Stat 236 Project: High-Dimensional Distributed Learning for Causal Inference

Larry Han, Christina Howe, Lara Maleyeff, Avi Swartz

December 12, 2020

Abstract

Distributed learning using electronic health records is a potentially attractive but underdeveloped area of research, particularly for causal modeling. We aim to develop novel procedures for the estimation of targeted average treatment effects in a distributed learning setting. The problem is challenging because patient-level data cannot be shared, there are costs to communicating even summary-level data, and there is a potentially high-dimensional covariate space to consider. We propose two approaches: (1) a model-based approach that leverages regularization in the first step for variable selection and double robustness in the second step for estimator development; (2) a gradient based approach that passes estimators of the data instead of aggregating the data in a central site. Through extensive simulations, we show good finite sample properties of our proposed methods. We also provide future avenues of research extending from our paper.

1 Introduction

The ability to integrate information across multiple institutions or sites can improve the generalizability of findings and accelerate appropriate decision-making, which is especially crucial during health crises (Friedman et al., 2010; Hripcsak et al., 2016). For example, the COVID-19 pandemic has highlighted the need for novel approaches to efficiently and safely analyze data across different populations and healthcare systems. Our paper is motivated by the international consortium (4CE) for COVID-19, a joint effort of

hundreds of hospitals across 50 sites and 17 countries to inform physicians, epidemiologists, and policymakers about COVID-19 patients using electronic health records (EHR) data (Brat et al., 2020). In this study, the aim is to estimate the average treatment effect (ATE) for a target population of interest, such as a single site (site-level), a few sites that are in the same country (country-level), or the overall population ATE (population-level) by leveraging information from all sites, where we are constrained by the inability to share patient-level information across sites.

EHRs contain large-scale data on the health status and treatment information of patients collected during routine clinical care. In the past several years, the increased adoption of EHRs has led to new opportunities for integrated analysis using EHRs (Payne et al., 2015). However, integrated analysis using EHRs is challenging because (i) it is often constrained by privacy provisions that make it impossible to share patient-level data across sites, (ii) there are communication costs of time and human resources associated with transferring even summary-level data between sites, and (iii) the dimension of observed covariates may be very large. These obstacles underscore a substantial need for privacy-preserving, communication-efficient integrative analysis methods that account for heterogeneity both within local sites and across healthcare systems.

In previous literature, distributed algorithms have been developed in the low-dimensional regression setting by decomposing tasks to be completed within each site so that patient-level information does not need to be pooled. These methods have been developed for linear regression (Chen et al., 2006), logistic regression (Wu et al., 2012; Duan et al., 2020), and time-to-event regression (Lu et al., 2015; Duan et al., 2020). Recently, much research has focused on the high-dimensional regression setting by imposing various sparsity assumptions (Lee et al., 2017; Battey et al., 2018) and constructing a surrogate likelihood function as approximation of the global likelihood function (Jordan et al., 2018).

Despite great interest in leveraging multi-site EHR data to understand causal effects of diffrent interventions, there has been little development of distributed learning algorithms for causal models, either in the low-dimensional or high-dimensional settings. In this paper, we aim to address this knowledge gap by proposing two novel approaches. The first approach is a two-step procedure that first fits a penalized regression at the central site to obtain the covariates to be used in the second step, where we robustly estimate target model parameters at local sites using only site-specific patient data and

summary level-information from other sites. The second approach leverages the gradient based optimization software infrastructure developed for neural network inference to move the estimators to the same location as the data, allowing the data to be used in inference without needing to remove the data from its secure location.

The rest of the paper is organized as follows. In Section 2, we formalize the setting, notation, and target parameter. In Section 3, we describe the regularization procedure and provide the model-based distributed learning algorithm and proposed estimators for approach 1 and the gradient based procedure for approach 2. In Section 4, we describe the data generation procedure for the simulation studies used to assess properties of our proposed estimators and we show results from those studies. In Section 5, we conclude by highlighting the significance of this work and provide avenues for future directions and extensions. In Section 6, we have all the relevant figures and tables for the paper.

2 Setting, Notation, and Target Parameter

We assume that we have K independent study sites with n_r patients in site r for $r = 1, ..., K$. Let $n = \sum_{r=1}^{K} n_r$ be the total sample size. Let $R \in \{1, ..., K\}$ denote the sites. For patient i in site r, denote the outcome as Y_i^r, which can be continuous or discrete, the p-dimensional baseline covariate vector as $\mathbf{X}_i^r$ (where p may be larger than n), and the treatment indicator as A_i, with $A = 1$ for a new therapy and $A = 0$ for placebo.

We assume we are in a sparse setting in which p^* covariates are significant with $p^* \ll p$. Note that under the usual potential outcomes framework, we denote $Y_i^r = A_i Y_i^{r(1)} + (1 - A_i) Y_i^{r(0)}$, where $Y_i^{r(a)}$ is the potential outcome for patient i in site r under treatment $A = a$. The observed data at each site is $\mathbf{D}_i^r = (Y_i^r, \mathbf{X}_i^{r\top}, A_i)^\top$. Assume also that the observations in site r, $\mathcal{D}^r = \{\mathbf{D}_i^r, i = 1, ..., n_r\}$ are independent and identically distributed. For identifiability, we require the standard causal inference assumptions:

$$\pi_{a,r}(\mathbf{x}) \equiv \mathbb{P}(A = a \mid \mathbf{X} = x, R = r) \in (0, 1) \tag{1}$$

$$(Y^{(1)}, Y^{(0)}) \perp (A, R) \mid \mathbf{X} \tag{2}$$

Assumption (1) states within all covariate levels, patients may receive either treatment or control. Assumption (2) implies that $\mathbf{X}$ includes all confounders

that may affect the primary outcome and treatment simultaneously (Imbens and Rubin, 2015; Rubin, 2005).

Our goal is to estimate a targeted average treatment effect (TATE) parameter for a specified target population, $\mathcal{T}$,

$$\Delta_{\mathcal{T}} = \mu_{1,\mathcal{T}} - \mu_{0,\mathcal{T}}, \quad \mu_{a,\mathcal{T}} = \mathbb{E}_{\mathbf{X}_{\mathcal{T}}}\{Y^{(a)} \mid R \in \mathcal{T}\}. \tag{3}$$

Note that the TATE across all sites reduces to the ATE, which is the special case when $\mathcal{T} = \{1, \ldots, K\}$. Throughout, we let $f(\mathbf{x} \mid R = r)$ be the density function of $\mathbf{X}$ in site r, and denote the density function of the target population as

$$f(\mathbf{x} \mid R \in \mathcal{T}) = \frac{\sum_{r \in \mathcal{T}} f(\mathbf{x} \mid R = r)\mathbb{P}(R = r)}{\mathbb{P}(R \in \mathcal{T})}.$$

3 Method

3.1 Approach 1

3.1.1 Regularization for Variable Selection

For the central site only, we fit a penalized linear regression with Y as the continuous outcome and $\mathbf{X}$ as the set of covariates available in the EHR. The method is valid for discrete outcomes as well, but we take Y to be continuous in the remainder of this paper. Penalized linear regression minimizes a penalized sum of squares of the form

$$\frac{1}{n}||Y - \mathbf{X}\beta||_2^2 + \rho_\lambda(\beta), \tag{4}$$

with respect to β. We consider three common penalty functions for ρ: LASSO, SCAD, and MCP (all as described in Section 3.2.1). To implement LASSO, we use the glmnet package in R, with λ selected using k-fold cross-validation. In this procedure, we fit the penalized regression over a grid of candidate λ values, each time leaving out $1/k$ of the data. We select the λ which minimizes the deviance-based cross-validation error. For LASSO, we take $k = 5$. To implement SCAD and MCP, we use the ncvreg package in R with $a = 3$ for MCP and $a = 3.7$ for SCAD. Here, λ is selected using 10-fold cross-validation.

From each of the three penalized regressions with cross-validated λ, we extract $\widehat{\mathbf{p}}_r^* \subseteq \{1, \ldots, p\}$, the indices of the covariates with a non-null relationship with Y^r. We then compute the estimators described on Section 3.1.3 using the $\{Y^r, A^r, \mathbf{X}^r_{\mathbf{p}_r^*}\}$.

