

# ECMA 31380 - Causal Machine Learning - Homework 4

Fernando Rocha Urbano

Autumn 2024

**Attention:** all code is available in

<https://github.com/Fernando-Urbano/causal-machine-learning/tree/main/hw4>.

---

## 1 Conditions on Nonparametric Estimators

We are studying the impact of a multi-valued treatment  $T \in \{0, 1, \dots, T\}$ , for some integer  $T$ , on an outcome  $Y$ . We observe  $Z = (Y, T, \mathbf{X})' \in \mathbb{R} \times \{0, 1, \dots, T\} \times \mathbb{R}^d$ . Define the potential outcomes as  $Y(t)$ , the propensity score  $p_t(\mathbf{x}) := \mathbb{P}[T = t \mid \mathbf{X} = \mathbf{x}]$ , and the regression functions  $\mu_t(x) = \mathbb{E}[Y(t) \mid \mathbf{X} = \mathbf{x}]$ .

Interesting estimands can be built from averages of  $\mu_t(\mathbf{x})$ . For example: the ATE of treatment level  $t$  is  $\tau_t = \mathbb{E}[\mu_t(\mathbf{X}) - \mu_0(\mathbf{X})]$ . If the treatment is a dose, then the effect of increasing the dose from  $t$  to  $t + 1$  is  $\mathbb{E}[\mu_{t+1}(\mathbf{X}) - \mu_t(\mathbf{X})]$ . And so on. So we will study  $\mu_t = \mathbb{E}[\mu_t(\mathbf{X})] = \mathbb{E}[Y(t)]$ . Suppose that  $p_t(x) = p_t$  is constant over  $x$ , so that this is a randomized experiment.

---

### 1.a Linear Regression Model and Assumptions

Provide a single linear regression model that yields identification of all  $\mu_t$ ,  $t \in \{0, \dots, T\}$ . What assumptions do you need? Describe the estimators  $\hat{\mu}$ . Provide regularity conditions so that the vector  $\hat{\mu}$  is asymptotically Normal, asymptotically unbiased, and characterize the asymptotic variance.

---

Consider the linear model:

$$Y = \sum_{t=0}^T \alpha_t \mathbf{1}\{T = t\} + \varepsilon,$$

where  $\mathbf{1}\{T = t\}$  is the indicator function that takes the value 1 if  $T = t$  and 0 otherwise.

Identification of the parameters  $\mu_t$  follows from:

$$\mu_t = \mathbb{E}[Y(t)] = \alpha_t.$$

Since we assume that the treatment assignment is independent of  $\mathbf{X}$  ( $p_t(\mathbf{x}) = p_t$  is constant), this implies that

$$\mathbb{E}[\varepsilon \mid T = t, \mathbf{X}] = 0.$$

Under these assumptions, the OLS estimators

$$\hat{\alpha}_t = \frac{1}{n_t} \sum_{i:T_i=t} Y_i,$$

where  $n_t = \sum_{i=1}^n \mathbf{1}\{T_i = t\}$ , are unbiased and consistent for  $\alpha_t$ . Hence,

$$\hat{\mu}_t = \hat{\alpha}_t.$$

To characterize the asymptotic behavior, let  $\hat{\mu} = (\hat{\mu}_0, \hat{\mu}_1, \dots, \hat{\mu}_T)'$  and  $\mu = (\mu_0, \mu_1, \dots, \mu_T)'$ . Under standard regularity conditions, including:

- Finite second moments:  $\mathbb{E}[Y(t)^2] < \infty$  for all  $t$ .
- The proportions  $p_t = \mathbb{P}(T = t)$  are fixed and strictly positive.
- Independence of treatment and potential outcomes:  $(Y(t))_{t=0}^T \perp T$ .

we have by the Central Limit Theorem:

$$\sqrt{n}(\hat{\mu} - \mu) \xrightarrow{d} N(0, \Sigma),$$

where  $\Sigma$  is a  $(T+1) \times (T+1)$  diagonal matrix given by

$$\Sigma = \text{diag} \left( \frac{\sigma_0^2}{p_0}, \frac{\sigma_1^2}{p_1}, \dots, \frac{\sigma_T^2}{p_T} \right),$$

and  $\sigma_t^2 = \text{Var}(Y(t))$ . Thus, each  $\hat{\mu}_t$  is asymptotically Normal and asymptotically unbiased with asymptotic variance  $\sigma_t^2/p_t$ .

The linear model above combined with the given assumptions and regularity conditions ensures that the estimators  $\hat{\mu}_t$  are consistent, asymptotically Normal, and asymptotically unbiased, and that the asymptotic variance is as described.

## 1.b Sufficient Conditions for Identification

Now suppose that  $p_t(\mathbf{x})$  is not constant. Provide sufficient conditions so that  $\mu_t$  is identified. Compare these conditions to what you found above.

For this question, we consider the following assumptions:

$$(Y(0), Y(1), \dots, Y(T)) \perp T \mid \mathbf{X}.$$

This condition, generally called CIA, ensures that the treatment assignment is independent of the potential outcomes once we condition on the covariates  $\mathbf{X}$ .

$$0 < p_t(\mathbf{x}) = \mathbb{P}(T = t \mid \mathbf{X} = \mathbf{x}) < 1 \text{ for all } t \in \{0, \dots, T\} \text{ and almost every } \mathbf{x}.$$

This positivity (overlap) condition ensures that every treatment arm has a nonzero probability of being assigned at each value of  $\mathbf{X}$ . It rules out degenerate cases where certain treatments never occur for some subsets of  $\mathbf{X}$ .

Given these assumptions, we have that

$$\mu_t = \mathbb{E}[Y(t)] = \int \mathbb{E}[Y \mid T = t, \mathbf{X} = \mathbf{x}] f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}.$$

By the unconfoundedness assumption,

$$\mathbb{E}[Y(t) \mid \mathbf{X} = \mathbf{x}] = \mathbb{E}[Y \mid T = t, \mathbf{X} = \mathbf{x}],$$

and by the law of total expectation,

$$\mu_t = \int \mathbb{E}[Y \mid T = t, \mathbf{X} = \mathbf{x}] f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}.$$

Since both  $\mathbb{E}[Y \mid T = t, \mathbf{X} = \mathbf{x}]$  and  $f_{\mathbf{X}}(\mathbf{x})$  can be identified from the data (with correct specification of the model and enough data),  $\mu_t$  is identified.

Comparing this to the case where  $p_t(\mathbf{x}) = p_t$  is constant, we previously needed only unconditional independence of treatment assignment. In that case, the identification was straightforward since no conditioning on  $\mathbf{X}$  was required.

Here, when  $p_t(\mathbf{x})$  is not constant, we need conditional independence and overlap conditions to ensure that each  $\mu_t$  is identified by integrating out the covariates  $\mathbf{X}$ .

Identification no longer comes from a simple randomization structure alone.

---

In class, we studied nonparametric regression using piecewise polynomials of degree  $p$  (fixed) and  $J = J_n \rightarrow \infty$  pieces and proved that the  $L_2$  convergence rate is (using the notation of the present context)

$$\|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2^2 = O_p\left(\frac{J^d}{n} + J^{-2(p+1)}\right).$$

Let the number of bins be  $J = Cn^\gamma$  for some constants (positive)  $C$  and  $\gamma$ . We will ignore  $C$  and focus on rates here.

First we study nonparametric estimation and inference.

### 1.c Range of $\gamma$ for Consistency

For what range of  $\gamma$  is  $\hat{\mu}_t(\mathbf{x})$  consistent? How does this range depend on the dimension and the polynomial order? Are there values of  $p$  and  $d$  such that this interval is empty?

---

For the given rate

$$\|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2^2 = O_p \left( \frac{J^d}{n} + J^{-2(p+1)} \right),$$

we substitute  $J = n^\gamma$  (ignoring the constant  $C$ ). Thus the rate becomes

$$\|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2^2 = O_p \left( n^{\gamma d - 1} + n^{-2\gamma(p+1)} \right).$$

For consistency, the entire expression should go to zero as  $n \rightarrow \infty$ . This requires that each exponent be negative:

$$\gamma d - 1 < 0 \implies \gamma < \frac{1}{d},$$

and

$$-2\gamma(p+1) < 0 \implies \gamma > 0.$$

Combining these two conditions, the parameter  $\gamma$  must lie in the interval

$$0 < \gamma < \frac{1}{d}.$$

This shows that the dimension  $d$  directly affects the range for  $\gamma$ .

As  $d$  increases, the upper bound  $1/d$  decreases: this makes it harder to find a  $\gamma$  that achieves consistency.

The polynomial order  $p$  affects the rate at which the second term vanishes but does not influence the existence of the interval  $(0, 1/d)$ .

We can see that, as long as  $\gamma > 0$ , the term  $n^{-2\gamma(p+1)}$  goes to zero. Thus, no matter the polynomial order  $p$ , there will always be some  $\gamma$  in  $(0, 1/d)$  for consistency. In conclusion, the interval is never empty for any fixed positive integer  $d$ .

## 1.d Optimal Value of $\gamma$

What value of  $\gamma$  is optimal in the sense that the rate is the fastest? Call this  $\gamma_{\text{mse}}^*$ . How does  $\gamma_{\text{mse}}^*$  vary with the dimension and the polynomial order?

Consider the rate

$$\|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2^2 = O_p \left( n^{\gamma d - 1} + n^{-2\gamma(p+1)} \right).$$

To find the optimal rate in terms of order, we choose  $\gamma$  to balance the two terms. Set

$$n^{\gamma d - 1} = n^{-2\gamma(p+1)}.$$

Taking logs, we have

$$\gamma d - 1 = -2\gamma(p+1).$$

$$\gamma d + 2\gamma(p+1) = 1,$$

$$\gamma(d + 2(p + 1)) = 1,$$

$$\gamma = \frac{1}{d + 2(p + 1)}.$$

Call this value  $\gamma_{\text{mse}}^*$ . It is the  $\gamma$  that equates the rates of the bias and variance terms. It optimizes the mean squared error rate.

This  $\gamma_{\text{mse}}^*$  depends on both the dimension  $d$  and the polynomial order  $p$  as follows:

$$\gamma_{\text{mse}}^* = \frac{1}{d + 2(p + 1)}.$$

As the dimension  $d$  increases, the denominator increases, making  $\gamma_{\text{mse}}^*$  smaller.

Increasing the polynomial order  $p$  also increases the denominator, leading to a smaller  $\gamma_{\text{mse}}^*$ .

Thus, higher dimensionality or smoother approximations (larger  $p$ ) both lead to a smaller optimal  $\gamma$ .

### 1.e Asymptotic Normality of $\hat{\mu}_t(x)$

For what range of  $\gamma$  is  $\hat{\mu}_t(\mathbf{x})$  asymptotically Normal when properly centered and scaled? That is, determine the range for  $\gamma$  such that

$$\sqrt{n/J^d}(\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})) \xrightarrow{d} \mathcal{N}(0, V).$$

(Don't worry about quantifying  $V$ ). How does this range depend on the dimension and the polynomial order? Are there values of  $p$  and  $d$  such that this interval is empty?

Substituting  $J = n^\gamma$ , we have  $J^d = n^{\gamma d}$  and thus

$$\sqrt{\frac{n}{J^d}} = n^{\frac{1}{2} - \frac{\gamma d}{2}}.$$

The asymptotic Normality with a non-degenerate limit requires that the bias be negligible relative to the chosen scaling. The bias is of order

$$J^{-(p+1)} = n^{-\gamma(p+1)},$$

so under the scaling we have

$$n^{-\gamma(p+1)} \cdot n^{\frac{1}{2} - \frac{\gamma d}{2}} = n^{\frac{1 - \gamma d}{2} - \gamma(p+1)}.$$

For the bias to vanish under this scaling, we need

$$\frac{1 - \gamma d}{2} - \gamma(p + 1) < 0.$$

Rearranging,

$$1 - \gamma d < 2\gamma(p+1) \implies 1 < \gamma(d + 2(p+1)) \implies \gamma > \frac{1}{d + 2(p+1)}.$$

Additionally, for a central limit theorem to apply to the binned means, the number of observations per bin  $n/J^d = n^{1-\gamma d}$  must tend to infinity. This gives

$$1 - \gamma d > 0 \implies \gamma < \frac{1}{d}.$$

Combining these two inequalities, we obtain the range for  $\gamma$ :

$$\frac{1}{d + 2(p+1)} < \gamma < \frac{1}{d}.$$

As the dimension  $d$  or the polynomial order  $p$  increases, the lower bound  $1/(d + 2(p+1))$  moves closer to zero, and the upper bound  $1/d$  decreases.

This leads us to conclude that the interval becomes narrower when either  $d$  or  $p$  is large, but it does not vanish. For all positive  $p$  and  $d$ , the interval for  $\gamma$  is never empty because  $d + 2(p+1) > d$  always.

## 1.f Range of $\gamma$ for Optimal Rate $\gamma_{\text{mse}}^*$

Is  $\gamma_{\text{mse}}^*$  in this range?

We saw that:

$$\gamma_{\text{mse}}^* = \frac{1}{d + 2(p+1)},$$

and the range for asymptotic Normality was:

$$\frac{1}{d + 2(p+1)} < \gamma < \frac{1}{d}.$$

Since  $\gamma_{\text{mse}}^* = \frac{1}{d+2(p+1)}$  is exactly at the lower boundary, it is not strictly within the interval.

Therefore,  $\gamma_{\text{mse}}^*$  is not in the open range required for asymptotic Normality.

The result is similar to the one found in a previous homework in which we review the optimal  $h$  for which  $f(x+h)$  is asymptotically normal and optimal  $h$  for which  $f(x+h)$  is optimal in terms of inference: like this, the  $h$  to improve inference is just outside the bound of asymptotically normality.

---

Now we study semiparametric estimation and inference.

