Causality-based Feature Selection: Methods and Evaluations

KUI YU and XIANJIE GUO, Hefei University of Technology, China
LIN LIU and JIUYONG LI, University of South Australia, Australia
HAO WANG and ZHAOLONG LING, Hefei University of Technology, China
XINDONG WU, Mininglamp Technology, China

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 article, 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 datasets. Finally, we discuss some challenging problems to be tackled in future research.

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

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

ACM Reference format:
Kui Yu, XianJie Guo, Lin Liu, Jiuyong Li, Hao Wang, Zhaolong Ling, and Xindong Wu. 2020. Causality-based Feature Selection: Methods and Evaluations. *ACM Comput. Surv.* 53, 5, Article 111 (September 2020), 36 pages.
https://doi.org/10.1145/3409382

This work is partially supported by the National Key Research and Development Program of China (under grant 2016YFB1000901), Australian Research Council Discovery Projects (under grant DP170101306), and the National Science Foundation of China (under grant 61876206).

Authors' addresses: K. Yu and X. Guo, Key Laboratory of Knowledge Engineering with Big Data of Ministry of Education (Hefei University of Technology) and School of Computer and Information, Hefei University of Technology, Hefei, 230601, China; emails: {yukui, xianjieguo}@hfut.edu.cn; L. Liu and J. Li, UniSA STEM, University of South Australia, Adelaide, 5095, Australia; emails: {Lin.Liu, Jiuyong.Li}@unisa.edu.au; H. Wang and Z. Ling, School of Computer and Information, Hefei University of Technology, Hefei, 230601, China; emails: jsjxwangh@hfut.edu.cn, z_dragonl@163.com; X. Wu, Mininglamp Academy of Sciences, Mininglamp Technology, Beijing, 100084, China; email: wuxindong@mininglamp.com.

0360-0300/2020/09-ART111 $15.00
https://doi.org/10.1145/3409382

1 INTRODUCTION

Feature selection plays an essential role in high-dimensional data analytics [13, 37, 47, 101] 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 a class variable of interest. For example, gene (i.e., feature) selection can identify a small number of informative genes from a high-dimensional gene dataset 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 dataset with high-dimensionality has become ubiquitous in various applications [110]. In the previous example, a gene expression dataset may easily have more than 10K predictive features [80]. For another example, the Web Spam Corpus 2011 collected approximately 16M predictive features for malicious web detection [98]. Almost all machine learning methods may not directly work on datasets of such high dimensionality without feature selection. As a result, in the past 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 [47].

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 article, we focus on causality-based feature selection, an emerging successful type of filter method. 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 [43]. A classical feature selection method aims to find a subset of relevant features based on the correlations between (predictive) features and the class variable [37]. 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 [5, 36]:

- Causal features can improve the explanatory capability of predictive models [66]. Correlations capture only the co-occurrence of features and the class variable. Hence, the selected features often do not provide a convincing explanation for predictions. For example, a strong correlation between *shoe size* (of a child in an elementary school (grades 1–5)) and *reading ability* (of the child) may be found, making *shoe size* a good predictive feature of *reading ability* of an elementary school child. However, clearly *shoe size* is not a reasonable explanation at all for *reading ability*. In fact, the causes of *reading ability*, such as *age*, is more explainable than *shoe size*.
- Causal features can improve the robustness of predictive models [9, 82]. 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 predict *reading ability* of a child in an elementary school using historical data. Based on the historical data, a predictive model built using non-causal features such as *shoe size* may not produce good predictions for a student in senior high school. In contrast, if the causes of *reading ability* of students (such as *age*) were selected as the predictive features, a model built on the historical data will be robust.

In recent years, causality-based feature selection methods have been developed to identify potential causal features using the Bayesian network (BN) and Markov boundary (MB) theory in both

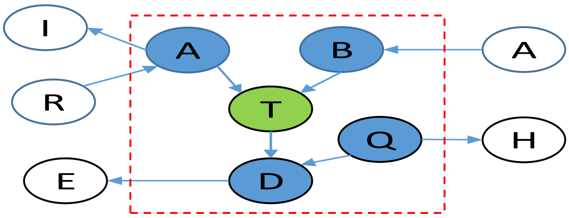

Fig. 1. An example of an MB of variable T (blue variables) in a Bayesian network.

machine learning and causal discovery domains [5, 12, 105]. The structure of a BN is represented by a directed acyclic graph (DAG) where nodes represent variables and edges represent the dependence relationship between the variables [64]. A DAG, or the structure of a BN, can be used to represent causal relationships of variables, when a directed edge $X \rightarrow Y$ is interpreted as a direct cause (X) and effect (Y) relationship [64]. In this case the BN is called a causal BN. The notion of MB was proposed in the context of a BN. If a BN satisfies the faithfulness assumption (see Definition 2.1 in Section 2.1), the MB of a variable in the BN is unique and consists of the parents (direct causes), children (direct effects), and spouses (i.e., other parents of the class variable's children) of the class variable [64]. Figure 1 gives an example of an MB in a BN. The MB of T includes A and B (parents), D (child), and Q (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 in its MB. Most importantly, all other features are probabilistically independent of the class variable conditioning on its MB [92]. Therefore, 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. Accordingly, we can learn a BN structure and then read off the MB of the class variable in the learnt structure for causality-based feature selection. In the past decades, many BN structure learning algorithms have been proposed [19, 87, 112]. However, learning a global BN structure is often computationally expensive especially with a high-dimensional dataset. Although some recently developed algorithms, such as the fGES algorithm, can learn a global BN structure with very high-dimensional data [76], in fact, it is not necessary and wasteful to find an entire BN structure when we are only interested in the MB of a variable of interest.

Thus, there is a need to develop causality-based feature selection methods that only focus on identifying the MB of a variable or a subset of the MB such as parents and children (PC) without learning an entire BN structure involving all features in a dataset. In the past decade, by combining BN structure learning methods with the MB theory, many causality-based feature selection algorithms have been proposed [5, 12]. The developed causality-based feature selection algorithms are divided into constraint-based methods and score-based methods, and they provide a new and complementary algorithmic methodology to enrich feature selection, especially for achieving explainable and robust machine learning.

To advance the research in causality-based feature selection, a comprehensive review of the state-of-the-art techniques in this area is needed. However, so far, there has not been such a review available. Aliferis et al. [5, 6] proposed a general local learning framework for causality-based feature selection, which are focused on three specific causality-based feature selection algorithms and their extensions to the BN structure learning, but the work is not a survey paper. Guyon et al. [36] 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. Yu et al. [105] theoretically analyzed the link between causality-based feature selection and classical feature selection in four levels: learning objectives, assumptions and optimization, search strategies, and practical implications, instead of presenting an extensive review of causality-based feature selection algorithms. There have been some recent reviews on causal inference, such as References [33, 35, 59, 112], but they mainly focused on the advances on learning causal relations between features. And almost all reviews regarding feature selection focused on classical feature selection methods in the past decades [13, 37, 47]. In summary, so far there is little work on a comprehensive review of causality-based feature selection algorithms.

Thus, in this article, we extensively review existing causality-based feature selection methods. Since identifying the causes of a class variable is crucial for robust predications where the training data and testing data have different distributions and almost all existing causality-based feature selection methods do not distinguish causes from effects, we also discuss some representative methods of distinguishing causes from effects. To our knowledge, this is the first attempt on presenting an extensive survey of causality-based feature selection and its recent advances.

In addition, there is no any open-source toolbox/package that 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 article, we develop the first comprehensive open-source package written in C language that implements the representative and state-of-the-art causality-based feature selection algorithms.

Finally, we conduct a comprehensively empirical evaluation on representative causality-based feature selection algorithms and classical feature selection methods using both synthetic and real-world datasets.

The rest of the article 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. Section 6 presents the open-source package. Section 7 reports the evaluation results. Section 8 concludes the article 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, BN, and causality-based feature selection, then we discuss the general strategy of existing causality-based feature selection methods.

2.1 Bayesian Network, Markov Boundary, and Causality-based Feature Selection

Let C be a class variable and $F = \{F_1, F_2, \ldots, F_M\}$ 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 $V = F \cup C = \{V_1, V_2, \ldots, V_{M+1}\}$, $V_i = F_i$ $(1 \leq i \leq M)$, 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 [64]. In the following, we introduce the key concepts and assumptions related to BN, Markov blanket, Markov boundary, and causality-based feature selection.

Definition 2.1 (Faithfulness). [64] Given a BN $< V, G, P(V) >$, G is faithful to $P(V)$ if and only if every conditional independence present in $P(V)$ is entailed by G and the Markov condition. $P(V)$ is faithful to G if and only if G is faithful to $P(V)$.

Definition 2.2 (Causal sufficiency). [64, 87] Causal sufficiency assumes that any common cause of two or more variables in V is also in V.

We first present the concepts of Markov blanket and Markov boundary from a statistical perspective. 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 2.3 (Markov Blanket, Mb). [64] 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 $V_i \in V \setminus (Mb(C) \cup C)$, $C \perp\!\!\!\perp V_i | Mb(C)$.

Definition 2.4 (Markov Boundary, MB). [64] 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 $MB(C)$.