3.1.2 Models

To robustly estimate TATE, we propose the estimation of three models:

$$m_{a,r}(\mathbf{x}) = \mathbb{E}(Y^r \mid A^r = a, \mathbf{X}^r = \mathbf{x}) = \mathbb{E}(Y^r \mid A^r = a, \mathbf{X}^r = \mathbf{x}, R = r) \quad (5)$$

$$\pi_{a,r}(\mathbf{x}) = \mathbb{P}(A^r = a \mid \mathbf{X}^r = \mathbf{x}) = \mathbb{P}(A^r = a \mid \mathbf{X}^r = \mathbf{x}, R = r) \quad (6)$$

$$w_r(\mathbf{x}) = \frac{f(\mathbf{x} \mid R = r)}{f(\mathbf{x} \mid R \in \mathcal{T})} \quad (7)$$

where (5) is the outcome regression model (OR), (6) is the propensity score model (PS), and (7) is the density ratio model (DR). First, $m_{a,r}(\mathbf{x})$ can be estimated distributively, i.e., each site will estimate $m_{a,r}(\mathbf{x})$ and transfer their estimate to the central site, which can then aggregate the estimates to obtain a sample-size weighted average

$$\widehat{m}_a(\mathbf{x}) = \sum_{r=1}^{K} \rho_r \widehat{m}_{a,r}(\mathbf{x}) \quad (8)$$

Note that we can aggregate the estimates in this way because we assume ignorability, i.e. $Y \perp R \mid \mathbf{X}$. We fit the OR model using a linear regression with Y as the outcome and $\{A^r, \mathbf{X}^r_{\widehat{\mathbf{p}}^*}\}$ as the covariates. Next, the propensity score, $\pi_{a,r}(\mathbf{x})$, can be estimated by imposing a parametric model, denoted $\pi_{1,r}(\mathbf{X}^r; \alpha)$, where α is a finite-dimensional parameter that can be estimated as the standard maximum likelihood estimator, $\widehat{\alpha}$. For example, we could use a logistic regression model where $\pi_1(\mathbf{X}; \alpha) = G\{\alpha^\top \Phi(\mathbf{X})\}$, where $\Phi(\mathbf{X})$ is a vector of basis functions of $\mathbf{X}$ to account for potential non-linear effects. We fit the PS model with A as the outcome and $\mathbf{X}^r_{\widehat{\mathbf{p}}_r^*}$ as the covariates, using logistic regression. Since the penalization methods result in a drastic dimension reduction $|\widehat{\mathbf{p}}^*| \ll p$, traditional linear and logistic regression theory holds.

Finally, $w_r(\mathbf{x})$ can be estimated by imposing a semiparametric model:

$$f(\mathbf{x} \mid R = r) = f(\mathbf{x} \mid R = 1) g(\mathbf{x}; \boldsymbol{\gamma}_r), \quad (9)$$

where $g(\cdot;\cdot)$ satisfies $g(\mathbf{x};\mathbf{0}) = 1$ and $\mathbb{E}\{g(\mathbf{X};\boldsymbol{\gamma}_{r,0}) \mid R = 1\} = 1$. For example, we could specify an exponential tilt model (Qin, 1998) where

$$g(\mathbf{x};\boldsymbol{\gamma}_r) = \exp(\boldsymbol{\gamma}_r^\top \psi(\mathbf{x}))$$

where $\psi(\cdot)$ is some specified basis. First, the central site will obtain its mean of $\mathbf{X}$ as

$$\bar{\mathbf{X}}^1 = n_1^{-1} \sum_{i=1}^{n_1} \mathbf{X}_i^1 \tag{10}$$

and pass $\bar{\mathbf{X}}_1$ to all of the local sites, which is sufficient since the density ratio model allows us to write

$$\bar{\mathbf{X}}^1 = \int \mathbf{x} f(\mathbf{x} \mid R = 1) d\mathbf{x} = \int \mathbf{x} f(\mathbf{x} \mid R = r) e^{\boldsymbol{\gamma}_r^\top \psi(\mathbf{x})} d\mathbf{x}. \tag{11}$$

Each local site will then be able to locally estimate their density ratio model parameter $\boldsymbol{\gamma}_r$ as the solution to the following estimating equation

$$n_r^{-1} \sum_{i=1}^{n_r} (\mathbf{X}_i^r)^\top g((\mathbf{X}_i^r)^\top;\boldsymbol{\gamma}_r) - \bar{\mathbf{X}}^1 = \mathbf{0}$$

and we denote the estimators as $\widehat{\boldsymbol{\gamma}}_r$ for $r = 2, ..., K$. We estimate $\widehat{\omega}_r(\mathbf{x})$ as

$$\widehat{\omega}_r(\mathbf{x}) = \exp(\widehat{\boldsymbol{\gamma}}_r^\top \mathbf{x}) \tag{12}$$

Finally, the density ratio weights can be estimated as

$$w_r(\mathbf{x}) = \frac{\omega_r(\mathbf{x})}{\sum_{r\prime} \rho_{r\prime} \omega_{r\prime}(\mathbf{x})}, \tag{13}$$

where $\rho_{r\prime} = \frac{n_{r\prime}}{n}$.

3.1.3 Proposed Estimators

We consider four different estimators of TATE: OR estimator, IPW estimator, doubly robust estimator, and weighted doubly robust estimator. We report the following estimators in site r, per each penalization method. The first is an outcome regression estimator, based on predicted values from Model 4, with $n_{a,r} = \sum_{i=1}^{n_r} I(A_i = a)$:

$$\widehat{\mathrm{OR}}_r = \frac{1}{n_{1,r}} \sum_{i=1}^{n_{1,r}} \widehat{m}_{1,r}(\mathbf{X}_i^r) I(A_i^r = 1) - \frac{1}{n_{0,r}} \sum_{i=1}^{n_{0,r}} \widehat{m}_{0,r}(\mathbf{X}_i^r) I(A_i^r = 0) \tag{14}$$

Next, we compute an inverse-probability weighted (IPW) estimator:

$$\widehat{\text{IPW}}_r = \frac{1}{n_r}\sum_{i=1}^{n_r}\left(\frac{Y_i^r I(A_i^r=1)}{\widehat{\pi}_{a,r}(\mathbf{X}_i^r)} - \frac{Y_i^r I(A_i^r=0)}{\widehat{\pi}_{a,r}(\mathbf{X}_i^r)}\right) \tag{15}$$

The OR estimator and IPW estimator will not give consistent estimates if the models are misspecified. For increased robustness to model misspecification, we consider a doubly robust estimator:

$$\widehat{\text{DR}}_r = \widehat{\mu}_{r,1} - \widehat{\mu}_{r,0}, \tag{16}$$

with

$$\widehat{\mu}_{r,a} = n_r^{-1}\sum_{i=1}^{n_r}\left\{\frac{Y_i^r I(A_i^r=a)}{\widehat{\pi}_{a,r}(\mathbf{X}_i^r)} - \frac{I(A_i^r=a)-\widehat{\pi}_{a,r}(\mathbf{X}_i^r)}{\widehat{\pi}_{a,r}(\mathbf{X}_i^r)}\widehat{m}_{a,r}(\mathbf{X}_i^r)\right\} \tag{17}$$

This estimator is doubly robust in the sense that the estimator will be consistent if at least one of the OR and PS models are correctly specified (Robins et al., 1994).

If there is between-site heterogeneity in the distribution of covariates, then the estimated ATE within each site will not be a consistent estimate of the ATE for the target site distribution. To account for this heterogeneity, we consider the weighted doubly robust estimator, which projects each patient covariate profile in a given site to the distribution in the target site(s) through the density ratio weight. The weighted doubly robust (WDR) estimator has the form:

$$\widehat{\text{WDR}}_r = \widetilde{\mu}_{r,1} - \widetilde{\mu}_{r,0}, \tag{18}$$

with

$$\widetilde{\mu}_{r,a} = n_r^{-1}\sum_{i=1}^{n_r} w_r(\mathbf{X}_i^r)\left\{\frac{Y_i^r I(A_i^r=a)}{\widehat{\pi}_{a,r}(\mathbf{X}_i^r)} - \frac{I(A_i^r=a)-\widehat{\pi}_{a,r}(\mathbf{X}_i^r)}{\widehat{\pi}_{a,r}(\mathbf{X}_i^r)}\widehat{m}_{a,r}(\mathbf{X}_i^r)\right\} \tag{19}$$

For each of the four estimators, to obtain a global estimator that combines the site-specific estimators, we take a sample-size adjusted mean.

