wei, 2018)
Local structure learning	PCD-by-PCD (Yin et al., 2008)
	CMB (Gao and Ji, 2015)
	MB-by-MB (Wang et al., 2014)
Local structure learning with experimental data	ODLP (Statnikov et al., 2015)
	ODLP* (Statnikov et al., 2015)

further by continuing the process. For a class variable in a large network, the local learning algorithms are able to greatly reduce CPU time through finding only a local structure around the class variable compared with the entire BN network learning approaches. Since all existing local learning methods employ the standard MB or PC learning algorithms, they inherit the drawbacks of these algorithms. Furthermore, the local learning algorithms need to sequentially find the MBs or PCs of the features until the causes and effects of the class variable have been distinguished, and thus their time complexity may not be controllable.

Local structure learning with experimental data. The local structure learning methods discussed above are fully dependent on observational data and may not be able to orient all edges with observational data alone (Aliferis et al., 2010b; Statnikov et al., 2015). Statnikov et. al. (Statnikov et al., 2015) proposed the ODLP algorithm for distinguishing direct causes and direct effects of the class variable using both observational and experimental data. In this work, two versions of the ODLP algorithm, ODLP* and ODLP, were proposed. Under the faithfulness assumption, the ODLP* algorithm first uses the HITON-PC or MMPC algorithm to find the set of PC of the target with observational data, then orients edges using experimental data to distinguish direct causes from direct effects of the class variable. The ODLP* algorithm is sound and complete. In order to increase the learning accuracy, by relaxing the faithfulness assumption, ODLP employs the TIE* algorithm to learn multiple PC sets of the class variable, then finds the true parents and children of the class variable from the union of the multiple PC sets with experimental data. The rational behind the idea is that a feature may have multiple sets of parents and children when the faithfulness assumption is violated (Statnikov et al., 2013).

6. The toolbox

There are several open-source toolboxes for Bayeisan or causal network learning, such as the well-known BNT in MATLAB [1](Murphy et al., 2001), PGM in R[2], bnlearn in

---

R^3(Scutari, 2009), tetrad in JAVA^4(Scheines et al., 1998), and pcalg in R^5(Kalisch et al., 2012). However, these toolboxes do not focus on causality-based feature selection, but Bayesian or causal network structure learning. For example, the bnlearn toolbox contains the several causality-based feature selection algorithms, such as GSMB, IAMB, Inter-IAMB, Fast-IAMB, MMPC, and HITON-PC, but it aims to use these algorithms for implementing algorithms of BN learning, inference, and classification. The Causal Explorer package (Statnikov et al., 2010) is a well-known local causal discovery package in MATLAB, including several representative causality-based feature selection algorithms, but it is not provided with source code and not available for public use now.

In this paper, we have developed the CausalFS toolbox for causality-based feature selection. The CausalFS toolbox provides the first comprehensive open-source library for use in C/C++ that implements the state-of-the-art algorithms of causality-based feature selection. The toolbox is designed to facilitate the development of new algorithms in this exciting research direction and make it easy to compare new methods and existing ones. The CausalFS toolbox is available from https://github.com/kuiy/CausalFS.

CausalFS was developed in Linux systems. The architecture of the CausalFS toolbox in Figure 5 contains three layers: application, algorithm, and data. The three layers are designed independently. This makes it easy to implement and extend CausalFS. One can easily add a new algorithm to the CausalFS toolbox and share it through the CausalFS framework without modifying the other layers. In the algorithm layer, CausalFS mainly implements 28 representative causality-based feature selection methods, including 24 constraint-based algorithms (i.e., 16 algorithms for learning a single MB, 2 algorithms for learning multiple MBs, and 6 algorithms for learning PC and 4 score-based MB and PC learning algorithms.

The algorithm layer of CausalFS can also support local-to-global BN structure learning. By applying the MB and PC learning algorithms in the algorithm layer, using CausalFS, it is easy to design different local-to-global structure learning methods. For example, using the MMMB algorithm, we can generate MMMB based local-to-global structure learning algorithm. All implementation details are included in the detailed documentation available at https://github.com/kuiy/CausalFS, where all algorithms and related data structures are explained in detail.

7. Evaluations of causality-based feature selection methods

In this section, we systematically evaluate the causality-based feature selection algorithms using the CausalFS package in two aspects as follows with synthetic and real-world data sets.

- For a synthetic data set, we can read the MB or PC or parents of a feature in the corresponding benchmark BN. We evaluate the quality of the MB or PC of a variable learnt by an algorithm by comparing the MB or PC of the variable with the true MB or PC of the variable in the BN.

---

3. http://www.bnlearn.com/
4. https://github.com/cmu-phil/tetrad
5. https://cran.r-project.org/web/packages/pcalg/index.html

Application Layer

Bayesian network structure learning

Classification

APIs

Algorithm Layer

constraint-based

MB discovery

single-MB

GSMB
IAMB
Inter_IAMB
Fast_IAMB
LRH
BAMB
EEMB
FBED

MMMB
PCMB
HITON_MB
Semi_HITON_MB
MBOR
IPC_MB
STMB
CCMB

multi-MB

KIAMB
TIE*

PC discovery

PC_simple
MBtoPC
HITON_PC
Semi_HITON_PC
GetPC
MMPC

score-based

MB discovery

single-MB

SLL
S^2TMB

PC discovery

SLL-PC
S^2TMB-PC

Data

Data Layer

Data import

Bechmark data

Real-word data

Figure 5: Architecture of the causality-based feature selection toolbox

- For a real-world data set, we evaluate a causality-based feature selection algorithm based on the classification performance of the selected features.

All experiments were conducted on a computer with Intel(R) i5-2600, 3.4GHz CPU, and 8GB memory. In the following experiments, the G^2 test is used for independence tests and the significance level for the tests is set to 0.05 and 0.01, respectively. In all tables reporting experiment results, "-" denotes that an algorithm cannot deal with a data set due to expensive computations (the running time of the algorithm exceeded the three-day time threshold and it was stopped).

7.1 Experiments on synthetic data

In this section, we systematically evaluate the algorithms in the CausalFS package using the four standard benchmark BNs as shown in Table 4. Using the four benchmark networks, we randomly generate two groups of data, one including 5 data sets with 500 data instances each, and the other also containing 5 data sets with 5,000 data instances each. Using the two groups of data sets, we evaluate the causality-based feature selection algorithms for learning MB and PC, respectively. The following metrics are used to compare the MB or PC of a variable learnt by a causality-based feature selection algorithms with the true MB or PC of the variable in the BN.

Table 4: Summary of benchmark BNs

Network	Num. Vars	Num. Edges	Max In/out-Degree	Min/Max |PCset|	Variable Domain
Child	20	25	2/7	1/8	2-6
Alarm	37	46	4/5	1/6	2-4
Pigs	441	592	2/39	1/41	3-3
Gene	801	972	4/10	0/11	3-5

- Precision. The number of true positives in the output (i.e., the features in the output belonging to the true MB or PC of a variable) divided by the number of features in the MB or PC output by an algorithm. Precision reports the false positive rate in the output of an algorithm.

- Recall. The number of true positives in the output divided by the total number of true positives (the size of the true MB or PC) Recall reports the true positive rate in the output of an algorithm.

- $F1 = 2 * precision * recall/(precision + recall)$. It 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 use the time consumed in seconds to measure the efficiency of an algorithm.

For an algorithm, we first use it to learn the MB or PC of each variable in a data set, then compute the average results of recall, precision, F1, and running time over all variables in the data set, and finally we report these average results over five data sets. In all tables of experiment results, a value pair A/B indicates the precision, recall or $F1$ of an algorithm when the significance level of independence tests is set to 0.01 (represented by A) and 0.05 (represented by B) respectively.

Results of PC learning. From Tables 5 and 6, for the time efficiency, we have the following observations.

- An algorithm using a forward strategy for PC learning may be faster than an algorithm using the backward strategy. Tables 5 and 6 illustrate that among the six constraint-based PC learning methods, MMPC, HITON-PC, and semi-HITON-PC are very competitive in terms of efficiency, but they are faster than the other PC algorithms. The explanation is that MMPC, HITON-PC, and semi-HITON-PC use a forward strategy, while Recongize-PC uses a backward strategy.

- The symmetry check makes a PC learning algorithm computationally expensive. GetPC uses a forward strategy, but it is not efficient since it employs the symmetry check using the AND rule in the PC learning phase. MBtoPC is slower than the other algorithms, because it performs the symmetry check using the OR rule. SLL-PC implements the symmetry check for both PC and spouse learning. S^{2}TMB-PC is faster than SLL-PC since it does not perform the symmetry check.

