# Neyman Orthogonalisation.

Eduardo Fé

# Introduction

Machine Learning Based Causal Inference is complicated.

Straightforward application of ML methods for inference is bound to fail.

Approaches exist that leverage the flexibility of ML for Causal Inference.

The "why works" of these methods is, however, complex.

The basic building blocks of these methods are

- Influence functions / Neyman Orthogonality
- Cross-Fitting

This first session provides an overview of these topics

# Introduction

These notes are based on:

- Belloni, Chernozhukov, Hansen (2014) Journal of Economic Perspectives
- Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, Robins (2018), Economic Journal
- Belloni, Chernozhukov, Hansen (2011), Review of Economic Studies
- Belloni,Chernozhukov, Chetverikov, Hansen, Kato. arXiv:1806.01888 (2018)
- Hines, Dukes, Díaz Ordaz, Vansteelandt (2021), arXiv:2107.0681
- Kennedy (2022), arXiv:2203.06469

... any errors/misconceptions are all mine.

# Introduction

Setting:

We have data $(Y_i, D_i, W_i)$ from $i$ independent individuals, where

- $Y$ is an outcome,
- $D$ is a binary "treatment" $D_i \in \{0, 1\}$
- $W$ is a set of pre-determined covariates.

The outcome

$$Y_i = Y_i(1) \cdot D_i + Y_i(0) \cdot (1 - D_i)$$

where $Y_i(D_i)$ are potential outcomes.

We are interested in drawing **inferences** about the effects of $D$ on $Y$, such as

$$ATE = E(Y_i(1) - Y_i(0))$$

# Introduction

Three identification challenges.

**Challenge 1**. Assumptions drive estimation and inference.

**Challenge 2**. Nuisance relationships (between $W$ and $D, Y$) affect estimation

**Challenge 3**. Estimators of $\tau$ depend on how well nuisance relationships are modeled[1].

In general if $\eta_{D,W} = \eta_{D,W}(W)$, $\eta_{Y,W} = \eta_{Y,W}(W)$ are the **nuisance** functional relationships between $W$ and $D, Y$,

$$\tau = \tau(\eta_{D,W}, \eta_{Y,W})$$

[1]For a very simple example of this, recall that in an OLS regression model, $Y_i = \alpha + \beta \cdot X_i + \varepsilon_i$, the estimator of $\alpha$ is $\hat{\alpha} = \bar{Y} - \hat{\beta} \cdot \bar{X}$, where $\hat{\beta}$ is the OLS estimator of $\beta$.

# Introduction

Machine Learning methods can assist with Challenges 1 - 3, by allowing us to make assumptions more credible, and estimating nuisance relationship flexibly.

HOWEVER ML introduces a bit of bias to improve the precision of *predictions*.

Used "naively" ML' **regularisation bias** will invalidate our inferences about $\tau$

We want an estimation framework to

- Take advantage of ML's ability to capture non-linearities and select relevant covariates
- While preventing regularisation biases invalidating inferences about $\tau$.

Such framework can be stated in terms of **Neyman Orthogonalisation**.

# Example: naive application of ML.

We simulated data from the model

$$Y_i = \tau \cdot D_i + \sum_{j=1}^{100} W_{i,j} \cdot \beta_j + \varepsilon_i$$

where

$$D_i = \sum_{j=1}^{100} W_{i,j} \cdot \gamma + 0.25 \cdot \nu_i$$

- $\varepsilon_i \sim N(0,1)$, $\nu_i \sim N(0,1)$
- $\tau = 0.5$
- $B = 500$
- $N = 100$
- $W_j$ are multivariate normal with $cov(W_j, W_k) = 0.7^{|j-k|}$
- and $\beta_j = \gamma_j = 1/j^2$.

# Example: naive application of ML.

1. Apply Lasso to select the variables in $Y = \tau \cdot D + W'\beta + \varepsilon$
2. Apply OLS of $Y$ on $D$ and the selected $W$,to estimate $\tau$
3. Draw inferences with the usual t-test.

Denote the estimator $\hat{\tau}_{Naive}$.

