ECON 293/MGTECON 634: Machine Learning and Causal Inference

Stefan Wager
Stanford University

Lecture 2: Average Treatment Effects

12 April 2018

A central goal of machine learning is to understand **what usually happens** in a given situation, e.g.,

- Given today's weather, what's the chance tomorrow's air pollution levels will be dangerously high?

Most economists want to predict **what would happen** if we changed the system, e.g.,

- How does the answer to the above question change if we reduce the number of cars on the road?

This class is about the interface of causal inference and machine learning, with both terms understood broadly:

- Our discussion of **causal inference** draws from a long tradition in economics and epidemiology on which questions about **counterfactuals** can be answered using a given type of data, and how these estimands can be **interpreted**.
- We use the term **machine learning** to describe an engineering heavy approach to data analysis. Given a well-defined task in which good performance can be **empirically validated**, we do not shy away from **computationally heavy** tools or **potentially heuristic** approaches (e.g., decision trees, neural networks, non-convex optimization).

Today's lecture is about **average treatment effects**:

- The **potential outcomes** model for causal inference in randomized experiments.
- Observational studies and the **propensity score**.
- **Double robustness**, or how to use machine learning for principled treatment effect estimation.

The potential outcomes framework

For a set of i.i.d. subjects $i = 1, ..., n$, we observe a tuple (X_i, Y_i, W_i), comprised of

- A **feature vector** $X_i \in \mathbb{R}^p$,
- A **response** $Y_i \in \mathbb{R}$, and
- A **treatment assignment** $W_i \in \{0, 1\}$.

Following the **potential outcomes** framework (Neyman, 1923; Rubin, 1974), we posit the existence of quantities $Y_i(0)$ and $Y_i(1)$, such that $Y_i = Y_i(W_i)$.

- Potential outcomes correspond to the response we **would have measured** given that the i-th subject received treatment ($W_i = 1$) or no treatment ($W_i = 0$).
- The **causal effect** of the treatment is $Y_i(1) - Y_i(0)$.

The potential outcomes framework

For a set of i.i.d. subjects $i = 1, \ldots, n$, we observe a tuple (X_i, Y_i, W_i), comprised of

- A **feature vector** $X_i \in \mathbb{R}^p$,
- A **response** $Y_i \in \mathbb{R}$, and
- A **treatment assignment** $W_i \in \{0, 1\}$.

Our first goal is to estimate the **average treatment effect (ATE)**

$$\tau = \mathbb{E}\left[Y_i(1) - Y_i(0)\right].$$

Of course, we only get to see $Y_i = Y_i(W_i)$. This **"missing data"** issue is a fundamental problem ins causal inference.

The potential outcomes framework

The simplest way to **identify** the ATE in the potential outcomes is via a **randomized trial**:

$$\{Y_i(0),\ Y_i(1)\} \perp\!\!\!\perp W_i.$$

In a randomized trial, we can check that:

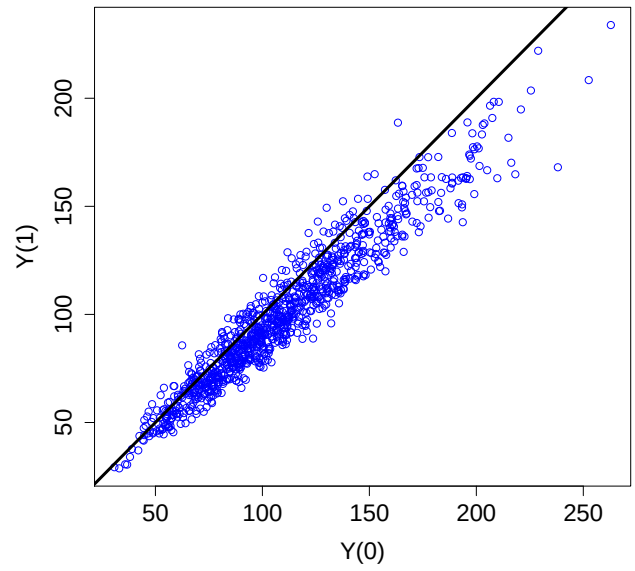
$$\begin{aligned}
\tau &= \mathbb{E}\left[Y_i(1)\right] - \mathbb{E}\left[Y_i(0)\right] \\
&= \mathbb{E}\left[Y_i(1) \,\middle|\, W_i = 1\right] - \mathbb{E}\left[Y_i(0) \,\middle|\, W_i = 0\right] \\
&= \mathbb{E}\left[Y_i \,\middle|\, W_i = 1\right] - \mathbb{E}\left[Y_i \,\middle|\, W_i = 0\right],
\end{aligned}$$

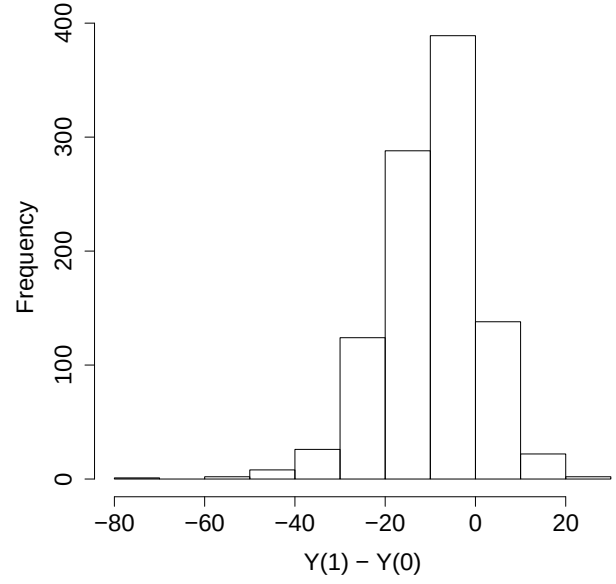
where the last line only has **observable moments**.

Thus, although we never observe $\tau_i = Y_i(1) - Y_i(0)$, we can **consistently estimate** $\tau = \mathbb{E}\left[\tau_i\right]$ in a randomized trial.

Example: The outcome Y_i is daily **air quality index**. The treatment imposes restrictions on driving to reduce traffic.

$Y_i(0)$	$Y_i(1)$	τ_i
154.68	153.49	-1.20
135.67	120.40	-15.27
103.46	117.68	14.23
117.62	95.08	-22.54
161.11	146.73	-14.39
117.89	105.05	-12.84
84.00	75.59	-8.41
73.32	65.68	-7.63
100.07	93.80	-6.28
103.81	82.30	-21.51
...	...	...
111.68	101.47	-10.21





Example: The outcome Y_i is daily **air quality index**. The treatment imposes restrictions on driving to reduce traffic.

$Y_i(0)$	$Y_i(1)$	τ_i
154.68	—	—
135.67	—	—
—	117.68	—
—	95.08	—
—	146.73	—
117.89	—	—
—	75.59	—
—	65.68	—
100.07	—	—
—	82.30	—
...	...	...
110.59	100.52	—

- In practice, we only ever observe a **single** potential outcome.
- However, in a RCT, we can use **averages** over the treated and controls to estimate the ATE.
- We **estimate** $\hat{\tau}$ as $110.59 - 100.52 = 10.07$.

ATE estimation in randomized trials

We have use the **potential outcomes** framework to justify the classical estimator of an **average treatment effect**:

$$\hat{\tau} = \frac{\sum_{\{i:W_i=1\}} Y_i}{|\{i : W_i = 1\}|} - \frac{\sum_{\{i:W_i=0\}} Y_i}{|\{i : W_i = 0\}|}.$$

This estimator is **unbiased**, **consistent**, **asymptotically Gaussian**, and also very **simple**. But is it the best we can do?

- If one has access to **covariates** X_i and can estimate $\mathbb{E}\left[Y_i \,\middle|\, X_i, W_i\right]$ accurately, then one can **improve the precision** of the above estimator.
- Any black-box predictor can be used for this (e.g., a forest, boosted trees, a deep net); the improvement in precision depends on **mean-squared error**.

ATE estimation in randomized trials

The simplest ATE estimator in an RCT is

$$\hat{\tau} = \frac{\sum_{\{i:W_i=1\}} Y_i}{|\{i : W_i = 1\}|} - \frac{\sum_{\{i:W_i=0\}} Y_i}{|\{i : W_i = 0\}|}.$$

How could we possibly improve on this?

- In the **air quality** example, weather has an effect on ozone (hot days have higher levels), independently of treatment.
- If we randomly assign treatment to more hot days and control to more cold days, our estimates we **exaggerate the treatment effect**, and vice-versa.
- In **large samples** these effects cancel out, but in **small samples** they matter. If we could **predict** and **eliminate** the effect of weather, we'd improve accuracy.

The traditional approach to this is via **stratified sampling**; here, we'll discuss an automatic approach that only assumes the existence of a **good predictor**.

ATE estimation in randomized trials

For a set of i.i.d. subjects $i = 1, \ldots, n$, we observe a tuple (X_i, Y_i, W_i), comprised of

- A **feature vector** $X_i \in \mathbb{R}^p$,
- A **response** $Y_i \in \mathbb{R}$, and
- A **treatment assignment** $W_i \in \{0, 1\}$.

