Multi-Source Causal Feature Selection

Kui Yu, Lin Liu, Jiuyong Li, Wei Ding, *Senior Member, IEEE*, and Thuc Duy Le

Abstract—Causal feature selection has attracted much attention in recent years, as the causal features selected imply the causal mechanism related to the class attribute, leading to more reliable prediction models built using them. Currently there is a need of developing multi-source feature selection methods, since in many applications data for studying the same problem has been collected from various sources, such as multiple gene expression datasets obtained from different experiments for studying the causes of the same disease. However, the state-of-the-art causal feature selection methods generally tackle a single dataset, and a direct application of the methods to multiple datasets will result in unreliable results as the datasets may have different distributions. To address the challenges, by utilizing the concept of causal invariance in causal inference, we first formulate the problem of causal feature selection with multiple datasets as a search problem for an invariant set across the datasets, then give the upper and lower bounds of the invariant set, and finally we propose a new Multi-source Causal Feature Selection algorithm, MCFS. Using synthetic and real world datasets and 16 feature selection methods, the extensive experiments have validated the effectiveness of MCFS.

Index Terms—Causal feature selection, Markov blanket, multiple datasets, Bayesian network, causal invariance

1 Introduction

Feature selection is an effective approach to reducing dimensionality by selecting features (variables) that are most relevant to the class attribute for better prediction. In recent years, causal feature selection [1], [11] is attracting more attentions and has been increasingly used in building prediction models, since the causal features selected can imply the causal mechanisms around the class attribute. Consequently, in contrast to traditional or non-causal feature selection, a prediction model built with causal features can be explained in terms of the causal relevance of the features with the class attribute. Moreover, causal features enable more reliable predictions in non-static environment where the distributions of testing and training data may be different, and allow the prediction of the outcomes of actions [11].

Many causal feature selection algorithms have been developed [1], [9], [20], with the aim to identify the Markov blanket (MB) of the class attributes or a subset of the MB. A MB of a variable contains its parents (direct causes), children (direct effects), and spouses (direct causes of children) when the relations between variables are represented using a Bayesian network [19].

However, all the methods are designed for causal feature selection from a single data set, whereas multiple datasets studying a same problem are ubiquitous nowadays. For example, multiple gene expression datasets may have been obtained from experiments conducted at different laboratories for the discovery of genetic causes of the same disease, such as lung cancer [10]. To develop strategies for effective promotion of a product, data may have been collected from various sources, such as A/B tests, customer surveys, and records of previous promotional campaigns. It is desirable to maximize the use of the richer information contained in the multiple datasets to develop better solutions. The challenge is that, however, existing causal feature selection methods are not able to be applied to multiple datasets directly because

- Unreliable results will be obtained if we simply pool the multiple datasets together and then apply an existing causal feature selection method to the pooled data. Although the multiple datasets are targeted at the same problem, they often have been produced from different experiments or sources, thus do not have identical distributions. For instance, to identify the impact of genes on a disease, in an experiment, the expression levels of some genes are manipulated (intervened), and then the expression changes of the marker genes of the disease are observed. As in different experiments different genes may be intervened, the distributions of the datasets obtained from these experiments may not be identical. Then in the pooled data, due to the different/inconsistent distributions, the relationship between a feature and the target attribute may not be detected any more (while it might be observed in a single dataset).
- It will not work well either if we apply an existing causal feature selection method to each dataset individually and then take the commonly selected features, because in this case we will lose useful information

- *K. Yu is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China, and also with the School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, SA 5095, Australia. E-mail: ykui713@gmail.com.*
- *L. Liu, J. Li, and T.D. Le are with the School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, SA 5095, Australia. E-mail: {Lin.Liu, Jiuyong.Li, Thuc.Le}@unisa.edu.au.*
- *W. Ding is with the Department of Computer Science, University of Massachusetts Boston, Boston, MA 02125. E-mail: wei.ding@umb.edu.*

Manuscript received 14 Aug. 2018; revised 3 Feb. 2019; accepted 21 Mar. 2019. Date of publication 29 Mar. 2019; date of current version 4 Aug. 2020.
(Corresponding author: Kui Yu.)
Recommended for acceptance by M. Sugiyama.
Digital Object Identifier no. 10.1109/TPAMI.2019.2908373

provided by the different datasets. For instance, suppose that a gene is important for predicting a disease, but it is manipulated in one training dataset while not in another, the commonly selected features from these datasets may not include the important gene for predicting the disease.

To tackle these problems, in the paper, we propose a multi-source causal feature selection approach by utilizing the concept of *causal invariance* [18], [22] in causal inference. The main idea behind causal invariance is that although in the experiments from which these datasets were obtained, different variables might have been intervened (resulting in different probability distributions of the datasets), since the datasets are for the same system, the underlying causal mechanism of the system should keep invariant across the experiments.

Based on the observations, we assume that there exists an invariant set S^* such that the conditional distribution of the class attribute C, $P(C|S^*)$ maintains the same across the datasets. As we will show in Section 4.2 (Theorem 6) that the set of direct causes (parents) of C is such an invariant set. As the ultimate goal of feature selection is to achieve good predictions, we would like to find a set of features S^* which not only satisfy the invariant property across the datasets, but also can maximize $P(C|S^*)$. Our goal is to search for such a feature set S^*.

In recent years, causal invariance has been employed to tackle domain adaptation problems [15], [23]. Particularly, based on causal invariance, a new method was proposed [15] to select a set of features that makes the predictions adaptable to a different domain. Our work is closely related to the existing work for cross-domain predictions since the causal features learnt from multiple training datasets carries richer and more reliable causal knowledge, and thus give more stable predictions in domains with different external environment/interventions. However, our work is mainly driven by the idea of better utilizing information in multiple sources to select a set of causal features for stable predictions, and the method is designed without assumed source (training) or target (testing) domains as in the previous work for domain adaptation.

The contribution of this paper can be summarized as follow:

- We analyze the properties of causal invariance for feature selection with multiple datasets, formulate the problem of multi-source causal feature selection as a search problem for an invariant set, and represent the search criterion using mutual information. Moreover, we give the upper and lower bounds of the invariant sets.
- Based on the theories established in the first contribution above, we propose a new Multi-source Causal Feature Selection algorithm MCFS. The effectiveness and efficiency of the MCFS algorithm are validated by a series of experiments using synthetic and real world data.

The rest of the paper is organized as follows. Section 2 reviews the related work, and Section 3 gives notations and definitions. Section 4 analyzes causal feature selection with multiple datasets, while Section 5 proposes our new algorithm. Section 6 describes and discusses the experiments and Section 7 concludes the paper.

2 Related Work

In the big data era, high-dimensional datasets have become ubiquitous in various applications [33]. And thus, feature selection is pressing more than ever, and thus many feature selection methods have been proposed. The most existing feature selection methods fall into three main categories, filter, wrapper, and embedded methods [13]. Filter feature selection methods are classifier independent, the other two types of methods are not. Excellent reviews of classical feature selection (i.e., filter, embedded, wrapper) algorithms can be found in [6], [12], [13] and the reference therein.

Causal feature selection has attracted much attention in recent years, since by bringing causality into play, it naturally provides causal interpretation about the relationships between features and the class attribute, enabling a better understanding of the mechanisms behind data [1], [11]. Additionly, the MB of the class attribute is a minimal set of features which renders the class attribute statistically independent from all the remaining features conditioned on the MB [19]. Causal feature selection did not become practical until Tsamardinos and Aliferis [26] proposed the IAMB family of algorithms, such as IAMB [26], inter-IAMB [28], IAMBnPC [28], and Fast-IAMB [30]. These algorithms attempt to find parents and children (PC) and spouses of a target variable simultaneously.

However, the IAMB family of algorithms is not able to distinguish parents and children from spouses of the target. In addition, they require a large number of data samples at least exponential to the size of the MB of the target, and thus they would not scale to thousands of variables in most real-world datasets with small numbers of data samples. To mitigate the problem, a divide-and conquer approach was proposed. The ideas behind the approach are that instead of discovering PC and spouses of a target variable simultaneously, it first finds the PC of the target, then discovers its spouses. The representative algorithms include HITION-MB [1], [2], MMMB [27], PCMB [20], and STMB [9]. However, existing causal feature selection algorithms only focus on selecting features from a single (training) dataset. Thus, there is a need for causal feature selection to specially selecting features from multiple datasets.

Recently, Yu et al. [31] theoretically analyzed under what conditions the correct MB of a target variable can be found and under what conditions the causes of the target variable are able to be identified via discovering its MB from multiple interventional datasets. And some methods have utilized the idea of causal invariance [18] for learning causal structures from multiple interventional datasets. Peters et al. [22] proposed the ICP algorithm to discover a target variable's direct causes from multiple interventional datasets by using the causal invariance. Zhang et al. [34] proposed an enhanced constraint-based algorithm for learning causal structures from heterogeneous data. And Mooij et al. [16] proposed a novel framework to unify causal structure learning with multiple interventional datasets.

However, the existing work uses the idea of causal invariance to discover causal structures, instead of finding causal

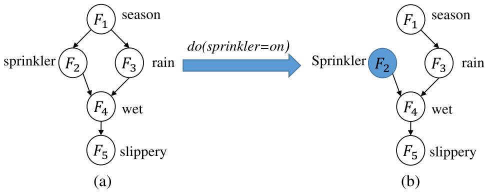

Fig. 1. Example of BN and interventions. (a) A simple BN representing dependencies among five variables; (b) An example of an intervention on variable "sprinkler".

features for building prediction models. In addition, [16] and [34] are both computational expensive or prohibitive when datasets contain large number of variables, and they need to specify a set of context variables (e.g., prior knowledge of interventions) to help causal structure learning, which may not be practical in many real-world applications.

Magliacane et al. [15] proposed a novel method to address domain adaptation problem, specifically transferable predictions. The idea behind [15] is to employ causal invariance to find a separating set to be used in the predictions in target domains. The proposed algorithm first uses a standard feature selection method such as Random Forests to generate a list of candidate feature sets, then identifies a set satisfying the invariance as a separate set. Both our work and the method in [15] utilize the idea of causal invariance and the causal features obtained by our method can also be used for predictions in different domains. However, they have the following differences: (1) The method in [15] needs to specify context variables while our work does not; (2) [15] assumes that datasets in the source domains (or the multiple training datasets) have the same distribution while our work deals with training datasets with different distributions; (3) As we will see later, our work can be scalable to thousands of variables, but as presented in [15], the method in practice only dealt with several variables; and (4) As introduced in Section 5, our method makes use of source domain data only, and it starts with candidate features selected from individual datasets by a causal feature selection method and then uses the invariance to select those can make stable predications; whereas the method in [15] utilizes data in both source and target domains, and starts with the candidate feature sets selected by a normal (non-causal) feature selection method and then uses causal inference method to filter out features that would not transfer to the target domain.

In summary, there is a lack of effective feature selection methods for selecting causal features from multiple datasets, thus, in this paper, we will focus on tackling causal feature selection with multiple datasets for stable predictions.

3 Notations and Definitions

In this section, we discuss some key concepts involved in tackling causal feature selection with multiple datasets. Specifically, Section 3.1 presents the concepts of Bayesian networks and Markov blankets w.r.t causal feature selection, and Section 3.2 discusses the intervention theory in causal inference, which is related to the idea of causal invariance.

Let $D = \{D_1, D_2, \ldots, D_K\}$ be K training datasets. $\forall i \in \{1, \ldots, K\}$, D_i is defined by $\{F, C\}$, i.e., the datasets all contain the same set of features $F = \{F_1, F_2, \ldots, F_N\}$ and the class attribute C. Let Υ_i ($\Upsilon_i \subset F$) be the features manipulated in the ith experiment, and $\Upsilon = \{\Upsilon_1, \ldots, \Upsilon_K\}$ the K intervention experiments producing $D_1, \ldots, D_K$, respectively. Note that in this paper we assume that the class attribute is not intervened in any of the experiments (more details in Section 3.2) (In the following, we use the two terms, class attribute and target variable, interchangeably). We use $\setminus$ to denote set subtraction. For simplicity, we abuse the notation and write $F\setminus\{F_i\}$ as $F\setminus F_i$ to indicate all features in F excluding F_i. F_i and F_j ($i \neq j$) are said to be conditionally independent given $S \subseteq F\setminus\{F_i, F_j\}$ if and only if $P(F_i, F_j|S) = P(F_i|S)P(F_j|S)$. We use $F_i \perp\!\!\!\perp F_j|S$ and $F_i \not\perp\!\!\!\perp F_j|S$ to represent that given S, F_i is conditionally independent of and dependent on F_j, respectively.

For the convenience of presentation, we let $F_{N+1} = C$ and $\mathcal{F} = \{F_1, F_2, \ldots, F_N, F_{N+1}\}$, representing the set of all variables under consideration, including all the features and the class attribute.

3.1 Bayesian Network and Markov Blanket

Let P be the joint probability distribution of D and represented by a directed acyclic graph (DAG) G over $\mathcal{F}$. A Bayesian network is defined as follows.

Definition 1 (Bayesian network) [19]. *The triplet $\langle \mathcal{F}, G, P \rangle$ is called a Bayesian network if $\langle \mathcal{F}, G, P \rangle$ satisfies the Markov condition: every variable is independent of any subset of its non-descendants conditioned on its parents in G.*

In this paper, we consider *causal Bayesian network* (CBN), a BN in which an edge $X \to Y$ indicates that X is a direct cause of Y [18]. For simple presentation, however, we use the term BN instead of CBN.

For example, Fig. 1a shows a simple yet typical BN [18]. A Bayesian network encodes the joint probability P over a set of variables $\mathcal{F}$ and decomposes P into a product of the conditional probability distributions of the variables given their parents in G. Let $pa(F_i)$ represent the set of parents of F_i in $\mathcal{F}$. We have the following decomposition of P:

$$P(\mathcal{F}) = \prod_{i=1}^{N+1} P(F_i|pa(F_i)). \tag{1}$$

Definition 2 (Faithfulness) [19]. *Given a Bayesian network $< \mathcal{F}, G, P >$, G is faithful to P if and only if every conditional independence present in P is entailed by G and the Markov condition. P is faithful if and only if there exists a DAG G such that G is faithful to P.*

Let $ch(F_i)$ and $sp(F_i)$ represent the sets of children and spouses of F_i in $\mathcal{F}$, then the Markov blanket of F_i in a BN is defined as follows.

