Chernozhukov et al. on Double / Debiased Machine Learning

Slides by Chris Felton Princeton University, Sociology Department

Sociology Statistics Reading Group November 2018

Papers

Papers

• Background:

• Robinson, Peter. 1988. "Root-*N*-Consistent Semiparametric Regression," *Econometrica* 56(4):931-954.

Papers

• Background:

• Robinson, Peter. 1988. "Root-*N*-Consistent Semiparametric Regression," *Econometrica* 56(4):931-954.

• Main paper:

 Chernozhukov, Victor, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. 2018. "Double / Debiased Machine Learning for Treatment and Structural Parameters," *Econometrics Journal* 21(1):1-68.

 Provides a general framework for estimating treatment effects using machine learning (ML) methods

- Provides a general framework for estimating treatment effects using machine learning (ML) methods
 - In particular, we can use any (preferably $n^{\frac{1}{4}}$ -consistent) ML estimator with this approach

- Provides a general framework for estimating treatment effects using machine learning (ML) methods
 - In particular, we can use any (preferably $n^{\frac{1}{4}}$ -consistent) ML estimator with this approach
- Enables us to construct valid **confidence intervals** for our treatment effect estimates

- Provides a general framework for estimating treatment effects using machine learning (ML) methods
 - In particular, we can use any (preferably $n^{\frac{1}{4}}$ -consistent) ML estimator with this approach
- Enables us to construct valid **confidence intervals** for our treatment effect estimates
- Introduces a \sqrt{n} -consistent estimator

- Provides a general framework for estimating treatment effects using machine learning (ML) methods
 - In particular, we can use any (preferably $n^{\frac{1}{4}}$ -consistent) ML estimator with this approach
- Enables us to construct valid **confidence intervals** for our treatment effect estimates
- Introduces a $\sqrt{n-consistent}$ estimator
 - As $n \to \infty$, the estimation error $\hat{\theta} \theta$ goes to zero at a rate of $n^{-\frac{1}{2}}$ (or $1/\sqrt{n}$)

- Provides a general framework for estimating treatment effects using machine learning (ML) methods
 - In particular, we can use any (preferably $n^{\frac{1}{4}}$ -consistent) ML estimator with this approach
- Enables us to construct valid **confidence intervals** for our treatment effect estimates
- Introduces a $\sqrt{n-consistent}$ estimator
 - As $n \to \infty$, the estimation error $\hat{\theta} \theta$ goes to zero at a rate of $n^{-\frac{1}{2}}$ (or $1/\sqrt{n}$)
 - We really like our estimators to be at least \sqrt{n} -consistent

- Provides a general framework for estimating treatment effects using machine learning (ML) methods
 - In particular, we can use any (preferably $n^{\frac{1}{4}}$ -consistent) ML estimator with this approach
- Enables us to construct valid **confidence intervals** for our treatment effect estimates
- Introduces a \sqrt{n} -consistent estimator
 - As $n \to \infty$, the estimation error $\hat{\theta} \theta$ goes to zero at a rate of $n^{-\frac{1}{2}}$ (or $1/\sqrt{n}$)
 - $\, \bullet \,$ We really like our estimators to be at least $\sqrt{n}\text{-consistent}$
 - $1/n^{\frac{1}{2}}$ will approach 0 more quickly than, e.g., $1/n^{\frac{1}{4}}$ as *n* grows

• In observational studies, we often estimate causal effects by conditioning on confounders

- In observational studies, we often estimate causal effects by conditioning on confounders
- We typically condition on confounders by making strong assumptions about the functional form of our model

- In observational studies, we often estimate causal effects by conditioning on confounders
- We typically condition on confounders by making **strong assumptions** about the functional form of our model
 - E.g., a standard OLS model assumes a linear and additive conditional expectation function

- In observational studies, we often estimate causal effects by conditioning on confounders
- We typically condition on confounders by making **strong assumptions** about the functional form of our model
 - E.g., a standard OLS model assumes a linear and additive conditional expectation function
- If we misspecify the functional form, we will end up with **biased estimates** of treatment effects even in the absence of unmeasured confounding

- In observational studies, we often estimate causal effects by conditioning on confounders
- We typically condition on confounders by making **strong assumptions** about the functional form of our model
 - E.g., a standard OLS model assumes a linear and additive conditional expectation function
- If we misspecify the functional form, we will end up with **biased estimates** of treatment effects even in the absence of unmeasured confounding
- Our parametric specifications often lack strong substantive justification

- In observational studies, we often estimate causal effects by conditioning on confounders
- We typically condition on confounders by making **strong assumptions** about the functional form of our model
 - E.g., a standard OLS model assumes a linear and additive conditional expectation function
- If we misspecify the functional form, we will end up with biased estimates of treatment effects even in the absence of unmeasured confounding
- Our parametric specifications often **lack strong substantive** justification
- ML provides a systematic framework for learning the form of the conditional expectation function from the data

We sometimes find ourselves working with high-dimensional data

- We sometimes find ourselves working with high-dimensional data
 - I.e., we have many covariates *p* relative to the number of observations *n*

- We sometimes find ourselves working with high-dimensional data
 - I.e., we have many covariates *p* relative to the number of observations *n*
- Two types of high-dimensionality:

- We sometimes find ourselves working with high-dimensional data
 - I.e., we have many covariates *p* relative to the number of observations *n*
- Two types of high-dimensionality:
 - We simply have many measured covariates (e.g., text data, genetic data)

- We sometimes find ourselves working with high-dimensional data
 - I.e., we have many covariates *p* relative to the number of observations *n*
- Two types of high-dimensionality:
 - We simply have many measured covariates (e.g., text data, genetic data)
 - 2 We have few measured covariates but wish to generate many non-linear transformations of and interactions between these covariates

- We sometimes find ourselves working with high-dimensional data
 - I.e., we have many covariates *p* relative to the number of observations *n*
- Two types of high-dimensionality:
 - We simply have many measured covariates (e.g., text data, genetic data)
 - 2 We have few measured covariates but wish to generate many non-linear transformations of and interactions between these covariates
- ML models perform much better in high dimensions than traditional statistical models do

Image: ML allows us to do causal inference with minimal assumptions about the functional form of our model

- Image: ML allows us to do causal inference with minimal assumptions about the functional form of our model
 - Warning: ML does not help us relax identification assumptions (e.g., no unmeasured confounding, parallel trends, exclusion restriction, etc.)

- Image: ML allows us to do causal inference with minimal assumptions about the functional form of our model
 - Warning: ML does not help us relax identification assumptions (e.g., no unmeasured confounding, parallel trends, exclusion restriction, etc.)
- ML allows us to do causal inference with high-dimensional data

• ML methods were designed for prediction

- ML methods were designed for prediction
- But off-the-shelf ML methods are biased estimators for treatment effects

- ML methods were designed for prediction
- But off-the-shelf ML methods are biased estimators for treatment effects
 - $\, \bullet \,$ To minimize $MSE = bias^2 + variance, we trade off variance for bias$

- ML methods were designed for prediction
- But off-the-shelf ML methods are biased estimators for treatment effects
 - $\, \bullet \,$ To minimize $MSE = bias^2 + variance, we trade off variance for bias$
- Consistent ML methods converge more slowly than $\frac{1}{\sqrt{n}}$

- ML methods were designed for prediction
- But off-the-shelf ML methods are biased estimators for treatment effects
 - $\, \bullet \,$ To minimize $MSE = bias^2 + variance, we trade off variance for bias$
- Consistent ML methods converge more slowly than $\frac{1}{\sqrt{n}}$
- Off-the-shelf methods also fail to provide confidence intervals for our treatment effect estimates

Eliminate the bias

- Eliminate the bias
- 2 Achieve \sqrt{n} -consistency

- Eliminate the bias
- ² Achieve \sqrt{n} -consistency
- ③ Construct valid confidence intervals

Introduce partially linear model set-up

- 1 Introduce partially linear model set-up
- ② Explain two sources of estimation bias from ML and how we overcome them

- 1 Introduce partially linear model set-up
- ② Explain two sources of estimation bias from ML and how we overcome them
 - Correct bias from regularization with Neyman orthogonality

- Introduce partially linear model set-up
- ② Explain two sources of estimation bias from ML and how we overcome them
 - Correct bias from regularization with Neyman orthogonality
 - We will see that we can achieve Neyman orthogonality using a residuals-on-residuals approach reminiscent of Robinson (1988) and the Frisch-Waugh-Lovell theorem

- Introduce partially linear model set-up
- ② Explain two sources of estimation bias from ML and how we overcome them
 - Correct bias from regularization with Neyman orthogonality
 - We will see that we can achieve Neyman orthogonality using a residuals-on-residuals approach reminiscent of Robinson (1988) and the Frisch-Waugh-Lovell theorem
 - Correct bias from overfitting using sample-splitting

