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 $T \in \{0, 1, \ldots, T\}$, for some integer T, on an outcome Y. We observe $Z = (Y, T, \mathbf{X}')' \in \mathbb{R} \times \{0, 1, \ldots, T\} \times \mathbb{R}^d$. 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 μ_t, $t \in \{0, \ldots, T\}$. 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 μ_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 α_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 Σ 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 σ_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 μ_t is identified. Compare these conditions to what you found above.

For this question, we consider the following assumptions:

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

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

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

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), μ_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 μ_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 $J = J_n \to \infty$ 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 γ. We will ignore C and focus on rates here.
First we study nonparametric estimation and inference.

1.c Range of γ for Consistency

For what range of γ 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 $n \to \infty$. 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 γ must lie in the interval

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

This shows that the dimension d directly affects the range for γ.

As d increases, the upper bound $1/d$ decreases: this make it harder to find a γ 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 γ in $(0, 1/d)$ for consistency. In conclusion, the interval is never empty for any fixed positive integer d.

1.d Optimal Value of γ

What value of γ is optimal in the sense that the rate is the fastest? Call this $\gamma^\star_{\text{mse}}$. How does $\gamma^\star_{\text{mse}}$ 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 γ 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 $\gamma^{\star}_{\text{mse}}$. It is the γ that equates the rates of the bias and variance terms. It optimizes the mean squared error rate.

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

$$\gamma^{\star}_{\text{mse}}=\frac{1}{d+2(p+1)}.$$

As the dimension d increases, the denominator increases, making $\gamma^{\star}_{\text{mse}}$ smaller.

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

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

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

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

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

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 γ is never empty because $d + 2(p+1) > d$ always.

1.f Range of γ for Optimal Rate $\gamma^\star_{\text{mse}}$

Is $\gamma^\star_{\text{mse}}$ in this range?

We saw that:

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

and the range for asymptotic Normality was:

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

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

Therefore, $\gamma^\star_{\text{mse}}$ 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 assynptotically 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 assynptotically 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

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

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 γ 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 $\gamma^{\star}_{\text{mse}}$ 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 $n \to \infty$, each exponent must be negative:

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

and

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

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 γ 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 γ for which inference is possible.

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

In the previous question we arrived to:

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

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

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

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

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

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:

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

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

Therefore, $\gamma^\star_{\text{mse}}$ 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

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

(i) For what range of γ 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 $\gamma^{\star}_{\text{mse}}$ in this range?

(iii) In terms of the allowed γ, 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:

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

As before, this implies

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

And the key additional requirement is:

$\sqrt{n}\|\hat{\mu}_t(\mathbf{x}) - \mu_t(\mathbf{x})\|_2\|\hat{p}_t(\mathbf{x}) - p_t(\mathbf{x})\|_2 \to 0$. 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:

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

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

For this to vanish:

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

and

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

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 γ 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 γ satisfies the conditions.

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

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

Compare $\gamma^\star_{\text{mse}}$ 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 $\gamma^\star_{\text{mse}} > 1/(4(p+1))$.

Next, compare $\gamma^\star_{\text{mse}}$ 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 $d < 2(p+1)$, $\gamma^\star_{\text{mse}}$ also satisfies $\gamma^\star_{\text{mse}} < 1/(2d)$.

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

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

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

(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 γ 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 $Y = Y_1 T + Y_0(1-T)$. 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 $\mu_0(\mathbf{x}) = \boldsymbol{\beta}_0'\mathbf{x}$ 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 $\hat{\tau}_{\mathrm{PI}}$ and $\hat{\tau}_{\mathrm{DR}}$ 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_β, s_0 are fixed?

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

(iii) How does s_β impact $\hat{\tau}_{\mathrm{PI}}$?

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

(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?

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

(i) What happens as n grows but d, s_β, 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_β 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 $\hat{\tau}_{\text{PI}}$ as d and s_0 change for fixed n?

Examining the plug-in estimator $\hat{\tau}_{\text{PI}}$ 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 persence of performance decline of the plug-in estimator. Lowering s_0 (i.e., reducing the number of non-zero coefficients in $\boldsymbol{\theta}_0$) 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 $\hat{\tau}_{\text{DR}}$ in practical applications.

(iii) How does s_β impact $\hat{\tau}_{\text{PI}}$?

As mentioned before, the sparsity level s_β, representing the number of non-zero coefficients in $\boldsymbol{\beta}_0$, significantly influences the performance of the plug-in estimator $\hat{\tau}_{\text{PI}}$. As s_β 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_β to $d/2$ with the same s_θ, the average error rises to 0.1507 and the standard error to 0.1668.
- Further increasing s_β 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_β 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_β 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_β, the plug-in estimator becomes less reliable, highlighting the importance of model sparsity for its effective application.

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

The doubly robust (DR) property of $\hat{\tau}_{\mathrm{DR}}$ 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_β Low) and Misspecified Propensity Score Model (s_θ 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_β High and s_θ 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_β or s_θ 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_β or low s_θ), 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?

(a) Use the Plug-in Estimator When:
- The dimensionality d is low to moderate.
- Models are expected to be well-specified with high sparsity (s_β, s_0 are low).

- Simplicity and computational efficiency are priorities, and model misspecification is unlikely.

(b) 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 $\boldsymbol{\theta}_0'\curvearrowleft$ 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 robus property of $\hat{\tau}_{\text{DR}}$.

(iii) Dicuss what your results mena for applied practice. When would you recommend the different estimators and why?

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 because, since we can still improve 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, specially with further 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 $\hat{\tau}_{\mathrm{PI}}$ and $\hat{\tau}_{\mathrm{DR}}$,

(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.

