Estimating Treatment Effect in the Wild via Differentiated Confounder Balancing

Kun Kuang*
Department of Computer Science
Tsinghua University
kk14@mails.tsinghua.edu.cn

Peng Cui
Department of Computer Science
Tsinghua University
cuip@tsinghua.edu.cn

Bo Li
School of Economics and Management
Tsinghua University
libo@sem.tsinghua.edu.cn

Meng Jiang
Department of Computer Science and Engineering, University of Notre Dame
mjiang2@nd.edu

Shiqiang Yang
Department of Computer Science
Tsinghua University
yangshq@tsinghua.edu.cn

ABSTRACT

The key challenge on estimating treatment effect in the wild observational studies is to handle confounding bias induced by imbalance of the confounder distributions between treated and control units. Traditional methods remove confounding bias by re-weighting units with supposedly accurate propensity score estimation under the unconfoundedness assumption. Controlling high-dimensional variables may make the unconfoundedness assumption more plausible, but poses new challenge on accurate propensity score estimation. One strand of recent literature seeks to directly optimize weights to balance confounder distributions, bypassing propensity score estimation. But existing balancing methods fail to do selection and differentiation among the pool of a large number of potential confounders, leading to possible underperformance in many high dimensional settings. In this paper, we propose a data-driven Differentiated Confounder Balancing (DCB) algorithm to jointly select confounders, differentiate weights of confounders and balance confounder distributions for treatment effect estimation in the wild high dimensional settings. The synergistic learning algorithm we proposed is more capable of reducing the confounding bias in many observational studies. To validate the effectiveness of our DCB algorithm, we conduct extensive experiments on both synthetic and real datasets. The experimental results clearly demonstrate that our DCB algorithm outperforms the state-of-the-art methods on treatment effect estimation.

KEYWORDS

Treatment Effect Estimation; Confounding Bias; Differentiated Confounder Balancing

*Tsinghua National Laboratory for Information Science and Technology (TNList).

KDD '17, August 13-17, 2017, Halifax, NS, Canada

DOI: 10.1145/3097983.3098032

1 INTRODUCTION

Owing to the popularity of Big Data, abundant data are accumulated in various domains such as healthcare and advertising. At the same time, many machine learning and data mining methods are proposed to exploit these data for prediction, aiming to estimate the future outcome in the application of interest. These methods have been proved to be successful in prediction-oriented applications. However, the lack of interpretability of most predictive algorithms makes them less attractive in many settings, especially those requiring decision making. How to improve the explanability of learning algorithms is of paramount importance for both academic research and real applications.

Causal inference is a powerful statistical modeling tool for explanatory analysis. One fundamental problem in causal inference is treatment effect estimation, and its key challenge is to remove the confounding bias induced by the different confounder distributions between treated and control units. The gold standard approach for removing confounding bias is to conduct randomized experiments like A/B testing. But fully randomized experiments are usually expensive [14] and sometimes infeasible [5]. Therefore, many methods are proposed to estimate treatment effect directly from observational data under the unconfoundedness assumption [22]. Most of them adopt the propensity score to reweight units for removing confounding bias [3, 4, 6]. Although these methods are gaining ground in applied work, they require correct model specification on treatment assignment or accurate propensity score estimation. In big data scenarios, controlling high-dimensional variables may make the unconfoundedness assumption more plausible, but poses new challenge on accurate propensity score estimation. Recently, some researchers proposed to balance confounder distributions by directly optimizing the weights, without modeling or estimating the propensity scores [2, 7, 11, 26]. But they balance all observed variables equally without screening and differentiation of confounders, leading to poor performance in high-dimensional settings. Overall, the previous methods can work well in well-designed experimental settings or observational studies with grounded model assumptions and prior knowledge.

In the wild big data scenarios, however, there are almost always a large number of additional or mostly uncontrolled confounders and identified variables, and the correlations

among them are complex and unknown in the real world [23]. Hence, we face the following challenges in estimating treatment effect in the wild observational studies: (1) **unknown model structure of the interactions among variables**: As stated in [23], pretty much everything in the real world interacts with everything else, to some degree, and their interactions are complicated due to the complex nature of the real world. We hardly know the real model structure among variables in the wild, so we cannot make any model specification a priori for removing confounding bias. (2) **high-dimensional and noisy variables**: In big data scenario, there are always a large number of observed variables, but not all these variables are confounders and different confounders contribute unequally to the confounding bias in data. Usually, we do not have sufficient prior knowledge to justify the inclusion of hundreds or even thousands of variables. How to differentiate the confounders and their confounding bias is quite difficult.

To address these challenges, we propose a data-driven method, named Differentiated Confounder Balancing (DCB) algorithm. The method is based on the framework of confounder balancing, but in contrast with previous methods which balance all variables equally, we argue that some variables should not be regarded as confounders and we theoretically prove that the weights of confounders should be differentiated in confounder balancing. Motivated by this, we propose an integrated regularization algorithm to jointly select confounders, differentiate weights of confounders and balance confounder distributions for treatment effect estimation. During the treatment effect estimation, the selected confounders and their weights are used to adjust the weights of units, so that the confounder distributions, approximated by their moments, over all units can be balanced in treated and control groups. We validate our DCB algorithm with extensive experiments on both synthetic and real datasets. The results show that our algorithm outperforms the state-of-the-art methods on treatment effect estimation in observational studies.

The main contributions of this paper are:

- We address the new challenges of estimating treatment effect in big data scenarios with high-dimensional noisy variables and insufficient prior knowledge on variable interactions, which is beyond the capability of previous methods.
- We propose a novel DCB algorithm to jointly select confounders, optimize the confounder weights and sample weights for confounder balancing, and simultaneously estimate the treatment effect in the wild observational studies.
- The advantages of DCB algorithm are demonstrated in both synthetic and real datasets. We also show that our method can significantly help to improve the prediction performance with real online advertising dataset.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduces our differentiated confounder balancing estimator. Section 4 proposes the algorithm that accurately infers the treatment effect. Section 5 gives the experimental results. Finally, Section 6 concludes the paper.

2 RELATED WORK

Existing weighting based treatment effect estimation methods in observational studies either employ propensity score or optimize balance weights directly.

The propensity score was first proposed by Rosenbaum and Rubin [22], where it was estimated via a logistic regression. Then many other machine learning algorithms (e.g., lasso [8, 10], boosting regression [18], bagged CART and neural network [25]) are employed for propensity score estimation. Various estimators have been proposed based on propensity score, such as propensity score matching, inverse propensity weighting and double robust estimators [3, 4, 6, 16]. But these estimators require correct model specification on treatment assignment or precise estimation of the propensity score, which may not be the case in many applications [2].

Recently, researchers proposed new weighting based estimators by focusing on confounder balancing directly [2, 7, 11, 12, 26]. Hainmueller [11] introduced entropy balancing to directly adjust sample weights to the specified sample moments while moving the sample weights as little as possible. Athey et al. [2] proposed approximate residual balancing for sample weights learning via a lasso residual regression adjustment. Zubizarreta [26] learnt the stable balancing weights via minimizing its variance and adjusting for confounder balancing directly. Chan et al. [7] considered a wide class calibration weights constructed to attain confounder balancing directly. Imai et al. [12] introduced covariate balancing propensity score, which models treatment assignment while optimizing covariates balancing. Most of these methods are non-parametrical and require no propensity score estimation, but they do not differentiate the confounders by treating all observed variables as confounders and balanced all of them equally, leading to possible poor performance on treatment effect estimation in the setting of high dimensional variables.

Hence, it is very likely to improve the treatment effect estimation efficiency by fine-tuned selection and differentiated methods. To achieve the goal, we propose a differentiated confounder balancing algorithm to jointly optimize confounder weights and sample weights for precise treatment effect estimation.

3 PROBLEM AND OUR ESTIMATOR

In this section, we first give the notations and problem formulation, then revisit traditional confounder balancing estimators, and propose a novel estimator via differentiated confounder balancing.

3.1 Notations and Problem Formulation

Our goal is to estimate the treatment effect based on potential outcome framework [13, 22]. With the framework, we define a treatment as a random variable T and a potential outcome as $Y(t)$ which corresponds to a specific treatment $T = t$.

Table 1: Symbols and definitions.