From a BN perspective, under the faithfulness assumption, a node's Markov boundary in a BN is unique and it is the same as the node's Markov blanket, as shown in Definition 2.5.

Definition 2.5 (Markov Blanket/Markov Boundary). [64] 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 [64], as shown in Proposition 2.6 below.

Proposition 2.6. *[64] 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.*

Proposition 2.6 illustrates that learning the MB of the class variable is actually a procedure of feature selection [5, 92]. Koller and Sahami [45] were the first to introduce the concept of MBs to feature selection. The work in References [92, 105] 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. Just as we discussed in Section 1, existing causality-based feature selection algorithms aim to learn the MB of the class variable or a subset of the MB (without learning an entire BN structure involving all features in a dataset) [5, 36].

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, 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 uses a backward phase for removing false positives from S [56]. The interleaving forward-backward feature selection (IFBS) strategy (Algorithm 2) performs the forward phase and backward phase alternatively [94]. Specifically, if there are new features added to S at the forward phase, IFBS immediately triggers the backward phase and implements

ALGORITHM 1: Standard Forward-Backward Selection (SFBS)

```
Input: Feature Set $F$ and the class variable $C$
Output: The set of selected $S$
$S = \emptyset$;
//Forward phase: Adding features to $S$
repeat
    Identify the most informative feature $X \in F$ by selection criterion $\Phi$;
    if $\Phi(S \cup X) > \Phi(S)$ then
        $S = S \cup X$ and $F = F \setminus X$;
    end if
until no features in $F$ are added to $S$;
//Backward phase: Removing features from $S$
repeat
    Find the least informative feature $Y \in S$ by selection criterion $\Phi$;
    if $\Phi(S \setminus Y) \geq \Phi(S)$ then
        $S = S \setminus Y$;
    end if
until no features in $S$ are removed
Output $S$
```

ALGORITHM 2: Interleaving Forward-Backward Selection (IFBS)

```
Input: Feature Set $F$ and the class variable $C$
Output: The set of selected $S$
$S = \emptyset$;
repeat
    //Forward phase: Adding features to $S$
    Identify the most informative feature $X \in F$ by selection criterion $\Phi$;
    if $\Phi(S \cup X) > \Phi(S)$ then
        $S = S \cup X$ and $F = F \setminus X$;
        //Backward phase: Removing features from $S$
        repeat
            Find the least informative feature $Y \in S$ by selection criterion $\Phi$;
            if $\Phi(S \setminus Y) \geq \Phi(S)$ then
                $S = S \setminus Y$;
            end if
        until no features in $S$ are not removed
    end if
until no features in $F$ are added to $S$;
Output $S$
```

both phases alternatively. Existing causality-based feature selection algorithms adopt either the SFBS strategy 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.

3 CONSTRAINT-BASED METHODS

In this section, we will discuss the constraint-based 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

Table 1. Representative Constraint-based Methods

Category	Representative algorithm
Simultaneous MB learning (learning PC and spouses simultaneously and do not distinguish PC from spouses)	GSMB [56]
	IAMB [92]
	IAMBnPC [94]
	IAMB-IP [75]
	Fast-IAMB [102]
	Inter-IAMB [94]
	Inter-IAMBnPC [94]
	FBED^K [12]
	PFBP [95]
Divide-and-conquer MB learning (learning PC and spouses separately)	MMMB [93]
	HITON-MB [7]
	Semi-HITON-MB [5]
	PCMB [70]
	IPCMB [26]
	MBOR [23]
	STMB [30]
	CCMB [100]
MB learning with interleaving PC and spouse learning	BAMB [50]
	EEMB [99]
MB learning with relaxed assumptions (e.g., the faithfulness assumption or causal sufficiency assumption)	KIAMB [70]
	TIE* [88]
	SGAI [108]
	LCMB [51]
	WLCMB [51]
	M3B [106]
MB learning with special purpose (e.g., multiple datasets, distribution shift, and weak supervision)	MIMB [104]
	MCFS [107]
	MIAMB and MKIAMB [52]
	BASSUM [15]
	Semi-IAMB [85]

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 3.1 and 3.2 below. Proposition 3.1 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 $V \setminus \{V_i, V_j\}$.

PROPOSITION 3.1. *[87] In a BN, if node V_i is a parent (or a child) of V_j, then $\forall S \subseteq V \setminus \{V_i, V_j\}$, $V_i \not\perp\!\!\!\perp V_j | S$.*

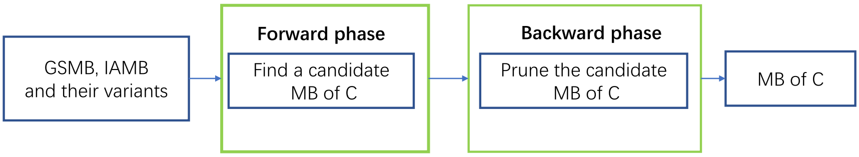

Fig. 2. Simultaneous MB learning.

Proposition 3.2 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, Q is the spouse of T in Figure 1. Q and T are independent (S is an empty set), but they are dependent conditioning on their common child D. Proposition 3.2 shows that Q (spouse) and D (common child) together carry more predictive information about T than D only. Proposition 3.2 also states that spouses of V_k consist of all parents of the children of V_k (excluding V_k).

Proposition 3.2. *[87] 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., $V_i \rightarrow V_j \leftarrow V_k$), if $\exists S \subseteq V \setminus \{V_i, V_j, V_k\}$ such that $V_i \perp\!\!\!\perp V_k | S$ and $V_i \not\!\perp\!\!\!\perp V_k | \{S, V_j\}$ hold, V_i is a spouse of V_k.*

Using the SFBS (or IFBS) strategy, existing constraint-based methods employ the statistical independence tests as the selection criteria, denoted as Φ, to add/remove features to/from S. Given the class variable C, in SFBS (or IFBS), at each iteration, let S be a set of features currently selected, if X ($X \in V \setminus \{C \cup S\}$) and C are conditionally independent conditioning on S (or a subset $S' \subset S$), X does not provide any predictive information to C conditioning on S (i.e., $\Phi(S \cup X) \leq \Phi(S)$). In this case, X is not to be added to S at the forward phase or removed from S at the backward phase.

There are five types of conditional independence tests used by current constraint-based methods, λ^2 test, G^2 test, mutual information for discrete features [58], Fisher's Z test for continuous features with linear relations and additive Gaussian errors [68], and kernel-based tests for continuous features with nonlinearity and non-Gaussian noise [111].

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 does not distinguish PC (parents and children) of C from its spouses during the MB learning. As shown in Figure 2, the simultaneous MB learning approach adopts a forward-backward strategy to greedily learn an 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 [56], IAMB [92], IAMB-nPC [94], Fast-IAMB [102], Inter-IAMB [94], Inter-IAMBnPC [94], IAMB-IP [75], FBED^K [12], and PFBP [95]. The GSMB algorithm was the first algorithm for learning an 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 dataset is not big enough, these

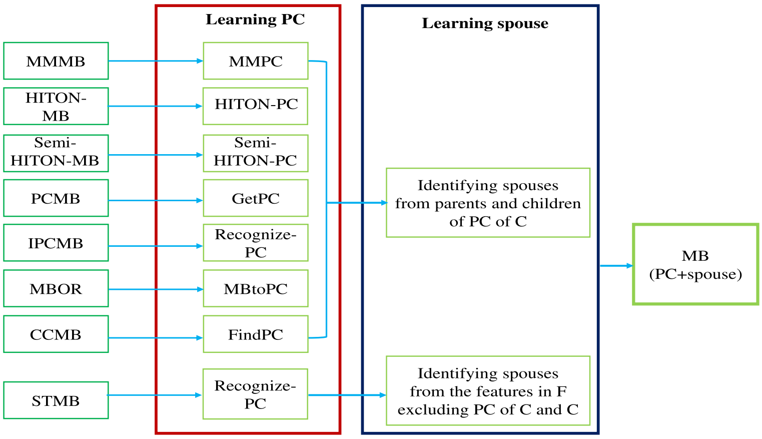

Fig. 3. Divide-and-conquer MB learning.

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. Thus, 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 $X \perp\!\!\!\perp C|S$ holds, this subset exploring process will terminate and X will be discarded and never considered again. However, for the simultaneous MB learning approach, the discarded features will be reconsidered many times (for identifying spouses).

