Assessing the optimality of CATE estimation

Daniel Cunha, Abe Winterscheidt, Shiwen Yang

December 1, 2021

Abstract

As of the 21st century, there is a wealth of observational data in many industries and contexts. Often, researchers/scientists are interested in understanding the causal effect of some underlying mechanism of those data, but without paying for an expensive(and often impossible) randomized control trial. However, under certain modeling assumptions, the observational data can be used to estimate causal effects, or the average treatment effect (ATE), of interventions. Sometimes we wish to know the effect of an intervention in a subpopulation of observations, or the conditional average treatment effect (CATE). In the following paper, we explicate a few estimators of the CATE, and for the DR-Learner, derive theory regarding finite sample bounds on the error of estimation. We end with an observational study of the causal effect of COVID-19 deaths on voter turnout in different racialized populations.

1 Introduction

The present paper is a short explication of the arguments made in *Optimal Doubly Robust Estimation of Heterogeneous Causal Effects* by Edward Kennedy. The general story telling of the paper goes like this. In causal inference, there are two foundational estimators of treatment effect, the conditioning argument plug-in estimator and the inverse probability treatment weighting (IPTW) estimator. Both of these are unbiased estimators of the

expected counterfactual outcome $\mathbb{E}[Y^a]$, that is, the expected outcome Y under the intervention $A = a$ (These estimators will be introduced in section 4). However, when these estimators are used for CATE, they suffer from high variance. It turns out that creating a doubly robust estimator that combines plug-in and IPTW not only reduces the variance in CATE estimation, but also yields interpretable finite error bounds and conditions for oracle efficiency. The paper proceeds by setting up the notation needed for the theorems, then a section to introduce plug-in and IPTW estimators. This is then followed by a section that introduces DR-Learner with derivations of some of its properties, and finally an application of DR-Learner to a voter turnout dataset. Please note that due to the notational complexity of the Theorem statements, we have quoted large portions of them to maintain mathematical integrity.

2 Setup

In this setup, we will introduce some notation, and functions that are needed for doing CATE analysis. Start by noting $X \in \mathbb{R}^d$ are covariates, $A \in \{0, 1\}$ is a binary indicator for treatment assignment, and $Y \in \mathbb{R}^d$ are outcome of interest. These three components make up a sample of observation $Z_i = (X_i, A_i, Y_i)$. In order to complete a CATE analysis, we will need the following estimands:

$$\pi(x) = P(A = 1|X = x)$$
$$\mu_a(x) = \mathbb{E}(Y|X = x, A = a)$$
$$\eta(x) = \mathbb{E}(Y|X = x)$$
$$\tau(x) = \mu_1(x) - \mu_0(x)$$

To put this into context, suppose we are trying to study whether a drug can have a

positive impact on the prognosis of some disease. Then as an illustrative example:

1. X can be the age of a patient, and then A is whether the patient is given the drug, and Y can be whether the patient is alive or dead in 3 years.
2. $\pi(x)$ gives the probability of an x-year-old patient being given the drug.
3. $\mu_a(x)$ is the probability of an x-year-old patient surviving in 3 years given the status of whether this patient were given the drug or not.
4. $\eta(x)$ is the marginal probability of an x-year-old patient surviving in 3 years in general.
5. $\tau(x)$ is the conditional average treatment effect(CATE), measures how much the drug has, hopefully, increased the survival probability among patients that are x-year-old patients.

3 Notation

We will use the following definitions throughout the remainder of the paper:

1. $\mathbb{P}_n(f) = \frac{1}{n}\sum_{i=1}^{n} f(Z_i)$
2. for $x \in \mathbb{R}^d, ||x||$ denotes the Euclidean norm of x given by $||x||^2 = \sum_{i=1}^{d} x_i^2$
3. for a function f, $||f||$ denotes the $L_2(\mathbb{P})$ of f, given by $||f||^2 = \int f^2(z) d\mathbb{P}(z)$
4. for two sequences a_n, b_n, $a_n \sim b_n$ if $\lim_{n\to\infty} \frac{a_n}{b_n} = 1$

For functions f, g and $\forall x$ in the domain of f, g:

1. $f \underset{\sim}{\leq} g$ if $\exists C \in \mathbb{R}$ such that $f(x) \leq Cg(x)$

2. $f \asymp g$ if $\exists c, C \in \mathbb{R}$ such that $cg(x) \leq f(x) \leq Cg(x)$

Furthermore, in this paper, the function f that we will be referring to, is a $\lfloor s \rfloor$-continuously differentiable function with bounded partial derivatives and also satisfies

$$|D^m f(x) - D^m f(x')| \lesssim ||x - x'||^{s - \lfloor s \rfloor} \quad \forall x, x' \tag{1}$$

where $m = \{m_1, ..., m_d\}$, such that $\sum_{i=1}^{d} m_i = \lfloor s \rfloor$, and $D^m = \frac{\partial^{\lfloor s \rfloor}}{\partial^{m_1} x_1 ... \partial^{m_d} x_d}$. We will assume these smoothness properties of the underlying functions π, ν, μ, τ in order to derive error bounds on estimation of different CATE estimators, and compare those errors to an oracle.