Define the conditional **response surfaces** as

$$\mu_{(w)}(x) = \mathbb{E}\left[Y_i \,\middle|\, X_i = x,\, W_i = w\right].$$

In the potential outcomes model, an **oracle** who knew the $\mu_{(w)}(x)$ could use

$$\bar{\tau} = \frac{1}{n}\sum_{i=1}^{n}\left(\mu_{(1)}(X_i) - \mu_{(0)}(X_i)\right).$$

Our approach starts by seeking to imitate this oracle.

ATE estimation via prediction

In the potential outcomes model, an **oracle** who knew the $\mu_{(w)}(x)$ could use

$$\bar{\tau} = \frac{1}{n} \sum_{i=1}^{n} \left(\mu_{(1)}(X_i) - \mu_{(0)}(X_i) \right).$$

A first, naive approach simply sets

$$\hat{\tau} = \frac{1}{n} \sum_{i=1}^{n} \left(\hat{\mu}_{(1)}(X_i) - \hat{\mu}_{(0)}(X_i) \right).$$

This is good if $\hat{\mu}_{(w)}(x)$ is obtained via **OLS**. But it breaks down if we use **regularization**.

Example. Suppose that $p \gg n$, but the true model is sparse,

$$\mathbb{E}\left[Y \,\middle|\, X = x, W = w \right] = 2X_1 + 0.1WX_2.$$

A **lasso** might set the coefficient on WX_2 to 0, and estimate $\hat{\tau} = 0$!

ATE estimation via prediction

A better estimator needs to **correct for regularization bias**:

$$\hat{\tau} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\mu}_{(1)}(X_i) - \hat{\mu}_{(0)}(X_i)\right) \qquad \text{(optimistic plug-in)}$$

$$+ \frac{\sum_{\{i:W_i=1\}}\left(Y_i - \hat{\mu}_{(1)}(X_i)\right)}{|\{i : W_i = 1\}|} \qquad \text{(bias correction for } \hat{\mu}_{(1)}(\cdot)\text{)}$$

$$- \frac{\sum_{\{i:W_i=0\}}\left(Y_i - \hat{\mu}_{(0)}(X_i)\right)}{|\{i : W_i = 0\}|} \qquad \text{(bias correction for } \hat{\mu}_{(0)}(\cdot)\text{)}$$

Modulo technical details, this is justified **for any** $\hat{\mu}_{(w)}(x)$. If $\hat{\mu}_{(w)}(x)$ can predict Y_i at all, can improve over basic estimator.

If $\hat{\mu}_{(w)}(x)$ is consistent, i.e., $\mathbb{E}\left[(\hat{\mu}_{(W)}(X) - \mu_{(W)}(X))^2\right] \to 0$, then this estimator is **optimal in large samples**.

Details: Wager et al. **High-Dim. Regression Adjust. in RCTs.** *PNAS*, 113(45), 2016.

ATE estimation via prediction

Example: We have $n = 1000$, $p = 400$, and $\mathbb{P}[W = 1] = 0.4$, with

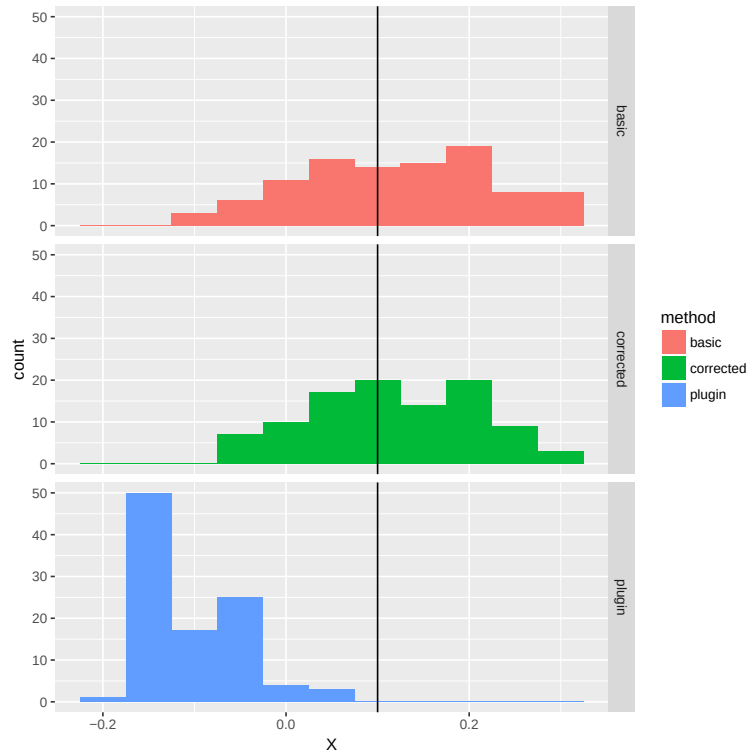
$$\mathbb{E}\left[Y \,\middle|\, X = x, W = w\right] = 2X_1 + 0.1WX_2, \quad X_{ij} \overset{\text{iid}}{\sim} U([0, 2]).$$

Predictions made via a **cross-validated lasso** (no intercept).

Consider 3 estimators, with **mean-square errors** for τ:

- **basic**: 0.105.
- **bias-corrected**: 0.092.
- **plug-in**: 0.210.

Distribution of estimates:



Today's lecture is about **average treatment effects**:

- The **potential outcomes** model for causal inference in randomized experiments.
- Observational studies and the **propensity score**.
- **Double robustness**, or how to use machine learning for principled treatment effect estimation.

Beyond randomized trials

The simplest way to move beyond randomized controlled trials is to let randomization probabilities depend on **covariate information**.

- We are interested in giving teenagers **cash incentives** to discourage them from **smoking**.
- A random subset of $\sim 5\%$ of teenagers in **Palo Alto, CA**, and a random subset of $\sim 20\%$ of teenagers in **Geneva, Switzerland** are eligible for the study.

Palo Alto	Non-S.	Smoker
Treat.	152	5
Control	2362	122

Geneva	Non-S.	Smoker
Treat.	581	350
Control	2278	1979

This is **not a randomized controlled study**, because Genevans are both more likely to smoke whether or not they get treated, and more likely to get treated.

Beyond randomized trials

The Palo Alto experiment and Geneva experiment are both individually randomized controlled studies—and looking at the numbers clearly shows that the treatment helps prevent smoking.

Palo Alto	Non-S.	Smoker
Treat.	152	6
Control	2362	122

Geneva	Non-S.	Smoker
Treat.	581	395
Control	2278	1979

Looking at aggregate data is misleading, and makes it look like the treatment hurts.

Palo Alto + Geneva	Non-Smoker	Smoker
Treatment	733	401
Control	4640	2101

This phenomenon is an example of Simpson's "paradox".

Beyond randomized trials

Formally, we have covariates $X_i \in \{\text{Palo Alto, Geneva}\}$, and know that the treatment assignment was random conditionally on X_i:

$$\{Y_i(0),\ Y_i(1)\} \perp\!\!\!\perp W_i \mid X_i.$$

We then estimate the overall average treatment effect as:

$$\hat{\tau} = \sum_{x \in \mathcal{X}} \frac{|\{X_i = x\}|}{n} \hat{\tau}(x),$$

$$\hat{\tau}(x) = \frac{\sum_{\{i: X_i = x,\ W_i = 1\}} Y_i}{|\{i : X_i = x,\ W_i = 1\}|} - \frac{\sum_{\{i: X_i = x,\ W_i = 0\}} Y_i}{|\{i : X_i = x,\ W_i = 0\}|}.$$

Covariates and unconfoundedness

For a set of i.i.d. subjects $i = 1, \ldots, n$, we observe a tuple (X_i, Y_i, W_i), comprised of

- A **feature vector** $X_i \in \mathbb{R}^p$,
- A **response** $Y_i \in \mathbb{R}$, and
- A **treatment assignment** $W_i \in \{0, 1\}$.

We assume that the treatment is **unconfounded** (aka selection on observables) (Rosenbaum & Rubin, 1983):

$$\{Y_i(0), Y_i(1)\} \perp\!\!\!\perp W_i \mid X_i.$$

We seek the ATE $\tau = \mathbb{E}[Y_i(1) - Y_i(0)]$. If the X_i is discrete, we can **stratify**: estimate an ATE for each x separately, and aggregate. But what if X is continuous and/or high-dimensional?

Covariates and unconfoundedness

Given **unconfoundedness** $\{Y_i(0),\ Y_i(1)\} \perp\!\!\!\perp W_i \mid X_i$, we have

$$\begin{aligned}
\tau &= \mathbb{E}\left[Y_i(1) - Y_i(0)\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[Y_i(1) \,\middle|\, X_i\right] - \mathbb{E}\left[Y_i(0) \,\middle|\, X_i\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[Y_i \,\middle|\, X_i,\ W_i = 1\right] - \mathbb{E}\left[Y_i \,\middle|\, X_i,\ W_i = 0\right]\right] \\
&= \mathbb{E}\left[\mu_{(1)}(X_i) - \mu_{(0)}(X_i)\right],
\end{aligned}$$

where $\mu_{(w)}(x) = \mathbb{E}\left[Y_i \,\middle|\, X_i = x,\ W_i = w\right]$. This suggests an **estimator** based on a **regression adjustment**:

$$\hat{\tau} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\mu}_{(1)}(X_i) - \hat{\mu}_{(0)}(X_i)\right),$$

where $\hat{\mu}_{(w)}(X_i)$ is obtained by regression Y_i on X_i on those observations with $W_i = w$.