Definition 3 (Markov blanket) [19]. *Under the faithfulness assumption, the Markov blanket of $F_i \in \mathcal{F}$ in a BN, noted as $MB(F_i)$, is unique and $MB(F_i) = \{pa(F_i) \cup ch(F_i) \cup sp(F_i)\}$.*

3.2 Interventions in BNs

To represent the intervention on a variable in an intervention experiment, Pearl [18] proposed the *do* operator $do(X = x)$ to indicate that the value of variable X is set to a constant x by the intervention. If we use a DAG to represent the causal relations between variables in $\mathcal{F}$, an intervention on a variable can be indicated by deleting all the edges pointing to the variable [18]. For example, to represent the intervention "turning the sprinkler On" (i.e., $do(sprinkler = on)$) in the network as shown in Fig. 1b, the link from F_1 to F_2 is deleted and F_2 is assigned the value "On".

Property 1 [18]. $P(F_j|pa(F_j)) = P(F_j|do(pa(F_j) = \zeta))$ if $F_j \notin \Upsilon_i$ *where* ζ *is a set of constant values of* $pa(F_j)$.

Property 2 [18]. *Assuming* $S \subseteq \mathcal{F} \setminus \{F_j, pa(F_j)\}$, *if* $F_j \notin \Upsilon_i$, $P(F_j|do(pa(F_j) = \zeta)), S) = P(F_j|do(pa(F_j) = \zeta))$.

Property 1 ensures that $P(F_j|pa(F_j))$ coincides with the effect (on F_j) of setting $pa(F_j)$ to the chosen values. Property 2 illustrates that once we control the direct causes of F_j (i.e., $pa(F_j)$), no other interventions will affect the probability of F_j. The DAG obtained after all the interventions of an intervention experiment are represented by the edge deletions is known as a post-manipulation DAG, and it is formal definition is given in the following.

Definition 4 (Post-manipulation DAG) [18]. *Let* $G = (\mathcal{F}, E)$ *be a DAG with variable set* $\mathcal{F}$ *and edge set* E. *After the intervention on the set of variables* Υ_i *(represented as* $do(\Upsilon_i = \gamma)$*), the post-manipulation DAG of* G *is* $G_i = (\mathcal{F}, E_i)$ *where* $E_i = \{(a, b)|(a, b) \in E, b \notin \Upsilon_i\}$. *The joint distribution of the post-manipulation DAG* G_i *with respect to the set* Υ_i *can be written as*

$$P(\mathcal{F}|do(\Upsilon_i = \gamma)) = \prod_{F_j \in \mathcal{F} \setminus \Upsilon_i} P(F_j|pa'(F_j), do(pa''(F_j) = \gamma), \tag{2}$$

where $pa'(F_j) \subseteq \mathcal{F} \setminus \Upsilon_i$ *and* $pa''(F_j) \subseteq \Upsilon_i$. *By Properties 1 and 2,* $P(F_j|pa'(F_j), do(pa''(F_j) = \gamma)$ *is the same as the conditional probability of* F_j *in Eq. (1) if* F_j *is not intervened, i.e.,* $P(F_j|pa(F_j))$ *remain invariant to interventions not involving* F_j *, while* $P(do(F_j = \gamma)|pa(F_j)) = 1$ *if* F_j *is intervened.*

For example, the post-manipulation DAG resulting from the intervention on variable "sprinkler" as shown in Fig. 1b is $P(F_1, F_2, F_3, F_4, F_5|do(F_2{=}On)) = P(F_1)P(F_3|F_1)P(F_4|F_3, F_2 = On)P(F_5|F_4)$.

4 Multi-Source Causal Feature Selection

As mentioned in the Introduction section, we formulate the problem of multi-source causal feature selection as a search problem for an invariant set across all the training datasets $D = \{D_1, D_2, \ldots, D_K\}$. Assuming $\forall D_i \in D$ and $\forall D_j \in D$ $(i \neq j)$, an invariant set S across D is defined as follows.

Definition 5 (Invariant set). *An invariant set* S *across* D *satisfies* $P^i(C|S) = P^j(C|S)$, *for* $\forall D_i, D_j \in D$.

As the goal of feature selection is to select a subset $S \subseteq F$ to maximize $P(C|S)$, given D, we would like to find a set of features S^* which is not only an invariant set across D, but also can maximize $P(C|S)$. Accordingly, the problem of causal feature selection with D is defined that given any dataset $D_i \in D$, then

$$\begin{aligned} S^* &= \arg\max\nolimits_{S \subseteq F} P^i(C|S) \\ &\quad s.t.\ P^i(C|S) = P^j(C|S)\ (\forall j,\ j \neq i). \end{aligned} \tag{3}$$

To tackle Eq. (3), in the following, Section 4.1 proposes the rationale of maximizing $P(C|S)$ for optimal prediction. Section 4.2 discusses the lower and upper bounds of S in Eq. (3) for search efficiency, and Section 4.3 analyzes the properties of the upper bound of S in D.

4.1 Rationale of Maximizing $P(C|S)$ for Optimal Prediction

For a subset $S \subseteq F$, why S is optimal for feature selection when S maximizes $P(C|S)$? We discuss the question using mutual information and the Bayes error rate. For classification, the minimum achievable classification error by any classifier is called the Bayes error rate [8]. The Bayes error rate is used for justifying $P(C|S)$ for optimal prediction since it is the tightest possible classifier-independent lower-bound by depending on predictive features and the class attribute alone.

Let $I(F_i, F_j)$ denote the mutual information of F_i and F_j, we can formulate $S^* = \arg\max_{S \subseteq F} P(C|S)$ as $S^* = \arg\max_{S \subseteq F} I(S; C)$, that is, maximizing $I(S; C)$ is equivalent to maximizing $P(C|S)$ [6]. Let P_{err} represent the Bayes error rate and $H(P_{err})^{-1}$ be the inverse of the entropy $H(P_{err})$, given C and $S \subseteq F$, the upper bound of P_{err} is given as Eq. (4) below [25]

$$H(P_{err})^{-1} \leq P_{err} \leq 1/2H(C|S). \tag{4}$$

Eq. (4) illustrates that minimizing $H(C|S)$ minimizes the Bayes error rate. By the term $I(C; S) = H(C) - H(C|S)$, maximizing $I(C; S)$ is equivalent to minimizing P_{err}. Accordingly, maximizing $P(C|S)$ is equivalent to minimizing P_{err}.

4.2 Bounds of S in Eq. (3)

In this section, using the concept of MBs in a BN, we will first discuss what S is exactly in Eq. (3) when D only contains a single training dataset that is sampled from the same distribution as the test dataset $(K = 1)$, then explore the bounds of S in Eq. (3) as $K > 1$.

Theorem 1 [19]. *Suppose* $MB(C)$ *is the MB of* C *in a BN,* $\forall S \subset \mathcal{F} \setminus \{MB(C) \cup C\}$, $P(C|MB, S) = P(C|MB)$.

By Theorem 1, Theorem 2 is achieved and it states that for $\forall S \subseteq F$, $I(C; MB(C)) \geq I(C; S)$ with equality if and only if $S = MB(C)$. By Theorem 2, we can see that all information that may influence the values of C is stored in the values of features of $MB(C)$.

Theorem 2. $I(C; MB(C))$ *is maximal.*

By Theorem 2 and Eq. (4), Theorem 3 below is achieved. Theorem 3 illustrates that $MB(C)$ is the optimal solution to Eq. (3) when $K = 1$ and the training and testing dataset are both generated from the same data distribution.

Theorem 3. $MB(C)$ *minimizes the Bayes error rate.*

Given multiple training datasets D $(K > 1)$, if the manipulated variables in both D and the testing dataset are not known, then what causal invariance properties will present in D? With these properties, what are the lower and upper bounds of S in Eq. (3)? Assuming that the class attribute C is not intervened and faithfulness holds, we discuss the first question above with Theorems 4 and 6, and the second one with Theorem 7 below.

Theorem 4. *Suppose $MB(C)$ is the MB of the class attribute C, if for $\forall \Upsilon_i \in \Upsilon$ and $\forall \Upsilon_j \in \Upsilon$ $(i \neq j)$, $ch(C) \not\subseteq \Upsilon_i$ and $ch(C) \not\subseteq \Upsilon_j$, $P^i(C|MB(C)) = P^j(C|MB(C))$ holds.*

According to Theorem 3, Theorem 4 states that if for all variables in $ch(C)$ are not manipulated in any datasets in D, $MB(C)$ is the largest invariant set across all datasets in D. Theorems 5 and 6 below illustrate that if C are not manipulated in any datasets in D, $pa(C)$ is not only an invariant set but also a minimal one across all datasets in D.

Theorem 5. *For $\forall D_i \in D$ and $\forall D_j \in D$ $(i \neq j)$, $P^i(C|pa_i(C)) = P^j(C|pa_j(C))$.*

Theorem 6. *$pa(C)$ is the minimal and invariant set across D with regard to C.*

By Theorems 4, 5, and 6, without variable manipulation information in D, the bounds of S in Eq. (3) is given in Theorem 7.

Theorem 7. *In Eq. (3), $pa(C) \subseteq S \subseteq MB(C)$.*

By Theorem 7, Eq. (3) is rewritten as Eq. (5) below:

$$\begin{aligned} S^* = & \arg\max\nolimits_{S \subseteq MB(C)} P^i(C|S) \\ & s.t.\ P^i(C|S) = P^j(C|S)\ (\forall j,\ j \neq i). \end{aligned} \tag{5}$$

4.3 Properties of $MB(C)$ in Multiple Datasets

How do we find $MB(C)$ from D without any variable manipulation information in each dataset? We discuss the problem with Theorems 8, 9, and 10 below.

Definition 6 [9]. *Υ is conservative, if $\forall F_j \in \bigcup_{i=1}^{K} \Upsilon_i$, $\exists \Upsilon_i \in \Upsilon$ such that $F_j \notin \Upsilon_i$.*

Definition 6 states that given the set of K interventional experiments, if for any variable that is intervened, we can always find an experiment in which the variable is not manipulated, then we say that the set of interventional experiments is conservative.

Theorem 8. *If Υ is conservative and $MB_i(C)$ represents MB (C) in D_i, the union $\bigcup_{i=1}^{K} MB_i(C) = MB(C)$ holds.*

Theorem 9. *If Υ is not conservative, $pa(C) \subseteq \bigcup_{i=1}^{K} MB_i(C) \subseteq MB(C)$.*

Theorem 8 states that if Υ is conservative, the union of $MB(C)$ in each dataset of D exactly equals $MB(C)$; if not, Theorem 9 shows that the union of $MB(C)$ is between $pa(C)$ and $MB(C)$. By Theorems 8 and 9, we get Theorem 10 as follows, which illustrates that $pa(C)$ is the minimal invariant set across D whatever Υ is conservative or not.

Theorem 10. *No matter Υ is conservative or not, $pa(C) \subseteq \bigcup_{i=1}^{K} MB_i(C)$.*

Theorems 8, 9, and 10, on the one hand, further illustrate the bounds shown in Theorem 7; on the other hand, these theorems discuss the properties of $MB(C)$ in D containing multiple interventional datasets without variable manipulation information. This also gives the basic ideas of finding $MB(C)$ from D by the algorithm presented in the next section.

5 The Proposed MCFS Algorithm

To solve Eq. (5), we propose the Multi-Source Causal Feature Selection (MCFS) algorithm (Algorithm 1) which has three phases. Phase 1 is carried out in Steps 2 to 5 for finding $MB(C)$ in D, Phase 2 is done in Steps 6 to 26 for discovering candidate invariant sets from D, and Phase 3 lies in Step 27 for selecting S^* from these candidate invariant sets.

Algorithm 1. *The MCFS Algorithm*

```
 Input: $D = \{D_1, D_2, \ldots, D_K\}$, $C$: the class attribute, $\alpha$: significance level
 Output: $S^*$
 1: $MB(C) = \emptyset$; $\rho = \emptyset$; $SelFea = \emptyset$;
 2: for i=1 to K do
 3:     /*Find $MB_i(C)$ in dataset $D_i$;
 4:     $MB(C) = MB(C) \cup MB_i(C)$;
 5: end
 6: for $S \subseteq MB(C)$ do
 7:     $avg_{MI} = 0$;
 8:     for i=1 to K do
 9:         $MI(i) = \emptyset$, $I^i(C;S)$=0;
10:         for j=1 to $|S|$ do
11:             /* computing $MI(i)$ on $D_i$ by Eq. (10)
12:             $MI(i) = MI(i) \cup I^i(C;F_j)$ $(F_j \in S)$;
13:             $I^i(C;S) = I^i(C;S) + I^i(C;F_j)$;
14:         end
15:         $ave_{MI} = ave_{MI} + \frac{1}{|S|} I^i(C;S)$;
16:     end
17:     $ave_{MI} = \frac{1}{K} ave_{MI}$;
18:     for i=1 to K do
19:         /* using t-test to calculate whether the mean of $MI(i)$ is
            identical to $ave_{MI}$
20:         $\rho_i$=get-p-value $(MI(i), ave_{MI})$;
21:         $\rho = \{\rho \cup \rho_i\}$;
22:     end
23:     if $\min(\rho) \geq \alpha$ then
24:         $SelFea = SelFea \cup S$;
25:     end
26: end
27: output $S^*$ with the highest prediction accuracy from $SelFea$.
```

5.1 Phase 1 (Steps 2 to 5): Discovering $MB(C)$ from D

By the analysis in Section 4.3, Phase 1 employs the HITON-MB algorithm, one of the best MB discovery algorithms [1] (any other up-to-date MB algorithms can be used here) to find $MB_i(C)$ in D_i, then union the found MBs in each dataset as $MB(C)$.

5.2 Phase 2 (Steps 6 to 26): Finding Candidate Invariant Sets in $MB(C)$

Mutual Information for Computing $P(C|S)$. In Eq. (5), it is difficult to calculate $P(C|S)$ especially for a large sized S [17]. Thus, Phase 2 uses mutual information as an alternative to compute $P(C|S)$ as follows. Given dataset $D_j \in D$, $j \in \{1, \ldots, K\}$, let $p(C|S, D_j)$ denote the true class distribution of D_j and $q(C|S, D_j)$ represent the predicted class distribution of D_j given S. Then the conditional likelihood of C given S is calculated by $L(C|S, D_j) = \prod_{i=1}^{M} q(c^i|s^i)$ where M is the number of data instances in D_j, c^i represents a value of C in the ith data instance, and s^i denotes a value set of S in the ith data instance. The (scaled) conditional log-likelihood of $L(C|S, D_j)$ is computed by

$$\ell(T|S, D_j) = \frac{1}{M}\sum_{i=1}^{M} \log q(c^i|s^i). \tag{6}$$

By [6], Eq. (6) can be rewritten as Eq. (7) where f^i denotes a value set of F in the ith data instance[1].

$$\begin{aligned} - \ell(T|S, D_j) = E_{cs}\left\{\log\frac{p(c^i|s^i)}{q(c^i|s^i)}\right\} + E_{cf}\left\{\log\frac{p(c^i|f^i)}{p(c^i|s^i)}\right\} \\ - E_{cf}\left\{\log p(c^i|f^i)\right\}. \end{aligned} \tag{7}$$

Eq. (7) can be further rewritten as Eq. (8) where $\overline{S} = F \setminus S$

$$\lim_{M\to\infty} -\ell(C|S, D_j) = KL(p(C|S)||q(C|S)) + I(C;\overline{S}|S) + H(C|F). \tag{8}$$

Since in Eq. (8), $KL(p(C|S)||q(C|S))$ will approach zero with a large M, by $I(C;F) = I(C;S) + I(C;\overline{S}|S)$ and Eq. (11), Eq. (8) can be rewritten as Eq. (9) below:

$$\lim\nolimits_{N\to\infty} -\ell(C|S, D_j) \approx H(C) - I(C;S). \tag{9}$$

For each dataset in D, since C is not intervened, we assume the probability of C keeps same and thus $H(C)$ will be the same across different datasets. Then for a subset of features S, if $I(C;S)$ in D_i and $I(C;S)$ in D_j are identical, S carries the equivalent information for predicting C.