---

## 1.g Semiparametric Estimation and Inference

In class, we showed that there was a problem with the two-step plug-in estimator  $\tilde{\mu}_t = \frac{1}{n} \sum_{i=1}^n \hat{\mu}(\mathbf{x}_i)$  and that it did not have the same influence function as the parametric regression-based plug-in estimator. However, Cattaneo and Farrell (2011) showed that it does in fact obtain an influence function representation, with the familiar influence function. That paper shows that if

$$\sqrt{n} \left( \frac{J^d}{n} + J^{-(p+1)} \right) \rightarrow 0$$

then

$$\sqrt{n}(\tilde{\mu}_t - \mathbb{E}[Y(t)]) = \frac{1}{\sqrt{n}} \sum_{i=1}^n \psi_i + o_p(1) \xrightarrow{d} \mathcal{N}(0, \mathbb{E}[\psi_t(Z)^2]),$$

where  $\psi_t(\mathbf{z}_i) = \mu(\mathbf{x}_i) - \mathbb{E}[Y(t)] + \mathbb{I}\{t_i = t\}(y_i - \mu(\mathbf{x}_i))/p_t(\mathbf{x}_i)$ .

- (i) For what range of  $\gamma$  is inference on the  $\mathbb{E}[Y(t)]$  possible? How does this range depend on  $p$  and  $d$ ? Are there values of  $p$  and  $d$  such that this interval is empty?
  - (ii) Is  $\gamma_{\text{mse}}^*$  in this range?
- 

Substitute  $J = n^\gamma$ . Then  $J^d = n^{\gamma d}$  and  $J^{-(p+1)} = n^{-\gamma(p+1)}$ , giving

$$\sqrt{n} \left( n^{\gamma d - 1} + n^{-\gamma(p+1)} \right) = n^{\gamma d - \frac{1}{2}} + n^{\frac{1}{2} - \gamma(p+1)}.$$

For these terms to vanish as  $n \rightarrow \infty$ , each exponent must be negative:

$$\gamma d - \frac{1}{2} < 0 \implies \gamma < \frac{1}{2d},$$

and

$$\frac{1}{2} - \gamma(p+1) < 0 \implies \gamma > \frac{1}{2(p+1)}.$$

Combining these inequalities, we find that for inference on  $\mathbb{E}[Y(t)]$  to be possible via the plug-in estimator:

$$\frac{1}{2(p+1)} < \gamma < \frac{1}{2d}.$$

The range for  $\gamma$  depends on both  $p$  and  $d$ . As  $d$  increases, the upper bound  $1/(2d)$  decreases, while as  $p$  increases, the lower bound  $1/(2(p+1))$  decreases. If the dimension  $d$  is large compared to the polynomial order  $p$ , it is possible for the interval

$$\left( \frac{1}{2(p+1)}, \frac{1}{2d} \right)$$

to be empty. Specifically, the interval is non-empty if and only if

$$\frac{1}{2(p+1)} < \frac{1}{2d} \implies d < p+1.$$

When  $p$  is sufficiently large relative to  $d$  ( $p \geq d$ ), there will always be some values of  $\gamma$  for which inference is possible.

When  $p$  is too small relative to  $d$ , the interval may be empty, and no such  $\gamma$  will exist.

In the previous question we arrived to:

$$\gamma_{\text{mse}}^* = \frac{1}{d + 2(p+1)},$$

and, again from the previous questions, the range of  $\gamma$  that allows inference on  $\mathbb{E}[Y(t)]$  is

$$\frac{1}{2(p+1)} < \gamma < \frac{1}{2d}.$$

Compare  $\gamma_{\text{mse}}^*$  to the lower bound  $1/(2(p+1))$ :

$$\gamma_{\text{mse}}^* = \frac{1}{d + 2(p+1)} \quad \text{and} \quad \frac{1}{2(p+1)}.$$

Since  $d > 0$ , we have  $d + 2(p+1) > 2(p+1)$ .

Therefore:

$$\frac{1}{d + 2(p+1)} < \frac{1}{2(p+1)},$$

meaning:

$$\gamma_{\text{mse}}^* < \frac{1}{2(p+1)}.$$

Because the allowed range for inference is  $\gamma > 1/(2(p+1))$ , and  $\gamma_{\text{mse}}^*$  is strictly less than  $1/(2(p+1))$ , we can check that  $\gamma_{\text{mse}}^*$  does not lie in the interval  $(1/(2(p+1)), 1/(2d))$ .

Therefore,  $\gamma_{\text{mse}}^*$  is not in the range that allows for inference on  $\mathbb{E}[Y(t)]$ .



## 1.h Influence Function-Based Estimator

Now consider the influence function-based estimator. Let  $\hat{\mu}_t(\mathbf{x})$  and  $\hat{p}_t(\mathbf{x})$  be partitioning-based estimators of the respective functions, which both have the rate of Equation (1). Define

$$\hat{\mu}_t = \frac{1}{n} \sum_{i=1}^n \left\{ \hat{\mu}_t(x_i) + \frac{\mathbb{I}\{t_i = t\}(y_i - \hat{\mu}_t(x_i))}{\hat{p}_t(x_i)} \right\}.$$

In class, we proved that the linear representation and asymptotic normality of Equation (3) holds (with  $\tilde{\mu}_t$  replaced by  $\hat{\mu}_t$ ) if

$$\|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2 \rightarrow 0, \quad \|\hat{p}_t(\mathbf{x}) - p_t(\mathbf{x})\|_2 \rightarrow 0, \quad \text{and} \quad \sqrt{n}\|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2 \|\hat{p}_t(\mathbf{x}) - p_t(\mathbf{x})\|_2 \rightarrow 0.$$

- (i) For what range of  $\gamma$  is inference on the  $\mathbb{E}[Y(t)]$  possible? How does this range depend on  $p$  and  $d$ ? Are there values of  $p$  and  $d$  such that this interval is empty?
- (ii) Is  $\gamma_{\text{mse}}^*$  in this range?
- (iii) In terms of the allowed  $\gamma$ , compare your findings to the previous part.

(i) Consider the conditions for the influence function-based estimator. Both  $\hat{\mu}_t(\mathbf{x})$  and  $\hat{p}_t(\mathbf{x})$  have the same rate:

$$\|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2^2 = O_p \left( \frac{J^d}{n} + J^{-2(p+1)} \right).$$

Substitute  $J = n^\gamma$ . Then

$$\|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2^2 = O_p \left( n^{\gamma d - 1} + n^{-2\gamma(p+1)} \right).$$

The same rate holds for  $\|\hat{p}_t(\mathbf{x}) - p_t(\mathbf{x})\|_2^2$ :

$$\|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2 = O_p \left( \sqrt{n^{\gamma d - 1} + n^{-2\gamma(p+1)}} \right),$$

and for  $\|\hat{p}_t(\mathbf{x}) - p_t(\mathbf{x})\|_2$ .

The conditions for the asymptotic normality of the influence function-based estimator require that:

$$\|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2 \rightarrow 0 \quad \text{and} \quad \|\hat{p}_t(\mathbf{x}) - p_t(\mathbf{x})\|_2 \rightarrow 0.$$

As before, this implies

$$0 < \gamma < \frac{1}{d}.$$

And the key additional requirement is:

$\sqrt{n}\|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2 \|\hat{p}_t(\mathbf{x}) - p_t(\mathbf{x})\|_2 \rightarrow 0$ . Since both norms share the same order, let  $\delta_n = \|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2$ . Then  $\|\hat{p}_t(\mathbf{x}) - p_t(\mathbf{x})\|_2 = O_p(\delta_n)$  and

$$\sqrt{n}\|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2 \|\hat{p}_t(\mathbf{x}) - p_t(\mathbf{x})\|_2 = \sqrt{n}\delta_n^2.$$

From those:

$$\begin{aligned}\delta_n^2 &= O_p(n^{\gamma d-1} + n^{-2\gamma(p+1)}). \\ \sqrt{n}\delta_n^2 &= O_p(n^{\gamma d-\frac{1}{2}} + n^{\frac{1}{2}-2\gamma(p+1)}).\end{aligned}$$

For this to vanish:

$$\gamma d - \frac{1}{2} < 0 \implies \gamma < \frac{1}{2d},$$

and

$$\frac{1}{2} - 2\gamma(p+1) < 0 \implies \gamma > \frac{1}{4(p+1)}.$$

Combining conditions:

$$\frac{1}{4(p+1)} < \gamma < \frac{1}{2d}$$

and also  $\gamma < 1/d$ . Since  $1/(2d) < 1/d$ , the effective upper bound is  $1/(2d)$ . Thus, the range of  $\gamma$  for which inference is possible is

$$\frac{1}{4(p+1)} < \gamma < \frac{1}{2d}.$$

The interval depends on  $p$  and  $d$ .

As  $d$  increases,  $1/(2d)$  decreases, and as  $p$  increases,  $1/(4(p+1))$  decreases. The interval is non-empty if and only if

$$\frac{1}{4(p+1)} < \frac{1}{2d} \implies d < 2(p+1).$$

If  $d \geq 2(p+1)$ , the interval is empty, and no  $\gamma$  satisfies the conditions.

(ii) Recall  $\gamma_{\text{mse}}^* = \frac{1}{d+2(p+1)}$ .

Thus, is  $\gamma_{\text{mse}}^*$  inside  $(1/(4(p+1)), 1/(2d))$ ?

Compare  $\gamma_{\text{mse}}^*$  with  $1/(4(p+1))$ :

$$\frac{1}{d+2(p+1)} > \frac{1}{4(p+1)} \iff 4(p+1) > d+2(p+1) \iff d < 2(p+1).$$

If  $d < 2(p+1)$ , then  $\gamma_{\text{mse}}^* > 1/(4(p+1))$ .

Next, compare  $\gamma_{\text{mse}}^*$  with  $1/(2d)$ : Since  $d+2(p+1) > 2d$  if and only if  $2(p+1) > d$ , and we are in the case  $d < 2(p+1)$ , we have

$$\frac{1}{d+2(p+1)} < \frac{1}{2d}.$$

If  $d < 2(p+1)$ ,  $\gamma_{\text{mse}}^*$  also satisfies  $\gamma_{\text{mse}}^* < 1/(2d)$ .

Therefore, when  $d < 2(p+1)$ ,

$$\frac{1}{4(p+1)} < \gamma_{\text{mse}}^* < \frac{1}{2d},$$

meaning  $\gamma_{\text{mse}}^*$  is inside the allowed range for inference. If  $d \geq 2(p+1)$ , then no  $\gamma$  satisfies the conditions, including  $\gamma_{\text{mse}}^*$ .

(iii) Previously, for the two-step plug-in estimator, the condition for inference was

$$\frac{1}{2(p+1)} < \gamma < \frac{1}{2d}.$$

Now, for the influence function-based estimator, the condition is

$$\frac{1}{4(p+1)} < \gamma < \frac{1}{2d}.$$

The influence function-based estimator relaxes the lower bound from  $1/(2(p+1))$  to  $1/(4(p+1))$ .

This enlarged feasible range makes it easier to satisfy the asymptotic normality conditions.

In particular, for given  $p$  and  $d$ , it may now be possible to select a  $\gamma$  that achieves both optimal MSE and valid inference, a scenario that was more restrictive under the two-step plug-in approach.

## 2 Propensity Score Weighting & ATT Estimation

*This is a continuation from homeworks 2 & 3.*

Assume that the random variables  $(Y_1, Y_0, T, \mathbf{X}')' \in \mathbb{R} \times \mathbb{R} \times \{0, 1\} \times \mathbb{R}^d$  obey  $\{Y_1, Y_0\} \perp T \mid \mathbf{X}$ . The researcher observes  $(Y, T, \mathbf{X})'$ , where  $Y = Y_1T + Y_0(1 - T)$ . Define the propensity score  $p(\mathbf{x}) = \mathbb{P}[T = 1 \mid \mathbf{X} = \mathbf{x}]$  and assume it is bounded inside  $(0, 1)$ . Define  $\mu_t = \mathbb{E}[Y(t) \mid T = 1]$  and  $\mu_t(\mathbf{x}) = \mathbb{E}[Y(t) \mid \mathbf{X} = \mathbf{x}]$ . The average treatment effect on the treated (ATT) is  $\tau = \mu_1 - \mu_0$ .

In homework 3, you studied a “plug-in” estimator of the ATT given by

$$\hat{\tau}_{\text{PI}} = \hat{\mu}_1 - \hat{\mu}_0 = \frac{1}{n} \sum_{i=1}^n \frac{t_i y_i}{\hat{p}} - \frac{1}{n} \sum_{i=1}^n \frac{(1 - t_i) \hat{p}(\mathbf{x}_i) y_i}{(1 - \hat{p}(\mathbf{x}_i))}.$$

In homework 2, you proved that

$$\mu_0 = \frac{1}{\mathbb{E}[T]} \mathbb{E} \left[ T \mu_0(\mathbf{X}) + \frac{(1 - T) p(\mathbf{X}) (Y - \mu_0(\mathbf{X}))}{(1 - p(\mathbf{X}))} \right]$$

and that this moment condition is doubly robust. This motivates a doubly robust estimator of the ATT given by

$$\hat{\tau}_{\text{DR}} = \hat{\mu}_1 - \hat{\mu}_0 = \frac{1}{n} \sum_{i=1}^n \left\{ \frac{t_i y_i}{\hat{p}} \right\} - \frac{1}{\hat{p}} \frac{1}{n} \sum_{i=1}^n \left\{ t_i \hat{\mu}_0(\mathbf{x}_i) + \frac{(1 - t_i) \hat{p}(\mathbf{x}_i) y_i}{(1 - \hat{p}(\mathbf{x}_i))} \right\}.$$

We will conduct a simulation study to examine various properties of these estimators. Make sure your simulation study obeys the data-generating process assumptions, including overlap. In this case, we know from theory that cross-fitting is not necessary, so we'll skip it unless specifically asked for.

## 2.a High-Dimensional Parametric Case

(a) First, we study the high-dimensional parametric case. Suppose that  $\mu_0(\mathbf{x}) = \beta_0' \mathbf{x}$  and  $p(\mathbf{x}) = (1 + \exp\{-\boldsymbol{\theta}_0' \mathbf{x}\})^{-1}$ . Use a penalized linear model for  $\hat{\mu}_0(\mathbf{x}_i)$  and a penalized logistic regression for  $\hat{p}(\mathbf{x})$ . Try both LASSO and ridge regression.