- Introduce partially linear model set-up
- ② Explain two sources of estimation bias from ML and how we overcome them
 - Correct bias from regularization with Neyman orthogonality
 - We will see that we can achieve Neyman orthogonality using a residuals-on-residuals approach reminiscent of Robinson (1988) and the Frisch-Waugh-Lovell theorem
 - Correct bias from overfitting using sample-splitting
 - Employ cross-fitting to avoid the loss of efficiency that normally comes with sample-splitting

- Introduce partially linear model set-up
- ② Explain two sources of estimation bias from ML and how we overcome them
 - Correct bias from regularization with Neyman orthogonality
 - We will see that we can achieve Neyman orthogonality using a residuals-on-residuals approach reminiscent of Robinson (1988) and the Frisch-Waugh-Lovell theorem
 - Correct bias from overfitting using sample-splitting
 - Employ cross-fitting to avoid the loss of efficiency that normally comes with sample-splitting
- ③ Outline a procedure for conducting inference with DML

- Introduce partially linear model set-up
- ② Explain two sources of estimation bias from ML and how we overcome them
 - Correct bias from regularization with Neyman orthogonality
 - We will see that we can achieve Neyman orthogonality using a residuals-on-residuals approach reminiscent of Robinson (1988) and the Frisch-Waugh-Lovell theorem
 - Correct bias from overfitting using sample-splitting
 - Employ cross-fitting to avoid the loss of efficiency that normally comes with sample-splitting
- ③ Outline a procedure for conducting inference with DML
- ④ Examine estimators for the ATE and variance that go beyond the partially linear model set-up

Don't worry if none of that makes sense yet!

Don't worry if none of that makes sense yet!

It's not as complicated as it sounds, and we will work through it slowly.

$$Y = D heta_0 + g_0(X) + U$$

 $D = m_0(X) + V$

$$egin{aligned} Y &= D heta_0 + g_0(X) + U \ D &= m_0(X) + V \end{aligned}$$

• Y: Outcome

$$Y = D heta_0 + g_0(X) + U$$

 $D = m_0(X) + V$

- Y: Outcome
- **D**: Treatment

$$Y = D\theta_0 + g_0(X) + U$$
$$D = m_0(X) + V$$

- Y: Outcome
- D: Treatment
- X: Measured confounders

$$egin{aligned} Y &= D heta_0 + g_0(X) + m{U} \ D &= m_0(X) + m{V} \end{aligned}$$

- Y: Outcome
- D: Treatment
- X: Measured confounders
- U and V are our error terms

$$Y = D heta_0 + g_0(X) + U$$

 $D = m_0(X) + V$

- Y: Outcome
- D: Treatment
- X: Measured confounders
- U and V are our error terms
- We assume zero conditional mean:

$$\mathsf{E}[U \mid X, D] = 0 \qquad \mathsf{E}[V \mid X] = 0$$

$$Y = D heta_0 + g_0(X) + U$$

 $D = m_0(X) + V$

• θ_0 : The true treatment effect

$$Y = D heta_0 + g_0(X) + U$$

 $D = m_0(X) + V$

- θ_0 : The true treatment effect
 - "theta-naught"

$$egin{aligned} Y &= D heta_0 + g_0(X) + U \ D &= m_0(X) + V \end{aligned}$$

- θ_0 : The true treatment effect
 - "theta-naught"
 - Warning: not necessarily the Average Treatment Effect

$$egin{aligned} Y &= D heta_0 + g_0(X) + U \ D &= m_0(X) + V \end{aligned}$$

- θ_0 : The true treatment effect
 - "theta-naught"
 - Warning: not necessarily the Average Treatment Effect
 - Regression gives us a weighted average of individual treatment effects where weights are determined by the conditional variance of treatment (see Aronow and Samii 2016)

$$Y = D\theta_0 + g_0(X) + U$$
$$D = m_0(X) + V$$

- θ_0 : The true treatment effect
 - "theta-naught"
 - Warning: not necessarily the Average Treatment Effect
 - Regression gives us a weighted average of individual treatment effects where weights are determined by the conditional variance of treatment (see Aronow and Samii 2016)
- g₀(·): some function mapping X to Y, conditional on D

$$Y = D\theta_0 + g_0(X) + U$$
$$D = m_0(X) + V$$

- θ_0 : The true treatment effect
 - "theta-naught"
 - Warning: not necessarily the Average Treatment Effect
 - Regression gives us a weighted average of individual treatment effects where weights are determined by the conditional variance of treatment (see Aronow and Samii 2016)
- g₀(·): some function mapping X to Y, conditional on D
- $m_0(\cdot)$: some function mapping X to D

$$egin{aligned} Y &= D heta_0 + g_0(X) + U \ D &= m_0(X) + V \end{aligned}$$

• This set-up allows both Y and D to be non-linear and interactive functions of X in contrast to standard OLS, which assumes a linear and additive model

$$egin{aligned} Y &= D heta_0 + g_0(X) + U \ D &= m_0(X) + V \end{aligned}$$

- This set-up allows both Y and D to be non-linear and interactive functions of X in contrast to standard OLS, which assumes a linear and additive model
- However, note that this partially linear model assumes that the effect of *D* on *Y* is additive and linear

$$egin{aligned} Y &= D heta_0 + g_0(X) + U \ D &= m_0(X) + V \end{aligned}$$

- This set-up allows both Y and D to be non-linear and interactive functions of X in contrast to standard OLS, which assumes a linear and additive model
- However, note that this partially linear model assumes that the effect of *D* on *Y* is additive and linear
- Our confounders can interact with one another, but not with our treatment!

$$egin{aligned} Y &= D heta_0 + g_0(X) + U \ D &= m_0(X) + V \end{aligned}$$

- This set-up allows both Y and D to be non-linear and interactive functions of X in contrast to standard OLS, which assumes a linear and additive model
- However, note that this partially linear model assumes that the effect of *D* on *Y* is **additive** and **linear**
- Our confounders can interact with one another, but not with our treatment!
- And remember we're still making the standard identification assumptions (unconfoundedness conditional on X, positivity, and consistency)

$$Y = D heta_0 + g_0(X) + U$$

 $D = m_0(X) + V$

 If ML is useful because it allows us to relax linearity and additivity, why would we assume linearity and additivity in D?

$$egin{aligned} Y &= D heta_0 + g_0(X) + U \ D &= m_0(X) + V \end{aligned}$$

- If ML is useful because it allows us to relax linearity and additivity, why would we assume linearity and additivity in D?
- We're just using this model for illustration!

$$Y = D heta_0 + g_0(X) + U$$

 $D = m_0(X) + V$

- If ML is useful because it allows us to relax linearity and additivity, why would we assume linearity and additivity in D?
- We're just using this model for illustration!
- We can assume a fully interactive and non-linear model when we actually use DML

$$egin{aligned} Y &= D heta_0 + g_0(X) + U \ D &= m_0(X) + V \end{aligned}$$

- If ML is useful because it allows us to relax linearity and additivity, why would we assume linearity and additivity in *D*?
- We're just using this model for illustration!
- We can assume a fully interactive and non-linear model when we actually use DML
- But our partially linear model set-up will allow us to better explain how DML works

• We've introduced the partially linear model set-up

- We've introduced the partially linear model set-up
- Next, we will introduce an intuitive procedure—"the naive approach"—for estimating θ_0 with ML assuming a partially linear model

- We've introduced the partially linear model set-up
- Next, we will introduce an intuitive procedure—"the naive approach"—for estimating θ_0 with ML assuming a partially linear model
- We will show that this estimation procedure is biased and not \sqrt{n} -consistent

- We've introduced the partially linear model set-up
- Next, we will introduce an intuitive procedure— "the naive approach"—for estimating θ₀ with ML assuming a partially linear model
- We will show that this estimation procedure is biased and not \sqrt{n} -consistent
- Then we're going to illustrate two sources of this bias and show how DML avoids these two types of bias

• The naive approach: estimate $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \widehat{U}$ using ML

- The naive approach: estimate $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \widehat{U}$ using ML
 - $\hat{\theta}_0$: our naive estimate of θ_0 , "theta-naught-hat"

- The naive approach: estimate $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \widehat{U}$ using ML
 - $\hat{\theta}_0$: our naive estimate of θ_0 , "theta-naught-hat"
- How might we estimate $\hat{\theta}_0$ and $\hat{g}_0(X)$?