The representative divide-and-conquer algorithms include MMMB [93], HITON-MB [7], semi-HITON-MB [7], PCMB [69], IPCMB [26], MBOR [23], STMB [30], and CCMB [100]. The main differences between those algorithms lie in the strategies of identifying $PC(C)$ and the strategies of finding $SP(C)$, as shown in Figure 3. This figure also presents the general steps of existing divide-and-conquer algorithms and the PC learning algorithms used by the eight representative MB methods, 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 learning phase and the spouse identifying phase alternatively. Specifically, once a candidate member of PC of C is added to the candidate $PC(C)$ at the PC

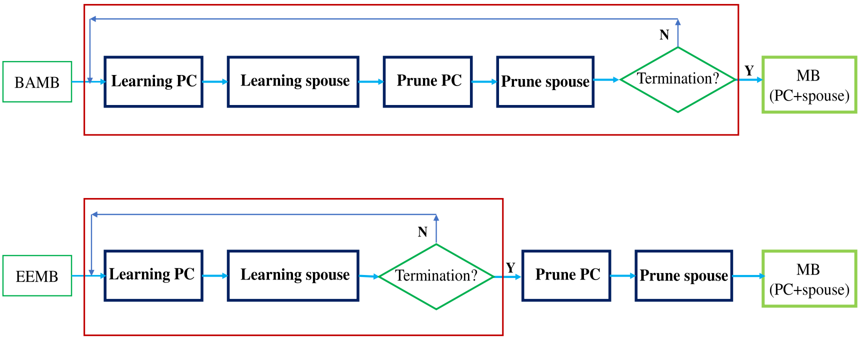

Fig. 4. MB learning with interleaving PC and spouse learning.

learning phase, this approach triggers the spouse learning phase immediately. The representative algorithms include BAMB [50] and EEMB [99]. 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.

When the faithfulness assumption is violated, the MB of a class variable in a dataset may not be unique [70, 88]. 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 [70], TIE* [88], SGAI [108], LCMB [51], and WLCMB [51]. 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 dataset, but it is often computationally expensive. 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.

When the causal sufficiency assumption is violated, if we still use an MB learning algorithm that assumes causal sufficiency, the learnt MB may not properly indicate the true causal relations. Yu et al. [106] proposed the M3B algorithm to tackle the violation of causal sufficiency. But M3B is designed based on the constraint-based approach and thus it also suffers from time efficiency and incorrect test problems.

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 an MB of a class variable from multiple datasets [104], the MCFS algorithm for stable prediction with distribution shift [107], the MIAMB and MKIAMB algorithms for learning an MB of multiple class variables [52], and the BASSUM and Semi-IAMB algorithms for MB learning with weak supervision [15, 85]. 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 attracted increasing attention in areas beyond feature selection. For example, causal knowledge has inspired efficient transfer learning

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 $CMB(C)$
```

and domain adaptation methods for accurate prediction across different domains [54, 79]. 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 first 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 algorithms, they will be introduced at the end.

GSMB. The Growing-Shrinking MB (GSMB) learning algorithm [55, 56] 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 9 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 16 in Algorithm 3), GSMB sequentially removes from $CMB(C)$ the false positive $Y \in CMB(C)$ satisfying $Y \perp\!\!\!\perp C | CMB(C) \setminus Y$. 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 $X \not\perp\!\!\!\perp C | CMB(C)$ 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, this heuristic makes GSMB ineffective in coping with a dataset 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 [92] 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 $X \not\perp\!\!\!\perp C | CMB(C)$ 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

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 $CMB(C)$
```

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 [94], Inter-IAMB [94], inter-IAMBnPC [94], and Fast-IAMB [102]. Compared to IAMB, IAMBnPC only substitutes the backward phase (Steps 11 to 15 in Algorithm 3) as implemented in IAMB with the PC algorithm [87]. To leverage prior knowledge, IAMB-IP (IAMB-Informative Prior) was proposed in Reference [75]. It can incorporate domain knowledge priors and structure sparsity priors to improve the performance of MB learning when the dataset 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, Inter-IAMBnPC 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 7 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 suggested by Agresti [3], 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. FBED^K (Forward-Backward selection with Early Dropping) [12] was developed from IAMB. In the forward phase, at each iteration IAMB should reconsider all remaining features (including all discarded features at each iteration) to find the next best candidate. To tackle the issue, $FBED^K$ 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 [95]. 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. In addition to the early dropping strategy proposed in Reference [12], 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 [93], HITON-MB [7], semi-HITON-MB [7], PCMB [69], IPCMB [26], MBOR [23], STMB [30], and CCMB [100]. 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 $PC(C)$: 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 [93] first employs the MMPC (Max-Min Parents and Children) algorithm to find candidate parents and children of C, $CPC(C)$. MMPC [93] utilizes the SFBS framework to search for $CPC(C)$ first, then prunes $CPC(C)$ at the backward phase, as shown in Algorithm 5. The novelty of MMPC lies in the fact that at Step 7 of the forward phase in Algorithm 5, MMPC proposes a max-min 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

ALGORITHM 5: The Instantiation of SFBS for PC Learning

```
Input: Feature set $F$ and the class variable $C$
Output: $CPC(C)$
$CPC(C) = \emptyset$;
// Filtering out irrelevant features by Proposition 3.1 (if $X \perp\!\!\!\perp C$ holds, $X \notin PC(C)$)
$R = F \setminus S'$ ($\forall X \in S',\ X \perp\!\!\!\perp C|\emptyset$);
//Forward phase: Adding candidate PC (or relevant) features to $CPC(C)$
repeat
    Select the best feature $X \in R$ with a greedy strategy;
    $CPC(C) = CPC(C) \cup \{X\}$; $R = R \setminus X$;
until no features in $R$ are added to $CPC(C)$;
//Backward phase: Removing false positives from $CPC(C)$;
repeat
    Consider each feature $Y \in CPC(C)$;
    if $\exists S \subseteq CPC(C) \setminus Y$ s.t. $Y \perp\!\!\!\perp C|S$ then
        $CPC(C) = CPC(C) \setminus Y$;
    end if
until no features in $CPC(C)$ are removed;
Output $CPC(C)$
```

ALGORITHM 6: The Instantiation of IFBS for PC Learning

```
Input: Feature set $F$ and the class variable $C$
Output: $CPC(C)$
$CPC(C) = \emptyset$;
$R = F \setminus S'$ ($\forall X \in S',\ X \perp\!\!\!\perp C|\emptyset$);
repeat
    //Forward phase: Adding candidate PC (relevant) features to $CPC(C)$
    Select the best feature $X \in R$ with a greedy strategy;
    $CPC(C) = CPC(C) \cup \{X\}$; $R = R \setminus X$;
    //Backward phase: Removing false positives from $CPC(C)$
    repeat
        Consider each feature $Y \in CPC(C)$
        if $\exists S \subseteq CPC(C) \setminus Y$ s.t. $Y \perp\!\!\!\perp C|S$ then
            $CPC(C) = CPC(C) \setminus Y$;
        end if
    until no features in $CPC(C)$ are removed;
until no features in $R$ are added to $CPC(C)$;
Output $CPC(C)$
```

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 $C \cup PC(C)$. 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, called $CSP(C)$. Then by Proposition 3.2, for each feature Y in the $CSP(C)$ 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

ALGORITHM 7: The Backward Framework for PC Learning

```
**Input**: Feature Set $F$ and the class variable $C$
**Output**: $CPC(C)$
$CPC(C) = \{F\}$;
i=0;
**repeat**
    **for** each feature $X$ in $CPC(C)$ **do**
        **if** $\exists S \subseteq CPC(C) \setminus X$ and $|S| = i$ such that $X \perp\!\!\!\perp C|S$ **then**
            $CPC(C) = CPC(C) \setminus X$;
        **end if**
    **end for**
    i=i+1;
**until** $i > |CPC(C)|$;
Output $CPC(C)$
```

subroutine) such that both $C \perp\!\!\!\perp Y|S$ and $C \not\perp\!\!\!\perp Y|X \cup S$ hold, then MMMB considers Y as a spouse of C.

HITON-MB and Semi-HITON-MB. HITON-MB uses the HITON-PC algorithm to discover $PC(C)$ [7]. 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 candidate PC learning and false PC 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 spouse learning, in the original version of the HITON-MB algorithm [7], the idea of HITON-MB is the same as that of MMMB.

However, Pena et al. [69, 70] 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. [96] 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, then C should be a child or a parent of X. Following this, Aliferis et al. [5] proposed a general local learning (GLL) framework and corrected the two flaws discussed above. In addition, in Reference [5], a new Semi-HITON-PC 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, Semi-HITON-MB was proposed accordingly [5].

PCMB. The parents-and-children-based MB (PCMB) algorithm [69, 70] was the first correct divide-and-conquer MB learning algorithm. Under the assumptions of faithfulness and causal sufficiency, PCMB returns the true MB of a target variable in the corresponding DAG. PCMB 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, then X will be removed from $CPC(C)$.