Find the sampling distribution of both estimators  $\hat{\tau}_{PI}$  and  $\hat{\tau}_{DR}$  as the data-generating process varies. In particular, try all combinations of the following:

- Sample size  $n = 1000$  and  $5000$ ,
  - Dimension  $d = \dim(\mathbf{x}) = \{10, 50, 500, 5000\}$ , and
  - Sparsity levels  $s_\beta = \|\boldsymbol{\beta}_0\|_0 = \{d/10, d/2, d\}$  and  $s_0 = \|\boldsymbol{\theta}_0\|_0 = \{d/10, d/2, d\}$ .
- (i) What happens as  $n$  grows but  $d, s_\beta, s_0$  are fixed?
  - (ii) What happens to  $\hat{\tau}_{PI}$  as  $d$  and  $s_0$  change for fixed  $n$ ?
  - (iii) How does  $s_\beta$  impact  $\hat{\tau}_{PI}$ ?
  - (iv) Verify the doubly robust property of  $\hat{\tau}_{DR}$ .
  - (v) What happens if you do not penalize in the first stage, but just use plain OLS and logistic regression?
  - (vi) Discuss what your results mean for applied practice. When would you recommend the different estimators and why?
-

Table 1: Simulation Results Part 1

N	D	sBeta	sTheta	Estimator	N / D	Avg Error	Std Error
1000	10	d	d	DR	100	0.8289625	1.0536459
1000	10	d	d	plugin	100	1.3654053	0.7661954
1000	10	d	d/10	DR	100	0.3367593	0.4148811
1000	10	d	d/10	plugin	100	0.3446483	0.2757023
1000	10	d	d/2	DR	100	0.5104013	0.6888760
1000	10	d	d/2	plugin	100	0.9729192	0.6514369
1000	10	d/10	d	DR	100	0.1756136	0.1576128
1000	10	d/10	d	plugin	100	0.2058596	0.2059385
1000	10	d/10	d/10	DR	100	0.1514799	0.1670930
1000	10	d/10	d/10	plugin	100	0.0897218	0.1131439
1000	10	d/10	d/2	DR	100	0.1751841	0.2254457
1000	10	d/10	d/2	plugin	100	0.1507190	0.1668584
1000	10	d/2	d	DR	100	0.3886681	0.3814184
1000	10	d/2	d	plugin	100	0.7457178	0.4798246
1000	10	d/2	d/10	DR	100	0.2345614	0.2901427
1000	10	d/2	d/10	plugin	100	0.2249306	0.2484568
1000	10	d/2	d/2	DR	100	0.4475076	0.5375286
1000	10	d/2	d/2	plugin	100	0.4405855	0.4271002
1000	50	d	d	DR	20	1.7425122	9.9705895
1000	50	d	d	plugin	20	4.2533481	4.4073251
1000	50	d	d/10	DR	20	0.5006210	0.4423222
1000	50	d	d/10	plugin	20	1.0306153	0.6412584
1000	50	d	d/2	DR	20	0.5591062	0.5087023
1000	50	d	d/2	plugin	20	2.6736825	0.8461582
1000	50	d/10	d	DR	20	0.1849620	0.1663687
1000	50	d/10	d	plugin	20	0.4193823	0.3124290
1000	50	d/10	d/10	DR	20	0.2092971	0.2199909
1000	50	d/10	d/10	plugin	20	0.2299031	0.2265041
1000	50	d/10	d/2	DR	20	0.2293015	0.3761419
1000	50	d/10	d/2	plugin	20	0.3532981	0.3791846
1000	50	d/2	d	DR	20	0.4818870	0.5570632
1000	50	d/2	d	plugin	20	1.8936701	0.6943594
1000	50	d/2	d/10	DR	20	0.4460281	0.4332426
1000	50	d/2	d/10	plugin	20	0.6195131	0.5038511
1000	50	d/2	d/2	DR	20	0.4914814	0.6696792
1000	50	d/2	d/2	plugin	20	1.4475228	0.6815314

Table 2: Simulation Results Part 2

N	D	sBeta	sTheta	Estimator	N / D	Avg Error	Std Error
1000	500	d	d	DR	2.0	17.9285568	5.6802628
1000	500	d	d	plugin	2.0	13.6074814	1.5161596
1000	500	d	d/10	DR	2.0	5.7253886	2.7610946
1000	500	d	d/10	plugin	2.0	4.2959873	1.5011151
1000	500	d	d/2	DR	2.0	12.4509127	4.5789888
1000	500	d	d/2	plugin	2.0	9.0328626	1.7916930
1000	500	d/10	d	DR	2.0	1.3897604	1.2487602
1000	500	d/10	d	plugin	2.0	1.3014280	0.5679175
1000	500	d/10	d/10	DR	2.0	0.4931566	0.6097872
1000	500	d/10	d/10	plugin	2.0	0.5660653	0.4280557
1000	500	d/10	d/2	DR	2.0	0.9843785	1.0184316
1000	500	d/10	d/2	plugin	2.0	0.9373360	0.4962284
1000	500	d/2	d	DR	2.0	7.1841114	4.5077776
1000	500	d/2	d	plugin	2.0	6.6334833	1.0719964
1000	500	d/2	d/10	DR	2.0	2.4606980	2.0880298
1000	500	d/2	d/10	plugin	2.0	2.4278438	1.2619697
1000	500	d/2	d/2	DR	2.0	5.2030787	3.5670419
1000	500	d/2	d/2	plugin	2.0	4.6560049	1.1749485
1000	5000	d	d	DR	0.2	101.6967727	16.3903423
1000	5000	d	d	plugin	0.2	51.4791610	8.2111318
1000	5000	d	d/10	DR	0.2	30.6814917	7.6773673
1000	5000	d	d/10	plugin	0.2	15.4883335	4.9457006
1000	5000	d	d/2	DR	0.2	70.1112416	12.8244860
1000	5000	d	d/2	plugin	0.2	35.6549907	6.5645416
1000	5000	d/10	d	DR	0.2	9.5862837	2.9843560
1000	5000	d/10	d	plugin	0.2	5.2867315	1.7584879
1000	5000	d/10	d/10	DR	0.2	3.0579028	2.1493572
1000	5000	d/10	d/10	plugin	0.2	1.7631723	1.2499734
1000	5000	d/10	d/2	DR	0.2	7.2532640	2.7787505
1000	5000	d/10	d/2	plugin	0.2	3.7315856	1.6107951
1000	5000	d/2	d	DR	0.2	50.0813579	8.4346763
1000	5000	d/2	d	plugin	0.2	25.5902178	4.1130824
1000	5000	d/2	d/10	DR	0.2	15.2335804	5.7816218
1000	5000	d/2	d/10	plugin	0.2	7.3823526	3.5377925
1000	5000	d/2	d/2	DR	0.2	35.7630466	7.5578942
1000	5000	d/2	d/2	plugin	0.2	17.9834509	3.8040092

Table 3: Simulation Results Part 3

N	D	sBeta	sTheta	Estimator	N / D	Avg Error	Std Error
5000	10	d	d	DR	500	0.8914523	1.4249042
5000	10	d	d	plugin	500	1.4554703	0.6746919
5000	10	d	d/10	DR	500	0.3220056	0.2971075
5000	10	d	d/10	plugin	500	0.3113658	0.2706293
5000	10	d	d/2	DR	500	0.4949858	0.5870928
5000	10	d	d/2	plugin	500	0.9325675	0.5506595
5000	10	d/10	d	DR	500	0.2803053	0.7183947
5000	10	d/10	d	plugin	500	0.2602036	0.2906967
5000	10	d/10	d/10	DR	500	0.0979840	0.1916589
5000	10	d/10	d/10	plugin	500	0.0732481	0.1494821
5000	10	d/10	d/2	DR	500	0.1189428	0.1522234
5000	10	d/10	d/2	plugin	500	0.1467472	0.1848383
5000	10	d/2	d	DR	500	0.4614483	0.6244108
5000	10	d/2	d	plugin	500	0.7628519	0.4777335
5000	10	d/2	d/10	DR	500	0.2362347	0.3828132
5000	10	d/2	d/10	plugin	500	0.1653414	0.2095628
5000	10	d/2	d/2	DR	500	0.4543969	0.7312549
5000	10	d/2	d/2	plugin	500	0.5514870	0.4593897
5000	50	d	d	DR	100	1.4363282	5.5370874
5000	50	d	d	plugin	100	4.0422361	2.2054465
5000	50	d	d/10	DR	100	0.6597514	0.7858726
5000	50	d	d/10	plugin	100	1.0154214	0.6958616
5000	50	d	d/2	DR	100	7.3203088	50.8724433
5000	50	d	d/2	plugin	100	5.6124618	24.4872406
5000	50	d/10	d	DR	100	0.3002097	0.5975881
5000	50	d/10	d	plugin	100	0.4637236	0.3468204
5000	50	d/10	d/10	DR	100	0.1591160	0.2228308
5000	50	d/10	d/10	plugin	100	0.1770431	0.2074685
5000	50	d/10	d/2	DR	100	0.3092281	1.0589211
5000	50	d/10	d/2	plugin	100	0.3885081	0.5501093
5000	50	d/2	d	DR	100	0.8199922	1.6445958
5000	50	d/2	d	plugin	100	1.9760798	0.6407744
5000	50	d/2	d/10	DR	100	0.4143681	0.5433724
5000	50	d/2	d/10	plugin	100	0.6399909	0.5122750
5000	50	d/2	d/2	DR	100	0.5535384	0.6747499
5000	50	d/2	d/2	plugin	100	1.3127844	0.6731987

Table 4: Simulation Results Part 4

N	D	sBeta	sTheta	Estimator	N / D	Avg Error	Std Error
5000	500	d	d	DR	10	1.5965368	0.4647268
5000	500	d	d	plugin	10	12.7880767	1.1957097
5000	500	d	d/10	DR	10	0.4899293	0.2896340
5000	500	d	d/10	plugin	10	3.8765982	0.9772925
5000	500	d	d/2	DR	10	1.0591391	0.4067111
5000	500	d	d/2	plugin	10	9.0435419	1.0222070
5000	500	d/10	d	DR	10	0.2543478	0.1383597
5000	500	d/10	d	plugin	10	1.2713848	0.3623185
5000	500	d/10	d/10	DR	10	0.1230677	0.1002656
5000	500	d/10	d/10	plugin	10	0.4706731	0.4097631
5000	500	d/10	d/2	DR	10	0.1858852	0.1441898
5000	500	d/10	d/2	plugin	10	0.9234009	0.4206549
5000	500	d/2	d	DR	10	0.8231587	0.3178069
5000	500	d/2	d	plugin	10	6.2106128	0.6802169
5000	500	d/2	d/10	DR	10	0.3109772	0.3449476
5000	500	d/2	d/10	plugin	10	1.9332039	0.7599734
5000	500	d/2	d/2	DR	10	0.5656529	0.2932155
5000	500	d/2	d/2	plugin	10	4.5804962	0.8052152
5000	5000	d	d	DR	1	82.5515704	4.2194055
5000	5000	d	d	plugin	1	46.5889106	2.2536606
5000	5000	d	d/10	DR	1	24.8908999	4.1195436
5000	5000	d	d/10	plugin	1	14.1842472	2.4203427
5000	5000	d	d/2	DR	1	56.3236189	4.1160045
5000	5000	d	d/2	plugin	1	32.0891902	2.2255182
5000	5000	d/10	d	DR	1	5.1174294	3.2897701
5000	5000	d/10	d	plugin	1	4.7336737	0.6904658
5000	5000	d/10	d/10	DR	1	1.4200906	1.4632792
5000	5000	d/10	d/10	plugin	1	1.3642926	0.6547067
5000	5000	d/10	d/2	DR	1	3.5219772	2.6770786
5000	5000	d/10	d/2	plugin	1	3.3314214	0.7733954
5000	5000	d/2	d	DR	1	38.2854633	4.0460698
5000	5000	d/2	d	plugin	1	23.4760905	1.4225787
5000	5000	d/2	d/10	DR	1	11.5091956	3.4449016
5000	5000	d/2	d/10	plugin	1	7.0572061	1.6700501
5000	5000	d/2	d/2	DR	1	25.9949260	3.8401054
5000	5000	d/2	d/2	plugin	1	16.0792181	1.6384815



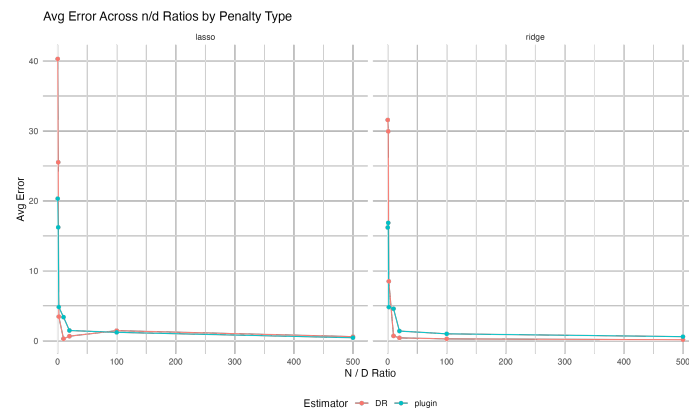
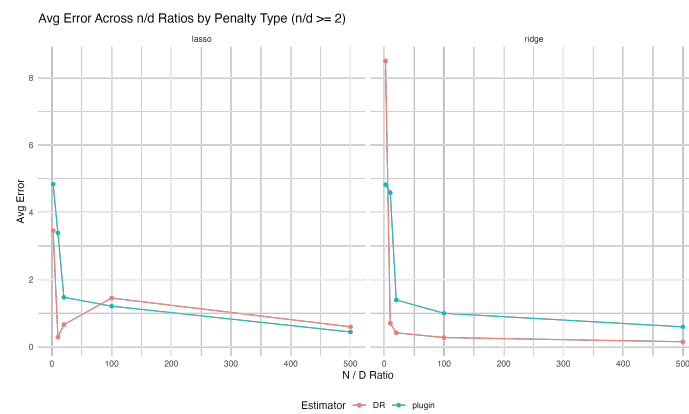
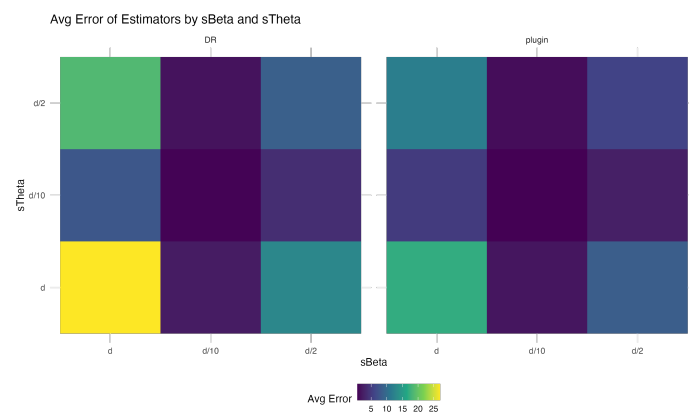
Figure 1: Average Error Across  $n / d$  by PenaltyFigure 2: Average Error Across  $n / d$  by Penalty Filter  $\frac{N}{D} > 2$ 