- The naive approach: estimate $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \widehat{U}$ using ML
 - $\hat{\theta}_0$: our naive estimate of θ_0 , "theta-naught-hat"
- How might we estimate $\hat{\theta}_0$ and $\hat{g}_0(X)$?
 - Remember, $m_0(X) = E[D \mid X]$, but $g_0(X) \neq E[Y \mid X]$

- The naive approach: estimate $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \widehat{U}$ using ML
 - $\hat{\theta}_0$: our naive estimate of θ_0 , "theta-naught-hat"
- How might we estimate $\hat{\theta}_0$ and $\hat{g}_0(X)$?
 - Remember, $m_0(X) = E[D \mid X]$, but $g_0(X) \neq E[Y \mid X]$
 - ${\scriptstyle \bullet }$ That's because $D\theta_0$ is also included in this model

- The naive approach: estimate $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \widehat{U}$ using ML
 - $\hat{\theta}_0$: our naive estimate of θ_0 , "theta-naught-hat"
- How might we estimate $\hat{\theta}_0$ and $\hat{g}_0(X)$?
 - Remember, $m_0(X) = E[D \mid X]$, but $g_0(X) \neq E[Y \mid X]$
 - ${\scriptstyle \bullet }$ That's because $D\theta_0$ is also included in this model
 - In the paper, the authors use $\ell_0(X)$ for $\mathsf{E}[Y \mid X]$

- The naive approach: estimate $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \widehat{U}$ using ML
 - $\hat{\theta}_0$: our naive estimate of θ_0 , "theta-naught-hat"
- How might we estimate $\hat{\theta}_0$ and $\hat{g}_0(X)$?
 - Remember, $m_0(X) = E[D \mid X]$, but $g_0(X) \neq E[Y \mid X]$
 - ${\scriptstyle \bullet }$ That's because $D\theta_0$ is also included in this model
 - In the paper, the authors use $\ell_0(X)$ for $\mathsf{E}[Y \mid X]$
- To estimate both $\hat{\theta}_0$ and $\hat{g}_0(X)$, we could use an iterative method that alternates between using random forest for estimating $\hat{g}_0(X)$ and OLS for estimating $\hat{\theta}_0$

- The naive approach: estimate $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \widehat{U}$ using ML
 - $\hat{\theta}_0$: our naive estimate of θ_0 , "theta-naught-hat"
- How might we estimate $\hat{\theta}_0$ and $\hat{g}_0(X)$?
 - Remember, $m_0(X) = E[D \mid X]$, but $g_0(X) \neq E[Y \mid X]$
 - ${\scriptstyle \bullet }$ That's because $D\theta_0$ is also included in this model
 - In the paper, the authors use $\ell_0(X)$ for $\mathsf{E}[Y \mid X]$
- To estimate both $\hat{\theta}_0$ and $\hat{g}_0(X)$, we could use an iterative method that alternates between using random forest for estimating $\hat{g}_0(X)$ and OLS for estimating $\hat{\theta}_0$
- Alternatively, we could generate many non-linear transformations of the covariates in X as well as interactions between these covariates and use LASSO to estimate the model

Bias from regularization

• To avoid overfitting the data with complex functional forms, ML algorithms use regularization

- To avoid overfitting the data with complex functional forms, ML algorithms use regularization
- This decreases the variance of the estimator and reduces overfitting...

- To avoid overfitting the data with complex functional forms, ML algorithms use regularization
- This decreases the variance of the estimator and reduces overfitting...
- ...but introduces bias and prevents \sqrt{n} -consistency

- To avoid overfitting the data with complex functional forms, ML algorithms use regularization
- This decreases the variance of the estimator and reduces overfitting...
- ...but introduces bias and prevents \sqrt{n} -consistency
- 2 Bias from overfitting

- To avoid overfitting the data with complex functional forms, ML algorithms use regularization
- This decreases the variance of the estimator and reduces overfitting...
- ...but introduces bias and prevents \sqrt{n} -consistency
- 2 Bias from overfitting
 - Sometimes our efforts to regularize fail to prevent overfitting

Bias from regularization

- To avoid overfitting the data with complex functional forms, ML algorithms use regularization
- This decreases the variance of the estimator and reduces overfitting...
- ...but introduces bias and prevents \sqrt{n} -consistency

Bias from overfitting

- Sometimes our efforts to regularize fail to prevent overfitting
- Overfitting: mistaking noise for signal

Bias from regularization

- To avoid overfitting the data with complex functional forms, ML algorithms use regularization
- This decreases the variance of the estimator and reduces overfitting...
- ...but introduces bias and prevents \sqrt{n} -consistency

Bias from overfitting

- Sometimes our efforts to regularize fail to prevent overfitting
- Overfitting: mistaking noise for signal
- More carefully, we overfit when we model the idiosyncrasies of our particular sample too closely, which may lead to poor out-of-sample performance

Bias from regularization

- To avoid overfitting the data with complex functional forms, ML algorithms use regularization
- This decreases the variance of the estimator and reduces overfitting...
- ...but introduces bias and prevents \sqrt{n} -consistency

Bias from overfitting

- Sometimes our efforts to regularize fail to prevent overfitting
- Overfitting: mistaking noise for signal
- More carefully, we overfit when we model the idiosyncrasies of our particular sample too closely, which may lead to poor out-of-sample performance
- $\, \bullet \,$ Overfitting $\, \rightarrow \,$ bias and slow convergence

• For clarity, we will isolate each type of bias

- For clarity, we will isolate each type of bias
- When we look at regularization bias, we will assume we have used sample-splitting to avoid bias from overfitting

- For clarity, we will isolate each type of bias
- When we look at regularization bias, we will assume we have used sample-splitting to avoid bias from overfitting
- When we look at bias from overfitting, we will assume we have used orthogonalization to prevent regularization bias

- For clarity, we will isolate each type of bias
- When we look at regularization bias, we will assume we have used sample-splitting to avoid bias from overfitting
- When we look at bias from overfitting, we will assume we have used orthogonalization to prevent regularization bias
- We'll explain how sample-splitting and orthogonalization work soon

Let's start by looking at the scaled estimation error in $\hat{\theta}_0$ when we use sample-splitting without orthogonalization

$$\sqrt{n}(\hat{\theta}_{0} - \theta_{0}) = \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_{i}^{2}\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_{i}U_{i}}_{:=a} + \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_{i}^{2}\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_{i}(g_{0}(X_{i}) - \hat{g}_{0}(X_{i}))}_{:=b}$$

$$\sqrt{n}(\hat{\theta}_{0} - \theta_{0}) = \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_{i}^{2}\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_{i}U_{i}}_{:=a} + \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_{i}^{2}\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_{i}(g_{0}(X_{i}) - \hat{g}_{0}(X_{i}))}_{:=b}$$

• This looks scary, so let's take it one term at a time

$$\sqrt{n}(\hat{\theta}_0 - \theta_0) = \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_iU_i}_{:=a} + \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_i(g_0(X_i) - \hat{g}_0(X_i))}_{:=b}$$

• This looks scary, so let's take it one term at a time • $\sqrt{n}(\hat{\theta}_0 - \theta_0)$ represents our scaled estimation error

$$\sqrt{n}(\hat{\theta}_{0} - \theta_{0}) = \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_{i}^{2}\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_{i}U_{i}}_{:=a} + \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_{i}^{2}\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_{i}(g_{0}(X_{i}) - \hat{g}_{0}(X_{i}))}_{:=b}$$

- This looks scary, so let's take it one term at a time • $\sqrt{n}(\hat{\theta}_0 - \theta_0)$ represents our scaled estimation error
- If we want consistency, we want our estimation error to go to zero

$$\sqrt{n}(\hat{\theta}_0 - \theta_0) = \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_iU_i}_{:=a} + \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_i^2\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_i(g_0(X_i) - \hat{g}_0(X_i))}_{:=b}$$

This looks scary, so let's take it one term at a time
√n(θ̂₀ - θ₀) represents our scaled estimation error
If we want consistency, we want this term to go to zero
a → N(0, Σ). Great!

$$\sqrt{n}(\hat{\theta}_{0} - \theta_{0}) = \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_{i}^{2}\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_{i}U_{i}}_{:=a} + \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_{i}^{2}\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_{i}(g_{0}(X_{i}) - \hat{g}_{0}(X_{i}))}_{:=b}$$

• **b** is the sum of terms that do not have mean zero divided by \sqrt{n}

Regularization Bias

$$\sqrt{n}(\hat{\theta}_{0} - \theta_{0}) = \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_{i}^{2}\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_{i}U_{i}}_{:=a} + \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_{i}^{2}\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_{i}(g_{0}(X_{i}) - \hat{g}_{0}(X_{i}))}_{:=b}$$

- b is the sum of terms that do not have mean zero divided by \sqrt{n}
- Specifically, $g_0(X_i) \hat{g}_0(X_i)$ will not have mean zero because \hat{g}_0 is biased

Regularization Bias

$$\sqrt{n}(\hat{\theta}_{0} - \theta_{0}) = \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_{i}^{2}\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_{i}U_{i}}_{:=a} + \underbrace{\left(\frac{1}{n}\sum_{i\in I}D_{i}^{2}\right)^{-1}\frac{1}{\sqrt{n}}\sum_{i\in I}D_{i}(g_{0}(X_{i}) - \hat{g}_{0}(X_{i}))}_{:=b}$$

- b is the sum of terms that do not have mean zero divided by \sqrt{n}
- Specifically, $g_0(X_i) \hat{g}_0(X_i)$ will not have mean zero because \hat{g}_0 is biased
- *b* will approach 0, but too slowly for our estimator to be \sqrt{n} -consistent!