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 $SP(C)$
```

GetPCD adopts the similar idea of MMPC, but they have two differences. First, GetPCD adopts the IFBS framework. 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, then X will be removed from $CPC(C)$. Pena et al. [69, 70] 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 $PC(X)$. 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, then Y is a spouse of C with regard to X. The above procedure of spouse learning is summarized in Algorithm 8 [5, 70]. The study in References [5, 70] has shown that if the input $CPC(C)$ and the PC learning algorithm used by Algorithm 8 are correct, then Algorithm 8 is complete and sound [5].

IPCMB. The Iterative Parent-Child based search of MB (IPCMB) algorithm [26] is quite similar to PCMB. The key difference between them is that IPCMB employs the RecognizePC algorithm [48] 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. And as an additional improvement, 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, then IPCMB does not implement Steps 6 to 8 and moves to the next feature in $CPC(C)$.

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 this computational efficiency problem, the simultaneous MB (STMB) algorithm [30] presents two new strategies. First, STMB [30] 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.

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 ($SP(C)$ from $F \setminus CPC(C)$ and removes false positives from $CPC(C)$ using the candidate spouses selected at this step 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.

But STMB still suffers from the problem of data inefficiency at Steps 3 and 4, since at the two steps it uses an entire set as a conditional set instead of a subset exhaustive search.

MBOR. The larger the size of a conditioning set in an independence test, the less reliable is the independence test. The MB learning algorithms discussed above, such as IAMB, MMMB, and PCMB, may miss true positives due to the unreliability of the conditional independence tests if the conditioning set is large. To address this problem, MBOR (Markov Boundary search using the OR condition) [23] was designed. 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 $PC(C)$. Since MBOR uses the MBtoPC algorithm [23] to find parents and children of C, which is the variant of the simultaneous MB discovery approach, it still suffers from the problem of data inefficiency.

CCMB. To further address the incorrect conditional independence tests, Wu et al. [100] 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 test. In the work, *PCMasking* 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 test 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 gets 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.

BAMB [50] and EEMB [99] 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 [50] 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 [99] 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 a 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., KIAMB [70], TIE* [88], SGAI [108], LCMB [51], WLCMB [51], and M3B [106].

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 $S_1 \perp\!\!\!\perp (S_2 \cup W)|Z$ holds [64]. The composition property assumption is much weaker than the faithfulness assumption. KIAMB [70] 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 $K \in [0, 1]$. In the forward step, IAMB greedily adds to $CMB(C)$ the feature with the highest association with C among all features excluding features currently in $CMB(C)$. In contrast with IAMB, KIAMB uses two sets for storing candidate MBs, i.e., $CMB(C)$ and $CMB1(C)$. In the forward phase, at each iteration by conditioning on $CMB(C)$, KIAMB first adds to $CMB1(C)$ the feature with the highest association with C among all features excluding features currently in $CMB(C)$. Then KIAMB randomly chooses a CanMB subset from $CMB1(C)$ with size $max(1, \llcorner(|CMB(C)| \cdot K)\lrcorner)$ and adds to $CMB(C)$ the feature with the highest associations with C in this CanMB set. If setting $K = 1$, KIAMB is reduced to IAMB, while if taking $K = 0$, KIAMB is a completely random approach that 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 [70]. However, KIAMB does not guarantee finding all MBs of the class variable and is computationally more expensive than IAMB, because it has to be run multiple times.

TIE*. Statnikov et al. [88] relaxed the composition assumption to the local composition assumption and proposed a family of the TIE* (Target Information Equivalence) algorithm for multiple MB learning. The joint probability distribution $P(V)$ satisfies the local composition property with respect to C if $C \perp\!\!\!\perp S_1|Z$ and $C \perp\!\!\!\perp S_2|Z, C \perp\!\!\!\perp (S_1 \cup S_2)|Z$. Specifically, TIE* mainly includes three steps. In Step 1, TIE* uses an existing single MB learning algorithm to learn a $MB(C)$ from a dataset D defined on F (i.e., the original distribution) and outputs $MB(C)$. In Step 2, TIE* uses a procedure to generate a new dataset 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 an 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 datasets D_{new} generated by the procedure used in Step 2 have been considered. It has been proved that TIE* can output all possible MBs of the class variable in a dataset when the faithfulness assumption is violated.

SGAI. Due to computational problems, it may not be tractable for TIE* to learn all possible MBs for feature selection. To deal with this problem, the SGAI (Selection via Group Alpha-Investing) algorithm was proposed [108]. Compared to the standard 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 dataset, 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 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 dataset, 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 dataset is not known, the representative set cannot guarantee to include the features of all possible MBs in the dataset. In this case, the final output of SGAI is not optimal for feature selection.

LCMB and WLCMB. To tackle incorrect independent tests, in Reference [51], 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 [51] 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 that 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 Reference [51], 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 [51]. WLCMB interleaves LCMB with a search-resuming procedure and has a higher computational complexity than LCMB.

M3B. When the causal sufficiency assumption is violated, some constraint-based BN learning algorithms have been proposed to learn a global Bayesian network structure with latent variables and these algorithms are computationally expensive [20, 87]. To tackle variables in a dataset having latent common causes, the maximal ancestral graph (MAG) has been developed to represent latent common causes [77]. In contrast to DAGs, when learning a MAG with latent common causes, we do not pre-determine the number of latent common causes and their exact locations with respect to other features [77].

In Reference [106], authors adopted the MAG to represent latent common causes. Since a MAG is different from a DAG, the work in Reference [106] first defines the concept of MB of the class variable in a MAG with latent common causes, i.e., MAG MB (MMB), and presents a theoretical analysis of its properties. Then M3B was proposed to learn the MMB of the class variable. M3B mainly includes two new algorithms to find the MMB of the class variable: the AdjV algorithm using a backward strategy, as shown in Algorithm 7, to find the PC of the class variable; and the RecSearch algorithm to discover the remaining features of the MMB of the class variable. Authors have proved that M3B finds the correct MMB in a dataset with latent common causes.

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 an MB of a class variable from multiple datasets [104], MCFS for stable predictions with distribution shift [107], MIAMB and MKIAMB for learning an MB of multiple class variables [52], and BASSUM and Semi-IAMB for weak supervision learning [15, 85].

MIMB. The MB learning algorithms discussed above all learn MBs from a single observational dataset. Yu et al. [104] recently studied the problems of MB learning in multiple interventional datasets. 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 the class variable through MB learning. Based on the theoretical analysis, authors designed the MIMB (Multiple Interventional MB) algorithm to learn MB in multiple Interventional datasets. MIMB also adopts a divide-and-conquer approach that consists of two new subroutines. One subroutine, called MIPC, was designed for learning $PC(C)$ from multiple interventional datasets using the IFBS framework as presented in Algorithm 6, and the other was proposed to identify spouses of C based on the framework as shown in Algorithm 8.

MCFS. To achieve stable predictions for multiple datasets with different distributions, based on the theoretical results in Reference [104], the MCFS (multi-source causal feature selection) algorithm was proposed [107]. By utilizing the concept of causal invariance [63, 71] and mutual information, MCFS formulates the problem of stable predictions in multiple datasets as a search for an invariant set across different datasets. 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 dataset. In addition, this work demonstrated that for multiple datasets with different distributions, the set of parents of a class variable is the promising invariant set for stable predictions, while the MB or PC of the class feature may not be.

MIAMB and MKIAMB. The algorithms described above all focus on learning an MB of a single class variable, e.g., $MB(C)$, the MB of C. The work in Reference [52] recently explored the problem of learning an 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 an 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 Reference [52]. For a set of class variables of interest, given an ordering that determines which class variable's MB needs to be learned in the current step, MIAMB and MKIAMB first find an MB of two class variables and then learn an MB of three class variables and so on until all the class variables are considered.

BASSUM. To leverage both unlabelled and labelled data to help MB learning (i.e., weak-supervision MB learning), Cai et al. [15] proposed a novel BAyesian Semi-SUpervised Method

Table 2. Representative Score-based Algorithms

Category	Algorithm
Divide-and conquer MB learning (learning PC and spouses separately using a BN structure learning algorithm)	SLL [61]
	S^2TMB [31]
	S^2TMB^+ [31]
	fGES-MB [76]
Simultaneous MB learning (learning PC and spouses simultaneously)	DMB [2]
	RPDMB [2]
MB learning with relaxed assumptions	BSS-MB [57]
	LMB-CSEM [29]

(BASSUM). To the best of our 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, there are no guarantees that the modified G^2 test will follow a chi-squared distribution, and this may lead to unpredictable results. Moreover, BASSUM cannot be applied in restricted semi-supervised environments, which assume that labelled examples are only from one class while all unlabelled data are labeled all positives or all negatives before learning starts [85].

Semi-IAMB. In restricted semi-supervised environments mentioned above, assuming that all missing labels are negative or assuming that they are positive, References [84, 85] proposed a generalization of the conditional independence tests 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.

Specifically, authors present a surrogate class variable for semi-supervised hypothesis testing. Let C_0 represent assigning 0 to all missing class labels and C_1 represent assigning 1 to all missing class labels, authors use the 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 the false negative rate, authors suggested using 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 [85], 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.

4 SCORE-BASED METHODS

This type of method 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 dataset 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 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 [21], BDeu [14], BDe [38], MDL/BIC [46], AIC [4], and MIT [16]. 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 read 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 [61] 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 remove 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 removes 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 [31] 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 computational efficiency of S^2TMB.

Different from SLL and S^2TMB, DMB and RPDMB [2] 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. The fGES-MB algorithm was developed based on the fGES algorithm [76]. By adopting several optimization techniques (e.g., score caching and parallelization), fGES greatly speeds up the GES algorithm [19] and can deal with high-dimensional data. As fGES-MB is based on fGES and given the high efficiency of fGES, fGES-MB can deal with high-dimensional data.

ALGORITHM 9: Candidate PC Learning by SLL

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

When the faithfulness assumption is violated, BSS-MB [57] was proposed to learn multiple MBs using a score criterion that is a score-based variant of KIAMB. When the causal sufficiency assumption is violated, LMB-CSEM [29] was the first score-based algorithm to learn the MB of C with latent variables in a DAG. BSS-MB does not guarantee finding all possible MBs, and it does not show significant 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 is violated, and existing algorithms may suffer the computational problem of BN structure learning and they are still based on the framework of the constraint-based MB learning. These algorithms do not show significant advantages over the constraint-based methods, and thus they have not attached as much attention as constraint-based methods in the causality-based feature selection 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 [61] first learns the PC set of a class variable as shown in Algorithms 9 and 10, and then it 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 References [19, 44], 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 $C \notin CPC(X)$, then 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.

ALGORITHM 10: PC Learning with Symmetric Check by SLL

