Causal Feature Selection in the Presence of Sample Selection Bias

SHUAI YANG, Anhui Agricultural University, China
XIANJIE GUO, KUI YU, and XIAOLING HUANG, Hefei University of Technology, China
TINGTING JIANG, JIN HE, and LICHUAN GU, Anhui Agricultural University, China

Almost all existing causal feature selection methods are proposed without considering the problem of sample selection bias. However, in practice, as data-gathering process cannot be fully controlled, sample selection bias often occurs, leading to spurious correlations between features and the class variable, which seriously deteriorates the performance of those existing methods. In this article, we study the problem of causal feature selection under sample selection bias and propose a novel Progressive Causal Feature Selection (PCFS) algorithm which has three phases. First, PCFS learns the sample weights to balance the treated group and control group distributions corresponding to each feature for removing spurious correlations. Second, based on the sample weights, PCFS uses a weighted cross-entropy model to estimate the causal effect of each feature and removes some irrelevant features from the confounder set. Third, PCFS progressively repeats the first two phases to remove more irrelevant features and finally obtains a causal feature set. Using synthetic and real-world datasets, the experiments have validated the effectiveness of PCFS, in comparison with several state-of-the-art classical and causal feature selection methods.

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

Additional Key Words and Phrases: Causal feature selection, sample selection bias, causal effect

ACM Reference format:
Shuai Yang, Xianjie Guo, Kui Yu, Xiaoling Huang, Tingting Jiang, Jin He, and Lichuan Gu. 2023. Causal Feature Selection in the Presence of Sample Selection Bias. *ACM Trans. Intell. Syst. Technol.* 14, 5, Article 78 (August 2023), 18 pages.
https://doi.org/10.1145/3604809

1 INTRODUCTION

In the big data era, high-dimensional datasets are ubiquitous in various real-world applications, such as text mining and image classification. Feature selection, which is to identify a subset

This work was supported by the National Key Research and Development Program of China (under grant 2021ZD0111801), the National Natural Science Foundation of China (under grant 62176082), and the University Synergy Innovation Program of Anhui Province (under grant GXXT-2022-046).
Authors' addresses: S. Yang, T. Jiang, J. He, and L. Gu, Anhui Agricultural University, School of Information and Computer, Hefei 230036, China; emails: {yangs, jiangtt, hejin, glc}@ahau.edu.cn; X. Guo, K. Yu, and X. Huang, Hefei University of Technology, Key Laboratory of Knowledge Engineering with Big Data of Ministry of Education (Hefei University of Technology), and School of Computer Science and Information Engineering, Hefei 230601, China; emails: xianjieguo@mail.hfut.edu.cn, yukui@hfut.edu.cn, hxl@chzu.edu.cn.

2157-6904/2023/08-ART78 $15.00
https://doi.org/10.1145/3604809

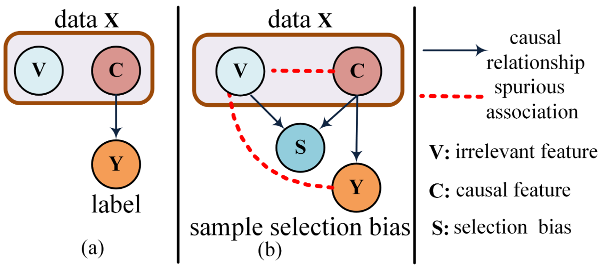


Fig. 1. (a) The dependencies among **C**, **V**, and **Y** in an unbiased dataset. (b) Selection bias (**S**) leads to spurious associations between **V** and (**C**, **Y**) in a biased dataset.

of relevant features from the origin features for building robust prediction models, is more urgent than ever, as it can well reduce the complexity of the high-dimensional learning tasks while improving the interpretability and robustness of the learning algorithm [33]. Due to its importance, feature selection has been well studied, and many methods for feature selection have been developed, which fall into two main categories, classical feature selection [8, 14] and causal feature selection [6, 7, 29, 34]. For instance, BASSUM [8] leverages knowledge from both labeled and unlabeled samples for feature selection. SELF [6] utilizes a structural equational likelihood framework and a hill climbing-based causal structure discovery algorithm for identifying causes and effects. THP [7] recovers causal structure on event sequences following different but related distributions and learns causal features of a given interesting variable using the learned causal structure. In contrast to classical feature selection, a prediction model built with causal features is more robust, as causal features imply the causal mechanism around the class variable [32].

Recently, many causal feature selection methods have been presented, with the aim at identifying the **Markov blanket** (**MB**) of a class variable, which contains its PC (parents and children) and spouses (other parents of the children of the class variable) [16]. In theory, the MB of the class variable is the optimal feature subset for feature selection, as the class variable is independent of the remaining features given the MB of the class variable. Existing methods can be mainly categorized into simultaneous MB learning and divide-and-conquer MB learning [32]. The former learns PC and spouses simultaneously and does not distinguish PC from spouses, such as GSMB [16], IAMB [24], and EAMB [10]. However, this type of method requires the number of samples to be exponential to the size of the MB set, and thus they are not data-efficient. To alleviate this problem, divide-and-conquer MB learning methods have been proposed, which learn PC and spouses separately, such as HITON-MB [1], EEMB [26], and CCMB [28].

While emerging successes have been made, existing causal feature selection algorithms rarely consider the problem of sample selection bias [4, 5] that is ubiquitous in practical applications, leading to limited performance. This can be explained from the perspective of causality. Let $\mathbf{Y}$ be the class label, $\mathbf{X} = \mathbf{C} \cup \mathbf{V}$ be the set of features in training data, where $\mathbf{C}$ is a set of causal features, $\mathbf{V}$ is a set of irrelevant features, and $\mathbf{C} \cap \mathbf{V} = \emptyset$. The dependencies among $\mathbf{C}$, $\mathbf{V}$, and $\mathbf{Y}$ are depicted in Figure 1(a), where the directed edges represent the causality, e.g., $\mathbf{C}$ is a direct cause of $\mathbf{Y}$. In practical applications, it is often hard to control the data-gathering process, leading to the collected data often suffering from the sample selection bias problem, as shown in Figure 1(b), where $\mathbf{S}$ denotes a binary indicator of a sample ($\mathbf{S} = 0$ represents that a sample is not selected into a training dataset and $\mathbf{S} = 1$, otherwise).

Taking image classification as an example, when building a camel image classification model, we may collect a training dataset where most of camel pictures were taken in the desert ($\mathbf{S} = 1$), excluding the camel pictures taken in other backgrounds ($\mathbf{S} = 0$). If sample selection bias occurs, the path **V**→ **S** ← **C**→ **Y** will be activated, as nodes **C**, **S**, and **V** form a V-structure [17]. That is, **V** is conditionally dependent on **C** given the selection bias **S**, and hence **V** is dependent on **Y** in this case. If using the biased dataset to perform causal feature selection, the background feature *desert* will be selected as a causal feature. However, in fact, the feature *desert* is not a causal feature of camel images. That is, sample selection bias will severely deteriorate the performance of existing causal feature selection methods. In addition, in practice, true causality can only be determined by using controlled experimentation, and thus the causal features learned by existing causal feature selection methods are only potentially causal features, not true causal features.

Then a question naturally arises: how to select true causal features under sample selection bias? To tackle this problem, in this article, we propose a novel **progressive causal feature selection** (**PCFS**) algorithm for selecting causal features. Our main contributions are summarized as follows:

— We investigate the important problem of causal feature selection under sample selection bias and propose an algorithm called PCFS, which progressively removes irrelevant features for identifying causal features by estimating the causal effect between features and the class variable.
— We have conducted extensive experiments using both synthetic and real-world datasets, and have compared PCFS with several state-of-the-art algorithms, to demonstrate the effectiveness of PCFS.

2 RELATED WORK

A variety of methods has been designed for feature selection recently, and they fall into two main categories, classical and causal feature selection methods. For the details on the former, we refer the readers to related literature, e.g., [14]. In this article, we mainly focus on the latter.

Existing causal feature selection methods can be broadly categorized into two different types. One is the simultaneous MB learning methods which learn PC and spouses of the class variable simultaneously without distinguishing spouses from its PC. To name a few, given the entire candidate MB currently selected, GSMB [16] employs a forward-backward strategy to greedily identify an MB of the class variable at each iteration. As a variant of GSMB, at each iteration, IAMB [24] selects the feature that has the highest association with the class variable into the candidate MB set and thus obtains better performance than GSMB. To further improve the efficiency of IAMB, two variants, Fast-IAMB [31] and FBED^K [3], have been proposed. Different from IAMB, in the forward phase, an aggressively greedy strategy is applied in Fast-IAMB and an early dropping strategy is adopted in FBED^K to accelerate MB learning. To tackle the problem of unreliable **conditional independence** (**CI**) tests, EAMB [10] first learns an MB subset and then recovers true positive MB features from discarded features. However, these methods encounter the data-inefficient problem, as the data samples required by those algorithms are exponential to the size of the MB set. When the data samples are insufficient, low-quality MBs would be learned.