Finding Candidate Invariant Sets. By the observations discussed above, for each subset $S \subseteq MB(C)$, Phase 2 tests whether $I^i(C;S)$ in D_i and $I^j(C;S)$ in D_j for $\forall i, j \in 1, \ldots, K$ are identical to identify a candidate invariant S. For computational efficiency, we use the well-known approach in Eq. (10) to approximately calculate $I(C;S)$ [21]

$$I(C;S) = \frac{1}{|S|}\sum\nolimits_{F_i \in S} I(F_i;C), \tag{10}$$

where $|S|$ is the size of the set S. At Step 12, $MI(i)$ is a set which stores mutual information of each feature in S with C in D_i. For data with discrete values, we calculate symmetrical uncertainty [32] instead of $I(F_i;C)$, which is defined by $SU(F_i, C) = \frac{2I(F_i;C)}{H(F_i)+H(C)}$. The advantage of $SU(F_i, C)$ over $I(F_i;C)$ is that $SU(F_i, C)$ normalizes the value of $I(F_i;C)$ between 0 and 1 to compensate for the bias of $I(F_i;C)$ toward features with more values. For data with numeric values, $I(F_i;C) = \frac{1}{2}\log(1-\rho^2)$ where ρ is the Pearson correlation coefficient [7]. At Step 17, avg_{MI} is the average value of $I(C;S)$ over K training datasets.

To determine whether a subset S is an invariant set, for $I^i(C;S)$ in D_i and $I^j(C;S)$ in D_j for $\forall i, j \in 1, \ldots, K$ and $i \neq j$, Steps 18 to 22 need to examine if each of them is identical. To avoid pairwise comparisons, the idea behind Steps 18 to 22 is that if $\exists S \in F$ such that for $\forall i \in 1, \ldots, K$, $I^i(C;S)$ is identical to $\frac{1}{K}\sum_{i=1}^{K} I^i(C;S)$, S is considered as an invariant set. Specially, Steps 18 to 22 calculate whether $\forall i \in \{1, \ldots, K\}$, the mean of $MI(i)$ is identical to avg_{MI} using t-test, and keep the corresponding p-value in the vector ρ. From Steps 23 to 25, if the minimum value in ρ is bigger or equals to α, S is added to $SelFea$ which stores candidate invariant sets.

5.3 Phase 3 (Step 27): Finding the Best S^* from the Candidate Invariant Sets by Using Prediction

In this step, for each subset S in $SelFea$, first, MCFS trains a classifier on each dataset in D independently, then gets K classifiers. Second, MCFS uses the K classifiers for predicting the class labels of data instances in the testing dataset individually. Third, in the testing dataset, the class label of each data instance has the K predicted class labels. When $K = 2$, i.e., D only includes two training datasets, if the two predicted labels are the same, for a data instance in the testing dataset, then it is assigned the predicted class label. If not, the class label of the data instance is randomly assigned. When $K > 2$, MCFS uses the majority voting method. In this case, the class label of each data instance in the testing dataset is the most frequent one among the K predicted class labels. Fourth, by comparing the predicted labels with the groud-truth of labels in the testing dataset, the prediction accuracy of S will be computed. Finally, MCFS outputs the subset S^* with the highest prediction accuracy.

5.4 Time Complexity

The time complexity of MCFS lies in Phase 1 and Phase 2. Phase 1 employs HITON-MB for discovering MBs in each dataset. Given a single dataset, HITON-MB first finds $PC(C)$ (parents and children of C). Then it discovers the spouses of C, for which HITON-MB needs to find the parents and children of each variable in $PC(C)$. In Phase 1, MCFS requires $O(|F||PC(C)|^2 2^{|PC(C)|})$ conditional independence tests (or mutual information computations). In Phase 2, let $\cup MB(C)$ represent the union of MBs of C found from all datasets, the time complexity of MCFS is $O(2^{|\cup MB(C)|})$. Therefore, the overall time complexity of MCFS is $O(2^{max(|\cup MB(C)|, |PC(C)|)})$.

6 Experiments

The goals of our experiments include: (1) evaluating the performance of the proposed MCFS algorithm, in comparison with existing MB discovery methods and other algorithms. We extensively evaluated our method through a series of experiments with synthetic and real world datasets (Sections 6.1 and 6.2); (2) Validating the lower and upper bounds of the invariant set proposed in Section 4.2 along with Theorems 6 and 7 using synthetic data (Section 6.1).

As there are no algorithms specifically developed for causal feature selection with multiple datasets for the experiments, we employ three representative causal feature selection methods, HITON-MB [2], IAMB [28], and STMB [9], two well-known mutual information based feature selection methods, FCBF [32] and mRMR [21], and the ICP algorithm [22].

Except for ICP, which is designed for finding causes from multiple datasets, the other five algorithms are designed for feature selection from a single dataset, so we apply these five algorithms to multiple datasets (for comparing with our proposed algorithm) in three different ways:

- *Use individual feature sets.* We first use an algorithm to select features from each training dataset, then use

[1] Please refer to Section 3.1 in [6] for the details on how to get Eqs. (6) and (7) in this paper.

TABLE 1
Summary of Compared Methods in Our Experiments

ID	Method	Output
1	ICP	Parents (direct causes) of C discovered from multiple training datasets
2	HITON-MB	MB of C found from a training dataset
3	IAMB	MB of C found from a training dataset
4	STMB	MB of C found from a training dataset
5	mRMR	Features selected by mRMR from a training dataset
6	FCBF	Features selected by FCBF from a training dataset
7	∪ HITON-MB	Union of the MB of C found from each training dataset by HITON-MB
8	∩ HITON-MB	Intersection of the MB of C found from each training dataset by HITON-MB
9	∪ IAMB	Union of the MB of C found from each training dataset by IAMB
10	∩ IAMB	Intersection of the MB of C found from each training dataset by IAMB
11	∪ STMB	Union of the MB of C found from each training dataset by STMB
12	∩ STMB	Intersection of the MB of C found from each training dataset by STMB
13	∪ mRMR	Union of the features selected by mRMR from each training dataset
14	∩ mRMR	Intersection of the features selected by mRMR from each training dataset
15	∪ FCBF	Union of the features selected by FCBF from each training dataset
16	∩ FCBF	Intersection of the features selected by FCBF from each training dataset

the set of selected features to train a classifier with the dataset.

- *Use the intersection*. We first select features from each training dataset, then train a classifier with the dataset using the intersection of the feature sets obtained from individual datasets.
- *Use the union*. We first select features from each training dataset, then train a classifier using the union of the feature sets selected from individual datasets.

With all the three approaches, for a test sample, we combine the prediction results by the trained classifiers via majority voting. Together with ICP, the three experiment configurations of applying the five rival algorithms give us 16 different methods for comparison as summarized in Table 1.

To evaluate the performance of the feature selection methods listed in Table 1 for classification, we use three types of classifiers, Naive Bayes (NB), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM). In all tables in this section about experiment results, the best results are highlighted in bold-face, and $\mathbf{A} \pm \mathbf{B}$ denotes that A is the average accuracy and B is the standard deviation.

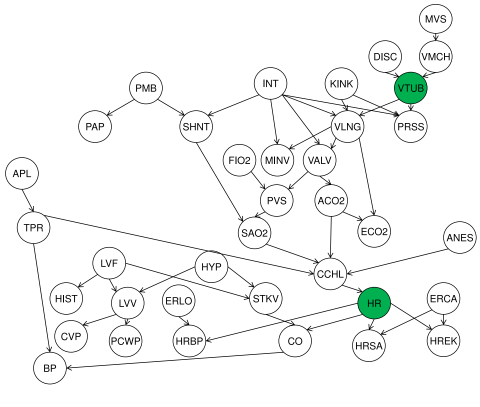

Fig. 2. The ALARM Bayesian network.

TABLE 2
Synthetic Datasets Used in the Experiments

Experiments	Number of training datasets	Number of testing datasets	Number of samples in a dataset
E5-500	5	1	500
E5-2000	5	1	2000
E10-500	10	1	500
E10-2000	10	1	2000

6.1 Experiments on Synthetic Data

Given a benchmark Bayesian network, we are able to read the MB of each variable in the network. Therefore, we can choose the variables in the MB of a target variable to intervene on their values as described in Section 3 to generate training and testing datasets and make the training and testing datasets not identically distributed. Then we apply our MCFS and the other competing methods listed in Table 1 to the training datasets to select features and evaluate the performance of the classifiers trained using the selected features by each method. As mentioned earlier, the experiments in this section with the synthetic data are for evaluating the performance of MCFS in classification (presented in Sections 6.1.1 (1A) and 6.1.2 (2A)), and for validating the bounds proposed in Theorems 6 and 7 (presented in Sections 6.1.1 (1B), 6.1.1 (1C), 6.1.2 (2B), and 6.1.2 (2C))).

We generate the training and testing datasets using a benchmark Bayesian network, the 37-variable A Logical Alarm Reduction Mechanism (ALARM) network [4],[2] as shown in Fig. 2. Two groups of datasets are generated by choosing the variables “HR” and “VTUB” respectively (the green nodes in Fig. 2) as the class attributes. The two variables have the largest sizes of MBs among all variables in the network. When generating an intervention dataset from the ALARM network, we randomly choose the variables in the MB of “HR” (or “VTUB) to intervene on them.

By Table 2, with each of the two chosen class attributes, we conduct two sets of experiments, E5 with 5 training datasets and 1 testing dataset; and E10 with 10 training datasets and 1 testing dataset. In addition, for E5 and E10 respectively, we conduct two experiments, one where each dataset contains 500 samples and another one where each dataset contains 2,000 samples. That is, for each of the two chosen class attributes, we conduct 4 experiments in total, E5-500, E5-2000, E10-500 and E10-2000. Each experiment is carried out for 5 runs, and for each experiment we compute and report the average prediction accuracy (i.e., the ratio of the number of correct predictions and total number of testing samples).

In the experiments, the significance level α for conditional independence tests for HITON-MB, IAMB, STMB, and MCFS is set to 0.01, while the threshold for FCBF is set to 0.01. Since the MBs of “HR” and “VTUB” are known in the network, the user-defined parameter k of mRMR is set to the size of the MB of “HR” and “VTUB”, respectively.

6.1.1 Experiment Results on “HR”

“HR” has the largest MB among all variables in the network and it has three distinct class labels (multiple classes). Its MB includes one parents, four children, and three spouses.

[2] Refer to www.bnlearn.com/bnrepository for the details of the network.

TABLE 3
Prediction Accuracy of MCFS Against Its Rivals When “HR” Is the Target (In the Table, (X)* Denotes that an Algorithm Succeeded by Returning a Non-Empty Feature Set for X Times Out of the Full 5 Runs)

Experiments		HITON-MB	IAMB	STMB	mRMR	FCBF	ICP	MCFS
E5-500	NB	0.8486 ± 0.0879	0.8420 ± 0.0673	0.8324 ± 0.1090	0.8164 ± 0.0999	0.8668 ± 0.0527	0.9133 ± 0.011(3)*	**0.9200 ± 0.0265**
	KNN	0.6864 ± 0.2708	0.7028 ± 0.2422	0.6980 ± 0.2924	0.8064 ± 0.0512	0.7836 ± 0.1340	0.9133 ± 0.011(3)*	**0.9276 ± 0.0352**
	SVM	0.7804 ± 0.0585	0.7628 ± 0.0599	0.7780 ± 0.0607	0.7716 ± 0.0596	0.8048 ± 0.0987	0.9093 ± 0.0127(3)*	**0.9112 ± 0.0206**
E5-2000	NB	0.8583 ± 0.1022	0.8520 ± 0.1032	0.8583 ± 0.1008	0.8446 ± 0.0888	0.8647 ± 0.1041	0.7675 ± 0(1)*	**0.9172 ± 0.0315**
	KNN	0.7134 ± 0.1448	0.6879 ± 0.1400	0.6960 ± 0.1733	0.7771 ± 0.0933	0.8106 ± 0.1211	0.7675 ± 0(1)*	**0.9346 ± 0.0414**
	SVM	0.7706 ± 0.1091	0.7793 ± 0.1070	0.7753 ± 0.1089	0.8155 ± 0.1183	0.8057 ± 0.1350	0.7675 ± 0(1)*	**0.9322 ± 0.0396**
E10-500	NB	0.8916 ± 0.0325	0.8864 ± 0.0439	0.8732 ± 0.0487	0.8684 ± 0.0566	0.8796 ± 0.0537	0.8880 ± 0.0113(2)*	**0.9168 ± 0.0386**
	KNN	0.8520 ± 0.1029	0.8332 ± 0.1237	0.8288 ± 0.1492	0.8460 ± 0.0578	0.8744 ± 0.0405	0.8880 ± 0.0113(2)*	**0.9244 ± 0.0447**
	SVM	0.7553 ± 0.1762	0.7477 ± 0.1959	0.7519 ± 0.1773	0.7746 ± 0.1443	0.7562 ± 0.1568	0.8922 ± 0.0032(2)*	**0.9498 ± 0.0183**
E10-2000	NB	0.8452 ± 0.0682	0.8457 ± 0.0672	0.8488 ± 0.0651	0.8494 ± 0.0592	0.8552 ± 0.0595	0 ± 0(0)*	**0.9158 ± 0.0349**
	KNN	0.8559 ± 0.1078	0.8504 ± 0.1215	0.8588 ± 0.1036	0.8387 ± 0.0511	0.8221 ± 0.0783	0 ± 0(0)*	**0.9284 ± 0.0380**
	SVM	0.7069 ± 0.1616	0.7244 ± 0.1811	0.7210 ± 0.1801	0.8375 ± 0.0700	0.8229 ± 0.1262	0 ± 0(0)*	**0.9403 ± 0.0342**

(1A) Performance of MCFS versus its rivals. In this part, we compare MCFS with the first six methods shown in Table 1 in terms of their prediction accuracy using the features selected by them (Table 3), the number of features selected from the true MB of “HR”, and their running time (Table 6).