Is this going to work? **Yes** with OLS, **no** in general.

Review: ATE estimation via OLS

A classical approach to the ATE involves estimating $\mu_{(0)}(x)$ and $\mu_{(1)}(x)$ via **ordinary least-squares regression** (OLS). Specifically, in `R` notation, we first run two separate regressions (recall that `lm` is the `R` command for running linear regression):

$$\hat{\beta}_{(0)} \leftarrow \texttt{lm}\left(Y_i \sim X_i,\ \texttt{subset}\ \ W_i = 0\right),$$
$$\hat{\beta}_{(1)} \leftarrow \texttt{lm}\left(Y_i \sim X_i,\ \texttt{subset}\ \ W_i = 1\right).$$

We then make predictions $\hat{\mu}_{(w)}(x) = \hat{\beta}_{(w)}x$, and obtain a **treatment effect** estimate as

$$\hat{\tau} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\mu}_{(1)}(X_i) - \hat{\mu}_{(0)}(X_i)\right) = \left(\hat{\beta}_{(1)} - \hat{\beta}_{(0)}\right)\overline{X},$$

where $\overline{X} = \sum_{i=1}^{n} X_i$. Note that, X implicitly includes an **intercept**.

Review: ATE estimation via OLS

```
library(sandwich) # for robust standard errors
treat_data = read.table("...../nswre74_treated.txt")
control_data = read.table("...../psid3_controls.txt")
combined = rbind(treat_data, control_data)
X = combined[,2:9]; Y = combined[,10]; W = combined[,1]

# First center the X, then run OLS with full W:X
# interactions. With this construction, the
# W-coefficient can be interpreted as ATE.
X.centered = scale(X, center = TRUE, scale = FALSE)
ols.fit = lm(Y ~ W * X.centered)

# Use robust standard errors
tau.hat = coef(ols.fit)["W"]
tau.se = sqrt(sandwich::vcovHC(ols.fit)["W", "W"])
print(paste0("95% CI: ", round(tau.hat),
             " +/- ", round(1.96 * tau.se)))
"95% CI: 2107 +/- 2379"
```

ATE estimation via the lasso?

OLS is optimal for learning linear models in **low dimensions**, i.e., with p predictors using n samples when $p \ll n$. In many modern applications, however, p may be of comparable size (or larger than) the sample size n. In this case, we often use the **lasso**:

$$\hat{\beta}_{lasso} = \text{argmin} \left\{ \frac{1}{n} \sum_{i=1}^{n} (Y_i - X_i \beta)^2 + \lambda \|\beta\|_1 \right\}.$$

By analogy to the low-dimensional case, may be tempted to use

$$\begin{aligned}
\hat{\beta}_{(0)} &\leftarrow \texttt{lasso}\left(Y_i \sim X_i,\ \texttt{subset}\ \ W_i = 0\right), \\
\hat{\beta}_{(1)} &\leftarrow \texttt{lasso}\left(Y_i \sim X_i,\ \texttt{subset}\ \ W_i = 1\right), \\
\hat{\tau} &= \left(\hat{\beta}_{(1)} - \hat{\beta}_{(0)}\right) \overline{X}.
\end{aligned}$$

Is this any good? The fundamental difference between the lasso and OLS is that the lasso has **bias** (and, in fact, any method in high dimensions must have bias).

Imposing Sparsity: LASSO Crash Course

We are in a p-dimensional linear model with n samples. Assume that there are at most a fixed number k of **non-zero coefficients**:

$$Y_i = X_i\beta + \varepsilon_i, \quad \mathbb{E}\left[\varepsilon_i \,\middle|\, X_i\right] = 0,$$

such that $\|\beta\|_0 \leq k$.

Lasso theory provides results on when we can **consistently estimate** β, even when the number of features p may be much larger than n.

The strength of the results depends on the amount of **sparsity**: The smaller k, then better guarantees we can get.

Imposing Sparsity: LASSO Crash Course

Suppose $X \in \mathbb{R}^{n \times p}$ satisfies a restricted eigenvalue condition: no small group of variables is nearly collinear. Then we can show:

$$\left\|\hat{\beta} - \beta\right\|_2 = \mathcal{O}_P\left(\sqrt{\frac{k \log(p)}{n}}\right), \quad \left\|\hat{\beta} - \beta\right\|_1 = \mathcal{O}_P\left(k\sqrt{\frac{\log(p)}{n}}\right).$$

At a high level, this error arises because the lasso **shrinks** each coefficient on the order of $\sqrt{\log(p)/n}$.

If $k \ll n/\log(p)$, we say the problem is **sparse**, and the lasso will make accurate **predictions**. For example, if

$$X_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1), \text{ then } \mathbb{E}\left[(X\hat{\beta} - X\beta))^2\right] = \left\|\hat{\beta} - \beta\right\|_2^2.$$

If $k \ll \sqrt{n}/\log(p)$, we say the problem is **ultra-sparse**, and we can build confidence intervals for β_j via the **debiased lasso**.

Why Lasso Regression Adjustments Don't Work

We have a design with **no treatment effect**, but with a different **covariate distribution** for the treated and controls. Specifically:

$$\mathbb{E}\left[X \,\middle|\, W = 0\right] = \mathbf{0}, \quad \mathbb{E}\left[X \,\middle|\, W = 1\right] = \mathbf{1},$$

such that $Y(w) = X\beta_{(w)} + \text{noise}$, $\text{Var}\left[Y(w) \,\middle|\, X\right] = \sigma^2$ and

$$\left(\beta_{(0)}\right)_j = \left(\beta_{(1)}\right)_j = 1\left(\{i \leq k\}\right)\sigma\sqrt{\log(p)/n}, \text{ and so}$$
$$\tau = \mathbb{E}\left[X\right] \cdot \left(\beta_{(1)} - \beta_{(0)}\right) = 0.$$

The lasso fits, $\hat{\mu}_{(w)}(x) = \hat{a}_{(w)} + x\hat{\beta}_{(w)}$, with **intercept** $\hat{a}_{(w)}$.

Why Lasso Regression Adjustments Don't Work

In order to **zero-out noise terms**, the lasso must eliminate all signals smaller than $\sigma\sqrt{2\log(p)/n}$. Here, all β are smaller than this cutoff, so the lasso pushes them to 0.

Thus, with high probability,

$$\hat{a}_{(0)}^{lasso} \approx 0, \quad \hat{\beta}_{(0)}^{lasso} = 0, \quad \text{and}$$
$$\hat{a}_{(1)}^{lasso} \approx k\sqrt{\log(p)/n}, \quad \hat{\beta}_{(1)}^{lasso} = 0.$$

Combining these into an ATE estimate, we get

$$\hat{\tau}^{lasso} = \hat{a}_{(1)}^{lasso} - \hat{a}_{(0)}^{lasso} + \overline{X} \cdot \left(\hat{\beta}_{(1)}^{lasso} - \hat{\beta}_{(0)}^{lasso}\right) \approx k\sigma\sqrt{\log(p)/n}.$$

Thus, the lasso has a **bias** on the order of $k\sigma\sqrt{\log(p)/n}$, despite the fact that the true **treatment effect is 0**.

Why Lasso Regression Adjustments Don't Work

- The lasso only looks for strong relationships between X and the outcome Y.
- But, for estimating ATE, it's also important to capture variables with a strong relationship between X and W.
- Strong variables in the propensity model can leak confounding effects, even if the corresponding X-Y effect is so small the lasso ignores it.
- This was not a problem with OLS, because OLS is unbiased (so it tries to fit every coefficient accurately, even if it's close to 0).