• To overcome this regularization bias, let's use orthogonalization

- To overcome this regularization bias, let's use orthogonalization
- Instead of fitting one ML model, we fit two:

- To overcome this regularization bias, let's use orthogonalization
- Instead of fitting one ML model, we fit two:
 - 1 Estimate $D = \widehat{m}_0(X) + \widehat{V}$, our treatment model

- To overcome this regularization bias, let's use orthogonalization
- Instead of fitting one ML model, we fit two:
 - 1 Estimate $D = \widehat{m}_0(X) + \widehat{V}$, our treatment model
 - 2 Estimate $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \hat{U}$ as we do in the naive approach, our outcome model

- To overcome this regularization bias, let's use orthogonalization
- Instead of fitting one ML model, we fit two:
 - 1 Estimate $D = \widehat{m}_0(X) + \widehat{V}$, our treatment model
 - 2 Estimate $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \hat{U}$ as we do in the naive approach, our outcome model

$${ ilde 3}$$
 Regress $Y - \hat g_0(X)$ on $\widehat V$

- To overcome this regularization bias, let's use orthogonalization
- Instead of fitting one ML model, we fit two:
 - 1 Estimate $D = \widehat{m}_0(X) + \widehat{V}$, our treatment model
 - 2 Estimate $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \hat{U}$ as we do in the naive approach, our outcome model

3 Regress
$$Y - \hat{g}_0(X)$$
 on \widehat{V}

 The resulting
 *θ*₀ ("theta-naught-check") is free of regularization bias!

- To overcome this regularization bias, let's use orthogonalization
- Instead of fitting one ML model, we fit two:
 - 1 Estimate $D = \widehat{m}_0(X) + \widehat{V}$, our treatment model
 - 2 Estimate $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \hat{U}$ as we do in the naive approach, our outcome model
 - 3 Regress $Y \hat{g}_0(X)$ on \widehat{V}
- The resulting $\check{\theta}_0$ ("theta-naught-check") is free of regularization bias!
- We can call this a "partialling-out" approach because we have partialled out the associations between X and D and between Y and X (conditional on D)

In notation:

$$\check{\theta}_0 = \left(\frac{1}{n}\sum_{i\in I}\,\widehat{V}_i D_i\right)^{-1} \frac{1}{n}\sum_{i\in I}\,\widehat{V}(Y_i - \hat{g}_0(X_i))$$

• In notation:

$$\check{\theta}_0 = \left(\frac{1}{n}\sum_{i\in I}\,\widehat{V}_i D_i\right)^{-1} \frac{1}{n}\sum_{i\in I}\,\widehat{V}(Y_i - \hat{g}_0(X_i))$$

• Look familiar?

In notation:

$$\check{\theta}_0 = \left(\frac{1}{n}\sum_{i\in I}\widehat{V}_i D_i\right)^{-1} \frac{1}{n}\sum_{i\in I}\widehat{V}(Y_i - \hat{g}_0(X_i))$$

- Look familiar?
- What about now?

$$\widehat{\beta}_{\mathsf{IV}} = (Z'D)^{-1}Z'y$$

In notation:

$$\check{\theta}_0 = \left(\frac{1}{n}\sum_{i\in I}\widehat{V}_i D_i\right)^{-1} \frac{1}{n}\sum_{i\in I}\widehat{V}(Y_i - \hat{g}_0(X_i))$$

- Look familiar?
- What about now?

$$\widehat{\beta}_{\mathsf{IV}} = (Z'D)^{-1}Z'y$$

 It's very similar to our standard linear instrumental variable estimator, two-stage least squares!

 Remember b from the scaled estimation error equation? Now we have b*:

 Remember b from the scaled estimation error equation? Now we have b*:

$$b^* = (E[V^2])^{-1} \frac{1}{\sqrt{n}} \sum_{i \in I} \underbrace{(\widehat{m}_0(X_i) - m_0(X_i))}_{\widehat{m}_0 \text{ estimation error}} \underbrace{(\widehat{g}_0(X_i) - g_0(X_i))}_{\widehat{g}_0 \text{ estimation error}}$$

 Remember b from the scaled estimation error equation? Now we have b*:

$$b^* = (E[V^2])^{-1} \frac{1}{\sqrt{n}} \sum_{i \in I} \underbrace{(\widehat{m}_0(X_i) - m_0(X_i))}_{\widehat{m}_0 \text{ estimation error}} \underbrace{(\widehat{g}_0(X_i) - g_0(X_i))}_{\widehat{g}_0 \text{ estimation error}}$$

 Because this term is based on the product of two estimation errors, it vanishes more quickly

 Remember b from the scaled estimation error equation? Now we have b*:

$$b^* = (E[V^2])^{-1} \frac{1}{\sqrt{n}} \sum_{i \in I} \underbrace{(\widehat{m}_0(X_i) - m_0(X_i))}_{\widehat{m}_0 \text{ estimation error}} \underbrace{(\widehat{g}_0(X_i) - g_0(X_i))}_{\widehat{g}_0 \text{ estimation error}}$$

- Because this term is based on the product of two estimation errors, it vanishes more quickly
- If \hat{g}_0 and \hat{m}_0 are each $n^{\frac{1}{4}}$ -consistent, $\hat{\theta}_0$ will be \sqrt{n} -consistent

 Remember b from the scaled estimation error equation? Now we have b*:

$$b^* = (E[V^2])^{-1} \frac{1}{\sqrt{n}} \sum_{i \in I} \underbrace{(\widehat{m}_0(X_i) - m_0(X_i))}_{\widehat{m}_0 \text{ estimation error}} \underbrace{(\widehat{g}_0(X_i) - g_0(X_i))}_{\widehat{g}_0 \text{ estimation error}}$$

- Because this term is based on the product of two estimation errors, it vanishes more quickly
- If \hat{g}_0 and \hat{m}_0 are each $n^{\frac{1}{4}}$ -consistent, $\hat{\theta}_0$ will be \sqrt{n} -consistent
- To see why, just note that $n^{\frac{1}{4}} \times n^{\frac{1}{4}} = n^{\frac{1}{2}}$

• Chernozhukov et al. use a second partialling-out estimator for partially linear models as well

- Chernozhukov et al. use a second partialling-out estimator for partially linear models as well
- This estimator is very similar to Robinson's partialling-out estimator, which is in turn very similar the Frisch–Waugh–Lovell partialling-out estimator

- Chernozhukov et al. use a second partialling-out estimator for partially linear models as well
- This estimator is very similar to Robinson's partialling-out estimator, which is in turn very similar the Frisch–Waugh–Lovell partialling-out estimator
- If you've taken an introductory statistics course, you've probably learned about the Frisch–Waugh–Lovell theorem

- Chernozhukov et al. use a second partialling-out estimator for partially linear models as well
- This estimator is very similar to Robinson's partialling-out estimator, which is in turn very similar the Frisch–Waugh–Lovell partialling-out estimator
- If you've taken an introductory statistics course, you've probably learned about the Frisch–Waugh–Lovell theorem
- But even if you haven't (or don't remember!), reviewing Frisch-Waugh-Lovell theorem and Robinson can help us build intuition for how DML works

The Frisch-Waugh-Lovell Theorem

The Frisch–Waugh–Lovell Theorem

• Let's say we want to estimate the following model using OLS:

The Frisch–Waugh–Lovell Theorem

• Let's say we want to estimate the following model using OLS:

•
$$Y = \beta_0 + \beta_1 D + \beta_2 X + U$$

The Frisch-Waugh-Lovell Theorem

• Let's say we want to estimate the following model using OLS:

• $Y = \beta_0 + \beta_1 D + \beta_2 X + U$

 The Frisch–Waugh–Lovell Theorem shows us that we can recover the OLS estimate of β₁ using a residuals-on-residuals OLS regression:

The Frisch-Waugh-Lovell Theorem

• Let's say we want to estimate the following model using OLS:

• $Y = \beta_0 + \beta_1 D + \beta_2 X + U$

- The Frisch–Waugh–Lovell Theorem shows us that we can recover the OLS estimate of β₁ using a residuals-on-residuals OLS regression:
 - (1) Regress D on X using OLS
 - Let \widehat{D} be the predicted values of D and let the residuals $\widehat{V} = D \widehat{D}$

The Frisch–Waugh–Lovell Theorem

• Let's say we want to estimate the following model using OLS:

• $Y = \beta_0 + \beta_1 D + \beta_2 X + U$

- The Frisch–Waugh–Lovell Theorem shows us that we can recover the OLS estimate of β₁ using a residuals-on-residuals OLS regression:
 - Regress D on X using OLS
 - Let \widehat{D} be the predicted values of D and let the residuals $\widehat{V} = D \widehat{D}$
 - 2 Regress Y on X using OLS
 - Let \widehat{Y} be the predicted values of Y and let the residuals $\widehat{W} = Y \widehat{Y}$