Table 3 shows that in all cases, MCFS is significantly better than all its rivals, including ICP, HITON-MB, IAMB, STMB, mRMR and FCBF, when the rivals only simply select features from each dataset and train a classifier individually. Note that for a feature selection algorithm, if it returns an empty set on a multiple dataset, we consider that the algorithm fails on the dataset and the corresponding prediction accuracy is 0.

In Experiment E5-500 (with 5 training datasets and 500 samples in each dataset), ICP returns a non-empty feature set in three out of five runs (see Table 3). The only feature selected by ICP in each of the three runs is “CCHL”, i.e., the parent of “HR”. When the number of data samples of each training dataset is set to 2,000, the only successful run of ICP returns two features, the parent and one child of “HR”. In Experiment E10-500 (with 10 training datasets and 500 samples each), ICP succeeds in two out of the five runs, and returns the parent of “HR” in one run and the parent and one child of “HR” in the other run. In Experiment E10-2000, ICP fails in all five runs without returning any features. Our observation shows that ICP does not necessarily guarantee to find the parents of a given target from multiple datasets.

From Table 3, the performance of HITON-MB, IAMB, STMB, mRMR, and FCBF seems to be competitive overall, but our algorithm MCFS still achieves higher prediction accuracy in all experiments. Using the KNN and SVM classifiers, when the number of datasets is set to 5, mRMR and FCBF achieve higher prediction accuracy than HITON-MB, IAMB, and STMB. On computational efficiency, from Table 6, ICP spends much more time than all the other algorithms. Compared to HITON-MB, IAMB, STMB, mRMR, and FCBF, MCFS has a reasonable running time and selects fewer features than these five algorithms.

In summary, from Table 3, the proposed MCFS algorithm is able to deal with the situation better than the other six algorithms where the training and testing datasets are not identically distributed.

(1B) Performance of MCFS, methods using intersections of feature sets, and the true parents of “HR”. In Section 4, Theorems 6 and 7 state that the set of all parents of the class attribute is the minimal and invariance subset across multiple interventional datasets when the class attribute is not manipulated. From the ALARM network, we can read the parents of “HR”. Thus, in this part, we compare the prediction accuracy of the true parent of “HR”, the set of features selected by MCFS, and the intersection of the sets selected by each other five algorithms on each training dataset, i.e., methods ∩ HITON-MB, ∩ IAMB, ∩ STMB, ∩ mRMR, and ∩ FCBF.

In Table 4, “TrueParent” denotes the ground-truth parents of “HR” in the ALARM network, that is, “CCHL”. We use the ground-truth parent of “HR” to train a classifier on each training dataset, and use majority voting to combine the prediction results on testing data attained.

From Table 4, we can see that MCFS achieves higher prediction accuracy than the other five methods using the intersections of selected feature sets (i.e., ∩ HITON-MB, ∩ IAMB, ∩ STMB, ∩ mRMR, and ∩ FCBF), and MCFS achieves similar prediction accuracy as that using the true parent as the feature. For ∩ HITON-MB, ∩ IAMB, ∩ STMB, ∩ mRMR, and ∩ FCBF, only the intersections of features selected by mRMR and FCBF from each training dataset are not empty. When the number of data samples is 2,000, we can see that the prediction accuracy of the true parent of “HR” is much higher than that of ∩ mRMR and ∩ FCBF. When the number of data samples is 500, the prediction accuracy of the true parent of “HR” is higher than ∩ mRMR and is very competitive with ∩ FCBF. When ∩ HITON-MB, ∩ IAMB, or ∩ STMB outputs a non-empty feature set, the performance of ∩ HITON-MB, ∩ IAMB, and ∩ STMB is not inferior to HITON-MB, IAMB, and STMB in Table 3.

By comparing Table 3 with Table 4, we can see that using the intersections of features selected from multiple datasets by mRMR and FCBF (i.e., ∩ mRMR and ∩ FCBF) gets higher prediction accuracy than using features selected by mRMR and FCBF. Moreover, in the experiments, we observe that the parent of “HR” is included in the output of all of ∩ HITON-MB, ∩ IAMB, ∩ STMB, ∩ mRMR, and ∩ FCBF. Especially, when the output of ∩ HITON-MB, ∩ IAMB, or ∩ STMB is not empty, it only includes the parent of “HR”. In summary, Table 4 illustrates that the different methods achieve similar prediction performance when they all use the parent set, indicating that the parent set is the invariant set.

TABLE 4
Prediction Accuracy of MCFS Against the Intersections of Features Selected by Its Rivals When "HR" Is the Target (In the Table, (X) * Denotes that an Algorithm Succeeded by Returning a Non-Empty Feature Set for X Times Out of the Full 5 Runs)

Experiments		∩ HITON-MB	∩ IAMB	∩ STMB	∩ mRMR	∩ FCBF	TrueParent	MCFS
E5-500	NB	0.9133 ± 0.0100(3)*	0 ± 0(0)*	0.9133 ± 0.0100(3)*	0.8940 ± 0.0313	0.9116 ± 0.0132	0.9088 ± 0.0100	**0.9200 ± 0.0265**
	KNN	0.9133 ± 0.0100(3)*	0 ± 0(0)*	0.9133 ± 0.0100(3)*	0.8988 ± 0.0198	0.9092 ± 0.0110	0.9088 ± 0.0100	**0.9276 ± 0.0352**
	SVM	0.9093 ± 0.0127(3)*	0 ± 0(0)*	0.9093 ± 0.0127(3)*	0.9012 ± 0.0134	0.9044 ± 0.0144	0.9040 ± 0.0123	**0.9112 ± 0.0206**
E5-2000	NB	0.8498 ± 0.1030	0.8513 ± 0.0727(3)*	0.8498 ± 0.1030	0.8430 ± 0.0911	0.8498 ± 0.1030	0.8955 ± 0.0071	**0.9172 ± 0.0315**
	KNN	0.8514 ± 0.0994	0.8513 ± 0.0727(3)*	0.8461 ± 0.1112	0.8191 ± 0.0897	0.7825 ± 0.1638	0.8955 ± 0.0071	**0.9346 ± 0.0414**
	SVM	0.8665 ± 0.0658	0.8513 ± 0.0727(3)*	0.8665 ± 0.0658	0.8413 ± 0.1004	0.8437 ± 0.1019	0.8955 ± 0.0071	**0.9322 ± 0.0396**
E10-500	NB	0.8884 ± 0.0114	0.8850 ± 0.0156(2)*	0.8864 ± 0.0119	0.8704 ± 0.0482	0.8940 ± 0.0248	0.8864 ± 0.0119	**0.9168 ± 0.0386**
	KNN	0.8864 ± 0.0119	0.8813 ± 0.0127(2)*	0.8864 ± 0.0119	0.8564 ± 0.0564	0.8936 ± 0.0240	0.8864 ± 0.0119	**0.9244 ± 0.0447**
	SVM	0.8669 ± 0.0664	0.8962 ± 0.0354(2)*	0.8658 ± 0.0657	0.7947 ± 0.1592	0.8337 ± 0.1142	0.8864 ± 0.0119	**0.9498 ± 0.0183**
E10-2000	NB	0.9043 ± 0.0059(3)*	0.8975 ± 0(1)*	0.9015 ± 0.0071(4)*	0.8478 ± 0.0598	0.8587 ± 0.0673	0.9008 ± 0.0064	**0.9158 ± 0.0349**
	KNN	0.9042 ± 0.0058(3)*	0.8975 ± 0(1)*	0.9015 ± 0.0071(4)*	0.8352 ± 0.0719	0.8794 ± 0.0488	0.9008 ± 0.0064	**0.9284 ± 0.0380**
	SVM	0.9042 ± 0.0058(3)*	0.8935 ± 0(1)*	0.9013 ± 0.0068(4)*	0.8521 ± 0.0904	0.8536 ± 0.1144	0.9008 ± 0.0064	**0.9403 ± 0.0342**

TABLE 5
Prediction Accuracy of MCFS Against Unions of Features Selected Its Rivals on "HR"

Experiments		∪ HITON-MB	∪ IAMB	∪ STMB	∪ mRMR	∪ FCBF	TrueMB	MCFS
E5-500	NB	0.8056 ± 0.1251	0.8064 ± 0.1257	0.8048 ± 0.1245	0.7752 ± 0.1073	0.8304 ± 0.0920	0.8056 ± 0.1251	**0.9200 ± 0.0265**
	KNN	0.7432 ± 0.1231	0.7812 ± 0.1371	0.7404 ± 0.1174	0.6728 ± 0.0740	0.7784 ± 0.0775	0.7876 ± 0.1391	**0.9276 ± 0.0352**
	SVM	0.7692 ± 0.0407	0.7528 ± 0.0530	0.7644 ± 0.0492	0.7216 ± 0.0052	0.7576 ± 0.0557	0.8008 ± 0.0772	**0.9112 ± 0.006**
E5-2000	NB	0.8455 ± 0.0978	0.8401 ± 0.0960	0.8441 ± 0.0960	0.8113 ± 0.0859	0.8546 ± 0.0999	0.8444 ± 0.0975	**0.9172 ± 0.0315**
	KNN	0.7920 ± 0.0929	0.7614 ± 0.1096	0.7860 ± 0.0863	0.7479 ± 0.1273	0.8051 ± 0.1154	0.7811 ± 0.0986	**0.9346 ± 0.0414**
	SVM	0.7850 ± 0.1444	0.7847 ± 0.1333	0.7750 ± 0.1381	0.7926 ± 0.1052	0.8061 ± 0.1381	0.7854 ± 0.1445	**0.9322 ± 0.0396**
E10-500	NB	0.8892 ± 0.0425	0.8684 ± 0.0483	0.8660 ± 0.0479	0.8400 ± 0.0803	0.8748 ± 0.0466	0.8776 ± 0.0468	**0.9168 ± 0.0386**
	KNN	0.8488 ± 0.1009	0.8664 ± 0.0924	0.8076 ± 0.1000	0.7544 ± 0.1054	0.7948 ± 0.0867	0.8600 ± 0.0801	**0.9244 ± 0.0447**
	SVM	0.7577 ± 0.1686	0.7548 ± 0.1728	0.7432 ± 0.1755	0.7724 ± 0.1302	0.7674 ± 0.1523	0.8052 ± 0.1301	**0.9498 ± 0.0183**
E10-2000	NB	0.8276 ± 0.0710	0.8317 ± 0.0772	0.8348 ± 0.0767	0.8273 ± 0.0661	0.8293 ± 0.0804	0.8311 ± 0.0765	**0.9158 ± 0.0349**
	KNN	0.8593 ± 0.0913	0.8592 ± 0.0934	0.8430 ± 0.0881	0.8106 ± 0.0478	0.8348 ± 0.0623	0.8581 ± 0.0922	**0.9284 ± 0.0380**
	SVM	0.7045 ± 0.1166	0.7130 ± 0.1332	0.7293 ± 0.0588	0.7851 ± 0.0258	0.8409 ± 0.0621	0.7165 ± 0.0954	**0.9403 ± 0.0342**

TABLE 6
Running Time (in Seconds) and Number of Selected Features on "HR"

Experiments		HITON-MB	IAMB	STMB	mRMR	FCBF	ICP	MCFS
E5-500	Running time	1.34 ± 0.9	0.72 ± 0.3	1.8 ± 0.4	0.24 ± 0.05	0.06 ± 0.01	6 ± 0	4.2 ± 0.4
	Number of selected features	4.2 ± 0.4	3 ± 0	5 ± 0	8 ± 0	4.6 ± 0.5	1 ± 0(3)*	3 ± 1
E5-2000	Running time	2.6 ± 0.9	1.8 ± 0.4	4 ± 0.7	0.32 ± 0.04	0.18 ± 0.04	26 ± 13	5.8 ± 1.6
	Number of selected features	4.8 ± 1.3	4.2 ± 0.4	5.4 ± 1.5	8 ± 0	3.2 ± 0.4	2 ± 0(1)*	3.6 ± 2.4
E10-500	Running time	2.6 ± 0.8	1 ± 0	3.8 ± 0.8	0.3 ± 0	0.2 ± 0	18 ± 5	9 ± 1.7
	Number of selected features	4.6 ± 0.9	3 ± 0	5.4 ± 1.5	8 ± 0	4.6 ± 0.5	1.5 ± 0.7(2)	3 ± 2
E10-2000	Running time	4.2 ± 1	2.6 ± 0.5	6.4 ± 2.2	0.6 ± 0	0.3 ± 0	23.4 ± 12	12.4 ± 5
	Number of selected features	4.6 ± 1.1	4.2 ± 0.8	5.2 ± 1.5	8 ± 0	3 ± 0.7	0 ± 0	2 ± 1.4

(1C) Performance of MCFS, methods using unions of feature sets, and the true MB of "HR". According to Theorem 8, when the feature interverion conforms to the conservative rule, the union of feature sets selected by each MB discovery algorithm from all training datasets equals to the true MB. Thus, to validate Theorems 8, we compare the prediction accuracy of using the true MB of "HR", the features outputed by MCFS, ∪ HITON-MB, ∪ IAMB, ∪ STMB, ∪ mRMR, and ∪ FCBF.