Symbol	Definition
n_t (n_c)	Sample size for treated (control) group
n	Sample size ($n = n_t + n_c$)
p	Dimension of observed (augmented) variables
$T \in \mathbb{R}^{n\times 1}$	Treatment
$Y \in \mathbb{R}^{n\times 1}$	Outcome
$\mathbf{X} \in \mathbb{R}^{n\times p}$	Observed variables
$\mathbf{X_t} \in \mathbb{R}^{n_t\times p}$	Observed variables of treated units
$\mathbf{M_t} \in \mathbb{R}^{n_t\times p}$	Augmented variables of treated units
$\mathbf{M_c} \in \mathbb{R}^{n_c\times p}$	Augmented variables of control units
$W \in \mathbb{R}^{n_c\times 1}$	Sample weights on control units
$\beta \in \mathbb{R}^{p\times 1}$	Confounder weights

In this paper, we only focus on binary treatment, that is $t \in \{0, 1\}$. We define the units which received treatment ($T = 1$) as treated units and the other units with $T = 0$ as control units. Then, for each unit indexed by $i = 1, 2, \cdots, n$, we observe a treatment T_i, an outcome Y_i^{obs} and a vector of observed variables $X_i \in \mathbb{R}^{p\times 1}$, where the observed outcome Y_i^{obs} of unit i denotes by:

$$Y_i^{obs} = Y_i(T_i) = T_i \cdot Y_i(1) + (1 - T_i) \cdot Y_i(0). \quad (1)$$

The numbers of treated and control units are equal to n_t and n_c, and the dimension of all observed variables is p. In our paper, for any column vector $\mathbf{v} = (v_1, v_2, \cdots, v_m)^T$, let $\|\mathbf{v}\|_\infty = \max(|v_1|, \cdots, |v_m|)$, $\|\mathbf{v}\|_2^2 = \sum_{i=1}^m v_i^2$, and $\|\mathbf{v}\|_1 = \sum_{i=1}^m |v_i|$.

Throughout this paper, we assume the unconfoundedness [22] condition is satisfied.

Assumption 1: Unconfoundedness. The distribution of treatment is independent of potential outcome when given the observed variables. Formally, $T \perp \big(Y(0), Y(1)\big) | \mathbf{X}$.

In this paper, we focus on estimating the Average Treatment effect on the Treated (ATT), which represents the mean (average) difference between the potential outcomes under treated and control status among the treated subgroup. Formally, the ATT is defined as:

$$ATT = E\big[Y(1)|T = 1\big] - E\big[Y(0)|T = 1\big], \quad (2)$$

where $Y(1)$ and $Y(0)$ represent the potential outcome of units with treatment status as treated $T = 1$ and control $T = 0$, respectively. Our method proposed in this paper can be readily extended to estimate the Average Treatment effect on the Control (ATC) and hence the Average Treatment Effect (ATE) for the whole population.

In Eq. (2), $E[Y(1)|T = 1]$ can be straightforwardly estimated by the sample analog $\sum_{i:T_i=1} \frac{1}{n_t} \cdot Y_i^{obs}$. But it is cumbersome to estimate $E[Y(0)|T = 1]$, since we cannot observe the potential outcome $Y(0)$ for the treated units. Under Assumption 1, $E[Y(0)|T = 1]$ is usually estimated by re-weighting techniques for removing the confounding bias. The re-weighting methods form the surrogates of the unobserved potential outcome $(Y(0)|T = 1)$ by re-weighting the control units with sample weights W to make the confounder distributions on control units mimic the distributions on treated units. Then with the sample weights W on control units, we can estimate the ATT by:

$$\widehat{ATT} = \textstyle\sum_{i:T_i=1} \frac{1}{n_t} \cdot Y_i^{obs} - \sum_{j:T_j=0} W_j \cdot Y_j^{obs}. \quad (3)$$

3.2 Revisiting on Confounder Balancing

It can been seen from Eq. (3) that the ATT estimation produces a sample weights learning problem. The classical approaches for sample weights learning are propensity score based methods [3, 4, 6]. The good performance of these methods hinges on the correct model specification for treatment assignment or accurate estimates of the propensity scores. Hence, the performance of these methods is often poor in the wild observational studies, where the model structure among variables is unknown.

To reduce the model dependency for applying on data in the wild, researchers proposed non-parametric methods to optimize the sample weights W by focusing on confounder balancing directly [2, 11]. The motivation behind these methods is that the confounders can be balanced by their moments, which uniquely determine their distributions. Therefore, they learn the sample weights W by:

$$W = \arg\min_W \|\overline{\mathbf{X}}_t - \textstyle\sum_{j:T_j=0} W_j \cdot X_j\|_2^2 \quad (4)$$

or

$$W = \arg\min_W \|\overline{\mathbf{X}}_t - \textstyle\sum_{j:T_j=0} W_j \cdot X_j\|_\infty^2, \quad (5)$$

where the $\overline{\mathbf{X}}_t$ represents the mean value of observed variables on treated units. The direct confounder balancing methods based on Eq. (4) or (5) can be applied on data in the wild. But they balance all observed variables equally without differentiating confounders, which results in poor performance in the setting of high dimensional variables.

3.3 Differentiated Confounder Balancing

To precisely estimate the treatment effect with high dimensional observational data in the wild, we propose to simultaneously learn confounder weights and sample weights. The confounder weights can determine which variable is included and its share of contribution on confounding bias, and the sample weights are designed for confounder balancing.

To be specific, we jointly optimize the confounder weights and sample weights by learning following optimization under some constraints to be clarified later.

$$W = \arg\min_W \big(\beta^T \cdot (\overline{\mathbf{X}}_t - \textstyle\sum_{j:T_j=0} W_j \cdot X_j)\big)^2, \quad (6)$$

where $W \in \mathbb{R}^{n_c\times 1}$ is sample weights and $\beta \in \mathbb{R}^{p\times 1}$ is the confounder weights. In Eq. (6), the confounder weights β differentiate the roles of each confounder in the balancing process, which helps for better removing the confounding bias in the wild observational studies.

Next, we give theoretical analysis on how to differentiate confounders weights with following proposition.

PROPOSITION 3.1. *In observational studies, different confounders make unequal confounding bias on ATT with their own weights, and the weights can be learned via regressing potential outcome $Y(0)$ on observed variables $\boldsymbol{X}$.*

The general relationship among observed variables $\mathbf{X}$, treatment T and outcome Y can be represented as:

$$Y = f(\mathbf{X}) + T \cdot g(\mathbf{X}) + \epsilon, \quad (7)$$

where the *true ATT* is $E(g(\mathbf{X}_t))$, and the potential outcome $Y(0)$ can be represented by:

$$Y(0) = f(\mathbf{X}) + \epsilon. \tag{8}$$

We prove Proposition 3.1 with following assumption.

Assumption 2: Linearity. The regression of potential outcome $Y(0)$ on observed variables $\mathbf{X}$ is linear, that is $f(\mathbf{X}) = c + \alpha\mathbf{X}$.

Under assumption 2, we can rewrite estimator of $\widehat{ATT}$ as:

$$\begin{aligned}\widehat{ATT} &= \textstyle\sum_{i:T_i=1} \frac{1}{n_t} Y_i^{obs} - \sum_{j:T_j=0} W_j Y_j^{obs} \\ &= \textstyle\sum_{i:T_i=1} \frac{1}{n_t}(c + \alpha X_i + g(X_i) + \epsilon_i) - \sum_{j:T_j=0} W_j(c + \alpha X_j + \epsilon_j) \\ &= E(g(\mathbf{X}_t)) + (\textstyle\sum_{i:T_i=1} \frac{1}{n_t}\alpha X_i - \sum_{j:T_j=0} W_j \alpha X_j) + \phi(\epsilon) \\ &= ATT + \textstyle\sum_{k=1}^{p} \alpha_k (\sum_{i:T_i=1} \frac{1}{n_t} X_{i,k} - \sum_{j:T_j=0} W_j X_{j,k}) + \phi(\epsilon).\end{aligned}$$

where $\phi(\epsilon) = \sum_{i:T_i=1} \frac{1}{n_t}\epsilon_i - \sum_{j:T_j=0} W_j \epsilon_j$ refers to the difference of noises between treated and control units, and $\phi(\epsilon) \simeq 0$ with Gaussian noise. To reduce the bias of $\widehat{ATT}$, we need regulate the term $\sum_{k=1}^{p} \alpha_k \cdot (\sum_{i:T_i=1} \frac{1}{n_t} X_{i,k} - \sum_{j:T_j=0} W_j X_{j,k})$, where $(\sum_{i:T_i=1} \frac{1}{n_t} X_{i,k} - \sum_{j:T_j=0} W_j X_{j,k})$ means the difference of the k^{th} confounder between treated and control units. The parameter α_k represents the confounding bias weight of the k^{th} confounder, and it is the coefficient of X_k in the function $f(\mathbf{X})$. Hence, we can learn the confounder weights from the regression of potential outcome $Y(0)$ on observed variables $\mathbf{X}$ under *Linearity* assumption.