```
**Input**: Feature set $F$ and the class variable $C$
**Output**: $PC(C)$
Find $CPC(C)$ using Algorithm 9;
//Symmetric check whether $C \in PC(X)(X \in CPC(C))$
**repeat**
    Select a feature $X \in CPC(C)$;
    $CPC(C) = CPC(C) \setminus X$;
    Obtain $CPC(X)$ using Algorithm 9;
    **if** $C \notin CPC(X)$ **then**
        $CPC(C) = CPC(C) \setminus X$;
    **end if**
**until** $CPC(C)$ is empty;
$PC(C) = CPC(C)$;
Output $PC(C)$.
```

ALGORITHM 11: Spouse Learning by SLL

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

- (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 computationally expensive when the size of $F \setminus PC(C)$ is large.

S^2TMB. SLL is computationally expensive to learn DAGs for symmetric checks in Steps 2 and 4, especially with a large size of the MB of C. 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^2TMB$^+$. 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.

fGES-MB. The fGES-MB algorithm [76] has the forward and backward phases. In the forward phase, by adding an edge between variables at each iteration, fGES-MB selects the variables that have the highest scores with the class variable as the class variable's candidate PC, then it learns the candidate PC of each of the variables in the candidate PC set of the class variable (i.e., candidate spouses of the class variable). To keep the size of the candidate MB in the forward phase as small as possible, fGES-MB assumes that if the score of X and C is negative, X is not considered as a candidate PC of C [18]. In the backward phase fGES-MB removes false positives by removing an edge at each time in the MB found in the forward phase until no more improvements in the score.

4.3.2 Simultaneous MB Learning Methods. DMB and RPDMB [2] 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 an MB of the class variable instead of learning many local DAGs. Specially, they first define two restricted search spaces, that is, CDAGs (Class-focused DAGs; see Definition 1 in Reference [2]) and CRPDAGs (Class-focused Restricted Partially Directed Acyclic Graphs; see Proposition 1 in Reference [2]). Then starting from an empty graph, using the hill-climbing-based search operators proposed in Reference [1], 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 off $MB(C)$ in the obtained graphs, respectively. Compared to SLL, S^2TMB and fGES-MB, DMB, and RPDMB do not need to learn DAGs many times. But the problem is of how to obtain the two restricted search spaces is not clear in the paper [1]. If the size of the restricted search space is large, the computational cost of learning DAGs may 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 [57] 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 that measures how well this feature subset acts as an MB.

LMB-CSEM [29]. When the causal sufficiency assumption is violated, we can use score-based BN learning algorithms with latent variables to learn a global Bayesian network structure and then get the MB [25]. However, it is very computationally expensive. LMB-CSEM [29] is specially designed for MB learning with latent variables. It 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 MB is obtained from the optimal MBs within each subspace. LMB-CSEM has three major steps. At Step 1, LMB-CSEM uses a standard MB discovery algorithm to find an MB of C from observed features as the baseline. At Step 2, using the baseline MB set, it employs CSEM to learn an 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 parents (direct causes) and children (direct effects) of a class variable is critical to the prediction of the consequence of the actions/interventions in decision making or robust predictions in machine learning. Existing studies have illustrated that the set of direct causes of a class variable can be used as the set of stable or invariant features for achieving robust predications when the training data and testing data are obtained from different distributions [107]. However, existing causality-based feature selection methods using conditional independence tests do not distinguish parents from children. To address this problem, Table 3 summarizes the approaches of global BN structure learning, local structure learning, neural networks for structuring learning, and learning cause-effect relationships.

Global structure learning. The local-to-global structure learning approach, such as GSBN [56], MMHC [96], and SLL+C/G [61], first learns each feature's MB (or PC) using existing causality-based feature selection methods, then constructs a DAG skeleton (i.e., an undirected graph) using the found MBs (or PCs), and finally orients the edges of the skeleton using independence tests or score criteria. To improve the MB learning efficiency, the TC algorithm [67] was proposed to use the Relief feature selection algorithm to identify an approximate MB and conditional independence tests to orient edges. Instead of finding the MBs of all features first, the GGSL algorithm [27] starts with a randomly selected feature, then gradually expands the learned structure through a series of local structure learning steps using a score-based MB learning algorithm. Developed from GGSL, the PSL algorithm is a parallel Bayesian network structure learning algorithm [32].

Local structure learning. The local-to-global BN learning approach can deal with a dataset with thousands of features. However, in many real-world applications, we are only interested in the causal relationships around a class variable (e.g., causal genes of a disease in a gene dataset), and it is not necessary to waste time and memory to learn a global BN structure. Then several local learning algorithms have been designed for learning a local causal structure around a class variable, such as CMB [28], PCD-by-PCD [103], MB-by-MB algorithms [97], and LCS-FS [49]. Given a class variable, these algorithms first find an MB or PC of the class variable and construct a local structure among the class variable and the features in the MB or PC, then sequentially find the MB or PC of the features connected to the class variable and simultaneously construct local structures along the paths starting from the class variable until the parents and children of the class variable have

Table 3. Representative Methods for Distinguishing Parents from Children

Category	Algorithm
Global BN structure learning	GSBN [56]
	MMHC [96]
	TC [67]
	SLL+C and SLL+G [61]
	GGSL [27]
	PSL [32]
Local BN structure learning	PCD-by-PCD [103]
	CMB [28]
	MB-by-MB [97]
	LCS-FS [49]
Neural networks for BN structuring learning	DAG-GNN [109]
	D-VAE [113]
	Bengio-method [11]
Learning cause-effect relationships	ODLP and ODLP*[89]
	IDA [53]
	LiNGAM [86]
	IGCI [22]
	ANM [39]

been distinguished or it is clear that the parents and children cannot be distinguished further by continuing the process. For a class variable in a large network, the local learning algorithms are able to greatly reduce CPU time compared with the entire BN network learning methods. However, the existing local BN 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.

Neural networks for structuring learning. Recently some work has been proposed for learning BN structures using neural networks. Yu et al. [109] proposed a deep generative model to learn BN structures. Zhang et al. [113] proposed a structure learning algorithm using a variational autoencoder. However, these algorithms are computationally expensive for learning global BN structures. Instead of learning a BN structure among multiple variables, the recent work in Reference [11] employed meta-learning for distinguishing causes from effects in the two-variable case (i.e., a dataset only containing data observations of two variables). In summary, using neural networks for BN structure learning is still a new research topic, and more work could be done along this direction.

Learning cause-effect relationships. Using purely observational data, on the one hand, the BN structure learning methods discussed above always obtain the Markov equivalence class of a BN structure and leave the directions of many edges unidentified [19]; on the other hand, these methods may not uncover true causal relationships in data [65]. It is well-known that intervention experiments can allow us to distinguish causes from effects [24]. Statnikov et al. [89] proposed the ODLP and ODLP* algorithms for distinguishing causes and effects of a variable of interest using both observational data and intervention experiments. Under the faithfulness assumption, since the MB of a variable is unique, ODLP* first uses MMPC/HITON-PC to learn the PC set of a variable, then identifies causes from the found PC set using intervention experiments. When a dataset violates the faithfulness assumption, the MB of a variable may not be unique [89]. Then

ODLP employs the TIE* algorithm to learn multiple PC sets of the variable and identifies causes from the union of the multiple PC sets using intervention experiments.