From Table 5, first, we can see that MCFS is significantly better than the true MB and the other five methods. This indicates that with multiple interventional datasets, the true MB of the class attribute may not be optimal for feature selection. Second, referring to Table 4, using the true parent of "HR" gets significantly better prediction accuracy than using the true MB of "HR". Thus, with multiple interventional datasets, when we do not know which features are intervened, the parents of the class attribute may be a more reliable subset for prediction. Third, ∪ HITON-MB, ∪ IAMB, and ∪ STMB achieves an accuracy very close to that of the true MB of "HR". This further validates Theorem 8, which demonstrates that when the feature interventions is conservative, the union of the MB of the class attribute discovered from

TABLE 7
Prediction Accuracy of MCFS Against Its Rivals on “VTUB” (In the Table, (X)* Denotes that an Algorithm Succeeded by Returning a Non-Empty Feature Set for X Times Out of the Full 5 Runs)

Experiments		HITON-MB	IAMB	STMB	mRMR	FCBF	ICP	MCFS
E5-500	NB	0.8440 ± 0.0699	0.8164 ± 0.1756	0.8088 ± 0.1555	0.7484 ± 0.1511	0.9200 ± 0.0642	0 ± 0(0)*	**0.9824 ± 0.0103**
	KNN	0.8556 ± 0.1104	0.8432 ± 0.1761	0.8636 ± 0.1653	0.8388 ± 0.0840	0.8824 ± 0.0944	0 ± 0(0)*	**0.9812 ± 0.0101**
	SVM	0.7872 ± 0.1625	0.7272 ± 0.2116	0.6280 ± 0.3027	0.6016 ± 0.3995	0.5856 ± 0.4221	0 ± 0(0)*	**0.8232 ± 0.2339**
E5-2000	NB	0.9184 ± 0.0424	0.9165 ± 0.0425	0.9235 ± 0.0480	0.5795 ± 0.3972	0.9258 ± 0.0406	0 ± 0(0)*	**0.9711 ± 0.0018**
	KNN	0.8296 ± 0.1698	0.8678 ± 0.1012	0.8230 ± 0.1281	0.7814 ± 0.1733	0.8987 ± 0.0922	0 ± 0(0)*	**0.9715 ± 0.0024**
	SVM	0.5056 ± 0.3124	0.5090 ± 0.3773	0.5051 ± 0.3116	0.5010 ± 0.3635	0.5063 ± 0.3822	0 ± 0(0)*	**0.7444 ± 0.2963**
E10-500	NB	0.9072 ± 0.0426	0.8324 ± 0.1711	0.8620 ± 0.1175	0.8700 ± 0.0578	0.9036 ± 0.0588	0 ± 0(0)*	**0.9752 ± 0.0033**
	KNN	0.8896 ± 0.0633	0.8436 ± 0.1686	0.9012 ± 0.0816	0.7864 ± 0.1527	0.8356 ± 0.0899	0 ± 0(0)*	**0.9752 ± 0.0033**
	SVM	0.4852 ± 0.2578	0.4976 ± 0.2435	0.4924 ± 0.2422	0.4776 ± 0.2533	0.4544 ± 0.2350	0 ± 0(0)*	**0.7040 ± 0.3437**
E10-2000	NB	0.8702 ± 0.1028	0.9067 ± 0.0695	0.8733 ± 0.0948	0.8851 ± 0.0848	0.9105 ± 0.0713	0 ± 0(0)*	**0.9685 ± 0.0034**
	KNN	0.9043 ± 0.0857	0.8775 ± 0.1442	0.8906 ± 0.0872	0.8765 ± 0.0771	0.9449 ± 0.0427	0 ± 0(0)*	**0.9685 ± 0.0034**
	SVM	0.5308 ± 0.1451	0.4720 ± 0.2216	0.6302 ± 0.2492	0.6966 ± 0.0985	0.5399 ± 0.2845	0 ± 0(0)*	**0.8338 ± 0.1294**

TABLE 8
Prediction Accuracy of MCFS Against Intersections of Features Selected Its Rivals on “VTUB” (In the Table, (X)* Denotes that an Algorithm Succeeded by Returning a Non-Empty Feature Set for X Times Out of the Full 5 Runs)

Experiments		∩ HITON-MB	∩ IAMB	∩ STMB	∩ mRMR	∩ FCBF	TureParent	MCFS
E5-500	NB	0.9100 ± 0.0122(3)*	0.904 ± 0(1)*	0.9160 ± 0.0025(3)*	0.9028 ± 0.1241	0.9220 ± 0.1312	0.9808 ± 0.0103	**0.9824 ± 0.0103**
	KNN	0.8720 ± 0.0537(3)*	0.6904 ± 0(1)*	0.9160 ± 0.0025(3)*	0.9020 ± 0.1213	0.9216 ± 0.1287	0.9808 ± 0.0103	**0.9812 ± 0.0101**
	SVM	0.7853 ± 0.2298(3)*	0.5200 ± 0(1)*	0.7913 ± 0.2350(3)*	0.6588 ± 0.3693	0.6276 ± 0.3787	0.5276 ± 0.4606	**0.8232 ± 0.2339**
E5-2000	NB	0.8324 ± 0.2537	0.8101 ± 0.2877(4)*	0.9295 ± 0.0697	0.7467 ± 0.3785	0.8907 ± 0.0904	**0.9711 ± 0.0018**	**0.9711 ± 0.0018**
	KNN	0.8697 ± 0.1692	0.8611 ± 0.1863(4)*	0.8940 ± 0.1475	0.9093 ± 0.0641	0.8451 ± 0.1327	0.9711 ± 0.0018	**0.9715 ± 0.0024**
	SVM	0.5376 ± 0.3468	0.4756 ± 0.3375(4)*	0.5142 ± 0.3871	0.4988 ± 0.3734	0.5123 ± 0.3836	0.4797 ± 0.4407	**0.7444 ± 0.2963**
E10-500	NB	0.4740 ± 0.6265(2)*	0 ± 0(0)*	0.5620 ± 0.5006(2)*	0.8608 ± 0.1887	0.9452 ± 0.0577	0.9724 ± 0.0038	**0.9752 ± 0.0033**
	KNN	0.4740 ± 0.6265(2)*	0 ± 0(0)*	0.5620 ± 0.5006(2)*	0.8704 ± 0.1938	0.9612 ± 0.0266	**0.9752 ± 0.0033**	**0.9752 ± 0.0033**
	SVM	0.6120 ± 0.4299(2)*	0 ± 0(0)*	0.4773 ± 0.3832(2)*	0.5292 ± 0.2827	0.6248 ± 0.2973	0.6960 ± 0.3544	**0.7040 ± 0.3437**
E10-2000	NB	0.8266 ± 0.1809	0.6681 ± 0.3688(4)*	0.9623 ± 0.0163	0.9582 ± 0.0188	0.9570 ± 0.0214	**0.9685 ± 0.0034**	**0.9685 ± 0.0034**
	KNN	0.8253 ± 0.1824	0.6643 ± 0.3652(4)*	0.9533 ± 0.0210	0.8860 ± 0.1185	0.9348 ± 0.0771	**0.9685 ± 0.0034**	**0.9685 ± 0.0034**
	SVM	0.4484 ± 0.3063	0.4495 ± 0.3665(4)*	0.4571 ± 0.3127	0.4665 ± 0.2521	0.4769 ± 0.2613	0.4665 ± 0.3152	**0.8338 ± 0.1294**

each interventional dataset equals to the true MB of the class attribute. ∪ mRMR achieves the worst result as shown in Table 5. The explanation is that it is hard to select the user-defined parameter k for mRMR to select the features to achieve desirable prediction accuracy.

6.1.2 Results on “VTUB”

“VTUB” has the second largest MB among all features in the ALARM network and has four distinct class labels (multiple classes). Its MB consists of two parents, two children and two spouses.

(2A) Performance of MCFS versus its rivals. From Table 7, we can see that MCFS is significantly better than the other six algorithms. For ICP, it returns an empty set on all five runs in all cases. Thus, this illustrates that the idea of ICP for finding parents of a given target from multiple datasets does not always work well. Meanwhile, using NB and KNN, FCBF achieves higher prediction accuracy than HITON-MB, IAMB, STMB, and mRMR. Compared to Tables 3, Tables 7 illustrates that FCBF also achieves satisfactory results. On computational efficiency, from Table 10, ICP is still the slowest one among the seven algorithms, although ICP uses the lasso method as a preprocess step. FCBF is faster than the other six algorithms. Compared to HITON-MB, IAMB, STMB, mRMR, and FCBF, MCFS has a reasonable running time and selects fewer features than these five algorithms. In summary, Tables 7 and 10 shows that MCFS is better than the other six algorithms to deal with multiple interventional datasets.

(2B) Performance of MCFS, methods using intersections of feature sets, and the true parents of “VTUB”. Table 8 illustrates that MCFS achieves highest prediction accuracy among the other six methods. Meanwhile, the se of true parents of “VTUB” achieves the same prediction accuracy as MCFS in 4 out of 8 cases, while in the other 4 cases, the prediction accuracy of the true parents of “VTUB” is almost the same as that of MCFS. However, it is a difficult problem to find the parents of a given target in data. For example, ICP is customized to discover parents of a given target from multiple interventional datasets, but Tables 3 and 7 illustrate that ICP always fails.

Table 8 shows that only ∩ mRMR and ∩ FCBF output a non-empty set over five runs. When the number of training datasets is 10, we can see that the intersections of features selected by FCBF achieve satisfactory prediction accuracy no matter for using NB or KNN. Compared to Table 7, Table 8 demonstrates that ∩ mRMR and ∩ FCBF get higher prediction accuracy than mRMR and FCBF. This further confirms that the set of parents of the class attribute is reliable for prediction with multiple interventional datasets.

TABLE 9
Prediction Accuracy of MCFS Against Unions of Features Selected Its Rivals on "VTUB"

Experiments		∪ HITON-MB	∪ IAMB	∪ STMB	∪ mRMR	∪ FCBF	TureMB	MCFS
E5-500	NB	0.8448 ± 0.0771	0.8448 ± 0.0771	0.8520 ± 0.0771	0.6792 ± 0.1959	0.7912 ± 0.1444	0.8440 ± 0.0875	**0.9824 ± 0.0103**
	KNN	0.8580 ± 0.0979	0.8664 ± 0.0891	0.8848 ± 0.0741	0.7396 ± 0.1747	0.8400 ± 0.0730	0.8784 ± 0.0875	**0.9812 ± 0.0101**
	SVM	0.4769 ± 0.2613	0.6552 ± 0.3235	0.5976 ± 0.3987	0.6920 ± 0.3325	0.7268 ± 0.3395	0.5172 ± 0.4218	**0.8232 ± 0.2339**
E5-2000	NB	0.9064 ± 0.0766	0.7445 ± 0.3388	0.8901 ± 0.0911	0.5377 ± 0.3961	0.7718 ± 0.2792	0.9064 ± 0.0766	**0.9711 ± 0.0018**
	KNN	0.7941 ± 0.1503	0.7320 ± 0.1911	0.7190 ± 0.2061	0.6689 ± 0.2331	0.8836 ± 0.0898	0.7941 ± 0.1503	**0.9715 ± 0.0024**
	SVM	0.4591 ± 0.3096	0.5190 ± 0.3059	0.5469 ± 0.2885	0.5180 ± 0.3804	0.5048 ± 0.3833	0.4592 ± 0.3098	**0.7444 ± 0.2963**
E10-500	NB	0.8480 ± 0.0887	0.8360 ± 0.1267	0.8896 ± 0.0630	0.6896 ± 0.1467	0.8204 ± 0.0505	0.8864 ± 0.0706	**0.9752 ± 0.0033**
	KNN	0.6976 ± 0.2978	0.7612 ± 0.1680	0.6424 ± 0.3122	0.5864 ± 0.1852	0.6148 ± 0.2744	0.7308 ± 0.2333	**0.9752 ± 0.0033**
	SVM	0.4180 ± 0.2337	0.4808 ± 0.2964	0.4444 ± 0.1849	0.4220 ± 0.2438	0.4120 ± 0.2704	0.4492 ± 0.2460	**0.7040 ± 0.3437**
E10-2000	NB	0.8887 ± 0.0908	0.8978 ± 0.0750	0.7716 ± 0.2381	0.7482 ± 0.2996	0.9004 ± 0.0944	0.8887 ± 0.0908	**0.9685 ± 0.0034**
	KNN	0.8789 ± 0.0865	0.8005 ± 0.1093	0.6824 ± 0.1380	0.6113 ± 0.1946	0.9189 ± 0.0535	0.8789 ± 0.0865	**0.9685 ± 0.0034**
	SVM	0.6030 ± 0.2126	0.5620 ± 0.1780	0.5948 ± 0.1824	0.5746 ± 0.1258	0.6163 ± 0.2132	0.6021 ± 0.2115	**0.8338 ± 0.1294**

TABLE 10
Running Time (in Seconds) and Number of Selected Features on "VTUB"

Experiments		HITON-MB	IAMB	STMB	mRMR	FCBF	ICP	MCFS
E5-500	Running time	2 ± 0	0.36 ± 0.05	2 ± 0	0.1 ± 0	0.08 ± 0.01	24 ± 15	2.8 ± 1.3
	Number of selected features	4.2 ± 0.8	2.4 ± 0.5	5 ± 1.4	6 ± 0	5 ± 0	0 ± 0	4 ± 0.7
E5-2000	Running time	3 ± 0.7	1 ± 0	3.4 ± 0.5	0.2 ± 0	0.1 ± 0	89.4 ± 38	3.6 ± 0.5
	Number of selected features	4.8 ± 0.4	4 ± 0	6.4 ± 1.1	6 ± 0	3.6 ± 0.5	0 ± 0	2.8 ± 0.8
E10-500	Running time	2.6 ± 0.5	1 ± 0	3.2 ± 0.4	0.2 ± 0	0.2 ± 0	53.4 ± 17	5.8 ± 1.9
	Number of selected features	3.6 ± 0.5	2.2 ± 0.4	4.4 ± 0.5	6 ± 0	5.4 ± 0.5	0 ± 0	3 ± 1
E10-2000	Running time	4.4 ± 0.9	2 ± 0	5.6 ± 0.5	0.4 ± 0	0.2 ± 0	254 ± 78	5.6 ± 1.4
	Number of selected features	4.4 ± 0.8	4 ± 0	5.4 ± 1.5	6 ± 0	3 ± 0	0 ± 0	3 ± 0.7

TABLE 11
Impact of α on Prediction Accuracy of HITON-MB, IAMB, STMB, and MCFS

Prediction accuracy (A/B) on "HR" using α=0.01 (A) or α=0.05 (B)										
Experiments		E5-500/E5-2000					E10-500/E10-2000			
		HITON-MB	IAMB	STMB	MCFS		HITON-MB	IAMB	STMB	MCFS
500	NB	0.8486/0.8624	0.8420/0.8424	0.8324/0.8324	0.9200/0.9200		0.8916/0.8784	0.8864/0.8868	0.8732/0.8760	0.9168/0.9176
	KNN	0.6864/0.7624	0.7028/0.7228	0.6980/0.7852	0.9276/0.9276		0.8520/0.8544	0.8332/0.8288	0.8288/0.8536	0.9244/0.9256
	SVM	0.7804/0.7804	0.7628/0.7624	0.7780/0.7892	0.9112/0.9112		0.7553/0.7565	0.7477/0.7473	0.7519/0.7399	0.9498/0.9556
2000	NB	0.8583/0.8583	0.8520/0.8582	0.8583/0.8558	0.9172/0.9172		0.8452/0.8499	0.8457/0.8512	0.8488/0.8481	0.9158/0.9163
	KNN	0.7134/0.7139	0.6879/0.7640	0.6960/0.7081	0.9346/0.9346		0.8559/0.8528	0.8504/0.8508	0.8588/0.8519	0.9284/0.9291
	SVM	0.7706/0.7706	0.7793/0.7804	0.7753/0.7779	0.9322/0.9343		0.7069/0.7451	0.7244/0.7201	0.7210/0.7516	0.9404/0.9259
Prediction accuracy (A/B) on "'VentTube" using α=0.01 (A) or α=0.05 (B)										
Experiments		E5-500/E5-2000					E10-500/E10-2000			
		HITON-MB	IAMB	STMB	MCFS		HITON-MB	IAMB	STMB	MCFS
500	NB	0.8440/0.8428	0.8164/0.8244	0.8088/0.8524	0.9824/0.9824		0.9072/0.9056	0.9685/0.8256	0.8620/0.8816	0.9752/0.9752
	KNN	0.8556/0.8712	0.8432/0.8384	0.8636/0.8708	0.9812/0.9812		0.8896/0.8944	0.8436/0.8540	0.9012/0.8832	0.9752/0.9752
	SVM	0.7872/0.7678	0.7272/0.7232	0.6280/0.5828	0.8232/0.8232		0.4852/0.5608	0.4976/0.4932	0.4924/0.4784	0.7040/0.7048
2000	NB	0.9184/0.9188	0.9165/0.9165	0.9235/0.8406	0.9711/0.9711		0.8702/0.8697	0.9067/0.9067	0.8733/0.9003	0.9685/0.9685
	KNN	0.8296/0.8250	0.8678/0.8581	0.8230/0.7435	0.9715/0.9715		0.9043/0.8994	0.8775/0.8784	0.8906/0.8797	0.9685/0.9685
	SVM	0.5056/0.5099	0.5090/0.5090	0.5051/0.4895	0.7444/0.7444		0.5308/0.5209	0.4720/0.4941	0.6302/0.5821	0.8338/0.8338

(2C) Performance of MCFS, methods using unions of feature sets, and the true MB of "VTUB". Table 9 shows that the prediction accuracy of MCFS is significantly better than that of the true MB of "VTUB". This further confirms that the true MB of the class attribute in a multiple interventional dataset may not be optimal for classification. Referring to Table 8, the set of true parents of "VTUB" gets significantly higher accuracy than the set of true MB of "VTUB". Thus, with multiple interventional datasets, the parents of the class attribute may be more reliable than its MB for prediction.