The Frisch-Waugh-Lovell Theorem

• Let's say we want to estimate the following model using OLS:

• $Y = \beta_0 + \beta_1 D + \beta_2 X + U$

- The Frisch–Waugh–Lovell Theorem shows us that we can recover the OLS estimate of β₁ using a residuals-on-residuals OLS regression:
 - Regress D on X using OLS
 - Let \widehat{D} be the predicted values of D and let the residuals $\widehat{V} = D \widehat{D}$
 - 2 Regress Y on X using OLS
 - Let \widehat{Y} be the predicted values of Y and let the residuals $\widehat{W} = Y \widehat{Y}$
 - 3 Regress \widehat{W} on \widehat{V} using OLS

The Frisch-Waugh-Lovell Theorem

• Let's say we want to estimate the following model using OLS:

• $Y = \beta_0 + \frac{\beta_1 D}{D} + \beta_2 X + U$

- The Frisch–Waugh–Lovell Theorem shows us that we can recover the OLS estimate of β₁ using a residuals-on-residuals OLS regression:
 - Regress D on X using OLS
 - Let \widehat{D} be the predicted values of D and let the residuals $\widehat{V} = D \widehat{D}$
 - 2 Regress Y on X using OLS
 - Let \widehat{Y} be the predicted values of Y and let the residuals $\widehat{W} = Y \widehat{Y}$
 - 3 Regress \widehat{W} on \widehat{V} using OLS
- The estimated coefficient on \hat{V} will be the same as the estimated coefficient $\hat{\beta}_1$ from regressing Y on D and X using OLS!

Robinson

Robinson

• The Frisch–Waugh–Lovell procedure:

Robinson

- The Frisch-Waugh-Lovell procedure:
 - 1 Linear regression of D on X

- The Frisch-Waugh-Lovell procedure:
 - 1 Linear regression of D on X
 - 2 Linear regression of Y on X

- The Frisch-Waugh-Lovell procedure:
 - 1 Linear regression of D on X
 - 2 Linear regression of Y on X
 - 3 Linear regression of the residuals from 2 on the residuals from 1

- The Frisch-Waugh-Lovell procedure:
 - 1 Linear regression of D on X
 - 2 Linear regression of Y on X
 - Linear regression of the residuals from (2) on the residuals from (1)
- Robinson's innovation: let's replace the linear regressions from 1 and 2 with some non-parametric regression

- The Frisch-Waugh-Lovell procedure:
 - 1 Linear regression of D on X
 - 2 Linear regression of Y on X
 - 3 Linear regression of the residuals from 2 on the residuals from 1
- Robinson's innovation: let's replace the linear regressions from 1 and 2 with some non-parametric regression
- Robinson's procedure:

- The Frisch-Waugh-Lovell procedure:
 - 1 Linear regression of D on X
 - 2 Linear regression of Y on X
 - 3 Linear regression of the residuals from 2 on the residuals from 1
- Robinson's innovation: let's replace the linear regressions from 1 and 2 with some non-parametric regression
- Robinson's procedure:
 - 1 Kernel regression of D on X

- The Frisch-Waugh-Lovell procedure:
 - 1 Linear regression of D on X
 - 2 Linear regression of Y on X
 - 3 Linear regression of the residuals from (2) on the residuals from (1)
- Robinson's innovation: let's replace the linear regressions from 1 and 2 with some non-parametric regression
- Robinson's procedure:
 - Kernel regression of D on X
 - 2 Kernel regression of Y on X

- The Frisch-Waugh-Lovell procedure:
 - 1 Linear regression of D on X
 - 2 Linear regression of Y on X
 - 3 Linear regression of the residuals from 2 on the residuals from 1
- Robinson's innovation: let's replace the linear regressions from 1 and 2 with some non-parametric regression
- Robinson's procedure:
 - Kernel regression of D on X
 - 2 Kernel regression of Y on X
 - Linear regression of the residuals from (2) on the residuals from (1)

• DML using residuals-on-residuals regression:

DML using residuals-on-residuals regression:
 ① Estimate D = m
₀(X) + V

DML using residuals-on-residuals regression:

(1) Estimate
$$D = \widehat{m}_0(X) + \widehat{V}$$

Stimate
$$Y = \hat{\ell}_0(X) + \widehat{U}$$

Note the absence of D and the switch from g₀(·) to ℓ₀(·), which is essentially E[Y | X]

- DML using residuals-on-residuals regression:
 - 1 Estimate $D = \widehat{m}_0(X) + \widehat{V}$
 - 2 Estimate $Y = \hat{\ell}_0(X) + \widehat{U}$
 - Note the absence of D and the switch from g₀(·) to ℓ₀(·), which is essentially E[Y | X]
 - 3 Regress \widehat{U} on \widehat{V} using OLS for an estimate $\check{\theta}_0$

• Robinson's procedure:

- Robinson's procedure:
 - 1 Predict *D* with *X* using kernel regression

- Robinson's procedure:
 - Predict D with X using kernel regression
 - 2 Predict Y with X using kernel regression

- Robinson's procedure:
 - Predict D with X using kernel regression
 - Predict Y with X using kernel regression
 - Linear regression of the residuals from (2) on the residuals from (1)
- DML residuals-on-residuals procedure:

- Robinson's procedure:
 - Predict D with X using kernel regression
 - Predict Y with X using kernel regression
 - 3 Linear regression of the residuals from 2 on the residuals from 1
- DML residuals-on-residuals procedure:
 - **1** Predict *D* with *X* using any $n^{\frac{1}{4}}$ -consistent ML model

- Robinson's procedure:
 - Predict D with X using kernel regression
 - Predict Y with X using kernel regression
 - Linear regression of the residuals from (2) on the residuals from (1)
- DML residuals-on-residuals procedure:
 - **1** Predict *D* with *X* using any $n^{\frac{1}{4}}$ -consistent ML model
 - 2 Predict Y with X using any $n^{\frac{1}{4}}$ -consistent ML model

- Robinson's procedure:
 - Predict D with X using kernel regression
 - Predict Y with X using kernel regression
 - 3 Linear regression of the residuals from 2 on the residuals from 1
- DML residuals-on-residuals procedure:
 - **1** Predict *D* with *X* using any $n^{\frac{1}{4}}$ -consistent ML model
 - 2 Predict Y with X using any $n^{\frac{1}{4}}$ -consistent ML model
 - Linear regression of the residuals from (2) on the residuals from (1)

Where We Are and Where We're Going

Where We Are and Where We're Going

• We saw that can eliminate regularization bias using orthogonalization

Where We Are and Where We're Going

- We saw that can eliminate regularization bias using orthogonalization
- Now we're going to show how we can eliminate bias from overfitting using sample-splitting and cross-fitting

• ML algorithms sometimes overfit our data due to their flexibility, and this can lead to bias and slower convergence

- ML algorithms sometimes overfit our data due to their flexibility, and this can lead to bias and slower convergence
- $\bullet\,$ Let's say we estimate $\check{\theta}_0$ using orthogonalization but without sample-splitting

- ML algorithms sometimes overfit our data due to their flexibility, and this can lead to bias and slower convergence
- $\bullet\,$ Let's say we estimate $\check{\theta}_0$ using orthogonalization but without sample-splitting
- That is, we fit our machine learning models and estimate our target parameter $\check{\theta}_0$ on the same set of observations

- ML algorithms sometimes overfit our data due to their flexibility, and this can lead to bias and slower convergence
- $\bullet\,$ Let's say we estimate $\check{\theta}_0$ using orthogonalization but without sample-splitting
- That is, we fit our machine learning models and estimate our target parameter $\check{\theta}_0$ on the same set of observations
- Our scaled estimation error $\sqrt{n}(\check{ heta}_0- heta_0)=a^*+b^*+c^*$

- ML algorithms sometimes overfit our data due to their flexibility, and this can lead to bias and slower convergence
- $\bullet\,$ Let's say we estimate $\check{\theta}_0$ using orthogonalization but without sample-splitting
- That is, we fit our machine learning models and estimate our target parameter $\check{\theta}_0$ on the same set of observations
- Our scaled estimation error $\sqrt{n}(\check{ heta}_0- heta_0)=a^*+b^*+c^*$
- We looked at b^{*} before, and we don't have to worry about a^{*}

• But c^* contains terms like this:

$$\frac{1}{\sqrt{n}}\sum_{i\in I}V_i(\hat{g}_0(X_i)-g_0(X_i))$$

• But *c*^{*} contains terms like this:

$$\frac{1}{\sqrt{n}}\sum_{i\in I}V_i(\hat{g}_0(X_i)-g_0(X_i))$$

• V: the error term (**not** the estimated residuals) from $D = m_0(X) + V$

But c* contains terms like this:

$$\frac{1}{\sqrt{n}}\sum_{i\in I} V_i(\hat{g}_0(X_i) - g_0(X_i))$$

- V: the error term (**not** the estimated residuals) from $D = m_0(X) + V$
- $\hat{g}_0(X_i) g_0(X_i)$: estimation error in \hat{g}_0 from $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \hat{U}$

But c* contains terms like this:

$$\frac{1}{\sqrt{n}}\sum_{i\in I} V_i(\hat{g}_0(X_i) - g_0(X_i))$$

- V: the error term (**not** the estimated residuals) from $D = m_0(X) + V$
- $\hat{g}_0(X_i) g_0(X_i)$: estimation error in \hat{g}_0 from $Y = D\hat{\theta}_0 + \hat{g}_0(X) + \hat{U}$
- What happens if we estimate θ₀ with the same set of observations we used to fit m₀(X) and g₀(X)?