36

Table 5: Results of PC learning methods on synthetic data sets (size=500)

hline Network	Algorithm	F1	Precision	Recall	Time
	PC-simple	0.89/0.89	0.94/0.91	0.86/0.88	0/0
	MMPC	0.90/0.87	0.93/0.87	0.89/0.89	0/0
	HITON-PC	0.91/0.88	0.94/0.89	0.89/0.91	0/0
Child	Semi-HITON-PC	0.91/0.89	0.94/0.90	0.89/0.91	0/0
	GetPC	0.84/0.84	0.93/0.91	0.80/0.81	0/0
	MBtoPC	0.86/0.83	0.95/0.90	0.81/0.80	0/0
	SLL-PC	0.87	0.95	0.83	0.47
	S^{2}TMB-PC	0.86	0.91	0.85	0.15
	PC-simple	0.82/0.84	0.89/0.88	0.79/0.84	0/0
	MMPC	0.85/0.84	0.90/0.84	0.83/0.87	0/0
	HITON-PC	0.84/0.83	0.89/0.83	0.83/0.87	0/0
Alarm	Semi-HITON-PC	0.85/0.84	0.89/0.85	0.83/0.87	0/0
	GetPC	0.77/0.82	0.88/0.89	0.73/0.80	0/0
	MBtoPC	0.86/0.85	0.95/0.92	0.81/0.82	0/0
	SLL-PC	0.91	0.93	0.91	0.38
	S^{2}TMB-PC	0.88	0.87	0.93	0.31
	PC-simple	0.99/0.95	0.98/0.93	1/1	0.02/0.03
	MMPC	0.91/0.77	0.86/0.66	1/1	0/0.01
	HITON-PC	0.91/0.77	0.86/0.66	1/1	0.01/0.01
Pigs	Semi-HITON-PC	0.91/0.77	0.87/0.67	1/1	0.01/0.01
	GetPC	0.99/0.95	0.98/0.92	1/1	0.04/0.05
	MBtoPC	0.93/0.89	0.97/0.90	0.92/0.92	0.02/0.03
	SLL-PC	-	-	-	-
	S^{2}TMB-PC	-	-	-	-
	PC-simple	0.96/0.91	0.97/0.89	0.96/0.96	0.01/0.02
	MMPC	0.83/0.70	0.79/0.60	0.92/0.93	0.01/0.01
	HITON-PC	0.83/0.70	0.79/0.60	0.92/0.93	0.01/0.01
Gene	Semi-HITON-PC	0.83/0.71	0.79/0.61	0.92/0.93	0.01/0.01
	GetPC	0.96/0.94	0.97/0.94	0.95/0.96	0.03/0.04
	MBtoPC	0.86/0.80	0.91/0.83	0.84/0.85	0.03/0.05
	SLL-PC	-	-	-	-
	S^{2}TMB-PC	-	-	-	-

Table 6: Results of PC learning methods on synthetic data sets (size=5,000)

Network	Algorithm	F1	Precision	Recall	Time
	PC-simple	1/0.96	0.99/0.94	1/1	0.01/0.02
	MMPC	0.98/0.92	0.96/0.87	1/1	0.01/0.01
	HITON-PC	0.98/0.92	0.97/0.88	1/1	0.01/0.01
Child	Semi-HITON-PC	0.98/0.92	0.97/0.88	1/1	0.01/0.01
	GetPC	1/1	1/0.99	1/1	0.05/0.05
	MBtoPC	0.97/0.93	0.98/0.92	0.97/0.97	0.02/0.02
	SLL-PC	0.97	1	0.94	5.2
	S^{2}TMB-PC	0.98	0.99	0.97	1.51
	PC-simple	0.98/0.97	0.97/0.96	0.99/0.99	0.02/0.02
	MMPC	0.97/0.93	0.97/0.90	0.99/0.99	0.01/0.01
	HITON-PC	0.97/0.92	0.97/0.89	0.99/0.99	0.01/0.01
Alarm	Semi-HITON-PC	0.97/0.93	0.97/0.90	0.99/0.99	0.01/0.01
	GetPC	0.98/0.98	1/1	0.97/0.97	0.04/0.04
	MBtoPC	0.98/0.96	0.99/0.97	0.97/0.97	0.03/0.03
	SLL-PC	0.97	0.96	0.98	3.49
	S^{2}TMB-PC	0.95	0.94	0.98	2.83
	PC-simple	0.99/0.95	0.98/0.93	1/1	0.64/0.78
	MMPC	0.91/0.78	0.86/0.68	1/1	0.19/0.2
	HITON-PC	0.91/0.78	0.86/0.68	1/1	0.24/0.26
Pigs	Semi-HITON-PC	0.91/0.79	0.86/0.69	1/1	0.29/0.3
	GetPC	0.99/0.97	0.99/0.96	1/1	10.23/10.43
	MBtoPC	0.97/0.95	0.98/0.94	0.99/0.99	0.52/0.76
	SLL-PC	-	-	-	-
	S^{2}TMB-PC	-	-	-	-
	PC-simple	0.98/0.92	0.97/0.90	0.99/0.98	0.40/0.59
	MMPC	0.83/0.72	0.78/0.62	0.94/0.94	0.12/0.13
	HITON-PC	0.83/0.72	0.78/0.62	0.94/0.94	0.15/0.19
Gene	Semi-HITON-PC	0.83/0.72	0.78/0.63	0.94/0.94	0.15/0.19
	GetPC	0.98/0.96	0.98/0.95	0.99/0.98	0.64/0.87
	MBtoPC	0.91/0.90	0.91/0.89	0.93/0.93	0.69/1.22
	SLL-PC	-	-	-	-
	S^{2}TMB-PC	-	-	-	-

- A score-based PC learning algorithm may be slower than a constraint-based method due to the computational complexity of the BN structure learning. In Tables 5 and 6, both score-based methods are slower than any constraint-based PC learning algorithms and failed to produce results for the two large-sized networks before they were terminated after three days.

In terms of learning performance, the following conclusions are obtained.

- A PC learning algorithm using the backward strategy or the symmetry check can remove more false positives, especially on large-sized BN networks. On the two small-sized networks, Recongize-PC, GetPC, MBtoPC are very competitive with MMPC, HITON-PC, and semi-HITON-PC in terms of precision and recall. On the two large-sized networks, Recongize-PC, GetPC, and MBtoPC significantly outperform MMPC, HITON-PC, and semi-HITON-PC in terms of precision. This indicates that on the large-sized networks, learning PC using a backward strategy or the symmetry check can remove much more false positives.

- A PC learning algorithm using the backward strategy or the symmetry check is not affected by the significance level in terms of precision (i.e., false positive rate). In Tables 5 and 6, the recall metric is not affected much by the significance level parameter, but the precision metrics of MMPC, HITON-PC, and semi-HITON-PC are influenced greatly by this parameter, while and Recongize-PC, GetPC, and MBtoPC are not affected. Meanwhile, for the learning performance, SLL-PC (performing the symmetry check) and S^2TMB-PC are very competitive to PC-simple, GetPC, and MBtoPC on the two small-sized networks.

In summary, the backward strategy or the symmetry check is a double-edged sword. They will make a PC learning algorithm computationally expensive, but they are able to make the PC learning algorithms output a more accurate MB or PC set.

Results of MB learning. From Tables 7 and 8, we have the following conclusions.

- Existing constraint-based simultaneous MB learning algorithms are inferior the other MB learning algorithms. From Tables 7 and 8, we can see that regardless of sample size (500 or 5000), on the four BN networks, the six simultaneous MB learning algorithms are inferior to the remaining ten MB learning algorithms, especially on the two large-sized BN networks. Meanwhile, GSMB has the worst precisions and recalls among all rivals. STMB has the lowest precision among all the MB learning algorithms except for GSMB. The explanation is that STMB adopts the same strategy as the simultaneous MB discovery algorithms in the false positive removal phase and this leads to a large data sample requirements.

- The backward strategy and the symmetry check make a MB learning algorithm more accurate using a large-sized data samples. Since GetPC, Recongize-PC, and MBtoPC outperform the other PC learning algorithms as shown in Tables 6 and 7, their corresponding MB learning algorithms, PCMB, IPCMB, and MBOR, are better than the other constraint-based 11 MB learning algorithms on the two large-sized BN networks. For example, IPCMB achieves the best performance on the two large networks,

38

since IPCMB not only adopts the Recongize-PC algorithm to learn the PC of each variable, but also it performs the symmetry check in the spouse discovery procedure (Recongize-PC gets the best performance as shown in Tables 5 and 6).

Duo to unreliable independence tests using a small-sized data samples, a MB learning algorithm using the AND rule may remove the true MB members from its output when performing the symmetry check, and thus this may degrade the performance of the algorithm on recall. For example, in Table 5, using 500 samples on the two small-sized networks, PCMB and IPCMB get much lower recall than MMMB, HITON-MB, and semi-HITON-MB. Thus it is an interesting problem of studying under what conditions we use the AND rule or the OR rule or combining both to make MB learning more accurate.

- Score-based methods have the similar performance as constraint-based methods in terms of F1. From Tables 7 and 8, S^{2}TMB and SLL achieve similar performance as the best constraint-based MB learning methods in terms of both precision and recall with the two small-sized networks. These results are consistent with the results of S^{2}TMB-PC and SLL-PC in Tables 6 and 7. On the large-sized networks, due to expensive computation costs, S^{2}TMB and SLL did not produce results within three days.

- Simultaneous MB learning methods are the fastest while score-based methods are the slowest. Since the six simultaneous MB learning methods do not need to perform an exhaustive subset search within the currently selected features, they are faster than the eight divide-and-conquer MB methods. In addition, these six simultaneous MB learning algorithms have the similar performance in terms of efficiency. PCMB and IPCMB are the slowest among the divide-and-conquer MB methods, since they need to perform the symmetry check. Moreover, the Recongize-PC algorithm used by IPCMB is slower than MMPC, HITON-PC, and semi-HITON-PC as shown in Tables 5 and 6. STMB and BAMB do not need to perform the symmetry check. However, they need to perform an exhaustive subset search for PC learning. The these two algorithms are not significantly faster than MMMB, HITON-MB, and semi-HITON-MB. S^{2}TMB and SLL are slower than all constraint-based MB learning algorithms, since they need to use a BN structure learning algorithm to learn MBs. S^{2}TMB is faster than SLL because S^{2}TMB removes the symmetry check.

7.2 Experiments on real-world data

In the section, we validate the causality-based feature selection algorithms using eight real-world datasets from the UCI Machine Learning Repository and NIPS2003 feature selection challenge datasets as shown in Table 9. Among these eight datasets, three are of low dimensionality but contains a large number of samples, two are high dimensional datasets with large number of samples, and the other three are also of high dimensionality but small size.