Actually, the regression of potential outcome $Y(0)$ against on observed variables $\mathbf{X}$ is infeasible, because of the counterfactual problem, that we cannot observe the potential outcome $Y(0)$ for treated units. Here we utilize the sample weights W again to facilitate the construction of surrogates for the potential outcomes $Y(0)$ of the treated units. We will elaborate on this later.

When the function $f(\mathbf{X})$ is nonlinear, that is $f(\mathbf{X})$ allows for powers and interactions among observed variables. It is conceptually easy to extend above results under *Linearity* assumption to bound the bias of *ATT* with Taylor expansion on $f(\mathbf{X})$ by balancing not only observed variables, but also their powers and interactions. Therefore, when $f(\mathbf{X})$ is nonlinear, we have to balance the augmented variables $\mathbf{M} = (\mathbf{X}, \mathbf{X}^2, X_iX_j, \mathbf{X}^3, X_iX_jX_k, \cdots)$, and learn the confounder weights by regressing the potential outcome $Y(0)$ on augmented variables $\mathbf{M}$.

The differentiated confounder balance strategy we propose focuses on mere bias control of the ATT estimator by mean modelling of the potential outcome. A possible refined strategy can be developed by incorporating variance modeling of the potential outcome, which allows one to control the variance and eventually the MSE of the resultant ATT estimator. We leave this extension to future work.

4 OPTIMIZATION

In this section, we give details of our DCB algorithm for treatment effect estimation, and introduce parameters tuning method for the "no ground truth" problem in observational causal inference.

4.1 Algorithm

With Proposition 3.1, we know the *ATT* estimator is affected by the unbalance of the observed variables, and their high order terms. That is the augmented variables $\mathbf{M}$:

$$\mathbf{M} = (\mathbf{X}, \mathbf{X}^2, X_iX_j, \mathbf{X}^3, X_iX_jX_k, \cdots). \tag{9}$$

Combining Eq. (6)&(9) and Proposition 3.1, we give our objective function to jointly optimize sample weights and confounder weights for ATT estimation in observational studies as:

$$\begin{aligned} \min \quad & \left(\beta^T \cdot (\overline{\mathbf{M}}_t - \mathbf{M}_c^T W)\right)^2, \\ s.t. \quad & \textstyle\sum_{j:T_j=0} (1 + W_j) \cdot (Y_j - M_j \cdot \beta)^2 \le \lambda, \\ & \|W\|_2^2 \le \delta, \quad \|\beta\|_2^2 \le \mu, \quad \|\beta\|_1 \le \nu, \\ & \mathbf{1}^T W = 1 \quad and \quad W \succeq 0, \end{aligned} \tag{10}$$

where W is the sample weights and β is the confounder weights. $\overline{\mathbf{M}}_t$ represents the mean value of augmented variables on treated units. $\sum_{j:T_j=0}(1+W_j)\cdot(Y_j - M_j\cdot\beta)^2$ refers to the loss function of potential outcome $Y(0)$ when learning the confounder weights, including potential outcome loss on both control units $\sum_{j:T_j=0}(Y_j - M_j\cdot\beta)^2$ and treated units $\sum_{j:T_j=0} W_j \cdot (Y_j - M_j\cdot\beta)^2$, which is again a surrogate by weighting. With the constraints $\|\beta\|_2^2 \le \mu$ and $\|\beta\|_1 \le \nu$, we can remove the non-confounders and smooth the confounder weights. The formula $\mathbf{1}^T W = 1$ normalizes the sample weights on control units to add up to one, with the sample weights on treated units. The terms $W \succeq 0$ constraint each of sample weights is non-negative. With norm $\|W\|_2^2 \le \delta$, we can reduce the variance of the sample weights to achieve stability.

These lead to the following optimization problem, which is to minimize $\mathcal{J}(W, \beta)$ with constraints on parameters W.

$$\begin{aligned} \mathcal{J}(W,\beta) \quad = \quad & \left(\beta^T \cdot (\overline{\mathbf{M}}_t - \mathbf{M}_c^T W)\right)^2 \\ & + \lambda \textstyle\sum_{j:T_j=0} (1 + W_j) \cdot (Y_j - M_j \cdot \beta)^2 \\ & + \delta\|W\|_2^2 + \mu\|\beta\|_2^2 + \nu\|\beta\|_1, \\ s.t. \quad & \mathbf{1}^T W = 1 \quad and \quad W \succeq 0. \end{aligned} \tag{11}$$

Here, we propose an iterative method to minimize the above objective function (11).

Firstly, we initialize sample weights $W = \{1/n_c, \cdots, 1/n_c\}^T$ and confounder weights $\beta = \{1/p, \cdots, 1/p\}^T$. Once the initial values are given, in each iteration, we first update β by fixing W, and then update W by fixing β. These steps are described below:

Update β: When fixing W, the problem (11) is equivalent to optimize following objective function:

$$\begin{aligned} \mathcal{J}(\beta) \quad = \quad & \left(\beta^T \cdot (\overline{\mathbf{M}}_t - \mathbf{M}_c^T W)\right)^2 + \mu\|\beta\|_2^2 + \nu\|\beta\|_1 \\ & + \lambda \textstyle\sum_{j:T_j=0} (1 + W_j) \cdot (Y_j - M_j \cdot \beta)^2 \end{aligned} \tag{12}$$

which is a standard ℓ_1 norm regularized least squares problem and can be solved with any LASSO (or elastic net) solver. Here, we use the proximal gradient algorithm [19] with proximal operator to solve the objective function in (12).

Algorithm 1 Differentiated Confounder Balancing (DCB)

Input: Tradeoff parameters $\lambda > 0$, $\delta > 0$, $\mu > 0$, $\nu > 0$, Augmented Variables Matrix on treat units $\mathbf{M}_t$, Augmented Variables Matrix on control units $\mathbf{M}_c$ and Outcome Y.

Output: Confounder Weights β and Sample Weights W

1: Initialize Confounder Weights $\beta^{(0)}$ and Sample Weights $W^{(0)}$
2: Calculate the current value of $\mathcal{J}(W,\beta)^{(0)} = \mathcal{J}(W^{(0)}, \beta^{(0)})$ with Equation (11)
3: Initialize the iteration variable $t \leftarrow 0$
4: **repeat**
5: $t \leftarrow t + 1$
6: Update $\beta^{(t)}$ by solving $\mathcal{J}(\beta^{(t-1)})$ in Equation (12)
7: Update $W^{(t)}$ by solving $\mathcal{J}(W^{(t-1)})$ in Equation (13)
8: Calculate $\mathcal{J}(W,\beta)^{(t)} = \mathcal{J}(W^{(t)}, \beta^{(t)})$
9: **until** $\mathcal{J}(W,\beta)^{(t)}$ converges or max iteration is reached
10: **return** β, W.