Figure 3: Average Error by sBeta and sTheta

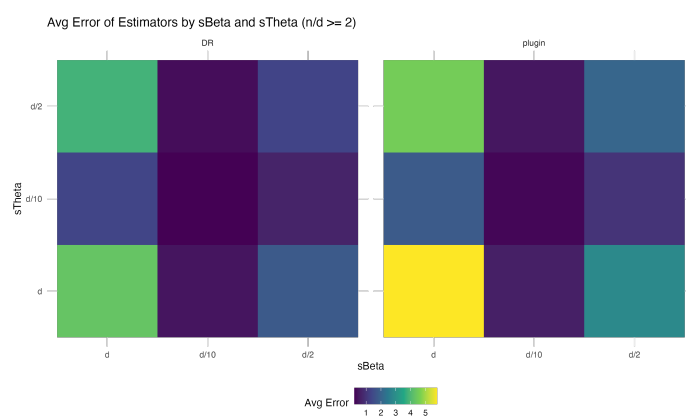
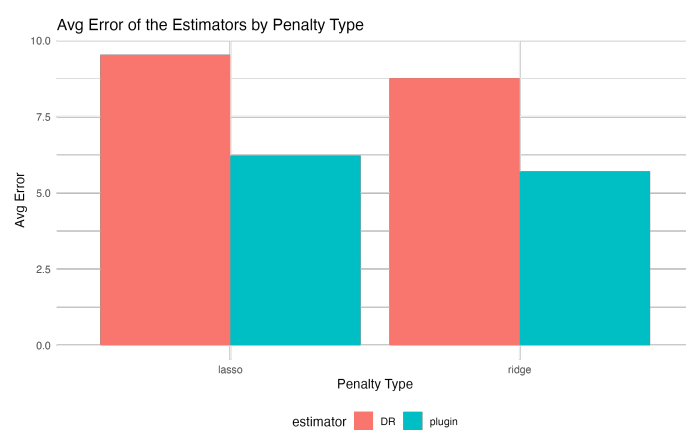
Figure 4: Average Error by sBeta and sTheta Filter  $\frac{N}{D} > 2$ 

Figure 5: Average Error of Estimators by Penalty

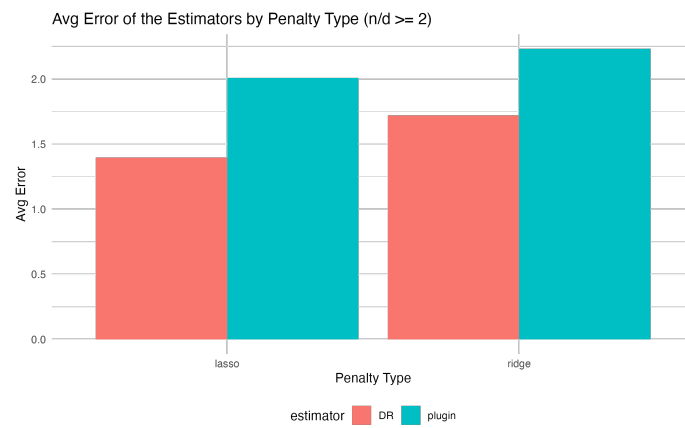
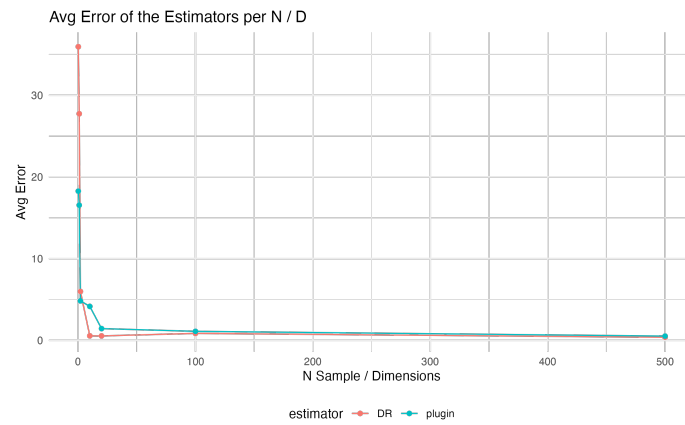
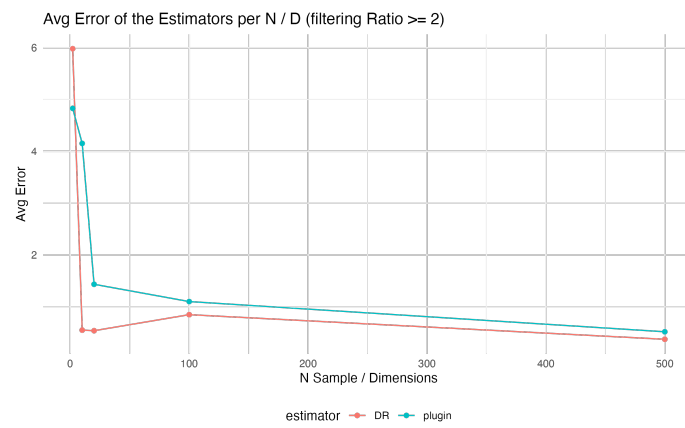
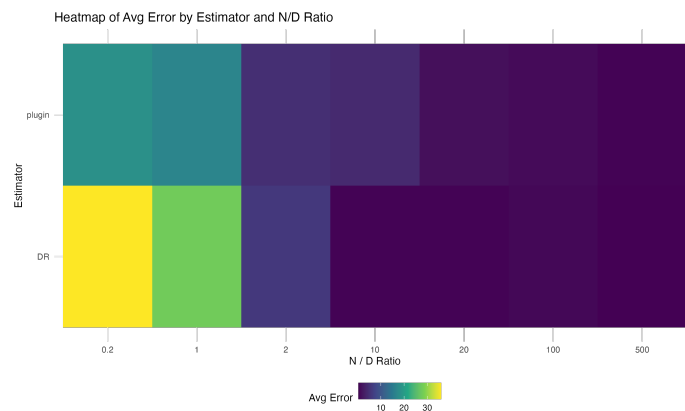
Figure 6: Average Error of Estimators by Penalty Filter  $\frac{N}{D} > 2$ 

Figure 7: Average Error of the Estimator per n / d

Figure 8: Average Error of the Estimator per  $n / d$  Filtering  $N / D \geq 2$ Figure 9: Heatmap of Average Error by Estimator and  $n / d$

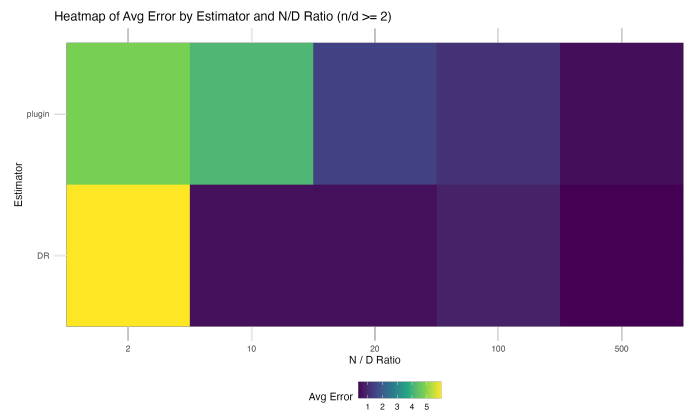


Figure 10: Heatmap of Average Error by Estimator and  $n / d$  Filter  $\frac{N}{D} > 2$

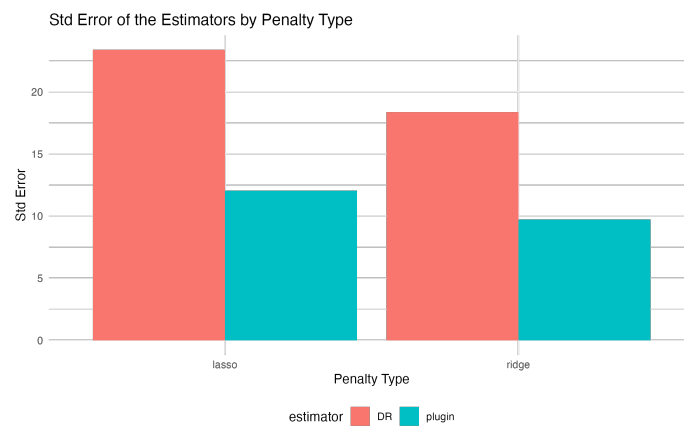


Figure 11: Standard Error of Estimators by Penalty

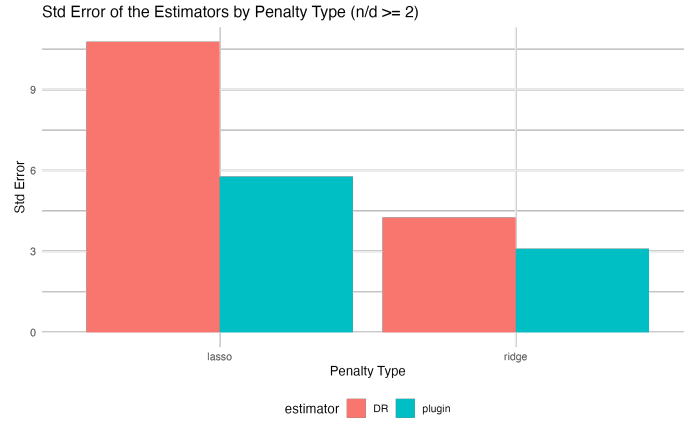


Figure 12: Standard Error of Estimators by Penalty Filter  $\frac{N}{D} > 2$

We run 75 simulations for each combination of the instructed parameters (10800 in total). The process required 8 parallel processes running for 12 hours.

(i) What happens as  $n$  grows but  $d, s_\beta, s_0$  are fixed?

As the sample size  $n$  increases from 1,000 to 5,000 while keeping the dimensionality  $d$  and sparsity levels  $s_\beta$  and  $s_0$  constant, the simulation results indicate that both the doubly robust (DR) and plug-in estimators generally improve in performance.

Specifically, the average error for the DR estimator decreases, demonstrating enhanced accuracy with larger  $n$ . For example, with  $d = 10$  and  $s_\beta = s_0 = d/10$ , the average error drops from 0.1515 at  $n = 1,000$  to 0.09798 at  $n = 5,000$ . Similarly, the plug-in estimator shows a reduction in average error from 0.0897 to 0.07325 under the same conditions.

Additionally, the standard error for both estimators tends to decrease as  $n$  increases, indicating more stable estimates. For instance, when  $d = 50$  and  $s_\beta = s_0 = d/10$ , the DR estimator's standard error decreases from 0.21999 at  $n = 1,000$  to 0.15911 at  $n = 5,000$ , and the plug-in estimator's standard error reduces from 0.2265 to 0.17704.

In high sparsity settings ( $s_\beta = s_0 = d$ ), the improvements are less pronounced, suggesting that high-dimensional parameter spaces may limit the benefits of increasing  $n$ .

(ii) What happens to  $\hat{\tau}_{PI}$  as  $d$  and  $s_0$  change for fixed  $n$ ?

Examining the plug-in estimator  $\hat{\tau}_{PI}$  with varying dimensionality  $d$  and sparsity levels  $s_0$  while keeping the sample size  $n$  fixed, the results reveal that increasing  $d$  leads to higher average errors and greater variability in the estimates.

For example, with  $n = 1,000$ , increasing  $d$  from 10 to 5,000 while maintaining  $s_\beta = s_0 = d$  results

in the average error rising from 1.3654 to 46.5889 and the standard error increasing from 0.7662 to 2.2537.

Additionally, higher sparsity levels  $s_0$  make it more clear the presence of performance decline of the plug-in estimator. Lowering  $s_0$  (i.e., reducing the number of non-zero coefficients in  $\theta_0$ ) improves the estimator's accuracy and reduces variability. For instance, with  $n = 1,000$  and  $d = 500$ , decreasing  $s_0$  from  $d$  to  $d/10$  lowers the average error from 13.6075 to 4.29599 and the standard error from 1.5162 to 1.5011.

The combination of high dimensionality and high sparsity significantly worsens the plug-in estimator's performance, highlighting its limitations in such settings. These findings suggest that the plug-in estimator is less reliable in high-dimensional, highly sparse environments, emphasizing the need for more robust estimation methods like the doubly robust estimator  $\hat{\tau}_{DR}$  in practical applications.

---

(iii) How does  $s_\beta$  impact  $\hat{\tau}_{PI}$ ?

---

As mentioned before, the sparsity level  $s_\beta$ , representing the number of non-zero coefficients in  $\beta_0$ , significantly influences the performance of the plug-in estimator  $\hat{\tau}_{PI}$ . As  $s_\beta$  increases, indicating a less sparse model with more non-zero coefficients, the plug-in estimator generally exhibits higher average error and greater variability. This trend can be observed across different dimensional settings:

For instance, consider the case where  $n = 1,000$  and  $d = 10$ :

- When  $s_\beta = d/10$  and  $s_\theta = d$ , the plug-in estimator has an average error of 0.0897 and a standard error of 0.1131.
- Increasing  $s_\beta$  to  $d/2$  with the same  $s_\theta$ , the average error rises to 0.1507 and the standard error to 0.1668.
- Further increasing  $s_\beta$  to  $d$ , the average error escalates to 1.3654 and the standard error to 0.7662.

A similar pattern is observed with higher dimensionality. For  $n = 1,000$  and  $d = 500$ :

- With  $s_\beta = d/10$  and  $s_\theta = d$ , the plug-in estimator records an average error of 4.29599 and a standard error of 1.501.
- Increasing  $s_\beta$  to  $d/2$ , the average error increases to 9.03286 and the standard error to 1.7917.
- When  $s_\beta = d$ , the average error reaches 13.6075 with a standard error of 1.5162.