4 Assumptions for counterfactual estimation

A counterfactual is a process that would have happened resulting from an intervention, basically a "what if?" scenario. For example, suppose we observe the following covariates from hospital data, vaccination status $A = a$, age X, and viral contraction Y. Then an example of a counterfactual estimation problem is "What would the expected number of viral contractions be, had everyone been vaccinated?". We represent this counterfactual as $Y^{(a)}$. Notice how this is different from answering the question, "What is the expected viral contraction given the patient received the vaccine?". There are two main assumptions needed to perform counterfactual estimation. The first is **consistency** of treatment assignment, $Y^{(a)} = Y|A = a$. That is, the counterfactual outcome and the observed outcome are equivalent when the treatment assignment is the same for both. The second assumption is **No Unobserved Confounding Assumption** (NUCA), which states there are no observed covariates that predict both treatment and the outcome, $Y^{(a)} \perp A|X$. We will use these assumptions to derive meaningful estimators of counterfactual outcomes.

5 The plug-in estimator and IPTW

5.1 The plug-in estimator via a conditioning argument

The plug-in estimator of a counterfactual outcome builds off of the following unbiasedness argument. Since the counterfactual outcome $Y^{(a)}$ is unobserved, we cannot hope to have an unbiased estimator for it. The next best we can hope for is an unbiased estimate of its expectation $\mathbb{E}[Y^{(a)}]$.

$$\begin{aligned}
\mathbb{E}[Y^{(a)}] &= \mathbb{E}_X[\mathbb{E}[Y^{(a)}|X = x]] && \text{(Tower property of expectations)} \\
&= \mathbb{E}_X[\mathbb{E}[Y^{(a)}|X = x, A = a]] && \text{(NUCA } Y^{(a)} \perp A|X) \\
&= \mathbb{E}_X[\mathbb{E}[Y|X = x, A = a]] && \text{(Consistency } Y = Y^{(a)}|A = a)
\end{aligned}$$

Notice how the assumptions of NUCA and consistency afford us the relationship $\mathbb{E}[Y^{(a)}] = \mathbb{E}_X[\mathbb{E}[Y|X = x, A = a]]$. This final term can be estimated the following way. The inner expectation $\mathbb{E}[Y|X = x, A = a]$ can be estimated with a regression $\hat{\mathbb{E}}[Y|X = x, A = a]$, and the outer expectation can be estimated using a Monte Carlo argument,

$$\hat{\mathbb{E}}[Y^{(a)}] = \frac{1}{n}\sum_{i=1}^{n} \hat{\mathbb{E}}[Y_i|X_i, A_i = a]$$

5.2 Inverse probability treatment weighting

Another approach to estimating counterfactuals is the inverse probability treatment weighting approach. The motivation for the approach is similar to the plug-in estimator, in that

both are unbiased estimators of $\mathbb{E}[Y^{(a)}]$. In order to see this, note the following,

$$\begin{aligned}
\mathbb{E}[\frac{Y * 1_{A=a}}{P(A=a|X)}] &= \mathbb{E}_X \mathbb{E}[\frac{Y * 1_{A=a}}{P(A=a|X)}|X] \\
&= \mathbb{E}_X \frac{1}{P(A=a|X)} \mathbb{E}[Y 1_{A=a}|X] \\
&= \mathbb{E}_X \frac{1}{P(A=a|X)} \mathbb{E}[Y^{(a)} 1_{A=a}|X] && \text{(Consistency)} \\
&= \mathbb{E}_X \frac{1}{P(A=a|X)} \mathbb{E}[Y^{(a)}|X] \mathbb{E}[1_{A=a}|X] && \text{(NUCA)} \\
&= \mathbb{E}_X \frac{1}{P(A=a|X)} \mathbb{E}[Y^{(a)}|X] P(A=a|X) \\
&= \mathbb{E}_X \mathbb{E}[Y^{(a)}|X] \\
&= \mathbb{E}[Y^{(a)}]
\end{aligned}$$

With a similar argument for $\mathbb{E}[\frac{1_{A=a}}{P(A=a|X)}] = 1$. Using these, we have the following Monte Carlo weight stabilized estimator,

$$\hat{\mathbb{E}}[Y^{(a)}] = \frac{\frac{1}{n}\sum_{i=1}^{n} \frac{Y_i 1_{A_i=a}}{\hat{P}(A_i=a|X_i)}}{\frac{1}{n}\sum_{i=1}^{n} \frac{1_{A_i=a}}{\hat{P}(A_i=a|X_i)}}$$

Where $\hat{P}(A_i = a|X_i)$ is a regression of the propensity score, or probability of treatment assignment.

6 Oracle Inequality

Since the plug-in estimator and IPTW estimator have high MSE, the author goes on to show that by combining sample splitting with a doubly robust estimator(i.e. that combines the plug-in and IPTW), they are able to derive explicit error bounds and reason about oracle efficiency. We start by defining a set of first and second stage regression functions, and

make a set of stability assumptions about those functions. We then follow with an oracle inequality that will enable us to reason about the optimality of various estimators.

Stability conditions. Let $\hat{\mathbb{E}}_n[Y|X=x]$ denote a generic estimator of the regression function $\mathbb{E}_n[Y|X=x]$, using the test data $(X_i, Y_i) \subseteq Z_i, i = 1, ..., n$. Then define the stability conditions of $\hat{\mathbb{E}}_n$ as follows,

1. $\hat{\mathbb{E}}_n[Y|X=x] + c = \hat{\mathbb{E}}_n[Y+c|X=x]$, $\forall c \in \mathbb{R}$

2. If $\mathbb{E}[W|X=x] = \mathbb{E}[Y|X=x]$ then
   $\mathbb{E}[(\hat{\mathbb{E}}_n[Y|X=x] - \mathbb{E}[Y|X=x])^2] \asymp \mathbb{E}[(\hat{\mathbb{E}}_n[W|X=x] - \mathbb{E}[W|X=x])^2]$

As the author notes, these stability conditions are reasonable, and enable a much cleaner approach to proving the following theorems.