To reduce the data requirements, divide-and-conquer MB learning methods have been proposed, which first learn PC, and then identify spouses of the class variable. One typical method is MMMB [25], which first discovers PC of the class variable using the **max-min parents and children** (**MMPC**) algorithm, and then learns a superset of MB by identifying the PC set of each feature within the PC set of the class variable, and finally removes false positives to discover spouses. Different from MMMB, HITON-MB [1] employs the HITON-PC algorithm that interleaves the forward phase and the backward phase to remove false positives from the PC set as early as possible for accurate PC learning. However, HINTON-MB may discard some true positive MB features due to unreliable CI tests. To alleviate this problem, Wu et al. present a concept of PCMasking to describe a type of incorrect CI tests during MB learning and propose the CCMB algorithm to

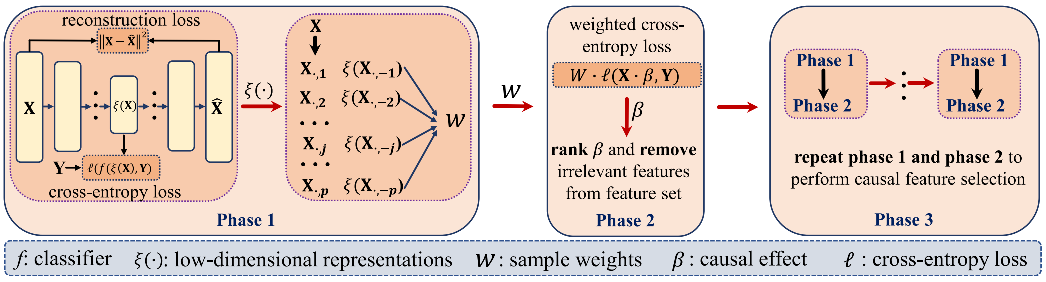

reconstruction loss
$\|X-\hat{X}\|^2$
X
$\xi(X)$
$\hat{X}$
Y
$\ell(f(\xi(X),Y)$
cross-entropy loss
$\xi(\cdot)$
X
$X_{\cdot,1}$ $\xi(X_{\cdot,-1})$
$X_{\cdot,2}$ $\xi(X_{\cdot,-2})$
$X_{\cdot,j}$ $\xi(X_{\cdot,-j})$
$X_{\cdot,p}$ $\xi(X_{\cdot,-p})$
W
Phase 1
W
weighted cross-entropy loss
$W\cdot\ell(X\cdot\beta,Y)$
β
rank β and remove irrelevant features from feature set
Phase 2
Phase 1
Phase 2
Phase 1
Phase 2
repeat phase 1 and phase 2 to perform causal feature selection
Phase 3
f: classifier
$\xi(\cdot)$: low-dimensional representations
W: sample weights
β: causal effect
ℓ: cross-entropy loss

Fig. 2. The framework of PCFS.

tackle the PCMasking phenomenon for accurate MB learning [28]. To achieve the tradeoff between time efficiency and data efficiency, BAMB [15] and EEMB [26] alternatively learn PC and spouses. Instead of learning causal features from a single dataset, MCFS [34] obtains causal features from multiple intervention datasets by using the concept of causal invariance.

However, just as we discussed in Section 1, they do not consider the sample selection bias problem and only learn potential causal features, resulting in limited performance.

3 PRELIMINARY WORK

Supervised AutoEncoder: An unsupervised autoencoder is a feed forward neural network with an input layer, one or more hidden layers, and an output layer. The autoencoder framework consists of an encoding phase and a decoding phase. To be specific, given input data $\mathbf{X} \in \mathbb{R}^{n\times d}$, where n and d are the number of training examples and features, respectively, the autoencoder first uses multiple nonlinear encoding processes to encoder it to learn low-dimensional representations $\xi(\mathbf{X})$ of $\mathbf{X}$, and then decodes $\xi(\mathbf{X})$ to obtain the output data $\hat{\mathbf{X}}$. The encoding and decoding processes can be formalized as follows:

$$\begin{aligned}\text{Encode}: \boldsymbol{\xi}^{(j)} &= \sigma(\boldsymbol{\xi}^{(j-1)}\mathbf{U}_1^{(j)} + \mathbf{b}_1^{(j)}), j = 1, 2, \ldots, l,\\ \text{Decode}: \boldsymbol{\psi}^{(j)} &= \sigma(\boldsymbol{\psi}^{(j-1)}\mathbf{U}_2^{(j)} + \mathbf{b}_2^{(j)}), j = 1, 2, \ldots, l,\end{aligned} \tag{1}$$

where σ is a nonlinear activation function (e.g., sigmoid function), and l is the number of hidden layers. Here, $\boldsymbol{\xi}^{(0)} = \mathbf{X}$. $\boldsymbol{\xi}^{(l)}$, denoted by $\xi(\cdot)$, is the low-dimensional representations of $\mathbf{X}$, and $\boldsymbol{\psi}^{(0)} = \boldsymbol{\xi}^{(l)}$. $\mathbf{U}_1^{(j)}$ and $\mathbf{U}_2^{(j)}$ are the weight matrixes. $\mathbf{b}_1^{(j)}$ and $\mathbf{b}_2^{(j)}$ are the bias vectors. The autoencoder optimizes $\xi(\mathbf{X})$ by minimizing reconstruction error between $\mathbf{X}$ and $\hat{\mathbf{X}}$. The objective function of an unsupervised autoencoder is formalized as follows:

$$\mathcal{L}_{AE} = \frac{1}{n}\|\mathbf{X} - \hat{\mathbf{X}}\|^2 + \lambda_1 \sum_{j=1}^{l}\sum_{i=1}^{2}\left(\left\|\mathbf{U}_i^{(j)}\right\|^2 + \left\|\mathbf{b}_i^{(j)}\right\|^2\right), \tag{2}$$

where λ_1 is the balancing parameter.

The supervised autoencoder further improves the quality of low-dimensional representations using the label information and incorporates a cross-entropy loss $\ell(\cdot)$ to the objective function as follows:

$$\mathcal{L}_{SAE} = \frac{1}{n}\|\mathbf{X} - \hat{\mathbf{X}}\|^2 + \lambda_1 \sum_{j=1}^{l}\sum_{i=1}^{2}\left(\left\|\mathbf{U}_i^{(j)}\right\|^2 + \left\|\mathbf{b}_i^{(j)}\right\|^2\right) + \lambda_2\ell(f(\xi(\mathbf{X})), \mathbf{Y}), \tag{3}$$

where f is a classifier and λ_2 is the balancing parameter.

4 PROPOSED PROGRESSIVE CAUSAL FEATURE SELECTION

Overview of PCFS. We propose the PCFS algorithm to learn causal features for building a robust prediction model. The framework of PCFS is shown in Figure 2. PCFS consists of three phases. Phase 1 learns weights for training samples to balance the distribution of the treated and control groups of each feature. Phase 2 first assigns sample with the weights learned in Phase 1, and then uses the weighted cross-entropy algorithm to compute the causal effect of each feature and removes some irrelevant features. Phase 3 repeats Phases 1 and 2 to perform causal feature selection by progressively removing irrelevant features.

Phase 1: Learning sample weights. The presence of sample selection bias often leads to some irrelevant features being selected by existing causal feature selection algorithms. In practice, given a single feature, if we know the causal effect between the feature and the class label, we can determine whether the feature is a causal feature or an irrelevant feature. Therefore, it is necessary to estimate the causal effect between features and the class variable when performing causal feature selection. In reality, the true causality can be identified by performing randomized experiments. However, in many practical applications, it is often hard to carry out randomized experiments. Instead, we estimate the causal effect using observational data.

The key challenge of estimating causal effect using observational data is to remove the confounding bias [19] induced by the confounders that affect both the treatment T and the class variable. To this end, a confounder balancing technique is adopted. Specifically, given a treatment feature T, when estimating the causal effect between T and the class variable, we first need to identify confounders. However, in observational studies, the prior knowledge of the causal structure is unknown, that is, we do not know which features are confounders, and thus the remaining features as regarded as confounders, as shown in Figure 3(a), the confounder set of T is $\mathbf{X} \setminus \{T\}$. Second, a new dataset is constructed by using these confounders as features. Third, according to the value of the samples in the feature T, the samples in the new dataset are divided into two groups, i.e., a treated group (T=1) and a control group (T=0). If the value of the sample in the feature is 1, the sample is divided into the treated group. Otherwise, the sample is assigned to the control group. We can estimate the causal effect of T on the class variable by comparing the average difference between treated and control groups.

