Proceedings of Machine Learning Research vol 140:1–16, 2022 1st Conference on Causal Learning and Reasoning

Differentially Private Estimation of Heterogeneous Causal Effects

Editor: Under Review for CLeaR 2022

Abstract

Estimating heterogeneous treatment effects in domains such as healthcare or social science often involves sensitive data where protecting privacy is important. We introduce a general meta-algorithm for estimating conditional average treatment effects (CATE) with differential privacy guarantees. Our meta-algorithm can work with simple, single-stage CATE estimators such as S-learner and more complex multi-stage estimators such as DR and R-learner. We perform a tight privacy analysis by taking advantage of sample splitting in our meta-algorithm and the parallel composition property of differential privacy. In this paper, we implement our approach using DP-EBMs as the base learner. DP-EBMs are interpretable, high-accuracy models with privacy guarantees, which allow us to directly observe the impact of DP noise on the learned causal model. Our experiments show that multi-stage CATE estimators incur larger accuracy loss than single-stage CATE or ATE estimators and that most of the accuracy loss from differential privacy is due to an increase in variance, not biased estimates of treatment effects.

Keywords: causal inference, differential privacy, interpretability, generalized additive models

1. Introduction

Understanding how treatment effects may vary across a population is critical for policy evaluations and optimizing treatment decisions. Estimating such treatment effect heterogeneity is common in medicine, marketing, and social science. Existing methods, however, may leak information about specific observations used in estimation. We propose a general way to estimate flexible heterogeneous treatment effect models in a privacy-preserving manner and show empirically how privacy guarantees impact estimation accuracy.

Using the potential outcomes framework (Neyman, 1923; Rubin, 1974), we focus on estimating the Conditional Average Treatment Effect (CATE) $\tau(x) = \mathbb{E}[Y(1) - Y(0)|X = x]$ where Y is the outcome of interest, X are observed covariates (features), T is a binary treatment, and $Y(1), Y(0)$ are the potential outcomes under treatment and control. The setting can be experimental, where T is a randomized manipulation, or observational, where T is not explicitly randomized.

Privacy concerns arise when data includes sensitive information, such as healthcare, finance, or advertising data. In these settings, mining population-wide patterns using algorithms without explicit privacy protection can reveal individuals' private information (Carlini et al., 2019; Melis et al., 2019; Shokri et al., 2017). The same concern extends to treatment effect estimation, especially CATE estimation, as the flexibly estimated CATE function at fine granularity could potentially reveal individual covariates, treatment status, or potential outcomes, all of which could be highly sensitive.

Differential privacy (DP) has emerged as a standard mathematical definition of privacy and has been widely adopted by US Census Bureau and companies like Apple and Google for data publication and analysis (Erlingsson et al., 2014; Abowd, 2018). DP provides a robust, cryptographically motivated framework for tracking privacy loss and protects against privacy attacks even when an

© 2022 .

adversary has significant external information (Dwork et al., 2006). In this work, we show how to add DP guarantees to many popular CATE estimation frameworks.

The main contributions of this paper are:

1. We introduce a general meta-algorithm for multi-stage learning under privacy constraints. We apply this to CATE estimation, but it may be independently useful for other multi-stage estimation tasks, especially those using machine learning models to estimate nuisance parameters. Our meta-algorithm can work with simple, single-stage CATE estimators such as S-learner and more complex multi-stage estimators such as DR and R-learner.

2. We perform a tight privacy analysis by taking advantage of sample splitting in our meta-algorithm and the parallel composition property of differential privacy.

3. We study the trade-off between privacy and estimation accuracy using synthetic experiments and a differentially private interpretable ML model, DP-EBM (Nori et al., 2021), as the base learner. Our work clearly shows the change in estimation quality as privacy guarantees and sample size change. We show that flexible multi-stage CATE estimators incur larger accuracy loss than single-stage CATE or ATE estimators and that most of the accuracy loss from differential privacy is due to an increase in variance, not biased estimates of treatment effects.

1.1. Related Work

Our work builds on those from CATE estimation and differential privacy.

The modern CATE literature has focused on adapting flexible machine learning models to estimate CATE while utilizing sample-splitting to avoid biases due to overfitting. This includes methods that specify particular ML methods (Athey and Imbens, 2016; Wager and Athey, 2018)) and "meta-learners" that can utilize general ML algorithms (Nie and Wager, 2020; Künzel et al., 2019; Kennedy, 2020; Chernozhukov et al., 2018; Van der Laan and Rose, 2011). Our paper builds on the latter line of literature and provides a general recipe to make the existing meta-learners differentially private.

Differential private algorithms have been proposed for many prediction tasks, such as classification, regression, and forecasting. These algorithms typically involve adding differential privacy guarantees to existing machine learning algorithms. For example, DP variants exist for linear models (Chaudhuri et al., 2011), random forests (Fletcher and Islam, 2015), and neural networks (Abadi et al., 2016). All of these models can be used as base components in our meta-algorithm.

Finally, there has been some initial work on differentially private causal inference methods. Lee et al. (2019) proposed a privacy-preserving inverse propensity score estimator for estimating average treatment effect (ATE). Their method ties with a specific differentially private prediction method for ATE estimation, which falls into the meta-algorithm framework in this paper. Agarwal and Singh (2021) studied estimation of a finite-dimensional causal parameter under the local differential privacy framework. Our paper uses the central differential privacy framework, which gives weaker privacy protection and retains much higher data utility.

2. Background

2.1. Heterogeneous Treatment Effect Estimation

We assume our data consists of iid observations $Z_i = (Y_i, T_i, X_i)$, where $Y \in \mathbb{R}$ is the outcome, $T \in \{0, 1\}$ is a binary treatment, and $X \in \mathbb{R}^d$ are the observed covariates, and $i \in 1, ..., n$. We define several nuisance functions:

$$e(x) = \mathbb{P}(T = 1|X = x)$$
$$\mu(t, x) = \mathbb{E}(Y|T = t, X = x)$$
$$\eta(x) = \mathbb{E}(Y|X = x)$$

where $e(\cdot)$ is the propensity score, $\mu(\cdot, \cdot)$ is the joint response surface, and $\eta(\cdot)$ is the mean outcome regression. We aim to estimate $\tau(x) = \mu(1, x) - \mu(0, x)$, which is referred to as CATE. We will assume standard causal assumptions such as no unmeasured confounding (conditional on X, treatment is independent of potential outcomes) and overlap of the treated and control propensity score distributions ($0 < e(x) < 1 \ \forall x$). These are trivially satisfied for randomized experiments but typically takes domain knowledge to verity for observational studies.