We report the

- Mean Bias $= B^{-1} \sum_{b=1}^{B} (\hat{\tau}_{Naive} - \tau)$
- Mean Squared Error (MSE) $= B^{-1} \sum_{b=1}^{B} (\hat{\tau}_{Naive} - \tau)^2$
- Mean Absolute Error (MAE) $= B^{-1} \sum_{b=1}^{B} |\hat{\tau}_{Naive} - \tau|$

# Results

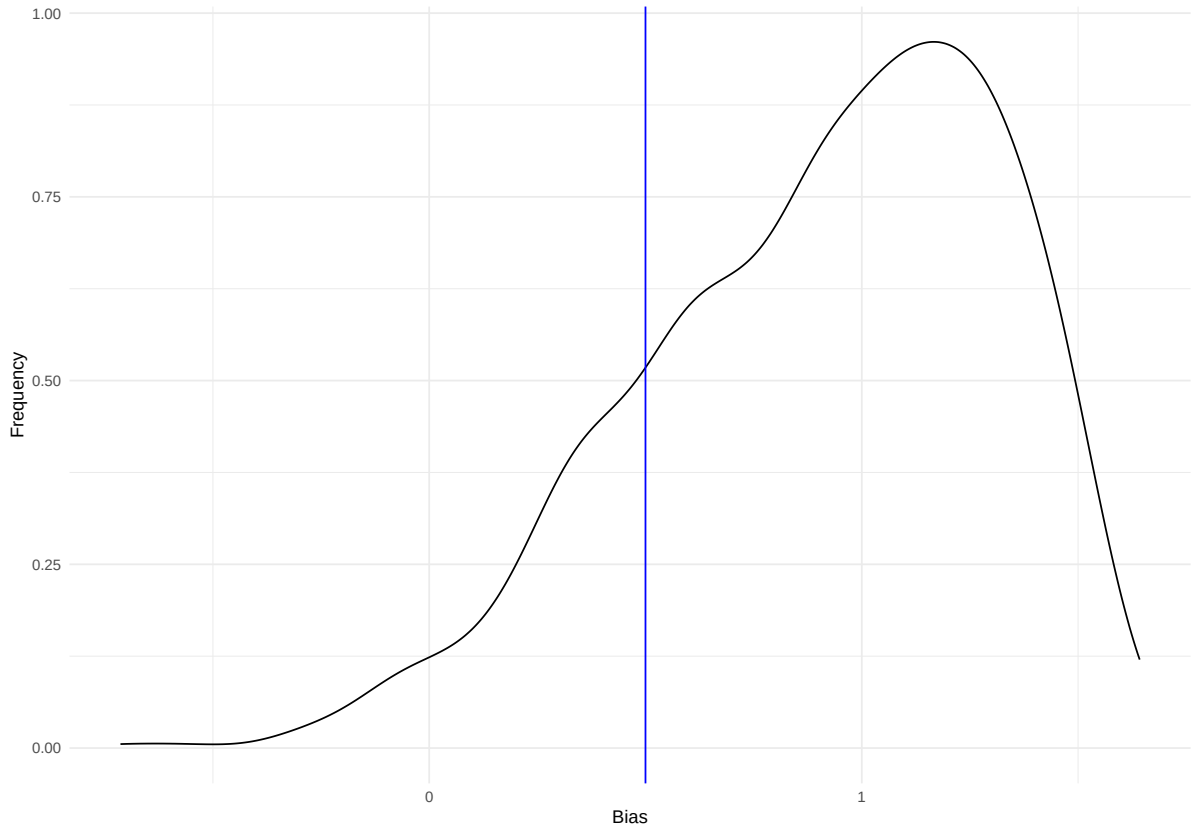


Figure 1: Distribution of the bias.

In the figure, the average estimate of $\tau$ over 500 observations was 0.909 with MSE$=$ 0.994, MAE$=$ 0.917.

# A *Naive* approach.

- The true model is linear
- Lasso estimates a linear model so, it should do a great job...

ML models balance bias and variance to achieve good predictions. In doing so,

- Regularisation makes "mistakes", dropping variables that are relevant to explain $Y$ OR $D$
- Small effects will tend to be wiped out (regularised)

As a result, the model will fail to take into account correlations with deleted variables. This will lead to biases in OLS $\rightarrow$ omitted variable bias.

Note that these biases come because we want to estimate the **nuisance** relationship between $Y$ and $W$.