However, due to sample selection bias, the distribution of the treated group is often different from that of the control group. To estimate the causal effect of a treatment feature T, we can learn a set of samples weight W to balance the data distribution between the treated group and the control group as follows:

$$\begin{aligned}\mathcal{L} = &\left\| \sum_{i=1}^{n} W_i \cdot x_i \cdot T_i - \sum_{i=1}^{n} W_i \cdot x_i \cdot (1 - T_i) \right\|_2^2 + \lambda_3 \left(\sum_{i=1}^{n} W_i - n \right)^2 \\ &+ \lambda_4 \sum_{i=1}^{n} (W_i - 1)^2,\end{aligned} \tag{4}$$

where x_i is the i^{th} sample of the data $\mathbf{X}$. W_i is the weight of x_i. $\sum_{i=1}^{n} W_i \cdot x_i \cdot T_i$ and $\sum_{i=1}^{n} W_i \cdot x_i \cdot (1-T_i)$ are the first-order moments of T on treated and control groups, respectively. λ_3 and λ_4 are the balancing parameters. The term $(\sum_{i=1}^{n} W_i - n)^2$ is used to guarantee that the sum of all sample weights is n. The term $\sum_{i=1}^{n} (W_i - 1)^2$ helps to reduce the variance of the sample weights. *After removing the confounding bias, the correlation between T and the class variable is the causal effect.*

To select causal features, we need to estimate the causal effects of all features. However, in practice, for each feature, learning a set of sample weights is infeasible, especially on high-dimensional

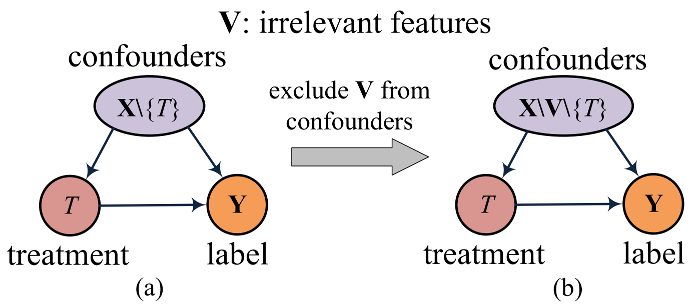


Fig. 3. The dependencies among confounders, treatment T, and the class variable $\mathbf{Y}$. (a) The identified confounder set contains irrelevant features. (b) The identified confounder set does not contain irrelevant features.

learning tasks. To this end, motivated by [11], the proposed algorithm learns a set of weights from a global perspective to align the distribution of the treated group and the control group corresponding to each feature as follows:

$$\begin{aligned}\mathcal{L} = \sum_{j=1}^{p} \left\| \sum{}_{i=1}^{n} W_i \cdot x_i \cdot T_i^j - \sum_{i=1}^{n} W_i \cdot x_i \cdot (1 - T_i^j) \right\|_2^2 + \lambda_3 \left(\sum_{i=1}^{n} W_i - n \right)^2 \\ + \lambda_4 \sum_{i=1}^{n} (W_i - 1)^2,\end{aligned} \tag{5}$$

where T^j is the j^{th} feature in $\mathbf{X}$.

However, in reality, there are often nonlinear relationships between features and there is often noise in the data, which are easy to disturb the balance of data distribution between the treated and control groups, resulting in poor quality of W. Since the supervised autoencoder has advantages in learning nonlinear relationships between features and compressing noise, the proposed algorithm uses a supervised autoencoder to map the treated and control group data into a low-dimensional nonlinear space and then balances the data distribution between them. Specifically, given the input data $\mathbf{X}$ and the class label $\mathbf{Y}$, we use Equation (3) to learn a supervised autoencoder model. Once $\mathbf{U}_i^{(j)}, \mathbf{b}_i^{(j)}$ $(i = 1, 2; j = 1, 2, \ldots, l)$ are learned, we can obtain the low-dimensional representations of the treated and control groups. And thus, the Equation (5) can be rewritten as follows:

$$\begin{aligned}\mathcal{L} = \sum_{j=1}^{p} \left\| \frac{\xi(\mathbf{X}_{\cdot,-j})^T \cdot (W \odot \mathbf{X}_{\cdot,j})}{W^T \odot \mathbf{X}_{\cdot,j}} - \frac{\xi(\mathbf{X}_{\cdot,-j})^T \cdot (W \odot (1 - \mathbf{X}_{\cdot,j}))}{W^T \odot (1 - \mathbf{X}_{\cdot,j})} \right\|_2^2 \\ + \lambda_3 \left(\sum_{i=1}^{n} W_i - n \right)^2 + \lambda_4 \sum_{i=1}^{n} (W_i - 1)^2,\end{aligned} \tag{6}$$

where $\mathbf{X}_{\cdot,j}$ is the j^{th} feature in $\mathbf{X}$, and $\mathbf{X}_{\cdot,-j} = \mathbf{X} \backslash \mathbf{X}_{\cdot,j}$ represents all the other features by removing the j^{th} feature in $\mathbf{X}$. $\odot$ is the Hadamard product.

Phase 2: Estimating the causal effect of each feature. Just as we discussed in Phase 1, the sample weights W learned in Phase 1 can be used to remove the confounding bias. Studies [2, 12] show that the correlation between a given feature T and the class variable is the causal effect if the confounding bias is removed. Motivated by this, we minimize a weighted cross-entropy loss to estimate the causal effect of each feature as follows:

$$-\sum_{i=1}^{n} W_i \cdot \left(Y_i \cdot \log \frac{1}{1+exp(-x_i \cdot \beta)} + (1-Y_i) \cdot \log\left(1 - \frac{1}{1+exp(-x_i \cdot \beta)}\right)\right) + \lambda_5 \|\beta\|_1, \quad (7)$$

where Y_i is the label of the i^{th} sample x_i.

The value of β_i is the causal effect between the i^{th} feature and the class variable. Given a fixed threshold $\delta > 0$, if $|\beta_i| \geq \delta$, the i^{th} feature is a causal feature. Otherwise, the i^{th} feature is an irrelevant feature. According to the value of β, we can obtain causal features.

Phase 3: Progressively removing irrelevant features. In Equation (6), when evaluating the causal effect of a treatment feature, all the other features are regarded as confounders, including irrelevant features. In reality, given a treatment feature, not all the other features are confounders. If some irrelevant features are viewed as confounders, the treated and control groups' data would contain irrelevant features and the balance of true positive confounders would be disturbed, resulting in that the learned W may be inaccurate. Studies [21, 30] have shown that failure to adjust for confounders can result in incorrect conclusions. That is, if confounders cannot be well balanced, causal effect estimation would be undesirable, and low-quality causal feature sets would be obtained. Therefore, it is necessary to remove irrelevant features from the confounder set for more reasonable causal effect estimation.

To this end, we progressively remove irrelevant features from the confounder set by iteratively carrying out Phases 1 and 2. To be specific, (1) given a treatment feature, we construct the confounder set (initially, all remaining features are regarded as confounders), and then we learn the sample weights W using Equation (6). (2) Based on the learned W, we use the weighted logistic regression algorithm (see Equation (7)) to identify some irrelevant features $\mathbf{V}$ and remove them from the confounder set, and obtain the confounder set $\mathbf{X}\setminus\mathbf{V}\setminus\{T\}$, as shown in Figure 3(b). (3) We repeat steps (1) and (2) for removing all irrelevant features and obtain causal features. In practice, it is difficult to determine the threshold δ for identifying irrelevant features. If the value of δ is set too large, some causal features would be discarded due to unreliable causal effect estimation. Instead, at each iteration, the top 10% of features with the smallest causal effect values are regarded as irrelevant features. That is, in previous iterations, we remove 10% of the features each time. In the last iteration, we use the threshold δ to determine causal features, as the causal effect estimation achieves optimal after several iterations.

5 EXPERIMENTS

In this section, we evaluate the performance of the proposed PCFS algorithm by comparing it with several state-of-the-art classical and causal feature selection methods.