We use the three classifiers, NB, KNN, and SVM to evaluate a selected feature subset for classification. The value of k for the KNN classifier is set to 3 and both SVM and KNN

Table 7: Results of MB learning methods on synthetic data sets (size=500)

Data	Algorithm	F1	Precision	Recall	Time
Child	GSMB	0.69/0.54	0.78/0.59	0.67/0.58	0/0
	IAMB	0.78/0.67	0.86/0.67	0.77/0.77	0/0
	Inter-IAMB	0.78/0.67	0.86/0.67	0.77/0.77	0/0
	Fast-IAMB	0.74/0.68	0.87/0.77	0.71/0.70	0/0
	LRH	0.80/0.68	0.86/0.67	0.80/0.82	0/0
	FBED	0.79/0.70	0.87/0.71	0.77/0.77	0/0
	MMMB	0.87/0.82	0.94/0.84	0.83/0.84	0/0
	PCMB	0.79/0.79	0.93/0.88	0.74/0.76	0/0
	HITON-MB	0.85/0.83	0.94/0.86	0.82/0.85	0/0
	Semi-HITON-MB	0.85/0.83	0.94/0.87	0.82/0.85	0/0
	MBOR	0.84/0.81	0.92/0.83	0.81/0.84	0/0
	IPCMB	0.80/0.82	0.93/0.88	0.74/0.79	0/0
	STMB	0.81/0.71	0.85/0.66	0.82/0.86	0/0
	BAMB	0.86/0.81	0.93/0.82	0.83/0.84	0/0
	EEMB	0.84/0.80	0.92/0.81	0.82/0.85	0/0
	SLL	0.84	0.95	0.79	0.85
	S^{2}TMB	0.83	0.95	0.77	0.14
Alarm	GSMB	0.29/0.19	0.35/0.21	0.27/0.20	0/0
	IAMB	0.76/0.73	0.89/0.79	0.71/0.74	0/0
	Inter-IAMB	0.77/0.72	0.89/0.77	0.72/0.75	0/0
	Fast-IAMB	0.71/0.67	0.85/0.77	0.65/0.66	0/0
	LRH	0.75/0.70	0.85/0.71	0.73/0.76	0/0
	FBED	0.77/0.74	0.90/0.80	0.71/0.73	0/0
	MMMB	0.82/0.82	0.90/0.85	0.78/0.84	0/0
	PCMB	0.74/0.79	0.88/0.89	0.68/0.76	0/0.01
	HITON-MB	0.82/0.81	0.90/0.84	0.78/0.85	0/0
	Semi-HITON-MB	0.82/0.83	0.90/0.86	0.78/0.85	0/0
	MBOR	0.85/0.85	0.93/0.89	0.81/0.85	0/0
	IPCMB	0.74/0.80	0.87/0.88	0.69/0.77	0/0.01
	STMB	0.67/0.59	0.69/0.54	0.76/0.84	0/0
	BAMB	0.80/0.80	0.90/0.85	0.75/0.80	0/0
	EEMB	0.80/0.79	0.90/0.85	0.75/0.87	0/0
	SLL	0.88	0.93	0.87	0.88
	S^{2}TMB	0.87	0.92	0.86	0.29
Pigs	GSMB	0.06/0.01	0.08/0.02	0.05/0.01	0/0
	IAMB	0.80/0.80	0.98/0.98	0.72/0.72	0.01/0.01
	Inter-IAMB	0.80/0.80	0.98/0.98	0.72/0.72	0.01/0.01
	Fast-IAMB	0.77/0.76	0.80/0.80	0.83/0.81	0.01/0.01
	LRH	0.69/0.68	0.90/0.87	0.60/0.60	0.01/0.02
	FBED	0.71/0.69	0.89/0.87	0.64/0.62	0/0
	MMMB	0.93/0.76	0.88/0.64	1/1	0.04/0.05
	PCMB	0.99/0.95	0.98/0.92	1/1	0.13/0.17
	HITON-MB	0.93/0.76	0.88/0.64	1/1	0.07/0.08
	Semi-HITON-MB	0.93/0.77	0.88/0.65	1/1	0.08/0.1
	MBOR	0.95/0.85	0.92/0.77	1/1	0.02/0.04
	IPCMB	0.99/0.97	0.99/0.95	1/1	0.60/0.67
	STMB	0.59/0.25	0.47/0.15	0.97/0.98	0.05/0.08
	BAMB	0.96/0.81	0.93/0.71	1/1	0.05/0.06
	EEMB	0.96/0.82	0.93/0.72	1/1	0.04/0.08
	SLL	-	-	-	-
	S^{2}TMB	-	-	-	-
Gene	GSMB	0.06/0.01	0.08/0.02	0.05/0.01	0/0
	IAMB	0.66/0.66	0.80/0.80	0.68/0.68	0.01/0.01
	Inter-IAMB	0.66/0.66	0.80/0.80	0.68/0.68	0.01/0.01
	Fast-IAMB	0.66/0.65	0.68/0.68	0.78/0.78	0.01/0.01
	LRH	0.71/0.67	0.89/0.82	0.67/0.67	0.01/0.02
	FBED	0.66/0.66	0.80/0.80	0.68/0.67	0.01/0.01
	MMMB	0.82/0.66	0.79/0.56	0.91/0.92	0.03/0.04
	PCMB	0.94/0.93	0.97/0.93	0.93/0.95	0.08/0.12
	HITON-MB	0.82/0.66	0.79/0.56	0.91/0.92	0.03/0.05
	Semi-HITON-MB	0.82/0.67	0.79/0.57	0.91/0.92	0.03/0.05
	MBOR	0.87/0.74	0.86/0.66	0.91/0.92	0.03/0.06
	IPCMB	0.96/0.94	0.98/0.94	0.94/0.96	0.04/0.07
	STMB	0.56/0.18	0.44/0.12	0.92/0.92	0.04/0.06
	BAMB	0.82/0.68	0.79/0.59	0.90/0.91	0.03/0.05
	EEMB	0.82/0.67	0.78/0.57	0.90/0.92	0.03/0.05
	SLL	-	-	-	-
	S^{2}TMB	-	-	-	-

Table 8: Results of MB learning methods on synthetic data sets (size=5,000)

Data	Algorithm	F1	Precision	Recall	Time
Child	GSMB	0.70/0.45	0.69/0.42	0.75/0.58	0/0
	IAMB	0.86/0.72	0.83/0.64	0.95/0.95	0.01/0.01
	Inter-IAMB	0.86/0.72	0.83/0.65	0.95/0.95	0.01/0.01
	Fast-IAMB	0.89/0.74	0.89/0.70	0.93/0.91	0.01/0.01
	LRH	0.85/0.71	0.81/0.63	0.97/0.96	0.02/0.03
	FBED	0.89/0.76	0.87/0.69	0.94/0.94	0.01/0.01
	MMMB	0.98/0.89	0.97/0.84	1/1	0.05/0.05
	PCMB	1/0.98	1/0.97	1/1	0.1/0.11
	HITON-MB	0.98/0.90	0.97/0.85	1/1	0.06/0.07
	Semi-HITON-MB	0.98/0.90	0.97/0.85	1/1	0.07/0.08
	MBOR	0.96/0.88	0.96/0.84	0.98/0.98	0.04/0.08
	IPCMB	1/0.98	1/0.97	1/1	0.09/0.11
	STMB	0.91/0.80	0.86/0.71	0.99/1	0.02/0.03
	BAMB	0.97/0.89	0.96/0.83	0.99/0.99	0.04/0.05
	EEMB	0.96/0.88	0.95/0.82	0.99/0.99	0.02/0.02
	SLL	0.96	1	0.93	9.1
	S^{2}TMB	0.98	1	0.96	1.3
Alarm	GSMB	0.33/0.20	0.34/0.20	0.35/0.26	0.01/0
	IAMB	0.92/0.80	0.94/0.75	0.91/0.92	0.02/0.02
	Inter-IAMB	0.92/0.79	0.94/0.72	0.92/0.93	0.02/0.03
	Fast-IAMB	0.91/0.79	0.94/0.75	0.91/0.92	0.01/0.01
	LRH	0.91/0.78	0.93/0.71	0.92/0.93	0.13/0.05
	FBED	0.93/0.83	0.96/0.80	0.91/0.92	0.01/0.01
	MMMB	0.97/0.93	0.98/0.90	0.97/0.98	0.04/0.04
	PCMB	0.97/0.98	1/1	0.96/0.96	0.08/0.09
	HITON-MB	0.97/0.92	0.98/0.89	0.97/0.98	0.04/0.05
	Semi-HITON-MB	0.97/0.93	0.98/0.9	0.97/0.98	0.05/0.06
	MBOR	0.97/0.96	0.98/0.95	0.97/0.98	0.05/0.06
	IPCMB	0.97/0.98	1/1	0.96/0.97	0.09/0.1
	STMB	0.79/0.71	0.76/0.63	0.95/0.97	0.04/0.04
	BAMB	0.96/0.92	0.98/0.91	0.94/0.96	0.03/0.04
	EEMB	0.96/0.93	0.99/0.93	0.94/0.95	0.02/0.03
	SLL	0.96	0.97	0.96	8.15
	S^{2}TMB	0.95	0.98	0.93	2.44
Pigs	GSMB	0.05/0.02	0.04/0.02	0.06/0.03	0.02/0.01
	IAMB	0.71/0.71	0.62/0.62	0.96/0.96	0.29/0.29
	Inter-IAMB	0.71/0.71	0.62/0.62	0.96/0.96	0.28/0.28
	Fast-IAMB	0.71/0.70	0.62/0.61	0.95/0.94	0.18/0.14
	LRH	0.80/0.76	0.75/0.68	0.96/0.96	0.87/1.22
	FBED	0.78/0.67	0.71/0.59	0.95/0.89	0.12/0.11
	MMMB	0.92/0.77	0.87/0.65	1/1	6.47/6
	PCMB	0.99/0.97	0.99/0.95	1/1	29.15/23.12
	HITON-MB	0.92/0.76	0.87/0.64	1/1	9.74/12.27
	Semi-HITON-MB	0.92/0.77	0.87/0.65	1/1	13.58/13.6
	MBOR	0.97/0.91	0.94/0.86	1/1	0.73/1.19
	IPCMB	1/0.98	1/0.97	1/1	11.17/11.37
	STMB	0.38/0.15	0.28/0.09	1/1	2.66/5.2
	BAMB	0.96/0.84	0.93/0.74	1/1	23.35/23.28
	EEMB	0.96/0.87	0.94/0.79	1/1	8.71/8.27
	SLL	-	-	-	-
	S^{2}TMB	-	-	-	-
Gene	GSMB	0.03/0.01	0.03/0.01	0.04/0.02	0.02/0.01
	IAMB	0.60/0.60	0.53/0.53	0.89/0.89	0.65/0.64
	Inter-IAMB	0.60/0.60	0.53/0.53	0.89/0.89	0.71/0.64
	Fast-IAMB	0.60/0.59	0.53/0.52	0.89/0.88	0.35/0.28
	LRH	0.66/0.61	0.60/0.55	0.89/0.88	0.98/3.3
	FBED	0.62/0.55	0.55/0.48	0.88/0.82	0.24/0.19
	MMMB	0.83/0.68	0.77/0.57	0.94/0.94	0.58/0.69
	PCMB	0.98/0.96	0.98/0.95	0.99/0.98	1.67/2.13
	HITON-MB	0.83/0.68	0.77/0.57	0.94/0.94	0.76/1.1
	Semi-HITON-MB	0.83/0.69	0.77/0.58	0.94/0.94	0.73/1.08
	MBOR	0.89/0.83	0.86/0.78	0.94/0.94	0.85/1.83
	IPCMB	0.99/0.97	0.99/0.96	0.99/0.99	1.92/2.96
	STMB	0.30/0.11	0.20/0.07	0.99/0.98	0.95/1.57
	BAMB	0.82/0.69	0.76/0.59	0.94/0.94	0.71/1.55
	EEMB	0.82/0.71	0.77/0.61	0.94/0.94	0.56/0.82
	SLL	-	-	-	-
	S^{2}TMB	-	-	-	-