# Neyman Orthogonality (Non-technical view).

Let $\tau$ be the causal parameter of interest and $\eta$ a **nuisance parameter**.

Most estimators are based on *moment* equations of the type[2]

$$E(f(W;\tau,\eta)) = \mathbf{0}$$

If we can find moment equations, say

$$E(\psi(W;\tau,\eta)) = \mathbf{0}$$

to obtain estimates of $\tau$, and these equations are *insensitive* to biases in the estimation of $\eta$, then we say that the equations $E(\psi(W;\tau,\eta)) = \mathbf{0}$ are **Neyman Orthogonal**.

---

[2]For instance,the linear regression model is estimated by OLS, which solves $E[\mathbf{X}'(\mathbf{y} - \mathbf{X}\beta) = 0$, where some of the elements in $= (\beta_1, ..., \beta_k)$ might not be of intrinsic interest (nuisance) parameters.

# Neyman Orthogonality.

**Double/Debiased Machine Learning (DML)**

1. Run Lasso regressions of $Y$ on $W$ (on the one hand) and $D$ on $W$ (on the other). Let $\hat{\gamma}_{YW}$ and $\hat{\gamma}_{DW}$ be the estimated coefficients of the variables selected by Lasso.

2. Construct the residuals

i) $\hat{u}_{Y,i} = Y_i - W_i'\hat{\gamma}_{YW}$ and
ii) $\hat{u}_{D,i} = D_i - W_i'\hat{\gamma}_{DW}$.

3. Run an Instrumental Variables regression of $\hat{u}_{Y,i}$ on $D$, using $\hat{u}_{D,i}$ as the instrument, to obtain the estimator of $\tau$:

$$\hat{\tau} = \frac{\sum_{i=1}^{n} \hat{u}_{Y,i} \cdot \hat{u}_{D,i}}{\sum_{i=1}^{n} D \cdot \hat{u}_{D,i}} \tag{1}$$

# Simulation (Continued)

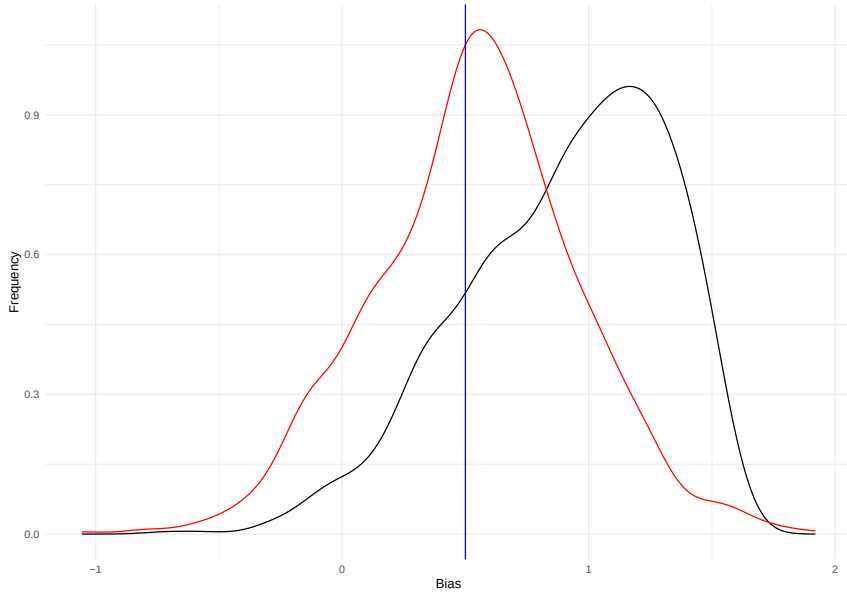


Figure 2: Distribution of the bias.

| Head | Ave | MSE | Bias | MAE |
|---|---|---|---|---|
| Naive | 0.909 | 0.335 | 0.409 | 0.499 |
| DML | 0.541 | 0.175 | 0.041 | 0.327 |

# Why does this work?

Let

- $\eta_0 = (\eta_1, \eta_2) = (\gamma'_{DW}, \gamma'_{YW})'$.
- $u_D = u_D(\eta_0) = D - W'\gamma_{DW}$
- $u_Y = u_Y(\eta_0) = Y - W'\gamma_{YW}$.