Theorem 1. Suppose $Z_0^n = (Z_{01}, ..., Z_{0n})$ and $Z^n = (Z_1, ..., Z_n)$ are independent training and test samples, respectively. Let $\hat{f}(z) = \hat{f}(z; Z_0^n)$ be an estimate of a function $f(z)$ using only the training data Z_0^n, and define $m(x) \equiv \mathbb{E}[f(Z)|X=x]$. Let $\hat{m}(x) = \hat{\mathbb{E}}_n[\hat{f}(Z)|X=x]$ denote the regression of $\hat{f}(Z)$ on X in the test samples, and let $\tilde{m}(x) = \hat{\mathbb{E}}_n[f(Z)|X=x]$ denote the corresponding oracle regression of $f(Z)$ on X. Define the error function $\hat{r}(x) = \hat{r}(x; Z_0^n) \equiv \mathbb{E}[\hat{f}(Z)|X=x, Z_0^n] - m(x)$.

Then under stability conditions stated above, the following holds,

$$\mathbb{E}[(\hat{m}(x) - m(x))^2] \lesssim \mathbb{E}[(\tilde{m}(x) - m(x))^2] + \mathbb{E}[\hat{r}(x)^2]$$

■

This result is particularly helpful for error analysis, because it decomposes the error obtained by the oracle from the remaining error due to noise in nuisance parameter estimation. As we will see later, this decomposition of the MSE enables us to reason about the oracle efficiency of estimators. For example, if we can show $\mathbb{E}[\hat{r}(x)^2]$ goes to zero faster than

$\mathbb{E}[(\tilde{m}(x) - m(x))^2]$, then the estimator is oracle efficient. Now that we have established a generalized error bound for pseudo-outcome regression, we will introduce the DR-Learner. Recall that the doubly robust estimator, through the machinery of unbiased estimating equations, achieves asymptotic normality and consistency. By combining the DR-Learner with sample splitting, we are able to apply Theorem 1 to compute error bounds on its estimation. Let us start by characterizing the estimation algorithm, and then extend Theorem 1 to apply specifically to DR-Learner.

Algorithm 1 (DR-Learner): $(D_{1a}^n, D_{1b}^n, D_2^n)$denote three independent samples of n observations of $Z_i = (X_i, A_i, Y_i)$.

1. Nuisance training:

   (a) Construct estimates $\widehat{\pi}$ of the propensity scores π using D_{1a}^n.

   (b) Construct estimates $(\widehat{\mu}_0, \widehat{\mu}_1)$ of the regression functions (μ_0, μ_1) using D_{1b}^n.

2. Pseudo-outcome regression: Construct the pseudo-outcome

$$\widehat{\varphi}(Z) = \frac{A - \widehat{\pi}(X)}{\widehat{\pi}(X)\{1 - \widehat{\pi}(X)\}} \{Y - \widehat{\mu}_A(X)\} + \widehat{\mu}_1(X) - \widehat{\mu}_0(X)$$

   and regress it on covariates X in the test sample D_2^n, yielding

$$\widehat{\tau}_{dr}(x) = \widehat{\mathbb{E}}_n\{\widehat{\varphi}(Z) \mid X = x\}$$

3. Cross-fitting (optional): Repeat Step 1-2 twice, first using (D_{1b}^n, D_2^n) for nuisance training and D_{1a}^n as the test sample, and then using (D_{1a}^n, D_2^n) for training and D_{1b}^n as the test sample. Use the average of the resulting three estimators as a final estimate of τ.

■

Let $\widehat{\psi}(Z) = \frac{A-\widehat{\pi}(X)}{\widehat{\pi}(X)\{1-\widehat{\pi}(X)\}}\{Y - \widehat{\mu}_A(X)\}$, then $\widehat{\phi}(Z) = \widehat{\psi}(Z) + \tau(X)$. We shall make the following observations:

1. $\mathbb{E}(\widehat{\psi}(Z)|X = x) = 0$ because $\mathbb{E}(A|X = x) = P(A = 1|X = x) = \mathbb{E}(\widehat{\pi}(x))$. Therefore we have $\widehat{\tau}_{dr}(x) = \widehat{\mathbb{E}}_n\{\widehat{\varphi}(Z) \mid X = x\}$.
2. $\widehat{\psi}(Z) = \begin{cases} \frac{Y-\widehat{\mu}_0(X)}{1-\widehat{\pi}(X)} & \text{when } A = 0 \\ \frac{Y-\widehat{\mu}_1(X)}{\widehat{\pi}(X)} & \text{when } A = 1 \end{cases}$ Here we are weighting each error, $Y - \widehat{\mu}_0(X)$ using its probability.

Now with the DR-Learner defined, we are in a position of extending Theorem 1 to this specific estimator. The oracle in this case is aware of the nuisance functions π and μ. So we can replace the MSE of the oracle with the estimation error as if π and μ are known, finally adding the error function for the uncertainty in π and μ.