Update W: By fixing β, we can obtain W by optimizing (11). It is equivalent to optimize following objective function:

$$\begin{aligned}\mathcal{J}(W) \quad &= \quad \left(\beta^T \cdot (\overline{\mathbf{M}}_t - \mathbf{M}_c^T W)\right)^2 + \delta\|W\|_2^2 \\ &\quad +\lambda \textstyle\sum_{j:T_j=0}(1+W_j)\cdot(Y_j - M_j\cdot\beta)^2, \\ s.t. \quad &\mathbf{1}^T W = 1 \quad and \quad W \succeq 0.\end{aligned} \tag{13}$$

For ensuring non-negative of W with constraint $W \succeq 0$, we let $W = \omega \odot \omega$, where $\omega \in \mathbb{R}^{n_c \times 1}$ and $\odot$ refers to the Hadamard product. Then the problem (13) can be reformulated as:

$$\begin{aligned}\mathcal{J}(\omega) \quad &= \quad \left(\beta^T \cdot (\overline{\mathbf{M}}_t - \mathbf{M}_c^T(\omega\odot\omega))\right)^2 + \delta\|\omega\odot\omega\|_2^2 \\ &\quad +\lambda \textstyle\sum_{j:T_j=0}(1+\omega_j\odot\omega_j)\cdot(Y_j - M_j\cdot\beta)^2, \\ s.t. \quad &\mathbf{1}^T(\omega\odot\omega) = 1.\end{aligned} \tag{14}$$

The partial gradient of term $\mathcal{J}(\omega)$ with respect to ω is:

$$\begin{aligned}\frac{\partial\mathcal{J}(\omega)}{\partial\omega} \quad &= \quad -4(\beta^T\cdot(\overline{\mathbf{M}}_t - \mathbf{M}_c^T(\omega\odot\omega)))\cdot\mathbf{M}_c\cdot\beta\odot\omega \\ &\quad +4\delta\omega\odot\omega\odot\omega + 2\lambda\omega\odot(Y_c - \mathbf{M}_c\cdot\beta)^2.\end{aligned}$$

Then we determine the step size a with line search, and update ω at t^{th} iteration as:

$$\omega^{(t)} = \omega^{(t-1)} - a\cdot\frac{\partial\mathcal{J}(\omega^{(t-1)})}{\partial\omega^{(t-1)}}.$$

With constraint $\mathbf{1}^T(\omega\odot\omega) = 1$, we normalize $\omega^{(t)}$ as:

$$\omega^{(t)} = \frac{\omega^{(t)}}{\sqrt{\mathbf{1}^T(\omega^{(t)}\odot\omega^{(t)})}}.$$

Then, we update $W^{(t)}$ at t^{th} iteration with:

$$W^{(t)} = \omega^{(t)}\odot\omega^{(t)}.$$

We update β and W iteratively until the objective function (11) converges. The whole algorithm is summarized in Algorithm 1.

Finally, with the optimized sample weights W by our DCB algorithm, we can estimate the ATT with Eq. (3).

REMARK 1. *The confounder weights β in our algorithm can be applied for outcome residual adjustment as [2] did. Then, with the optimized β and W in our algorithm, we can estimate the ATT by combining confounder balancing and residual adjustment as:*

$$\widehat{ATT} = \textstyle\sum_{i:T_i=1}\frac{1}{n_t}\cdot Y_i^{obs} - \left(\overline{\mathbf{M}}_t\cdot\beta + \sum_{j:T_j=0} W_j(Y_j^{obs} - M_i\cdot\beta)\right).$$

4.2 Complexity Analysis

During the procedure of optimization, the main cost is to calculate the loss $\mathcal{J}(W,\beta)$, update confounder weights β and sample weights W. We analyze the time complexity of each of them respectively. For the calculation of the loss, its complexity is $O(np)$, where n is the sample size and p is the dimension of (augmented) variables. For updating β, this is standard LASSO problem and its complexity is $O(np)$. For updating W, the complexity is dominated by the step of calculating the partial gradients of function $\mathcal{J}(\omega)$ with respect to variable ω. The complexity of $\frac{\partial\mathcal{J}(\omega)}{\partial\omega}$ is $O(np)$.

In total, the complexity of each iteration in Algorithm 1 is $O(np)$.

4.3 Parameters Tuning

No ground truth for parameters tuning is the main challenge of causal inference in observational studies. To address this challenge, we apply matching method to estimate the ATT and set it as the "approximal ground truth" as [1, 15, 21] did. Specially, for each treated unit i, we find its closet match among control units as follow:

$$match(i) = \arg\min_{j:T_j=0}\|X_i - X_j\|_2^2. \tag{15}$$

To make the matching approximate to exactly matching, we drop unit i if $match(i) > \varepsilon$. Then, we can obtain the "approximal ground truth" by comparing the average outcome between the matched treated and control units.

With the "approximal ground truth", we can tune parameters for our algorithm and baselines with cross validation by grid searching.

5 EXPERIMENTS

In this section, we evaluate our algorithm on both synthetic and real world datasets, comparing with the state-of-the-art methods.

5.1 Baseline Estimators

We implement following baseline estimators to evaluate the ATT for comparison.

- *Directly Estimator* $\widehat{ATT}_{dir}$: It evaluates the ATT by directly comparing the average outcome between the treated and control units. It ignores the confounding bias in data.
- *IPW Estimator* $\widehat{ATT}_{IPW}$ [22]: It evaluates the ATT via reweighting units with inverse of propensity score. It relies on correct model specification on treatment assignment.
- *Doubly Robust Estimator* $\widehat{ATT}_{DR}$ [4]: It evaluates the ATT with combination of IPW and regression method. It relies on correct specification of propensity score or outcome regression models.

- *Entropy Balancing Estimator* $\widehat{ATT}_{ENT}$ [11]: It evaluates the ATT by directly balancing on confounders and entropy loss on sample weights. It ignores the confounder weights.
- *Approximate Residual Balancing Estimator* $\widehat{ATT}_{ARB}$ [2]: It evaluates the ATT by combining weighting adjustment via directly balancing on confounders and regression adjustment on outcome. It ignores the confounder weights.

In this paper, we implemented $\widehat{ATT}_{IPW}$ and $\widehat{ATT}_{DR}$ with **lasso regression** for variables selection.

5.2 Experiments on Synthetic Data

In this section, we introduce how to generate the synthetic datasets and demonstrate the effectiveness of our DCB algorithm with extensive experiments.

5.2.1 Dataset. To generate the synthetic datasets, we consider two sample sizes $n = \{2000, 5000\}$ and also vary the dimension of observed variables $p = \{50, 100\}$. We first generate the observed variables $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p)$ with independent Gaussian distributions as:

$$\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p \quad \overset{iid}{\sim} \quad \mathcal{N}(0, 1),$$

where $\mathbf{x}_i$ represents value of the i^{th} variable in $\mathbf{X}$.

To test the robustness of all estimators, we generate the binary treatment variable T from a logistic function (T_{logit}) and a misspecified function (T_{missp}) as

$$T_{logit} \sim Bernoulli(1/(1+\exp(-\textstyle\sum_{i=1}^{p\cdot r_c} s_c \cdot x_i + \mathcal{N}(0,1)))), and$$
$$T_{missp} = 1 \; if \; \textstyle\sum_{i=1}^{p\cdot r_c} s_c \cdot x_i + \mathcal{N}(0,1) > 0, \; T_{missp} = 0 \; otherwise$$

where we vary both **confounding rate** r_c and **confounding strength** s_c from 0 to 1. The confounding rate represents the ration of confounders to all observed variables, and the confounding strength refers to the bias strength of confounders on treatment.

We generate the outcome Y from a linear function (Y_{linear}) and a nonlinear function (Y_{nonlin}) as:

$$\begin{aligned} Y_{linear} &= T + \textstyle\sum_{j=1}^{p}\{I(mod(j,2) \equiv 0) \cdot (\frac{j}{2} + T) \cdot \mathbf{x}_j\} + \mathcal{N}(0,3), \\ Y_{nonlin} &= T + \textstyle\sum_{j=1}^{p}\{I(mod(j,2) \equiv 0) \cdot (\frac{j}{2} + T) \cdot \mathbf{x}_j\} + \mathcal{N}(0,3) \\ &\quad + \textstyle\sum_{j=1}^{p-1}\{I(mod(j,10) \equiv 1) \cdot \frac{p}{2} \cdot (x_j^2 + x_j \cdot x_{j+1})\}, \end{aligned}$$

where the $I(\cdot)$ is the indicator function and function $mod(x, y)$ returns the modulus after division of x by y.

Under different settings on treatment T and outcome Y, we know the *true* ATT in simulation. We evaluate the ATT with our algorithm, comparing with baselines.