Additionally, we can see that ∪ HITON-MB gets very close accuracy with the true MB of "VTUB", while ∪ mRMR still gets the worst prediction accuracy in Table 9.

6.1.3 Impact of the Parameter α

Table 11 reports the impact of the significance level α for conditional independence tests for HITON-MB, IAMB, STMB, and MCFS. From Table 11, we can see that α almost has no impact on the performance of MCFS. Meanwhile, for HITON-MB, IAMB, and STMB, in most cases, with a different value of α, the prediction accuracy of HITON-MB, IAMB, and STMB is able to keep stable, thus α does not have a significant influence on these algorithms.

6.1.4 Time Complexity of the Rivals of MCFS

The time complexity of MCFS, FCBF, mRMR, HITON-MB, IAMB, and STMB is measured in the number of conditional

TABLE 12
Variables in the Educational Attainment Data Set and Their Meanings

Variable	Meaning
education	Years of education completed (target variable, binarized to completed a BA or not in this paper)
gender	Student gender, male or female
ethnicity	Afam/Hispanic/Other
score	Base year composite test score. (These are achievement tests given to high school seniors in the sample)
fcollege	Father is a college graduate or not
mcollege	Mother is a colllege graduate or not
home	Family owns a house or not
urban	School in urban area or not
unemp	County unempolyment rate in 1980
wage	State hourly wage in manufacturing in 1980
distance	Distance to the nearest 4-year college
tuition	Avg. state 4-year college tuition in \$1000's
income	Family income >\$25,000 per year or not
region	Student in the western states or other states

independence tests (or mutual information computations) executed. For IAMB, the average time complexity is $O(|F||MB(C)|)$ and the worst case time complexity is $O(|F|^2)$ with $|MB(C)| = |F|$. STMB also finds $PC(C)$ first, then discovers spouses. Different from HITON-MB, STMB finds spouses from $F \setminus PC(C)$, instead of all parents and children of variables of $PC(C)$. Then the overall time complexity of STMB is $O(|PC(C)||F \setminus PC(C)|2^{|PC(C)|})$. However, STMB is not able to deal with datasets with high-dimensionality and small number of samples. Since the user-defined parameter k of mRMR is set to the size of $MB(C)$, mRMR and FCBF need $O(|MB(C)|^2)$ pairwise mutual information computations, and thus the time complexity of FCBF and mRMR is not exponential with the size of $MB(C)$. However, it is hard to select a suitable value of k for mRMR and FCBF, and they are not specifically designed for MB discovery.

In summary, we can see that FCBF, and mRMR in general are faster than HITON-MB, IAMB, and STMB. Comparing to HITON-MB, IAMB, and STMB, MCFS has competitive efficiency with synthetic data and when the sizes of the MBs of variables "VTUB" and "HR" are small, (see Tables 6 and 10), although MCFS has an additional step to find the invariant sets. When the size of the MB of C found by IAMB and STMB is much larger than that by MCFS, IAMB and STMB are much slower than MCFS, as shown in Table 17 in next Section using real-world datasets.

6.2 Results on Real-World Data

In this section, we will study the performance of MCFS with two real-world datasets. The details of these two datasets and the corresponding experimental results are reported as follows.

6.2.1 Results on the Student Dataset

The Student dataset is a real-world dataset about educational attainment of teenagers and it was provided in [24]. The original Student dataset includes records of 4,739 pupils from approximately 1,100 US high schools and 14 attributes as shown in Table 12. Following the method in [22], considering variable *distance* being the manipulated variable, the original Student dataset is split into two intervention datasets (for which the *distance* variable is intervened): one including 2,231 data instances of all pupils who live closer to a 4-year college than the median distance of 10 miles, and the other including 2,508 data instances of all pupils who live at least 10 miles from the nearest 4-year college. Then the variable *education* is selected as the target variable and we make it into a binary target, that is, whether a pupil received a Bachelor of Arts (BA) degree or not. With KNN and NB classifiers, we use MCFS and all the 16 methods listed in Table 1 to select features from the above described two intervention datasets for predicting the value of the target *education*.

Specifically, we select 2,000 data instances from the two intervention datasets to construct two training datasets (each with 2,000 training instances). The 231 instances and 508 instances remained from the two intervention datasets respectively are merged to form 739 data instances as the testing dataset. Then we use MCFS and its rivals to select features from the two training datasets. In each of the two training datasets, we train the NB, KNN, and SVM classifiers using the selected features and make predictions on the testing dataset. We repeat the experiment with each method ten times and report the average prediction accuracy, number of selected features, and running time in Tables 13, 14, and 15, respectively.

With the results in Tables 13 and 14, to compare MCFS with its rivals, we conduct t-tests at a 95 percent confidence level under the null-hypothesis, which states that whether the performance of MCFS and that of its rivals have no significant difference in prediction accuracy.

TABLE 13
Prediction Accuracy on Student Dataset ("•" Indicates that MCFS Is Statistically Better than the Compared Method)

Algorithm	NB		KNN		SVM	
	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.05$
MCFS	**0.7669 ± 0.0134**	**0.7698 ± 0.0160**	**0.7646 ± 0.0150**	**0.7671 ± 0.0187**	**0.7683 ± 0.0144**	**0.7707 ± 0.0157**
HITON-MB	0.7403 ± 0.0177•	0.7432 ± 0.0170•	0.7353 ± 0.0146•	0.7315 ± 0.0161•	0.7558 ± 0.0166•	0.7571 ± 0.0174•
∪ HITON-MB	0.7468 ± 0.0133•	0.7479 ± 0.0122•	0.7227 ± 0.0310•	0.7288 ± 0.0269•	0.7583 ± 0.0138	0.7572 ± 0.0146•
∩ HITON-MB	0.7463 ± 0.0201•	0.7422 ± 0.0216•	0.7440 ± 0.1830•	0.7423 ± 0.0213•	0.7511 ± 0.0156•	0.7531 ± 0.0166•
IAMB	0.7475 ± 0.0147•	0.7498 ± 0.0119•	0.7486 ± 0.0128•	0.7440 ± 0.0152•	0.7580 ± 0.0156	0.7587 ± 0.0151•
∪ IAMB	0.7483 ± 0.0121•	0.7482 ± 0.0120•	0.7406 ± 0.0154•	0.7369 ± 0.0174•	0.7595 ± 0.0145	0.7591 ± 0.0145•
∩ IAMB	0.7477 ± 0.0204•	0.7468 ± 0.0146•	0.7433 ± 0.0183•	0.7457 ± 0.0136•	0.7528 ± 0.0149•	0.7510 ± 0.0154•
STMB	0.7491 ± 0.0168•	0.7461 ± 0.0188•	0.7446 ± 0.0151•	0.7269 ± 0.0189•	0.7564 ± 0.0153•	0.7593 ± 0.0146•
∪ STMB	0.7430 ± 0.0126•	0.7683 ± 0.0652	0.7437 ± 0.0281•	0.7217 ± 0.0170•	0.7562 ± 0.0152•	0.7607 ± 0.0172
∩ STMB	0.7495 ± 0.0210•	0.7509 ± 0.0150•	0.7458 ± 0.0211•	0.7494 ± 0.0181•	0.7549 ± 0.0173•	0.7557 ± 0.0153•

TABLE 14
Prediction Accuracy on Student Dataset ("•" Indicates that MCFS Is Statistically Better than the Compared Method)

Algorithm	NB	KNN	SVM
MCFS	**0.7669 ± 0.0134**	**0.7646** ± 0.0150•	**0.7683 ± 0.0144**
ICP	0.7513 ± 0.0173•	0.7440 ± 0.0163•	0.7520 ± 0.0171•
mRMR	0.7535 ± 0.0153	0.6606 ± 0.0281•	0.7614 ± 0.0171
∪ mRMR	0.7444 ± 0.0154•	0.7352 ± 0.0257•	0.7604 ± 0.0158
∩ mRMR	0.7610 ± 0.0169	0.6742 ± 0.0602•	0.7586 ± 0.0198
FCBF	0.7450 ± 0.0170•	0.7346 ± 0.0368•	0.7507 ± 0.0108•
∪ FCBF	0.7556 ± 0.0159	0.7465 ± 0.0158•	0.7539 ± 0.0163•
∩ FCBF	0.7491 ± 0.0187•	0.7396 ± 0.0241•	0.7538 ± 0.0130•

In Table 13, when $\alpha = 0.01$ (i.e., the value of parameter α (significance level) is set for MCFS, IAMB, HITON-MB, and STMB), both using NB and KNN, we observe that all null-hypotheses are rejected, and thus MCFS is significantly better than all 9 rivals of MCFS on prediction accuracy. For SVM, using t-tests, except for IAMB, ∪ IAMB, and ∪ HITON-MB, we observe that MCFS is significantly better than the 6 remaining rivals. When $\alpha = 0.05$, using NB, KNN, and SVM, expect for ∪ STMB, all null-hypotheses are also rejected, then we can state that MCFS is significantly better than all rivals of MCFS (expect for ∪ STMB) on prediction accuracy. For ∪ STMB, using SVM and NB, the two null-hypotheses are accepted, then MCFS and ∪ STMB have no significant difference on prediction accuracy.

In Table 14, by conducting t-tests at a 95 percent confidence level, on prediction accuracy, we observe that using NB, MCFS is significantly better than ICP, ∪ mRMR, FCBF, and ∩ FCBF, while MCFS is not significantly better than mRMR, ∩ mRMR, and ∪ FCBF. When using KNN, MCFS is significantly better than all its rivals. When using SVM, except for mRMR, ∪ mRMR, MCFS is significantly better than the remaining rivals. Then we can conclude that no matter for setting α = 0.01 or α = 0.05 for MCFS, at most cases, MCFS is significantly better than its rivals on prediction accuracy. Moreover, from Tables 13 and 14, we can see that the feature subset selected by MCFS achieves more stable prediction accuracy than those of its rivals on NB, KNN, and SVM.

For computational efficiency, compared to IAMB, STMB, and HITON-MB, the running time of MCFS is reasonable, and MCFS is almost 70 times faster than ICP. mRMR and FCBF are the fastest algorithms. As about the correctly selected features, MCFS and its rivals are all competitive.

Over the ten runs, the features most frequently selected by MCFS include *score* and *mcollege* while ICP selects *fcollege*. As we have not the ground truth of the parents and the MB of variable *education* in this real-world dataset, we use the Max-Min Hill Climbing (MMHC) algorithm [29], a well-known algorithm for learning a Bayesian network structure from the original Student dataset. Fig. 3 gives the local Bayesian network structure around the target *education*. Using the parents and the MB of *education* in Fig. 3, over the ten runs, the average accuracies of the trained NB, KNN, and SVM classifiers are 0.7419, 0.7532, and 0.7574, respectively.

6.2.2 Gene Expression Datasets

In this section, we use three microarray gene expression datasets, Harvard, Michigan, and Stanford, which come from three laboratories studying lung cancer [3], [5]. They have been obtained from different patient samples and from different experimental environments. The three datasets were preprocessed by removing duplicated genes and genes with missing values in the datasets, resulting in three datasets each containing common 1,962 genes (features) and the following listed numbers of instances respectively [14]:

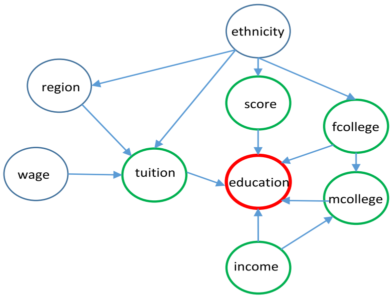

Fig. 3. A local causal structure around *education* learned from the original educational attainment dataset.

- Harvard: 156 instances, including 139 tumor and 17 normal samples.
- Stanford: 46 instances, including 41 tumor and 5 normal samples.
- Michigan: 96 instances, including 86 tumor and 10 normal samples.

Since the three datasets are class-imbalanced, we use AUC to evaluate MCFS and its rivals instead of prediction accuracy. We conduct three experiments corresponding to the three different settings of multiple datasets as shown in Table 16. In each of the three experiments, the AUC of MCFS is compared with the AUCs obtained by all the methods listed in Table 1, except for ∩ HITON-MB, ∩ IAMB, ∩ STMB, ∩ mRMR, and ∩ FCBF, as their outputs are empty.

Experiment 1. In this experiment, we have the Harvard and Stanford datasets for training while using the Michigan dataset for testing, and the results are reported in Figs. 4, 5, and 6. From these three figures (using NB, KNN, and SVM respectively), we can observe that except for ICP, the remaining 10 rivals are significantly worse than MCFS on the AUC metric. Using KNN and SVM, the values of AUC of both MCFS and ICP are up to 1 while the AUC of ∪ IAMB is only 0.5 (or 0.55) using NB and SVM (or KNN).