2.2. Differential Privacy

Differential privacy is a widely adopted privacy definition based on the notion of a randomized algorithm's sensitivity to the presence or absence of any data point (Dwork et al., 2006). It is formally defined as:

Definition 1 ((ε, δ)-Differential Privacy) *A randomized algorithm $\mathcal{A} : \mathbb{Z}^n \to \mathcal{H}$ is (ε ,δ)-differentially private if for all neighboring datasets S, $S' \in \mathbb{Z}^n$, and for all $\mathcal{O} \subset \mathcal{H}$:*

$$\mathbb{P}(\mathcal{A}(S) \in \mathcal{O}) \leq e^{\varepsilon}\mathbb{P}(\mathcal{A}(S') \in \mathcal{O}) + \delta,$$

where neighboring datasets are defined as two datasets that differ in at most one observation.

Intuitively, differential privacy is a promise to each user that the output of any DP algorithm will be approximately the same with or without their data. The privacy parameters ε and δ parameterize the strength of the privacy guarantee – smaller values of ε and δ ensure that the output distributions of $\mathcal{A}(S)$ and $\mathcal{A}(S')$ are closer, thereby ensuring more privacy. Algorithms typically achieve differential privacy guarantees by adding calibrated, symmetric noise to their outputs. Decreasing ε or δ necessitates adding more noise to a model, which leads to an implicit trade-off between privacy and accuracy. We explore this trade-off for the CATE setting in Section 5.

We also leverage a refined notion of differential privacy called f-DP, which provides better mathematical tools for the privacy analysis of our algorithms (Dong et al., 2019). f-DP conceptualizes the privacy task as providing minimum error rates on statistical tests that would try to determine which of two potential samples were used to generate a statistic. It therefore parameterizes the privacy guarantees with a function, called a "trade-off function," that relates how Type-I and Type-II error can be traded-off in such a test. (ε, δ)-DP is a special case of the f-DP framework. Our main theorem will be presented in terms of the more general f-DP space. Our experimental results will be shown in the more widely used (ε, δ)-DP space. The following definitions are due to Dong et al. (2019), which we copy here for completeness.

Definition 2 (f-Differential Privacy) *Let f be a trade-off function. A randomized algorithm $\mathcal{A}$ is said to be f-differentially private if for all neighboring datasets S and S',*

$$T\big(\mathcal{A}(S), \mathcal{A}(S')\big) \geq f.$$

Definition 3 (Trade-off Functions) *Let P and Q be any two probability distributions on the same space. Let ϕ be any (possibly randomized) rejection rule for testing $H_0 : P$ against $H_1 : Q$. Define the trade-off function $T(P, Q) : [0, 1] \to [0, 1]$ as*

$$\alpha \mapsto \inf_{\phi}\{1 - \mathbb{E}_Q[\phi] : \mathbb{E}_P[\phi] \leq \alpha\},$$

where the infimum is taken over all rejection rules.

3. Privacy-preserving CATE Learners

In this section, we present a meta-algorithm of conditional treatment effect estimation with simple and tight privacy guarantees. The meta-algorithm can be used with leading CATE learners in the literature, including DR-learner, R-learner, and S-learner, among others. Multiple sample splitting, i.e., using different parts of the sample to estimate different components of an estimator, is the key feature of this algorithm and enables the application of the parallel composition property of differential privacy. The privacy guarantee we give is learner-model-agnostic, i.e., it applies for any meta-learner in the literature and sub-algorithm modules as long as each sub-algorithm module is differentially private on its own.

3.1. Construction

The algorithm below describes our proposed meta-algorithm.

Algorithm 1 A two-stage algorithm with sample splitting and data transformation

input Dataset $\mathcal{S} = \{Z_i\}_{i\in[n]} = \{(X_i, T_i, Y_i)\}_{i\in[n]}$, sample splitting ratio $\lambda_{1,1}, \ldots, \lambda_{1,k}$ and λ_2 with $\sum_{i=1}^{k} \lambda_{1,i} + \lambda_2 = 1$

output $\mathcal{M}(\mathcal{S}) = (\mathcal{A}_1(S_{1,1}), \ldots, \mathcal{A}_k(S_{1,k}), \mathcal{B} \circ \mathcal{T}(S_2; \mathcal{A}_1(S_{1,1}), \mathcal{A}_2(S_{1,2}), \ldots, \mathcal{A}_k(S_{1,k})))$, where $\mathcal{M}$ denotes the full algorithm, $\mathcal{A}_i, i \in [k]$ and $\mathcal{B}$ denote algorithm modules, $\mathcal{T}$ is a data transformation operator, $S_{1,1}, \ldots, S_{1,k}$ and S_2 form a partition of the full dataset with sample ratio $\lambda_{1,1}, \ldots, \lambda_{1,k}$ and λ_2

1: Sample splitting: let $(S_{1,1}, \ldots, S_{1,k}, S_2) = \texttt{Partition}(\mathcal{S}; \lambda_{1,1}, \ldots, \lambda_{1,k}, \lambda_2)$ be a partition of $\mathcal{S}$ that is chosen uniformly at random among all the partitions of $\mathcal{S}$ with sizes $n_{1,1} = \lambda_{1,1} \cdot n, \ldots, n_{1,k} = \lambda_{1,k} \cdot n$ and $n_2 = \lambda_2 \cdot n$

2: Run first-stage algorithms on disjoint sets of samples: $\mathcal{A}_1(S_{1,1}), \ldots, \mathcal{A}_k(S_{1,k})$

3: Construct data with transformed outcome, $\tilde{S}_2$, using S_2 and the output of first-stage algorithms: $\tilde{S}_2 = \{(X_i, \hat{\psi}_i)\}_{i\in S_2} = \mathcal{T}(S_2)$, in which $\hat{\psi}_i = \varphi(Z_i; \mathcal{A}_1(S_{1,1}), \mathcal{A}_2(S_{1,2}), \ldots, \mathcal{A}_k(S_{1,k}))$

4: Run the second-stage algorithm with the transformed outcome: $\mathcal{B}\left(\tilde{S}_2\right)$

The meta-algorithm above is expressive enough to represent many leading CATE learners. We next show how to map the DR-learner, R-learner, and S-learner to this framework by specifying

their $k, \mathcal{A}_1, \ldots, \mathcal{A}_k, \mathcal{T}, \mathcal{B}$. Use $L(Y \sim X)$ to denote a regression estimator, which estimates $x \to \mathbb{E}[Y|X = x]$.