use the linear kernel. In the following, we evaluate two types of feature subsets selected by the causality-based feature selection algorithms for classification, which are the MB and PC of the class variable, using the following metrics.

- Prediction accuracy. The number of correctly predicted data samples divided by the total number of data samples in a test data set.

- Compactness. The size of the feature subset selected by an algorithm.

- AUC. Area Under the ROC Curve for demonstrating the performance of a classifier for predicting a binary class variable with imbalanced classes.

From Tables 10 to 12, our observations are summarized as follows.

- The classification performance using the PC set of a class variable is not inferior to that of using the MB of the class variable. And PC learning is much more efficient than MB learning. The findings are consistent with the results in (Aliferis et al., 2010a). Thus in terms of feature selection, PC learning algorithms are practical in real-world applications, .

- When the size of data samples is large, the simultaneous MB learning approach (except for GSMB) is significantly faster than the other MB and PC learning algorithms, and they achieve very competitive prediction accuracy with their rivals. Surprisingly, FBED is the fastest algorithm and its performance is comparable with the others as shown in Tables 10 to 11.

- Existing score-based MB or PC learning algorithms are still computationally expensive or prohibitive when the size of PC or MB is large, even if the number of features in a data set is small. For example, on the *spambase* data set, we observe that when the size of MB is over 30, SLL and S^2TMB are very computationally expensive using the exact score-based BN learning algorithms. Using both synthetic and real-world data sets, existing score-based MB or PC learning algorithms do not show significant advantages over the constrain-based algorithms. The results also explain why the score-based MB learning approach is not the focus in the causality-based feature selection research.

- With the three class-imbalanced data sets, *infant*, *bankruptcy*, and *dorothea*, all MB learning algorithms achieve almost the same high prediction accuracy in Tables 10 to 11. In Table 12, on *infant* and *bankruptcy*, it is the same for all PC learning algorithms. However, in Tables 13 to 14, we can see that all MB and PC learning algorithms achieve much low values of AUC. Here we do not report the value of AUC of each algorithm on the remaining five class-balanced date sets, since the MB and PC learning algorithms obtain almost the same prediction accuracy and AUC. The results indicate that the existing causality-base feature selection algorithms are not able to deal with a data set with imbalanced classes well.

- The compactness and symmetry check. In terms of compactness, in Tables 10 to 11, STMB selects the most features than the other rivals for all data sets, while the score-based MB algorithms achieve the worst performance among all algorithms under

comparison. Meanwhile, the symmetry check may not be helpful for selecting a good feature subset for classification in real-world applications. MMMB, HITON-MB, and semi-HITON-MB achieve more stable and better prediction accuracy than the other rivals, although their outputs include more false positives than the other rivals as discussed above. The possible explanation is that when we use causality-based feature selection to deal with real-world data sets for classification, these real-world data sets may violate the faithfulness or causal sufficiency assumption. And this is the same for MMPC, HITON-PC, and semi-HITON-PC in Tables 12.

Table 9: Summary of real-world datasets

Dataset	Number of features	Number of instances	Category ratio
infant	86	5,337	0.94/0.06
spambase	57	4,601	0.61/0.39
bankruptcy	147	7,063	0.89/0.11
madelon	500	2,000	0.50/0.50
gisette	5,000	6,000	0.50/0.50
arcene	10,000	100	0.56/0.44
dexter	20,000	300	0.50/0.50
dorothea	100,000	800	0.90/0.10

8. Conclusion and open problems

In this paper, we first reviewed the state-of-the art causality-based feature selection algorithms, then described our developed open-source software package which implements the representative causality-based feature selection algorithms, and finally we evaluated the representative algorithms using synthetic and real-world data sets.The experiments using synthetic data sets have demonstrated that a backward strategy or the symmetry check significantly improves the precision of causality-based feature selection algorithms (i.e., helps removing more false positives). However, with real-world data sets, due to noise or small-sized data samples, allowing more false positives in the output of causality-based feature selection algorithms may be useful for classification. Using real world data sets, selecting a PC of the class variable for classification would be better and is much faster than selecting a MB. The score-based PC or MB learning algorithms do not suffer from the problem of data efficiency compared to the constraint-based MB learning algorithms.

Although a significant number of causality-based feature selection algorithms have been developed for big data analytics in the past two decades, more efforts are still required in this research direction to tackle the following challenges.

- Low-quality data. Missing or noise data are ubiquitous in many real-world application domains, which means that the values for one or more features in a data set are incorrect or missing from recorded observations. All existing causality-based feature selection methods assume that all features involved in a dataset do not have missing or noise values. It is challenging to address causality-based feature selection methods with low-quality big data.

- Streaming data. Existing causality-based feature selection methods assume that all data instances (or samples) in a data set are given before the learning starts. In fact,

Table 10: Comparison of MB methods on real-world data sets in prediction accuracy (1)