OLS solves the sample equivalent to finding $b$ such that $E\Big[(Y - X'b)'u\Big] = 0$.

Therefore, the Double Lasso method thus finds $\tau$ such that

$$E\Big[(u_Y(\eta_0) - \tau \cdot u_D(\eta_0)) \cdot u_D(\eta_0)\Big] = 0.$$

It turns out that the "*derivative*" of the above moment with respect to $\eta_0$ is 0.

That is, we can estimate $\tau$ accurately if we have a "good" estimate of $\eta_0$ (and Lasso provides that).

# Why does this work?

In contrast, OLS after Lasso, solves the sample equivalent to

$$M(\tau, \eta) = E\Big[(Y - \tau D - W'\beta)D\Big] = 0,$$

with

$$\partial_b = E[DW] \neq 0$$

in general.

An exception occurs if $D$ is randomly allocated (as in experiments), in which case Double Lasso and OLS Post Lasso both will perform well.

# Why does this work?

Under certain complex regularity conditions,

$$\sqrt{n}(\hat{\tau} - \tau) \sim N(0, V)$$

where $V$ can be estimated as the variance of the IV estimator.

# Cross Fitting

The properties of the DML estimator depend on how well the relationship between $Y$ and $W$ and $D$ and $W$ are estimated (say $\eta_{YW}$, $\eta_{DW}$).

The ML estimation of the nuisance parameters still incurs in some regularisation error.

Specifically, there is a chance of overfitting (that is, our model does not "smooth" the data and instead "traces" the data -essentially replicating each observation).

Overfitting implies that our estimates of nuisance parameters include spurious variation, which is another source of bias.

# Cross Fitting

For example, suppose data come from the model:

$$Y = \tau \cdot D + W'\beta + \varepsilon \tag{2}$$

$$D = W'\gamma + \nu \tag{3}$$

$$E(\varepsilon|W, D) = E(\nu|W) = 0 \tag{4}$$

Then over-fitting in the DML estimator of $\tau$ will lead to a bias that depends on the variance $\nu$, in a way such that

$$Bias = n^{\delta} V(\nu) \text{ where } 0 < \delta < 1/2 \to \infty \tag{5}$$

This bias can thus be substantive.

# Cross Fitting

We can mitigate the effect of these relationships through **Cross-Fitting**. A basic implementation is as follows:

1. Split the sample into two subsamples, at random, $(Y_i^{(1)}, W_i^{(1)}, D_i^{(1)})$ and $(Y_i^{(2)}, W_i^{(2)}, D_i^{(2)})$.
2. Estimate (with a ML method) the relationships between $Y$ and $W$ and $D$ and $W$ using the first sub-sample, say $\hat{g}(X_i)$ and $\hat{m}(X_i)$ respectively.
3. Get the predicted values of $D$ and $Y$ in the second sub-sample, using the models estimated in step two, $\hat{g}(W_i^{(2)})$ and $\hat{m}(W_i^{(2)})$ respectively.
4. Obtain the residuals $\hat{u}_{Y,i}^{(2)} = Y_i^{(2)} - \hat{g}(W_i^{(2)})$, $\hat{u}_{D,i}^{(2)} = D_i^{(2)} - \hat{m}(W_i^{(2)})$,
5. Regress $\hat{u}_{Y,i}^{(2)}$ on $\hat{u}_{D,i}^{(2)}$, to obtain an estimator of $\theta$.

# Cross Fitting

The previous implementation has the shortcoming of wasting half of the sample.

What we can do is to apply the algorithm twice

- First, $(Y_i^{(1)}, W_i^{(1)}, D_i^{(1)})$ works as the *training sample* and $(Y_i^{(2)}, W_i^{(2)}, D_i^{(2)})$ as the *test sample* to obtain a first estimate $\widehat{\tau}_1$...
- Then, $(Y_i^{(1)}, W_i^{(1)}, D_i^{(1)})$ works as the *test sample* and $(Y_i^{(2)}, W_i^{(2)}, D_i^{(2)})$ as the *training sample* to obtain a second estimate $\widehat{\tau}_2$... - The final cross-fitting estimate is just the average of $\widehat{\tau}_1$ and $\widehat{\tau}_2$.