- DR-learner from Kennedy (2020) uses a two-stage doubly robust learning approach. This learner divides the data into three samples $S_{1,1}, S_{1,2}, S_2$, so $k = 2$. The first stage estimates two nuisance functions. $\mathcal{A}_1$ uses $S_{1,1}$ to estimate the propensity score $\hat{e} = L_{1,1}(T \sim X)$. $\mathcal{A}_2$ uses $S_{1,2}$ to estimate the joint response surface $\hat{\mu} \sim L_{1,2}(Y \sim (T, X))$. The data transformation step uses $\hat{e}$ and $\hat{\mu}$ to construct the doubly robust score, $\hat{\psi}(Z) = \hat{\mu}(1, X) - \hat{\mu}(0, X) + \frac{Y - \hat{\mu}(1,X)}{\hat{e}(X)} \mathbb{1}(T = 1) - \frac{Y - \hat{\mu}(0,X)}{1 - \hat{e}(X)} \mathbb{1}(T = 0)$. The second stage, $\mathcal{B}$, uses the transformed data generated from S_2 to run the pseudo-outcome regression $\hat{\tau} = L_2(\hat{\psi} \sim X)$.

- R-learner from Nie and Wager (2020) uses a two-stage debiased machine learning approach. This learner divides the data into three samples $S_{1,1}, S_{1,2}, S_2$, so $k = 2$. The first stage estimates two nuisance functions. $\mathcal{A}_1$ uses $S_{1,1}$ to estimate the propensity score $\hat{e} = L_{1,1}(T \sim X)$. $\mathcal{A}_2$ uses $S_{1,2}$ to estimate the mean outcome regression $\hat{\eta} = L_{1,2}(Y \sim X)$. The data transformation step produces residuals from two regressions $\hat{\psi}(Z) = (Y - \hat{\eta}(X), T - \hat{e}(X))$. The second stage, $\mathcal{B}$, uses the transformed data generated from S_2 to run an empirical loss minimization $\hat{\tau} = \arg\min_\tau \sum_{i \in S_2} \left[\{Y_i - \hat{\eta}(X_i)\} - \{T_i - \hat{e}(X_i)\} \tau(X_i)\right]^2$.

- S-learner from Künzel et al. (2019) uses the whole dataset $S_2 = \mathcal{S}$ to estimate the joint response surface $\hat{\mu} \sim L_2(Y \sim (T, X))$ and then estimates $\hat{\tau}(x) = \hat{\mu}(1, x) - \hat{\mu}(0, x)$. In this case $k = 0$ and $S_2 = \mathcal{S}$.

Multiple sample splitting is a key design choice of Algorithm 1. The practice of using separate samples for estimating the nuisance functions in the first stage and the CATE function in the second stage is popular in the literature (Chernozhukov et al., 2018; Kennedy, 2020; Newey and Robins, 2018; Nie and Wager, 2020; Van der Laan and Rose, 2011). This is motivated by the fact that sample splitting avoids the bias due to overfitting. Kennedy (2020) and Newey and Robins (2018) point out that using different parts of the sample to estimate different components of an estimator, called double sample splitting, leads to simpler and sharper risk bound analysis. On top of these, we will see in Section 3.2 that multiple sample splitting leads to tight differential privacy guarantees by exploiting parallel composition and post-processing properties of differential privacy.

Remark 4 Algorithm 1 releases all outputs of intermediates algorithm modules. For example, in the DR-learner case, it releases the estimated propensity score function, estimated regression function, and the final estimated CATE function to the public. From a practical point of view, releasing these intermediate outputs enables researchers to understand the data better and perform diagnostic tests. From a theoretical point of view, it is simpler to derive DP guarantees for compositions when intermediate outputs are released. Moreover, we cannot save any privacy budget by keeping the intermediate output private given the generality of our meta-algorithm. In this sense, it is privacy-budget-free to release all intermediates outputs.

Remark 5 Many classical semiparametric estimators and popular approaches for applying machine learning methods to causal inference follow a multi-stage estimation scheme as Algorithm 1. Our Algorithm can be used to construct sample-split versions of these estimators. Notable examples include augmented inverse-propensity weighting (AIPW) estimators for average treatment effect

(Robins et al., 1994), debiased machine learning for for treatment and structural parameters (Chernozhukov et al., 2018; Foster and Syrgkanis, 2019), and estimating nonparametric instrumental variable models (Newey and Powell, 2003; Hartford et al., 2017) among others. Though the focus of this paper is specifically on conditional treatment effect estimation, our privacy guarantee applies to sample-split versions of all of these estimation algorithms.

3.2. Privacy Guarantees

In this section we formally prove Algorithm 1 is private under the conditions that each of the sub algorithm modules is private. To give a tight analysis of the privacy guarantee, we adopt the f-DP framework proposed by Dong et al. (2019).

Theorem 6 *Suppose*

1. *algorithm module $\mathcal{A}_i : \mathbb{Z}^{n_{1,i}} \to \mathcal{H}_{1,i}$ is $f_{1,i}$-DP for $i = 1, \ldots, k$, where $\mathcal{H}_{1,i}$ is the image space of $\mathcal{A}_i$,*
2. *algorithm module $\mathcal{B} : (\mathbb{X} \times \Psi)^{n_2} \to \mathcal{H}_2$ is f_2-DP, where Ψ is the image space of the data transformation $\varphi : \mathbb{Z} \times \mathcal{H}_1 \times \ldots \times \mathcal{H}_k \to \Psi$,*

then the composed meta-algorithm $\mathcal{M}$ is f-DP with

$$f = \min\{f_{1,1}, \ldots, f_{1,k}, f_2\}^{**},$$

where $g^*(y) := \sup_{x \in \mathbb{R}} yx - g(x)$ denotes the convex conjugate of a generic function g and $g^{**} = (g^*)^*$ denotes the double conjugate.