3.2 Approach 2

3.2.1 Overview

In an alternate approach, we leverage the gradient based optimization software infrastructure developed for neural network inference. In particular, we focus on two major factors of the problem:

1. We assume that there are only a small number of covariates that actually impact the treatment effect.

2. We are constrained in sharing our data. We cannot take patient-level data from different sites and pool them for a single estimation algorithm due to privacy concerns.

In order to address concern 1, we use penalized regression for variable selection by modeling the effect of different covariates on the outcome Y and the treatment A. Because of the large amount of resources devoted to improving gradient based optimization techniques for training neural networks, as well as another reason that will become apparent shortly, we decided to use PyTorch (Paszke et al., 2019) to implement a regularized regression algorithm. There is a graphical depiction of our algorithm in Figure 1.

Given a set of covariate data $\mathbf{X}$ and an observed outcome Y, we used the PyTorch Adamax optimizer to minimize the regularized regression loss and infer the sparse coefficient vector β:

$$L(\beta) = \frac{1}{n}||Y - \mathbf{X}\beta||_2^2 + \rho_\lambda(|\beta|) \tag{20}$$

where $\rho_\lambda(|\beta|)$ is a elementwise penalization term on β to encourage sparsity. Specifically, the penalization terms we used for the three types of regularizations in Approach 2 are (Breheny and Huang, 2011)

- LASSO: $\rho_\lambda(\beta) = \lambda||\beta||_1$
- SCAD: $\rho_\lambda(\beta) = \begin{cases} \lambda\beta & \text{if} \quad \beta \leqslant \lambda \\ \frac{a\lambda\beta - 0.5(\beta^2+\lambda^2)}{a-1} & \text{if} \quad \lambda < \beta \leqslant a\lambda \\ \frac{\lambda^2(a^2-1)}{2(a-1)} & \text{if} \quad \beta > a\lambda \end{cases}$

  where $a > 2$ (we took $a = 3.7$ according to the suggestion on Lecture 4 Part 1 Slide 52)

- MCP: $\rho_\lambda(\beta) = \begin{cases} \lambda\beta - \frac{\beta^2}{2a} & \text{if} \quad \beta \leqslant a\lambda \\ \frac{1}{2}a\lambda^2 & \text{if} \quad \beta > a\lambda \end{cases}$

  where $a > 1$ (we took $a = 2.7$)

Since we are concerned about the potential misspecification that could be introduced by conducting variable selection only at the central site in Approach 1, we instead pass the inferred parameters from site to site in Approach 2. We start out with a random guess for β. Then we pass β to the first site, which uses the optimizer to minimize the loss function calculated using only the data at the first site. Then we pass the β to the next site, which uses the data at the next site to calculate a new loss function, which is then optimized. The passing continues in a circle until the β converges.

Importantly, unlike most regression implementations that automatically infer the best β for the data at one site, we only take 10 optimizer steps before passing the β values to the next site. As each site takes 10 optimizer steps at a time, each site adjusts the inference only a little bit at a time. Then after the β has been passed around sufficiently, it will settle on a value that is consistent with all the sites at the same time. This is why we needed to reimplement regression using a gradient based optimizer.

3.2.2 Distributed Regression

We wrote a PyTorch function that would take as input lists of $\mathbf{X}$ and Y data from each site and a specific loss function. The function would then optimize the loss function without ever looking at data from different sites at the same time. By passing the loss function from equation 20 and the relevant regularization term $\rho_\lambda(|\beta|)$ from Section 3.2.1, we created a distributed linear regression, a distributed logistic regression, a distributed LASSO regression, a distributed MCP regression, and a distributed SCAD regression. Furthermore, as the optimization code can operate with any loss function passed to it, it will easily generalize to any other use case.

Upon testing, we did find that the regularized regression did not quite push unimportant variables to 0. However, by setting a cutoff of $0.1 \cdot \max(\beta)$ and setting all values of β to 0 if they were below the threshold, we were able to enforce sparsity. After doing so, we found this implementation quite successful (Figure 2).

In testing, the distributed LASSO outperforms the solution found by running a prepackaged LASSO implementation (we used PICASSO (Ge et al., 2019)) on each site separately and then averaging the β together. The distributed LASSO is even able to outperform the solution found by running the prepackaged LASSO on all the data aggregated together, indicating that this implementation is able to perform distributed penalized regression quite

well.

3.2.3 ATE Estimation with Distributed Regression

After developing the distributed regression code, we used it to infer the average treatment effect from site partitioned data.

- We use cross validation to determine the optimal λ value for the regularized regression of interest (either LASSO, SCAD, or MCP). For each site, we split the data into five partitions and take a series of 100 λ values chosen by the PICASSO package. Then for each λ, we use PICASSO to regress on four of the five partitions and evaluate on the fifth partition, and then average over all five partitions to determine the score for that λ. Finally, we average the score for each λ across all the sites to determine which λ is best for the entire data.

- Then given the optimal λ, we can run the distributed regression on the entire data. This allows us to utilize the entire dataset for the regression without breaking privacy restrictions.

- From here, there are multiple options. To estimate the average treatment effect, it suffices to take the important variables from the previous step, add in the treatment indicator variable A, and run a distributed linear regression using only those important variables. Then we can simply extract the coefficient of A. The results of this method will be examined further in Section 4.3.

- Alternatively, we can take the important variables and move back to the work in Section 3.1.2 to get the TATE. While the rest of the inference will proceed as in approach 1, the non-null variables will have been selected using more data, which means that the whole process should be more accurate.

- Finally, there are other possible ways to infer the TATE using this approach, which we will explore in the discussion.

4 Simulation Study

4.1 Data Generation

We conducted a simulation study of the proposed method. We have $K = 50$ study sites with $n \in \{100, 200, 500\}$ subjects in each site. For $p \in \{10, 100, 500\}$, let $\mathbf{X_r} = (X_1, \ldots, X_p)^T$ be multivariate normal with mean $\mu_{p,r}$ and covariance matrix with $\sigma^2 = 4$ on the diagonal and 0 on the off-diagonals. The $\mu_{p,r}$ are also multivariate normal, with mean 0 and covariance matrix with 1 on the diagonal and 0 on the off-diagonals. Let A be a binary variable representing treatment assignment that depends only on the $p^* = 5$ covariates $(X_1, X_2, X_3, X_4, X_5)$ such that:

$$\pi_{a,r}(\mathbf{x}) = \mathbb{P}(A^r = a \mid \mathbf{X}^r = \mathbf{x}) = \text{expit}(.05X_1^r + .1X_2^r + .1X_3^r - .05X_4^r - .1X_5^r)$$

Let Y be a continuous outcome variable that depends only on the assigned treatment A and the same p^* covariates such that:

$$m_{a,r}(\mathbf{x}) = \mathbb{E}(Y^r \mid A^r = a, \mathbf{X}^r = \mathbf{x}) = \beta_A A^r + .5X_1^r + X_2^r + .5X_3^r - .5X_4^r - X_5^r$$

We allow the treatment effect β_A to be either 0 or 10.

4.2 Approach 1 Results

Across various settings, we compare the properties of five different estimators for the TATE: outcome regression, inverse probability weighted, doubly robust using all sites, doubly robust using data from the target sites only, and a weighted doubly robust using data from all sites.

1. $\widehat{\text{OR}}_r$
2. $\widehat{\text{IPW}}_r$
3. $\widehat{\text{DR}}_r$ where $r \in (1, \ldots, K)$
4. $\widehat{\text{DR}}_r$ where $r \in \mathcal{T}$ a.k.a. "Naive Targeted Doubly Robust"
5. $\widehat{\text{WDR}}_r$

For all except the Weighted Doubly Robust estimator, the mean of the site-specific estimators is chosen as the overall estimator. For the Weighted Doubly Robust estimator, the choice of aggregation method is important. We tried several methods, of which two are shown in greater detail. In this approach, we additionally specify the target site $\mathcal{T}$ to be either all 50 sites or a smaller set of 5 sites.