5.2.2 Results. To evaluate the performance of our proposed method, we carry out the experiments for 100 times independently. Based on the estimated ATT ($\widehat{ATT}$), we calculate its $Bias$, standard deviations (SD), mean absolute errors (MAE) and root mean square errors ($RMSE$) with following definitions:

$$\begin{aligned} Bias &= |\tfrac{1}{K}\textstyle\sum_{k=1}^{K}\widehat{ATT}_k - ATT| \\ SD &= \sqrt{\tfrac{1}{K}\textstyle\sum_{k=1}^{K}(\widehat{ATT}_k - \tfrac{1}{K}\textstyle\sum_{k=1}^{K}\widehat{ATT}_k)^2} \\ MAE &= \tfrac{1}{K}\textstyle\sum_{k=1}^{K}|\widehat{ATT}_k - ATT| \end{aligned}$$

$$RMSE = \sqrt{\tfrac{1}{K}\textstyle\sum_{k=1}^{K}(\widehat{ATT}_k - ATT)^2}$$

where K is the experimental times, $\widehat{ATT}_k$ is the estimated ATT in k^{th} experiment and ATT represents the *true treatment effect*.

We report the results in Table 2 for settings $T = T_{logit}, Y = Y_{linear}$ and $T = T_{missp}, Y = Y_{nonlin}$. The results under other settings are reported in the supplemental[1] to save space.

From Table 2, we have following observations and analyses:

- Directly estimator fails when confounders are associated with both treatment and outcome. From our results, we find Directly estimator makes huge error on ATT estimation, since it ignores the confounding bias in data.
- IPW and DR estimators have poor performance in the setting of high dimensional variables or when the model specifications are incorrect. IPW and DR estimators make huge error under setting 3 and setting 4, where $T = T_{missp}$ and $Y = Y_{nonlin}$.
- ENT estimator has good performance only when the parameters $s_c = 0.2$ under setting 2, where the confounding bias is small in data, but its performance deteriorates as the confounding bias increasing. Since it ignores the confounder weights, which makes it unable to effectively remove the confounding bias in data.
- ARB estimator achieves better performance than other baselines in most of time, since it is nonparametric method with regression adjustment. However, it is far inferior to our proposed estimator. The key reason is that it balances all observed variables equally.
- Our proposed DCB estimator, by jointly optimizing both sample weights and confounder weights, achieves significant improvements over the baselines in different settings, when varying sample size n, dimension of variables p, confounding rate r_c and confounding strength s_c.

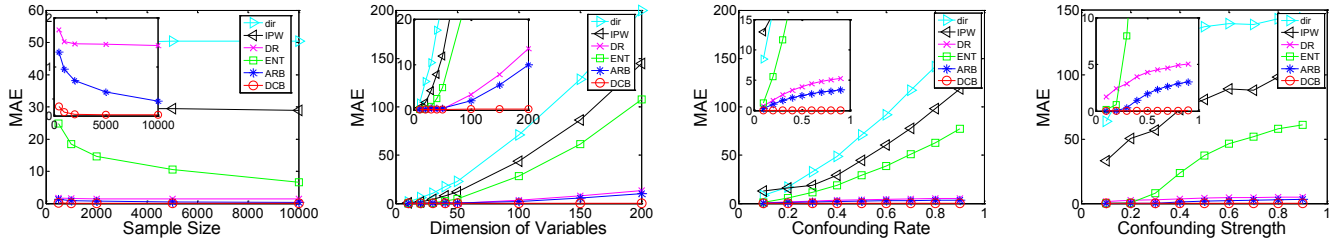
We show the robustness of our estimator in Figure 1 by varying the sample size n, dimension of variables p, confounding rate r_c and confounding strength s_c. From Figure 1, we find that as we decrease n or increase p, r_c and s_c, the MAE of our estimator is consistent stable and small, while the MAE of baseline estimators increases continuously. This demonstrates that our proposed estimator is more precise and robust than the baselines.

5.2.3 Parameter Analysis. In our DCB algorithm, we have hype-parameters λ, δ, μ and ν. As mentioned before, we tuned these parameters in our experiments with cross validation by grid searching, and each parameter is uniformly varied from {0.001,0.01,0.1,1,10,100,1000}. We displayed the $Bias$ of treatment effect estimation with respect to λ, δ, μ and ν, respectively. As seen from Figure 2, the $Bias$ do not change too much and the performance are relatively stable when parameters $\lambda \geq 1$ and $\delta, \mu, \nu \leq 1$. From Figure 2a, we can see the $Bias$ is huge when parameter λ is too small. The main reason is that small value of λ would slack the

[1] https://www.dropbox.com/s/99s9rmkybo7jnbv/paper_supplemental.pdf

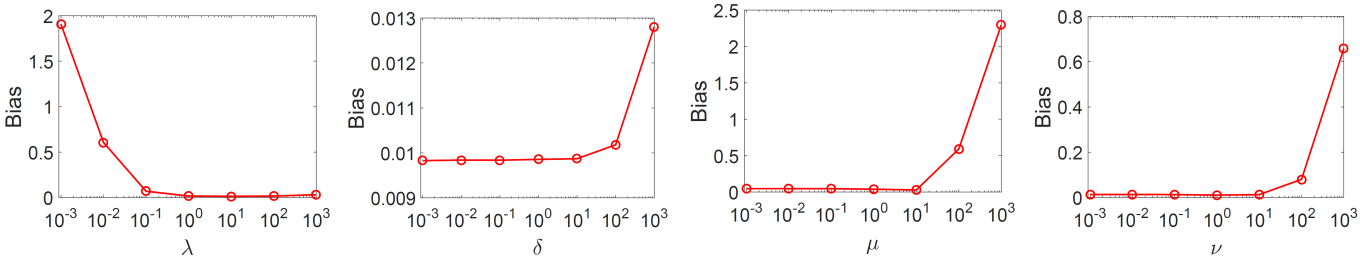
Table 2: Results on synthetic dataset in different settings. The *Bias* refers to the absolute error between the true and estimated ATT. The *SD*, *MAE* and *RMSE* represent the standard deviations, mean absolute errors and root mean square errors of estimated ATT ($\widehat{ATT}$) after 100 times independently experiments, respectively. The smaller *Bias*, *SD*, *MAE* and *RMSE*, the better.