5.1 Experimental Settings

5.1.1 Datasets. To validate the effectiveness of PCFS, experiments on synthetic and real-world datasets are performed.

Synthetic dataset. First, we generate observed features $\mathbf{X} = \{\mathbf{C},\mathbf{V}\} = \{C_1,\ldots,C_{p_c},V_1,\ldots,V_{p_v}\} \backsim \mathcal{N}(0,1)$ with independent Gaussian distribution, where $p_c + p_v = p$. To make $X_i \in \mathbf{X}$ binary, we set $X_i = 1$ when $X_i>0$, otherwise $X_i = 0$, where X_i represents the i^{th} variable in $\mathbf{X}$. To simulate complicated causal relationships, we separate the causal features into a linear part $\mathbf{C}_l$ and a non-linear part $\mathbf{C}_n$. Then, we generate the class variable Y using the following generation function [11].

$$Y = 1/(1 + exp\left(-\sum\nolimits_{X_i \in \mathbf{C}_l} \alpha_i \cdot X_i - \sum\nolimits_{X_j \in \mathbf{C}_n} \beta_j \cdot X_j \cdot X_{j+1}\right) + \mathcal{N}(0, 0.2),$$

where $\alpha_i = (-1)^i \cdot (i\%3 + 1) \cdot p/3$ and $\beta_j = p/2$. To make Y binary, we set $Y = 1$ when $Y \geq 0.5$, otherwise $Y = 0$.

To verify the effectiveness of our method on data with sample selection bias, we generate a set of environments e by varying $P(Y|\mathbf{V})$ with a bias rate $r \in (0,1)$. Specifically, for each sample, it is selected with probability r if $V_i = Y$, otherwise, it is selected with probability 1-r.

Amazon Review is a cross-domain sentiment classification dataset of product reviews collecting from four types of products: *Books* (B), *DVDS* (D), *Electronics* (E), and *Kitchen appliances* (K), each of which has about 1,000 positive and 1,000 negative reviews. In our experiments, we use the preprocessed version of Amazon Review reported in [27], and construct twelve tasks: B→D, B→E, . . ., K→E, where B→D represents that B and D are used as the training data and testing data, respectively.

Office-Caltech10 *with SURF features* consists of 2,533 images collected from four real-world domains: *Caltech-256* (C), *Amazon* (A), *Webcam* (W), and *DSLR* (D). In the experiments, we generate twelve tasks: C→A, C→W, . . ., D→W. For this dataset, the preprocessed version reported in [22] is adopted.

Note that for Amazon Review and Office-Caltech10 datasets, we convert features to binary ones using one-hot encoding.

5.1.2 Baselines. We compare PCFS with two classical feature selection methods, FCBF [35] and LUFS [23], and four causal feature selection methods, BAMB [15], EEMB [26], EAMB [10], and CVS [13]. CVS first selects a causal feature using prior knowledge and then performs CI tests to learn the dependencies between this causal feature and other features for selecting other causal features. Note that PCFS only uses the learned causal features, i.e., a subset of the original features, for building the classifier.

5.1.3 Implementation Details. On the synthetic datasets, the values of λ_1, λ_2, λ_3, λ_4, and λ_5, δ are set to 0.0001, 10, 0.0001, 0.001, 0.01, and 0.1, respectively. On the Amazon Review and Office-Caltech10 dataset, the values of λ_1, λ_2, λ_3, λ_4, and λ_5, δ are set to 0.0001, 1, 0.0001, 0.001, 1, and 0.001, respectively. On all datasets, the number of stacked layers is set to 2. All experimental results are conducted on Windows 10 with Intel(R) i7-10700, 2.90 GHz CPU, and 64GB memory.

5.1.4 Evaluation Metrics. For synthetic datasets, the **Root Mean Square Error (RMSE)**, Average_Error, and Stability_Error metrics [11] are used to evaluate the performance.

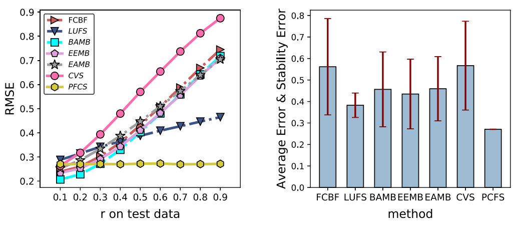
$$\mathrm{RMSE}(\mathbf{D}^e) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

where $\mathbf{D}^e$ is the dataset from environment $e \in \varepsilon$, ε is a set of all environments. y_i and $\hat{y}_i$ are the true and predicted label of the i^{th} sample, respectively. Average_Error and Stability_Error are defined as follows:

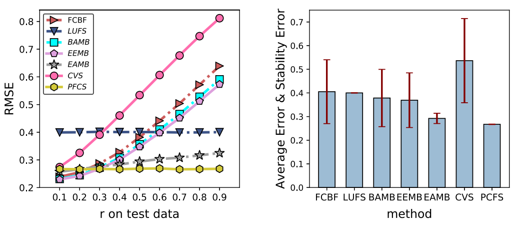
$$\mathrm{Average\_Error} = \frac{1}{|\varepsilon|}\sum_{e\in\varepsilon}\mathrm{RMSE}(\mathbf{D}^e),$$

$$\mathrm{Stability\_Error} = \sqrt{\frac{1}{|\varepsilon|-1}\sum_{e\in\varepsilon}(\mathrm{RMSE}(\mathbf{D}^e) - \mathrm{Average\_Error})^2}.$$

For real-world datasets, the classification accuracy of testing data is used as the quality metric. In the following experiments, the logistic regression classifier is used.

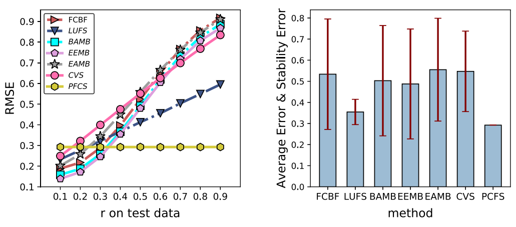


(a) Trained on $n = 2000,\ p = 20,\ r = 0.15$

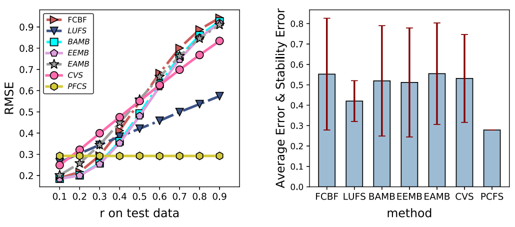


(b) Trained on $n = 2000,\ p = 20,\ r = 0.25$

Fig. 4. RMSE, Average_Error, and Stability_Error of PCFS and its rivals on various test datasets by varying bias rate r ($p = 20$).



(a) Trained on $n = 2000,\ p = 40,\ r = 0.15$



(b) Trained on $n = 2000,\ p = 40,\ r = 0.25$

Fig. 5. RMSE, Average_Error, and Stability_Error of PCFS and its rivals on various test datasets by varying bias rate r ($p = 40$).

5.2 Experiment Results on Synthetic Data

We generated data with 2,000 samples by varying selection bias rate $r = \{0.15, 0.25\}$ and the dimension of variables $p = \{20, 40, 60\}$. The results of PCFS and its rivals are shown in Figures 4–6, from which we have the following observations.

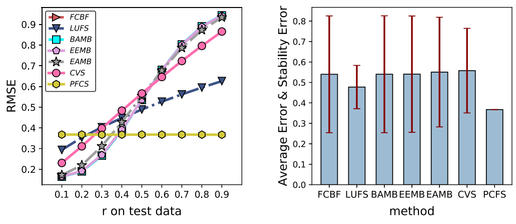
— Regardless of the number of variables p and selection bias rate r, PCFS is superior to its rivals. Specifically, PCFS achieves lower RMSE than the baseline methods. Furthermore, PCFS achieves lowest Average_Error and Stability_Error values, and this confirms the robustness of our method. The reason is that the baselines utilize co-occurrence relationships between the class variable and features or use CI tests to select features, which are easy to select irrelevant features under sample selection bias. In contrast, PCFS selects causal features by estimating the causal effect between the class variable and features, which can well tackle the sample selection bias problem.

— LUFS performs worse than PCFS but better than the other methods. This is because LUFS utilizes $l_{2,1}$-norm to regularize the feature self-representation term and uses l_1-norm to regularize the graph Laplacian term, making the model more robust to irrelevant (noisy) features. PCFS can well identify irrelevant features by estimating the causal effect, and thus achieves lower Average_Error and Stability_Error values than LUFS.

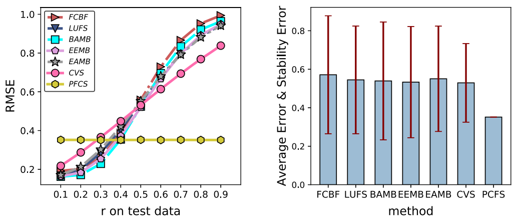
— The performance of all algorithms decreases with the increase of the number of features, which indicates that the increase of feature dimension increases the difficulty of learning. However, PCFS achieves the best performance in different feature dimensions, which shows the effectiveness of our method.