In the high-dimensional scenario where  $d = 5,000$  and  $n = 1,000$ :

- For  $s_\beta = d/10$  and  $s_\theta = d$ , the plug-in estimator shows an average error of 1.7632 and a standard error of 0.6547.
- Increasing  $s_\beta$  to  $d/2$ , the average error grows to 25.5902 and the standard error to 1.4226.
- At  $s_\beta = d$ , the average error soars to 46.5889 with a standard error of 2.2537.

In settings with higher  $s_\beta$ , the plug-in estimator becomes less reliable, highlighting the importance of model sparsity for its effective application.

---

(iv) Verify the doubly robust property of  $\hat{\tau}_{\text{DR}}$ .

---

The doubly robust (DR) property of  $\hat{\tau}_{\text{DR}}$  implies that the estimator remains consistent for the ATT if either the outcome model  $\mu_0(\mathbf{x})$  or the propensity score model  $p(\mathbf{x})$  is correctly specified, but not necessarily both. To verify this property using the simulation results, we examine scenarios where one of the models is correctly specified (low sparsity) while the other is misspecified (high sparsity).

1. Scenario 1: Correct Outcome Model ( $s_\beta$  Low) and Misspecified Propensity Score Model ( $s_\theta$  High)
  - For  $n = 1000$ ,  $d = 10$ ,  $s_\beta = d/10$ , and  $s_\theta = d$ :
    - DR Estimator: Average error = 0.1515, Standard error = 0.1671
    - Plug-in Estimator: Average error = 0.0897, Standard error = 0.1131
    - Despite the propensity score model being highly sparse (potentially misspecified), the DR estimator maintains a low average error, indicating consistency due to the correctly specified outcome model.
2. Scenario 2: Both Models Misspecified ( $s_\beta$  High and  $s_\theta$  High)
  - For  $n = 1000$ ,  $d = 10$ ,  $s_\beta = d$ , and  $s_\theta = d$ :
    - DR Estimator: Average error = 0.82896, Standard error = 1.0536
    - Plug-in Estimator: Average error = 1.3654, Standard error = 0.7662
    - When both models are misspecified, the DR estimator does not maintain consistency, as expected. The average error increases significantly, reflecting the breakdown of the doubly robust property when both models are incorrect.
3. Additional Observations:
  - High Dimensionality (e.g.,  $d = 5000$ ): The DR estimator consistently shows lower average errors compared to the plug-in estimator when either  $s_\beta$  or  $s_\theta$  is low, reinforcing the doubly robust property in high-dimensional settings.
  - Varying Sparsity Levels: Across various dimensions ( $d = 10, 50, 500, 5000$ ), the DR estimator maintains low average errors when at least one of the models is correctly specified (low  $s_\beta$  or low  $s_\theta$ ), while the plug-in estimator's performance deteriorates more rapidly under model misspecification.

---

(v) What happens if you do not penalize in the first stage, but just use plain OLS and logistic regression?

---

When opting to not doing penalization in the first stage and instead utilize plain Ordinary Least Squares (OLS) for estimating  $\hat{\mu}_0(\mathbf{x})$  and standard logistic regression for estimating  $\hat{p}(\mathbf{x})$ , the performance of both the plug-in estimator  $\hat{\tau}_{\text{PI}}$  and the doubly robust estimator  $\hat{\tau}_{\text{DR}}$  decreases.



## 1. Increased Bias and Variability in High-Dimensional Settings:

- Higher Dimensionality ( $d$ ) with Limited Sample Size ( $n$ ): Without penalization, OLS and logistic regression are prone to overfitting, especially when  $d$  is large relative to  $n$ .
  - For  $n = 1,000$ ,  $d = 5,000$ ,  $s_\beta = s_\theta = d$ , the plug-in estimator exhibits an average error of 46.5889 and a standard error of 2.2537, indicating substantial bias and variability due to the high dimensionality and lack of regularization.

## 2. Degradation of Estimator Performance Across Sparsity Levels:

- In scenarios where the sparsity levels are high ( $s_\beta = s_\theta = d$ ), plain OLS and logistic regression fail to effectively identify and estimate the relevant predictors, leading to biased propensity scores and outcome models.
  - For  $n = 1,000$ ,  $d = 500$ ,  $s_\beta = s_\theta = d$ , the plug-in estimator records an average error of 13.6075 and a standard error of 1.5162, which are significantly higher compared to penalized approaches.
- Lower Sparsity Levels ( $s_\beta = s_\theta = d/10$ ): While reduced sparsity mitigates some of the negative impacts, the performance still lags behind penalized methods, particularly in very high-dimensional settings.

---

(vi) Discuss what your results mean for applied practice. When would you recommend the different estimators and why?

---

## 1. Use the Plug-in Estimator When:

- The dimensionality  $d$  is low to moderate.
- Models are expected to be well-specified with high sparsity ( $s_\beta, s_\theta$  are low).
- Simplicity and computational efficiency are priorities, and model misspecification is unlikely.

## 2. Use the Doubly Robust Estimator When:

- Dealing with high-dimensional data ( $d$  is large).
- Sparsity levels are moderate to low, making model estimation challenging.
- There is uncertainty about the correct specification of the outcome or propensity score models.
- Robustness to model misspecification is crucial for reliable ATT estimation.
- Regularization techniques (e.g., LASSO, ridge) are employed to handle high-dimensionality effectively.

## 2.b Nonparametrics and Low-Dimensional Case

(b) Now we turn to nonparametrics and lower-dimensional functions. Suppose that  $\mu_0(\mathbf{x})$  and  $p(\mathbf{x})$  are completely unknown functions. In your data-generating process, make them nonlinear functions of  $\mathbf{x}$ . Try  $n = \{1000, 5000, 15000\}$  and  $d = \dim(\mathbf{x}) = \{1, 3, 5, 10\}$ , including designs with sparsity. Use deep nets and random forests (and anything else you care to try).

For logit, by "nonlinear" we mean that  $p(\mathbf{x})$  has the logic form but the linear index  $\theta'_0 \wedge$  is replaced with something nonlinear.

Sparsity here is not based on slope coefficients, but rather it means that of the  $D$  covariates, only a subset enter the nonlinear function.

- (i) What happens as  $n$  grows but  $d$  is fixed?
- (ii) Verify the doubly robust property of  $\hat{\tau}_{\text{DR}}$ .
- (iii) Discuss what your results mean for applied practice. When would you recommend the different estimators and why?

Table 5: Simulation Results Part 1

N	D	S	Estimator	N / D	Avg DR Error	Std DR Error	Avg Plug-In Error	Std Plug-In Error
1000	1	1	deepnet	1000.0000	0.1188596	0.0865510	0.0694198	0.0454449
1000	1	1	rf	1000.0000	0.0683270	0.0523763	0.1115632	0.0546863
1000	3	1	deepnet	333.3333	0.1164264	0.0885983	0.0666727	0.0456159
1000	3	1	rf	333.3333	0.0655111	0.0495241	0.1079366	0.0519269
1000	3	3	deepnet	333.3333	0.1125104	0.0952149	0.0830327	0.0601980
1000	3	3	rf	333.3333	0.0667173	0.0490497	0.1136744	0.0660015
1000	5	1	deepnet	200.0000	0.1019231	0.0779841	0.0743787	0.0511270
1000	5	1	rf	200.0000	0.0665608	0.0510175	0.1102007	0.0532295
1000	5	3	deepnet	200.0000	0.1305549	0.0913589	0.0690302	0.0525333
1000	5	3	rf	200.0000	0.0654779	0.0464492	0.1162378	0.0683032
1000	5	5	deepnet	200.0000	0.5813612	0.1654391	0.1694454	0.0929925
1000	5	5	rf	200.0000	0.1122385	0.0741803	0.3828467	0.0807155
1000	10	1	deepnet	100.0000	0.1104556	0.0771283	0.0738879	0.0512893
1000	10	1	rf	100.0000	0.0622163	0.0472267	0.1064542	0.0495990
1000	10	3	deepnet	100.0000	0.1222687	0.0984759	0.0804333	0.0611218
1000	10	3	rf	100.0000	0.0614085	0.0472006	0.1147334	0.0680407
1000	10	5	deepnet	100.0000	0.5502481	0.1733933	0.1832190	0.0879109
1000	10	5	rf	100.0000	0.1599291	0.0812053	0.3806761	0.0896622
1000	10	10	deepnet	100.0000	1.4681068	0.2736488	0.2918145	0.1347418
1000	10	10	rf	100.0000	0.6989819	0.1172553	0.8138144	0.1026496
5000	1	1	deepnet	5000.0000	0.1064164	0.0655856	0.0579077	0.0350236
5000	1	1	rf	5000.0000	0.0313263	0.0231269	0.1020683	0.0244159
5000	3	1	deepnet	1666.6667	0.0937061	0.0600549	0.0606401	0.0346842
5000	3	1	rf	1666.6667	0.0242721	0.0189115	0.1022634	0.0235945
5000	3	3	deepnet	1666.6667	0.1194380	0.0860687	0.0848311	0.0570162
5000	3	3	rf	1666.6667	0.0299362	0.0229466	0.1075437	0.0316622
5000	5	1	deepnet	1000.0000	0.0942494	0.0660110	0.0595314	0.0352359
5000	5	1	rf	1000.0000	0.0266170	0.0216179	0.1045059	0.0255324
5000	5	3	deepnet	1000.0000	0.1027449	0.0756725	0.0851633	0.0518942
5000	5	3	rf	1000.0000	0.0292544	0.0218934	0.1086323	0.0313192

Table 6: Simulation Results Part 2

N	D	S	Estimator	N / D	Avg DR Error	Std DR Error	Avg Plug-In Error	Std Plug-In Error
5000	5	5	deepnet	1000	0.5165095	0.1435848	0.2362147	0.0897948
5000	5	5	rf	1000	0.0364207	0.0271482	0.3756248	0.0363502
5000	10	1	deepnet	500	0.0910494	0.0638729	0.0653223	0.0396288
5000	10	1	rf	500	0.0257117	0.0212406	0.1041807	0.0248034
5000	10	3	deepnet	500	0.1099078	0.0793048	0.0847055	0.0581834
5000	10	3	rf	500	0.0290623	0.0227746	0.1080859	0.0310643
5000	10	5	deepnet	500	0.5951550	0.1328736	0.1946967	0.0823967
5000	10	5	rf	500	0.0641742	0.0349222	0.3763468	0.0365609
5000	10	10	deepnet	500	1.6098024	0.1966549	0.3345875	0.1259575
5000	10	10	rf	500	0.5354508	0.0476285	0.8033817	0.0447472
15000	1	1	deepnet	15000	0.1035508	0.0571626	0.0584310	0.0284899
15000	1	1	rf	15000	0.0186054	0.0146221	0.1054463	0.0137249
15000	3	1	deepnet	5000	0.0954707	0.0593477	0.0617562	0.0280536
15000	3	1	rf	5000	0.0151110	0.0113118	0.1032908	0.0125218
15000	3	3	deepnet	5000	0.1171102	0.0817226	0.0763702	0.0468641
15000	3	3	rf	5000	0.0179146	0.0125474	0.1097410	0.0178791
15000	5	1	deepnet	3000	0.0937354	0.0573410	0.0603272	0.0306608
15000	5	1	rf	3000	0.0155132	0.0116021	0.1048228	0.0145629
15000	5	3	deepnet	3000	0.1141821	0.0829260	0.0770501	0.0466527
15000	5	3	rf	3000	0.0181026	0.0130724	0.1097178	0.0185147
15000	5	5	deepnet	3000	0.4484596	0.1458692	0.2330061	0.0776500
15000	5	5	rf	3000	0.0199905	0.0152260	0.3749759	0.0208752
15000	10	1	deepnet	1500	0.0950993	0.0580284	0.0634426	0.0343292
15000	10	1	rf	1500	0.0155375	0.0109449	0.1044657	0.0140622
15000	10	3	deepnet	1500	0.1027783	0.0749304	0.0824905	0.0516532
15000	10	3	rf	1500	0.0177048	0.0140744	0.1055849	0.0177291
15000	10	5	deepnet	1500	0.4877920	0.1370227	0.2389437	0.0795032
15000	10	5	rf	1500	0.0251532	0.0161137	0.3771937	0.0217643
15000	10	10	deepnet	1500	1.3764137	0.1999090	0.4351173	0.1361148
15000	10	10	rf	1500	0.4310972	0.0290622	0.7990867	0.0290638

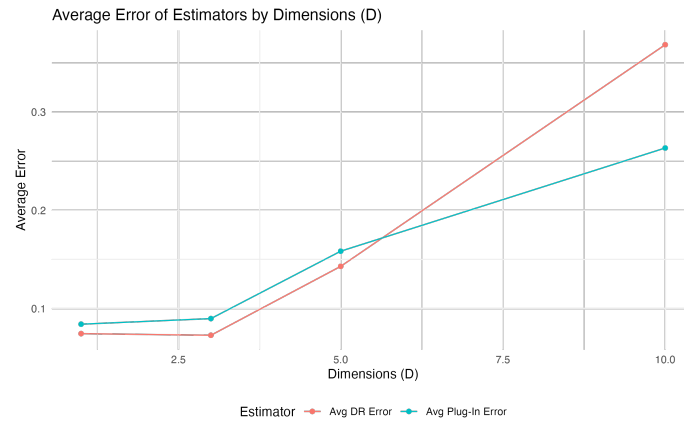


Figure 13: Average Error by Estimator and Dimensionality

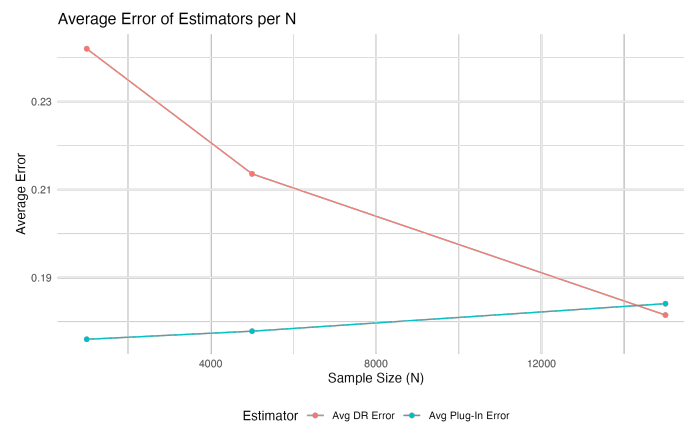


Figure 14: Average Error by Estimator and Sample Size (N).