Setting 1: $T = T_{logit}$, $Y = Y_{linear}$ **and** $s_c = 1$													
	n/p	$n = 2000, p = 50$			$n = 2000, p = 100$			$n = 5000, p = 50$			$n = 5000, p = 100$		
r_c	Estimator	*Bias* (SD)	MAE	RMSE	*Bias* (SD)	MAE	RMSE	*Bias* (SD)	MAE	RMSE	*Bias* (SD)	MAE	RMSE
$r_c = 0.2$	$\widehat{ATT}_{dir}$	6.483 (3.460)	6.682	7.349	18.60 (8.859)	18.67	20.61	6.420 (2.050)	6.420	6.739	18.53 (5.148)	18.53	19.23
	$\widehat{ATT}_{IPW}$	2.220 (6.224)	4.866	6.609	8.365 (15.40)	14.47	17.52	1.907 (4.092)	3.648	4.514	8.033 (9.852)	10.52	12.71
	$\widehat{ATT}_{DR}$	0.118 (0.307)	0.253	0.329	1.591 (0.512)	1.591	1.672	0.059 (0.174)	0.145	0.183	1.446 (0.337)	1.446	1.485
	$\widehat{ATT}_{ENT}$	0.371 (0.477)	0.453	0.605	4.924 (3.167)	5.052	5.855	0.046 (0.254)	0.210	0.258	2.425 (1.229)	2.429	2.719
	$\widehat{ATT}_{ARB}$	0.074 (0.472)	0.376	0.477	0.868 (0.435)	0.881	0.971	0.027 (0.269)	0.217	0.270	0.365 (0.371)	0.447	0.520
	$\widehat{ATT}_{DCB}$	**0.014** (0.121)	**0.099**	**0.122**	**0.006** (0.119)	**0.101**	**0.119**	**0.001** (0.073)	**0.053**	**0.073**	**0.001** (0.085)	**0.067**	**0.085**
$r_c = 0.8$	$\widehat{ATT}_{dir}$	51.06 (3.725)	51.06	51.19	143.0 (9.389)	143.0	143.3	50.45 (1.900)	50.45	50.48	142.1 (5.647)	142.1	142.2
	$\widehat{ATT}_{IPW}$	29.99 (4.048)	29.99	30.26	98.24 (8.462)	98.24	98.60	29.38 (2.216)	29.38	29.46	96.86 (5.899)	96.86	97.04
	$\widehat{ATT}_{DR}$	0.345 (0.253)	0.367	0.428	4.492 (0.333)	4.492	4.504	0.338 (0.136)	0.338	0.365	4.306 (0.227)	4.306	4.312
	$\widehat{ATT}_{ENT}$	15.06 (1.745)	15.06	15.16	63.02 (4.551)	63.02	63.19	10.09 (1.473)	10.09	10.19	51.99 (3.206)	51.99	52.09
	$\widehat{ATT}_{ARB}$	0.231 (0.645)	0.553	0.685	2.909 (0.491)	2.909	2.951	0.189 (0.504)	0.428	0.538	2.259 (0.468)	2.259	2.307
	$\widehat{ATT}_{DCB}$	**0.003** (0.127)	**0.102**	**0.127**	**0.020** (0.135)	**0.114**	**0.136**	**0.003** (0.088)	**0.072**	**0.088**	**0.012** (0.088)	**0.073**	**0.089**
Setting 2: $T = T_{logit}$, $Y = Y_{linear}$ **and** $r_c = 0.5$													
	n/p	$n = 2000, p = 50$			$n = 2000, p = 100$			$n = 5000, p = 50$			$n = 5000, p = 100$		
s_c	Estimator	*Bias* (SD)	MAE	RMSE	*Bias* (SD)	MAE	RMSE	*Bias* (SD)	MAE	RMSE	*Bias* (SD)	MAE	RMSE
$s_c = 0.2$	$\widehat{ATT}_{dir}$	11.80 (3.243)	11.80	12.24	43.38 (9.170)	43.38	44.34	11.53 (2.142)	11.53	11.73	42.64 (6.103)	42.64	43.07
	$\widehat{ATT}_{IPW}$	3.897 (2.759)	4.144	4.775	18.37 (8.317)	18.38	20.17	3.873 (2.055)	3.875	4.384	17.13 (5.971)	17.13	18.14
	$\widehat{ATT}_{DR}$	0.053 (0.150)	0.124	0.159	1.255 (0.265)	1.255	1.283	0.056 (0.104)	0.090	0.118	1.148 (0.180)	1.148	1.162
	$\widehat{ATT}_{ENT}$	0.023 (0.168)	0.128	0.170	0.174 (0.193)	0.208	0.260	**0.001** (0.116)	0.090	0.116	0.089 (0.119)	0.120	0.149
	$\widehat{ATT}_{ARB}$	**0.002** (0.170)	0.129	0.170	**0.011** (0.184)	0.151	0.185	0.004 (0.119)	0.094	0.120	0.006 (0.121)	0.093	0.122
	$\widehat{ATT}_{DCB}$	0.011 (0.107)	**0.086**	**0.107**	0.013 (0.098)	**0.080**	**0.099**	0.003 (0.065)	**0.053**	**0.065**	**0.004** (0.073)	**0.060**	**0.073**
$s_c = 0.8$	$\widehat{ATT}_{dir}$	22.81 (3.610)	22.81	23.09	69.28 (9.608)	69.28	69.94	21.91 (1.908)	21.91	21.99	68.72 (5.410)	68.72	68.93
	$\widehat{ATT}_{IPW}$	9.984 (4.878)	10.15	11.11	40.64 (12.48)	40.64	42.51	9.263 (3.615)	9.263	9.943	40.31 (7.185)	40.31	40.94
	$\widehat{ATT}_{DR}$	0.185 (0.256)	0.256	0.316	3.234 (0.449)	3.234	3.265	0.177 (0.166)	0.205	0.243	3.051 (0.245)	3.051	3.061
	$\widehat{ATT}_{ENT}$	2.805 (1.153)	2.805	3.033	23.53 (4.432)	23.53	23.94	0.742 (0.447)	0.759	0.866	15.97 (2.519)	15.97	16.16
	$\widehat{ATT}_{ARB}$	0.059 (0.564)	0.455	0.567	1.861 (0.491)	1.861	1.924	0.005 (0.408)	0.327	0.408	1.133 (0.451)	1.133	1.219
	$\widehat{ATT}_{DCB}$	**0.007** (0.124)	**0.102**	**0.124**	**0.015** (0.123)	**0.102**	**0.124**	**0.001** (0.083)	**0.067**	**0.083**	**0.017** (0.076)	**0.063**	**0.078**
Setting 3: $T = T_{missp}$, $Y = Y_{nonlin}$ **and** $s_c = 1$													
	n/p	$n = 2000, p = 50$			$n = 2000, p = 100$			$n = 5000, p = 50$			$n = 5000, p = 100$		
r_c	Estimator	*Bias* (SD)	MAE	RMSE	*Bias* (SD)	MAE	RMSE	*Bias* (SD)	MAE	RMSE	*Bias* (SD)	MAE	RMSE
$r_c = 0.2$	$\widehat{ATT}_{dir}$	6.527 (5.367)	7.041	8.450	18.67 (14.04)	20.01	23.36	7.340 (3.425)	7.366	8.099	20.54 (9.992)	20.54	22.84
	$\widehat{ATT}_{IPW}$	5.061 (8.998)	8.542	10.32	17.31 (19.22)	21.90	25.86	6.707 (6.494)	7.934	9.336	19.81 (15.04)	21.79	24.87
	$\widehat{ATT}_{DR}$	6.334 (8.628)	8.562	10.70	23.65 (26.32)	29.16	35.38	6.493 (6.698)	7.637	9.329	23.44 (16.62)	24.77	28.74
	$\widehat{ATT}_{ENT}$	3.770 (2.166)	3.842	4.348	13.46 (5.854)	13.58	14.68	3.096 (1.285)	3.102	3.352	12.16 (3.585)	12.16	12.68
	$\widehat{ATT}_{ARB}$	0.643 (0.292)	0.647	0.706	3.757 (0.483)	3.757	3.788	0.512 (0.247)	0.517	0.569	3.288 (0.262)	3.288	3.299
	$\widehat{ATT}_{DCB}$	**0.016** (0.316)	**0.263**	**0.317**	**0.021** (0.364)	**0.294**	**0.365**	**0.017** (0.169)	**0.139**	**0.169**	**0.082** (0.214)	**0.183**	**0.230**
$r_c = 0.8$	$\widehat{ATT}_{dir}$	53.26 (5.308)	53.26	53.53	145.2 (13.47)	145.2	145.9	53.12 (3.673)	53.12	53.24	145.2 (9.247)	145.2	145.4
	$\widehat{ATT}_{IPW}$	39.46 (6.404)	39.46	39.97	113.0 (16.91)	113.0	114.3	39.04 (4.424)	39.04	39.29	111.7 (10.19)	111.7	112.1
	$\widehat{ATT}_{DR}$	15.12 (8.433)	15.40	17.31	34.07 (28.29)	37.09	44.28	14.26 (5.613)	14.28	15.33	30.92 (15.90)	31.70	34.77
	$\widehat{ATT}_{ENT}$	29.83 (1.795)	29.83	29.89	97.32 (6.507)	97.32	97.54	25.73 (1.155)	25.73	25.76	85.63 (3.114)	85.63	85.68
	$\widehat{ATT}_{ARB}$	1.342 (0.337)	1.342	1.384	7.440 (0.566)	7.440	7.462	1.102 (0.230)	1.102	1.126	6.526 (0.325)	6.526	6.535
	$\widehat{ATT}_{DCB}$	**0.076** (0.321)	**0.255**	**0.330**	**0.024** (0.388)	**0.298**	**0.389**	**0.003** (0.207)	**0.171**	**0.207**	**0.021** (0.304)	**0.248**	**0.305**
Setting 4: $T = T_{missp}$, $Y = Y_{nonlin}$ **and** $r_c = 0.5$													
	n/p	$n = 2000, p = 50$			$n = 2000, p = 100$			$n = 5000, p = 50$			$n = 5000, p = 100$		
s_c	Estimator	*Bias* (SD)	MAE	RMSE	*Bias* (SD)	MAE	RMSE	*Bias* (SD)	MAE	RMSE	*Bias* (SD)	MAE	RMSE
$s_c = 0.2$	$\widehat{ATT}_{dir}$	18.01 (5.556)	18.01	18.84	59.49 (14.13)	59.49	61.15	18.01 (3.178)	18.01	18.29	60.34 (8.923)	60.34	60.99
	$\widehat{ATT}_{IPW}$	7.288 (6.605)	8.429	9.836	32.24 (19.66)	33.23	37.76	7.372 (4.505)	7.516	8.639	33.39 (12.87)	33.39	35.78
	$\widehat{ATT}_{DR}$	3.408 (5.953)	5.735	6.859	13.87 (21.90)	21.33	25.92	3.130 (4.146)	4.360	5.194	13.87 (12.53)	15.54	18.69
	$\widehat{ATT}_{ENT}$	1.812 (0.818)	1.812	1.988	25.54 (6.241)	25.54	26.29	0.273 (0.160)	0.282	0.317	14.49 (2.800)	14.49	14.76
	$\widehat{ATT}_{ARB}$	0.159 (0.254)	0.244	0.300	2.960 (0.385)	2.960	2.985	0.055 (0.150)	0.131	0.160	1.899 (0.241)	1.899	1.915
	$\widehat{ATT}_{DCB}$	**0.005** (0.223)	**0.178**	**0.223**	**0.011** (0.288)	**0.228**	**0.288**	**0.012** (0.120)	**0.095**	**0.120**	**0.025** (0.158)	**0.125**	**0.160**
$s_c = 0.8$	$\widehat{ATT}_{dir}$	24.58 (5.276)	24.58	25.14	72.30 (13.95)	72.30	73.63	24.10 (3.219)	24.10	24.31	71.20 (8.771)	71.20	71.74
	$\widehat{ATT}_{IPW}$	18.34 (6.819)	18.34	19.56	57.07 (18.02)	57.07	59.85	17.65 (4.755)	17.65	18.28	54.95 (9.861)	54.95	55.83
	$\widehat{ATT}_{DR}$	11.23 (8.757)	12.46	14.24	32.35 (26.22)	35.39	41.65	11.17 (5.492)	11.17	12.44	28.06 (14.24)	28.29	31.46
	$\widehat{ATT}_{ENT}$	12.88 (1.956)	12.88	13.03	48.40 (5.818)	48.40	48.75	10.46 (1.315)	10.46	10.55	40.79 (2.773)	40.79	40.88
	$\widehat{ATT}_{ARB}$	0.993 (0.343)	0.993	1.050	6.052 (0.525)	6.052	6.075	0.807 (0.255)	0.807	0.846	5.176 (0.279)	5.176	5.183
	$\widehat{ATT}_{DCB}$	**0.042** (0.310)	**0.246**	**0.313**	**0.023** (0.364)	**0.306**	**0.365**	**0.006** (0.211)	**0.167**	**0.211**	**0.013** (0.237)	**0.194**	**0.238**