Improving the Properties of ATE Estimation in High Dimensions: A "Double-Selection" Method

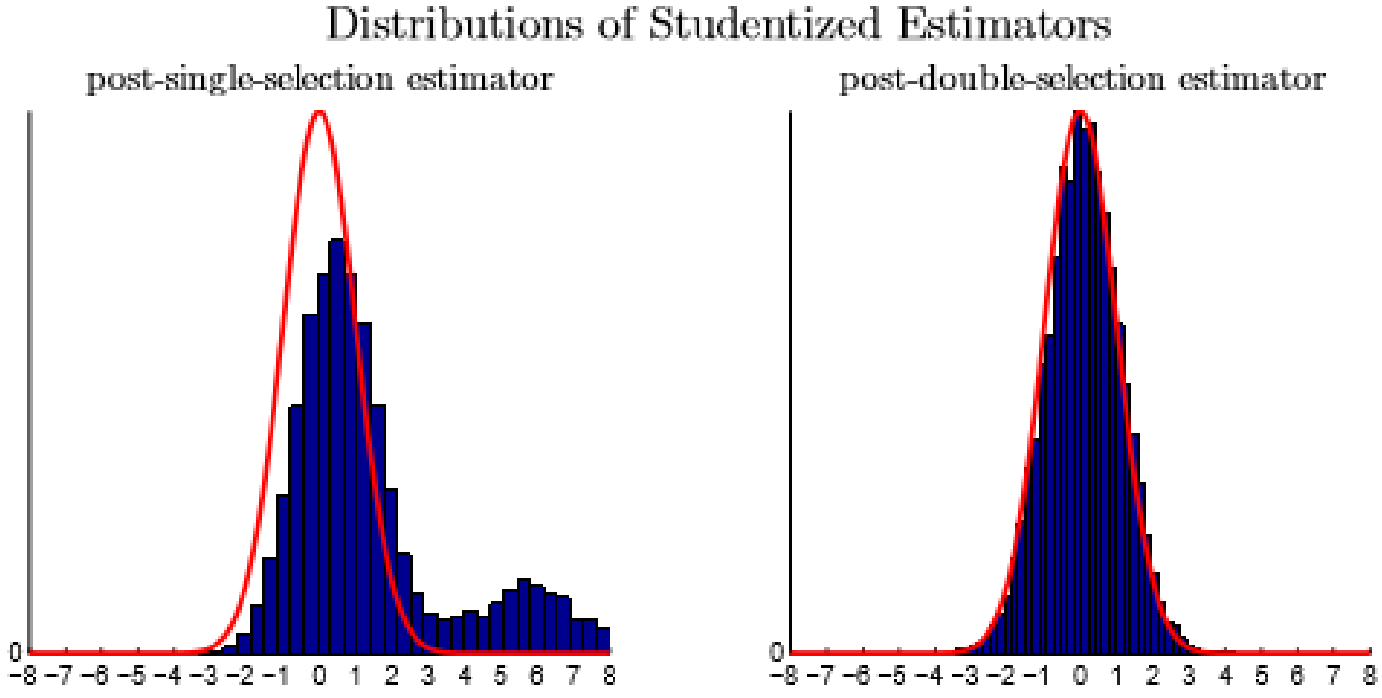
Belloni, Chernozukov, and Hansen (2014) propose a simple fix to this problem.

- Run a LASSO of W on X. Select variables with non-zero coefficients at a selected λ (e.g. cross-validation).
- Run a LASSO of Y on X on both the treated on control samples. Select variables with non-zero coefficients at a selected λ (may be different than first λ).
- Run OLS of Y on W interacted with the union of selected variables. Conclude as in the regular OLS case.

The third step above is not as good at purely **predicting** Y as using only second set. But it is more accurate for the ATE.

Result: under "approximate sparsity" of BOTH the propensity and outcome models, and constant treatment effects, estimated ATE is asymptotically normal and estimation is efficient.

Single v. Double Selection in BCH Algorithm



Recap: ATE estimation via the lasso

The **lasso** can be use to estimate conditional response functions $\mu_{(w)}(x) = \mathbb{E}\left[Y_i(w) \,\middle|\, X_i = x\right]$ as

$$\hat{\mu}^{lasso}_{(w)}(x) = \hat{a}^{lasso}_{(w)} + x\hat{\beta}^{lasso}_{(w)}.$$

Because we're in high dimensions, we need to **regularize**, and this leads to **bias**.

- The lasso is calibrated to make good **predictions**, but not necessarily to make good **ATE estimates**.
- The simple lasso regression adjustment $\hat{\tau} = \frac{1}{n}\sum_{i=1}^{n}(\hat{\mu}^{lasso}_{(1)}(X_i) - \hat{\mu}^{lasso}_{(0)}(X_i))$ may fail badly.

The BCH method provides a fix to this problem, by "un-regularizing" features that are important in the propensity model.

- Does this idea generalize beyond **linear models**?
- What if the propensity model isn't **sparse**?

The propensity score

The confounding effects of X_i can alternatively be captured via the **propensity score**,

$$e(x) = \mathbb{P}\left[W_i = 1 \,\middle|\, X_i = x\right].$$

The key fact about the propensity score is that

$$\tau = \mathbb{E}\left[\frac{W_i Y_i}{e(X_i)} - \frac{(1 - W_i) Y_i}{1 - e(X_i)}\right].$$

The same idea underlies **importance weighting**, **Horvitz-Thompson sampling**, etc.

The propensity score

Inverse-propensity weighting is unbiased because:

$$
\begin{aligned}
\tau &= \mathbb{E}\left[Y_i(1) - Y_i(0)\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[Y_i(1) \,\middle|\, X_i\right] - \mathbb{E}\left[Y_i(0) \,\middle|\, X_i\right]\right] \\
&= \mathbb{E}\left[\frac{\mathbb{E}\left[W_i \,\middle|\, X_i\right]\mathbb{E}\left[Y_i(1) \,\middle|\, X_i\right]}{e(X_i)} - \frac{\mathbb{E}\left[1 - W_i \,\middle|\, X_i\right]\mathbb{E}\left[Y_i(0) \,\middle|\, X_i\right]}{1 - e(X_i)}\right] \\
&= \mathbb{E}\left[\frac{\mathbb{E}\left[W_i Y_i(1) \,\middle|\, X_i\right]}{e(X_i)} - \frac{\mathbb{E}\left[(1 - W_i)\, Y_i(0) \,\middle|\, X_i\right]}{1 - e(X_i)}\right] \\
&= \mathbb{E}\left[\frac{W_i Y_i}{e(X_i)} - \frac{(1 - W_i) Y_i}{1 - e(X_i)}\right].
\end{aligned}
$$

The 5-th equality depends on consistency of the **potential outcomes**, and the 4-th equality relies on **unconfoundedness**,

$$\{Y_i(0),\ Y_i(1)\} \perp\!\!\!\perp W_i \,\middle|\, X_i.$$

Inverse-propensity weighting

We know that the **average treatment effect** is

$$\tau = \mathbb{E}\left[\frac{W_i Y_i}{e(X_i)} - \frac{(1 - W_i) Y_i}{1 - e(X_i)}\right].$$

A natural idea is to **estimate** $\hat{e}(\cdot)$ via some machine learning method (e.g., an L_1-penalized logistic regression in high dimensions), and then use

$$\hat{\tau} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{W_i Y_i}{\hat{e}(X_i)} - \frac{(1 - W_i) Y_i}{1 - \hat{e}(X_i)}\right).$$

This strategy has several pitfalls, however:

- Getting properly **calibrated** $\hat{e}(\cdot)$ estimates is hard.
- **Regularization bias** is still a problem.

The rest of this lecture is about a strategy to **overcome** this issue.

Augmented Inverse-Propensity Weighting

There is a more flexible approach to using **machine learning** methods for ATE estimation that relies on both **outcome regression** and the **propensity score**

$$\mu_{(w)}(x) = \mathbb{E}\left[Y_i \,\middle|\, X_i = x,\, W_i = w\right], \quad e(x) = \mathbb{P}\left[W_i = 1 \,\middle|\, X_i = x\right].$$

Suppose that we have estimates $\hat{\mu}_{(w)}(x)$ from any machine learning method, and also have propensity estimate $\hat{e}(x)$. AIPW then uses:

$$\hat{\tau} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\mu}_{(1)}(X_i) - \hat{\mu}_{(0)}(X_i) + \frac{W_i}{\hat{e}(X_i)}\left(Y_i - \hat{\mu}_{(1)}(X_i)\right) - \frac{1 - W_i}{1 - \hat{e}(X_i)}\left(Y_i - \hat{\mu}_{(0)}(X_i)\right)\right).$$

In considerable generality, this is a good estimator of the ATE.

Augmented Inverse-Propensity Weighting

To interpret AIPW, it is helpful to write it as

$$\hat{\tau}_{AIPW} = D + R$$

$$D = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\mu}_{(1)}(X_i) - \hat{\mu}_{(0)}(X_i)\right)$$

$$R = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{W_i}{\hat{e}(X_i)}\left(Y_i - \hat{\mu}_{(1)}(X_i)\right) - \frac{1-W_i}{1-\hat{e}(X_i)}\left(Y_i - \hat{\mu}_{(0)}(X_i)\right)\right).$$

D is the direct **regression adjustment** estimator using $\hat{\mu}_{(w)}(x)$, and R is an IPW estimator applied to the **residuals** $Y_i - \hat{\mu}_{(W_i)}(X_i)$.

Qualitatively, AIPW uses propensity weighting on the residuals to **debias** the direct estimate.

A Simple Example

Consider an example with $X_i \sim \mathcal{N}(0, I)$, $n = 1,000$ and $p = 20$:

$$e(x) = 1/(1 + e^{-x_1}), \quad \mu_{(0)}(x) = (x_1 + x_2)_+, \quad \mu_{(1)}(x) = (x_1 + x_3)_+.$$

Here, we need to model $\mu_{(w)}(x)$ **non-parametrically**, but there's not quite enough data to nail the functional form quite right.

$$\hat{\tau}_{AIPW} = D + R$$

$$D = \frac{1}{n} \sum_{i=1}^{n} \left(\hat{\mu}_{(1)}(X_i) - \hat{\mu}_{(0)}(X_i) \right)$$

$$R = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{W_i}{\hat{e}(X_i)} \left(Y_i - \hat{\mu}_{(1)}(X_i) \right) - \frac{1 - W_i}{1 - \hat{e}(X_i)} \left(Y_i - \hat{\mu}_{(0)}(X_i) \right) \right).$$

Here, the **direct** regression adjustment D is on average **0.18**, the **correction** R is on average **-0.12**, and **AIPW** gives us on average **0.06**, which is closer to the **correct answer** $\tau = 0$.