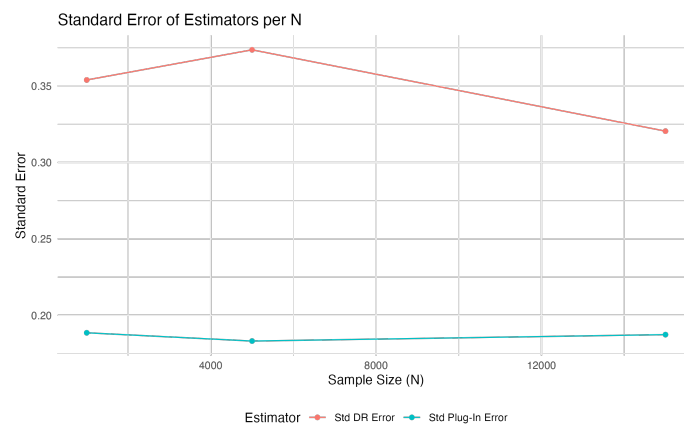


Figure 15: Standard Error by Estimator and Sample Size (N).

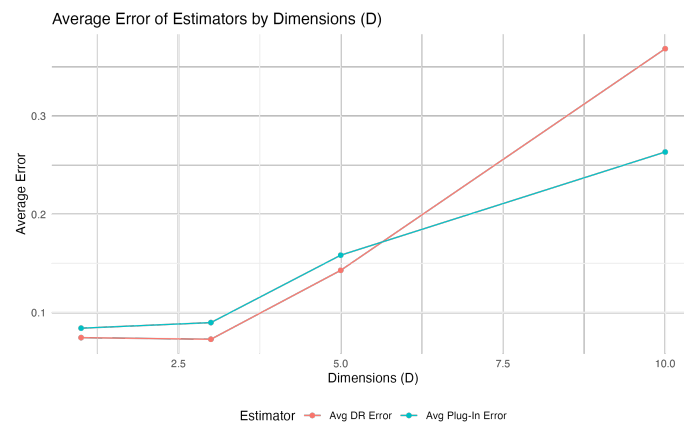


Figure 16: Average Error by Estimator and Dimensionality (D).

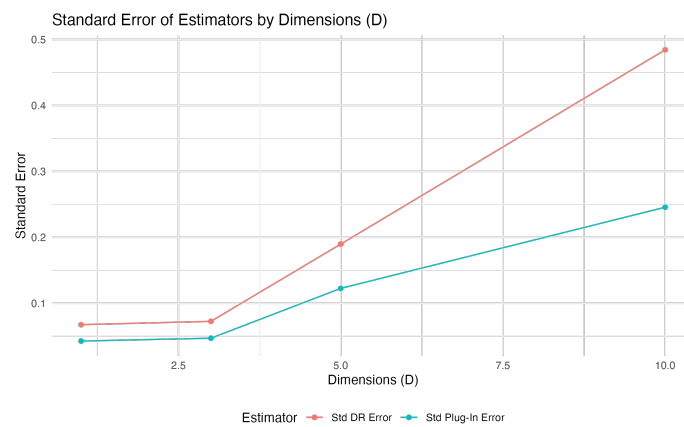


Figure 17: Standard Error by Estimator and Dimensionality (D).

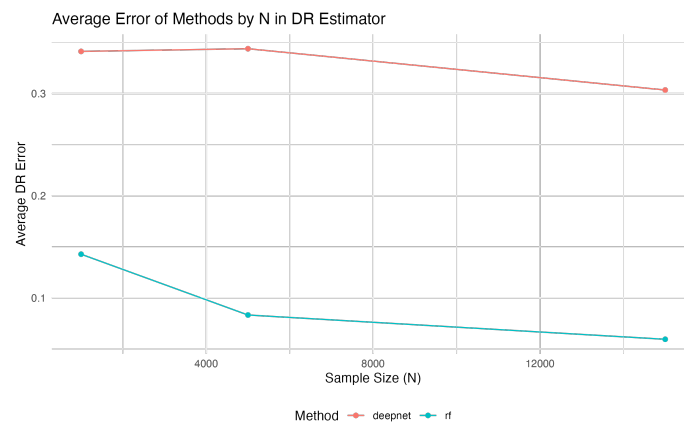


Figure 18: Average Error by Method and Sample Size (N) in DR Estimator.

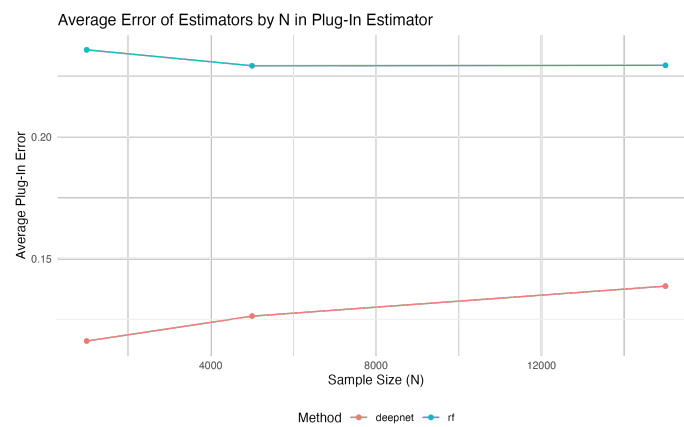


Figure 19: Average Error by Method and Sample Size (N) in Plug-In Estimator.

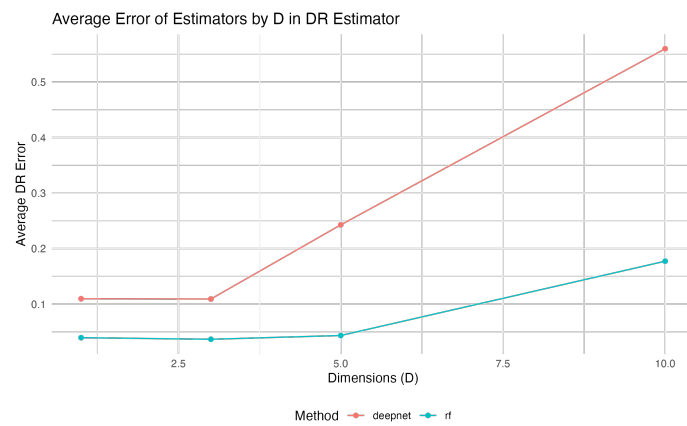


Figure 20: Average Error by Method and Dimensionality (D) in DR Estimator.

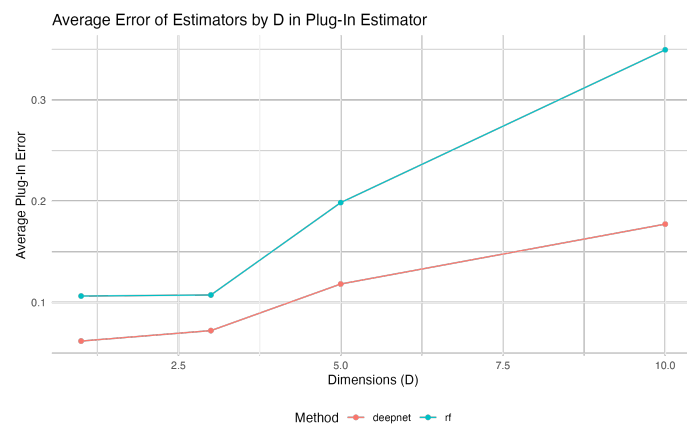


Figure 21: Average Error by Method and Dimensionality (D) in Plug-In Estimator.

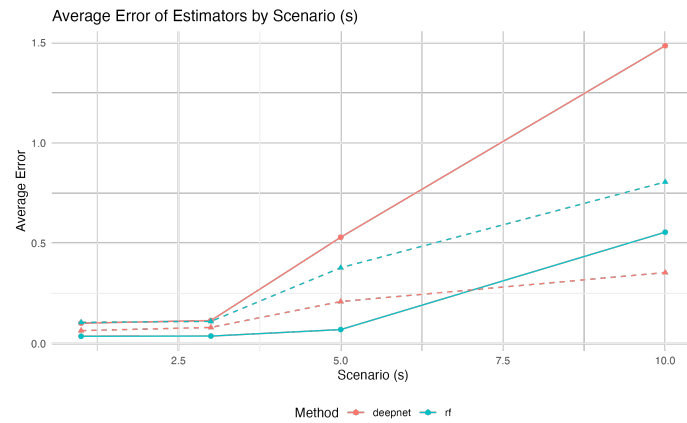


Figure 22: Average Error by Scenario (S) and Method.

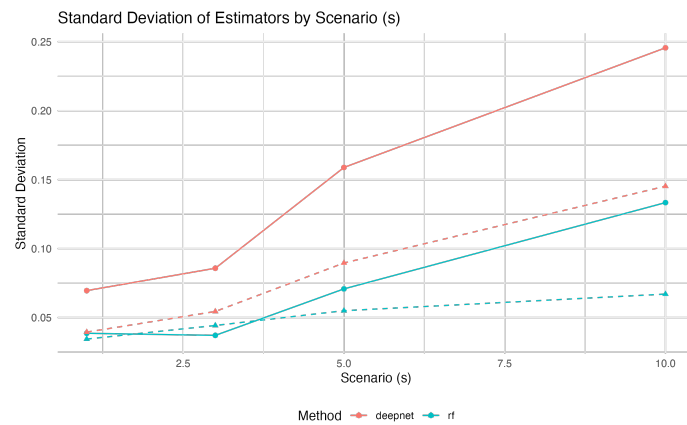
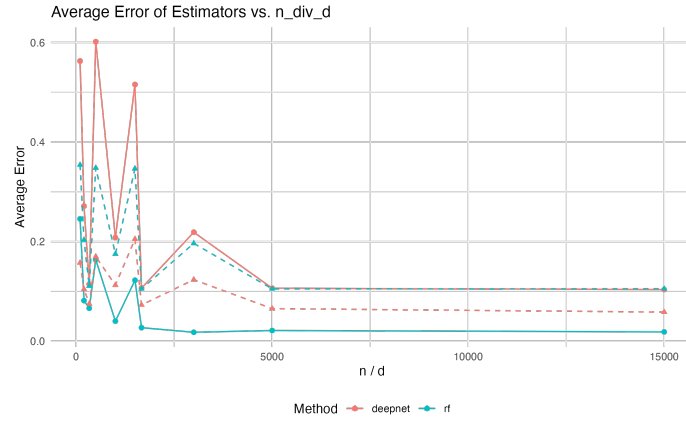
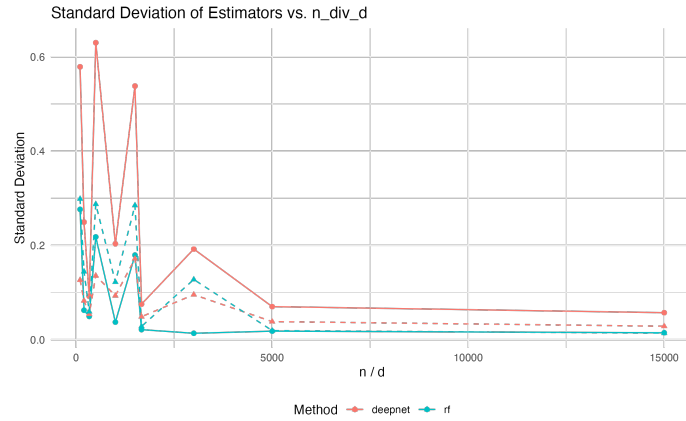


Figure 23: Standard Deviation by Scenario (S) and Method.



Figure 24: Average Error vs. Ratio of Sample Size to Dimensionality ( $n / d$ ).Figure 25: Standard Deviation vs. Ratio of Sample Size to Dimensionality ( $n / d$ ).

We run simulations for 5 hours.

As the sample size  $n$  increases while keeping the dimensionality  $d$  fixed, both the doubly robust (DR) and plug-in estimators exhibit a decrease in their average errors and standard deviations.

Specifically, for fixed  $d$ , increasing  $n$  from 1,000 to 15,000 leads to a reduction in the average DR error for both deep neural networks and random forests.

For instance, with  $d = 1$ , the DR error for the deepnet estimator decreases from approximately 0.119 to 0.104, and for the random forest estimator, it decreases from 0.068 to 0.019.

The simulation results support the doubly robust property of  $\hat{\tau}_{\text{DR}}$ . The DR estimator consistently shows competitive or superior performance compared to the plug-in estimator across various settings of  $n$  and  $d$ .

For example, when  $d = 1$  and  $n = 1,000$ , the DR estimator using random forests has a lower average error (0.068) compared to the plug-in estimator (0.112). Similarly, even as  $d$  increases,

the DR estimator maintains relatively stable performance, whereas the plug-in estimator's error may increase. This confirms its ability to remain consistent provided that either the propensity score model or the outcome model is correctly specified.

When dealing with datasets where the number of covariates  $d$  is relatively low and the sample size  $n$  is moderate to large, random forests emerge as a strong choice for estimating the ATT due to their lower average DR error and stability across different settings.

In high-dimensional settings or when there is sparsity in the covariates, the DR estimator using random forests still performs reliably, whereas deep neural networks may suffer from increased errors.

The simulation results reinforce the theoretical expectations regarding the performance and robustness of the doubly robust estimator in ATT estimation.

The results can still be improved, since we can provide better hyperparameter tuning. We are more certain about the outstanding performance of the DR estimator. Regarding the use of RF and DNN, further gridsearch can improve performance, specially for DNN with more regularization.

---

Now real data. Return to the Census data from class to find the ATT of sex on the log wage rate.

## 2.c Discuss Results

(c) Show results:

- (i) Both estimators  $\hat{\tau}_{PI}$  and  $\hat{\tau}_{DR}$ ,
- (ii) With and without cross-fitting,
- (iii) Using different first-step estimators for the propensity score  $\hat{p}(x_i)$  and regression function  $\hat{\mu}_0(x_i)$ , including forests, neural networks, LASSO, and parametric models.

Discuss the results.

---

Firstly, it is evident that the choice of propensity score model and outcome model significantly impacts the estimated ATT values.