Theorem 2. Let $\widehat{\tau}_{dr}(x)$ denote the DR-Learner estimator detailed in Algorithm 1. Assume:

1. The propensity score estimates are bounded as $\epsilon \leq \widehat{\pi}(x) \leq 1 - \epsilon$ for some $\epsilon > 0$ wp1. second-stage regression $\widehat{\mathbb{E}}_n(\cdot \mid X = x)$ satisfies Assumptions $1 - 2$ of Theorem 1 .

Let $\widetilde{\tau}(x) = \widehat{\mathbb{E}}_n\left(Y^1 - Y^0 \mid X = x\right)$ denote an oracle estimator that regresses the difference $(Y^1 - Y^0)$ on X, where Y^a is such that $\mathbb{P}\left(Y^a \leq y \mid X = x\right) = \mathbb{P}(Y \leq y \mid X = x, A = a)$. Then

$$\mathbb{E}\left[\{\widehat{\tau}_{dr}(x) - \tau(x)\}^2\right] \lesssim R^*(x) + \mathbb{E}\left[\{\widehat{\pi}(x) - \pi(x)\}^2\right] \sum_{a=0}^{1} \mathbb{E}\left[\{\widehat{\mu}_a(x) - \mu_a(x)\}^2\right]$$

where $R^*(x) = \mathbb{E}\left[\{\widetilde{\tau}(x) - \tau(x)\}^2\right]$ is the oracle risk. ■

The importance of this theorem is that, the error bound we obtain is independent from any model. It only requires the second-stage regression to satisfy some very reasonable assumptions given in Theorem 1. This is powerful because then we can apply this to nuisance functions and CATE that has known bounds for their MSE, for example, local polynomial. The next corollary demonstrates this point.

Corollary 1. Suppose the assumptions of Theorem 2 hold. Further assume:

1. The propensity score π is α-smooth, and is estimated with squared error $n^{\frac{-2\alpha}{2\alpha+d}}$.
2. The regression functions μ_a are β-smooth, and are estimated with squared error $n^{\frac{-2\beta}{2\beta+d}}$.
3. The CATE τ is γ-smooth.

If the second-stage estimator $\hat{\mathbb{E}}_n$ yields the minimax optimal squared error rate $n^{\frac{-2\gamma}{2\gamma+d}}$ for γ-smooth functions, then

$$\mathbb{E}[(\hat{\tau}_{dr}(x) - \tau(x))^2] \lesssim n^{\frac{-2\gamma}{2\gamma+d}} + n^{\frac{-2\alpha}{2\alpha+d}+\frac{-2\beta}{2\beta+d}}$$

And thus the DR-Learner is oracle efficient if

$$\alpha\beta \geq \frac{d^2}{4} - \frac{(\alpha+\frac{d}{2})(\beta+\frac{d}{2})}{1+\frac{2\gamma}{d}}$$

■

7 Example: Race, COVID-19, and Voter Turnout

7.1 Introduction

Race is one of the most salient factors in American politics, perhaps second only to partisanship. Racial disparities occur across almost every facet of the political system, from

education to voting to officeholders, and to voter turnout (eg. Fraga 2015, Hill Leighley 1999). Simultaneously, the current COVID-19 pandemic has become a huge factor affecting voters. The crisis led many states to implement remote-voting measure, for the 2020 election and many other states to pass anti-voting laws in retaliation soon thereafter (Flanders et al. 2020). All three of these variables are intrinsically linked – COVID-19 affected minority communities disproportionately, particularly, Native, Latino, and Black communities in the first year of the pandemic; though, political fallacies among white conservatives have helped make up some of that gap since the vaccination became widely available (APM Research 2021). In this section of the paper, we explore a single fundamental question regarding these three variables: how have pandemic fatality rates affect turnout among different races.

7.2 Methodology

Using a racial binary as our "treatment," we conduct three cross-fitted CATE models to derive $\tau(x)$ estimates at different fatality rates. Three binary treatments are used, *wb*, *wa*, *wh*. In each, white voters are treated as the control, where $A = 0$, and one of three minority groups (Blacks, Asians, and Hispanics) are the "treatment" where $A = 1$. Data is collected at the state level using CDC and Census data for fatality rates and turnout by race, respectively. Importantly, while the CDC provides fatality data for Whites, Blacks, Asians, Hispanics, Hawaiian and Pacific Islander, and Native Americans/Indigenous peoples, the Census only provides turnout data for the first four groups. All data is collected with the 2020 election in mind, so fatality rates were aggregated into a single variable by month from January 1, 2020 to October 31, 2020; data is also collected at the state level, providing $n = 200$ overall and $n = 100$ for each model (50 states, each with data for white voters and voters of the minority population of interest).

Models were designed following the DR-Learner algorithm provided in Kennedy (2020). Code for these models is provided in Appendix A. In interpreting the results of the models, one important theoretical note is that the direction of causality between race and COVID deaths. As implied in the CATE model, when looking for a causal relationships, A should be an intervention causing Y among subgroups X and the effects of A should vary due to X. However, in reality, race is likely to play an important role in COVID fatality rates. Due to the importance of race as a "treatment" in political science literature, and the difficulty of converting fatality rates into a functional binary variable while still allowing it to capture the nuance of the data, we chose to proceed with the aforementioned method, despite its theoretical shortcomings.