• But c^* contains terms like this:

$$\frac{1}{\sqrt{n}}\sum_{i\in I}V_i(\hat{g}_0(X_i)-g_0(X_i))$$

• But *c*^{*} contains terms like this:

$$\frac{1}{\sqrt{n}}\sum_{i\in I}V_i(\hat{g}_0(X_i)-g_0(X_i))$$

• Overfitting: modeling the noise too closely

• But *c*^{*} contains terms like this:

$$\frac{1}{\sqrt{n}}\sum_{i\in I}V_i(\hat{g}_0(X_i)-g_0(X_i))$$

- Overfitting: modeling the noise too closely
- When \hat{g}_0 is overfit, it will pick up on some of the noise U from the outcome model

Bias from Overfitting

• But c* contains terms like this:

$$\frac{1}{\sqrt{n}}\sum_{i\in I}V_i(\hat{g}_0(X_i)-g_0(X_i))$$

- Overfitting: modeling the noise too closely
- When \hat{g}_0 is overfit, it will pick up on some of the noise U from the outcome model
- If our noise terms V and U are associated, estimation error in \hat{g}_0 from overfitting might be associated with V

Bias from Overfitting

• But c* contains terms like this:

$$\frac{1}{\sqrt{n}}\sum_{i\in I}V_i(\hat{g}_0(X_i)-g_0(X_i))$$

- Overfitting: modeling the noise too closely
- When \hat{g}_0 is overfit, it will pick up on some of the noise U from the outcome model
- If our noise terms V and U are associated, estimation error in \hat{g}_0 from overfitting might be associated with V
- Note: \hat{g}_0 and V might also be associated if we have any unmeasured confounding, but we're assuming that away

• To break the association between V and \hat{g}_0 and avoid bias from overfitting, we employ **sample-splitting**:

- To break the association between V and \hat{g}_0 and avoid bias from overfitting, we employ **sample-splitting**:
 - Randomly partition our data into two subsets

- To break the association between V and \hat{g}_0 and avoid bias from overfitting, we employ **sample-splitting**:
 - Randomly partition our data into two subsets
 - ② Fit your ML models \hat{g}_0 and \hat{m}_0 on the first subset

- To break the association between V and \hat{g}_0 and avoid bias from overfitting, we employ **sample-splitting**:
 - Randomly partition our data into two subsets
 - ② Fit your ML models \hat{g}_0 and \hat{m}_0 on the first subset
 - 3 Estimate $\check{\theta}_0$ in the second subset using the \hat{g}_0 and \hat{m}_0 functions we fit in the first subset

- To break the association between V and \hat{g}_0 and avoid bias from overfitting, we employ **sample-splitting**:
 - Randomly partition our data into two subsets
 - ② Fit your ML models \hat{g}_0 and \hat{m}_0 on the first subset
 - 3 Estimate $\check{\theta}_0$ in the second subset using the \hat{g}_0 and \hat{m}_0 functions we fit in the first subset

• The standard approach to sample-splitting will reduce efficiency and statistical power

- The standard approach to sample-splitting will reduce efficiency and statistical power
- We can avoid the loss of efficiency and power using cross-fitting:

- The standard approach to sample-splitting will reduce efficiency and statistical power
- We can avoid the loss of efficiency and power using cross-fitting:
 - 1 Randomly partition your data into two subsets

- The standard approach to sample-splitting will reduce efficiency and statistical power
- We can avoid the loss of efficiency and power using cross-fitting:
 - Randomly partition your data into two subsets
 - ② Fit two ML models $\hat{g}_{0,1}$ and $\hat{m}_{0,1}$ in the first subset

- The standard approach to sample-splitting will reduce efficiency and statistical power
- We can avoid the loss of efficiency and power using cross-fitting:
 - Randomly partition your data into two subsets
 - 2 Fit two ML models $\hat{g}_{0,1}$ and $\hat{m}_{0,1}$ in the first subset
 - 3 Estimate θ_{0,1} in the second subset using the ĝ_{0,1} and m̂_{0,1} functions we fit in the first subset

- The standard approach to sample-splitting will reduce efficiency and statistical power
- We can avoid the loss of efficiency and power using cross-fitting:
 - Randomly partition your data into two subsets
 - 2 Fit two ML models $\hat{g}_{0,1}$ and $\hat{m}_{0,1}$ in the first subset
 - 3 Estimate $\check{\theta}_{0,1}$ in the second subset using the $\hat{g}_{0,1}$ and $\hat{m}_{0,1}$ functions we fit in the first subset
 - 4 Fit two ML models $\hat{g}_{0,2}$ and $\hat{m}_{0,2}$ in the second subset

- The standard approach to sample-splitting will reduce efficiency and statistical power
- We can avoid the loss of efficiency and power using cross-fitting:
 - Randomly partition your data into two subsets
 - 2) Fit two ML models $\hat{g}_{0,1}$ and $\hat{m}_{0,1}$ in the first subset
 - 3 Estimate $\check{\theta}_{0,1}$ in the second subset using the $\hat{g}_{0,1}$ and $\hat{m}_{0,1}$ functions we fit in the first subset
 - ④ Fit two ML models $\hat{g}_{0,2}$ and $\hat{m}_{0,2}$ in the second subset
 - S Estimate
 θ
 _{0,2} in the first subset using the
 ĝ
 _{0,2} and
 m
 _{0,2} functions we fit in the second subset

- The standard approach to sample-splitting will reduce efficiency and statistical power
- We can avoid the loss of efficiency and power using cross-fitting:
 - Randomly partition your data into two subsets
 - 2) Fit two ML models $\hat{g}_{0,1}$ and $\hat{m}_{0,1}$ in the first subset
 - 3 Estimate $\check{\theta}_{0,1}$ in the second subset using the $\hat{g}_{0,1}$ and $\hat{m}_{0,1}$ functions we fit in the first subset
 - ④ Fit two ML models $\hat{g}_{0,2}$ and $\hat{m}_{0,2}$ in the second subset
 - S Estimate θ_{0,2} in the first subset using the ĝ_{0,2} and m_{0,2} functions we fit in the second subset
 - (6) Average our two estimates $\check{\theta}_{0,1}$ and $\check{\theta}_{0,2}$ for our final estimate $\check{\theta}_0$

Where We Are and Where We're Going

Where We Are and Where We're Going

 We now (hopefully) have a good intuition for how fitting two ML models allows us to remove bias and achieve faster convergence

Where We Are and Where We're Going

- We now (hopefully) have a good intuition for how fitting two ML models allows us to remove bias and achieve faster convergence
- Next we're going to look at how the authors formally define Neyman orthogonality and DML

• Remember: orthogonalize D with respect to $X \rightarrow$ eliminate regularization bias

- Remember: orthogonalize D with respect to $X \rightarrow$ eliminate regularization bias
- Now we formalize this "Neyman orthogonality" condition

- Remember: orthogonalize D with respect to $X \rightarrow$ eliminate regularization bias
- Now we formalize this "Neyman orthogonality" condition

$$\eta_0 = (g_0, m_0), \eta = (g, m)$$

- Remember: orthogonalize D with respect to $X \rightarrow$ eliminate regularization bias
- Now we formalize this "Neyman orthogonality" condition

Let

$$\eta_0 = (g_0, m_0), \eta = (g, m)$$

• We have a new friend! Her name is η_0 ("eta-naught")

- Remember: orthogonalize D with respect to $X \rightarrow$ eliminate regularization bias
- Now we formalize this "Neyman orthogonality" condition

$$\eta_0 = (g_0, m_0), \eta = (g, m)$$

- We have a new friend! Her name is η_0 ("eta-naught")
- η_0 is our *nuisance parameter*

- Remember: orthogonalize D with respect to $X \rightarrow$ eliminate regularization bias
- Now we formalize this "Neyman orthogonality" condition

$$\eta_0 = (g_0, m_0), \eta = (g, m)$$

- We have a new friend! Her name is η_0 ("eta-naught")
- η_0 is our *nuisance parameter*
 - We don't really care what g_0 and m_0 are