5.3 Experiments Results on Real-world Data

The experimental results of PCFS and its rivals on Amazon Review and Office-Caltech10 datasets are presented in Tables 1 and 2, respectively. We observe that PCFS outperforms its rivals on



(a) Trained on $n = 2000$, $p = 60$, $r = 0.15$



(b) Trained on $n = 2000$, $p = 60$, $r = 0.25$

Fig. 6. RMSE, Average_Error, and Stability_Error of PCFS and its rivals on various test datasets by varying bias rate r ($p = 60$).

Table 1. Accuracy (%) of the 12 Cross-domain Tasks on Amazon Review based on the Logistic Regression Classifier

Methods	FCBF	LUFS	BAMB	EEMB	EAMB	CVS	PCFS
B→D	73.84	73.74	73.24	74.24	74.34	76.94	**77.94**
B→E	67.02	68.57	72.32	73.32	73.27	71.32	**76.73**
B→K	69.08	72.59	73.14	75.34	75.29	73.24	**77.44**
D→B	70.25	70.60	71.60	71.30	70.30	70.35	**72.75**
D→E	72.77	72.77	72.17	72.27	72.57	**74.17**	73.97
D→K	76.59	73.14	75.34	74.14	76.39	75.09	**77.24**
E→B	66.60	72.00	66.10	66.65	68.05	69.35	**72.95**
E→D	69.33	72.89	68.93	69.28	70.04	72.59	**73.79**
E→K	78.69	81.64	78.49	79.24	79.99	81.89	**82.79**
K→B	69.20	68.35	68.50	69.70	**69.75**	68.35	**69.75**
K→D	70.34	73.99	71.84	72.24	70.99	73.99	**74.59**
K→E	78.53	81.33	78.63	80.23	78.73	81.33	**81.93**
Avg	71.85	73.47	72.53	73.16	73.31	74.05	**75.99**
Avg rank	5.54	4.25	5.50	4.17	3.96	3.46	1.13

The best result is highlighted in bold.

most tasks, especially on the tasks of B→E, B→K, C→D, and W→C. We also see that the average classification accuracies of PCFS on the Amazon Review and Office-Caltech10 datasets are 75.99% and 44.35%, respectively. Compared to the best baseline CVS, the average performance of PCFS gains 1.94%, 1.24% of improvement, respectively, which demonstrates the effectiveness of PCFS.

To further demonstrate the robustness of the causal features selected by our method, we also use two classifiers, i.e., SVM and Naive Bayes, to compute the classification accuracies achieved by using the selected feature subsets. The experimental results are reported in Tables 3–6.

We observe that regardless of the classifier used, PCFS achieves better average accuracies than its rivals and achieves the lowest average rank value, which indicates the effectiveness of the proposed method. We also see that CVS performs better than BAMB, EEMB, and EAMB. The reason for this is that BAMB, EEMB, and EAMB cannot well deal with high-dimensional complex data, when the dimension increases, especially when there are hundreds of features, causal feature learning becomes unreliable given the limited number of samples. In contrast, CVS uses priori knowledge to select a causal feature and leverages it to obtain other causal features by performing a single CI test on per feature, which can well alleviate the data-inefficient problem, and thus can obtain more high-quality causal features than BAMB, EEMB, and EAMB. In addition, we observe that BAMB,

Table 2. Accuracy (%) of the 12 Cross-domain Tasks on Office-Caltech10 based on the Logistic Regression Classifier

Methods	FCBF	LUFS	BAMB	EEMB	EAMB	CVS	PCFS
C→A	39.77	46.35	41.54	42.80	38.20	48.64	**49.90**
C→W	33.12	36.94	36.31	38.22	35.03	40.13	**41.40**
C→D	26.44	36.61	30.51	33.56	26.44	36.27	**40.00**
A→C	34.73	38.82	34.82	34.11	34.37	38.82	**42.30**
A→W	28.66	36.31	32.48	29.94	28.66	34.39	**35.03**
A→D	31.53	29.15	30.85	28.47	30.85	**39.66**	34.92
W→C	28.58	33.84	26.27	30.37	29.03	33.84	**36.24**
W→A	29.02	36.85	30.17	30.58	29.23	**36.85**	36.53
W→D	65.61	76.43	59.24	64.33	65.61	**76.43**	72.61
D→C	25.47	29.83	25.11	20.48	26.18	31.70	**32.68**
D→A	25.26	32.05	27.14	21.71	25.68	32.05	**34.03**
D→W	52.54	70.17	45.08	32.20	53.90	68.47	**76.61**
Avg	35.06	41.95	34.96	33.90	35.26	43.11	**44.35**
Avg rank	5.63	2.63	5.21	5.42	5.42	2.21	1.50

The best result is highlighted in bold.

Table 3. Accuracy (%) of the 12 Cross-domain Tasks on Amazon Review based on the SVM Classifier

Methods	FCBF	LUFS	BAMB	EEMB	EAMB	CVS	PCFS
B→D	72.24	70.74	72.29	72.09	72.99	74.79	**77.04**
B→E	66.02	70.47	71.37	72.27	72.57	74.37	**75.48**
B→K	67.63	73.84	71.54	73.89	74.34	74.49	**77.59**
D→B	69.70	72.40	70.10	70.55	70.00	71.85	**75.55**
D→E	72.87	72.27	71.22	71.27	72.12	74.67	**75.48**
D→K	75.79	74.24	73.49	73.04	75.79	74.24	**78.54**
E→B	66.05	**72.15**	66.50	65.90	67.35	69.80	71.50
E→D	68.93	71.79	69.03	68.53	68.68	72.04	**72.59**
E→K	78.69	**82.99**	78.59	78.84	78.99	80.84	82.34
K→B	67.70	**71.45**	66.75	69.20	67.80	**71.45**	70.55
K→D	70.34	72.39	70.39	70.99	70.89	72.39	**72.79**
K→E	78.03	**81.63**	78.98	79.23	77.83	**81.63**	81.23
Avg	71.17	73.86	71.69	72.15	72.45	74.38	**75.89**
Avg rank	5.63	3.25	5.58	5.17	4.46	2.42	1.50

The best result is highlighted in bold.

EEMB, and EAMB do not perform better than two classical feature selection methods, i.e., FCBF and LUFS. This is because the collected data are complex and contain noise that has a great impact on the CI tests, BAMB, EEMB, and EAMB achieves low-quality causal features due to unreliable CI tests.

5.3.1 Statistical Tests. To further compare the performance difference between PCFS and the baseline methods, we conduct the Friedman test and Nemenyi test [9]. Here, we use the results achieved by the logistic regression classifier. Specifically, first, the Friedman test at the 5% significance level under the null-hypothesis that all methods are equivalent (i.e., the average ranks of

Table 4. Accuracy (%) of the 12 Cross-domain Tasks on Amazon Review based on the Naive Bayes Classifier

Methods	FCBF	LUFS	BAMB	EEMB	EAMB	CVS	PCFS
B→D	72.29	72.29	72.54	71.59	72.29	77.09	**79.84**
B→E	72.77	70.47	71.82	71.97	72.97	74.87	**77.58**
B→K	76.24	74.19	73.94	75.49	76.19	76.94	**79.14**
D→B	70.20	73.75	69.25	69.75	70.80	73.70	**78.10**
D→E	72.87	75.38	70.97	70.92	72.02	76.38	**77.93**
D→K	76.39	74.64	75.29	75.04	75.79	75.39	**79.09**
E→B	66.50	73.50	64.30	65.25	66.55	66.15	**72.70**
E→D	68.68	73.99	68.08	67.13	68.83	71.94	**74.64**
E→K	78.49	**83.34**	78.24	76.24	78.64	81.64	83.19
K→B	69.20	72.80	67.30	69.05	69.40	72.80	**74.55**
K→D	71.14	75.74	71.04	71.89	72.09	75.74	**76.39**
K→E	78.53	82.23	79.18	80.48	79.28	82.23	**83.43**
Avg	72.77	75.19	71.83	72.07	72.90	75.41	**78.05**
Avg rank	4.58	3.46	6.08	5.92	4.00	2.79	1.17

The best result is highlighted in bold.