7.3 Results

Interpreting the results, we find a similar tend among all three models: a minimum difference of turnout somewhere around the median fatality rate followed by a steep negative increase for lower fatality rates (the data are almost entirely negative, and as $\tau(x_i)$ indicates a difference in turnout between black and white voters, the humps indicate minimums as they are closer to zero). The trend is most pronounced among black voters and least pronounced among Asians, who have a majority of their τs indistinguishable from zero, indicating very similar turnout between Asian and White voters until a steep decrease around the median.

7.4 Discussion

Overall, the CATE provides evidence of voter turnout differences between all four races which vary across the range of COVID-19 death rates. There are some major flaws with the

results obtained from the DR-Learner method. First, and most importantly, the splitting of data had a huge impact on the τ values. Depending on which observations were grouped where, the shape of the curve would change dramatically.[1] This variation may be the result of limited data, with each model only provided 100 observations. Second, in the models for Asian and Hispanic populations, particularly the latter, there were huge τs. Some models, depending on sub-grouping, had τ values in the negative thousands. Obviously, as all the continuous data is bound between $[0, 1]$, these differences are clearly flawed. Having examined the code and the data multiple times, the only cause we could find for these issues were smaller $\hat{\pi}$ values, which may inflate the first term from the denominator, though there wasn't a clear consistency among low $\hat{\pi}$ estimates and high τs. Interestingly, despite being derived from the same base code, this issue did not appear to affect the model on Black voters and was not as severe or persistent in the models of Hispanic voters. Obviously, these two issues present a clear and present danger for the viability of the DR-Learner CATE algorithm, flaws that warrant closer examination than this paper can provide given its limited scope in question. That said, it is reasonable to consider the possibility that these flaws may be specific to the data and models presented here and may not expand to other areas of interest; though Kennedy (2020)'s limited example, based on narrowly simulated data designed for the sole purpose of determining the oracle estimator leaves a bit to be desired in terms of the approach's true applicability outside of a theoretical discussion.

[1] See **Appendix B** for additional, almost all of which tell a different story, along with the results for when all steps are run on all the data

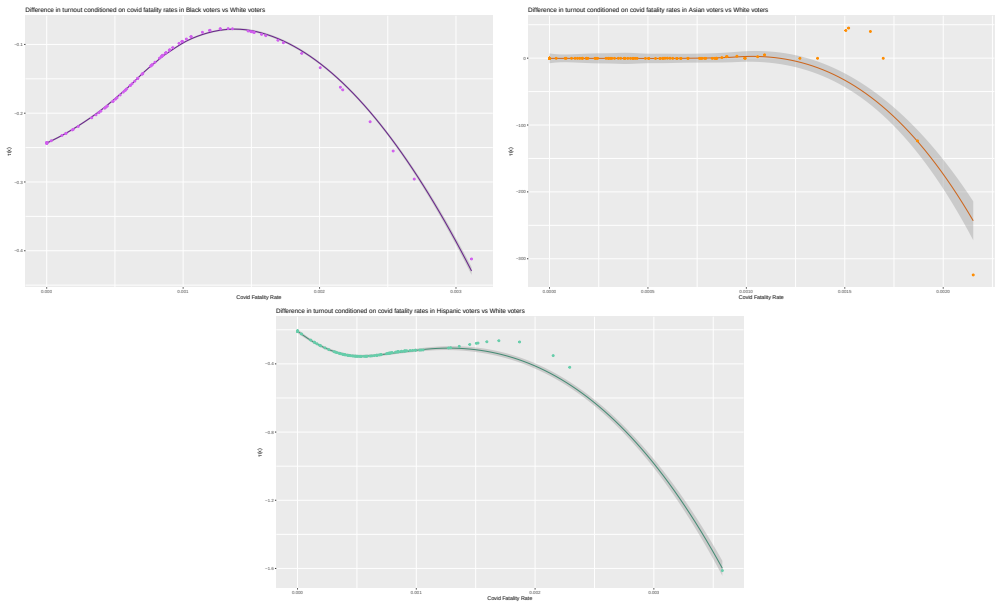


Figure 1: $\tau(x)$ values in Blacks, Asians, and Hispanics Across COVID Fatality Rates

Bibliography

Kennedy, E. 2020. "Optimal Doubly Robust Estimation of Heterogeneous Causal Effects." https://arxiv.org/pdf/2004.14497.pdf/

APM Research. 2021. "Color of Cornonavirus: COVID-19 Deaths by Race and Ethnicity". *APM Reseatch Lab.*

Flanders, W. Dana, William D. Flanders, and Michael Goodman. 2020. "The association of voter turnout with county-level coronavirus disease 2019 occurrence early in the pandemic." *Annals of Epidemiology*, 49(1): 42-29.

Fraga, Bernard L. 2015. "Redistricting and the Causal Impact of Race on Voter Turnout." *The Journal of Politics*, 78(1): 19-34.