However, intervention experiments are not always feasible in practice [73]. Pearl [62] proposed the structural causal model (SCM) and invented the do-calculus to simulate physical intervention experiments [62], which opens a new door to infer causal effects from observational data without requiring any actual intervention experiments. By combining existing BN structure learning algorithms with the do-calculus, the IDA algorithm [53] is well-established for inferring causal effects directly from observational data. Meanwhile, based on SCM, in the past decade, researchers have proposed many methods for distinguishing causes from effects purely from observational data in the two-variable case [72, 74]. These methods are divided into two types: methods based on additive noise models, such as LiNGAM and ANM [39, 86]; and methods based on information geometric causal inference, such as IGCI [22, 41]. Mooij et al. [59] proposed an excellent survey on the advancement in learning causal relationships in the two-variable case and thus more references on this topic can be found in this survey.

6 THE TOOLBOX

There are several open-source toolboxes for Bayesian or causal network learning, such as the well-known BNT in MATLAB [60], PGM in R[1], bnlearn in R [83], tetrad in JAVA [81], and pcalg in R [42]. But these tools do not focus on causality-based feature selection, but Bayesian 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 [90] 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 article, 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.

[1] http://mensxmachina.org/en/software/probabilistic-graphical-model-toolbox/.

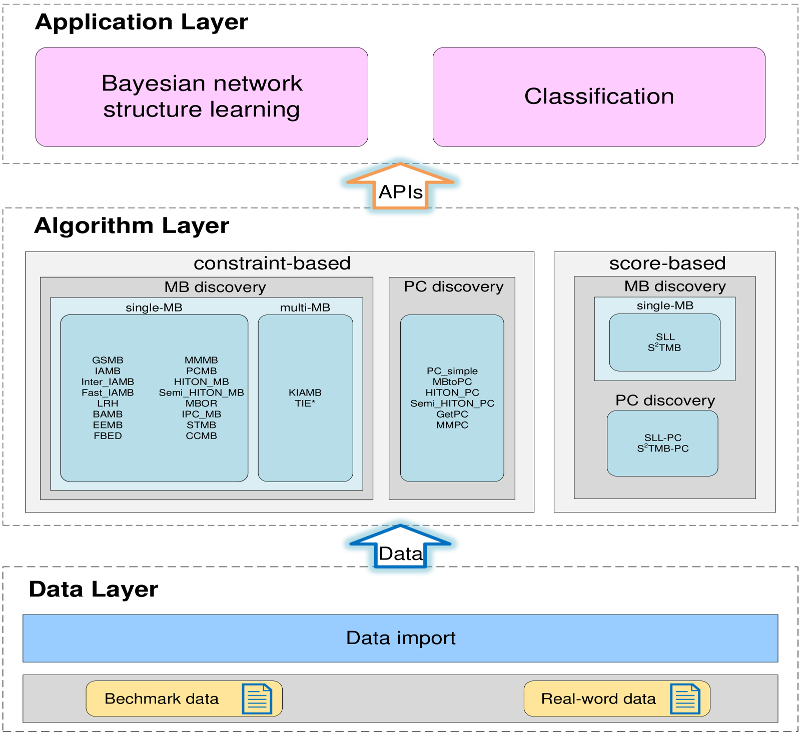

Fig. 5. Architecture of the causality-based feature selection toolbox.

7 EVALUATIONS OF CAUSALITY-BASED FEATURE SELECTION METHODS

In this section, we systematically evaluate the causality-based feature selection algorithms using the CausalFS package using synthetic and real-world datasets. For a synthetic dataset, 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, and the experimental results and findings are given in Section S-1 in the Supplement. We also evaluate these algorithms on the dense variables that have either large-sized MBs or take a large number of discrete values in Section S-2 in the Supplement. For a real-world dataset, we evaluate a causality-based feature selection algorithm based on the classification performance of the selected features and compare them with three well-established non-causal feature selection algorithms. Using eight real-world datasets from the UCI Machine Learning Repository and NIPS2003 feature selection challenge datasets, the evaluation results and findings are reported in Section S-3 in the Supplement. Here, we only summarize some main findings as follows:

- The backward strategy or the symmetry check is a double-edged sword. First, an algorithm using a forward strategy for learning the MB/PC set of a class variable may be faster than an algorithm using the backward strategy. For example, the experimental results have illustrated that MMPC, HITON-PC, and semi-HITON-PC are faster than Recognize-PC. Using synthetic data, Recognize-PC and its corresponding MB learning algorithm IPCMB are better than the other 11 constraint-based PC/MB learning algorithms on the two large-sized BN networks in learning accuracy. Second, the symmetry check (the AND or OR rule) will make an MB/PC learning algorithm very computationally expensive. However, this symmetry check can make MB/PC learning more accurate using large-sized data samples, while the OR rule will make MB/PC learning more accurate using small-sized data samples. For example, in the experiments, we found that IPCMB, PCMB, and MBOR are better than the other 11 constraint-based MB learning algorithms on the two large-sized BN networks. However, for real datasets, the symmetry check (i.e., the AND rule) may not be helpful for selecting a good feature subset for classification. PCMB and IPCMB using the AND rule are inferior to MMMB, HITON-MB, and semi-HITON-MB, the algorithms that do not use the AND rule. The possible explanation is that an algorithm using the AND rule may remove many true positives due to unreliable independence tests when a dataset has dimensionality and a small number of data samples. However, using the OR rule, MBOR always achieves stable and good prediction accuracy, especially with a dataset of high dimensionality and containing small-sized data samples. Thus, it is an interesting problem for studying regarding the conditions for using the AND rule or the OR rule or combining both to make MB learning more accurate.

- In the experiments, we have validated that the simultaneous MB learning methods are the fastest algorithms among all MB algorithms using both synthetic and real datasets. As a score-based method, fGES-MB is very computationally efficient and it is the fastest one among the three score-based methods. When the number of data samples is large, the simultaneous MB learning methods achieve very competitive prediction accuracy with their rivals, while they are inferior to the constraint-based MB learning algorithms when the size of data samples is small. Surprisingly, FBED is the fastest algorithm and its performance is comparable with the others.
- 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 learning the PC set of the class variable for feature selection is much more efficient than learning the MB of the class variable. These findings are consistent with the results in Reference [5]. Thus, in terms of feature selection, PC learning algorithms are practical in real-world applications. In addition, all types of causality-based feature selection methods cannot deal with a variable with both a large-sized MB and a large number of discrete values.
- For three non-causal feature selection algorithms, we have observed that the computational efficiency of FCBF is very competitive with the simultaneous MB learning methods, and the three algorithms are faster than the divide-and-conquer MB learning methods. The computational efficiency of mRMR is related to the number of the selected features. SPEC_CMI is the most computationally expensive and it cannot produce results for some real-world datasets within three days. Regarding learning accuracy, FCBF and mRMR also achieve good performance. However, the performance of FCBF and mRMR is determined by the user-defined parameter, i.e., the number of selected features, and it is hard to determine a well-performing value for the parameter.

8 CONCLUSION AND OPEN PROBLEMS

In this article, we first reviewed the state-of-the art causality-based feature selection algorithms, then described our developed open-source software package that implements the representative causality-based feature selection algorithms, and finally, we evaluated the representative algorithms using synthetic and real-world datasets. Although a significant number of causality-based feature selection algorithms have been developed in the past decade, many issues in big data analytics are still not addressed at all. In the future, more efforts are still required in this research direction to tackle the following challenges:

- Non-IID data. Almost all existing causality-based feature selection methods are limited to the IID data. For the non-IID data, these methods still face great challenges. One type of the non-IID scenario is that the training data and testing data have different distributions (i.e., distribution shift data). For this type of data, it is reasonable to assume that the causal mechanism of the system (i.e., direct causes) is invariant in different conditions or environments, and thus the direct causes can be used as the set of invariant features for obtaining stable predications [54, 107]. Causality-based feature selection has the potential to deal with distribution shift data, but they cannot identify causes of a class variable from data without intervention experiments. Another type of the non-IID situation is that data observations are highly interdependent (i.e., correlated data). It has shown that ignoring the correlation between data observations may significantly deteriorate the performance of BN structure learning [10], and the same problem may persist with existing causality-based feature selection methods. So far, little research has been done in development of causality-based feature selection algorithms to address the issue of correlated data.
- 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 dataset 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 in a dataset are given in advance. In fact, many real-world datasets 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 practice, a dataset 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 datasets. Thus, it is interesting to exploit unlabeled data instances to help causality-based feature selection methods with a few labeled data instances.
- Imbalanced class data. The majority of existing causality-based methods cannot deal with datasets with imbalanced classes, which, however, exist in many real-world applications. It is important to develop new feature selection methods to address this problem.
- Causal effect estimation. Existing causality-based feature selection methods do not distinguish parents from children, and thus they cannot directly help with causal effect estimation or prediction of the effects of actions (causes). However, existing causality-based feature selection methods provide the basis for efficient local causal structure learning from high-dimensional data, which in turn lays the foundation for estimating the causal effect of a cause variable on its effect variable using existing causal effect estimation methods [17, 40]. It is interesting to apply causality-based feature selection methods to calculate causal effects with high-dimensional data.