Understanding Augmented Inverse-Propensity Weighting

To understand why AIPW works, we can compare it to an **oracle** that gets to use the true values of $\mu_{(w)}(x)$ and $e(x)$:

$$\tilde{\tau}_{AIPW} = \frac{1}{n}\sum_{i=1}^{n}\left(\mu_{(1)}(X_i) - \mu_{(0)}(X_i) + \frac{W_i}{e(X_i)}\left(Y_i - \mu_{(1)}(X_i)\right) - \frac{1-W_i}{1-e(X_i)}\left(Y_i - \mu_{(0)}(X_i)\right)\right).$$

"Theorem." If the first-stage function estimates satisfy

$$\mathbb{E}\left[\left(\hat{\mu}_{(w)}(X) - \mu_{(w)}(X)\right)^2\right]^{\frac{1}{2}}\mathbb{E}\left[\left(\hat{e}(X) - e(X)\right)^2\right]^{\frac{1}{2}} = o_P\left(\frac{1}{\sqrt{n}}\right),$$

and we also have **overlap**, then $\hat{\tau}_{AIPW}$ and $\tilde{\tau}_{AIPW}$ satisfy

$$\sqrt{n}\left(\hat{\tau}_{AIPW} - \tilde{\tau}_{AIPW}\right) \to_p 0.$$

In other words, $\hat{\tau}_{AIPW}$ and $\tilde{\tau}_{AIPW}$ are first-order equivalent.

Understanding Augmented Inverse-Propensity Weighting

The upshot of this result is that we can study $\tilde{\tau}_{AIPW}$ instead of $\hat{\tau}_{AIPW}$. Because $\tilde{\tau}_{AIPW}$ is just an average of independent terms, a direct application of the **central limit theorem** implies that

$$\sqrt{n}\left(\tilde{\tau}_{AIPW} - \tau\right) \Rightarrow \mathcal{N}\left(0,\, V^*\right),$$

$$V^* = \text{Var}\left[\mu_{(1)}(X) - \mu_{(0)}(X)\right] + \mathbb{E}\left[\frac{\text{Var}\left[Y_i(1)\right] \mid X_i}{e(X_i)}\right] + \mathbb{E}\left[\frac{\text{Var}\left[Y_i(0)\right] \mid X_i}{1 - e(X_i)}\right].$$

Because $\hat{\tau}_{AIPW}$ and $\tilde{\tau}_{AIPW}$ are equivalent on the $\sqrt{n}$-scale, we then immediately get, whenever the result from the previous slide holds,

$$\sqrt{n}\left(\hat{\tau}_{AIPW} - \tau\right) \Rightarrow \mathcal{N}\left(0,\, V^*\right).$$

Moreover, it can be shown that this behavior is **optimal** for any ATE estimator, assuming a generic non-parametric setup.

Inference with Augmented Inverse-Propensity Weighting

We are considering $\hat{\tau}_{AIPW} = n^{-1} \sum_{i=1}^{n} \widehat{\Gamma}_i$, with

$$\widehat{\Gamma}_i = \hat{\mu}_{(1)}(X_i) - \hat{\mu}_{(0)}(X_i) + \frac{W_i}{\hat{e}(X_i)} \left(Y_i - \hat{\mu}_{(1)}(X_i)\right) - \frac{1 - W_i}{1 - \hat{e}(X_i)} \left(Y_i - \hat{\mu}_{(0)}(X_i)\right)$$

If the first-stage estimates are reasonably accurate, this estimator is to first order **not affected by errors** in $\hat{e}(\cdot)$ and $\hat{\mu}_{(w)}(\cdot)$.

As a consequence, for **inference**, we can act as though $\hat{\tau}$ were the average of independent terms $\widehat{\Gamma}_i$, with variance

$$\widehat{\mathrm{Var}}\left[\hat{\tau}_{AIPW}\right] = \widehat{V}_n := \frac{1}{n(n-1)} \sum_{i=1}^{n} \left(\widehat{\Gamma}_i - \hat{\tau}_{AIPW}\right)^2.$$

We can use this to build Gaussian **confidence intervals**:

$$\mathbb{P}\left[\tau \in \left\{\hat{\tau}_{AIPW} \pm z_{1-\alpha/2} \widehat{V}_n^{1/2}\right\}\right] \to 1 - \alpha.$$

Details #1: The assumptions

Our result relies on the assumption that

$$\mathbb{E}\left[\left(\hat{\mu}_{(w)}(X) - \mu_{(w)}(X)\right)^2\right]^{\frac{1}{2}} \mathbb{E}\left[\left(\hat{e}(X) - e(X)\right)^2\right]^{\frac{1}{2}} = o_P\left(\frac{1}{\sqrt{n}}\right).$$

One simple way to achieve this condition is if both $\hat{\mu}$ and $\hat{e}$ are $o(1/n^{1/4})$-consistent in root-mean squared error.

- In other words, our final estimate $\hat{\tau}_{AIPW}$ can be an **order of magnitude** more accurate than either nuisance component ($1/n^{1/2}$ vs $1/n^{1/4}$).
- The reason for this phenomenon is that, to first order, the errors in the $\hat{\mu}$ and $\hat{e}$ regressions **cancel out**.
- This is known as the **orthogonal moments** construction, and plays a key role in semiparametric statistics.

Of course, $o(1/n^{1/4})$-consistency is still a strong assumption, and may not always hold. The topic of when we can get $o(1/n^{1/4})$-consistency, and also when we can improve on the assumptions stated above, is the topic of a large literature.

Details #2: Cross-fitting

To get good behavior out of AIPW, we recommend **cross-fitting**

$$\hat{\tau} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\mu}_{(1)}^{(-i)}(X_i) - \hat{\mu}_{(0)}^{(-i)}(X_i) + \frac{W_i}{\hat{e}^{(-i)}(X_i)}\left(Y_i - \hat{\mu}_{(1)}^{(-i)}(X_i)\right) - \frac{1-W_i}{1-\hat{e}^{(-i)}(X_i)}\left(Y_i - \hat{\mu}_{(0)}^{(-i)}(X_i)\right)\right).$$

In other words, when estimating $e(X_i)$, use a model that **did not have access** to the i-th training example during training.

- A simple approach is to cut the data into K **folds**. Then, for each $k = 1, ..., K$, train a model on all but the k-th fold, and evaluate its predictions on the k-th fold.
- With forests, **leave-one-out** estimation is natural, i.e., $\hat{e}^{(-i)}(X_i)$ is trained on all but the i-th sample.

Chernozhukov et al. (2017) emphasize the role of cross-fitting in proving flexible efficiency results for AIPW.

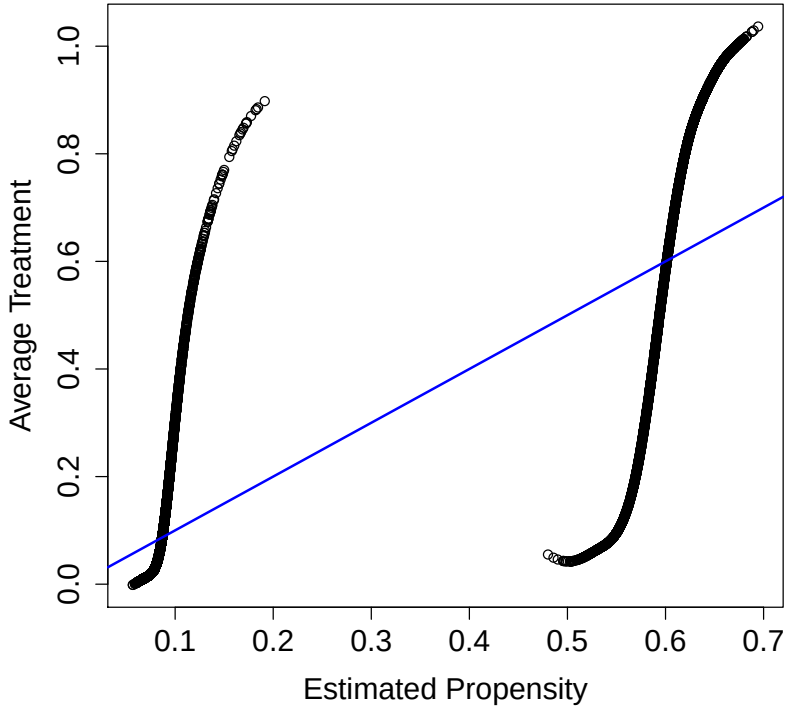
Details #2: Cross-fitting

Example from tutorial, with $n = 9750$ and $p = 21$.