(a) sample size n (b) dimension of variables p (c) confounding rate r_c (d) confounding strength s_c

Figure 1: MAE on ATT estimation when varying different parameters, with setting $T = T_{logit}$, $Y = Y_{linear}$. The sub-figure on the top left corner of each main figure is plot by freezing MAE on Y-axis with a limit. The results show our proposed DCB estimator is more precise and robust than the baselines.



(a) (b) (c) (d)

Figure 2: The effect of hyper-parameters λ, δ, μ and ν.

constrain on confounder weights learning, resulting in imprecise confounder weights, even the trivial solution $\beta = 0$. From Figure 2c and 2d, we find that the *Bias* increased as the increasing of μ and ν. This is because that large value of μ and ν would enforce the confounder weights close to *zero*. The Figure 2b demonstrates that the performance is insensitive to the parameter δ. To sum up, we can easily obtain the best hype-parameters for our DCB algorithm.

5.3 Experiments on Real Data

In this section, we apply our DCB algorithm on two real datasets for ATT estimation and application, including the LaLonde dataset and an online advertising dataset.

5.3.1 LaLonde Dataset. First, we apply our DCB algorithm on the LaLonde [17] dataset[2], a canonical benchmark in the causal inference literature [9, 11]. The LaLonde dataset used in our paper consists of two parts. The first part comes from a randomized experiment on a large scale job training program, the National Support Work Demonstration (NSW)[3]. In the second part data, as [11] did, we replace the control group in randomized experiment with another control group drawn from the Current Population Survey-Social Security Administration file (CPS-1) where the measured covariates are the same with the experimental data. The treatment in this data is whether the participant attend the particular job training program or not, and the outcome is the earning in the year 1978. The data contains 10 raw observed variables, including earnings and employment status for year 1974 and 1975, education status (years of schooling and an indicator for completed high school degree), age, ethnicity (indicators for black and hispanic) and the married status.

Overall, there are *185* program participants (the treated units) and *260* nonparticipants (the control units) in the experimental data NSW. In the observational data CPS-1, we have *185* program participants and *15,992* nonparticipants. The randomized experimental data NSW provide the ground truth for estimating the ATT of the program. We estimate the ATT with the observational data CPS-1, comparing our proposed algorithm with the baselines.

Experimental Settings. In our experiments, we randomly split the observational data CPS-1 as 6 partitions, with the first 3 partitions, we train our model and baseline models for parameters tuning with cross validation by grid searching, and test model performance and robustness with the last 3 partitions. We conduct our DCB algorithm and baselines on two variables sets, V-RAW and V-INTERACTION. The V-RAW refers to the 10 raw observed variables, and the V-INTERACTION refers to the set of raw variables, their pairwise one-way interaction, and their squared terms.

Results. We report the results in Table 3, where the smaller *Bias* and *SD*, the better. From the results, we have following observations. (1) Directly estimator failed due to the existing of confounding bias in the LaLonde data. (2) IPW generates a big error on ATT estimation in both V-RAW and V-INTERACTION settings. The main reason is that

[2]The dataset is available at http://users.nber.org/~rdehejia/data/nswdata2.html

[3]Notice that we focus on the Dehejia and Wahha sampled dataset of the LaLonde.

Table 3: ATT estimation results on LaLonde dataset, where the true ATT from randomized experiment is 1,794. The smaller *Bias* and SD, the better.

Variables Set	V-RAW		V-INTERACTION	
Estimator	$\widehat{ATT}$	*Bias* (SD)	$\widehat{ATT}$	*Bias* (SD)
$\widehat{ATT}_{dir}$	-8471	10265 (374)	-8471	10265 (374)
$\widehat{ATT}_{IPW}$	-4481	6275 (971)	-4365	6159 (1024)
$\widehat{ATT}_{DR}$	1154	639 (491)	1590	204 (812)
$\widehat{ATT}_{ENT}$	1535	259 (995)	1405	388 (787)
$\widehat{ATT}_{ARB}$	1537	257 (996)	1627	167 (957)
$\widehat{ATT}_{DCB}$	1958	**164** (728)	1836	**43** (716)

Table 4: Confounder weights learnt from our DCB algorithm with V-RAW variables set.

Rank	Confounder	Weight
1	Earnings 1975	0.335
2	Earnings 1974	0.241
3	Employed 1975	0.141
4	Education	0.138
5	Employed 1974	0.050
6	Married	0.039
7	High School Degree 1975	0.017
8	Age	-0.013
9	Black	-0.003
10	Hispanic	-0.001

the specification model of IPW is incorrect and the sample size between treated and control units is unbalanced. (3) Our proposed DCB estimator achieves the best performance compared with the baselines, since our estimator simultaneously optimizes sample weights and confounder weights, and requires no any model specification on treatment assignment. (4) Our estimator obtains a more accurate ATT estimation with V-INTERACTION variables set than V-RAW variables set. This demonstrates that our estimator can achieve a better confounder balancing with including the high order terms of observed variables in augmented variables.

In Table 4, we show the confounder weights optimized by our DCB algorithm with V-RAW variables set. From this table, we know that the confounders Earnings 1975 & 1974 and Education are very important for the outcome (Earning 1978), but the Black and Hispanic have few effects on the outcome. Thus the confounders of Earnings 1975 & 1974 and Education are more important, and should be balanced first.