Hill, Kim Quaile and Jan E. Leighley. 1999. "Racial Diversity, Voter Turnout, and Mobi-

lizing Institutions in the United States." *American Politics Quarterly*, 27(3): 275-295

Appendix A: Code

```
library(readxl)
library(aod)
library(ggplot2)
library(readxl)
library(dplyr)
data <- read_excel("Desktop/FILE_5188.xlsx")
d1 = subset(data,wb!=2)
d2 = subset(data,wa!=2)
d3 = subset(data,wh!=2)



grouper <- function(df, n) {

        # create a random number for each row
        random <- sample(1:nrow(df), replace =
            FALSE, nrow(df))

        # divide the random number by the group
            size
        d<- df$group_number <- ceiling(random / (
            nrow(df) / n))
```

```
        return(d)
}

d1$s<-grouper(d1,3)
d2$s<-grouper(d2,3)
d3$s<-grouper(d3,3)



d1a= subset(d1,s==1)
d2a= subset(d2,s==1)
d3a= subset(d3,s==1)

d1b= subset(d1,s==2)
d2b= subset(d2,s==2)
d3b= subset(d3,s==2)

d1c= subset(d1,s==3)
d2c= subset(d2,s==3)
d3c= subset(d3,s==3)



#Black
#1
pihat <- predict(glm(wb~pd,family=binomial(link="
    logit")),
```

```
                                    data=d1a),newdata=d1,type
                                        ='response')
mu1hat <- predict(smooth.spline(d1$pd[d1$wb==1&d1
    $s==2],
                                    d1$pt[d1$wb==1&d1$s==2]),
                                        d1$pd)$y
mu0hat <- predict(smooth.spline(d1$pd[d1$wb==0&d1
    $s==2],
                                    d1$pt[d1$wb==0&d1$s==2]),
                                        d1$pd)$y
## construct estimators
d1$pseudo <- ((d1$wb-pihat)/(pihat*(1-pihat)))*
                    (d1$pt-d1$wb*mu1hat-(1-d1$wb)*
                        mu0hat) + mu1hat-mu0hat


d1$drl <- predict(smooth.spline(d1$pd[d1$s==3],
                    d1$pseudo[d1$s==3]),d1$pd)$y
#2
pihatb <- predict(glm(wb~pd,family=binomial(link=
    "logit"),
                    data=d1b),newdata=d1,type='
                        response')
mu1hatb <- predict(smooth.spline(d1$pd[d1$wb==1&
    d1$s==3],
                    d1$pt[d1$wb==1&d1$s==3]),d1$pd)$y
```

```
mu0hatb <- predict(smooth.spline(d1$pd[d1$wb==0&
    d1$s==3],
                    d1$pt[d1$wb==0&d1$s==3]),d1$pd)$y

d1$pseudo2 <- ((d1$wb-pihatb)/(pihatb*(1-pihatb))
    )*
                    (d1$pt-d1$wb*mu1hatb-(1-d1$wb)*
                        mu0hatb) + mu1hatb-mu0hatb

d1$dr12 <- predict(smooth.spline(d1$pd[d1$s==1],
                    d1$pseudo2[d1$s==1]),d1$pd)$y
#3
pihatc <- predict(glm(wb~pd,family=binomial(link=
    "logit"),
                    data=d1a),newdata=d1,type='
                        response')
mu1hatc <- predict(smooth.spline(d1$pd[d1$wb==1&
    d1$s==3],
                    d1$pt[d1$wb==1&d1$s==3]),d1$pd)$y
mu0hatc <- predict(smooth.spline(d1$pd[d1$wb==0&
    d1$s==3],
                    d1$pt[d1$wb==0&d1$s==3]),d1$pd)$y

d1$pseudo3 <- ((d1$wb-pihatc)/(pihatc*(1-pihatc))
    )*
```

```
                                    (d1$pt-d1$wb*mu1hatc-(1-d1$wb)*
                                        mu0hatc) + mu1hatc-mu0hatc

                    d1$drl3 <- predict(smooth.spline(d1$pd[d1$s==2],
                                    d1$pseudo3[d1$s==2]),d1$pd)$y

                    for (i in 1:100) {
                            d1$drlavg[i]<-(d1$drl[i]+d1$drl2[i]+d1$
                                drl3[i])/3
                    }
                    ggplot(data=d1,aes(pd,drlavg))+
                    geom_smooth(show.legend = TRUE, colour="
                        darkorchid4")+
                    geom_point(colour = "mediumorchid2")+
                    xlab("Covid Fatality Rate")+
                    ylab(expression(paste(tau,"(x)")))+
                    ggtitle("Difference in turnout conditioned on
        covid fatality rates in Black
        voters vs White voters")

                    #Asian
                    #1
                    pihat2 <- predict(glm(wa~pd,family=binomial(link=
                        "logit"),
```

```
                    data=d2a),newdata=d2,type='
                        response')
mu1hat2 <- predict(smooth.spline(d2$pd[d2$wa==1&
    d2$s==2],
                    d2$pt[d2$wa==1&d2$s==2]),d2$pd)$y
mu0hat2 <- predict(smooth.spline(d2$pd[d2$wa==0&
    d2$s==2],
                    d2$pt[d2$wa==0&d2$s==2]),d2$pd)$y

d2$pseudo <- ((d2$wa-pihat2)/(pihat2*(1-pihat2)))
    *
                    (d2$pt-d2$wa*mu1hat2-(1-d2$wa)*
                        mu0hat2) + mu1hat2-mu0hat2

d2$drl <- predict(smooth.spline(d2$pd[d2$s==3],
                    d2$pseudo[d2$s==3]),d2$pd)$y

#2
pihat2b <- predict(glm(wa~pd,family=binomial(link
    ="logit"),
                    data=d2b),newdata=d2,type='
                        response')
mu1hat2b <- predict(smooth.spline(d2$pd[d2$wa==1&
    d2$s==3],
                    d2$pt[d2$wa==1&d2$s==3]),d2$pd)$y
```

```
mu0hat2b <- predict(smooth.spline(d2$pd[d2$wa==0&
    d2$s==3],
                    d2$pt[d2$wa==0&d2$s==3]),d2$pd)$y

d2$pseudo2 <- ((d2$wa-pihat2b)/(pihat2b*(1-
    pihat2b)))*
                    (d2$pt-d2$wa*mu1hat2b-(1-d2$wa)*
                        mu0hat2b) + mu1hat2b-mu0hat2b

d2$dr12 <- predict(smooth.spline(d2$pd[d2$s==1],
                    d2$pseudo2[d2$s==1]),d2$pd)$y

#3
pihat2c <- predict(glm(wa~pd,family=binomial(link
    ="logit"),
                    data=d2a),newdata=d2,type='
                        response')
mu1hat2c <- predict(smooth.spline(d2$pd[d2$wa==1&
    d2$s==3],
                    d2$pt[d2$wa==1&d2$s==3]),d2$pd)$y
mu0hat2c <- predict(smooth.spline(d2$pd[d2$wa==0&
    d2$s==3],
                    d2$pt[d2$wa==0&d2$s==3]),d2$pd)$y
```

```
d2$pseudo3 <- ((d2$wa-pihat2c)/(pihat2c*(1-
    pihat2c)))*
                    (d2$pt-d2$wa*mu1hat2c-(1-d2$wa)*
                        mu0hat2c) + mu1hat2c-mu0hat2c

d2$drl3 <- predict(smooth.spline(d2$pd[d2$s==2],
                    d2$pseudo3[d2$s==2]),d2$pd)$y

for (i in 1:100) {
          d2$drlavg[i]<-(d2$drl[i]+d2$drl2[i]+d2$
              drl3[i])/3
}
ggplot(data=d2,aes(pd,drlavg))+
geom_smooth(show.legend = TRUE, colour="
    chocolate3")+
geom_point(colour = "darkorange")+
xlab("Covid Fatality Rate")+
ylab(expression(paste(tau,"(x)")))+
ggtitle("Difference in turnout conditioned on
                    covid fatality rates in Asian
    voters vs White voters")


##Hispanic
#1
```

```
pihat3 <- predict(glm(wh~pd,family=binomial(link=
    "logit"),
                  data=d3a),newdata=d3,type='
                      response')
mu1hat3 <- predict(smooth.spline(d3$pd[d3$wh==1&
    d3$s==2],
                  d3$pt[d3$wh==1&d3$s==2]),d3$pd)$y
mu0hat3 <- predict(smooth.spline(d3$pd[d3$wh==0&
    d3$s==2],
                  d3$pt[d3$wh==0&d3$s==2]),d3$pd)$y


d3$pseudo <- ((d3$wh-pihat3)/(pihat3*(1-pihat3)))
    *
                  (d3$pt-d3$wh*mu1hat3-(1-d3$wh)*
                      mu0hat3) + mu1hat3-mu0hat3


d3$drl <- predict(smooth.spline(d3$pd[d3$s==3],
                  d3$pseudo[d3$s==3]),d3$pd)$y
#2
pihat3b <- predict(glm(wh~pd,family=binomial(link
    ="logit"),
                  data=d3b),newdata=d3,type='
                      response')
mu1hat3b <- predict(smooth.spline(d3$pd[d3$wh==1&
    d3$s==3],
```

```
                    d3$pt[d3$wh==1&d3$s==3]),d3$pd)$y
mu0hat3b <- predict(smooth.spline(d3$pd[d3$wh==0&
    d3$s==3],
                    d3$pt[d3$wh==0&d3$s==3]),d3$pd)$y

d3$pseudo2 <- ((d3$wh-pihat3b)/(pihat3b*(1-
    pihat3b)))*
                    (d3$pt-d3$wh*mu1hat3b-(1-d3$wh)*
                        mu0hat3b) + mu1hat3b-mu0hat3b

d3$dr12 <- predict(smooth.spline(d3$pd[d3$s==1],
                    d3$pseudo2[d3$s==1]),d3$pd)$y

#3

pihat3c <- predict(glm(wh~pd,family=binomial(link
    ="logit"),
                    data=d3a),newdata=d3,type='
                        response')
mu1hat3c <- predict(smooth.spline(d3$pd[d3$wh==1&
    d3$s==3],
                    d3$pt[d3$wh==1&d3$s==3]),d3$pd)$y
mu0hat3c <- predict(smooth.spline(d3$pd[d3$wh==0&
    d3$s==3],
                    d3$pt[d3$wh==0&d3$s==3]),d3$pd)$y
```