The impact of cross-fitting is further illustrated across different models. While cross-fitting generally aims to reduce overfitting and improve estimator stability, its effects are not perfectly consistent. With the `NeuralNetworkClassifier` for propensity and `LassoRegression` for the outcome, cross-fitting changes  $\hat{\tau}_{DR}$  from  $-0.940$  to  $-1.952$ , demonstrating a more substantial impact. The same is not true for RF and Logistic Regression. This is probably because the neural network and LASSO without proper tuning are more prone to overfitting and cross-fitting helps to mitigate this issue.

Another point of interest is the comparison between the two estimators. The plug-in estimator ( $\hat{\tau}_{PI}$ ) consistently provides positive estimates across most scenarios, whereas the doubly robust estimator ( $\hat{\tau}_{DR}$ ) often yields negative values.

This discrepancy suggests that  $\hat{\tau}_{DR}$  may be more sensitive to model misspecification or that it captures different aspects of the treatment effect under varying model assumptions.

We also highlight the fact that the Random Forest estimates are often unstable and unreliable, as evidenced by the drastic changes in the estimated ATT values when using different models and cross-

fitting. This instability is particularly pronounced in the Random Forest Classifier for the propensity model, where the estimated ATT values vary significantly between cross-fitting and non-cross-fitting scenarios. Thus, we recommend that the analysis is not done using the RF model.

Ideally, we should aim to capture more covariates which might influence the result.

Furthermore, this problem is difficult in its nature: the covariates are posttreatment, meaning that  $T$  might influence the covariates and the outcome. This is different from a situation in which the covariates define to the probability of treatment: in this case, the treatment probably helps define some of the covariates (the causal relationship is inverse) and the covariates act as mediators: ideally, we should use a different orthogonal score function to make estimation, since the covariates are post-treatment. The covariates acting as mediators lead to frontdoor path which should be closed through the use of frontdoor adjustment.

To give a simple explanation, frontdoor adjustment is a method used to estimate the causal effect of treatment  $T$  on outcome  $Y$  when there is unmeasured confounding that cannot be addressed using backdoor adjustment. It leverages a mediator  $M$  that lies on the causal path from  $T$  to  $Y$ .

The causal effect of  $T$  on  $Y$  can be expressed using the frontdoor adjustment formula:

$$\mathbb{P}[Y(t)] = \sum_M \mathbb{P}[M \mid T = t] \sum_{t'} \mathbb{P}[Y \mid M, T = t'] \mathbb{P}[T = t']. \quad (1)$$

$t'$  acts as a dummy variable of integration (or summation) which, in our case, assumes values in  $\{0, 1\}$ . The  $\mathbb{P}[Y(t)]$  is computed by marginalizing over both the mediator  $M$  and any variation in the treatment  $T$ .

### Results with Adjustment Similar to the Code in R Provided in Class

In one the lectures, before doing analysis on the data, we filter:

- hours > 500
- income > 5000
- age < 60

First, we provide the results doing the same filtering as in the class code.

Table 7: Estimator Performance under Various Models

Propensity Model	Outcome Model	CrossFitting	Tau_PI	Tau_DR
LogisticRegression	LinearRegression	False	1.366433	-2.226286
LogisticRegression	LinearRegression	True	1.363280	-2.234279
LogisticRegression	RandomForestRegressor	False	1.366433	-2.230346
LogisticRegression	RandomForestRegressor	True	1.363280	-2.238295
LogisticRegression	NeuralNetworkRegressor	False	1.366433	-2.173630
LogisticRegression	NeuralNetworkRegressor	True	1.363280	-2.255142
LogisticRegression	LassoRegression	False	1.366433	-2.251588
LogisticRegression	LassoRegression	True	1.363280	-2.259363
RandomForestClassifier	LinearRegression	False	1.371529	-2.141202
RandomForestClassifier	LinearRegression	True	-0.650575	-10.374873
RandomForestClassifier	RandomForestRegressor	False	1.371529	-2.145262
RandomForestClassifier	RandomForestRegressor	True	-0.650575	-10.378888
RandomForestClassifier	NeuralNetworkRegressor	False	1.371529	-2.088546
RandomForestClassifier	NeuralNetworkRegressor	True	-0.650575	-10.395736
RandomForestClassifier	LassoRegression	False	1.371529	-2.166504
RandomForestClassifier	LassoRegression	True	-0.650575	-10.399956
NeuralNetworkClassifier	LinearRegression	False	1.509397	-2.020649
NeuralNetworkClassifier	LinearRegression	True	1.424044	-2.146075
NeuralNetworkClassifier	RandomForestRegressor	False	1.509397	-2.024709
NeuralNetworkClassifier	RandomForestRegressor	True	1.424044	-2.150091
NeuralNetworkClassifier	NeuralNetworkRegressor	False	1.509397	-1.967993
NeuralNetworkClassifier	NeuralNetworkRegressor	True	1.424044	-2.166938
NeuralNetworkClassifier	LassoRegression	False	1.509397	-2.045951
NeuralNetworkClassifier	LassoRegression	True	1.424044	-2.171158

Now, we provide the results without the filtering.

**Results without Random Forest Adjustment**

Table 8: Estimator Performance under Various Models

<b>Propensity Model</b>	<b>Outcome Model</b>	<b>CrossFitting</b>	<b>Tau_PI</b>	<b>Tau_DR</b>
LogisticRegression	LinearRegression	False	0.874867	-1.995883
LogisticRegression	LinearRegression	True	0.871097	-2.004188
LogisticRegression	RandomForestRegressor	False	0.874867	-1.951904
LogisticRegression	RandomForestRegressor	True	0.871097	-1.958615
LogisticRegression	NeuralNetworkRegressor	False	0.874867	-2.220714
LogisticRegression	NeuralNetworkRegressor	True	0.871097	-2.090103
LogisticRegression	LassoRegression	False	0.874867	-2.036195
LogisticRegression	LassoRegression	True	0.871097	-2.042983
RandomForestClassifier	LinearRegression	False	0.958338	-1.843984
RandomForestClassifier	LinearRegression	True	–	–
RandomForestClassifier	RandomForestRegressor	False	0.958338	-1.800005
RandomForestClassifier	RandomForestRegressor	True	–	–
RandomForestClassifier	NeuralNetworkRegressor	False	0.958338	-2.068815
RandomForestClassifier	NeuralNetworkRegressor	True	–	–
RandomForestClassifier	LassoRegression	False	0.958338	-1.884296
RandomForestClassifier	LassoRegression	True	–	–
NeuralNetworkClassifier	LinearRegression	False	1.067900	-1.644606
NeuralNetworkClassifier	LinearRegression	True	0.920863	-1.913626
NeuralNetworkClassifier	RandomForestRegressor	False	1.067900	-1.600627
NeuralNetworkClassifier	RandomForestRegressor	True	0.920863	-1.868053
NeuralNetworkClassifier	NeuralNetworkRegressor	False	1.067900	-1.869437
NeuralNetworkClassifier	NeuralNetworkRegressor	True	0.920863	-1.999540
NeuralNetworkClassifier	LassoRegression	False	1.067900	-1.684918
NeuralNetworkClassifier	LassoRegression	True	0.920863	-1.952421

## Results with Random Forest Adjustment

Table 9: ATT Estimates under Various Modeling Scenarios

Propensity Model	Outcome Model	CrossFitting	Tau_PI	Tau_DR
LogisticRegression	LinearRegression	False	0.896997	-1.103917
LogisticRegression	LinearRegression	True	0.895288	-1.106173
LogisticRegression	RandomForestRegressor	False	0.896997	-1.058235
LogisticRegression	RandomForestRegressor	True	0.895288	-1.059637
LogisticRegression	NeuralNetworkRegressor	False	0.896997	-1.326090
LogisticRegression	NeuralNetworkRegressor	True	0.895288	-1.180861
LogisticRegression	LassoRegression	False	0.896997	-1.125998
LogisticRegression	LassoRegression	True	0.895288	-1.128644
RandomForestClassifier	LinearRegression	False	0.912058	-1.016369
RandomForestClassifier	LinearRegression	True	-3.515407	-9.518771
RandomForestClassifier	RandomForestRegressor	False	0.912058	-0.987425
RandomForestClassifier	RandomForestRegressor	True	-3.515407	-9.780664
RandomForestClassifier	NeuralNetworkRegressor	False	0.912058	-1.229146
RandomForestClassifier	NeuralNetworkRegressor	True	-3.515407	-9.763415
RandomForestClassifier	LassoRegression	False	0.912058	-1.037920
RandomForestClassifier	LassoRegression	True	-3.515407	-9.635466
NeuralNetworkClassifier	LinearRegression	False	1.273571	-0.917413
NeuralNetworkClassifier	LinearRegression	True	1.103739	-1.002065
NeuralNetworkClassifier	RandomForestRegressor	False	1.273571	-0.869977
NeuralNetworkClassifier	RandomForestRegressor	True	1.103739	-0.957053
NeuralNetworkClassifier	NeuralNetworkRegressor	False	1.273571	-1.164582
NeuralNetworkClassifier	NeuralNetworkRegressor	True	1.103739	-1.081366
NeuralNetworkClassifier	LassoRegression	False	1.273571	-0.940831
NeuralNetworkClassifier	LassoRegression	True	1.103739	-1.952421

## Extra: DML Estimation of ATE

As an extra, we also estimate the ATE using Double Machine Learning. We provide the result of the model with the smallest standard error, which in this case is a Random Forest.

```

1 DoubleML-based Doubly Robust ATE estimate: 0.12342314601251633
2 Standard Error: 0.17182105633995212
3

```

Listing 1: Double ML estimation of the ATE (not ATT)

### 3 An Application

The file `data_for_HW4.csv` contains data from two independent sources, as indicated by the variable `e`. Both have data on the same outcome  $y$ , same treatment  $t$ , and the same set of pre-treatment variables  $\mathbf{x}.1, \mathbf{x}.2, \mathbf{x}.3, \mathbf{x}.4, \mathbf{x}.5$ . The treatment in the first data source may have been targeted based on some or all of the  $\mathbf{x}$  variables. The second data source is a fully randomized experiment. Both obey our other assumptions (SUTVA, consistency, CIA, overlap).

#### 3.a Ignoring $\mathbf{x}$ Variables

Ignore the  $\mathbf{x}$  variables to compute the ATE and a confidence interval for it in each of the data sources. Comment on your findings and possible explanations for them.

```

1 Results for data source e=1 (observational data):
2 ATE estimate: -1.060766
3 95% CI: -1.136993 to -0.9845383
4
5 Results for data source e=2 (fully randomized):
6 ATE estimate: 1.938148
7 95% CI: 1.869419 to 2.006877
8

```

Listing 2: ATE and Confidence Interval Estimates Ignoring Covariates

The results show a discrepancy between the two data sources. In the first data source, where treatment assignment was potentially targeted based on observed pre-treatment characteristics, the estimated average treatment effect ignoring covariates is approximately

$$\hat{\tau}_{e=1} \approx -1.06.$$

The 95% confidence interval shows:

$$-1.14 \leq \tau_{e=1} \leq -0.98$$

In the second data source, which is fully randomized, the estimated average treatment effect ignoring covariates is

$$\hat{\tau}_{e=2} \approx 1.94.$$

The corresponding confidence interval is approximately

$$1.87 \leq \tau_{e=2} \leq 2.01,$$

suggesting a positive and statistically significant effect of the treatment in the randomized setting. These findings can be explained by the difference in treatment assignment mechanisms. For the first data source, if the treatment was assigned based on variables correlated with the outcome, the simple difference-in-means estimator is not unbiased. Due to non-random assignment, treated and control units differ systematically in ways that influence their outcomes, leading to a biased estimate of the treatment effect. Mathematically, ignoring the covariates, the conditional independence assumption does not hold, and we have

$$E[Y(0) | T = 1] \neq E[Y(0) | T = 0],$$

causing the observed difference in means

$$\hat{\tau}_{\text{obs}} = E[Y | T = 1] - E[Y | T = 0]$$

to deviate from the true average treatment effect.

In contrast, for the fully randomized second data source, the assignment is independent of potential outcomes:

$$T \perp (Y(0), Y(1)),$$

ensuring that

$$E[Y(0) | T = 1] = E[Y(0) | T = 0].$$

Thus, the simple difference-in-means here recovers an unbiased estimate of the treatment effect, producing a positive and significant result. This once more illustrates the importance of randomization for obtaining unbiased treatment effect estimates or adjusting for observed confounders.

### 3.b Linear Model with Interactions

Use a linear model with interactions to obtain the CATEs in each data source, plot the distribution of the CATEs, obtain the ATE and its confidence interval. Compare your findings on the ATEs to the previous part.

```

1 Results for data source e=1:
2 ATE: 1.362815
3 95% CI: 1.252676 to 1.472955
4
5 Results for data source e=2:
6 ATE: 1.994436
7 95% CI: 1.958671 to 2.030201
8
```

Listing 3: ATE and Confidence Interval Estimates Ignoring Covariates



Estimate	$e_1$	$e_2$
Mean	1.36	1.99
Std	0.897	1.83
Skewness	0.00223	0.00222
Kurtosis	2.99	2.99
Q10	0.217	-0.346
Q25	0.758	0.759
Median	1.36	1.99
Q75	1.97	3.23
Q90	2.51	4.36

Figure 27: Descriptive Statistics of Conditional Average Treatment Effects (CATEs)

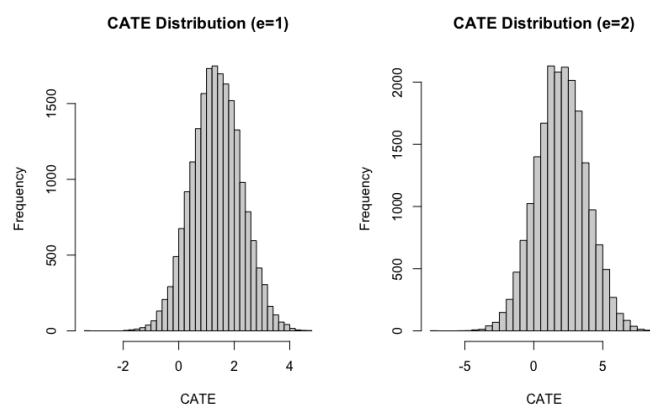


Figure 26: Distribution of Conditional Average Treatment Effects (CATEs) in Each Data Source

The findings indicate that after incorporating covariates and allowing for treatment-covariate interactions, both data sources produce a positive average treatment effect estimate. For the first data source, the previously obtained raw difference-in-means estimate suggested a negative treatment effect. In contrast, the adjusted model now yields

$$\hat{\tau}_{e=1} \approx 1.36,$$

with a 95% confidence interval

$$[1.25, 1.47].$$

For the second data source, where the assignment was fully randomized, the estimate remains consistently positive and similar to the earlier unadjusted results:

$$\hat{\tau}_{e=2} \approx 1.99,$$

with a 95% confidence interval

$$[1.96, 2.03].$$

The distribution of the conditional average treatment effects (CATEs) in each data source shows that, when controlling for pre-treatment variables, the CATEs are roughly symmetric with near-zero skewness and close-to-normal kurtosis. For the first data source, the mean CATE is around 1.36, while for the second it is around 1.99. This suggests that, within each data source, adjusting for covariates and including interactions reveals a more consistent and positive treatment effect across individuals.