A proof of theorem 6 can be found in the appendix.

Remark 7 When all algorithm modules are (ε, δ)-DP with the same (ε, δ), Theorem 6 suggests the whole algorithm is also (ε, δ)-DP.

4. DP-EBM CATE learner

In this section we introduce DP-EBMs as the base learner in the meta-algorithm and present the full algorithm of a specific differentially private CATE leaner we call DP-EBM-DR-learner.

4.1. DP-EBM

DP-EBM is a differentially private and interpretable machine learning algorithm recently introduced by Nori et al. (2021), which adds differential privacy guarantees to Explainable Boosting Machines (EBM) (Nori et al., 2019; Lou et al., 2012, 2013). EBMs and DP-EBMs belong to the family of Generalized Additive Models (GAMs), which are restricted models of the form:

$$g(\mu(x)) = \alpha + f_1(x_1) + ... + f_d(x_d),$$

where each $\mu(x) := \mathbb{E}[Y|X]$ is the regression function, f_i is a univariate function that operates on each input feature x_i, α is an intercept, and g is a link function which adapts the model to different settings like classification and regression. GAMs are a relaxation of generalized linear models in which each function f_i is restricted to linear. In the GAM setting, the models are allowed

to flexibly learn complex functions of each feature, which has been shown to yield more expressive and accurate models (Chang et al., 2021; Caruana et al., 2015; Nori et al., 2019). Moreover, the additive structure of the models and inability to learn complex interactions between features (e.g.: $f(x_1, x_2, x_3)$) allows GAMs to remain interpretable. At prediction time, the contribution of each feature i is exactly $f_i(x_i)$. These term contributions can be directly compared, sorted, and reasoned about. In addition, each function f_i can be visualized as a graph to show how the model's predictions change with varying inputs.

DP-EBMs add privacy guarantees to EBMs by adding calibrated Gaussian noise during each iteration of the training process. This noise has been shown to cause distortion in the individual feature functions f_i, especially under strong privacy guarantees. We use DP-EBMs to highlight our DP CATE learning algorithm for two main reasons: DP-EBMs have been shown to yield high accuracy under strong privacy guarantees on tabular datasets, and the interpretability of DP-EBMs is useful to highlight how varying privacy guarantees affect CATE estimation (Nori et al., 2021).

4.2. DP-EBM-DR-learner

Previously, Algorithm 1 outlined a general recipe to construct DP CATE learners. Now that we have introduced DP-EBM, a concrete prediction algorithm with privacy guarantee, we will present a complete CATE estimator that we call DP-EBM-DR-learner.

Algorithm 2 DP-EBM-DR-learner

input Dataset $\mathcal{S} = \{Z_i\}_{i\in[n]} = \{(Y_i, T_i, X_i)\}_{i\in[n]}$, privacy parameters $\varepsilon \in (0, \infty)$, $\delta \in (0, 1)$, sample splitting ratio $\lambda_{1,1}, \ldots, \lambda_{1,k}$ and λ_2 with $\sum_{i=1}^{k} \lambda_{1,i} + \lambda_2 = 1$,

output $\mathcal{M}(\mathcal{S}) = (\hat{e}, \hat{\mu}, \hat{\tau})$, where $\mathcal{M}$ denotes the full algorithm

1: Sample splitting: let $(S_{1,1}, S_{1,2}, S_2) = \texttt{Partition}(\mathcal{S}; \lambda_{1,1}, \lambda_{1,2}, \lambda_2)$ be a partition of $\mathcal{S}$ that is chosen uniformly at random among all the partitions of $\mathcal{S}$ with sizes $n_{1,1} = \lambda_{1,1} \cdot n, n_{1,2} = \lambda_{1,2} \cdot n$ and $n_2 = \lambda_2 \cdot n$

2: Run first-stage algorithms on disjoint sets of samples:

   (a) Use $S_{1,1}$ and DP-EBM classifier to estimate propensity scores $\hat{e} = \text{DP-EBM}(T \sim X; S_{1,1}, \varepsilon, \delta)$

   (b) Use $S_{1,2}$ and DP-EBM regression to estimate joint response surface $\hat{\mu} = \text{DP-EBM}(Y \sim (T, X); S_{1,2}, \varepsilon, \delta)$

3: Construct data with transformed outcome, $\tilde{S}_2$, using S_2 and the output of first-stage algorithms: $\tilde{S}_2 = \{(X_i, \hat{\psi}_i)\}_{i\in S_2}$, in which $\hat{\psi}_i = \hat{\mu}(1, X_i) - \hat{\mu}(0, X_i) + \frac{Y_i - \hat{\mu}(1, X_i)}{\hat{e}(X_i)} \mathbb{1}(T_i = 1) - \frac{Y_i - \hat{\mu}(0, X_i)}{1 - \hat{e}(X_i)} \mathbb{1}(T_i = 0)$

4: Use the transformed data $\tilde{S}_2$ and DP-EBM regression to estimate CATE $\hat{\tau} = \text{DP-EBM}(\hat{\psi} \sim X; \tilde{S}_2, \varepsilon, \delta)$

DP-EBM-DR-learner is (ε, δ)-DP by theorem 6 and remark 7.

Similarly, we can construct DP-EBM-R-learner and DP-EBM-S-learner following Section 3.1 and use DP-EBM for fitting the regression model (or empirical loss minimization).

Our main focus here is DP-EBM-DR-learner due to the favorable theoretical properties of DR-learner, including double robustness (Kennedy, 2020), connection to semiparametric efficiency (Robins et al., 1994), and its interpretability. More precisely, DR-learner will always estimate the projection of the true CATE function onto the space of functions over which we optimize in the final regression, even if the true CATE function does not lie in this function space. This fact partially mitigates the problem of model misspecification. We pick DP-EBM prediction algorithm due to its exact model interpretability and high accuracy under privacy constraints.

R-learner shares many of the theoretical properties of DR-learner. DP-EBM-R-learner also has similar empirical performance as DP-EBM-DR-learner in the following experiments.