- Causality for neural networks. Recently, using causality for neural network learning has become a hot topic [8, 34, 78, 91], and using causal knowledge to solve problems in neural network learning is a right direction to follow. However, it is challenging to develop causality-based feature selection algorithms to identify features in neural network representation.

REFERENCES

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

[2] Silvia Acid, Luis M. de Campos, and Moisés Fernández. 2013. Score-based methods for learning Markov boundaries by searching in constrained spaces. *Data Mining Knowl. Disc.* 26, 1 (2013), 174–212.

[3] Alan Agresti and Maria Kateri. 2011. *Categorical Data Analysis.* Springer.

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

[5] Constantin F. Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D. Koutsoukos. 2010. Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation. *J. Mach. Learn. Res.* 11 (2010), 171–234.

[6] Constantin F. Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D. Koutsoukos. 2010. Local causal and markov blanket induction for causal discovery and feature selection for classification part ii: Analysis and extensions. *J. Mach. Learn. Res.* 11, Jan. (2010), 235–284.

[7] 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.

[8] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *Arxiv Preprint Arxiv:1907.02893* (2019).

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

[10] Harold Bae, Stefano Monti, Monty Montano, Martin H. Steinberg, Thomas T. Perls, and Paola Sebastiani. 2016. Learning Bayesian networks from correlated data. *Sci. Rep.* 6, 1 (2016), 1–14.

[11] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. 2019. A meta-transfer objective for learning to disentangle causal mechanisms. *Arxiv Preprint:1901.10912* (2019).

[12] Giorgos Borboudakis and Ioannis Tsamardinos. 2019. Forward-backward selection with early dropping. *J. Mach. Learn. Res.* 20, 1 (2019), 276–314.

[13] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. 2012. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *J. Mach. Learn. Res.* 13, Jan. (2012), 27–66.

[14] Wray Buntine. 1991. Theory refinement on Bayesian networks. In *Proceedings of the Uncertainty in Artificial Intelligence Conference (UAI'91).* Morgan Kaufmann Publishers Inc., 52–60.

[15] Ruichu Cai, Zhenjie Zhang, and Zhifeng Hao. 2011. BASSUM: A Bayesian semi-supervised method for classification feature selection. *Pattern Recog.* 44, 4 (2011), 811–820.

[16] Luis M. de Campos. 2006. A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. *J. Mach. Learn. Res.* 7, Oct. (2006), 2149–2187.

[17] Debo Cheng, Jiuyong Li, Lin Liu, Jixue Liu, Kui Yu, and Thuc Duy Le. 2020. Causal query in observational data with hidden variables. *Arxiv Preprint:2001.10269* (2020).

[18] David Maxwell Chickering. 2002. Learning equivalence classes of Bayesian-network structures. *J. Mach. Learn. Res.* 2, 3 (2002), 445–498.

[19] David Maxwell Chickering. 2002. Optimal structure identification with greedy search. *J. Mach. Learn. Res.* 3, Nov. (2002), 507–554.

[20] Diego Colombo, Marloes H. Maathuis, Markus Kalisch, and Thomas S. Richardson. 2012. Learning high-dimensional directed acyclic graphs with latent and selection variables. *Ann. Statist.* 40, 1 (2012), 294–321.

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

[22] Povilas Daniusis, Dominik Janzing, Joris Mooij, Jakob Zscheischler, Bastian Steudel, Kun Zhang, and Bernhard Schölkopf. 2012. Inferring deterministic causal relations. *Arxiv Preprint Arxiv:1203.3475* (2012).

[23] Sergio Rodrigues De Morais and Alex Aussem. 2008. A novel scalable and data efficient feature subset selection algorithm. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD'08).* Springer, 298–312.

[24] Byron Ellis and Wing Hung Wong. 2008. Learning causal Bayesian network structures from experimental data. *J. Amer. Statist. Assoc.* 103, 482 (2008), 778–789.

[25] Robin J. Evans et al. 2018. Margins of discrete Bayesian networks. *Ann. Statist.* 46, 6A (2018), 2623–2656.

[26] Shunkai Fu and Michel C. Desmarais. 2008. Fast Markov blanket discovery algorithm via local learning within single pass. In *Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence.* Springer, 96–107.

[27] Tian Gao, Kshitij Fadnis, and Murray Campbell. 2017. Local-to-global Bayesian network structure learning. In *Proceedings of the International Conference on Machine Learning (ICML'17).* JMLR.org, 1193–1202.

[28] Tian Gao and Qiang Ji. 2015. Local causal discovery of direct causes and effects. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'15).* 2512–2520.

[29] Tian Gao and Qiang Ji. 2016. Constrained local latent variable discovery. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'16)*. 1490–1496.

[30] Tian Gao and Qiang Ji. 2017. Efficient Markov blanket discovery and its application. *IEEE Trans. Cyber.* 47, 5 (2017), 1169–1179.

[31] Tian Gao and Qiang Ji. 2017. Efficient score-based Markov blanket discovery. *Int. J. Approx. Reas.* 80 (2017), 277–293.

[32] Tian Gao and Dennis Wei. 2018. Parallel Bayesian network structure learning. In *Proceedings of the International Conference on Machine Learning (ICML'18)*. 1671–1680.

[33] Clark Glymour, Kun Zhang, and Peter Spirtes. 2019. Review of causal discovery methods based on graphical models. *Front. Genet.* 10 (2019).

[34] Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michèle Sebag. 2017. Causal generative neural networks. *Arxiv Preprint:1711.08936* (2017).

[35] Ruocheng Guo, Lu Cheng, Jundong Li, P. Richard Hahn, and Huan Liu. 2020. A survey of learning causality with data: Problems and methods. *ACM Computing Surveys (CSUR)* 53, 4 (2020), 1–37.

[36] Isabelle Guyon, Constantin Aliferis, et al. 2007. Causal feature selection. In *Computational Methods of Feature Selection*. Chapman and Hall/CRC, 75–97.

[37] Isabelle Guyon and Andre Elisseeff. 2003. An introduction to variable and feature selection. *J. Mach. Learn. Res.* 3 (2003), 1157–1182.

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

[39] Patrik O. Hoyer, Dominik Janzing, Joris M. Mooij, Jonas Peters, and Bernhard Schölkopf. 2009. Nonlinear causal discovery with additive noise models. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 689–696.

[40] Antti Hyttinen, Frederick Eberhardt, and Matti Järvisalo. 2015. Do-calculus when the true graph is unknown. In *Proceedings of the Uncertainty in Artificial Intelligence Conference (UAI'15)*. Citeseer, 395–404.

[41] Dominik Janzing, Joris Mooij, Kun Zhang, Jan Lemeire, Jakob Zscheischler, Povilas Daniušis, Bastian Steudel, and Bernhard Schölkopf. 2012. Information-geometric approach to inferring causal directions. *Artif. Intell.* 182 (2012), 1–31.

[42] Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H. Maathuis, Peter Bühlmann, et al. 2012. Causal inference using graphical models with the R package pcalg. *J. Statist. Softw.* 47, 11 (2012), 1–26.

[43] Ron Kohavi and George H. John. 1997. Wrappers for feature subset selection. *Artif. Intell.* 97, 1–2 (1997), 273–324.

[44] Mikko Koivisto and Kismat Sood. 2004. Exact Bayesian structure discovery in Bayesian networks. *J. Mach. Learn. Res.* 5, May (2004), 549–573.

[45] Daphne Koller and Mehran Sahami. 1996. Toward optimal feature selection. In *Proceedings of the International Conference on Machine Learning (ICML'96)*. Morgan Kaufmann Publishers Inc., 284–292.

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

[47] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. 2017. Feature selection: A data perspective. *Comput. Surv.* 50, 6 (2017), 94.

[48] Jiuyong Li, Lin Liu, and Thuc Duy Le. 2015. *Practical Approaches to Causal Relationship Exploration*. Springer.

[49] Zhaolong Ling, Kui Yu, Hao Wang, Lei Li, and Xindong Wu. 2020. Using feature selection for local causal structure learning. *IEEE Trans. Emerg. Topics Comput. Intell.* DOI:10.1109/TETCI.2020.2978238 (2020).

[50] Zhaolong Ling, Kui Yu, Hao Wang, Lin Liu, Wei Ding, and Xindong Wu. 2019. BAMB: A balanced Markov blanket discovery approach to feature selection. *ACM Trans. Intell. Syst. Technol.* 10, 5 (2019), 1–25.

[51] Xuqing Liu and Xinsheng Liu. 2016. Swamping and masking in Markov boundary discovery. *Mach. Learn.* 104, 1 (2016), 25–54.

[52] Xu-Qing Liu and Xin-Sheng Liu. 2018. Markov blanket and Markov boundary of multiple variables. *J. Mach. Learn. Res.* 19, 1 (2018), 1658–1707.

[53] Marloes H. Maathuis, Markus Kalisch, Peter Bühlmann, et al. 2009. Estimating high-dimensional intervention effects from observational data. *Ann. Stat.* 37, 6A (2009), 3133–3164.

[54] 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 *Proceedings of the Conference and Workshop on Neural Information Processing Systems (NeurIPS'18)*. 10846–10856.

[55] Dimitris Margaritis. 2009. Toward provably correct feature selection in arbitrary domains. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'09)*. 1240–1248.

[56] Dimitris Margaritis and Sebastian Thrun. 2000. Bayesian network induction via local neighborhoods. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS'00)*. 505–511.

[57] Andrés R. Masegosa and Serafín Moral. 2012. A Bayesian stochastic search method for discovering Markov boundaries. *Knowl.-based Syst.* 35 (2012), 211–223.