TABLE 15
Time and Number of Selected Features on Student Dataset

Algorithm	Time		$Features	
	α=0.01	α=0.05	α=0.01	α=0.05
MCFS	5.9±2.3	10±3.2	3.5±0.8	3.9±1.2
HITON-MB	4.89±2.4	7.6±2.4	4.8±0.6	6±0.4
∪HITON-MB			6.9±1	8.2±1.1
∩HITON-MB			2.3±0.5	3±0.8
IAMB	0.2±0	0.2±0	4.1±0.3	4.6±0.5
∪IAMB			6±0	6.2±0.4
∩IAMB			2±0.4	2.3±0.5
STMB	0.47±0.1	1.1±0.5	4.4±0.7	7±1.1
∪STMB			5.5±0.8	9.5±1.4
∩STMB			2.8±0.6	3.7±1
ICP	349±52		1.7±0.7	
mRMR	0.06±0		6±0	
∪mRMR			8.1±0.7	
∩mRMR			4.1±0.3	
FCBF	0.02±0		3.2±1	
∪FCBF			3.5±1.6	
∩FCBF			2.1±0.7	

TABLE 16
Summary of the Multiple Datasets in the Three Experiments

Experiment	Training data	Testing data
1	Harvard and Stanford	Michigan
2	Michigan and Stanford	Harvard
3	Michigan and Harvard	Stanford

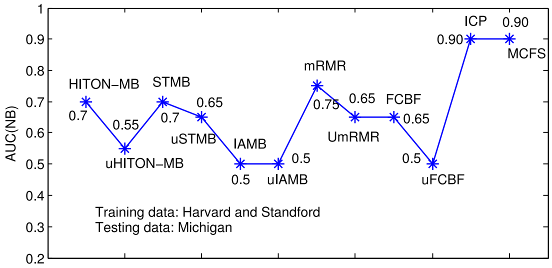

Fig. 4. AUC of NB using the features selected by MCFS and its rivals in Experiment 1.

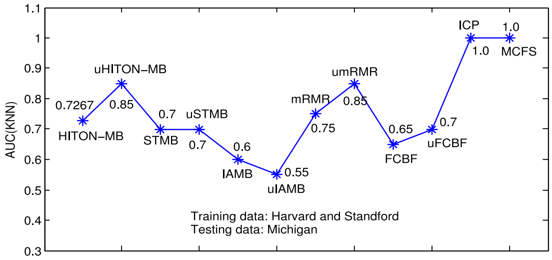

Fig. 5. AUC of KNN using the features selected by MCFS and its rivals in Experiment 1.

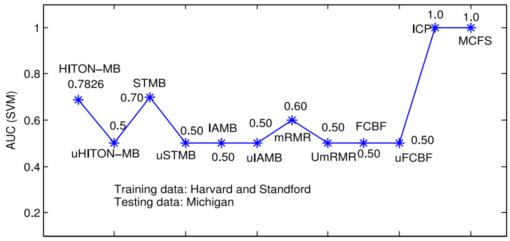

Fig. 6. AUC of SVM using the features selected by MCFS and its rivals in Experiment 1.

Experiment 2. In this experiment, the Michigan and Stanford datasets are for training while the Harvard dataset is for testing. From Figs. 7, 8, and 9, we can see that using NB, MCFS is significantly better than its 10 rivals except for mRMR. Using KNN, MCFS is significantly better than its 7 rivals, while for the AUC values of HITION-MB, IAMB, ∪ mRMR, and mRMR are close to that of MCFS, but they still achieves lower AUC than MCFS. Using SVM, except for IAMB, MCFS is significantly better than the other rivals. Moreover, MCFS and HITON-MB achieve stable AUC values, while the other rivals get fluctuating AUC values.

Experiment 3. In this experiment, we have the Michigan and Harvard datasets as the training datasets and the Stanford dataset as the testing dataset. In Figs. 10, 11, and 12, for NB and KNN, IAMB gets the worst result while for SVM, STMB is the worst. Except for STMB, ∪ STMB, FCBF, and ∪ FCBF, using NB, MCFS is the best in Fig. 10, while using KNN, except for HITION-MB and ∪ HITON-MB, MCFS is significantly better than the other rivals in Fig. 11. Using SVM, except for HITION-MB, ∪ HITON-MB, and ∪ mRMR, MCFS is significantly better than the remaining rivals. The AUC of NB with features selected by STMB, FCBF and ∪ FCBF is 1, but using KNN, the AUC of KNN with features selected by STMB, FCBF and ∪ FCBF is only up to 0.8, 0.7 and 0.7, respectively. And the similar unstable AUC values with features selected by HITION-MB and ∪ HITON-MB using NB and KNN. However, no matter for NB or KNN or SVM, the AUC when using features selected by MCFS is always 1.

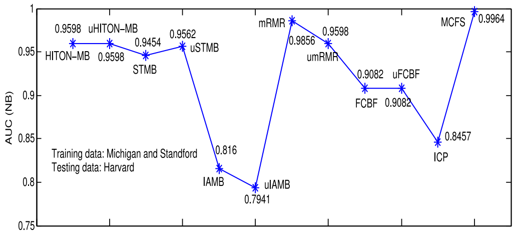

Fig. 7. AUC of NB using the features selected by MCFS and its rivals in Experiment 2.

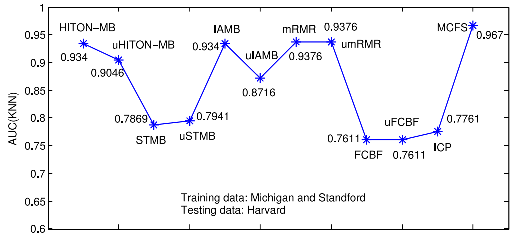

Fig. 8. AUC of KNN using the features selected by MCFS and its rivals in Experiment 2.

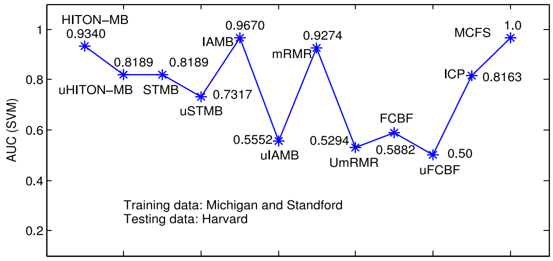

Fig. 9. AUC of SVM using the features selected by MCFS and its rivals in Experiment 2.

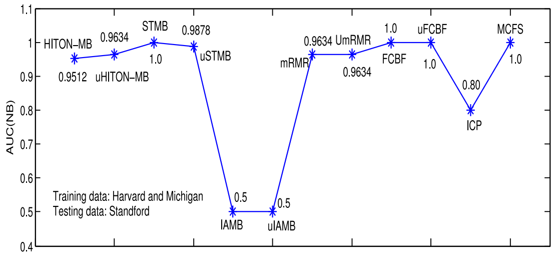

Fig. 10. AUC of NB using the features selected by MCFS and its rivals in Experiment 3.

Table 17 shows the number of selected features and running time of MCFS and its rivals. We can see that ICP selects

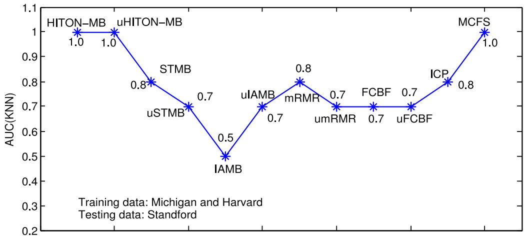

Fig. 11. AUC of KNN using the features selected by MCFS and its rivals in Experiment 3.

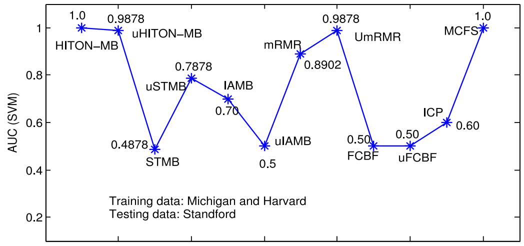

Fig. 12. AUC of SVM using the features selected by MCFS and its rivals in Experiment 3.

TABLE 17
Number of Selected Features and Running Time (E1, E2, and E3 Refer to Experiments 1, 2 and 3 Respectively)

Algorithm	#Feature			Time		
	E1	E2	E3	E1	E2	E3
MCFS	4	3	2	44	38	53
ICP	1	1	1	10	14	21
HITON-MB	5	5	6	39	35	50
UHITON-MB	10	9	9			
IAMB	92	64	114	298	193	358
UIAMB	183	125	224			
STMB	24	28	27	385	142	440
USTMB	47	55	53			
mRMR	15	15	15	7	11	12
UmRMR	20	28	24			
FCBF	20	24	21	2	2	2
UFCBF	41	47	37			

TABLE 18
Average AUC of MCFS and Its Rivals

Algorithm	NB	KNN	SVM
MCFS	**0.9655 ± 0.0567**	**0.9890 ± 0.0191**	**0.9890 ± 0.0191**
ICP	0.8486 ± 0.0501	0.8587 ± 0.1230	0.8054 ± 0.2002
HITON-MB	0.8703 ± 0.1416	0.8682 ± 0.1533	0.8741 ± 0.1642
UHITON-MB	0.8244 ± 0.2376	0.9182 ± 0.0759	0.7689 ± 0.2477
IAMB	0.6054 ± 0.1826	0.6780 ± 0.2273	0.7223 ± 0.2343
UIAMB	0.5980 ± 0.1698	0.7072 ± 0.1609	0.5184 ± 0.0319
STMB	0.8818 ± 0.1598	0.7623 ± 0.0543	0.6692 ± 0.1682
USTMB	0.8647 ± 0.1866	0.7314 ± 0.0543	0.6732 ± 0.1562
mRMR	0.8997 ± 0.1301	0.8292 ± 0.0971	0.8059 ± 0.1793
UmRMR	0.8577 ± 0.1799	0.8292 ± 0.1202	0.6724 ± 0.2735
FCBF	0.8527 ± 0.1815	0.7037 ± 0.0556	0.5294 ± 0.0509
UFCBF	0.8207 ± 0.2662	0.7204 ± 0.0353	0.5000 ± 0.0

the smallest number of features, while IAMB selects the most number of features. For computational efficiency, IAMB and STMB are the slowest since they select more features than the other algorithms, while FCBF is the fastest algorithm. Meanwhile, in Table 17, the running time and the number of selected features of MCFS also look reasonable.

Finally, we report the average results of AUC and the deviations in the three experiments in Table 18, where we can see that MCFS is significantly better than the other methods. In summary, Figs. 4, 5, 6, 7, 8, 9, 10, 11, and 12, and Table 18 show that MCFS gets significantly higher AUC and always achieves much more stable performance than its rivals.

7 Conclusion

W have analyzed causal interventions and invariance in feature selection with multiple datsets, and have proposed a new algorithm, MCFS, for causal feature selection with multiple datasets. Experiments on synthetic and real-world datasets have illustrated that if the distributions between training and testing datasets are different, MCFS is significantly better than the existing causal and non-causal feature selection algorithms.

Additionally, we empirically analyzed the bounds proposed in Theorems 6 and 7. The experiments have illustrated that with multiple intervention datasets, the set of parents of the class attribute is promising for reliable prediction while the MB of the class attribute may not be for optimal prediction. In future, on the one hand, we will explore MCFS to tackle large MBs and propose efficient methods to find invariant sets in Phase 2 of MCFS; on the other hand, our work also can be put in the context of domain adaptation, although here we focus on causal feature selection for stable predictions. In next work, we will systematically explore our work proposed in the paper for domain adaptation.

Appndix A
Mutual Information

Given two random variables X and Y, the mutual information $I(X,Y)$ and the conditional mutual information $I(X;Y|Z)$ are calculated in Eqs. (11) and (12) below [7]

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) \\ &= \Sigma_{x\in X, y\in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}. \end{aligned} \tag{11}$$

The entropy $H(X)$ and $H(X|Y)$ are defined as $H(X) = -\Sigma_{x\in X} P(x) \log P(x)$ and $H(X|Y) = -\Sigma_{y\in Y} P(y) \Sigma_{x\in X} P(x|y) \log P(x|y)$, respectively. $P(x)$ is the prior probability of value x that feature X takes, and $P(x|y)$ is the posterior probability of x given the value y that feature Y takes

$$\begin{aligned} I(X;Y|Z) &= H(X|Z) - H(X|YZ) \\ &= \Sigma_{z\in Z} P(z) \Sigma_{x\in X, y\in Y} P(x,y|z) \log \frac{P(x,y|z)}{P(x|z)P(y|z)}. \end{aligned} \tag{12}$$

Appndix B
Proofs of Theorems in Section 4

By Eqs. (11) and (12), we get Lemmas 1 and 2 as follows.

Lemma 1. *$I(F_i;F_j) \geq 0$ with equality if and only if $P(F_i,F_j) = P(F_i)P(F_j)$.*

Lemma 2. *$I(F_i;F_j|S) \geq 0$ with equality if and only if $P(F_i, F_j|S) = P(F_i|S)P(F_j|S)$.*

Proof of Theorem 2. Case 1: For $\forall S \subseteq \mathcal{F} \setminus \{C \cup MB(C)\}$, by Eq. (12), we can get the following equation:

$$I(C; S|MB(C) = E_{\{C,S,MB(C)\}}\log\frac{P(C,S|MB(C))}{P(C|MB(C))P(S|MB(C))}.$$

By Theorem 1, $P(C,S|MB(C)) = P(C|MB(C))P(S|MB(C))$ holds, and thus we get $I(C; S|MB(C)) = 0$. By the chain rule, $I((S, MB(C); C) = I(C; MB(C)) + I(C; S|MB(C)) = I(C; S) + I(C; MB(C)|S)$. Since $I(C; S|MB(C)) = 0$ holds, then $I(C; MB(C)) = I(C; S) + I(C; MB(C)|S)$ holds. By Lemmas 1 and 2, we get $I(C; MB(C)) \geq I(C; S)$ with equality if S equals to $MB(C)$.

Case 2: For $\forall S \subset MB(C)$ and $S' = MB(C) \setminus S$, by $I(C; MB(C)) - I(C; S) = I(C; S \cup S') - I(C; S) = I(C; S) + I(C; S'|S) - I(C; S) = I(C; S'|S)$, then $I(C; MB(C)) \geq I(C; S)$ holds.