Table 5. Accuracy (%) of the 12 Cross-domain Tasks on Office-Caltech10 based on the SVM Classifier

Methods	FCBF	LUFS	BAMB	EEMB	EAMB	CVS	PCFS
C→A	38.73	48.02	44.99	42.80	39.67	50.63	**54.07**
C→W	31.21	43.95	43.31	40.76	35.67	49.04	**52.23**
C→D	26.78	33.90	32.20	33.22	26.10	36.61	**36.95**
A→C	34.82	38.65	37.49	34.55	34.37	41.14	**41.76**
A→W	33.76	31.85	29.30	33.76	29.94	32.48	**34.39**
A→D	29.49	28.81	30.17	29.15	27.46	**33.56**	31.53
W→C	26.80	28.76	25.73	29.12	28.41	28.76	**30.01**
W→A	28.08	28.81	28.18	29.12	28.29	28.81	**30.38**
W→D	64.97	71.97	61.15	66.88	68.79	71.97	**75.80**
D→C	24.93	21.99	21.37	18.97	**25.02**	23.06	23.78
D→A	19.52	16.08	18.37	16.18	**20.15**	17.01	16.70
D→W	46.78	32.88	36.27	28.81	**47.46**	42.71	42.71
Avg	33.82	35.47	34.05	33.61	34.28	37.98	**39.19**
Avg rank	4.71	4.21	5.08	4.71	4.67	2.83	1.79

The best result is highlighted in bold.

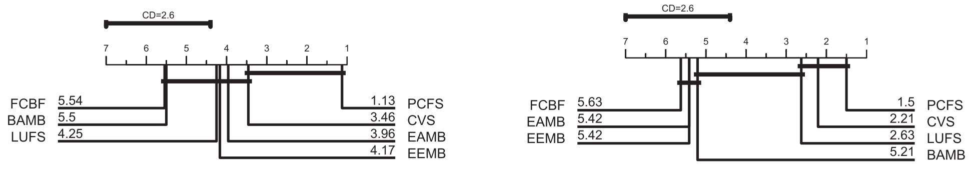
all methods are the same) is performed. The average ranks of all methods on the Amazon Review and Office-Caltech10 datasets are reported in the last column of Tables 1 and 2, respectively. We observe that the null hypothesis is rejected on the two datasets.

To further check whether there are significant differences between PCFS and its rivals, we conduct the Nemenyi test, which states that there exists a significant difference between two algorithms if the average ranks of the two algorithms differ by at least one **critical difference** (**CD**). The results are depicted in Figure 7. We observe that on Amazon Review, PCFS achieves comparable performance against CVS, and it completely surpasses the other methods. On Office-Caltech10,

Table 6. Accuracy (%) of the 12 Cross-domain Tasks on Office-Caltech10 based on the Naive Bayes Classifier

Methods	FCBF	LUFS	BAMB	EEMB	EAMB	CVS	PCFS
C→A	34.13	39.14	35.70	35.28	34.34	**39.77**	39.25
C→W	35.67	43.95	36.31	35.67	38.85	38.22	**38.85**
C→D	26.44	36.61	31.86	29.49	26.10	**41.36**	38.98
A→C	31.17	31.43	30.19	27.87	31.17	31.97	**32.06**
A→W	24.84	33.12	30.57	31.21	27.39	34.39	**35.67**
A→D	30.85	33.90	31.53	33.22	30.85	36.61	**36.95**
W→C	29.56	31.17	26.00	28.32	28.76	31.17	**31.79**
W→A	31.84	35.07	32.05	31.32	32.67	34.03	**36.22**
W→D	54.78	61.15	54.14	50.96	53.50	57.32	**65.61**
D→C	24.58	26.98	24.49	20.39	24.76	27.16	**28.58**
D→A	21.29	26.93	23.38	18.89	21.50	**28.39**	28.08
D→W	43.05	51.86	36.95	28.47	43.39	52.20	**57.29**
Avg	32.35	37.61	32.76	30.92	32.77	37.72	**39.11**
Avg rank	5.63	2.63	5.17	6.04	5.04	2.13	<u>1.38</u>

The best result is highlighted in bold.



(a) Amazon Review (b) Office-Caltech10

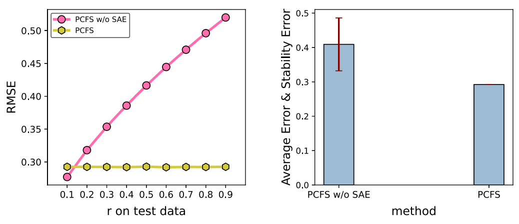
Fig. 7. Comparison of the PCFS against its rivals with the Nemenyi test.

PCFS achieves comparable performance against CVS and LUFS, and it significantly outperforms the other methods. We also see that PCFS is ranked the first place.

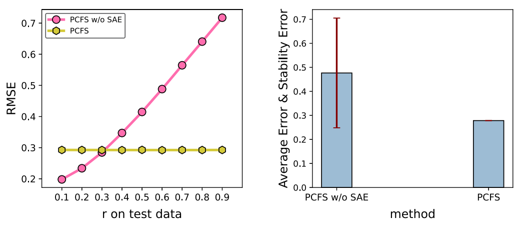
5.4 Analysis

To verify the effectiveness of mapping the treated and control group data into a low-dimensional nonlinear space using a supervised autoencoder, we propose a variant of PCFS, which uses the original treated and control group data to learn sample weights, referred to as "PCFS w/o SAE". PCFS is compared with "PCFS w/o SAE" using two synthetic datasets. The experiment results are shown in Figure 8. We observe that the RMSE value of PCFS is lower than that of "PCFS w/o SAE". Furthermore, PCFS achieves lower Average_Error and Stability_Error values, which shows the necessity of mapping the treated and control group data into a low-dimensional nonlinear space to improve the quality of learned sample weights. The reason for this is two-fold. First, learning low-dimensional representations of the treated and control group data captures non-linear relationships between features and reduces the impact of noisy features. Second, the number of features has a great impact on the quality of learned sample weights. Mapping the treated and control group data into a low-dimensional nonlinear space reduces the difficulty of learning sample weights, as the dimension of features decreases.

To further demonstrate the superiority of using a supervised autoencoder for learning low-dimensional representations of the treated and control group data, we introduce a variant of

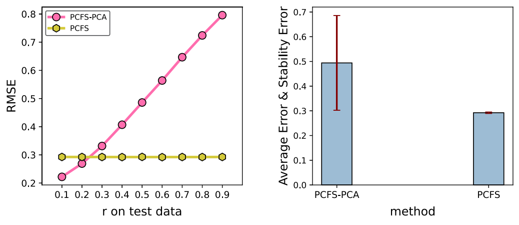


(a) Trained on $n = 2000$, $p = 40$, $r = 0.15$

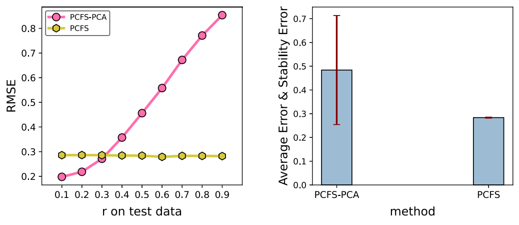


(b) Trained on $n = 2000$, $p = 40$, $r = 0.25$

Fig. 8. Comparison of PCFS against "PCFS w/o SAE".

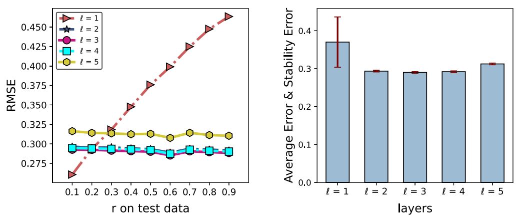


(a) Trained on $n = 2000$, $p = 40$, $r = 0.15$

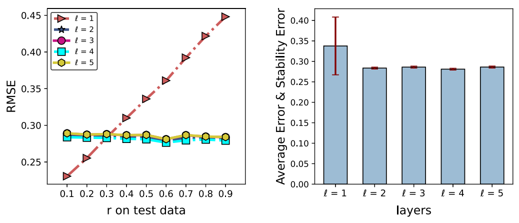


(b) Trained on $n = 2000$, $p = 40$, $r = 0.25$

Fig. 9. Comparison of PCFS against PCFS-PCA.



(a) Trained on $n = 2000$, $p = 40$, $r = 0.15$



(b) Trained on $n = 2000$, $p = 40$, $r = 0.25$

Fig. 10. RMSE, Average_Error, and Stability_Error of PCFS by varying the number of stacked layers l.

PCFS that utilizes PCA [18] to learn low-dimensional representations, and we refer to this version of PCFS as PCFS-PCA. For the sake of fairness, regardless of using autoencoders and PCA, the dimensions remain consistent after dimensionality reduction. Figure 9 shows the results of PCFS and PCFS-PCA on two synthetic datasets. We can see that PCFS achieves a lower RMSE value than that of PCFS-PCA. Additionally, the values of Average_Error and Stability_Error of PCFS-PCA are higher than those of PCFS, indicating that using the supervised autoencoder can learn high-quality feature representations. This is because the supervised autoencoder captures more complex nonlinear relationships between features compared with PCA.