- Remember: orthogonalize D with respect to $X \rightarrow$ eliminate regularization bias
- Now we formalize this "Neyman orthogonality" condition

$$\eta_0 = (g_0, m_0), \eta = (g, m)$$

- We have a new friend! Her name is η₀ ("eta-naught")
- η₀ is our *nuisance parameter*
 - We don't really care what g_0 and m_0 are
 - We just want to use them to get good estimates of θ_0

- Remember: orthogonalize D with respect to $X \rightarrow$ eliminate regularization bias
- Now we formalize this "Neyman orthogonality" condition

$$\eta_0 = (g_0, m_0), \eta = (g, m)$$

- We have a new friend! Her name is η₀ ("eta-naught")
- η_0 is our *nuisance parameter*
 - We don't really care what g_0 and m_0 are
 - We just want to use them to get good estimates of θ_0
 - The exact form of our nuisance parameter isn't a scientifically substantive quantity of interest

$$\psi(W;\theta,\eta_0) = \underbrace{(D-m_0(X))}_V \times \underbrace{(Y-g_0(X)-(D-m_0(X))\theta)}_U$$

• Introduce the following *score function*:

$$\psi(W;\theta,\eta_0) = \underbrace{(D-m_0(X))}_V \times \underbrace{(Y-g_0(X)-(D-m_0(X))\theta)}_U$$

Looks scary! Let's break it down

$$\psi(W;\theta,\eta_0) = \underbrace{(D-m_0(X))}_V \times \underbrace{(Y-g_0(X)-(D-m_0(X))\theta)}_U$$

- Looks scary! Let's break it down
- ψ : our score function "psi"

$$\psi(W;\theta,\eta_0) = \underbrace{(D-m_0(X))}_V \times \underbrace{(Y-g_0(X)-(D-m_0(X))\theta)}_U$$

- Looks scary! Let's break it down
- ψ : our score function "psi"
- W: our data

$$\psi(W;\theta,\eta_0) = \underbrace{(D-m_0(X))}_V \times \underbrace{(Y-g_0(X)-(D-m_0(X))\theta)}_U$$

- Looks scary! Let's break it down
- ψ : our score function "psi"
- W: our data
- Each underbraced term represents a noise term from our partially linear model

• Introduce the following *moment condition*:

 $\psi(W; \theta, \eta_0) = (D - m_0(X)) \times (Y - g_0(X) - (D - m_0(X))\theta) = 0$

• Introduce the following *moment condition*:

 $\psi(W; \theta, \eta_0) = (D - m_0(X)) \times (Y - g_0(X) - (D - m_0(X))\theta) = 0$

• So we want our score function to = 0. Why?

• Introduce the following *moment condition*:

 $\psi(W; \theta, \eta_0) = (D - m_0(X)) \times (Y - g_0(X) - (D - m_0(X))\theta) = 0$

• So we want our score function to = 0. Why?

$$Y = V\theta_0 + g_0(X) + U$$
$$Y = (D - m_0(X))\theta_0 + g_0(X) + U$$

• Introduce the following *moment condition*:

 $\psi(W; \theta, \eta_0) = (D - m_0(X)) \times (Y - g_0(X) - (D - m_0(X))\theta) = 0$

• So we want our score function to = 0. Why?

$$Y = V\theta_0 + g_0(X) + U$$
$$Y = (D - m_0(X))\theta_0 + g_0(X) + U$$

• $V = (D - m_0(X))$ is our regressor

• Introduce the following *moment condition*:

 $\psi(W; \theta, \eta_0) = (D - m_0(X)) \times (Y - g_0(X) - (D - m_0(X))\theta) = 0$

• So we want our score function to = 0. Why?

$$Y = V\theta_0 + g_0(X) + U$$
$$Y = (D - m_0(X))\theta_0 + g_0(X) + U$$

• $V = (D - m_0(X))$ is our regressor

• $U = (Y - g_0(X) - (D - m_0(X))\theta)$ is our error term

 We're saying we want our regressor V and our error term U to be orthogonal to one another¹

¹Moment conditions like this will look more familiar to people who know Generalized Method of Moments (GMM). My understanding is that while GMM is popular in economics, it's less common in political science and sociology, which is why I explain what's going on in a little more depth here.

- We're saying we want our regressor V and our error term U to be orthogonal to one another¹
- Two vectors are orthogonal to one another when their dot product equals 0

¹Moment conditions like this will look more familiar to people who know Generalized Method of Moments (GMM). My understanding is that while GMM is popular in economics, it's less common in political science and sociology, which is why I explain what's going on in a little more depth here.

- We're saying we want our regressor V and our error term U to be orthogonal to one another¹
- Two vectors are orthogonal to one another when their dot product equals 0
- This moment condition is very similar to saying we want our error to be uncorrelated with our regressor

¹Moment conditions like this will look more familiar to people who know Generalized Method of Moments (GMM). My understanding is that while GMM is popular in economics, it's less common in political science and sociology, which is why I explain what's going on in a little more depth here.

- We're saying we want our regressor V and our error term U to be orthogonal to one another¹
- Two vectors are orthogonal to one another when their dot product equals 0
- This moment condition is very similar to saying we want our error to be uncorrelated with our regressor
- It is also very similar to (but slightly weaker than) the standard zero conditional mean assumption for OLS

¹Moment conditions like this will look more familiar to people who know Generalized Method of Moments (GMM). My understanding is that while GMM is popular in economics, it's less common in political science and sociology, which is why I explain what's going on in a little more depth here.

• Now we can define Neyman orthogonality!

 $\partial_{\eta}\mathsf{E}[\psi(W;\theta_0,\eta_0)][\eta-\eta_0]=0$

• Now we can define Neyman orthogonality!

$$\partial_{\eta}\mathsf{E}[\psi(W;\theta_0,\eta_0)][\eta-\eta_0]=0$$

• In words: the (Gateaux) derivative of our score function with respect to our nuisance parameter is 0

$$\partial_{\eta}\mathsf{E}[\psi(W;\theta_0,\eta_0)][\eta-\eta_0]=0$$

- In words: the (Gateaux) derivative of our score function with respect to our nuisance parameter is 0
- Recall that a derivative represents our instantaneous rate of change

$$\partial_{\eta}\mathsf{E}[\psi(W;\theta_0,\eta_0)][\eta-\eta_0]=0$$

- In words: the (Gateaux) derivative of our score function with respect to our nuisance parameter is 0
- Recall that a derivative represents our instantaneous rate of change
- Thus, when the derivative = 0, our score function is robust to small perturbations in η_0

$$\partial_{\eta}\mathsf{E}[\psi(W;\theta_0,\eta_0)][\eta-\eta_0]=0$$

- In words: the (Gateaux) derivative of our score function with respect to our nuisance parameter is 0
- Recall that a derivative represents our instantaneous rate of change
- Thus, when the derivative = 0, our score function is robust to small perturbations in η_0
- It doesn't change much when η_0 moves around a little

Now we're ready to formally define DML!

Defining DML

- Take a K-fold random partition (I_k)^K_{k=1} of observation indices [N] = 1, ..., N such that the size of each fold I_k is n = N/K. Also, for each k ∈ [K] = 1, ..., K, define I^c_k := 1, ..., N \ I_k.
 - Create K equally sized partitions
 - I^c_k is the complement of I_k: if we have 100 observations and I_k is the set of observations 1-20, then I^c_k is the set of observations 21-100

Defining DML

② For each
$$k \in [K]$$
, construct an ML estimator

$$\hat{\eta}_{0,\boldsymbol{k}} = \hat{\eta}_0((W_i)_{i \in \boldsymbol{I}_k^c})$$

Important: The estimator we use for fold k was fit in I^c_k!
This is the sample splitting we talked about earlier that removes bias from overfitting

Defining DML

3 Construct the estimator $\tilde{\theta}_0$ ("theta-naught-tilde") as the solution to

$$\frac{1}{K}\sum_{k=1}^{K}E_{n,k}[\psi(W;\tilde{\theta}_{0},\eta_{0,k})]=0$$

- Note that $\tilde{\theta}_0$ is not indexed by k, but the nuisance parameter $\hat{\eta}_{0,k}$ is
- We're finding the $\tilde{\theta}_0$ that minimizes the average of the scores across all folds, where the scores vary by fold due to $\hat{\eta}_{0,k}$
- This is a slightly different² version of the **cross-fitting** approach we talked about earlier that enables us to do sample splitting without loss of efficiency

²In particular, we are no longer taking the average of k different estimates of $\tilde{\theta}_0$ but instead finding one estimate that minimizes the average of the k different score functions. Chernozhukov et al. recommend this latter approach because it behaves better in smaller samples.