Case 3: Let $S' \subset MB(C)$ and $S'' \subset F \setminus \{C \cup MB(C)\}$, and $S = \{S' \cup S''\}$, then by Theorem 8, we get Eq. (13) below. By $I(C; MB(C)) + I(C; S|MB(C)) = I(C; S) + I(C; MB(C)|S)$ and Eq. (13), in the case, $I(C; MB(C)) \geq I(C; S)$ holds

$$\begin{aligned}&\frac{P(C,S|MB(C))}{P(C|MB(C))P(S|MB(C))}\\ &= \frac{P(C,S'',MB(C))}{P(C|MB(C))P(S'',MB(C))}\\ &= \frac{P(C|S'',MB(C))P(S'',MB(C))}{P(C|MB(C))P(S'',MB(C)))} = 1.\end{aligned} \tag{13}$$

By Cases 1 to 3, $I(C; MB(C)) \geq I(C; S)$ with equality holds if S equals to $MB(C)$. □

Proof of Theorem 4. Suppose $S = \mathcal{F} \setminus \{C \cup MB(C)\}$ and $S' = \mathcal{F} \setminus MB(C)$. Let $P(sp(C)) = \prod_{k=1}^{|sp(C)|} P(F_k|Pa(F_k))$, $P(pa(C)) = \prod_{m=1}^{|pa(C)|} P(F_m|Pa(F_m))$, and $P(ch(C)) = \prod_{j=1}^{|ch(C)|} P(F_j|Pa(F_j))$, then by Eq. (1), $P(C|MB(C))$ is calculated as follows:

$$\begin{aligned}P(C|MB(C)) &= \frac{P(C,MB(C))}{P(MB(C))}\\ &= \frac{\sum_S \prod_{i=1}^{|S|} P(F_i|pa(F_i))P(C|pa(C))P(sp(C))P(ch(C))P(pc(C))}{\sum_{S'} \prod_{i=1}^{|S'|} P(F_i|pa(F_i))P(C|pa(C))P(sp(C))P(C|pa(C))P(ch(C))P(pc(C))}\\ &= \frac{P(C|pa(C))P(ch(C))\sum_S \prod_{i=1}^{|S|} P(F_i|pa(F_i))P(sp(C))P(pc(C))}{\sum_C P(C|pa(C))P(ch(C))\sum_S \prod_{i=1}^{|S|} P(F_i|pa(F_i))P(sp(C))P(pc(C))}\\ &= \frac{P(C|pa(C))\prod_{j=1}^{|ch(C)|} P(F_j|pa(F_j))}{\sum_C P(C|pa(C))\prod_{j=1}^{|ch(C)|} P(F_j|pa(F_j))}.\end{aligned} \tag{14}$$

By Eq. (2), the post-manipulation distribution of an intervention Υ_i can be factorized as

$$\begin{aligned}&P^i(\mathcal{F}|do(\Upsilon_i = \gamma_i)) = P(C|pa(C))\\ &\times \prod\nolimits_{F_j \in ch(C)} P(F_j|pa(V_j)) \times \prod\nolimits_{F_j \notin \{\Upsilon_i \cup ch(C)\}} P(F_j|pa(F_j)).\end{aligned} \tag{15}$$

By Eq. (15), since C and the variables in $ch(C)$ are not manipulated, $\forall D_i \in D$, $P^i(C|pa(C)) = P(C|pa(C))$ and $\prod_{F_j \in ch(C)} P^i(F_j|pa(V_j)) = \prod_{F_j \in ch(C)} P(F_j|pa(V_j))$ hold. Thus, by Eq. (14), the theorem is proven. □

Proof of Theorem 5. a) If $pa(C) \notin \Upsilon_i \; \forall i$, by Eq. (15), then $P^i(C|pa_i(C)) = P^j(C|pa_j(C)) \; (i \neq j)$ holds; (b) If $pa(C) \in \Upsilon_i \; \forall i$, by Properties 1 and 2, the theorem holds. □

Proof of Theorem 6. Since C is not intervened, $pa(C)$ is invariant across D. Case 1: for $\forall D_i \in D$ and $ch(C) \nsubseteq \Upsilon_i$, by Theorem 4, $MB(C)$ remains invariant across D and $pa(C) \subseteq MB(C)$. Case 2: for $\forall D_i \in D$, $\exists S \subset ch(C)$ and $S \subseteq \Upsilon_i$, for the invariant set S' across D, $pa(C) \subseteq S'$. Case 3: for $\forall D_i \in D$, if $ch(C) \subseteq \Upsilon_i$, $ch(C)$ and the corresponding $sp(C)$ are not in $MB_i(C)$, by Theorem 5, $pa(C)$ remains invariant across D. Thus, considering the three cases, $pa(C)$ is the minimally invariant set across D. □

Proof of Theorem 8. Since C is not manipulated, (1) for $\forall D_i \in D$, $pc(C)$ keeps invariant across D. Thus for $\forall MB_i(C)$, $pa(C)$ in $MB_i(C)$ holds; (2) If $\exists F_j \in ch(C)$ and $F_j \in \Upsilon$, by the conservative rule, there must exist a set Υ_m and $F_j \notin \Upsilon_m$. Then in D_m, F_j is not manipulated, and the edge between C and F_j is not deleted. Then $F_j \in MB_m(C)$. Since F_j is not manipulated in D_m, the edges between F_j and its parents (C and $C's$ spouses w.r.t F_j) are not deleted. Then the set $sp(C)$ with respect to $F_j \in ch(C)$ is in $MB_m(C)$; (3) If $\exists F_j \in ch(C)$ and $F_j \notin \Upsilon$, F_j is not manipulated. Thus, for $\forall D_i \in D$, as the same as the proof in (2), F_j and the corresponding $sp(C)$ are in $MB_i(C)$. □

Proof of Theorem 9. (1) C is not manipulated, then for $\forall MB_i(C)$, $pa(C)$ in $MB_i(C)$ holds. (2) Since Υ is not conservative, if $\exists F_j \in ch(C)$ and for $\forall \Upsilon_i \in \Upsilon$, $F_j \in \Upsilon_i$ holds, then for $\forall D_i \in D$, F_j is manipulated. Thus F_j and the corresponding $sp(C)$ are not in $MB_i(C)$. Then $\bigcup_{i=1}^{K} MB_i(C) \subset MB(C)$ holds. Otherwise, if $ch(C) \nsubseteq \Upsilon$ and $\forall F_j \in ch(C)$, for $\forall D_i \in D$, F_j is not manipulated, and $\{ch(C) \cup sp(C)\} \subset MB_i(C)$. In the case, $\bigcup_{i=1}^{K} MB_i(C) = MB(C)$ holds. □

ACKNOWLEDGMENTS

This work is partly supported by the Australian Research Council (ARC) Discovery Project (under grant DP170101306), the National Key Research and Development Program of China (under grant 2016YFB1000901), and the National Science Foundation of China (under grants 61876206 and 61673152).

REFERENCES

[1] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos, "Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation," *J. Mach. Learn. Res.*, vol. 11, no. Jan., pp. 171–234, 2010.

[2] C. F. Aliferis, I. Tsamardinos, and A. Statnikov, "HITON: A novel Markov blanket algorithm for optimal variable selection," in *Proc. AMIA Annu. Symp. Proc.*, 2003, Art. no. 21.

[3] D. G. Beer, S. L. Kardia, C.-C. Huang, T. J. Giordano, A. M. Levin, D. E. Misek, L. Lin, G. Chen, T. G. Gharib, D. G. Thomas, et al., "Gene-expression profiles predict survival of patients with lung adenocarcinoma," *Nature Med.*, vol. 8, no. 8, 2002, Art. no. 816.

[4] I. A. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper, "The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks," in *Proc. Conf. Artif. Intell. Med. Care*, 1989, pp. 247–256.

[5] A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, et al., "Classification of human lung carcinomas by MRNA expression profiling reveals distinct adenocarcinoma subclasses," *Proc. Nat. Academy Sci. United States America*, vol. 98, no. 24, pp. 13790–13795, 2001.

[6] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *J. Mach. Learn. Res.*, vol. 13, no. Jan., pp. 27–66, 2012.

[7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.

[8] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York, NY, USA: Academic, 2013.

[9] T. Gao and Q. Ji, "Efficient Markov blanket discovery and its application," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1169–1179, May 2017.

[10] M. E. Garber, O. G. Troyanskaya, K. Schluens, S. Petersen, Z. Thaesler, M. Pacyna-Gengelbach, M. Van De Rijn, G. D. Rosen, C. M. Perou, R. I. Whyte, et al., "Diversity of gene expression in adenocarcinoma of the lung," *Proc. Nat. Academy Sci. United States America*, vol. 98, no. 24, pp. 13784–13789, 2001.

[11] I. Guyon, C. Aliferis, and A. Elisseeff, "Causal feature selection," *Comput. Methods Feature Selection*, Chapman and Hall/CRC, 2007, pp. 75–97.

[12] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, no. Mar., pp. 1157–1182, 2003.

[13] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surveys*, vol. 50, no. 6, 2017, Art. no. 94.

[14] L. Liu, Y. Li, B. Liu, and J. Li, "A simple yet effective data integration approach to tree-based microarray data classification," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2010, pp. 1503–1506.

[15] S. Magliacane, T. van Ommen, T. Claassen, S. Bongers, P. Versteeg, and J. M. Mooij, "Domain Adaptation by Using Causal Inference to Predict Invariant Conditional Distributions," in *NeurIPS'18.*, 2018, pp. 10846–10856.

[16] J. M. Mooij, S. Magliacane, and T. Claassen, "Joint causal inference from multiple contexts," arXiv:1611.10351, 2016.

[17] L. Paninski, "Estimation of entropy and mutual information," *Neural Comput.*, vol. 15, no. 6, pp. 1191–1253, 2003.

[18] J. Pearl, *Causality*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[19] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Amsterdam, The Netherlands: Elsevier, 2014.

[20] J. M. Pena, R. Nilsson, J. Björkegren, and J. Tegnér, "Towards scalable and data efficient learning of Markov boundaries," *Int. J. Approximate Reasoning*, vol. 45, no. 2, pp. 211–232, 2007.

[21] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[22] J. Peters, P. Bühlmann, and N. Meinshausen, "Causal inference by using invariant prediction: Identification and confidence intervals," *J. Roy. Statistical Soc.: Series B (Statistical Methodology)*, vol. 78, no. 5, pp. 947–1012, 2016.

[23] M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters, "Invariant models for causal transfer learning," *J. Mach. Learn. Res.*, vol. 19, no. 36, pp. 1–34, 2018.

[24] C. E. Rouse, "Democratization or diversion? The effect of community colleges on educational attainment," *J. Bus. Econ. Statist.*, vol. 13, no. 2, pp. 217–224, 1995.

[25] D. Tebbe and S. Dwyer, "Uncertainty and the probability of error (Corresp.)," *IEEE Trans. Inf. Theory*, vol. 14, no. 3, pp. 516–518, May 1968.

[26] I. Tsamardinos and C. F. Aliferis, "Towards principled feature selection: Relevancy, filters and wrappers," in *Proc. 9th Int. Workshop Artif. Intell. Statist.*, 2003.

[27] I. Tsamardinos, C. F. Aliferis, and A. Statnikov, "Time and sample efficient discovery of Markov blankets and direct causal relations," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 673–678.

[28] I. Tsamardinos, C. F. Aliferis, A. R. Statnikov, and E. Statnikov, "Algorithms for large scale Markov blanket discovery," in *Proc. Int. FLAIRS Conf.*, 2003, pp. 376–380.

[29] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, "The max-min hill-climbing Bayesian network structure learning algorithm," *Mach. Learn.*, vol. 65, no. 1, pp. 31–78, 2006.

[30] S. Yaramakala and D. Margaritis, "Speculative Markov blanket discovery for optimal feature selection," in *Proc. 5th IEEE Int. Conf. Data Mining*, 2005, pp. 809–812.

[31] K. Yu, L. Liu, and J. Li, "Discovering Markov blanket from multiple interventional datasets," arXiv:1801.08295, 2018.

[32] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, no. Oct., pp. 1205–1224, 2004.

[33] Y. Zhai, Y.-S. Ong, and I. W. Tsang, "The emerging "big dimensionality"," *IEEE Comput. Intell. Mag.*, vol. 9, no. 3, pp. 14–26, Aug. 2014.

[34] K. Zhang, B. Huang, J. Zhang, B. Schölkopf, and C. Glymour, "Discovery and visualization of nonstationary causal models," arXiv:1509.08056, 2015.

Kui Yu received the PhD degree in computer science from the Hefei University of Technology, China, in 2013. He is currently a professor with the Hefei University of Technology, China. From 2015 to 2018, he was a research fellow with the University of South Australia, Australia. From 2013 to 2015, he was a postdoctoral fellow with the School of Computing Science, Simon Fraser University, Canada. His main research interests include data mining and causal discovery.

Lin Liu received the bachelor's and master's degrees in electronic engineering from Xidian University, China, in 1991 and 1994, respectively, and the PhD degree in computer systems engineering from the University of South Australia, in 2006. She is c1urrently an associated professor with the University of South Australia. Her research interests include data mining, causal inference, and bioinformatics.

Jiuyong Li received the PhD degree in computer science from Griffith University in Australia. He is currently a professor with the University of South Australia. His main research interests include data mining, privacy preservation, and bioinformatics. His research work has been supported by six Australian Research Council Discovery projects, and he has published more than 120 research papers.

Wei Ding received the PhD degree in computer science from the University of Houston, in 2008. She is an associate professor of computer science with the University of Massachusetts Boston. Her research interests include data mining, machine learning, artificial intelligence, computational semantics, and with applications to health sciences, astronomy, geosciences, and environmental sciences. She has published more than 130 refereed research papers, one book, and has three patents. She is an associate editor of the *ACM Transactions on Knowledge Discovery from Data* (TKDD) and the *Knowledge and Information Systems* (KAIS). She served as an editorial board member of the *Journal of Information System Education* (JISE), the *Journal of Big Data*, and the *Social Network Analysis and Mining Journal*. She is the recipient of the Best Paper Award at the 2011 IEEE International Conference on Tools with Artificial Intelligence (ICTAI), the Best Paper Award at the 2010 IEEE International Conference on Cognitive Informatics (ICCI), the Best Poster Presentation award at the 2008 ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPAITAL GIS). She received the 2019 WISAY Distinguished Woman in Science Award from Yale University, 2018 Outstanding Alumni Award and the Best PhD Work Award between 2007 and 2010 from the University of Houston. Her research projects have been sponsored by NSF, NIH, NASA, and DOE. She is a senior member of the IEEE and ACM.

Thuc Duy Le is a senior lecturer with the School of Information Technology and Mathematical Sciences, University of South Australia. His research interests include causal discovery methods and their applications in bioinformatics. He is also a National Health & Medical Research Council (NHMRC) Early Career Research Fellow (2017-2020). He has served as an academic editor of the *Plos One* and as a reviewer for many top conferences in data mining and journals in bioinformatics.