We treat S-learner as the baseline due to its simplicity. Since the approach is relatively simple and does not use estimated propensity scores, S-learner often has larger bias (integrated squared bias) and smaller variance (integrated variance) for moderate sample sizes. DP-EBM-S-learner reduces to a differentially private average treatment effect (ATE) estimator. This happens because the functional form restriction of DP-EBM enforces the CATE estimate of S-learner to be a constant $\hat{\tau}(x) = \hat{\mu}(1, x) - \hat{\mu}(0, x) = \hat{f}_T(1) - \hat{f}_T(0)$, where $\hat{f}_T$ is the estimated univariate function of the treatment. As a result, DP-EBM-S-learner, an ATE estimator, has a large irreducible bias and a small variance for estimating CATE.

5. Experiments

In this section, we present experimental results for DP CATE learners using data from a real-world experiment with synthetic treatment effect in addition to five simulated datasets. Imposing privacy constraints necessarily compromises accuracy in estimating CATE. Understanding this privacy-accuracy trade-off is the main goal of our experiments.

We present our analysis to illustrate how DP noise affects the estimated CATE function. We show how different DP CATE learners applied to different sample sizes are affected differently by varying the level of privacy constraints. Further, we elaborate on the privacy-accuracy trade-off by decomposing the increase in MSE into increases in bias and variance.

5.1. Setups

5.1.1. ALGORITHM SPECIFICATION

In our experiments, we evaluate three DP CATE learners: DP-EBM-DR-learner, DP-EBM-R-learner, and DP-EBM-S-learner. For the first two, the sample splitting ratio is set to $(\lambda_{1,1}, \lambda_{1,2}, \lambda_2) = (0.25, 0.25, 0.5)$, which means estimating propensity score, outcome regression, and CATE function each uses a quarter, a quarter, and a half of the whole sample, respectively. We use the DP-EBM classifier and regressor implemented in the python package `interpret` introduced in Nori et al. (2019) with all hyperparameters set to their default values. Except for propensity scores, which are estimated using a DP-EBM classifier, all other functions in the experiments are fitted using a DP-EBM regressor.

5.1.2. DATA DESCRIPTION

We describe how we generate different datasets $\{(Y_i, T_i, X_i)_{i \in \mathcal{S}}\}$ in all experiments using the following language. We use P_X to denote the covariate distribution, $e(x) := P(T = 1|X = x)$ to denote the propensity score, and $b(x) := \mathbb{E}[Y(0)|X = x]$ for the control baseline function.

The main dataset we consider is a real dataset from Arceneaux et al. (2006), which studies the effect of paid get-out-the-vote calls on voter turnout. We refer to this dataset as the voting data. This data is originally generated from a stratified experiment with varying treatment probability and has previously been used for comparing and evaluating the performance of different causal effect estimators (Arceneaux et al., 2006; Nie and Wager, 2020). Nie and Wager (2020) points out that the true treatment effect in this data is close to non-existent, so we can spike the original dataset with a synthetic treatment effect $\tau(\cdot)$ to make the task of estimating heterogeneous treatment effects non-trivial. The modified dataset therefore has the covariate distribution and propensity function from the true data and has a synthetic treatment effect. Because we know the true synthetic treatment effect $\tau(\cdot)$, we can compare the estimated $\hat{\tau}(\cdot)$ to it and evaluate its performance.

Following Nie and Wager (2020), we focus on a subset of 148,160 samples containing 59,264 treated units and 88,896 controls. The covariates have 11 variables, which include state, county, age, gender, past voting record, etc. The synthetic treatment effect is set to $\tau(X_i) = -\frac{\text{VOTE00}_i}{2+100/\text{AGE}_i}$, where VOTE00i indicates whether the i-th unit voted in the year 2000. The synthetic treatment effect is added by strategically flipping the binary outcome. Denote the original outcome in the data by Y^*. With probability $1 - \tau(X_i)$, we set $Y_i(0) = Y_i(1) = Y^*$. With probability $\tau(X_i)$, we set $Y_i(0) = 1, Y_i(1) = 0$. Finally, we set $Y_i = Y_i(T_i)$. The treatment heterogeneity as measured by $\text{Var}(\tau(X)) = 0.016$ is moderate. This number can also be interpreted as the MSE of using the true ATE as a CATE estimator.

All five other datasets are generated by simulation based on the same recipe:

$$X_i \sim P_X,\ T_i|X_i \sim \text{Ber}(e(X_i)), \epsilon_i \sim \text{N}(0,1)$$
$$Y_i = b(X_i) + T_i \cdot \tau(X_i) + \epsilon_i.$$

We specify setups A to E with different (P_X, b, e, τ) to cover a variety of settings.

Setup	P_X	b	e	τ
A	$\text{U}(0,1)^d$	$\sin(\pi x_1 x_2) + 2(x_3 - 0.5)^2$ $+x_4 + 0.5x_5$	$\text{trim}_{0.1}\{\sin(\pi x_1 x_2)\}$	$(x_1 + x_2)/2$
B	$\text{N}(0, I_d)$	$\max\{x_1 + x_2, x_3, 0\}$ $+\max\{x_4 + x_5, 0\}$	0.5	$x_1 + \log(1 + e^{x_2})$
C	$\text{N}(0, I_d)$	$2\log(1 + e^{x_1+x_2+x_3})$	$1/(1 + e^{x_2+x_3})$	1
D	$\text{N}(0, I_d)$	$\max\{x_1 + x_2 + x_3, 0\}$ $+\max\{x_4 + x_5, 0\}$	$1/(1 + e^{-x_1} + e^{-x_2})$	$\max\{x_1 + x_2 + x_3, 0\}$ $-\max\{x_4 + x_5, 0\}$
E	$\text{N}(0, \Sigma_d)$	$\sum_{i=1}^{6} i x_i + x_1 x_6$ $+\mathbb{1}(-0.5 < x_3 < 0.5)$	$1/(1 + e^{x_1+x_6})$	$1/(1 + e^{x_1}) - x_2$ $+\sum_{i=3}^{6} x_i$

Table 1: Experiment Setup

For all setups, $d = 6$. The trim function is specified as $\text{trim}_{0.1}(x) = \min\{\max\{0.1, x\}, 0.9\}$. Setup A, B, C, and D are from Section 6 of Nie and Wager (2020). Setup A has a complicated baseline function. Setup B is a randomized controlled trial with $e = 0.5$. Setup C has a constant treatment effect with $\tau = 1$. Setup D has a nondifferentiable treatment effect function. Setup E has correlated covariates with Σ_d generated from a random data generator. Its baseline function is discontinuous and contains an interaction term between variables.