Models Correctly Specified First, we evaluate the properties of these five estimators in the setting where the outcome regression and propensity scores are known. With 50 simulations in each of $n \in (100, 200, 500)$, $p \in (10, 100, 500)$, $\beta_A \in (0, 10)$, and both the small target of five sites and the full target of $K = 50$, we see that all of the estimators are indeed unbiased (Table 1). We see differences in the empirical standard errors of these estimators. Notably, the Weighted Doubly Robust estimator $\widehat{\text{WDR}}_r$ is less efficient than the other estimators when the target is the entire sample size (Figure 3). Here, we aggregated the site-specific Weighted Doubly Robust estimators by taking an unweighted median. Overall, all of the estimators perform reasonably well in all tested simulation settings when all models were correctly specified.

Model Misspecification Using the same data generation mechanism and simulation settings, we approached model misspecification in two different ways. First, we misspecified the outcome regression or the propensity score model by removing two of the five p^* important covariates from the linear and logistic regression used to estimate the outcome and propensity score respectively. As expected, this results in bias in the Outcome Regression estimator $\widehat{\text{OR}}_r$ and Inverse Probability Weighted estimator $\widehat{\text{IPW}}_r$ respectively. All three doubly robust estimators are unaffected by these misspecifications, again as expected. Figure 4 shows these results for 50 simulations in the setting when $n = 100, p = 100$, the true treatment effect is 10, and the target is all $K = 50$ study sites.

Secondly, to show the additional robustness of the target-specific Doubly Robust estimator ($\widehat{\text{DR}}_r$ where $r \in \mathcal{T}$) and the Weighted Doubly Robust Estimator $\widehat{\text{WDR}}_r$, we allow the treatment effect β_A to vary across the sites. Specifically, the site-specific means of X_1^r and X_2^r were distributed $N(-2, 1)$ rather than $N(0, 1)$ in the small five-site target. Additionally, X_1^r was made to be an effect modifier of the treatment effect, with a coefficient chosen so

that the treatment effect would be approximately 10% larger in the target population than in the overall population. Results of 50 simulations for $n = 100, p = 100$, the true treatment effect is 10 are shown in Figure 5. Notably, the Outcome Regression estimator and the Doubly Robust estimator that was not site-dependent show bias in the case when the target site is small and has a different covariate distibution from the other sites, but are unbiased when the target is all fifty sites. In both of these settings, the Weighted Doubly Robust estimator was aggregated with a weighted median, which performs reasonably well in the small target though appears to be biased when used to estimate the average treatment effect across all sites.

High Dimensional Distributed Estimators Lastly, we combine the central-site regularization step with the distributed step to represent the implementation of this method in the high-dimensional setting. The details of the regularization testing are discussed earlier. As mentioned, we compared three regularization methods: LASSO, MCP, and SCAD. In the full approach, the LASSO struggled when $p = 500$ and $n = 100$, where it often selected many extraneous covariates and did not always converge (Table 2). MCP and SCAD performed better in this case. As the LASSO was most likely to show model misspecification and the Inverse Probability Weighted Estimator is not robust to misspecification of the propensity score model, it has a large empirical standard error in the $n = 100, p = 500$ case (Figure 6). The bias and empirical standard errors from 20 simulations in each setting are given in Tables 3 and 4.

4.3 Approach 2 Results

In order to validate the average treatment effect inference algorithm described in Section 3.2.3, we used the same data simulation set up as described in Section 4.1, just without any target specification. We then varied the number of covariates, the number of patients per site, as well as the treatment effect. In all cases, we took the coefficient of the treatment indicator variable as an estimator of the treatment effect. We also simulated the data 20 times for each set of parameters to generate and plot a prediction interval for the estimator, which we thought would be more useful than a confidence interval.

- **Varying the number of covariates** (Figure 7). We can see that both in the case of a large treatment effect and a zero treatment effect,

the predicted treatment effect is accurate. Furthermore, the width of the predictive interval does not particularly change as the number of covariates increases after 100 (with the exception of a strange and suspicious kink in the curve at no treatment effect and 100 covariates), indicating that the algorithm is able to handle a wide sparsity range: in this case from $\frac{5}{100}$ to $\frac{5}{500}$ significant.

- **Varying the number of patients at each site** (Figure 8). Here, the predicted treatment is accurate for both the large and zero treatment effect. Furthermore, there does seem to be a narrowing of the predictive interval as the data size increases. This means that predictions get more accurate as the data size increases, which makes sense.

- **Varying the treatment effect size** (Figure 9). In this case, there seems to be a drop in predictive accuracy for negative treatment effects. However, this is not reflected in a symmetric drop in accuracy for positive treatment effects and the code took much longer to run for the negative values than the position values of the treatment effect. It seems that there is an assumption being violated in the model somewhere that is causing strange behavior. Additionally, the error in the prediction is very small relative to the estimator being predicted, so this seems not too bad.

Also of interest in the results is that there did not seem to be large differences in the final estimation between the different regularization methods. Despite the regularization differing in the bias and convexity of the penalty, it seems that the average treatment effect estimation here is sufficiently robust to the differences.

5 Discussion

5.1 Discussion of Approach 1

We have proposed a two-step procedure for robust estimation of a targeted average treatment effect parameter in a distributed setting, where patient-level information cannot be shared between potentially heterogeneous sites. In step 1, the method takes advantage of sparsity in the regression of $Y \mid A, \mathbf{X}$ to reduce a high-dimensional regression problem to a low-dimensional problem. Notably, this penalized regression only needs to be fit in the central site

rather than all sites. We explore several existing methods for conducting this penalized regression, including LASSO, SCAD, and MCP. Following variable selection, in step 2, we propose a model-based procedure that requires the estimation of outcome regression, propensity score, and density ratio model estimation. We develop doubly robust and weighted doubly robust estimators to safely leverage information from all sites in the estimation of the average treatment effect in a target population of interest. Notably, our method allows the user to specify which target sites are of interest, so that findings can be targeted at the site-, country-, or even population-level. Through extensive simulation studies, we show minimal empirical bias and consistent estimators that are robust to various model misspecification scenarios.

In future work, we aim to derive the asymptotic variance of the proposed estimators and to discuss the efficiency of the proposed estimator from the following perspectives. First, we may consider the loss due to data sharing. We would show that our estimator achieves the same efficiency compared to the case when the outcome regression model and the density ratio model can be estimated without constraints on data sharing. This will normally lead to a constraint on the relationship between n and K, i.e., K cannot be too large. Second, we may consider the benefit of leveraging multiple datasets. Under the above regime of n and K, when all three models are are correctly specified, we may compare our proposed estimator with the semiparametric efficient estimator obtained from using only the target population to examine how much efficiency gain we obtain. Third, we may consider the best possible scenario when all of the data are homogeneous or when the target population is the overall population. Then, our proposed estimator can achieve the same efficiency as the efficient estimator in the corresponding settings.

An alternative approach to explore would be to consider regularized calibrated estimators with LASSO penalties and carefully chosen loss functions for fitting the propensity score models and outcome regression models (Tan et al., 2020; Tan, 2020). This may allow us to reduce our two-step procedure to a single step, although we would need to be careful about which covariates are ultimately included in the models.

5.2 Discussion of Approach 2

We have proposed a method of running a regression on a set of data without needing to look at the entire data at once. We have shown that the distributed regression is comparable or even better than the regression per-

formed by looking at the whole data at once and that the results of the regression is much better than the regression performed by averaging the results of all the sites individually (Figure 2).

This means we have a way of addressing two of the three concerns in the analysis of electronic health records brought up in the introduction. We do not require patient level data to be shared from one institution to another, respecting privacy concerns. We are also able to reduce the dimensions of the covariates using the penalization. However, we have a large number of data transfers that must be made to move the estimate of β from one site to the next, so we do not address the concern of communication-efficiency in this approach. Nonetheless, the costs of transmitting data are probably the least of the three concerns, so this may be a reasonable tradeoff.

As mentioned in Section 3.2.3, there are multiple ways of inferring the treatment effect after we have selected meaningful variables with the distributed regression. We have shown in Section 4.3 that simply running a distributed regression using the selected variables and the treatment indicator A against the the response variable Y yields the coefficient of A as an excellent estimator of the average treatment effect.

In future work, we aim to use the distributed regression approach to improve the estimation of the targeted average treatment effect. One straightforward strategy would be to use the distributed regression to perform variable selection, and then proceed with the rest of approach 1.