To study the effect of stacked layers l, the experiments on two synthetic datasets by varying the number of l are performed, and the experimental results are depicted in Figure 10. From the experimental results, we observe that PCFS achieves poor performance when the number of l is set to 1. The possible reason is that the autoencoder with one hidden layer may not well capture the complex nonlinear relationships between features. We also note that PCFS obtains promising performance when the number of l is greater than or equal to 2. Since the number of parameters is exponential with the number of l, the number of l is set to 2 in our experiments.

To demonstrate the effectiveness of progressively removing irrelevant features, we perform experiments on two synthetic datasets by varying the number of iterations (k) of removing irrelevant

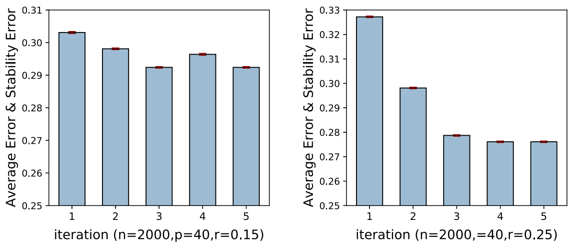


Fig. 11. Average_Error and Stability_Error of PCFS by varying the number of removing irrelevant features.

features. The results are depicted in Figure 11. We observe that PCFS performing only one iteration achieves the highest Average_Error value. Furthermore, we see that the value of Average_Error decreases with the increase of the number of iterations when $k \leq 3$. We also note that the value of Average_Error tends to be stable when $k \geq 3$. In summary, from the experimental results, we can draw the conclusion that the proposed strategy of progressively removing irrelevant features is effective. The reason for this is that PCFS gradually removes irrelevant features and reduces the influence of irrelevant features on sample weighting, and thus improves the quality of learned sample weights and obtains more accurate causal effect estimation.

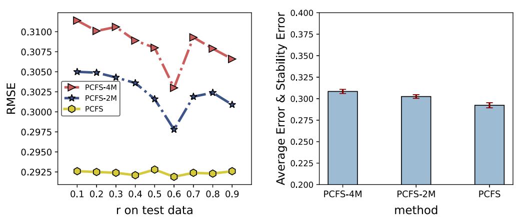
To evaluate the effectiveness of using first-order moments to measure the distribution discrepancy between the treated group and the control group, we propose two variants of PCFS, which use second-order moments and fourth-order moments instead of first-order moments used in Equation (6) to measure the distribution difference, referred as PCFS-2M and PCFS-4M, respectively. The calculation process of second-order moments is defined as follows:

$$\sum_{j=1}^{p} \left\| \frac{1}{W^T \odot \mathbf{X}_{\cdot,j}} \sum_{i=1}^{n} \Delta_t \otimes \Delta_t - \frac{1}{W^T \odot (1 - \mathbf{X}_{\cdot,j})} \sum_{i=1}^{n} \Delta_c \otimes \Delta_c \right\|_2^2,$$

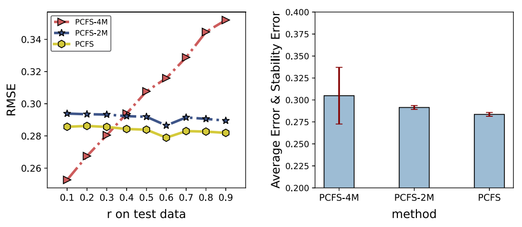
where $\Delta_t = \mathbf{X}_{\cdot,j} \cdot (W_i \cdot \xi(\mathbf{X}_{\cdot,-j})^T - \mu_t)$, $\Delta_c = (1 - \mathbf{X}_{\cdot,j}) \cdot (W_i \cdot \xi(\mathbf{X}_{\cdot,-j})^T - \mu_c)$, $\mu_t = \frac{\xi(\mathbf{X}_{\cdot,-j}^T)\cdot(W \odot \mathbf{X}_{\cdot,j})}{W^T \odot \mathbf{X}_{\cdot,j}}$, $\mu_c = \frac{\xi(\mathbf{X}_{\cdot,-j}^T)\cdot(W \odot (1-\mathbf{X}_{\cdot,j}))}{W^T \odot (1-\mathbf{X}_{\cdot,j})}$, $\Delta_t \otimes \Delta_t = \Delta_t^T \Delta_t$. The computation process of fourth-order moments is formalized as follows:

$$\sum_{j=1}^{p} \left\| \frac{1}{W^T \odot \mathbf{X}_{\cdot,j}} \sum_{i=1}^{n} \Delta_t \otimes \Delta_t \otimes \Delta_t \otimes \Delta_t - \frac{1}{W^T \odot (1 - \mathbf{X}_{\cdot,j})} \sum_{i=1}^{n} \Delta_c \otimes \Delta_c \otimes \Delta_c \otimes \Delta_c \right\|_2^2.$$

We carry out experiments on two synthetic datasets and have plotted the RMSE, Average_Error, and Stability_Error of PCFS, PCFS-2M, and PCFS-4M in Figure 12. It can be seen that PCFS is superior to PCFS-2M and PCFS-4M. To be specific, the RMSE and Average_Error values obtained by PCFS are the lowest. References [11, 20] indicate that first-order moments can also achieve promising performance when the features are binary variables. Compared to first-order moments, in theory, second-order and fourth-order moments can better measure the distribution discrepancy. That is to say, it is advantageous to use second-order and fourth-order moments to align the distribution between the treated and control groups corresponding to a single feature. However, PCFS needs to balance the distribution between the treated group and the control group corresponding to each feature. Compared with first-order moments, the calculation processes of second-order and fourth-order moments are more complex, and thus it is more difficult to optimize them and learn desirable sample weights for balancing the distribution between different groups, especially for high-dimensional datasets.

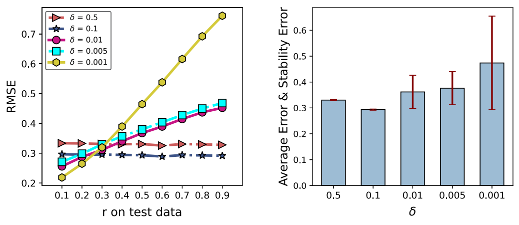


(a) Trained on $n = 2000$, $p = 40$, $r = 0.15$

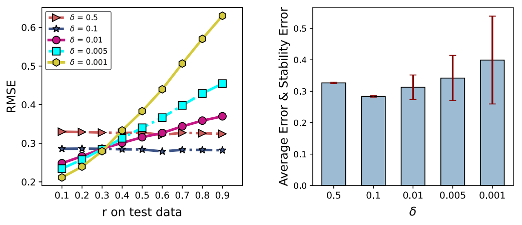


(b) Trained on $n = 2000$, $p = 40$, $r = 0.25$

Fig. 12. Comparison of PCFS against PCFS-2M and PCFS-4M.



(a) Trained on $n = 2000$, $p = 40$, $r = 0.15$



(b) Trained on $n = 2000$, $p = 40$, $r = 0.25$

Fig. 13. RMSE, Average_Error, and Stability_Error of PCFS by varying the value of δ.

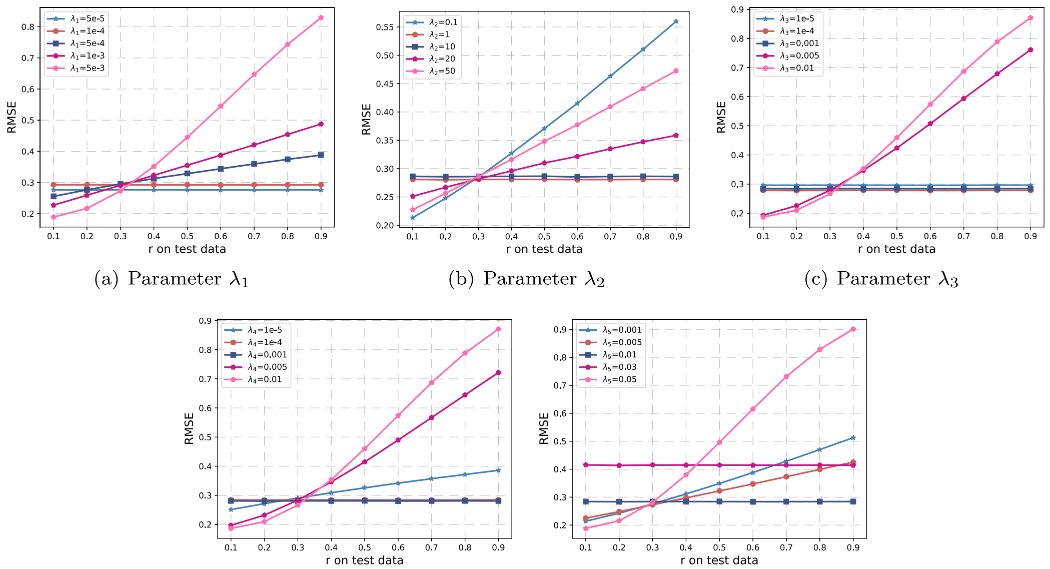
To investigate the sensitivity of PCFS with respect to the threshold δ, we conduct experiments on two synthetic datasets by varying the values of δ that are selected from the set $\{0.5, 0.1, 0.01, 0.005, 0.001\}$. The experimental results are shown in Figure 13. We can see from the figure that PCFS is sensitive to the value of δ. If the value of δ is too large, partial causal features may be discarded, while if it is too small, some irrelevant features may be selected, leading to a degradation in generalization performance.