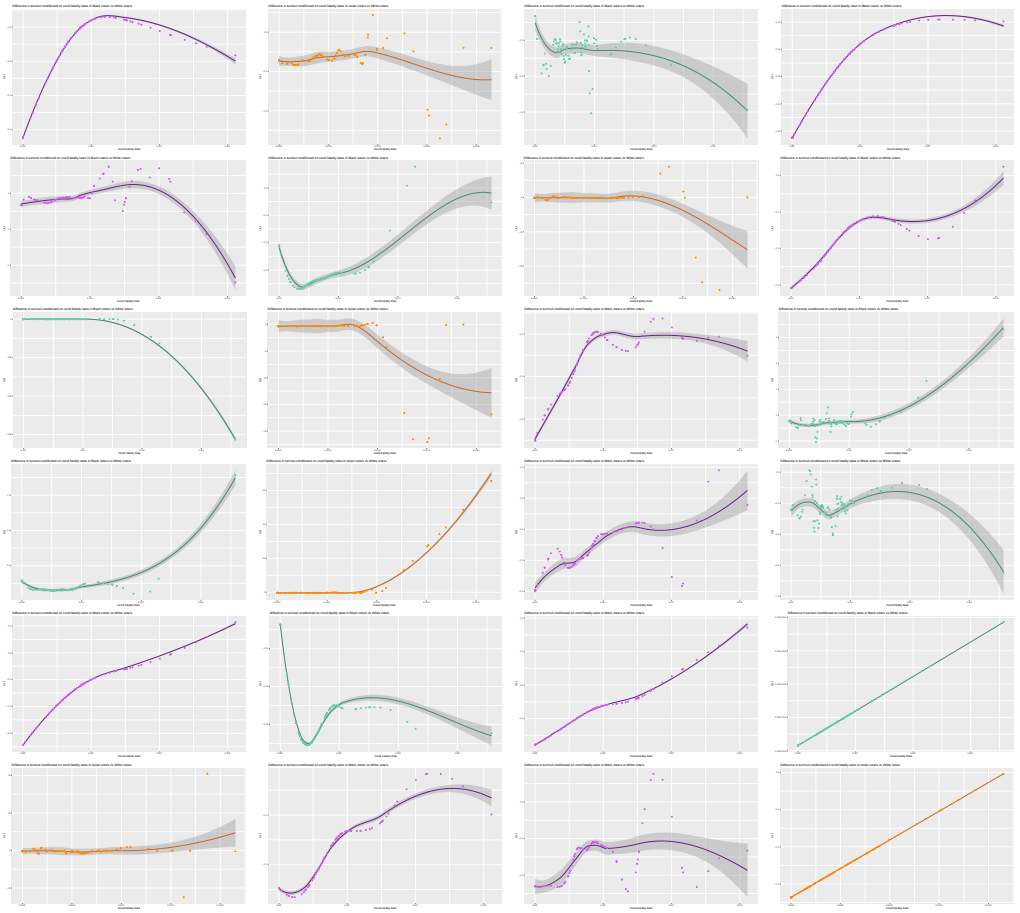
```
d3$pseudo3 <- ((d3$wh-pihat3c)/(pihat3c*(1-
    pihat3c)))*
                    (d3$pt-d3$wh*mu1hat3c-(1-d3$wh)*
                        mu0hat3c) + mu1hat3c-mu0hat3c


d3$drl3 <- predict(smooth.spline(d3$pd[d3$s==2],
                    d3$pseudo3[d3$s==2]),d3$pd)$y
for (i in 1:100) {
          d3$drlavg[i]<-(d3$drl[i]+d3$drl2[i]+d3$
              drl3[i])/3
}
ggplot(data=d3,aes(pd,drlavg))+
geom_smooth(show.legend = TRUE, colour="
    aquamarine4")+
geom_point(colour = "mediumaquamarine")+
xlab("Covid Fatality Rate")+
ylab(expression(paste(tau,"(x)")))+
ggtitle("Difference in turnout conditioned on
        covid fatality rates in Hispanic
    voters vs White voters")
```