Alternatively, we could extend the loss optimization method of Approach 2 further. For each patient, we could calculate a weighting of how similar that patient is to the target. For example, we could use the density ratio weights from Section 3.1.2. Then when calculating the distributed regression to determine the coefficient on the treatment indicator A, we could weight each patient's contribution to the loss to correspond to their similarity to the target. This way more similar patients contribute more to the treatment effect, resulting in the estimation of a targeted average treatment effect.

6 Figures and Tables

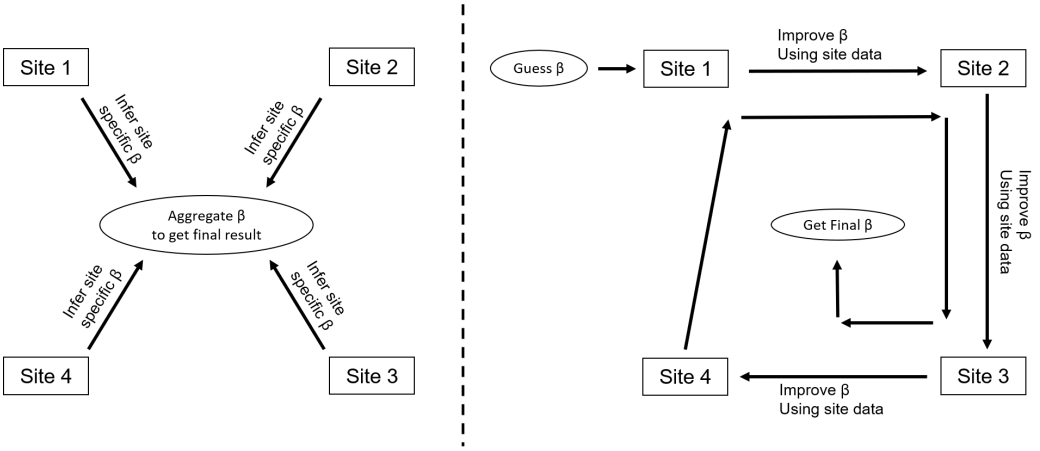

Figure 1: A visual depiction of the distributed regression model used in Approach 2. There are here four sites with data that we wish to infer the regression coefficient β from. However, none of the sites can send their data out due to privacy restrictions.
In the naive model on the left, each site calculates their own β, which can be sent out and aggregated. In our proposed model on the right, the first site takes a random guess of β and improves it slightly using its own data. Then it sends β to the next site, which uses its own data to improve β slightly again. This continues in a circle until the β converges to a value consistent with all the sites' data.

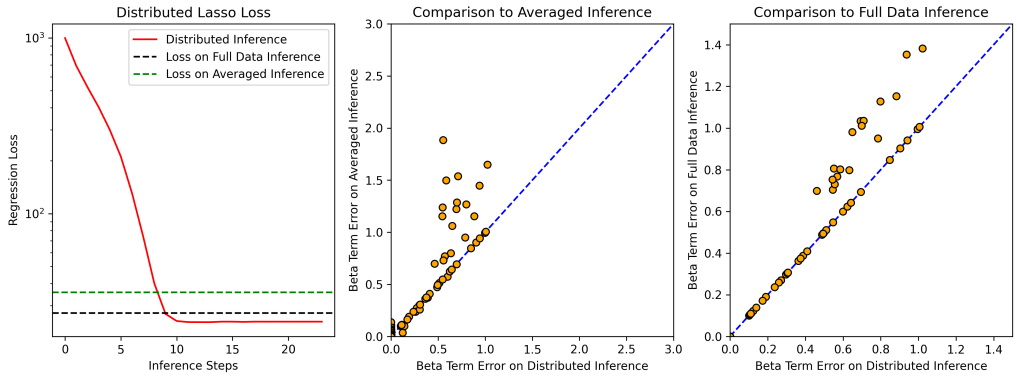

Figure 2: A comparison between the distributed LASSO inference as described in Section 3.2.2, the full data inference, which runs prepackaged LASSO with the PICASSO package on all the data simultaneously by ignoring privacy concerns, and the averaged inference, which runs prepackaged LASSO on all the different sites independently and averages the inferred β together. This data was generated by taking 20 sites of 20 people of 1000 covariates. The covariates are $\sim \mathcal{N}(0,1)$ and the 50 nonzero entries of β are also $\sim \mathcal{N}(0,1)$.
In the left plot, we can see that the distributed LASSO (red) reaches a better loss than both the other methods. In the middle plot, we can see that the distributed LASSO predicts every entry of β more accurately than the averaged inference, and in the right plot we can see that the distributed LASSO predicts every entry of β more accurately than the full data inference.
The fact that the distributed LASSO inference outperforms the full data LASSO despite them seeing all the same data indicates that either the stepwise optimization of the distributed LASSO has a beneficial effect or the PyTorch Adamax optimizer for the distributed LASSO is better than the optimizer used for the prepackaged LASSO.

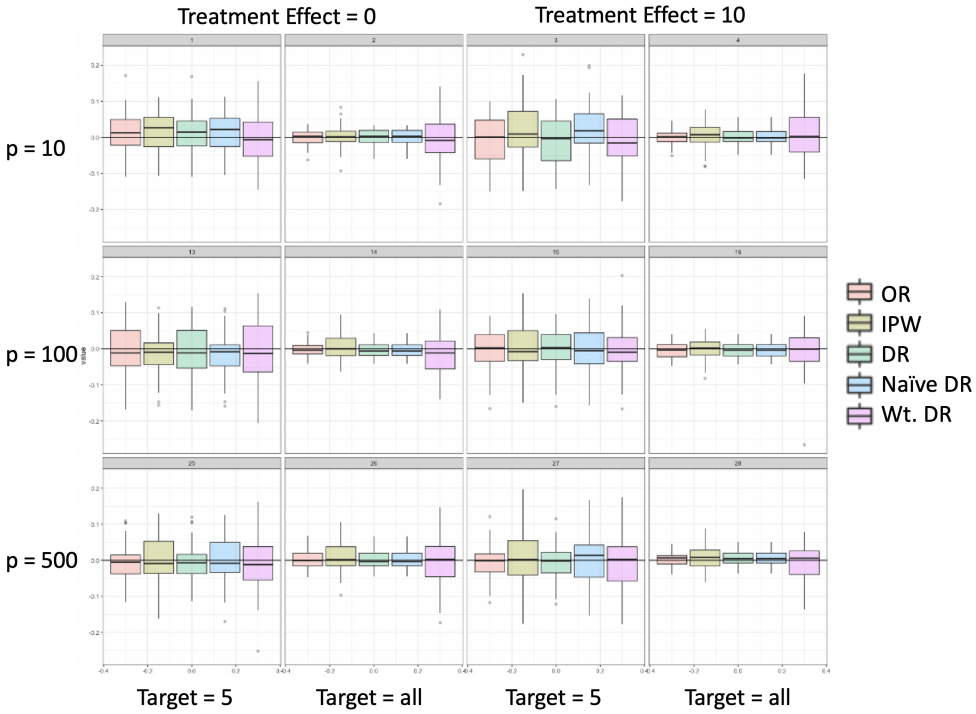

Figure 3: A comparison between five estimators of the TATE, $\widehat{\mathrm{OR}}_r$, $\widehat{\mathrm{IPW}}_r$, $\widehat{\mathrm{DR}}_r$ where $r \in (1, \ldots, K)$, $\widehat{\mathrm{DR}}_r$ where $r \in \mathcal{T}$, and $\widehat{\mathrm{WDR}}_r$ for correctly specified models. $K = 50, n = 100, p \in \{10, 100, 500\}, \beta_A \in \{0, 10\}$, and $\mathcal{T}$ is either five sites or all $K = 50$ sites with no structured between-site heterogeneity. Note that all of the estimators are unbiased and that the Weighted Doubly Robust estimator has a similar empirical standard error to the other estimators for the five-site target but is less efficient when estimating the overall average treatment effect.