• Even under unconfoundedness, OLS does not (necessarily) give us the ATE

- Even under unconfoundedness, OLS does not (necessarily) give us the ATE
- We set up the following moment condition for estimation of the ATE:

$$\psi(W;\theta,\eta) := \\ (g(1,X)-g(0,X)) + \frac{D(Y-g(1,X))}{m(X)} - \frac{(1-D)(Y-g(0,X))}{1-m(X)} - \theta$$

- Even under unconfoundedness, OLS does not (necessarily) give us the ATE
- We set up the following moment condition for estimation of the ATE:

$$\psi(W;\theta,\eta) := (g(1,X)-g(0,X)) + \frac{D(Y-g(1,X))}{m(X)} - \frac{(1-D)(Y-g(0,X))}{1-m(X)} - \theta$$

• Recall the score function has to = 0

- Even under unconfoundedness, OLS does not (necessarily) give us the ATE
- We set up the following moment condition for estimation of the ATE:

$$\psi(W; \theta, \eta) :=$$

 $(g(1, X) - g(0, X)) + \frac{D(Y - g(1, X))}{m(X)} - \frac{(1 - D)(Y - g(0, X))}{1 - m(X)} - \theta$

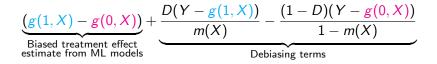
- Recall the score function has to = 0
- So we're saying we want the three big terms in the middle to $= \theta$

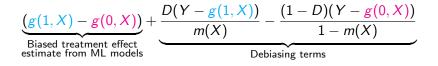
- Even under unconfoundedness, OLS does not (necessarily) give us the ATE
- We set up the following moment condition for estimation of the ATE:

$$\psi(W; \theta, \eta) :=$$

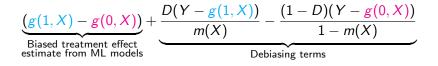
 $(g(1, X) - g(0, X)) + \frac{D(Y - g(1, X))}{m(X)} - \frac{(1 - D)(Y - g(0, X))}{1 - m(X)} - \theta$

- Recall the score function has to = 0
- So we're saying we want the three big terms in the middle to $= \theta$
- Let's look at them more closely

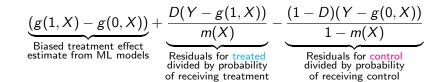


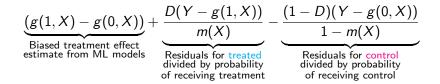


 g(1, X): predicted outcome given X when D = 1, i.e., predicted outcome for treated units

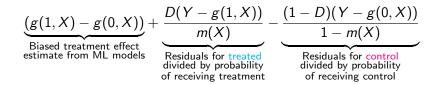


- g(1, X): predicted outcome given X when D = 1, i.e., predicted outcome for treated units
- g(0, X): predicted outcome given X when D = 0, i.e., predicted outcome for control units

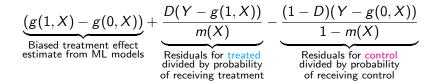




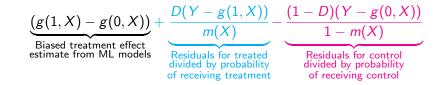
 Recall that we can define the ATE as E[Y(1) - Y(0)] in potential outcomes notation, where Y(1) and Y(0) represent potential outcomes under treatment and control, respectively

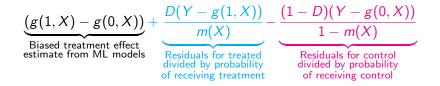


- Recall that we can define the ATE as E[Y(1) Y(0)] in potential outcomes notation, where Y(1) and Y(0) represent potential outcomes under treatment and control, respectively
- When Ŷ(1) is downwardly biased, our ATE will be biased
 downward

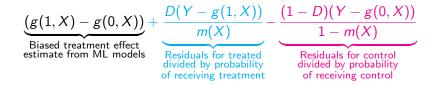


- Recall that we can define the ATE as E[Y(1) Y(0)] in potential outcomes notation, where Y(1) and Y(0) represent potential outcomes under treatment and control, respectively
- When $\widehat{Y}(1)$ is downwardly biased, our \widehat{ATE} will be biased downward
- When $\widehat{Y}(0)$ is downwardly biased, our \widehat{ATE} will be biased **upward**

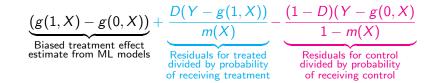


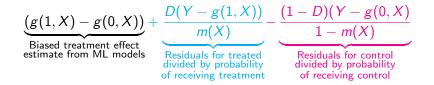


 That's why we want to add the residuals for the treated units and subtract the residuals for the control units

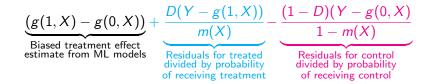


- That's why we want to add the residuals for the treated units and subtract the residuals for the control units
- To see this, note that, e.g., D(Y g(1, X)) represents the observed potential outcome under treatment minus the predicted potential outcome under treatment

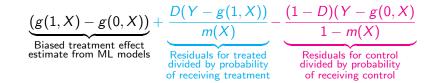


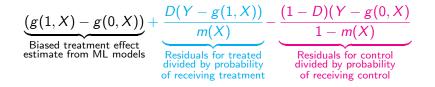


Finally, we weight by the inverse probability of treatment 1/m(X) for the treated units because units with high probability of treatment will be overrepresented among the treated units

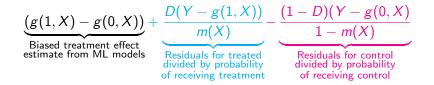


- Finally, we weight by the inverse probability of treatment $\frac{1}{m(X)}$ for the **treated** units because units with high probability of treatment will be overrepresented among the treated units
- And we weight by the inverse probability of control 1/(1-m(X)) for the control units because units with high probability of control will be overrepresented among the control units

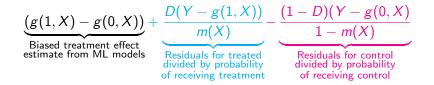




Important: be sure to assess common support!



- Important: be sure to assess common support!
- If the probability of treatment or control is very low in some strata of X, the debiasing terms will blow up



- Important: be sure to assess common support!
- If the probability of treatment or control is very low in some strata of *X*, the debiasing terms will blow up
- $\bullet\,$ Lack of common support $\rightarrow\,$ unstable estimates of treatment effects

 To get valid confidence intervals, we assume that score are linear in the following sense:

 $\psi(w;\theta,\eta) = \psi^{a}(w;\eta)\theta + \psi^{b}(w;\eta) \quad \forall \quad w \in \mathcal{W}, \theta \in \Theta, \eta \in \mathcal{T}$

• To get valid confidence intervals, we assume that score are linear in the following sense:

$$\psi(w;\theta,\eta) = \psi^{a}(w;\eta)\theta + \psi^{b}(w;\eta) \quad \forall \quad w \in \mathcal{W}, \theta \in \Theta, \eta \in \mathcal{T}$$

We use this new term ψ^a(w; η) to estimate the asymptotic variance of our estimator

• We use the following estimator for the asymptotic variance of DML:

• We use the following estimator for the asymptotic variance of DML:

$$\hat{\sigma}^{2} = \underbrace{\hat{J}_{0}^{-1}}_{\hat{J}_{0} \text{ inverse}} \frac{1}{K} \sum_{k=1}^{K} \mathsf{E}_{n,k} \underbrace{[\psi(W; \tilde{\theta}_{0}, \hat{\eta}_{0,k})}_{\text{Score function}} \underbrace{\psi(W; \tilde{\theta}_{0}, \hat{\eta}_{0,k})'}_{\substack{\text{Score function} \\ \text{transpose}}} \underbrace{[\hat{J}_{0}^{-1}]'}_{\hat{J}_{0} \text{ inverse}} \underbrace{[\hat{J}_{0}^{-1}]'}_{\text{transpose}}$$

where

$$\hat{\mathbf{J}}_0 = \frac{1}{K} \sum_{k=1}^{K} \mathsf{E}_{n,k}[\psi^a(\boldsymbol{W}; \hat{\eta}_{0,k})]$$

• Useful for selection-on-observables with high-dimensional confounding

- Useful for selection-on-observables with high-dimensional confounding
- Avoids strong functional form assumptions

- Useful for selection-on-observables with high-dimensional confounding
- Avoids strong functional form assumptions
- Two sources of bias: regularization and overfitting

- Useful for selection-on-observables with high-dimensional confounding
- Avoids strong functional form assumptions
- Two sources of bias: regularization and overfitting
- Two tools for eliminating bias: orthogonalization and cross-fitting

- Useful for selection-on-observables with high-dimensional confounding
- Avoids strong functional form assumptions
- Two sources of bias: regularization and overfitting
- Two tools for eliminating bias: orthogonalization and cross-fitting
- \sqrt{n} consistency if both nuisance parameter estimators are $n^{\frac{1}{4}}\text{-consistent}$

- Useful for selection-on-observables with high-dimensional confounding
- Avoids strong functional form assumptions
- Two sources of bias: regularization and overfitting
- Two tools for eliminating bias: orthogonalization and cross-fitting
- \sqrt{n} consistency if both nuisance parameter estimators are $n^{\frac{1}{4}}$ -consistent
- Asymptotically valid confidence intervals