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 1: 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 2: Estimator Performance under Various Models

Propensity Model	**Outcome Model**	**CrossFitting**	**Tau_PI**	**Tau_DR**
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 3: 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

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 inconsistent. For instance, with the `LogisticRegression` propensity model and `RandomForestRegressor` outcome model, cross-fitting has a minimal effect on $\hat{\tau}_{\text{PI}}$ but slightly alters $\hat{\tau}_{\text{DR}}$. With the `NeuralNetworkClassifier` for propensity and `LassoRegression` for the outcome, cross-fitting changes $\hat{\tau}_{\text{DR}}$ from -0.940 to -1.952, demonstrating a more substantial impact. 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 ($\hat{\tau}_{\text{PI}}$) consistently provides positive estimates across most scenarios, whereas the doubly robust estimator ($\hat{\tau}_{\text{DR}}$) often yields negative values.

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

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

Furthermore, this problem is difficult in its nature: a lot of covariates are postreatment, 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 helps define the covariates (the causal relationship is inverse)

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 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.

```
Results for data source e=1 (observational data):
ATE estimate: -1.060766
95% CI: -1.136993 to -0.9845383

Results for data source e=2 (fully randomized):
ATE estimate: 1.938148
95% CI: 1.869419 to 2.006877

```

Listing 1: 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

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

The 95% confidence interval shows:

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

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

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

The corresponding confidence interval is approximately

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

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

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

causing the observed difference in means

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

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

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

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.

```
Results for data source e=1:
ATE: 1.362815
95% CI: 1.252676 to 1.472955

Results for data source e=2:
ATE: 1.994436
95% CI: 1.958671 to 2.030201

```

Listing 2: 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 2: Descriptive Statistics of Conditional Average Treatment Effects (CATEs)

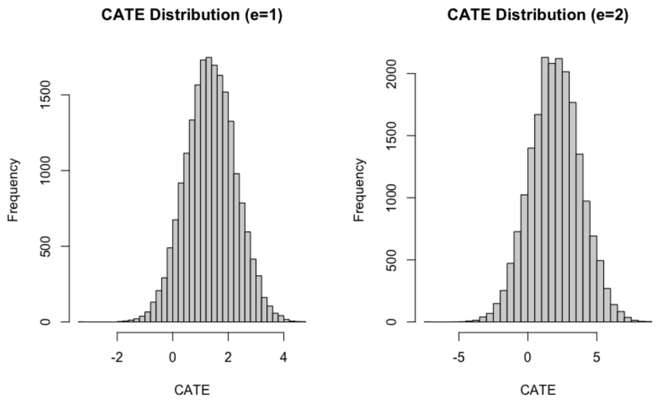

Figure 1: Distribution of Conditional Average Treatment Effects (CATEs) in Each Data Source

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

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

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:

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

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:

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

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

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

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

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

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

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

3.c Doubly Robust Estimation

Combine the estimators of $\mu_t(x) = \mathbb{E}[Y(t) \mid X = x]$ with a parametric logistic regression estimate of the propensity score $p(x) = \mathbb{P}[T = 1 \mid X = x]$ 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 than running a proper double ML function 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

```
Results for data source e=1 (doubly robust, corrected):
ATE: 1.281753
95% CI: 1.202279 to 1.361226

Results for data source e=2 (doubly robust, corrected):
ATE: 1.994437
95% CI: 1.95078 to 2.038093

```

Listing 3: Doubly Robust ATE Estimation

Estimator Using Double Machine Learning

```
Results for data source e=1 (doubly robust, corrected):
coef    std err          t          P>|t|      2.5 %     97.5 %
d   0.445511   0.030286   14.71035   5.532050e-49   0.386152   0.504869

Results for data source e=2 (doubly robust, corrected):
coef    std err           t  P>|t|      2.5 %     97.5 %
d   1.993791   0.022734   87.701211     0.0   1.949233   2.038348
```

Listing 4: 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 $m(X_i) = \mathbb{E}[Y_i \mid X_i]$, $\pi(X_i) = \mathbb{P}[T_i = 1 \mid X_i]$, and $g(X_i) = \mathbb{E}[T_i \mid X_i]$. 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 4: 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 5: 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%	Data Source
RandomForest	0.502296	0.032106	15.644891	3.599873e-55	0.439369	0.565223	e1
GradientBoosting	0.518262	0.032348	16.021308	9.072225e-58	0.454860	0.581663	e1
DeepNN	0.500243	0.032388	15.445140	8.135033e-54	0.436763	0.563723	e1
LASSO	0.443051	0.027604	16.050380	5.681481e-58	0.388948	0.497153	e1

Table 6: 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%**	**Data Source**
RandomForest	2.000420	0.021877	91.438273	0.000000e+00	1.957541	2.043299	e2
GradientBoosting	1.990438	0.021741	91.550747	0.000000e+00	1.947826	2.033051	e2
DeepNN	1.907454	0.021751	87.694617	0.000000e+00	1.864823	1.950085	e2
LASSO	1.993790	0.022269	89.530360	0.000000e+00	1.950142	2.037437	e2

Table 7: 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 coeffient, 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 8: 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:

- $\mathcal{L}_{RCT}(\theta) = \sum_{i:e_i=2}(Y_i - \mu(T_i, X_i; \theta))^2$, capturing the fit of the outcome model for the RCT.
- $\mathcal{L}_{Obs}(\theta, \gamma)$ 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 θ and γ 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

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

```

Listing 5: ATE and Confidence Interval Estimates Ignoring Covariates

Random Forest Estimation Estimation

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

Listing 6: ATE and Confidence Interval Estimates Ignoring Covariates