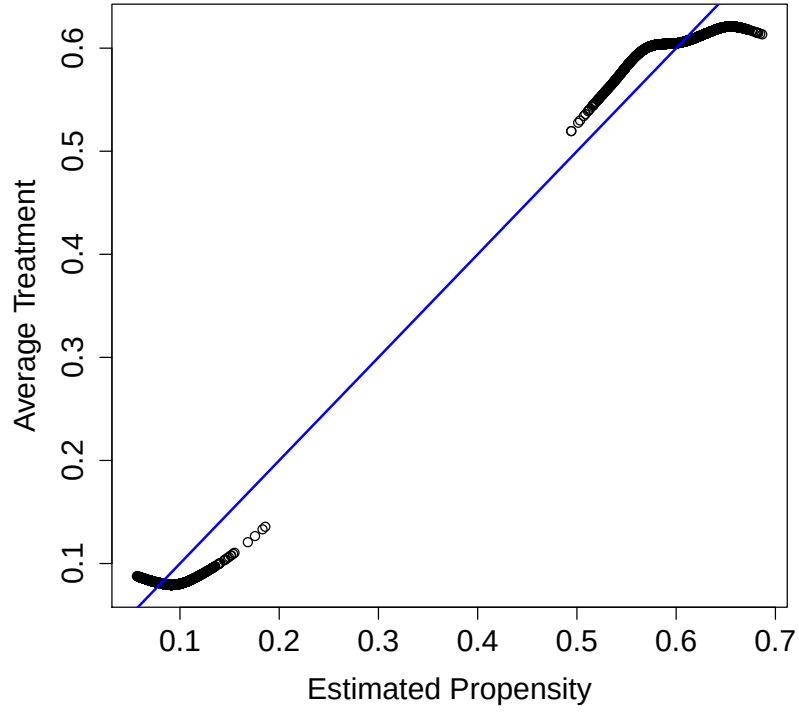
```
library(grf)
propensity_fit = regression_forest(Xmod, Wmod)

# If you ask a forest to predict without giving it a test
# set, it automatically does OOB on the training set.

ehat_oob = predict(propensity_fit)$predictions
ehat_naive = predict(propensity_fit,
                     newdata = Xmod)$predictions

c(OOB=mean(Wmod / ehat_oob), NAIVE=mean(Wmod / ehat_naive))

    OOB          NAIVE
    1.0173325    0.9093602
```

Details #2: Cross-fitting

Calibration plots run a single non-parametric regression of W_i against $\hat{e}(X_i)$, and are a good way to assess quality of a propensity fit. Ideally, the calibration curve should be close to the diagonal.



without crossfitting



with crossfitting

Details #3: Overlap

Overlap means that propensity scores are bounded away from 0 and 1:

$$\eta \leq \mathbb{P}\left[W_i = 1 \,\middle|\, X_i = x\right] \leq 1 - \eta, \quad \eta > 0,$$

for all possible value of x. The proof assumes overlap, and even the limiting **variance** gets bad as overlap gets bad:

$$V^* = \text{Var}\left[\mu_{(1)}(X) - \mu_{(0)}(X)\right] + \mathbb{E}\left[\frac{\text{Var}\left[Y_i(1)\right] \mid X_i}{e(X_i)}\right] + \mathbb{E}\left[\frac{\text{Var}\left[Y_i(0)\right] \mid X_i}{1 - e(X_i)}\right].$$

In applications, it is important to check overlap.

The role of overlap

Note that we need $e(x) \in (0, 1)$ to be able to calculate treatment effects for all x.

- Intuitively, how could you possibly infer $[Y(0)|X_i = x]$ if $e(x) = 1$?
- Note that for discrete x, the variance of ATE is infinite when $e(x) = 0$.
- "Moving the goalposts": Crump, Hotz, Imbens, Miller (2009) analyze trimming, which entails dropping observations where $e(x)$ is too extreme. Typical approaches entail dropping bottom and top 5% or 10%.
- Approaches that don't directly require propensity score weighting may seem to avoid the need for this, but important to understand role of extrapolation.
- If we subset the data, need to be mindful of what the estimand is.

Propensity Score Plots: Assessing Overlap

The causal inference literature has developed a variety of conventions, broadly referred to as "supplementary analysis," for assessing credibility of empirical studies. One of the most prevalent conventions is to plot the propensity scores of treated and control groups to assess overlap.

- Idea: for each $q \in (0, 1)$, plot the fraction of observations in the treatment group with $e(x) = q$, and likewise for the control group.
- Even if there is overlap, when there are large imbalances, this is a sign that it may be difficult to get an accurate estimate of the treatment effect.

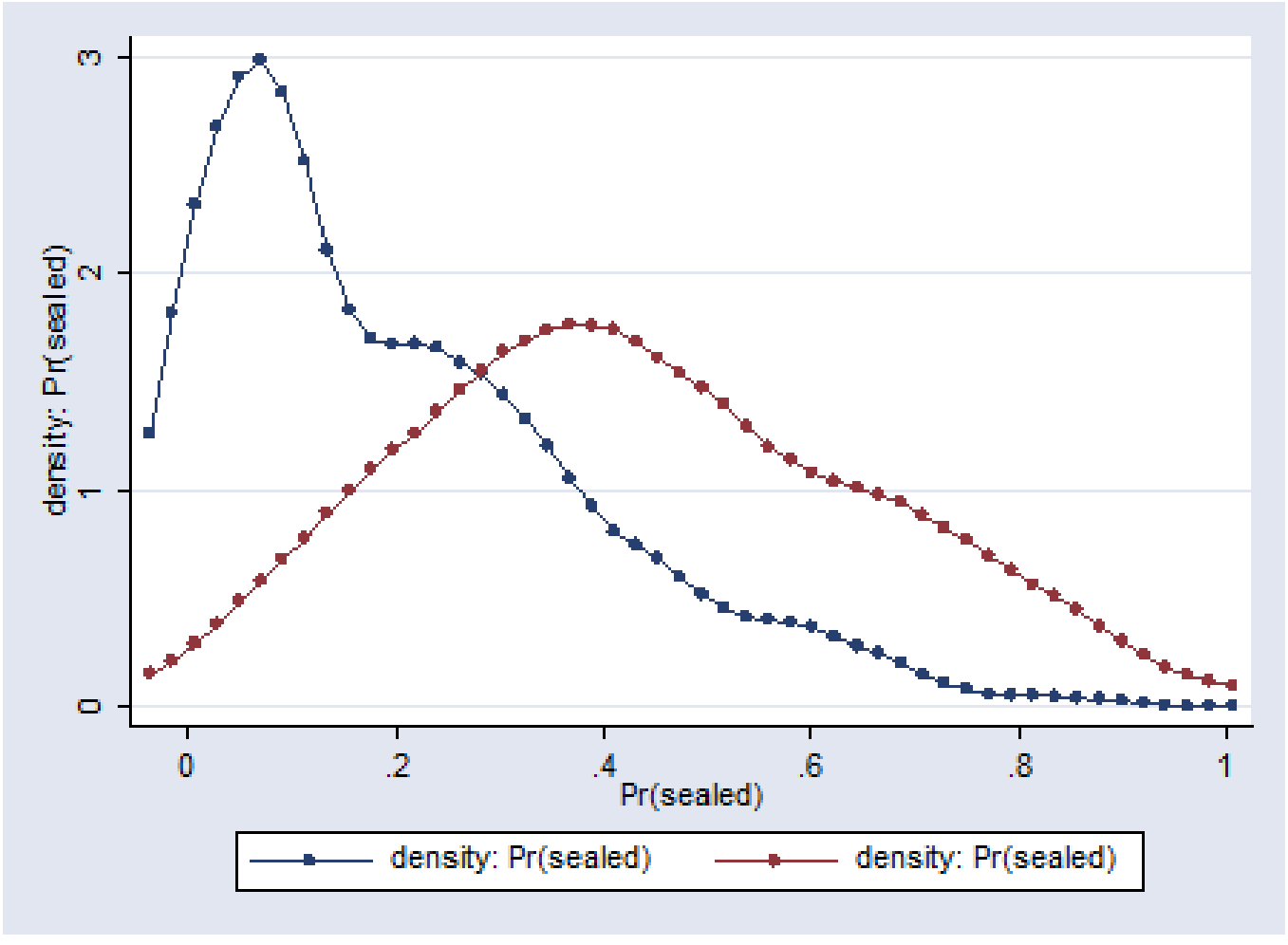
Propensity Score Plots: Assessing Overlap

Example: Athey, Levin and Seira analysis of timber auctions.

- The paper studies consequences of awarding contracts to harvest timber via first price sealed auction or open ascending auction.
- Assignment to first price sealed auction or open ascending auction:
  - In Idaho, auction mechanism is randomized for subset of tracts with different probabilities in different geographies;
  - In California, auction mechanism is determined by small v. large sales (with cutoffs varying by geography).
- So $W = 1$ if auction is sealed, and X represents geography, size and year.