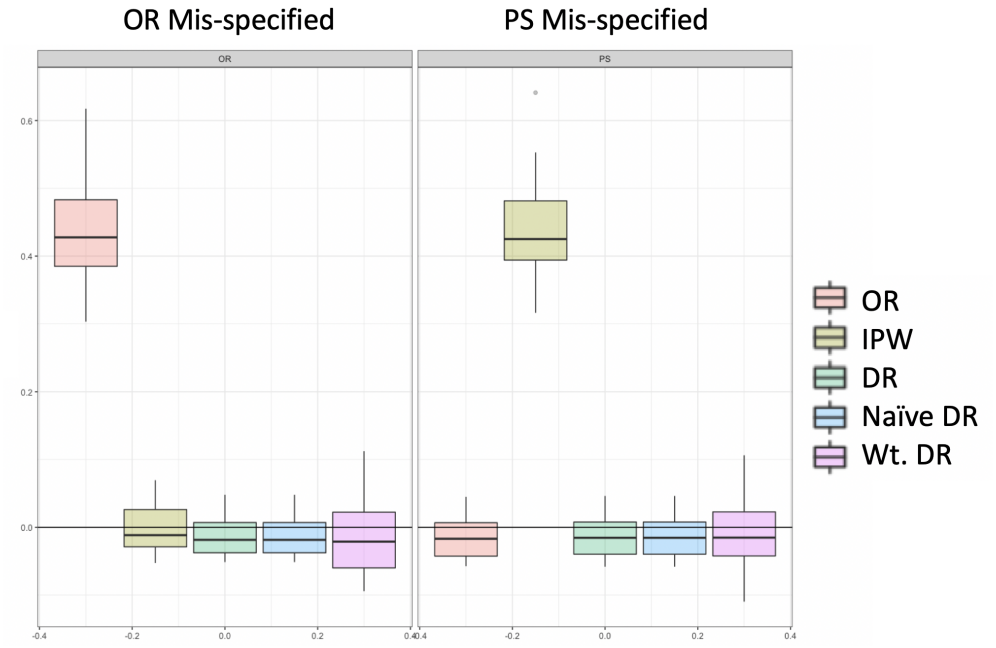

Figure 4: A comparison between five estimators of the TATE, $\widehat{\text{OR}}_r$, $\widehat{\text{IPW}}_r$, $\widehat{\text{DR}}_r$ where $r \in (1, \ldots, K)$, $\widehat{\text{DR}}_r$ where $r \in \mathcal{T}$, and $\widehat{\text{WDR}}_r$, when either the outcome regression or propensity score model is purposefully misspecified by not including covariates used to generate the outcome and propensity score respectively. In this figure, $K = 50, n = 100, p = 100, \beta_A = 10$, and $\mathcal{T}$ is all $K = 50$ sites with no between-site heterogeneity. Note that $\widehat{\text{OR}}_r$ is biased when the outcome regression is misspecified and $\widehat{\text{IPW}}_r$ is biased when the propensity score model is misspecified. The doubly robust models are unbiased as long as at least one of the two models is correctly specified.

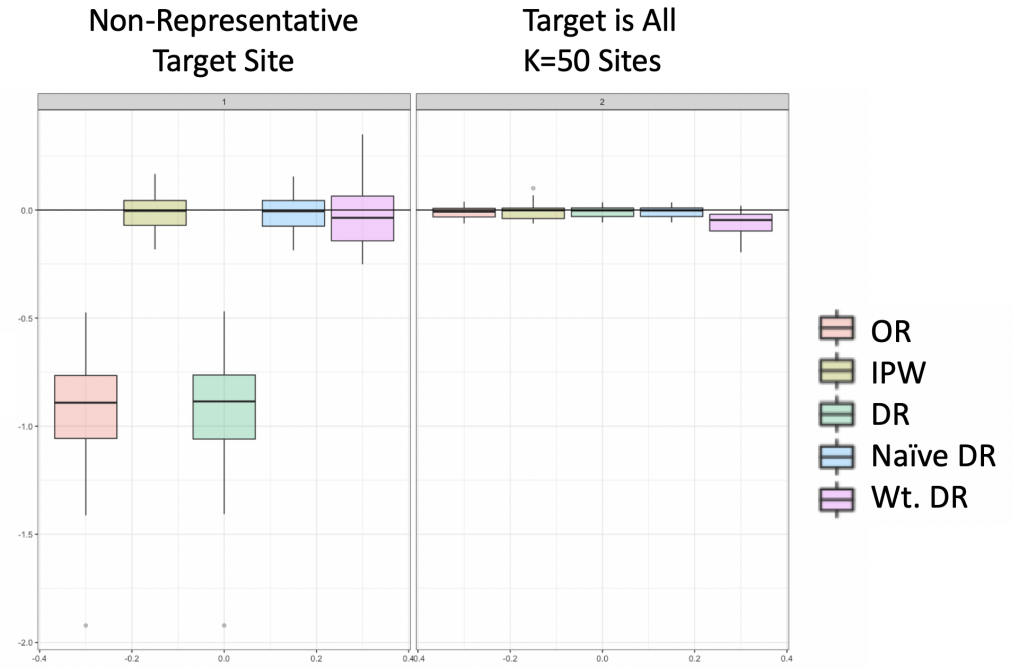

Figure 5: A comparison between five estimators of the TATE, $\widehat{\text{OR}}_r$, $\widehat{\text{IPW}}_r$, $\widehat{\text{DR}}_r$ where $r \in (1, \ldots, K)$, $\widehat{\text{DR}}_r$ where $r \in \mathcal{T}$, and $\widehat{\text{WDR}}_r$, with structured covariate heterogeneity between the five sites in the small target and the other forty-five sites and a covariate-dependent treatment effect. $K = 50, n = 100, p = 100, \beta_A = 10$, and $\mathcal{T}$ is either the five heterogeneous sites or all $K = 50$ sites. Note that $\widehat{\text{OR}}_r$ and $\widehat{\text{DR}}_r$ where $r \in (1, \ldots, K)$ are biased when the target site is not well-represented by the overall study population.

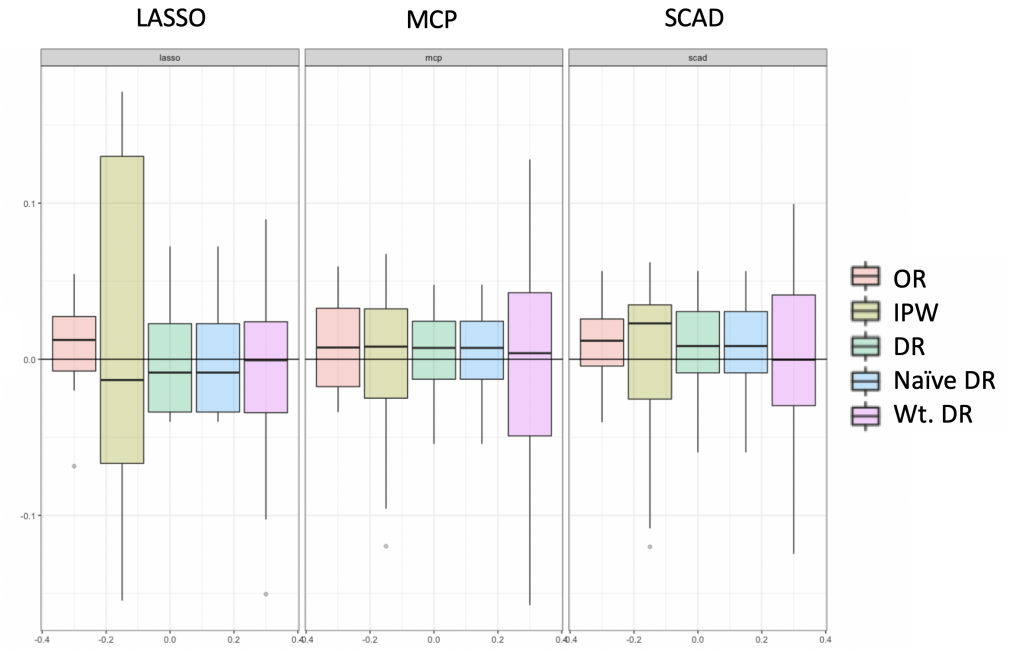

Figure 6: A comparison between five estimators of the TATE, $\widehat{\text{OR}}_r$, $\widehat{\text{IPW}}_r$, $\widehat{\text{DR}}_r$ where $r \in (1, \ldots, K)$, $\widehat{\text{DR}}_r$ where $r \in \mathcal{T}$, and $\widehat{\text{WDR}}_r$, in the high-dimensional setting. Three regularization methods are compared: LASSO, MCP, and SCAD. Here, $K = 50, n = 100, p = 500$ with $p^* = 5$ non-zero entries in the true β vector. Here, $\beta_A = 10$, and $\mathcal{T}$ is all $K = 50$ sites. Note that the LASSO failed to converge in several of the cases in this setting, so extreme outliers were removed. Note that $\widehat{\text{IPW}}_r$ is not robust to misspecification of the propensity score model and the LASSO often chose incorrect covariates in this case, leading to the relatively larger empirical standard error.