Data	Algorithm	NB	KNN	SVM	Compactness	Time
madelon	GSMB	0.58/0.55	0.51/0.52	0.56/0.55	7.00/7.00	0.01/0.00
	IAMB	0.58/0.58	0.62/0.62	0.63/0.63	7.00/7.00	0.21/0.22
	Inter-IAMB	0.58/0.58	0.62/0.61	0.63/0.63	7.00/7.00	0.22/0.21
	Fast-IAMB	0.57/0.57	0.57/0.56	0.61/0.60	6.00/6.00	0.06/0.06
	LRH	0.60/0.61	0.61/0.60	0.62/0.63	8.00/8.00	0.16/0.30
	FBED	0.60/0.59	0.61/0.57	0.62/0.61	7.10/7.90	0.06/0.03
	MMMB	0.58/0.59	0.55/0.63	0.60/0.64	5.70/9.40	0.14/0.23
	PCMB	0.56/0.58	0.50/0.52	0.55/0.58	1.50/3.50	0.24/0.42
	HITON-MB	0.58/0.60	0.57/0.60	0.61/0.63	5.60/8.40	0.15/0.24
	Semi-HITON-MB	0.58/0.60	0.56/0.60	0.60/0.63	5.50/8.30	0.15/0.26
	MBOR	0.60/0.60	0.59/0.60	0.61/0.61	6.40/9.40	0.38/0.78
	IPCMB	0.59/0.60	0.54/0.54	0.59/0.61	3.70/6.10	0.16/0.35
	STMB	0.60/0.60	0.56/0.52	0.62/0.60	24.00/83.40	0.13/0.31
	BAMB	0.60/0.59	0.61/0.65	0.63/0.64	7.20/9.60	0.27/0.47
	EEMB	0.60/0.61	0.59/0.62	0.62/0.62	6.60/8.10	0.17/0.26
	SLL	0.57	0.52	0.57	6.1	64.47
	S^{2}TMB	0.58	0.52	0.57	5.2	61.59
arcene	GSMB	0.71/0.70	0.68/0.71	0.66/0.67	3.60/3.70	0.00/0.00
	IAMB	0.69/0.69	0.64/0.64	0.66/0.66	4.00/4.00	0.10/0.11
	Inter-IAMB	0.69/0.69	0.64/0.64	0.66/0.66	4.00/4.00	0.10/0.11
	Fast-IAMB	0.70/0.67	0.66/0.64	0.70/0.68	3.00/3.00	0.05/0.06
	LRH	0.59/0.61	0.61/0.58	0.59/0.66	3.30/4.00	0.13/0.30
	FBED	0.68/0.63	0.58/0.60	0.66/0.65	3.90/4.00	0.03/0.03
	MMMB	0.80/0.79	0.66/0.74	0.79/0.76	3.90/6.60	1.61/2.33
	PCMB	0.61/0.62	0.60/0.60	0.61/0.63	1.70/2.00	2.88/5.01
	HITON-MB	0.74/0.70	0.67/0.69	0.69/0.73	3.70/6.90	1.71/2.48
	Semi-HITON-MB	0.69/0.71	0.62/0.62	0.66/0.73	3.50/5.80	1.56/2.20
	MBOR	0.66/0.71	0.63/0.73	0.66/0.71	3.30/6.80	0.12/0.59
	IPCMB	0.61/0.63	0.61/0.62	0.60/0.62	1.50/2.80	0.57/1.38
	STMB	0.63/0.61	0.71/0.71	0.71/0.67	56.10/293.10	0.64/0.87
	BAMB	0.73/0.70	0.63/0.65	0.66/0.70	3.90/7.30	0.53/1.28
	EEMB	0.75/0.72	0.67/0.66	0.69/0.72	3.80/5.90	0.85/1.13
	SLL	-	-	-	-	-
	S^{2}TMB	0.74	0.7	0.73	6.1	1648.84
dexter	GSMB	0.70/0.63	0.67/0.55	0.70/0.63	4.80/4.60	0.00/0.00
	IAMB	0.81/0.81	0.73/0.73	0.82/0.82	5.00/5.00	0.36/0.36
	Inter-IAMB	0.81/0.81	0.73/0.73	0.82/0.82	5.00/5.00	0.35/0.35
	Fast-IAMB	0.73/0.73	0.72/0.74	0.76/0.77	4.00/4.00	0.12/0.13
	LRH	0.78/0.75	0.80/0.77	0.79/0.77	4.80/5.00	0.30/1.02
	FBED	0.81/0.81	0.73/0.73	0.82/0.82	5.00/5.00	0.05/0.06
	MMMB	0.86/0.88	0.84/0.86	0.86/0.87	9.20/19.40	4.10/7.17
	PCMB	0.81/0.85	0.71/0.83	0.81/0.84	6.60/12.30	11.74/32.99
	HITON-MB	0.85/0.87	0.81/0.86	0.85/0.86	9.80/19.50	4.44/6.99
	Semi-HITON-MB	0.85/0.87	0.79/0.85	0.84/0.86	8.40/18.80	3.92/6.87
	MBOR	0.91/0.90	0.87/0.89	0.88/0.89	21.60/31.00	4.38/10.83
	IPCMB	0.82/0.86	0.71/0.85	0.82/0.85	5.90/12.40	6.19/13.08
	STMB	0.88/0.90	0.82/0.75	0.90/0.90	54.50/244.50	3.03/9.61
	BAMB	0.86/0.89	0.84/0.88	0.86/0.88	9.90/17.90	2.04/3.01
	EEMB	0.85/0.87	0.82/0.87	0.85/0.86	9.20/14.20	3.32/4.45
	SLL	-	-	-	-	-
	S^{2}TMB	-	-	-	-	-
dorothea	GSMB	0.93/0.90	0.92/0.90	0.93/0.90	4.30/7.00	0.05/0.01
	IAMB	0.93/0.93	0.93/0.94	0.93/0.94	7.00/7.00	42.48/44.64
	Inter-IAMB	0.93/0.93	0.93/0.93	0.93/0.93	7.00/7.00	40.89/41.32
	Fast-IAMB	0.93/0.93	0.92/0.90	0.93/0.92	6.00/6.00	12.64/12.89
	LRH	0.94/0.94	0.92/0.92	0.93/0.93	5.90/6.30	45.22/303.98
	FBED	0.93/0.93	0.93/0.93	0.93/0.93	7.00/7.00	2.72/5.32
	MMMB	0.93/0.93	0.92/0.91	0.93/0.93	8.60/16.60	288.00/555.40
	PCMB	0.45/0.92	0.45/0.92	0.45/0.92	0.60/2.50	331.04/1259.19
	HITON-MB	0.93/0.93	0.91/0.92	0.93/0.93	10.20/19.90	333.89/638.10
	Semi-HITON-MB	0.94/0.93	0.92/0.92	0.93/0.93	8.90/14.60	302.03/497.61
	MBOR	0.93/0.93	0.92/0.93	0.93/0.93	10.50/31.80	171.26/1870.99
	IPCMB	0.36/0.83	0.36/0.82	0.36/0.83	0.50/1.40	75.16/538.33
	STMB	-/-	-/-	-/-	-/-	-/-
	BAMB	0.93/0.93	0.90/0.91	0.93/0.93	10.50/20.20	329.52/2941.22
	EEMB	-/-	-/-	-/-	-/-	-/-
	SLL	-	-	-	-	-
	S^{2}TMB	-	-	-	-	-

Table 11: Comparison of MB methods on real-world data sets in prediction accuracy (2)

Data	Algorithm	NB	KNN	SVM	Compactness	Time
gisette	GSMB	0.63/0.58	0.60/0.55	0.63/0.57	3.00/3.00	0.07/0.07
	IAMB	0.84/0.84	0.91/0.91	0.91/0.91	4.00/4.00	5.94/5.86
	Inter-IAMB	0.84/0.84	0.91/0.91	0.91/0.91	4.00/4.00	5.82/5.85
	Fast-IAMB	0.84/0.84	0.83/0.83	0.84/0.84	3.00/3.00	1.25/1.25
	LRH	0.86/0.86	0.90/0.90	0.91/0.91	6.00/6.00	294.35/460.95
	FBED	0.85/0.84	0.89/0.91	0.90/0.91	4.00/4.00	3.52/3.69
	MMMB	0.90/0.89	0.97/0.96	0.75/0.75	255.10/334.40	332.94/601.17
	PCMB	0.86/0.87	0.93/0.95	0.87/0.77	46.60/113.90	1335.92/3115.83
	HITON-MB	0.89/0.88	0.97/0.97	0.75/0.75	361.30/507.50	496.91/1035.98
	Semi-HITON-MB	0.90/0.89	0.97/0.97	0.75/0.73	315.80/433.50	546.27/1117.32
	MBOR	-/-	-/-	-/-	-/-	-/-
	IPCMB	0.86/0.86	0.94/0.96	0.83/0.73	80.90/192.30	2904.15/6057.50
	STMB	-/-	-/-	-/-	-/-	-/-
	BAMB	-/-	-/-	-/-	-/-	-/-
	EEMB	-/-	-/-	-/-	-/-	-/-
	SLL	-	-	-	-	-
	S^{2}TMB	-	-	-	-	-
bankrupty	GSMB	0.89/0.89	0.85/0.88	0.89/0.89	7.20/6.00	0.02/0.01
	IAMB	0.89/0.89	0.89/0.89	0.90/0.90	10.00/10.00	0.30/0.30
	Inter-IAMB	0.89/0.89	0.89/0.89	0.90/0.90	10.00/10.00	0.30/0.31
	Fast-IAMB	0.89/0.89	0.86/0.86	0.89/0.89	9.00/9.00	0.08/0.09
	LRH	0.89/0.89	0.88/0.88	0.89/0.89	9.70/9.70	4.28/6.26
	FBED	0.89/0.89	0.89/0.89	0.90/0.90	10.00/10.00	0.16/0.18
	MMMB	0.84/0.82	0.88/0.88	0.90/0.89	60.70/76.90	32.53/79.24
	PCMB	0.86/0.85	0.88/0.89	0.89/0.89	38.90/54.40	99.06/225.14
	HITON-MB	0.84/0.82	0.88/0.88	0.90/0.89	58.30/73.70	54.81/132.83
	Semi-HITON-MB	0.84/0.83	0.88/0.88	0.90/0.89	55.80/72.30	58.64/161.94
	MBOR	0.87/0.85	0.88/0.88	0.90/0.90	32.80/39.90	67.79/80.22
	IPCMB	0.86/0.85	0.88/0.88	0.89/0.89	39.70/56.00	76.54/243.60
	STMB	0.80/0.80	0.88/0.88	0.89/0.89	89.70/113.80	125.16/488.26
	BAMB	0.85/0.84	0.89/0.89	0.90/0.90	42.60/52.80	431.61/1523.60
	EEMB	0.85/0.85	0.89/0.89	0.90/0.90	37.00/44.90	217.67/647.90
	SLL	0.88	0.84	0.89	12.22	11737.17
	S^{2}TMB	0.89	0.83	0.89	9.1	1958.75
infant	GSMB	0.95/0.95	0.95/0.95	0.96/0.95	4.80/3.40	0.02/0.00
	IAMB	0.95/0.95	0.95/0.95	0.96/0.96	5.00/5.00	0.06/0.06
	Inter-IAMB	0.95/0.95	0.95/0.95	0.96/0.96	5.00/5.00	0.06/0.06
	Fast-IAMB	0.95/0.95	0.94/0.94	0.96/0.96	4.30/4.30	0.02/0.02
	LRH	0.95/0.95	0.95/0.95	0.96/0.96	6.00/6.60	0.10/0.15
	FBED	0.95/0.95	0.95/0.95	0.96/0.96	5.00/5.00	0.02/0.02
	MMMB	0.95/0.95	0.95/0.95	0.96/0.96	6.90/11.30	0.12/0.24
	PCMB	0.95/0.95	0.95/0.95	0.96/0.96	5.10/7.80	0.41/0.93
	HITON-MB	0.95/0.95	0.95/0.95	0.96/0.96	6.40/11.30	0.14/0.33
	Semi-HITON-MB	0.95/0.95	0.95/0.95	0.96/0.96	6.50/11.30	0.18/0.47
	MBOR	0.95/0.95	0.95/0.95	0.96/0.96	7.10/8.30	0.30/0.58
	IPCMB	0.95/0.95	0.94/0.95	0.96/0.96	3.50/7.70	0.26/0.56
	STMB	0.95/0.94	0.95/0.95	0.96/0.95	14.60/28.00	0.13/0.48
	BAMB	0.95/0.95	0.95/0.95	0.96/0.96	5.20/9.20	0.17/0.64
	EEMB	0.95/0.95	0.95/0.95	0.96/0.96	5.20/8.30	0.09/0.28
	SLL	0.96	0.94	0.96	5.17	3087.95
	S^{2}TMB	0.95	0.94	0.95	6.5	94.77
spambase	GSMB	0.79/0.78	0.78/0.78	0.80/0.80	8.50/8.90	0.00/0.00
	IAMB	0.90/0.90	0.91/0.91	0.91/0.91	9.00/9.00	0.06/0.06
	Inter-IAMB	0.90/0.90	0.91/0.91	0.91/0.91	9.00/9.00	0.06/0.06
	Fast-IAMB	0.88/0.88	0.90/0.90	0.89/0.89	8.00/8.00	0.01/0.01
	LRH	0.90/0.90	0.91/0.90	0.91/0.91	8.20/8.90	0.88/1.12
	FBED	0.90/0.90	0.91/0.91	0.91/0.91	9.00/9.00	0.04/0.05
	MMMB	0.88/0.88	0.93/0.93	0.93/0.93	44.10/49.80	12.30/23.11
	PCMB	0.88/0.88	0.93/0.93	0.93/0.93	41.30/47.80	58.64/90.65
	HITON-MB	0.88/0.88	0.93/0.93	0.93/0.93	43.10/49.10	27.06/52.15
	Semi-HITON-MB	0.88/0.88	0.93/0.93	0.93/0.93	42.90/49.00	34.29/65.51
	MBOR	0.89/0.89	0.92/0.93	0.93/0.93	37.40/42.90	25.74/29.35
	IPCMB	0.88/0.88	0.92/0.93	0.93/0.93	42.00/47.80	66.02/111.47
	STMB	0.88/0.89	0.93/0.93	0.93/0.93	50.00/52.90	67.55/147.72
	BAMB	0.89/0.89	0.92/0.93	0.93/0.93	35.40/43.30	147.33/432.71
	EEMB	0.90/0.90	0.92/0.93	0.93/0.93	31.60/38.00	58.51/160.65
	SLL	-	-	-	-	-
	S^{2}TMB	-	-	-	-	-