5.1.3. EXPERIMENT HYPERPARAMETERS

We run all three DP CATE learners on all six datasets with seven levels of training sample size growing exponentially $\{500, 1000, 2000, 4000, 8000, 16000, 32000\}$. For setups A-E, we generate a fixed test sample of size 250000 using their data generating process and use it throughout the experiments. For the voting data, we construct the training data by random sampling (stratified by treatment) from the whole data and use all data left as the test set. The privacy parameter δ is fixed as 10^{-5} and ε varies between $\{1, 2, 4, 8, 16\}$. $\varepsilon = 16$ is seen as approximately non-private since the privacy guarantee is extremely weak. $\varepsilon = 1$ and $\varepsilon = 2$ are seen as strong guarantees.

For each specification, we train the algorithm twice on two separate training sets with the specified sample size. We then generate three empirical MSE corresponding to the predicted CATE using these two trained algorithms and the average of these on the test set. These three MSE are denoted as $\widehat{\text{MSE}}_1, \widehat{\text{MSE}}_2, \widehat{\text{MSE}}_{avg}$. We generate estimated $\widehat{\text{MSE}} = (\widehat{\text{MSE}}_1 + \widehat{\text{MSE}}_2)/2$, $\widehat{\text{Bias}} = 2\widehat{\text{MSE}}_{avg} - \widehat{\text{MSE}}$, $\widehat{\text{Var}} = 2\widehat{\text{MSE}} - \widehat{\text{Bias}}$. Here $\widehat{\text{Bias}}$ is meant to estimate the integrated squared bias $\int (\mathbb{E}_{\mathcal{S}}[\hat{\tau}_{\mathcal{S}}(x)] - \tau(x))^2 dF(x)$, in which $\mathbb{E}_{\mathcal{S}}[\hat{\tau}_{\mathcal{S}}(x)]$ denotes the expected estimate at point x over the distribution of data $\mathcal{S}$. $\widehat{\text{Var}}$ is meant to estimate the integrated variance $\int \text{Var}(\hat{\tau}_{\mathcal{S}}(x)) dF(x)$. This calculation is motivated by the fact that MSE of the average prediction is the sum of the integrated squared bias plus half of the integrated variance. For each of the specification we run this process five times and report the average $\widehat{\text{MSE}}$, $\widehat{\text{Bias}}$, $\widehat{\text{Var}}$.

5.2. Insights from Experiment Results

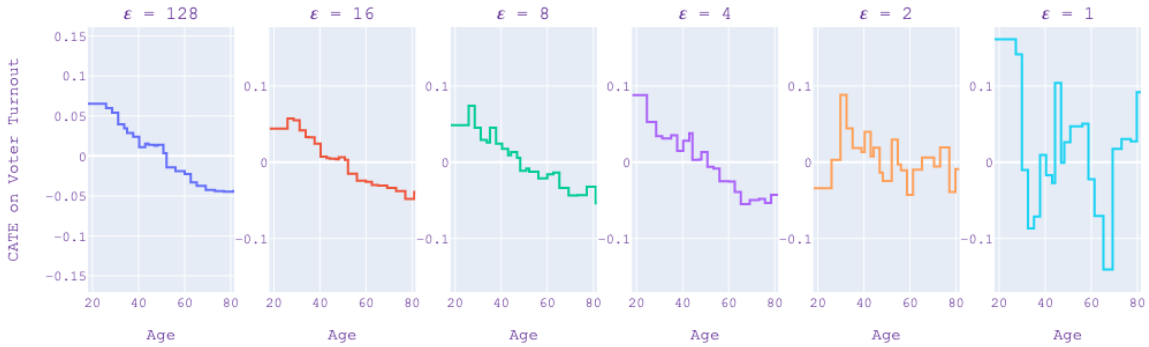

Figure 1: Estimate of CATE on voter turnout as a function of Age (one of the 11 features) using voting data with sample size 16000 and DP-EBM-DR-learner with varying ε.

Figure 1 shows the shape function of the conditioning variable "Age" that the DP-EBM-DR-learner learned from the voting data with sample size 16000 at five different levels of ε. The main purpose of this figure is to transparently show the effect of adding differential privacy constraints on the estimated CATE function. Since the true shape function corresponds to the projection of the synthetic CATE function $\tau^*(X_i) = -\frac{\text{VOTE00}_i}{2+(100/\text{AGE}_i)}$ onto the additive separable function space, it is smooth and monotonically decreasing. DP-EBM-DR-learner learns this well when the algorithm is nonprivate or the privacy is low with $\varepsilon = 16$ or 8. When privacy gets stronger with $\varepsilon = 4$ or 2 or 1, the estimated shape functions become jumpier and less smooth. Intuitively, this happens because differential privacy is achieved by adding Gaussian noise in the training process. Higher

levels of noise under strong privacy lead to less well-behaved nonparametric function estimates. Similar behavior has been observed when DP-EBMs are applied to general prediction tasks (Nori et al., 2021).