More generally, we can base Cross Fitting in $K$ folds (instead of 2 folds).

# Cross Fitting

What this achieve is to break the correlation between any over fitting biases in and the error term:

- The observations are being assumed independent
- By splitting the sample, we get independent sub-samples
- Overfitting biases from the training subsamples will be uncorrelated with error terms in the test folds/subsamples.

# Simulation

We created data from

$$Y = \tau \cdot D + g(X_i) + \varepsilon_i \text{ where } \varepsilon_i \sim N(0,1) \tag{6}$$
$$D = m(X_i) + \nu_i \text{ where } \varepsilon_i \sim N(0,1) \tag{7}$$

where

$$g(X_i) = \cos(X_i'\beta)^2 \tag{8}$$
$$m(X_i) = \sin(X_i'\gamma) + \cos(X'\gamma) \tag{9}$$

$\beta_j = 1/j^2$, $j = 1, ..., 10$,and $X_{j,i}$ are multivariate normal with covariance $cov(X_j, X_k) = 0.7^{|j-k|}$.

As before, $\tau = 0.5$. We generated data from the above 1000 times, each time estimating the Double/Debiased estimator with and without cross-fitting (based on Random Forest).

# Simulation

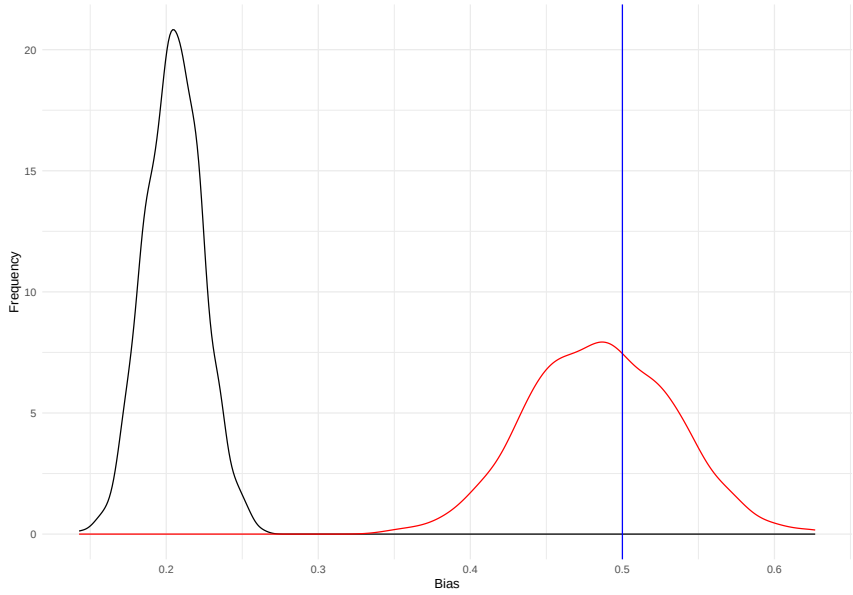


Figure 3: Distribution of the bias.

| Head | Ave | MSE | Bias | MAE |
|---|---|---|---|---|
| X-fit | 0.486 | 0.002 | -0.014 | 0.040 |
| DML | 0.205 | 0.087 | -0.295 | 0.295 |

# Conclusion

- ML has been devised with "prediction" in mind.
- Regularisation allows for a degree of bias, and this means that inference based on ML methods (applied *naively*) will be invalid:
  - Regularisation eliminiates variables that might be important to explain the effect of a specific covariate (this creates omitted variable biases)
  - Regularisation also forces coefficients of selected variables towards 0 (another source of bias)
- We can apply "Orthonogalisation" techniques to take advantage of ML's estimation abilities whilst also undertaking inference.
- We often will benefit from combining Orthogonalisation with Cross Fitting (to eliminate biases inherited from the ML prediction step).

# Conclusion

However:

- Implicit assumption is that our data has ALL RELEVANT variables to explain $Y$ and
- If some important variable is missing in your data (or measured with error) the procedures described before will fail:

**Sophistication must not trump Identification.**

Also, we have assumed our data contains independent observations. If that is not the case:

- Your estimates are likely to still be accurate
- Although Cross Fitting is not guaranteed to work
- Tests will be generally invalid.