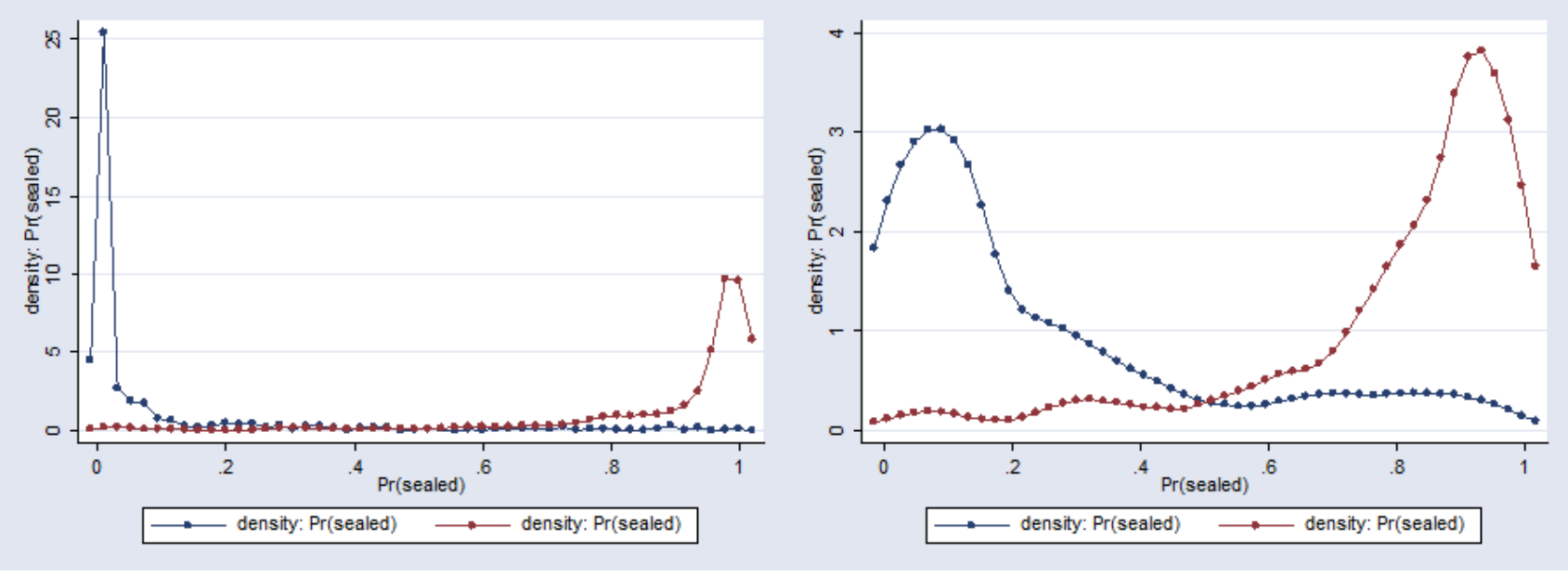
Propensity Score Plots: Assessing Overlap in ID

Very few observations with extreme propensity scores



Propensity Score Plots: Assessing Overlap in CA

Untrimmed v. trimmed so that $e(x) \in [.025, .975]$



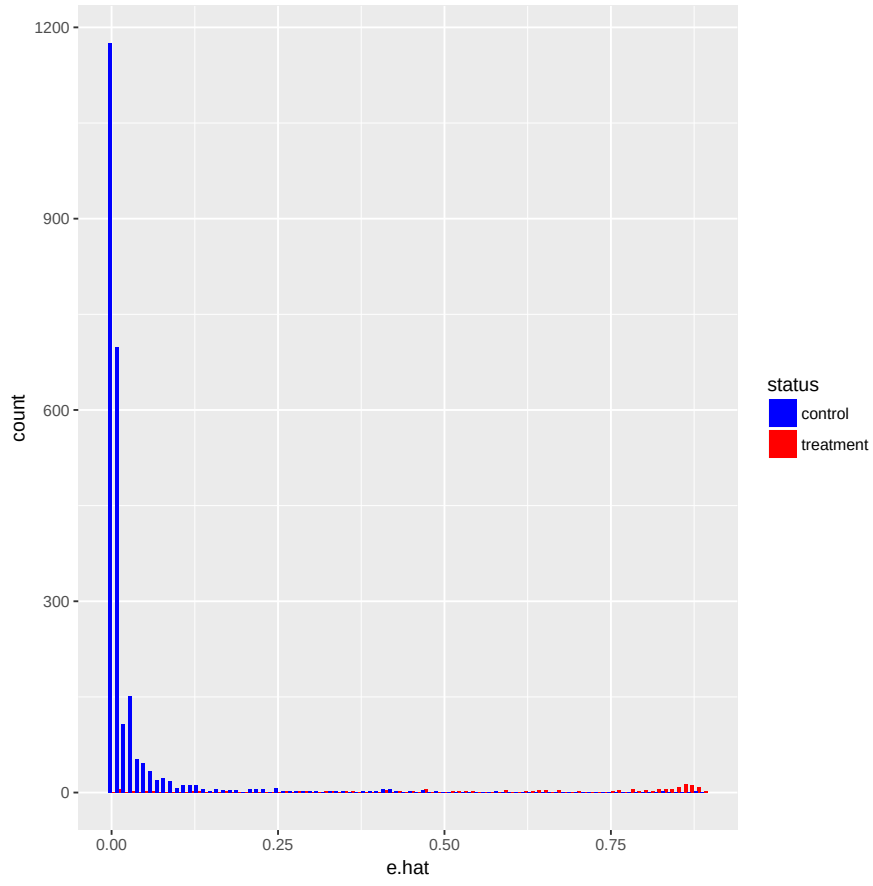
Estimation Strategies with Poor Overlap

Lalonde (1986) collected a classical dataset for **program evaluation**. For now, we consider the following sample:

- A sample of $n_1 = 297$ **treated** people, randomly selected among participants in the National Supported Work Demonstration.
- A sample of $n_0 = 2490$ **control** people, collected via the Population Survey of Income Dynamics.

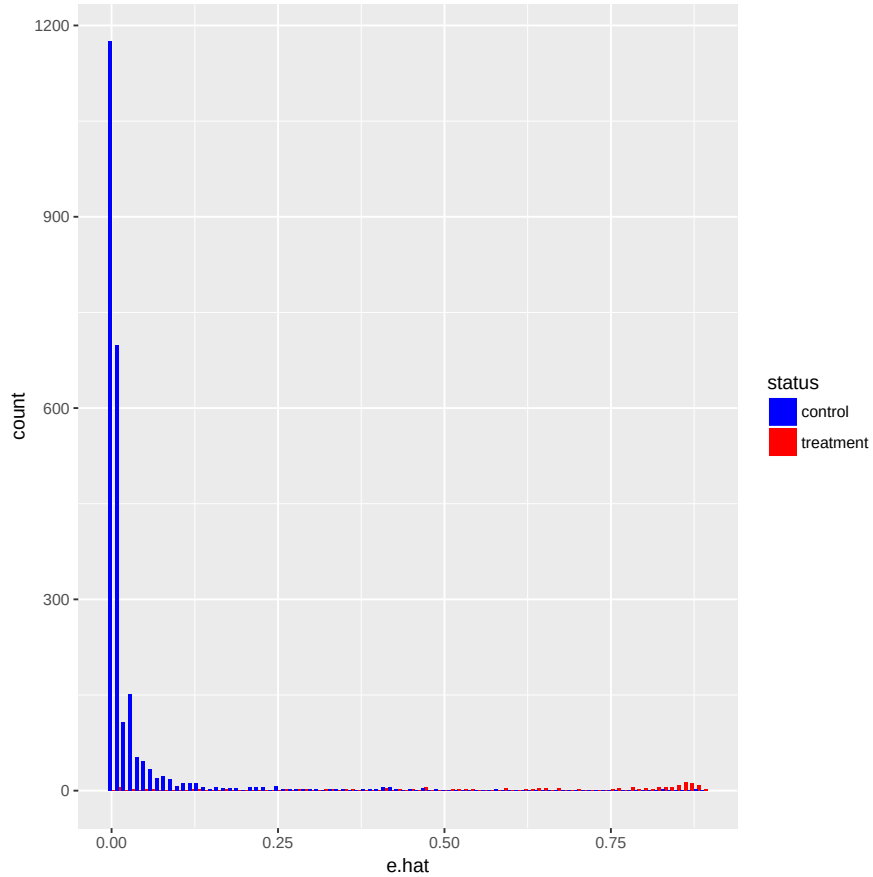
For some values of x, we essentially know the person is a control and so $e(x) \approx 0$. (For example, if the person was already employed at the start of the study.)

Overlap in the Lalonde data



Overlap on the Lalonde dataset, with full set of PSID controls. Many of the controls have essentially 0 propensity, but there is no overlap problem near 1.

Overlap in the Lalonde data



We can find **propensity-matches** for the treated units, but not for all the controls. A simple way to trim away the overlap problem is to estimate an **average treatment effect on the treated**. Here, this may also be better conceptually justified.

Estimands for Causal Inference

The **average treatment effect on the treated** (ATT)
$\mathbb{E}\left[Y_i(1) - Y_i(0) \,\middle|\, W_i = 1\right]$ often has simple interpretation.