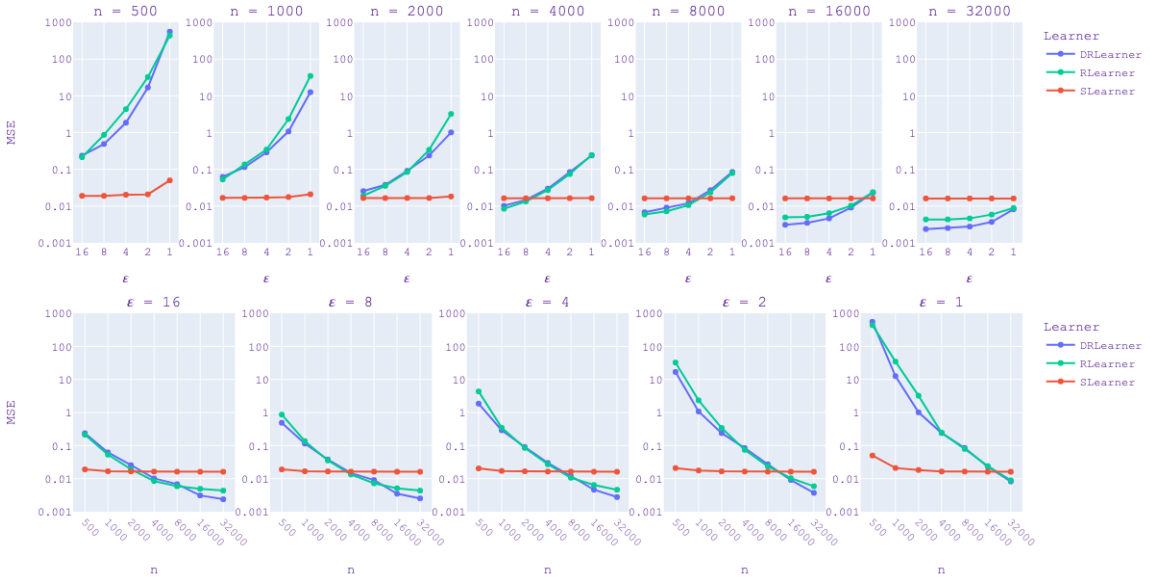

Figure 2: Comparison of DP-EBM-DR-learner, DP-EBM-R-learner, and DP-EBM-S-learner on voting data with varying DP guarantee ε and sample size n. All MSE values are averages of 25 runs reported on an independent test set.

Figure 2 plots the MSE of DP-EBM-DR-learner, DP-EBM-R-learner, and DP-EBM-S-learner under seven different sample sizes of the voting data at five different levels of privacy. Each curve in the upper row shows how MSE increases as the privacy gets stronger at a given sample size. Each curve in the bottom row shows how MSE decreases as the sample size increases at a given privacy level.

The performance of the less flexible DP-EBM-S-learner stays remarkably stable with MSE equal to about 0.016 for most designs and is larger than this only at only $n = 500$ and $\varepsilon = 1$ where the sample size is the smallest and the privacy is the strongest. This happens because DP-EBM-S-learner outputs a constant scalar estimate of ATE, which is easily estimable with hundreds of samples despite the privacy constraint. In fact, its MSE approximately equals $\mathrm{Var}(\tau^*(X)) = 0.016$, which is the MSE of the true ATE.

The more flexible DP-EBM-DR-learner and DP-EBM-R-learner are much more sensitive to the privacy parameter ε. This is especially true when the sample size n is small. MSE increases by a factor of 10^3 going from $\varepsilon = 16$ to $\varepsilon = 1$ at $n = 500$; while it only increases by a factor less than 10 from $\varepsilon = 16$ to $\varepsilon = 1$ at $n = 32000$. For the intermediate sample size $n = 16000$, DR-learner and R-learner are better until privacy is at its strongest at $\varepsilon = 1$, as also suggested by Figure 1.

A researcher interested in exploiting heterogeneity in treatment effects under privacy constraints will need to decide whether to use a flexible CATE estimator or a simple ATE estimator. The lower plot shows that when there is almost no privacy at $\varepsilon = 16$, flexible CATE estimators perform better with 3000 or more samples on this dataset. As the privacy requirements get stronger, it takes more

samples for the flexible CATE estimator to perform better. At $\varepsilon = 1$ the simple ATE estimator outperforms the CATE estimators until about 20000 samples on this dataset.

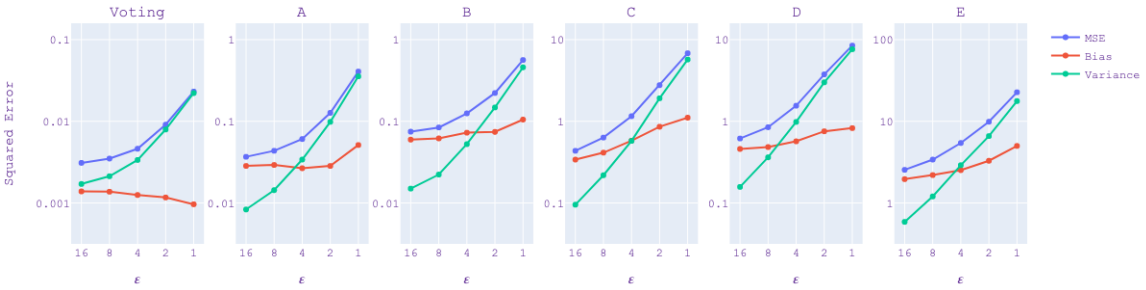

Figure 3: Performance of DP-EBM-DR-learner on six datasets. Datasets Voting, A, and B use 16000 samples, and C, D, and E use 4000 samples. All mean squared error (MSE), bias, and variance values are averages of 25 runs from an independent test set.

Figure 3 shows the mean squared error, integrated squared bias, and integrated variance values of the DP-EBM-DR learner on all six datasets. Moving from no privacy to strong privacy, the variance of this learner often increases by about a factor of 10 while the bias changes by at most a factor of 2. This bias-variance decomposition reveals that accuracy loss due to privacy constraints comes mostly from the increase in variance. In other words, the privacy-variance trade-off captures most of the privacy-accuracy trade-off.

Our main experimental findings can be summarized as:

1. Stronger privacy constraints lead to noisier and jumpier estimated CATE functions.
2. Most of the additional accuracy loss due to privacy can be attributed to a significant increase in variance, while bias often increases at a much slower rate.
3. Flexible, multi-stage CATE learners incur larger accuracy loss from differential privacy than less flexible, single-stage CATE or ATE learners. It takes a significantly larger sample size to exploit treatment effect heterogeneity under strong privacy constraints.

6. Discussions and Conclusions

We presented a meta-algorithm for differentially private estimation of the Conditional Average Treatment Effect (CATE). The method works with a variety of CATE estimation methods and provides tight privacy bounds. Experiments with interpretable DP-EBMs as the base learners show the impact of DP noise on the learned causal models. Significantly, experiments that vary the size of the training data and the strength of the privacy guarantee indicate that most of the accuracy loss in the high-privacy, low-to-modest sample size regime is due to an increase in variance, not due to model bias. This suggests that when data is large relative to the noise injected by DP, it is beneficial to use more complex multi-stage CATE estimation procedures, but when the sample size is not large enough to overcome DP noise, simpler estimation procedures with potentially larger bias and lower variance may be preferred.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

John M Abowd. The us census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2867–2867, 2018.