Comparing these results to the previous part, we see a clear difference for the first data source. Without adjusting for covariates, the estimate was negative, indicating that units selected for treatment may have been systematically different, likely with lower expected outcomes, violating the conditional independence assumption. Once we incorporate the covariates and their interactions, we effectively control for the selection mechanism:

$$E[Y(0) | T = 1, X] = E[Y(0) | T = 0, X],$$

which brings the adjusted estimate closer to what might be the true effect. Mathematically, we had previously

$$\hat{\tau}_{e=1, \text{unadjusted}} = E[Y | T = 1] - E[Y | T = 0] < 0,$$

but after conditioning on  $X$  and modeling the interactions, the conditional expectation of the untreated potential outcome given treatment and  $X$  aligns with that of the controls, yielding

$$\hat{\tau}_{e=1, \text{adjusted}} = E_Y[T = 1, X] - E_Y[T = 0, X] > 0.$$

For the second data source, where treatment is randomized and thus independent of  $X$ ,

$$T \perp (Y(0), Y(1), X),$$

the unadjusted difference-in-means already provided an unbiased estimate of the average treatment effect. The inclusion of covariates and interactions only slightly refines this estimate, reaffirming that the simple difference-in-means was appropriate and stable. Here, the adjusted ATE remains close to the previously estimated value, thus confirming

$$\hat{\tau}_{e=2, \text{adjusted}} \approx \hat{\tau}_{e=2, \text{unadjusted}}.$$

---

### 3.c Doubly Robust Estimation

Combine the estimators of  $\mu_t(x) = \mathbb{E}[Y(t) | X = x]$  with a parametric logistic regression estimate of the propensity score  $p(x) = \mathbb{P}[T = 1 | X = x]$  to estimate the ATE and confidence interval in each data source using the doubly robust estimator. Compare your findings on the ATEs to the previous two parts.

---

Here, we run the regression only adjusting by the probability and then running a proper double ML model with orthogonal score function.

---

First, in both cases, we see a decrease in the standard error of the ATE when compared to (3.b). The decrease in uncertainty is specially visible when using Double Machine Learning method.

The estimate for  $e_1$  differs considerably between the first and second method (1.28 and 0.44). On the other hand, the estimate for  $e_2$  is very similar between the two methods (1.99 and 1.99). Method 2 (DML) provides smaller standard errors for both data sources, which is expected given the bigger robustness of the method.

### Estimator Using Logistic Regression

```

1 Results for data source e=1 (doubly robust, corrected):
2 ATE: 1.281753
3 95% CI: 1.202279 to 1.361226
4
5 Results for data source e=2 (doubly robust, corrected):
6 ATE: 1.994437
7 95% CI: 1.95078 to 2.038093
8

```

Listing 4: Doubly Robust ATE Estimation

### Estimator Using Double Machine Learning

```

1 Results for data source e=1 (doubly robust, corrected):
2 coef    std err          t      P>|t|     2.5 %    97.5 %
3 d   0.445511   0.030286   14.71035   5.532050e-49   0.386152   0.504869
4
5 Results for data source e=2 (doubly robust, corrected):
6 coef    std err          t      P>|t|     2.5 %    97.5 %
7 d   1.993791   0.022734   87.701211   0.0   1.949233   2.038348

```

Listing 5: Doubly Robust ATE Estimation

## 3.d Flexible/Nonparametric Versions

Replace your estimates of  $\mu_t(x)$  and  $p(x)$  with flexible/nonparametric/ML versions, and repeat the doubly robust estimation and inference. Try a few different nonparametric estimators for practice.

In this question, we use DoubleML estimator to implement the doubly robust estimation with flexible/nonparametric/ML versions of the treatment effect and propensity score models. The following propensity score function is used:

$$\psi(Y_i, T_i, X_i; \eta) = \left( \frac{T_i - g(X_i)}{\pi(X_i)(1 - \pi(X_i))} \right) (Y_i - m(X_i)) + (m_1(X_i) - m_0(X_i)) - \tau$$

In which  $m(X_i) = \mathbb{E}[Y_i | X_i]$ ,  $\pi(X_i) = \mathbb{P}[T_i = 1 | X_i]$ , and  $g(X_i) = \mathbb{E}[T_i | X_i]$ . Because the treatment is binary,  $\pi(X_i) = g(X_i)$ .

We use the same flexible model for both the treatment effect and propensity score models in each of our tentatives. We test with grid search Random Forest, Gradient Boosting, DeepNN, LASSO.

To avoid overfitting of the most complicated models, we use cross-fitting with 5 folds. The results are presented in the following tables:

**Doubly Robust ATE Estimation for  $e_1$  without Grid Search**

Method	Coef	Std Err	t	p-value	2.5%	97.5%
RandomForest	0.484574	0.031210	15.526405	2.299137e-54	0.423405	0.545744
GradientBoosting	0.521255	0.032215	16.180598	6.911977e-59	0.458116	0.584395
DeepNN	0.496879	0.032290	15.388224	1.963492e-53	0.433593	0.560166
LASSO	0.443827	0.027538	16.116643	1.949143e-58	0.389852	0.497801

Table 10: Doubly Robust ATE Estimation for  $e_1$

**Doubly Robust ATE Estimation for  $e_2$  without Grid Search**

Method	Coef	Std Err	t	p-value	2.5%	97.5%
RandomForest	1.997878	0.022700	88.013960	0.000000e+00	1.953388	2.042369
GradientBoosting	1.996379	0.021839	91.415537	0.000000e+00	1.953576	2.039182
DeepNN	1.927452	0.022653	85.085649	0.000000e+00	1.883053	1.971852
LASSO	1.993349	0.022390	89.029704	0.000000e+00	1.949466	2.037232

Table 11: Doubly Robust ATE Estimation for  $e_2$

**Doubly Robust ATE Estimation for  $e_1$  with Grid Search**

Method	Coef	Std Err	t	p-value	2.5%	97.5%
RandomForest	0.502296	0.032106	15.644891	3.599873e-55	0.439369	0.565223
GradientBoosting	0.518262	0.032348	16.021308	9.072225e-58	0.454860	0.581663
DeepNN	0.500243	0.032388	15.445140	8.135033e-54	0.436763	0.563723
LASSO	0.443051	0.027604	16.050380	5.681481e-58	0.388948	0.497153

Table 12: Doubly Robust ATE Estimation for  $e_1$

**Doubly Robust ATE Estimation for  $e_2$  with Grid Search**

Method	Coef	Std Err	t	p-value	2.5%	97.5%
RandomForest	2.000420	0.021877	91.438273	0.000000e+00	1.957541	2.043299
GradientBoosting	1.990438	0.021741	91.550747	0.000000e+00	1.947826	2.033051
DeepNN	1.907454	0.021751	87.694617	0.000000e+00	1.864823	1.950085
LASSO	1.993790	0.022269	89.530360	0.000000e+00	1.950142	2.037437

Table 13: Doubly Robust ATE Estimation for  $e_2$ 

We see that the results differ from (3.b) for the observational data. The RCT maintains the ATE extremely similar, allowing us to validate our code.

Among the RCT models, Gradient Boosting has the smallest standard error. The ATE estimates are all significant at 1%.

For the observational data, LASSO has the smallest standard error, with the most different coefficient, when compared to the others estimates with the observational data. The coefficient estimates are all significant at 1%.

### 3.e Combined Model for Both Datasets

Propose and estimate a model (parametric or not) that combines and uses the two datasets as one. In other words, your model should have a single loss function, shared or common parameters, and appropriate assumptions as you deem fit. You must use data from both sources. Discuss your choice of specification and the properties of your proposed estimator.

Using Double Machine Learning (DML) to separately estimate the propensity score (for the observational data) and the outcome variable (for both datasets) can be a good approach.

We propose this first approach and a second approach using  $e$  as covariate and again DML.

#### First Approach

Given that we have two datasets, indexed by  $e \in \{1, 2\}$ :

- $e = 1$ : Observational data with non-random treatment assignment.
- $e = 2$ : RCT data with randomized treatment assignment.

We should be able to retrieve the true ATE if the following assumptions hold:

- No Unmeasured Confounding (for  $e = 1$ ):  $(Y(0), Y(1)) \perp T \mid X$ .
- Randomization in  $e = 2$ :  $T \perp X$ .
- Overlap: There exists  $\epsilon > 0$  such that  $\epsilon \leq P(T = 1|X) \leq 1 - \epsilon$ .

Double Machine Learning for Observational Data ( $e = 1$ ) involves:

- Estimating the propensity score,  $\hat{p}(X) = P(T = 1|X, e = 1)$ , using machine learning models.
- Estimating the conditional outcome regression,  $\hat{\mu}(T, X) = E[Y|T, X]$ , for both  $T = 0$  and  $T = 1$ .
- Constructing orthogonal score functions to estimate the treatment effect, ensuring robustness to regularization bias in the nuisance parameter estimation.

For both datasets, we define the conditional outcome model:

$$\mu(T, X; \theta) = E[Y|T, X].$$

Additionally, for the observational dataset ( $e = 1$ ), we define a propensity model:

$$p(X; \gamma) = P(T = 1|X, e = 1).$$

For the RCT dataset ( $e = 2$ ), the treatment assignment is known:

$$P(T = 1|X, e = 2) = p_0,$$

where  $p_0$  is constant. We check that by looking at the empirical probability of treatment in the RCT dataset. To verify, we group the covariates by percentiles and check the percentage of observations with treatment in each group. For the RCT dataset, we see a constant percentage of treatment across all groups.

e	P0-P24	P25-P49	P50-P74	P75-P100	x
1	0.0326	0.101	0.109	0.219	x1
1	0.0146	0.0322	0.0758	0.339	x2
1	0.118	0.119	0.112	0.113	x3
1	0.119	0.112	0.117	0.113	x4
1	0.114	0.113	0.122	0.112	x5
2	0.508	0.507	0.508	0.500	x1
2	0.495	0.509	0.506	0.514	x2
2	0.512	0.504	0.504	0.504	x3
2	0.509	0.512	0.496	0.507	x4
2	0.506	0.500	0.511	0.507	x5

Table 14: Data Table

On observational data, the probability of treatment varies considerably in  $x_1$  and  $x_2$ . For  $x_3$ ,  $x_4$ , and  $x_5$ , the probability of treatment is relatively constant across percentiles. In the RCT data, the probability of treatment is constant across all covariates.

The loss function incorporates contributions from both datasets:

$$\mathcal{L}(\theta, \gamma) = \mathcal{L}_{\text{RCT}}(\theta) + \mathcal{L}_{\text{OBS}}(\theta, \gamma),$$

where:

- $\mathcal{L}_{\text{RCT}}(\theta) = \sum_{i:e_i=2} (Y_i - \mu(T_i, X_i; \theta))^2$ , capturing the fit of the outcome model for the RCT.
- $\mathcal{L}_{\text{Obs}}(\theta, \gamma)$  is based on doubly-robust moment conditions for the observational data:

$$\mathcal{L}_{\text{OBS}}(\theta, \gamma) = \sum_{i:e_i=1} \psi_i(\theta, \gamma),$$

where:

$$\psi_i(\theta, \gamma) = \frac{T_i - p(X_i; \gamma)}{p(X_i; \gamma)(1 - p(X_i; \gamma))} \cdot (Y_i - \mu(T_i, X_i; \theta)) + \mu(1, X_i; \theta) - \mu(0, X_i; \theta).$$

The parameters  $\theta$  and  $\gamma$  can be estimated jointly by minimizing  $\mathcal{L}(\theta, \gamma)$ , potentially using iterative or optimization-based algorithms. Machine learning methods (e.g., random forests, gradient boosting, or neural networks) can flexibly estimate  $\mu(T, X)$  and  $p(X)$ , with sample splitting to ensure orthogonality and reduce overfitting bias.

The estimator remains consistent for the treatment effect under the stated assumptions. The RCT data ensures unbiased identification of the treatment effect, while the observational data provides additional precision.

For the observational component ( $e = 1$ ), the estimator remains consistent if either the propensity model  $p(X; \gamma)$  or the outcome model  $\mu(T, X; \theta)$  is correctly specified.

## Second Approach

Again, we propose the use of DML. In this scenario, we join the datasets and use  $e$  as a covariate. Furthermore, we also use interaction between  $e$  and  $x$ , to allow for different effects of the covariates on the probability and on the target variable for each dataset.

In this situation, we propose a bigger level of regularization for the parameters of  $x$  (the parameters for the features which multiply  $e$  by  $x$ ) that aim to estimate the propensity score for the RCT dataset, since we expect that the treatment assignment is roughly 50%.

In this scenario, we can use a single loss function and estimate the propensity score and the target variable simultaneously for both datasets. Here,  $e$  becomes an important covariate, separating the two estimates.

## Deep Neural Networks Estimation

```

1 coef      std err          t    P>|t|      2.5 %      97.5 %
2 t      1.5903    0.018785   84.658239      0.0    1.553482    1.627117
3
```

Listing 6: ATE and Confidence Interval Estimates Ignoring Covariates

**Random Forest Estimation Estimation**

```
1 coef      std err          t  P>|t|      2.5 %      97.5 %  
2 t    1.314469    0.020126   65.310582    0.0    1.275022    1.353917
```

Listing 7: ATE and Confidence Interval Estimates Ignoring Covariates