45

Table 12: Comparison of the PC methods on real-world data sets in prediction accuracy

Data	Algorithm	NB	KNN	SVM	Compactness	Time
madelon	PC-simple	0.60/0.60	0.53/0.49	0.59/0.60	3.60/4.70	0.03/0.08
	MMPC	0.58/0.59	0.52/0.57	0.60/0.60	4.70/6.10	0.03/0.04
	HITON-PC	0.59/0.59	0.52/0.54	0.59/0.60	4.70/5.90	0.03/0.05
	Semi-HITON-PC	0.59/0.59	0.51/0.54	0.59/0.60	4.60/5.50	0.03/0.05
	GetPC	0.56/0.58	0.50/0.51	0.56/0.58	1.50/2.70	0.15/0.23
	MBtoPC	0.60/0.59	0.52/0.52	0.60/0.60	3.40/4.50	0.24/0.43
	SLL-PC	0.57	0.52	0.57	5.1	22.94
	S^{2}TMB-PC	0.58	0.52	0.57	5.1	61.63
arcene	PC-simple	0.61/0.63	0.61/0.62	0.60/0.61	1.50/3.60	0.25/0.37
	MMPC	0.78/0.80	0.66/0.75	0.79/0.77	3.80/5.30	0.34/0.40
	HITON-PC	0.74/0.72	0.67/0.68	0.69/0.71	3.70/5.60	0.34/0.40
	Semi-HITON-PC	0.69/0.73	0.62/0.61	0.66/0.69	3.50/4.90	0.34/0.40
	GetPC	0.61/0.62	0.60/0.60	0.61/0.63	1.70/2.00	1.66/2.42
	MBtoPC	0.66/0.69	0.68/0.65	0.71/0.69	2.70/3.20	0.12/0.61
	SLL-PC	-	-	-	-	-
	S^{2}TMB-PC	0.74	0.68	0.73	5.8	1633.66
dexter	PC-simple	0.82/0.86	0.71/0.84	0.82/0.85	6.00/10.90	1.01/2.06
	MMPC	0.86/0.87	0.81/0.87	0.86/0.86	8.20/13.40	0.49/0.63
	HITON-PC	0.85/0.86	0.79/0.87	0.85/0.86	9.00/13.40	0.49/0.61
	Semi-HITON-PC	0.85/0.86	0.78/0.85	0.85/0.86	7.80/13.00	0.49/0.64
	GetPC	0.81/0.85	0.72/0.82	0.81/0.84	6.20/10.60	4.16/7.18
	MBtoPC	0.80/0.79	0.75/0.75	0.82/0.81	5.10/5.50	4.10/9.95
	SLL-PC	-	-	-	-	-
	S^{2}TMB-PC	-	-	-	-	-
dorothea	PC-simple	0.73/0.93	0.73/0.92	0.73/0.93	1.30/2.20	36.72/249.97
	MMPC	0.93/0.93	0.92/0.91	0.93/0.93	8.60/15.30	28.47/46.96
	HITON-PC	0.93/0.93	0.91/0.91	0.93/0.93	9.90/17.80	28.09/42.30
	Semi-HITON-PC	0.94/0.94	0.92/0.92	0.93/0.93	8.80/13.90	28.04/39.01
	GetPC	0.54/0.92	0.54/0.92	0.54/0.92	0.80/2.20	266.76/615.19
	MBtoPC	0.94/0.93	0.92/0.92	0.93/0.94	6.40/7.60	165.69/1726.01
	SLL-PC	-	-	-	-	-
	S^{2}TMB-PC	-	-	-	-	-
gisette	PC-simple	0.89/0.90	0.93/0.94	0.90/0.87	22.30/39.60	1154.45/3168.57
	MMPC	0.91/0.91	0.96/0.96	0.84/0.82	53.70/73.90	67.47/169.92
	HITON-PC	0.91/0.91	0.96/0.97	0.83/0.80	48.80/67.90	121.95/329.02
	Semi-HITON-PC	0.91/0.90	0.96/0.96	0.84/0.82	46.50/63.60	135.25/372.99
	GetPC	0.87/0.88	0.89/0.91	0.90/0.88	11.50/22.70	291.91/640.68
	MBtoPC	-/-	-/-	-/-	-/-	-/-
	SLL-PC	-	-	-	-	-
	S^{2}TMB-PC	-	-	-	-	-
bankrupty	PC-simple	0.88/0.87	0.86/0.87	0.89/0.89	23.30/31.30	29.92/92.30
	MMPC	0.87/0.87	0.87/0.87	0.89/0.89	30.00/37.40	12.56/30.69
	HITON-PC	0.87/0.87	0.87/0.88	0.89/0.89	28.50/36.00	22.43/56.45
	Semi-HITON-PC	0.88/0.87	0.87/0.87	0.89/0.89	27.00/35.00	26.36/72.68
	GetPC	0.89/0.88	0.86/0.87	0.89/0.89	15.40/24.30	56.52/163.50
	MBtoPC	0.89/0.89	0.88/0.89	0.89/0.90	8.20/9.90	4.05/5.17
	SLL-PC	0.89	0.84	0.89	9.3	2668.09
	S^{2}TMB-PC	0.89	0.84	0.89	8.9	1984.47
infant	PC-simple	0.95/0.95	0.95/0.95	0.96/0.96	3.80/6.70	0.06/0.14
	MMPC	0.95/0.95	0.95/0.95	0.96/0.96	5.00/7.90	0.02/0.06
	HITON-PC	0.95/0.95	0.95/0.95	0.96/0.96	5.00/8.10	0.03/0.10
	Semi-HITON-PC	0.95/0.95	0.95/0.95	0.96/0.96	5.00/7.90	0.04/0.15
	GetPC	0.95/0.95	0.95/0.95	0.96/0.96	4.00/6.40	0.11/0.25
	MBtoPC	0.95/0.95	0.95/0.95	0.96/0.96	5.00/5.00	0.14/0.19
	SLL-PC	0.95	0.94	0.95	5.4	2653.88
	S^{2}TMB-PC	0.95	0.94	0.95	5.9	94.8
spambase	PC-simple	0.90/0.90	0.92/0.92	0.93/0.93	21.10/26.80	19.17/26.59
	MMPC	0.91/0.91	0.92/0.92	0.93/0.93	23.30/28.60	2.06/4.08
	HITON-PC	0.90/0.91	0.92/0.92	0.93/0.93	23.30/28.10	5.02/9.97
	Semi-HITON-PC	0.91/0.91	0.92/0.92	0.93/0.93	23.20/28.10	6.18/12.78
	GetPC	0.90/0.91	0.92/0.92	0.93/0.93	21.90/27.10	25.86/49.17
	MBtoPC	0.90/0.90	0.90/0.91	0.91/0.91	9.30/9.70	0.99/1.02
	SLL-PC	-	-	-	-	-
	S^{2}TMB-PC	-	-	-	-	-

Table 13: Comparison of MB learning methods on class-imbalanced real-world data in AUC