Anish Agarwal and Rahul Singh. Causal inference with corrupted data: Measurement error, missing values, discretization, and differential privacy. *arXiv preprint arXiv:2107.02780*, 2021.

Kevin Arceneaux, Alan S. Gerber, and Donald P. Green. Comparing experimental and matching methods using a large-scale voter mobilization experiment. *Political Analysis*, 14(1):37—62, 2006.

Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. *USENIX Security*, 2019.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1721–1730, 2015.

Chun-Hao Chang, Sarah Tan, Ben Lengerich, Anna Goldenberg, and Rich Caruana. How interpretable and trustworthy are GAMs? In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 95–105, 2021.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.

Sam Fletcher and Md Zahidul Islam. A differentially private decision forest. *AusDM*, 15:99–108, 2015.

Dylan J Foster and Vasilis Syrgkanis. Orthogonal statistical learning. *arXiv preprint arXiv:1901.09036*, 2019.

Jason Hartford, Greg Lewis, Kevin Leyton-Brown, and Matt Taddy. Deep IV: A flexible approach for counterfactual prediction. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1414–1423. PMLR, 2017.

Edward H Kennedy. Optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*, 2020.

Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10):4156–4165, 2019.

Si Kai Lee, Luigi Gresele, Mijung Park, and Krikamol Muandet. Privacy-preserving causal inference via inverse probability weighting. *arXiv preprint arXiv:1905.12592*, 2019.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158, 2012.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631, 2013.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019.

Whitney K. Newey and James L. Powell. Instrumental variable estimation of nonparametric models. *Econometrica*, 71(5):1565–1578, 2003.

Whitney K Newey and James R Robins. Cross-fitting and fast remainder rates for semiparametric estimation. *arXiv preprint arXiv:1801.09138*, 2018.

Jerzy S Neyman. On the application of probability theory to agricultural experiments. essay on principles. section 9. *Roczniki Nauk Rolniczych Tom X[In Polish]*, 10:1–51, 1923.

X Nie and S Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108 (2):299–319, 2020.

Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

Harsha Nori, Rich Caruana, Zhiqi Bu, Judy Hanwen Shen, and Janardhan Kulkarni. Accuracy, interpretability, and differential privacy via explainable boosting. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8227–8237. PMLR, 2021.

James M. Robins, Andrea Rotnitzky, and Lue Ping Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, 89(427):846–866, 1994.

Donald B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

Josh Smith, Hassan Jameel Asghar, Gianpaolo Gioiosa, Sirine Mrabet, Serge Gaspers, and Paul Tyler. Making the most of parallel composition in differential privacy. *arXiv preprint arXiv:2109.09078*, 2021.

Mark J Van der Laan and Sherri Rose. *Targeted learning: Causal inference for observational and experimental data*. Springer Science & Business Media, 2011.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

Appendix: Proof of Theorem 6

We begin our proof by introducing a parallel composition theorem of f-DP recently introduced in Smith et al. (2021). This theorem gives a cumulative privacy guarantee on a sequence of analysis on a private dataset, in which each step is done using a new set of samples and is informed by previous steps only through accessing intermediate outputs.

Let $M_1 : \mathbb{Z}^{n_1} \to \mathcal{H}_1$ be the first algorithm and $M_2 : \mathbb{Z}^{n_2} \times \mathcal{H}_1 \to \mathcal{H}_2$ be the second algorithm. $\mathcal{H}_i$ denotes the image space of M_i. The joint algorithm $M : \mathbb{Z}^{n_1+n_2} \to \mathcal{H}_1 \times \mathcal{H}_2$ is defined as

$$M(S_1, S_2) = (h_1, M_2(S_2, h_1)),$$

where $S_1 \in \mathbb{Z}^{n_1}$ and $S_2 \in \mathbb{Z}^{n_2}$ are disjoint datasets and $h_1 = M_1(S_1)$. We can follow this recipe to define an l-fold composed algorithm with disjoint inputs iteratively and get the following privacy guarantee.

Lemma 8 *Let $M_i(\cdot; h_1, \ldots, h_{i-1}) : \mathbb{Z}^{n_i} \to \mathcal{H}_i$ be f_i-DP for all $h_1 \in \mathcal{H}_1, \ldots, h_{i-1} \in \mathcal{H}_{i-1}$. The l-fold composed algorithm with disjoint inputs $M : \mathbb{Z}^n \to \mathcal{H}_1 \times \cdots \times \mathcal{H}_l$ is f-DP, where $n = n_1 + \ldots + n_l$ and $f = \min\{f_1, \ldots, f_l\}^{**}$.*

Intuitively, this theorem builds on a combination of parallel composition property and post-processing processing property of differential privacy. Since different algorithm modules use disjoint sets of samples, the whole algorithm uses each single individual observation only once, thereby avoiding the privacy costs of sequential composition.

Our proof of Theorem 6 follows from this theorem.

Proof Use $\tilde{\mathcal{B}}(\cdot) = \mathcal{B} \circ \mathcal{T}(\cdot; h_1, \ldots, h_k)$ to denote the combined procedure of the data transformation $\mathcal{T}$ followed by algorithm module $\mathcal{B}$. Because $\mathcal{B}$ is f_2-DP and $\mathcal{T}(\cdot; h_1, \cdots, h_k)$ is a 1-stable transformation, i.e. neighboring datasets are still neighbors after transformation for any $h_1 \in \mathcal{H}_1, \ldots, h_k \in \mathcal{H}_k$, $\tilde{\mathcal{B}}(\cdot; h_1, \ldots, h_k)$ is f_2-DP for any $h_1, \ldots, h_k$.

The full algorithm, $\mathcal{M} : \mathcal{S} \mapsto (\mathcal{A}_1(S_{1,1}), \ldots, \mathcal{A}_k(S_{1,k}), \tilde{\mathcal{B}}(S_2; \mathcal{A}_1(S_{1,1}), \mathcal{A}_2(S_{1,2}), \ldots, \mathcal{A}_k(S_{1,k})))$, is a $k+1$-composed algorithm with disjoint inputs. Each algorithm module has its corresponding DP guarantee. The conditions of lemma 8 check. Hence, $\mathcal{M}$ is f-DP with $f = \min\{f_{1,1}, \ldots, f_{1,k}, f_2\}^{**}$. ■