. . .

Appendix B: Flawed Outputs

Provided below are some of the various $\tau(x_i)$ estimates generated using different random data samples for cross-fitting. All models were provided by the same data. Make note of the near linear fits and the models that completely invert the direction of the difference (ie. τs tending upwards at higher fatality rates)



Below are the results when the DR-Learner method was applied without sample splitting:

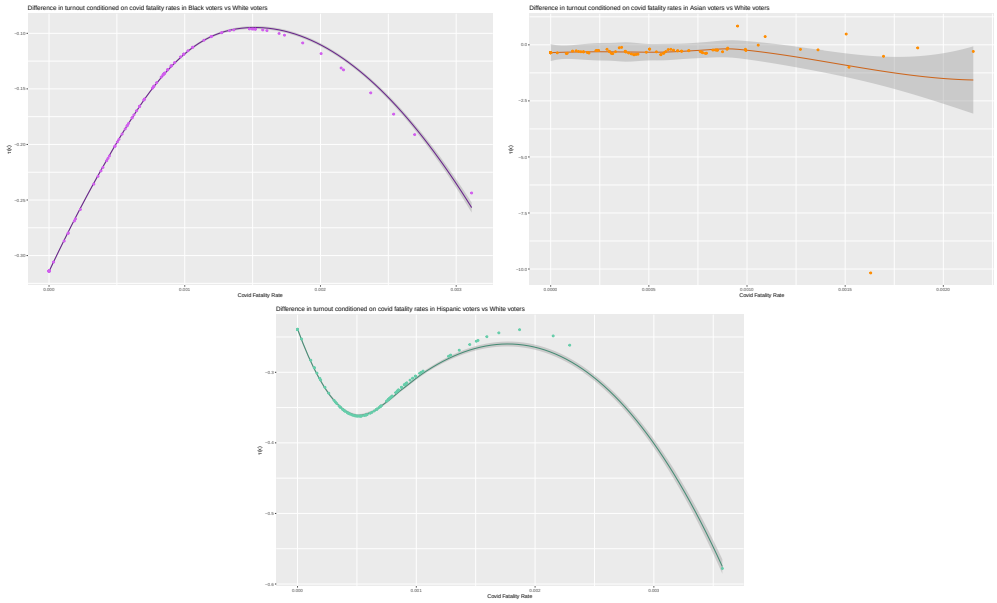


Figure 2: Models created without data splitting