Data	Algorithm	NB-AUC	KNN-AUC	SVM-AUC
dorothea	GSMB	0.73/0.50	0.68/0.51	0.73/0.50
	IAMB	0.78/0.78	0.78/0.78	0.82/0.82
	Inter-IAMB	0.78/0.79	0.77/0.79	0.82/0.82
	Fast-IAMB	0.76/0.76	0.74/0.67	0.78/0.76
	LRH	0.79/0.78	0.75/0.74	0.78/0.79
	FBED	0.79/0.79	0.78/0.76	0.82/0.82
	MMMB	0.75/0.73	0.73/0.72	0.79/0.77
	PCMB	0.28/0.68	0.28/0.68	0.28/0.69
	HITON-MB	0.74/0.74	0.73/0.73	0.77/0.71
	Semi-HITON-MB	0.78/0.74	0.75/0.72	0.79/0.72
	MBOR	0.73/0.78	0.72/0.71	0.78/0.75
	IPCMB	0.23/0.59	0.23/0.57	0.23/0.60
	STMB	-/-	-/-	-/-
	BAMB	0.72/0.73	0.68/0.71	0.77/0.74
	EEMB	-/-	-/-	-/-
	SLL	-	-	-
	S^2TMB	-	-	-
bankrupty	GSMB	0.50/0.50	0.51/0.50	0.50/0.50
	IAMB	0.55/0.55	0.70/0.70	0.59/0.59
	Inter-IAMB	0.55/0.55	0.70/0.70	0.59/0.59
	Fast-IAMB	0.50/0.50	0.73/0.70	0.50/0.50
	LRH	0.56/0.56	0.69/0.69	0.52/0.53
	FBED	0.54/0.55	0.69/0.70	0.59/0.59
	MMMB	0.78/0.78	0.66/0.64	0.56/0.54
	PCMB	0.74/0.77	0.64/0.66	0.52/0.53
	HITON-MB	0.78/0.78	0.65/0.65	0.56/0.54
	Semi-HITON-MB	0.78/0.78	0.65/0.65	0.56/0.54
	MBOR	0.74/0.76	0.65/0.65	0.56/0.57
	IPCMB	0.75/0.78	0.65/0.66	0.54/0.54
	STMB	0.78/0.78	0.63/0.63	0.53/0.51
	BAMB	0.78/0.79	0.68/0.68	0.58/0.57
	EEMB	0.77/0.78	0.67/0.67	0.57/0.57
	SLL	0.62	0.66	0.50
	S^2TMB	0.51	0.73	0.5
infant	GSMB	0.72/0.65	0.59/0.58	0.68/0.65
	IAMB	0.74/0.74	0.66/0.66	0.69/0.69
	Inter-IAMB	0.74/0.74	0.66/0.66	0.69/0.69
	Fast-IAMB	0.72/0.72	0.53/0.53	0.68/0.68
	LRH	0.74/0.73	0.66/0.65	0.69/0.68
	FBED	0.74/0.74	0.66/0.66	0.69/0.69
	MMMB	0.74/0.74	0.67/0.65	0.69/0.69
	PCMB	0.73/0.73	0.64/0.65	0.69/0.69
	HITON-MB	0.73/0.74	0.66/0.67	0.69/0.70
	Semi-HITON-MB	0.73/0.74	0.66/0.66	0.69/0.69
	MBOR	0.74/0.74	0.67/0.67	0.69/0.69
	IPCMB	0.72/0.73	0.55/0.63	0.68/0.69
	STMB	0.73/0.74	0.66/0.67	0.69/0.67
	BAMB	0.74/0.74	0.67/0.67	0.69/0.69
	EEMB	0.73/0.74	0.65/0.66	0.69/0.69
	SLL	0.71	0.50	0.71
	S^2TMB	0.71	0.50	0.71

47

Table 14: Comparison of PC learning methods on class-imbalanced real-world data in AUC

Data	Algorithm	NB-AUC	KNN-AUC	SVM-AUC
dorothea	PC-simple	0.51/0.71	0.51/0.65	0.51/0.72
	MMPC	0.75/0.73	0.72/0.72	0.79/0.76
	HITON-PC	0.74/0.73	0.72/0.73	0.77/0.73
	Semi-HITON-PC	0.78/0.74	0.75/0.72	0.79/0.74
	GetPC	0.33/0.68	0.33/0.67	0.33/0.69
	MBtoPC	0.77/0.78	0.72/0.73	0.81/0.82
	SLL-PC	-	-	-
	S^{2}TMB-PC	-	-	-
bankrupty	PC-simple	0.72/0.74	0.64/0.67	0.50/0.51
	MMPC	0.76/0.77	0.67/0.67	0.50/0.51
	HITON-PC	0.74/0.76	0.67/0.68	0.50/0.51
	Semi-HITON-PC	0.74/0.77	0.66/0.67	0.50/0.51
	GetPC	0.63/0.70	0.65/0.65	0.50/0.51
	MBtoPC	0.54/0.55	0.67/0.69	0.50/0.59
	SLL-PC	0.54	0.70	0.50
	S^{2}TMB-PC	0.51	0.74	0.50
infant	PC-simple	0.73/0.74	0.61/0.65	0.69/0.69
	MMPC	0.73/0.74	0.66/0.66	0.69/0.69
	HITON-PC	0.73/0.74	0.66/0.66	0.69/0.68
	Semi-HITON-PC	0.73/0.74	0.66/0.65	0.69/0.68
	GetPC	0.73/0.73	0.62/0.62	0.69/0.69
	MBtoPC	0.74/0.74	0.66/0.66	0.69/0.69
	SLL-PC	0.71	0.50	0.71
	S^{2}TMB-PC	0.71	0.50	0.71

many real-world data sets are available in streams. There is a need to develop online causality-based feature selection algorithms to deal with streaming data.

- Weak-supervision data. In real-world applications, a data set may have very few labeled data instances, while abundant unlabeled data instances are available. However, existing causality-based feature selection algorithms are unable to work well with such data sets. Thus, it is interesting and challenging to exploit unlabeled data instances to facilitate weak-supervision causality-based feature selection methods.

- Imbalanced class data. The majority of existing causality-based methods cannot deal with data sets with imbalanced classes, which, however, exist in many real-world applications. It is important to develop new feature selection methods to address this problem.

- Distribution shift. Causality-based feature selection has the potential to deal with data with distribution shifts, i.e., training data and testing data do not have identical distributions, since in this case the set of all direct causes of a class variable is the invariance and minimal set between training data and testing data. However, it is an open problem to distinguish causes from effects of a class variable.

References

Silvia Acid, Luis M de Campos, and Javier G Castellano. 2005. Learning Bayesian network classifiers: Searching in a space of partially directed acyclic graphs. *Machine Learning* 59, 3 (2005), 213–235.

Silvia Acid, Luis M de Campos, and Moisés Fernández. 2013. Score-based methods for learning Markov boundaries by searching in constrained spaces. *Data mining and knowledge discovery* 26, 1 (2013), 174–212.

Alan Agresti and Maria Kateri. 2011. *Categorical data analysis*. Springer.

Hirotugu Akaike. 1974. A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*. Springer, 215–222.

Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. 2010a. Local causal and markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation. *Journal of Machine Learning Research* 11 (2010), 171–234.

Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. 2010b. Local causal and markov blanket induction for causal discovery and feature selection for classification part ii: Analysis and extensions. *Journal of Machine Learning Research* 11, Jan (2010), 235–284.

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.

Susan Athey. 2017. Beyond prediction: Using big data for policy problems. *Science* 355, 6324 (2017), 483–485.

Marco Benjumeda, Concha Bielza, and Pedro Larrañaga. 2019. Learning tractable Bayesian networks in the space of elimination orders. *Artificial Intelligence* 274 (2019), 66–90.

Giorgos Borboudakis and Ioannis Tsamardinos. 2016. Towards robust and versatile causal discovery for business applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1435–1444.

Giorgos Borboudakis and Ioannis Tsamardinos. 2019. Forward-backward selection with early dropping. *The Journal of Machine Learning Research* 20, 1 (2019), 276–314.

Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. 2012. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of machine learning research* 13, Jan (2012), 27–66.

Wray Buntine. 1991. Theory refinement on Bayesian networks. In *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 52–60.

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.

Cassio P de Campos and Qiang Ji. 2011. Efficient structure learning of Bayesian networks using constraints. *Journal of Machine Learning Research* 12, Mar (2011), 663–689.

Luis M de Campos. 2006. A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. *Journal of Machine Learning Research* 7, Oct (2006), 2149–2187.

David Maxwell Chickering. 2002. Optimal structure identification with greedy search. *Journal of machine learning research* 3, Nov (2002), 507–554.

David Maxwell Chickering, David Heckerman, and Christopher Meek. 2004. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research* 5, Oct (2004), 1287–1330.

Gregory F Cooper and Edward Herskovits. 1992. A Bayesian method for the induction of probabilistic networks from data. *Machine learning* 9, 4 (1992), 309–347.

Sergio Rodrigues De Morais and Alex Aussem. 2008. A novel scalable and data efficient feature subset selection algorithm. In *ECML PKDD'08*. Springer, 298–312.

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.

Tian Gao, Kshitij Fadnis, and Murray Campbell. 2017. Local-to-global Bayesian network structure learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1193–1202.

Tian Gao and Qiang Ji. 2015. Local causal discovery of direct causes and effects. In *Advances in Neural Information Processing Systems*. 2512–2520.

Tian Gao and Qiang Ji. 2016. Constrained Local Latent Variable Discovery.. In *IJCAI*. 1490–1496.

Tian Gao and Qiang Ji. 2017a. Efficient Markov blanket discovery and its application. *IEEE transactions on cybernetics* 47, 5 (2017), 1169–1179.

Tian Gao and Qiang Ji. 2017b. Efficient score-based Markov Blanket discovery. *International Journal of Approximate Reasoning* 80 (2017), 277–293.

Tian Gao and Dennis Wei. 2018. Parallel Bayesian network structure learning. In *International Conference on Machine Learning*. 1671–1680.

Asish Ghoshal and Jean Honorio. 2017. Learning identifiable gaussian bayesian networks in polynomial time and sample complexity. In *Advances in Neural Information Processing Systems*. 6457–6466.

Clark Glymour, Kun Zhang, and Peter Spirtes. 2019. Review of Causal Discovery Methods Based on Graphical Models. *Frontiers in Genetics* 10 (2019).

Ruocheng Guo, Lu Cheng, Jundong Li, P Richard Hahn, and Huan Liu. 2018. A Survey of Learning Causality with Data: Problems and Methods. *arXiv: Artificial Intelligence* (2018).

Isabelle Guyon, Constantin Aliferis, and others. 2007. Causal feature selection. In *Computational methods of feature selection*. Chapman and Hall/CRC, 75–97.

Isabelle Guyon and Andre Elisseeff. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research* 3 (2003), 1157–1182.

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

Jake M Hofman, Amit Sharma, and Duncan J Watts. 2017. Prediction and explanation in social systems. *Science* 355, 6324 (2017), 486–488.

Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H Maathuis, Peter Bühlmann, and others. 2012. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software* 47, 11 (2012), 1–26.

Ron Kohavi and George H John. 1997. Wrappers for feature subset selection. *Artificial intelligence* 97, 1-2 (1997), 273–324.

Mikko Koivisto and Kismat Sood. 2004. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research* 5, May (2004), 549–573.

Daphne Koller and Mehran Sahami. 1996. Toward optimal feature selection. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 284–292.

Wai Lam and Fahiem Bacchus. 1994. Learning Bayesian belief networks: An approach based on the MDL principle. *Computational intelligence* 10, 3 (1994), 269–293.