To investigate the effect of the hyper-parameters λ_1, λ_2, λ_3, λ_4, and λ_5, parameter sensitivity analysis on one synthetic dataset is performed. When we tune a specific parameter, the values of the remaining parameters are freezed. The results are depicted in Figure 14. We see that PCFS is sensitive to the five hyper-parameters. Additionally, for $\lambda_1 \sim \lambda_5$, [5e-5,1e-4], [1,10], [1e-4,0.001], [1e-4,001], [0.01,0.03] are the optimal value ranges.

6 CONCLUSION

In this article, we investigate the problem of causal feature selection under sample selection bias and have proposed a novel algorithm, PCFS, for causal feature selection. Different from previous causal feature selection methods, PCFS progressively removes irrelevant features to learn causal features by estimating the causal effect between features and the class variable. Extensive experiments have been performed on synthetic and real-world datasets, and the experimental results have illustrated that PCFS has a stronger generalization ability than other state-of-the-art algorithms. The success of PCFS indicates the importance of estimating the causal effect when performing causal feature selection.

Despite the encouraging performance, PCFS still suffers from two limitations. First, PCFS can only deal with discrete binary features, because when estimating the causal effect of the feature T, it needs to divide samples into a treated group ($T = 1$) or a control group ($T = 0$) based on the value of the samples in the feature T. Second, PCFS needs to learn the weight for each sample, and thus it is difficult for PCFS to deal with datasets with large training samples. In the future, we will explore PCFS to handle continuous, high-dimensional datasets.



(a) Parameter λ_1 (b) Parameter λ_2 (c) Parameter λ_3

(d) Parameter λ_4 (e) Parameter λ_5

Fig. 14. Parameter sensitivity study for PCFS (Trained on n = 2,000, p = 40, r = 0.25).

REFERENCES

[1] Constantin F. Aliferis, Ioannis Tsamardinos, and Alexander R. Statnikov. 2003. HITON: A novel Markov blanket algorithm for optimal variable selection. In *Proceedings of the American Medical Informatics Association Annual Symposium.*

[2] Susan Athey, Guido W. Imbens, and Stefan Wager. 2016. *Efficient Inference of Average Treatment Effects in High Dimensions via Approximate Residual Balancing.* Technical Report.

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

[4] Ruichu Cai, Jiawei Chen, Zijian Li, Wei Chen, Keli Zhang, Junjian Ye, Zhuozhang Li, Xiaoyan Yang, and Zhenjie Zhang. 2021. Time series domain adaptation via sparse associative structure alignment. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence.* 6859–6867.

[5] Ruichu Cai, Jiahao Li, Zhenjie Zhang, Xiaoyan Yang, and Zhifeng Hao. 2020. DACH: Domain adaptation without domain information. *IEEE Transactions on Neural Networks and Learning Systems* 31, 12 (2020), 5055–5067.

[6] Ruichu Cai, Jie Qiao, Zhenjie Zhang, and Zhifeng Hao. 2018. SELF: Structural equational likelihood framework for causal discovery. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence.* 1787–1794.

[7] Ruichu Cai, Siyu Wu, Jie Qiao, Zhifeng Hao, Keli Zhang, and Xi Zhang. 2022. THPs: Topological hawkes processes for learning causal structure on event sequences. *IEEE Transactions on Neural Networks and Learning Systems.* (2022), 1–15. DOI: 10.1109/TNNLS.2022.3175622

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

[9] Janez Demsar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7 (2006), 1–30.

[10] Xianjie Guo, Kui Yu, Fuyuan Cao, Pei-Pei Li, and Hao Wang. 2022. Error-aware Markov blanket learning for causal feature selection. *Information Sciences* 589 (2022), 849–877.

[11] Kun Kuang, Peng Cui, Susan Athey, Ruoxuan Xiong, and Bo Li. 2018. Stable prediction across unknown environments. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 1617–1626.

[12] Kun Kuang, Peng Cui, Bo Li, Meng Jiang, and Shiqiang Yang. 2017. Estimating treatment effect in the wild via differentiated confounder balancing. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 265–274.

[13] Kun Kuang, Haotian Wang, Yue Liu, Ruoxuan Xiong, Runze Wu, Weiming Lu, Yue Ting Zhuang, Fei Wu, Peng Cui, and Bo Li. 2023. Stable prediction with leveraging seed variable. *IEEE Transactions on Knowledge and Data Engineering* 35, 6 (2023), 6392–6404.

[14] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. 2018. Feature selection: A data perspective. *Computing Surveys* 50, 6 (2018), 94:1–94:45.

[15] Zhaolong Ling, Kui Yu, Hao Wang, Lin Liu, Wei Ding, and Xindong Wu. 2019. BAMB: A balanced Markov blanket discovery approach to feature selection. *ACM Transactions on Intelligent Systems and Technology* 10, 5 (2019), 52:1–52:25.

[16] Dimitris Margaritis and Sebastian Thrun. 1999. Bayesian network induction via local neighborhoods. *Advances in Neural Information Processing Systems* 12 (1999), 505–511.

[17] Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

[18] Karl Pearson. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2, 11 (1901), 559–572.

[19] Donald B. Rubin. 1973. Matching to remove bias in observational studies. *Biometrics* (1973), 159–183.

[20] Zheyan Shen, Peng Cui, Kun Kuang, Bo Li, and Peixuan Chen. 2018. Causally regularized learning with agnostic data selection bias. In *Proceedings of the 2018 ACM Multimedia Conference on Multimedia Conference*. 411–419.

[21] Claudia Shi, David M. Blei, and Victor Veitch. 2019. Adapting neural networks for the estimation of treatment effects. In *Proceedings of the Advances in Neural Information Processing Systems*. 2503–2513.

[22] Baochen Sun, Jiashi Feng, and Kate Saenko. 2016. Return of frustratingly easy domain adaptation. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. 2058–2065.

[23] Chang Tang, Xinzhong Zhu, Jiajia Chen, Pichao Wang, Xinwang Liu, and Jie Tian. 2018. Robust graph regularized unsupervised feature selection. *Expert Systems with Applications* 96 (2018), 64–76.

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

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

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

[27] Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S. Yu. 2018. Visual domain adaptation with manifold embedded distribution alignment. In *Proceedings of the ACM Multimedia Conference on Multimedia Conference*. 402–410.

[28] Xingyu Wu, Bingbing Jiang, Kui Yu, Chunyan Miao, and Huanhuan Chen. 2020. Accurate Markov boundary discovery for causal feature selection. *IEEE Transactions on Cybernetics* 50, 12 (2020), 4983–4996.

[29] Shuai Yang, Kui Yu, Fuyuan Cao, Lin Liu, Hao Wang, and Jiuyong Li. 2023. Learning causal representations for robust domain adaptation. *IEEE Transactions on Knowledge and Data Engineering* 35, 3 (2023), 2750–2764.

[30] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. 2021. A survey on causal inference. *ACM Transactions on Knowledge Discovery from Data* 15, 5 (2021), 74:1–74:46.

[31] Sandeep Yaramakala and Dimitris Margaritis. 2005. Speculative Markov blanket discovery for optimal feature selection. In *Proceedings of the 5th IEEE International Conference on Data Mining*. 4.

[32] Kui Yu, Xianjie Guo, Lin Liu, Jiuyong Li, Hao Wang, Zhaolong Ling, and Xindong Wu. 2020. Causality-based feature selection: Methods and evaluations. *Computing Surveys* 53, 5 (2020), 111:1–111:36.

[33] Kui Yu, Lin Liu, and Jiuyong Li. 2021. A unified view of causal and non-causal feature selection. *ACM Transactions on Knowledge Discovery from Data* 15, 4 (2021), 63:1–63:46.

[34] Kui Yu, Lin Liu, Jiuyong Li, Wei Ding, and Thuc Duy Le. 2020. Multi-source causal feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 9 (2020), 2240–2256.

[35] Lei Yu and Huan Liu. 2004. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research* 5 (2004), 1205–1224.

Received 18 November 2022; revised 5 May 2023; accepted 5 June 2023