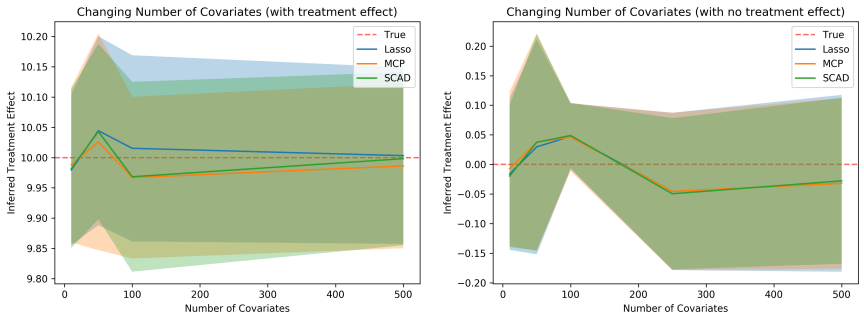

Figure 7: We fixed the data to have 50 sites of 100 patients each. Then we ranged the number of covariates from 10 to 500, with 5 of them set to a nonzero value. The bands are roughly 95% prediction intervals (2 times the sample standard deviation, generated from 20 repeats). For all the number of covariates tested, the algorithm was able to get within 0.2 or so of the correct average treatment effect.

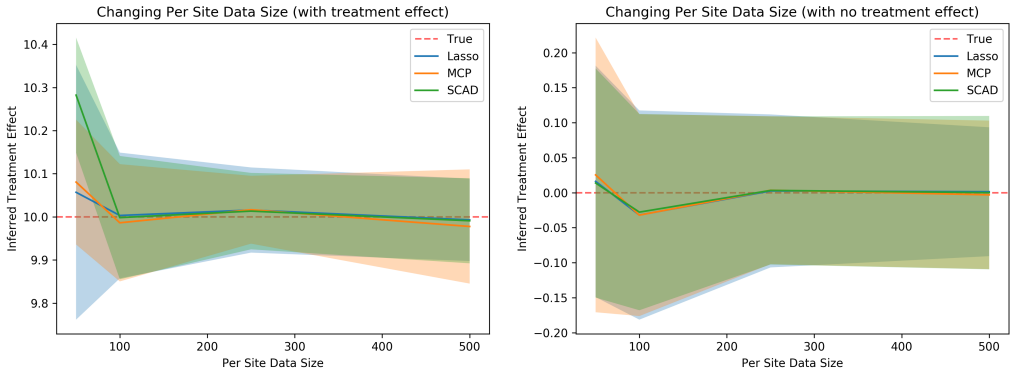

Figure 8: We fixed the data to have 50 sites, where the number of patients at each site varied from 10 to 500. We fixed the number of covariates to 5 significant out of 500. For both with and without a treatment effect, the algorithm was able to get within 0.4 of the correct average treatment effect, with the prediction interval narrowing as the number of patients increases.

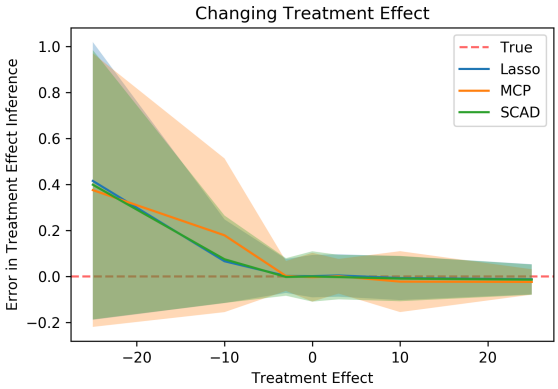

Figure 9: We varied the treatment effect from -25 to 25. We kept the data at 50 sites of 500 people each, and 5 significant covariates out of 500.
The algorithm performs well for positive treatment effects, with a difference between predicted an observed average treatment effect of < 0.2 even for large ATEs. However, these is a surprising asymmetry between the positive and negative treatment effects. Nonetheless, even the largest ATE estimation error of ≈ 1 is still small compared to the large in magnitude value of $ATE = -25$, so the results are not terrible.

p	n	True ATE	Target Sites	OR	IPW	DR	Naive DR	Wt. DR
10	100	0	sample	0.015	0.017	0.015	0.016	-0.001
100	100	0	sample	-0.003	-0.013	-0.003	-0.013	-0.003
500	100	0	sample	-0.005	0.001	-0.005	-0.001	-0.01
10	200	0	sample	-0.002	-0.004	-0.002	-0.004	-0.006
100	200	0	sample	0.006	-0.003	0.006	-0.004	0.002
500	200	0	sample	0.004	-0.002	0.004	-0.003	0.007
10	500	0	sample	0.001	0	0.001	0	-0.001
100	500	0	sample	0.003	0.001	0.004	0.001	0.006
500	500	0	sample	0.01	-0.003	0.01	-0.003	0.009
10	100	10	sample	-0.005	0.019	-0.006	0.023	-0.01
100	100	10	sample	-0.001	0.003	-0.001	0	-0.001
500	100	10	sample	-0.004	0	-0.004	-0.002	-0.01
10	200	10	sample	0.002	-0.005	0.002	-0.006	0
100	200	10	sample	-0.007	0.009	-0.007	0.008	-0.009
500	200	10	sample	-0.006	-0.002	-0.007	-0.001	-0.004
10	500	10	sample	0.001	0.003	0.001	0.002	0.001
100	500	10	sample	0	0.003	0	0.003	-0.001
500	500	10	sample	-0.001	0	0	0.001	-0.002
10	100	0	all	0.001	0.003	0.001	0.001	-0.006
100	100	0	all	-0.002	0.005	-0.003	-0.003	-0.013
500	100	0	all	0.003	0.008	0.002	0.002	0
10	200	0	all	-0.001	0.002	0	0	0
100	200	0	all	0.001	0.003	0.001	0.001	0.004
500	200	0	all	-0.001	0.003	-0.001	-0.001	0.006
10	500	0	all	0	0.002	0	0	-0.004
100	500	0	all	-0.002	-0.001	-0.002	-0.002	0.001
500	500	0	all	0.002	0.001	0.001	0.001	-0.001
10	100	10	all	0.002	0.005	0.003	0.003	0.01
100	100	10	all	-0.004	-0.001	-0.003	-0.003	-0.003
500	100	10	all	0.005	0.007	0.005	0.005	-0.01
10	200	10	all	0	0.001	0.001	0.001	0
100	200	10	all	-0.001	-0.001	-0.001	-0.001	-0.005
500	200	10	all	0.004	0	0.004	0.004	0.006
10	500	10	all	0	-0.001	0.001	0.001	-0.004
100	500	10	all	0.002	0.003	0.002	0.002	0.004
500	500	10	all	0.001	0.002	0.001	0.001	0.005

Table 1: Empirical Bias when all models are correctly specified with no structured between-site heterogeneity. The target sites $\mathcal{T}$ are either a small sample (5 sites) or all sites. Note that in this case, all proposed estimators are unbiased.