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. 2017. Feature Selection: A Data Perspective. *Comput. Surveys* 50, 6 (2017), 94.

Jundong Li and Huan Liu. 2017. Challenges of Feature Selection for Big Data Analytics. *IEEE Intelligent Systems* 32, 2 (2017), 9–15.

Jiuyong Li, Lin Liu, and Thuc Duy Le. 2015. *Practical approaches to causal relationship exploration*. Springer.

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 Transactions on Intelligent Systems and Technology,in press* (2019).

Xuqing Liu and Xinsheng Liu. 2016. Swamping and masking in Markov boundary discovery. *Machine Learning* 104, 1 (2016), 25–54.

Xu-Qing Liu and Xin-Sheng Liu. 2018. Markov blanket and Markov boundary of multiple variables. *The Journal of Machine Learning Research* 19, 1 (2018), 1658–1707.

Sara Magliacane, Thijs van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M Mooij. 2018. Domain Adaptation by Using Causal Inference to Predict Invariant Conditional Distributions. In *Advances in Neural Information Processing Systems*. 10846–10856.

Dimitris Margaritis. 2009. Toward provably correct feature selection in arbitrary domains. In *Advances in neural information processing systems*. 1240–1248.

Dimitris Margaritis and Sebastian Thrun. 2000. Bayesian network induction via local neighborhoods. In *Advances in neural information processing systems*. 505–511.

Andrés R Masegosa and Serafín Moral. 2012. A Bayesian stochastic search method for discovering Markov boundaries. *Knowledge-Based Systems* 35 (2012), 211–223.

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

Kevin Murphy and others. 2001. The bayes net toolbox for matlab. *Computing science and statistics* 33, 2 (2001), 1024–1034.

Siqi Nie, Cassio P de Campos, and Qiang Ji. 2017. Efficient learning of Bayesian networks with bounded tree-width. *International Journal of Approximate Reasoning* 80 (2017), 412–427.

T Niinimki and Pekka Parviainen. 2012. Local structure discovery in bayesian networks. In *Proceedings of UAI'12, Workshop on Causal Structure Learning, 2012*. 634–643.

Judea Pearl. 2009. *Causality*. Cambridge university press.

Judea Pearl. 2014. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

Jean-Philippe Pellet and André Elisseeff. 2008. Using Markov blankets for causal structure learning. *Journal of Machine Learning Research* 9, Jul (2008), 1295–1342.

Jose M Peña. 2008. Learning gaussian graphical models of gene networks with false discovery rate control. In *European conference on evolutionary computation, machine learning and data mining in bioinformatics*. Springer, 165–176.

Jose M Peña, Johan Björkegren, and Jesper Tegnér. 2005. Scalable, efficient and correct learning of Markov boundaries under the faithfulness assumption. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*. Springer, 136–147.

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.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. 2016. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 78, 5 (2016), 947–1012.

Adam Pocock, Mikel Luján, and Gavin Brown. 2012. Informative priors for markov blanket discovery. In *Artificial Intelligence and Statistics*. 905–913.

Marco Tulio De Freitas Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *knowledge discovery and data mining* (2016), 1135–1144.

Thomas Richardson, Peter Spirtes, and others. 2002. Ancestral graph Markov models. *The Annals of Statistics* 30, 4 (2002), 962–1030.

Raanan Y Rohekar, Yaniv Gurwicz, Shami Nisimov, Guy Koren, and Gal Novik. 2018. Bayesian Structure Learning by Recursive Bootstrap. In *Advances in Neural Information Processing Systems*. 10525–10535.

M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters. 2018. Invariant Models for Causal Transfer Learning. *Journal of Machine Learning Research* 19, 36 (2018), 1–34.

Yvan Saeys, Inaki Inza, and Pedro Larranaga. 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics* 23, 19 (2007), 2507–2517.

Richard Scheines, Peter Spirtes, Clark Glymour, Christopher Meek, and Thomas Richardson. 1998. The TETRAD project: Constraint based aids to causal model specification. *Multivariate Behavioral Research* 33, 1 (1998), 65–117.

Gideon Schwarz and others. 1978. Estimating the dimension of a model. *The annals of statistics* 6, 2 (1978), 461–464.

Marco Scutari. 2009. Learning Bayesian networks with the bnlearn R package. *arXiv preprint arXiv:0908.3817* (2009).

Konstantinos Sechidis and Gavin Brown. 2015. Markov blanket discovery in positive-unlabelled and semi-supervised data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 351–366.

Konstantinos Sechidis and Gavin Brown. 2018. Simple strategies for semi-supervised feature selection. *Machine Learning* 107, 2 (2018), 357–395.

Ricardo Silva and Zoubin Ghahramani. 2009. The hidden life of latent variables: Bayesian learning with mixed graph models. *Journal of Machine Learning Research* 10, Jun (2009), 1187–1238.

Peter Spirtes, Clark N Glymour, Richard Scheines, David Heckerman, Christopher Meek, Gregory Cooper, and Thomas Richardson. 2000. *Causation, prediction, and search*. MIT press.

Alexander Statnikov, Nikita I Lytkin, Jan Lemeire, and Constantin F Aliferis. 2013. Algorithms for discovery of multiple Markov boundaries. *Journal of Machine Learning Research* 14, Feb (2013), 499–566.

Alexander Statnikov, Sisi Ma, Mikael Henaff, Nikita Lytkin, Efstratios Efstathiadis, Eric R Peskin, and Constantin F Aliferis. 2015. Ultra-scalable and efficient methods for hybrid observational and experimental local causal pathway discovery. *The Journal of Machine Learning Research* 16, 1 (2015), 3219–3267.

Alexander Statnikov, Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. 2010. Causal explorer: A matlab library of algorithms for causal discovery and variable selection for classification. *Challenges in machine learning* 2 (2010), 267–278.

Ioannis Tsamardinos and Constantin Aliferis. 2003. Towards Principled Feature Selection: Relevancy, Filters and Wrappers. In *in Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*. Citeseer.

Ioannis Tsamardinos, Constantin F Aliferis, and Alexander Statnikov. 2003a. 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.

Ioannis Tsamardinos, Constantin F Aliferis, Alexander R Statnikov, and Er Statnikov. 2003b. Algorithms for Large Scale Markov Blanket Discovery.. In *FLAIRS conference*, Vol. 2. 376–380.

Ioannis Tsamardinos, Giorgos Borboudakis, Pavlos Katsogridakis, Polyvios Pratikakis, and Vassilis Christophides. 2019. A greedy feature selection algorithm for Big Data of high dimensionality. *Machine learning* 108, 2 (2019), 149–202.

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.

Changzhang Wang, You Zhou, Qiang Zhao, and Zhi Geng. 2014. Discovering and orienting the edges connected to a target variable in a DAG via a sequential local learning approach. *Computational Statistics & Data Analysis* 77 (2014), 252–266.

De Wang, Danesh Irani, and Calton Pu. 2012. Evolutionary study of web spam: Webb Spam Corpus 2011 versus Webb Spam Corpus 2006. (2012), 40–49.

Hao Wang, Zhaolong Ling, Kui Yu, and Xindong Wu. 2020. Towards efficient and effective discovery of Markov blankets for feature selection. *Information Science* 509 (2020), 227–242.

Xingyu Wu, Bingbing Jiang, Kui Yu, Chunyan Miao, and Huanhuan Chen. in press, 2019. Accurate Markov Boundary Discovery for Causal Feature Selection. *IEEE Transactions on Cybernetics* (in press, 2019).

Xindong Wu, Kui Yu, Wei Ding, Hao Wang, and Xingquan Zhu. 2013. Online Feature Selection with Streaming Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 5 (2013), 1178–1192.

Sandeep Yaramakala and Dimitris Margaritis. 2005. Speculative Markov blanket discovery for optimal feature selection. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*. IEEE, 4–pp.

Raanan Yehezkel and Boaz Lerner. 2009. Bayesian network structure learning by recursive autonomy identification. *Journal of Machine Learning Research* 10, Jul (2009), 1527–1570.

Jianxin Yin, You Zhou, Changzhang Wang, Ping He, Cheng Zheng, and Zhi Geng. 2008. Partial orientation and local structural learning of causal networks for prediction. In *Causation and Prediction Challenge.* 93–105.

Kui Yu, Lin Liu, and Jiuyong Li. 2018a. Discovering Markov Blanket from Multiple interventional Datasets. *arXiv preprint arXiv:1801.08295* (2018).

Kui Yu, Lin Liu, and Jiuyong Li. 2018b. A Unified View of Causal and Non-causal Feature Selection. *arXiv preprint arXiv:1802.05844* (2018).

Kui Yu, Lin Liu, Jiuyong Li, and Huanhuan Chen. 2018c. Mining Markov Blankets Without Causal Sufficiency. *IEEE transactions on neural networks and learning systems* 99 (2018), 1–15.

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).

Kui Yu, Xindong Wu, Wei Ding, Yang Mu, and Hao Wang. 2017. Markov blanket feature selection using representative sets. *IEEE transactions on neural networks and learning systems* 28, 11 (2017), 2775–2788.

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* 11, 2 (2016), 16.

Yiteng Zhai, Yewsoon Ong, and Ivor W Tsang. 2014. The Emerging Big Dimensionality. *IEEE Computational Intelligence Magazine* 9, 3 (2014), 14–26.

Kun Zhang, Mingming Gong, and Bernhard Schölkopf. 2015. Multi-source domain adaptation: A causal view. In *Twenty-ninth AAAI conference on artificial intelligence.*

Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2012. Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775* (2012).

Kun Zhang, Bernhard Schölkopf, Peter Spirtes, and Clark Glymour. 2017. Learning causality and causality-related learning: some recent progress. *National science review* 5, 1 (2017), 26–29.

Jing Zhou, Dean P Foster, Robert A Stine, and Lyle H Ungar. 2006. Streamwise feature selection. *Journal of Machine Learning Research* 7, Sep (2006), 1861–1885.