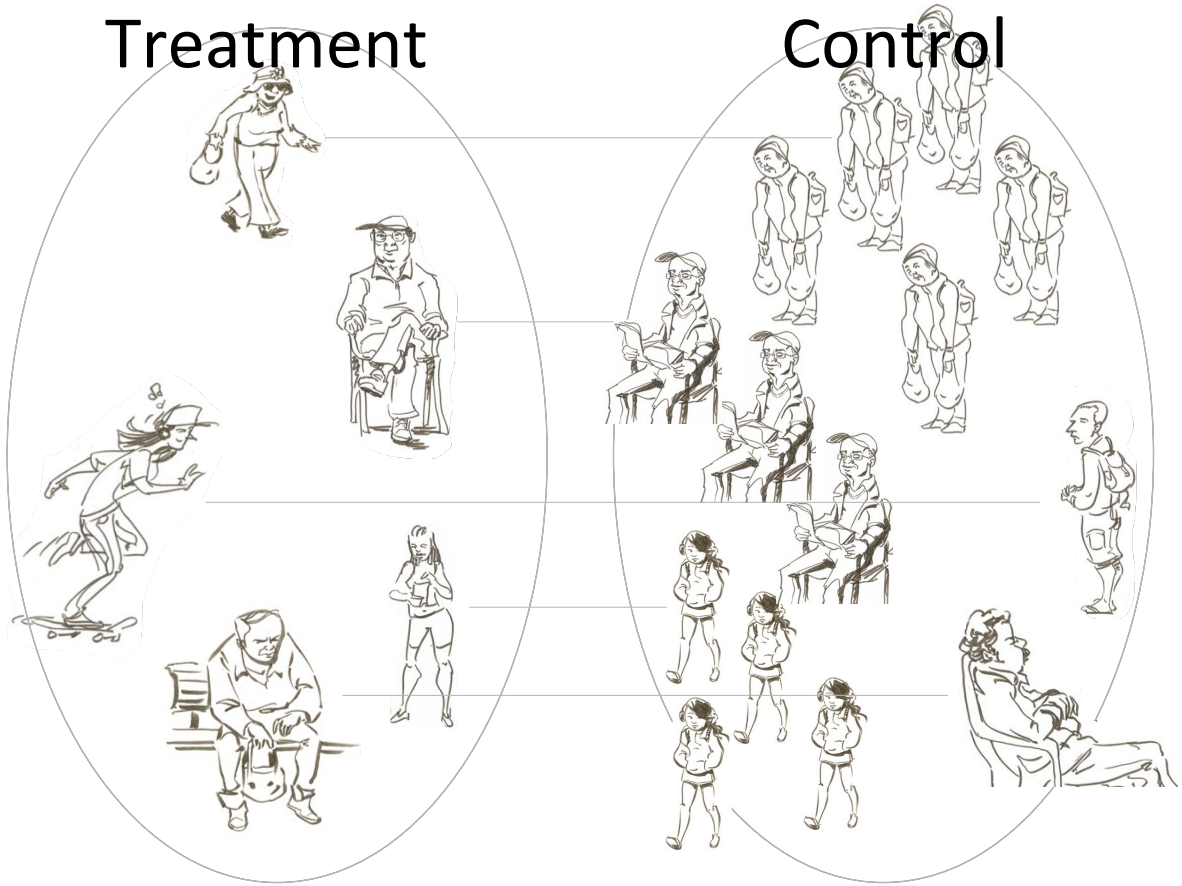


Image credit: J. Zubizarreta.

Average treatment effect on the treated

Recall that the average treatment effect on the treated is

$$\tau_{ATT} = \mathbb{E}\left[Y_i(1) - Y_i(0) \,\middle|\, W_i = 1\right].$$

As usual, we can estimate it via several strategies. The direct **regression adjustment** estimator fits a model $\hat{\mu}_{(0)}(x)$ to the controls, and then uses it to impute what would have happened to the treated units (on average) in the control condition

$$\hat{\tau}_{ATT} = \frac{1}{n_1} \sum_{W_i=1} \left(Y_i - \hat{\mu}_{(0)}(X_i)\right).$$

The **propensity-weighted** estimator uses

$$\hat{\tau}_{ATT} = \frac{1}{n_1} \sum_{W_i=1} Y_i - \sum_{W_i=0} \frac{\hat{e}(X_i)}{1-\hat{e}(X_i)} Y_i \Bigg/ \sum_{W_i=0} \frac{\hat{e}(X_i)}{1-\hat{e}(X_i)}.$$

We never divide by $\hat{e}(x)$, so **propensities near 0** aren't a problem.

Average treatment effect on the treated

The **augmented propensity-weighted** estimator combines both

$$\hat{\tau}_{ATT} = \frac{1}{n_1} \sum_{W_i=1} \left(Y_i - \hat{\mu}_{(0)}(X_i)\right) - \sum_{W_i=0} \frac{\hat{e}(X_i)}{1-\hat{e}(X_i)} \left(Y_i - \hat{\mu}_{(0)}\right) \Bigg/ \sum_{W_i=0} \frac{\hat{e}(X_i)}{1-\hat{e}(X_i)}.$$

Again, this estimator is **asymptotically optimal** via an **orthogonal moments** argument. Also, we can write $\hat{\tau}_{ATT} = n^{-1} \sum_{i=1}^{n} \widehat{\Gamma}_i$,

$$\frac{\widehat{\Gamma}_i}{n} = \frac{W_i \left(Y_i - \hat{\mu}_{(0)}(X_i)\right)}{n_1} - \frac{(1-W_i)\frac{\hat{e}(X_i)}{1-\hat{e}(X_i)} \left(Y_i - \hat{\mu}_{(0)}\right)}{\sum_{W_i=0} \frac{\hat{e}(X_i)}{1-\hat{e}(X_i)}},$$

and, by the same argument as before, we can estimate variance via $\widehat{V}_n = \sum_{i=1}^{n} (\widehat{\Gamma}_i - \hat{\tau}_{ATT})^2 / (n(n-1))$ to build **confidence intervals**.

Overlap in the Lalonde data

```
library(grf) # for random forests
treat_data = read.table("...../nswre74_treated.txt")
control_data = read.table("...../psid_controls.txt")
combined = rbind(treat_data, control_data)
X = combined[,2:9]; Y = combined[,10]; W = combined[,1]
cf = causal_forest(X, Y, W)

ate.hat = average_treatment_effect(cf,
                                   target.sample = "all")
print(paste0("95% CI: ", round(ate.hat["estimate"]),
             " +/- ", round(1.96 * ate.hat["std.err"])))
Warning: Estimated treatment propensities go as low as 0.003...
[1] "95% CI: -3039 +/- 6388"

att.hat = average_treatment_effect(cf,
                                   target.sample = "treated")
print(paste0("95% CI: ", round(att.hat["estimate"]),
             " +/- ", round(1.96 * att.hat["std.err"])))
[1] "95% CI: 1142 +/- 1510"
```

Addressing failures in overlap

If there are some observations with propensities very near 0 and some very near 1, we need **more aggressive** methods:

- One idea is to fit a model for $\hat{e}(x)$, throw away all observations with $\hat{e}(X_i) \leq 0.1$ or $\hat{e}(X_i) \geq 0.9$, and estimate an ATE on the rest.
- Another idea is the weight observations by $\hat{e}(X_i)(1 - \hat{e}(X_i))$, so all observations with extreme weights are strongly enough discounted not to inflate variance.

In both cases, **interpretation** requires care.

Recap

Treatment effects are important in many scientific analyses.

Once we have **identified** treatment effects via unconfoundedness, we can **estimate** them by combining flexible **machine learning** methods with **augmented IPW**.

- Formally, AIPW yields **semiparametrically efficient** estimates of the treatment effect, provided the inputs from machine learning methods are accurate enough.
- In practice, AIPW makes our procedure robust to **regularization bias**.
- AIPW allows from simple **confidence intervals** that do not depend on which specific machine learning method we used.

AIPW lets machine learning focus on what it's good at (i.e., accurate predictions), and then uses its outputs for efficient treatment effect estimation.

Recap

The **AIPW** estimator can be written as $\hat{\tau} = \frac{1}{n}\sum_{i=1}^{n} \widehat{\Gamma}_i$,

$$\widehat{\Gamma}_i = \hat{\mu}_{(1)}(X_i) - \hat{\mu}_{(0)}(X_i) + \frac{W_i}{\hat{e}(X_i)}\left(Y_i - \hat{\mu}_{(1)}(X_i)\right) - \frac{1 - W_i}{1 - \hat{e}(X_i)}\left(Y_i - \hat{\mu}_{(0)}(X_i)\right).$$

We can form level-α **confidence intervals** $\mathcal{I}_\alpha$ as follows (recall that the validity of these confidence intervals is not trivial, and relies on an orthogonal moments argument):

$$\mathcal{I}_\alpha = \hat{\tau} \pm z_{1-\alpha/2}\widehat{V}^{\frac{1}{2}}, \quad \widehat{V} = \frac{1}{n(n-1)}\sum_{i=1}^{n}\left(\widehat{\Gamma}_i - \hat{\tau}\right)^2.$$

The `grf` package has a forest-based **implementation** of AIPW:

```
cf = causal_forest(X, Y, W)
ate.hat = average_treatment_effect(cf)
```