True ATE	n_r	Penalty	Sensitivity	Specificity
0	100	LASSO	0.84	0.85
		MCP	0.80	0.98
		SCAD	0.81	0.94
	200	LASSO	0.84	0.86
		MCP	0.80	0.99
		SCAD	0.80	0.97
	500	LASSO	0.83	0.87
		MCP	0.80	0.99
		SCAD	0.80	0.98
10	100	LASSO	1.00	0.84
		MCP	1.00	0.99
		SCAD	1.00	0.98
	200	LASSO	1.00	0.85
		MCP	1.00	0.99
		SCAD	1.00	0.97
	500	LASSO	1.00	0.86
		MCP	1.00	0.99
		SCAD	1.00	0.98

Table 2: The data considered here was generated with 50 sites of n_r people (varying from 100 to 500), each one with 100 covariates $\sim \mathcal{N}(0, 1)$. Treatment effect varied between 0 and 10. There were only 5 non-zero entries in the true β vector (described in Section 4.1). Sensitivity refers to the percentage of correctly identified non-zero β's. Specificity refers to the percentage of correctly identified zero β's.

p	n	True ATE	Reg Penalty	OR	IPW	DR	Naive DR	Wt. DR
100	100	0	lasso	0.067	0.066	0.07	0.07	0.041
500	100	0	lasso	0.071	0.059	0.086	0.086	0.083
100	200	0	lasso	0.065	0.062	0.063	0.063	0.03
500	200	0	lasso	0.098	0.097	0.097	0.097	0.048
100	100	10	lasso	-0.014	0.001	-0.008	-0.008	0.01
500	100	10	lasso	-0.009	0.004	-0.012	-0.012	-0.012
100	200	10	lasso	0.011	0.016	0.012	0.012	0.009
500	200	10	lasso	-0.005	0.015	-0.004	-0.004	0.018
100	100	0	mcp	0.108	0.107	0.107	0.107	0.09
500	100	0	mcp	0.085	0.09	0.086	0.086	0.042
100	200	0	mcp	0.082	0.087	0.081	0.081	0.061
500	200	0	mcp	0.099	0.099	0.098	0.098	0.084
100	100	10	mcp	-0.008	-0.006	-0.006	-0.006	-0.016
500	100	10	mcp	-0.006	-0.002	-0.007	-0.007	-0.014
100	200	10	mcp	0.012	0.013	0.012	0.012	0.009
500	200	10	mcp	-0.004	-0.011	-0.005	-0.005	-0.011
100	100	0	scad	0.097	0.087	0.097	0.097	0.069
500	100	0	scad	0.09	0.087	0.092	0.092	0.043
100	200	0	scad	0.083	0.089	0.082	0.082	0.064
500	200	0	scad	0.097	0.099	0.096	0.096	0.07
100	100	10	scad	-0.009	-0.007	-0.008	-0.008	-0.011
500	100	10	scad	-0.004	-0.004	-0.005	-0.005	0.005
100	200	10	scad	0.011	0.015	0.012	0.012	0.011
500	200	10	scad	-0.004	-0.011	-0.005	-0.005	-0.008

Table 3: Empirical Bias of each estimator in the high-dimensional setting across 20 simulations, with $\mathcal{T}$ as all $K = 50$ sites.

p	n	True ATE	Reg Penalty	OR	IPW	DR	Naive DR	Wt. DR
100	100	0	lasso	0.066	0.082	0.08	0.08	0.081
500	100	0	lasso	0.051	0.09	0.059	0.059	0.089
100	200	0	lasso	0.047	0.054	0.047	0.047	0.088
500	200	0	lasso	0.03	0.056	0.033	0.033	0.073
100	100	10	lasso	0.033	0.103	0.059	0.059	0.074
500	100	10	lasso	0.042	0.127	0.05	0.05	0.092
100	200	10	lasso	0.021	0.039	0.022	0.022	0.043
500	200	10	lasso	0.019	0.086	0.017	0.017	0.079
100	100	0	mcp	0.053	0.061	0.057	0.057	0.084
500	100	0	mcp	0.035	0.044	0.038	0.038	0.089
100	200	0	mcp	0.034	0.039	0.035	0.035	0.056
500	200	0	mcp	0.029	0.034	0.029	0.029	0.053
100	100	10	mcp	0.031	0.047	0.03	0.03	0.055
500	100	10	mcp	0.036	0.061	0.035	0.035	0.082
100	200	10	mcp	0.021	0.028	0.021	0.021	0.049
500	200	10	mcp	0.018	0.024	0.018	0.018	0.024
100	100	0	scad	0.056	0.077	0.06	0.06	0.099
500	100	0	scad	0.037	0.077	0.04	0.04	0.049
100	200	0	scad	0.034	0.036	0.035	0.035	0.066
500	200	0	scad	0.029	0.032	0.029	0.029	0.06
100	100	10	scad	0.031	0.049	0.03	0.03	0.055
500	100	10	scad	0.035	0.061	0.034	0.034	0.09
100	200	10	scad	0.021	0.031	0.022	0.022	0.049
500	200	10	scad	0.019	0.025	0.019	0.019	0.028

Table 4: Empirical standard error of each estimator in the high-dimensional setting across 20 simulations, with $\mathcal{T}$ as all $K = 50$ sites. Note that when $n = 100$ and the True ATE is large $\beta_A = 10$, the LASSO often chooses many extraneous covariates, causing the propensity score model to be more biased and thus increasing the empirical standard error of the Inverse Probability Weight estimator

7 Code

The code for this project is uploaded at `https://github.com/azswartz/Stat236-Final`

References

Battey, H., J. Fan, H. Liu, J. Lu, and Z. Zhu (2018). Distributed testing and estimation under sparse high dimensional models. *Annals of statistics 46*(3), 1352.

Brat, G. A., G. M. Weber, N. Gehlenborg, P. Avillach, N. P. Palmer, L. Chiovato, J. Cimino, L. R. Waitman, G. S. Omenn, A. Malovini, et al. (2020). International electronic health record-derived covid-19 clinical course profiles: the 4ce consortium. *medRxiv*.

Breheny, P. and J. Huang (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics 5*(1), 232–253.

Chen, Y., G. Dong, J. Han, J. Pei, B. W. Wah, and J. Wang (2006). Regression cubes with lossless compression and aggregation. *IEEE Transactions on Knowledge and Data Engineering 18*(12), 1585–1599.

Duan, R., M. R. Boland, Z. Liu, Y. Liu, H. H. Chang, H. Xu, H. Chu, C. H. Schmid, C. B. Forrest, J. H. Holmes, et al. (2020). Learning from electronic health records across multiple sites: A communication-efficient and privacy-preserving distributed algorithm. *Journal of the American Medical Informatics Association 27*(3), 376–385.

Duan, R., C. Luo, M. H. Schuemie, J. Tong, J. C. Liang, H. H. Chang, M. R. Boland, J. Bian, H. Xu, J. H. Holmes, et al. (2020). Learning from local to global-an efficient distributed algorithm for modeling time-to-event data. *bioRxiv*.

Friedman, C. P., A. K. Wong, and D. Blumenthal (2010). Achieving a nationwide learning health system. *Science translational medicine 2*(57), 57cm29–57cm29.

Ge, J., X. Li, H. Jiang, H. Liu, T. Zhang, M. Wang, and T. Zhao (2019). Picasso: A sparse learning library for high dimensional data analysis in r and python. *J. Mach. Learn. Res. 20*, 44–1.

Hripcsak, G., P. B. Ryan, J. D. Duke, N. H. Shah, R. W. Park, V. Huser, M. A. Suchard, M. J. Schuemie, F. J. DeFalco, A. Perotte, et al. (2016). Characterizing treatment pathways at scale using the ohdsi network. *Proceedings of the National Academy of Sciences 113*(27), 7329–7336.

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

Jordan, M. I., J. D. Lee, and Y. Yang (2018). Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*.

Lee, J. D., Q. Liu, Y. Sun, and J. E. Taylor (2017). Communication-efficient sparse regression. *The Journal of Machine Learning Research 18*(1), 115–144.

Lu, C.-L., S. Wang, Z. Ji, Y. Wu, L. Xiong, X. Jiang, and L. Ohno-Machado (2015). Webdisco: a web service for distributed cox model learning without patient-level data sharing. *Journal of the American Medical Informatics Association 22*(6), 1212–1219.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc.

Payne, T. H., S. Corley, T. A. Cullen, T. K. Gandhi, L. Harrington, G. J. Kuperman, J. E. Mattison, D. P. McCallie, C. J. McDonald, P. C. Tang, et al. (2015). Report of the amia ehr-2020 task force on the status and future direction of ehrs. *Journal of the American Medical Informatics Association 22*(5), 1102–1110.

Qin, J. (1998). Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika 85*(3), 619–630.

Robins, J. M., A. Rotnitzky, and L. P. Zhao (1994). Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association 89*(427), 846–866.

Rubin, D. B. (2005). Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association 100*(469), 322–331.

Tan, Z. (2020). Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data. *Biometrika 107*(1), 137–158.

Tan, Z. et al. (2020). Model-assisted inference for treatment effects using regularized calibrated estimation with high-dimensional data. *Annals of Statistics 48*(2), 811–837.

Wu, Y., X. Jiang, J. Kim, and L. Ohno-Machado (2012). Grid binary logistic regression (glore): Building shared models without sharing data. *Journal of the American Medical Informatics Association 19*(5), 758–764.