[58] John H. McDonald. 2009. *Handbook of Biological Statistics.* Vol. 2. Sparky House Publishing, Baltimore, MD.

[59] Joris M. Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. 2016. Distinguishing cause from effect using observational data: Methods and benchmarks. *J. Mach. Learn. Res.* 17, 1 (2016), 1103–1204.

[60] Kevin Murphy et al. 2001. The Bayes net toolbox for Matlab. *Comput. Sci. Statist.* 33, 2 (2001), 1024–1034.

[61] T. Niinimki and Pekka Parviainen. 2012. Local structure discovery in Bayesian networks. In *Proceedings of the Workshop on Causal Structure Learning of UAI'12.* 634–643.

[62] Judea Pearl. 1995. Causal diagrams for empirical research. *Biometrika* 82, 4 (1995), 669–688.

[63] Judea Pearl. 2009. *Causality.* Cambridge University Press, Cambridge, UK.

[64] Judea Pearl. 2014. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann.

[65] Judea Pearl et al. 2009. Causal inference in statistics: An overview. *Statist. Surv.* 3 (2009), 96–146.

[66] Judea Pearl and Dana Mackenzie. 2018. *The Book of Why: the New Science of Cause and Effect.* Basic Books.

[67] Jean-Philippe Pellet and André Elisseeff. 2008. Using Markov blankets for causal structure learning. *J. Mach. Learn. Res.* 9, July (2008), 1295–1342.

[68] Jose M. Peña. 2008. Learning Gaussian graphical models of gene networks with false discovery rate control. In *Proceedings of the European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics.* Springer, 165–176.

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

[70] Jose M. Pena, Roland Nilsson, Johan Björkegren, and Jesper Tegnér. 2007. Towards scalable and data efficient learning of Markov boundaries. *Int. J. Approx. Reas.* 45, 2 (2007), 211–232.

[71] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. 2016. Causal inference by using invariant prediction: Identification and confidence intervals. *J. Roy. Statist. Soc.: Series B (Statist. Methodol.)* 78, 5 (2016), 947–1012.

[72] Jonas Peters, Dominik Janzing, and Bernhard Scholkopf. 2011. Causal inference on discrete data using additive noise models. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 12 (2011), 2436–2450.

[73] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of Causal Inference: Foundations and Learning Algorithms.* The MIT Press, Cambridge, UK.

[74] Jonas Peters, Joris M. Mooij, Dominik Janzing, and Bernhard Schölkopf. 2011. Identifiability of causal graphs using functional models. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence.* 589–598.

[75] Adam Pocock, Mikel Luján, and Gavin Brown. 2012. Informative priors for Markov blanket discovery. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics (AI and Statistics'12).* 905–913.

[76] Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. 2017. A million variables and more: The fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *Int. J. Data Sci. Anal.* 3, 2 (2017), 121–129.

[77] Thomas Richardson, Peter Spirtes, et al. 2002. Ancestral graph Markov models. *Ann. Stat.* 30, 4 (2002), 962–1030.

[78] Raanan Y. Rohekar, Shami Nisimov, Yaniv Gurwicz, Guy Koren, and Gal Novik. 2018. Constructing deep neural networks by Bayesian network structure learning. In *Proceedings of the Conference and Workshop on Neural Information Processing Systems (NeurIPS'18).* 3047–3058.

[79] M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters. 2018. Invariant models for causal transfer learning. *J. Mach. Learn. Res.* 19, 36 (2018), 1–34.

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

[81] Richard Scheines, Peter Spirtes, Clark Glymour, Christopher Meek, and Thomas Richardson. 1998. The TETRAD project: Constraint based aids to causal model specification. *Multivar. Behav. Res.* 33, 1 (1998), 65–117.

[82] Bernhard Schölkopf. 2019. Causality for machine learning. *Arxiv Preprint:1911.10500* (2019).

[83] Marco Scutari. 2009. Learning Bayesian networks with the bnlearn R package. *Arxiv Preprint:0908.3817* (2009).

[84] Konstantinos Sechidis and Gavin Brown. 2015. Markov blanket discovery in positive-unlabelled and semi-supervised data. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD'15).* Springer, 351–366.

[85] Konstantinos Sechidis and Gavin Brown. 2018. Simple strategies for semi-supervised feature selection. *Mach. Learn.* 107, 2 (2018), 357–395.

[86] Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. 2006. A linear non-Gaussian acyclic model for causal discovery. *J. Mach. Learn. Res.* 7, Oct. (2006), 2003–2030.

[87] Peter Spirtes, Clark N. Glymour, Richard Scheines, David Heckerman, Christopher Meek, Gregory Cooper, and Thomas Richardson. 2000. *Causation, Prediction, and Search*. The MIT Press, Cambridge, MA.

[88] Alexander Statnikov, Nikita I. Lytkin, Jan Lemeire, and Constantin F. Aliferis. 2013. Algorithms for discovery of multiple Markov boundaries. *J. Mach. Learn. Res.* 14, Feb. (2013), 499–566.

[89] 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. *J. Mach. Learn. Res.* 16, 1 (2015), 3219–3267.

[90] 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. *Chall. Mach. Learn.* 2 (2010), 267–278.

[91] Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. 2019. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *Proceedings of the International Conference on Machine Learning (ICML'19)*. 6056–6065.

[92] Ioannis Tsamardinos and Constantin Aliferis. 2003. Towards principled feature selection: Relevancy, filters and wrappers. In *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics*. Citeseer.

[93] Ioannis Tsamardinos, Constantin F. Aliferis, and Alexander Statnikov. 2003. Time and sample efficient discovery of Markov blankets and direct causal relations. In *Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD'03)*. ACM, 673–678.

[94] Ioannis Tsamardinos, Constantin F. Aliferis, Alexander R. Statnikov, and Er Statnikov. 2003. Algorithms for large scale Markov blanket discovery. In *Proceedings of the Florida Artificial Intelligence Research Society Conference (FLAIRS'03)*, Vol. 2. 376–380.

[95] Ioannis Tsamardinos, Giorgos Borboudakis, Pavlos Katsogridakis, Polyvios Pratikakis, and Vassilis Christophides. 2019. A greedy feature selection algorithm for big data of high dimensionality. *Mach. Learn.* 108, 2 (2019), 149–202.

[96] Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. 2006. The max-min hill-climbing Bayesian network structure learning algorithm. *Mach. Learn.* 65, 1 (2006), 31–78.

[97] 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. *Comput. Statist. Data Anal.* 77 (2014), 252–266.

[98] De Wang, Danesh Irani, and Calton Pu. 2012. Evolutionary study of web spam: Webb spam corpus 2011 versus webb spam corpus 2006. In *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'12)*. IEEE, 40–49.

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

[100] Xingyu Wu, Bingbing Jiang, Kui Yu, Chunyan Miao, and Huanhuan Chen. 2019. Accurate Markov boundary discovery for causal feature selection. *IEEE Trans. Cyber.* (2019). DOI: https://doi.org/10.1109/TCYB.2019.2940509

[101] Xindong Wu, Kui Yu, Wei Ding, Hao Wang, and Xingquan Zhu. 2013. Online feature selection with streaming features. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 5 (2013), 1178–1192.

[102] Sandeep Yaramakala and Dimitris Margaritis. 2005. Speculative Markov blanket discovery for optimal feature selection. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'05)*. IEEE, 4–9.

[103] 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 *Proceedings of the Workshop on the Causation and Prediction Challenge*. 93–105.

[104] Kui Yu, Lin Liu, and Jiuyong Li. 2018. Discovering Markov blanket from multiple interventional datasets. *Arxiv Preprint:1801.08295* (2018).

[105] Kui Yu, Lin Liu, and Jiuyong Li. 2018. A unified view of causal and non-causal feature selection. *Arxiv Preprint:1802.05844* (2018).

[106] Kui Yu, Lin Liu, Jiuyong Li, and Huanhuan Chen. 2018. Mining Markov blankets without causal sufficiency. *IEEE Trans. Neural Netw. Learn. Syst.* 99 (2018), 1–15.

[107] Kui Yu, Lin Liu, Jiuyong Li, Wei Ding, and Thuc Le. 2019. Multi-source causal feature selection. *IEEE Trans. Pattern Anal. Mach. Intell.* DOI: 10.1109/TPAMI.2019.2908373 (2019).

[108] Kui Yu, Xindong Wu, Wei Ding, Yang Mu, and Hao Wang. 2017. Markov blanket feature selection using representative sets. *IEEE Trans. Neural Netw. Learn. Syst.* 28, 11 (2017), 2775–2788.

[109] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. 2019. DAG-GNN: DAG structure learning with graph neural networks. In *Proceedings of the International Conference on Machine Learning (ICML'19)*. 7154–7163.

[110] Yiteng Zhai, Yewsoon Ong, and Ivor W. Tsang. 2014. The emerging big dimensionality. *IEEE Comput. Intell. Mag.* 9, 3 (2014), 14–26.

[111] Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2012. Kernel-based conditional independence test and application in causal discovery. *Arxiv Preprint:1202.3775* (2012).

[112] Kun Zhang, Bernhard Schölkopf, Peter Spirtes, and Clark Glymour. 2017. Learning causality and causality-related learning: Some recent progress. *Nat. Sci. Rev.* 5, 1 (2017), 26–29.

[113] Muhan Zhang, Shali Jiang, Zhicheng Cui, Roman Garnett, and Yixin Chen. 2019. D-VAE: A variational autoencoder for directed acyclic graphs. In *Proceedings of the Conference and Workshop on Neural Information Processing Systems (NeurIPS'19)*. 1586–1598.

Received November 2019; revised June 2020; accepted June 2020