5.3.2 Online Advertising Dataset. The real online advertising dataset we used is collected from Tencecnt WeChat App[4] during September 2015. In WeChat, each user can share (receive) posts to (from) his/her friends as like the Twitter and Facebook. Then the advertisers could push their advertisements to users, by merging them into the list of the user's wallposts. For each advertisement, there are two types of feedbacks: "Like" and "Dislike". When the user clicks the "Like" button, his/her friends will receive the advertisements with this action.

The online advertising campaign used in our paper is about LONGCHAMP handbags for young ladies[5]. This campaign contains 14,891 user feedbacks with Like and 93,108 Dislikes. For each user, we have 56 features including (1) demographic attributes, such as age, gender, (2) number of friends, (3) device (iOS or Android), and (4) the user settings on WeChat, for example, whether allowing strangers to see his/her album and whether installing the online payment service.

Experimental Settings. In our experiments, we set the feedback of users on the advertisement as outcome Y. Specifically, we set the outcome $Y_i = 1$ when user i likes the advertisement and $Y_i = 0$ when user i dislikes it. And we alternatively set one of the user features as the treatment T and others as the observed variables $\mathbf{X}$. Therefore, we can estimate the ATT for each user feature. We tuned the parameters in our algorithm and baseline methods with the "approximal ground truth" via cross validation by grid searching.

Evaluation and baselines. In this dataset, we have no ground truth about the treatment effect of each user feature, but we are interesting in whether the top k features ranked by our proposed DCB estimator is able to get good performance in predicting the Like and Dislike behaviors of users, comparing with all above ATT baseline estimators and two commonly used methods for correlation-based feature selection, including MRel (Maximum Relevance) [24] and mRMR (Maximum Relevance Minimum Redundancy) [20]. Our estimator and other ATT baseline estimator rank the user features by their absolute causal effect. We use MAE as the evaluation metric, which is defined as:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|\hat{Y}_i - Y_i|,$$

where m is the number of users in test data, $\hat{Y}_i$ and Y_i represent the predict and actual feedback of user i on the advertisement.

Results. We plot the results in Figure 3. From the results, we can find that our proposed DCB estimator achieves the best prediction accuracy with different number of features. Also, our method can get almost the optimal prediction performance with much less features than other baselines. The main reason is that with differentiating the confounders, our estimator can estimate the causal effect of each user feature more precise by better confounding bias removing. Another important observation is that the two commonly used correlation-based feature selection methods perform worse than our method and even the other causal estimators. This is because of the sample selection bias between the training and testing datasets, the correlation-based methods cannot handle this issue, while the causal estimators can solve the problem to a certain extent by balancing treated and control units and removing the confounding bias.

The results demonstrate that treatment effect estimation can significantly help to improve the prediction performance, as long as the confounding problems are appropriately addressed.

[4]http://www.wechat.com/en/

[5]http://en.longchamp.com/en/womens-bags

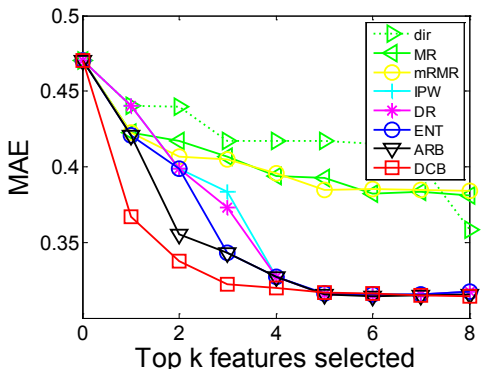


Figure 3: Our proposed DCB estimator outperforms the baselines when selecting the top k significant causal features to predict whether user will like or dislike an advertisement.

6 CONCLUSION

In this paper, we focus on how to estimate the treatment effect more precisely with high dimensional observational data in the wild. We argued that most previous weighting based estimators do not take confounder differentiation into account or require model specification, leading to poor performance in the setting of high dimensional variables or in the wild. Therefore, we proposed the concept of confounder weights for confounders differentiation with theoretical analysis. We proposed a differentiated confounder balancing algorithm to jointly optimize the confounder weights and sample weights for treatment effect estimation. Extensive experiments on both synthetic and real datasets demonstrated that our proposed algorithm can significantly and consistently outperforms the start-of-the-art methods. We also demonstrated that the top ranked features by our algorithm have the best prediction performance on an online advertising dataset.

Our future work will focus on causal inference with unobserved confounders in observational studies by data driven approach.

7 ACKNOWLEDGEMENT

This work was supported by National Program on Key Basic Research Project, No. 2015CB352300; National Natural Science Foundation of China, No. 61370022, No. 61521002, No. 61531006 and No. 61210008. Thanks for the research fund of Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology. Bo Li's research was supported by the Tsinghua University Initiative Scientific Research Grant, No. 20165080091; National Natural Science Foundation of China, No. 71490723 and No. 71432004; Science Foundation of Ministry of Education of China, No. 16JJD630006.

REFERENCES

[1] S. Athey and G. W. Imbens. Machine learning methods for estimating heterogeneous causal effects. *stat*, 1050:5, 2015.

[2] S. Athey, G. W. Imbens, and S. Wager. Approximate residual balancing: De-biased inference of average treatment effects in high dimensions. *arXiv preprint arXiv:1604.07125*, 2016.

[3] P. C. Austin. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate behavioral research*, 46(3):399–424, 2011.

[4] H. Bang and J. M. Robins. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4), 2005.

[5] L. Bottou, J. Peters, J. Q. Candela, D. X. Charles, M. Chickering, E. Portugaly, D. Ray, P. Y. Simard, and E. Snelson. Counterfactual reasoning and learning systems: the example of computational advertising. *Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

[6] D. Chan, R. Ge, O. Gershony, T. Hesterberg, and D. Lambert. Evaluating online ad campaigns in a pipeline: causal models at scale. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 7–16. ACM, 2010.

[7] K. C. G. Chan, S. C. P. Yam, and Z. Zhang. Globally efficient non-parametric inference of average treatment effects by empirical balancing calibration weighting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2015.

[8] V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, et al. Double machine learning for treatment and causal parameters. *arXiv preprint arXiv:1608.00060*, 2016.

[9] A. Diamond and J. S. Sekhon. Genetic matching for estimating causal effects: A general multivariate matching method for achieving balance in observational studies. *Review of Economics and Statistics*, 95(3):932–945, 2013.

[10] M. H. Farrell. Robust inference on average treatment effects with possibly more covariates than observations. *Journal of Econometrics*, 189(1):1–23, 2015.

[11] J. Hainmueller. Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies. *Political Analysis*, 20(1):25–46, 2012.

[12] K. Imai and M. Ratkovic. Covariate balancing propensity score. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):243–263, 2014.

[13] G. W. Imbens and D. B. Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

[14] R. Kohavi and R. Longbotham. Unexpected results in online controlled experiments. *ACM SIGKDD Explorations Newsletter*, 12(2):31–35, 2011.

[15] K. Kuang, P. Cui, B. Li, M. Jiang, S. Yang, and F. Wang. Treatment effect estimation with data-driven variable decomposition. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. AAAI, 2017.

[16] K. Kuang, M. Jiang, P. Cui, and S. Yang. Steering social media promotions with effective strategies. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 2016.

[17] R. J. LaLonde. Evaluating the econometric evaluations of training programs with experimental data. *The American economic review*, pages 604–620, 1986.

[18] D. F. McCaffrey, G. Ridgeway, and A. R. Morral. Propensity score estimation with boosted regression for evaluating causal effects in observational studies. *Psychological methods*, 9(4):403, 2004.

[19] N. Parikh, S. P. Boyd, et al. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[20] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[21] C. A. Rolling and Y. Yang. Model selection for estimating treatment effects. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4):749–769, 2014.

[22] P. R. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[23] R. M. Shiffrin. Drawing causal inference from big data, 2016.

[24] K. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of machine learning research*, 3(Mar):1415–1438, 2003.

[25] D. Westreich, J. Lessler, and M. J. Funk. Propensity score estimation: neural networks, support vector machines, decision trees (cart), and meta-classifiers as alternatives to logistic regression. *Journal of clinical epidemiology*, 63(8):826–833, 2010.

[26] J. R. Zubizarreta. Stable weights that balance covariates for estimation with incomplete outcome data. *Journal of the American Statistical Association*, 110(511):910–922, 2015.