

ML in Economics and Finance: Where do We Go Now? - Part I

Raul Riva

FGV EPGE

December, 2025

INSPER - São Paulo

Intro

Who is this guy?

- Just joined [FGV EPGE](#) as an Assistant Professor;
- PhD in Finance from [Northwestern University](#);
- Asset Pricing + Macro-Finance + Econometrics;

Who is this guy?

- Just joined [FGV EPGE](#) as an Assistant Professor;
- PhD in Finance from [Northwestern University](#);
- Asset Pricing + Macro-Finance + Econometrics;

I am **not** an ML developer, but maybe a mildly sophisticated consumer

Where are we?

- Last 20-30 years: explosion of computational power and popularization of ML techniques;
- Last 15 years: we economists imported several techniques from CS and Stats;

Where are we?

- Last 20-30 years: explosion of computational power and popularization of ML techniques;
- Last 15 years: we economists imported several techniques from CS and Stats;
- Many challenges in this translation:
 - Causality vs pattern recognition;
 - Interpretability;
 - Sophisticated notions of equilibrium;
 - Time series dynamics;

Where are we?

- Last 20-30 years: explosion of computational power and popularization of ML techniques;
- Last 15 years: we economists imported several techniques from CS and Stats;
- Many challenges in this translation:
 - Causality vs pattern recognition;
 - Interpretability;
 - Sophisticated notions of equilibrium;
 - Time series dynamics;

Right now:

- Better understanding of the limitations of "plug and play" ML;
- Great stuff: new hybrid methods designed by and for economists;
- Bad stuff: we are flooded with tutorials, books, videos, bootcamps...

Where do we go now?

- The Econ/Finance forecasting crowd was really fast in adopting ML...
- But what else? What is worth knowing about ML in Econ and Finance? What's the frontier?

Where do we go now?

- The Econ/Finance forecasting crowd was really fast in adopting ML...
- But what else? What is worth knowing about ML in Econ and Finance? What's the frontier?

What I will do:

- My own economist-crafted definition of ML methods and how to think about them;
- Three very cool agendas where ML can help economists;
- Causality in HD, seriously heterogeneous treatment effects, and solving large models;

Where do we go now?

- The Econ/Finance forecasting crowd was really fast in adopting ML...
- But what else? What is worth knowing about ML in Econ and Finance? What's the frontier?

What I will do:

- My own economist-crafted definition of ML methods and how to think about them;
- Three very cool agendas where ML can help economists;
- Causality in HD, seriously heterogeneous treatment effects, and solving large models;

What I will not do:

- Teach you how to code;

Where do we go now?

- The Econ/Finance forecasting crowd was really fast in adopting ML...
- But what else? What is worth knowing about ML in Econ and Finance? What's the frontier?

What I will do:

- My own economist-crafted definition of ML methods and how to think about them;
- Three very cool agendas where ML can help economists;
- Causality in HD, seriously heterogeneous treatment effects, and solving large models;

What I will not do:

- Teach you how to code;
- Walk you through proofs and be super formal;

Where do we go now?

- The Econ/Finance forecasting crowd was really fast in adopting ML...
- But what else? What is worth knowing about ML in Econ and Finance? What's the frontier?

What I will do:

- My own economist-crafted definition of ML methods and how to think about them;
- Three very cool agendas where ML can help economists;
- Causality in HD, seriously heterogeneous treatment effects, and solving large models;

What I will not do:

- Teach you how to code;
- Walk you through proofs and be super formal;
- Lie to you and say you can easily perform any of this in Stata! 😊

Who is this for?

- Students starting their empirical research agendas;
- Fellow empirical researchers trying to grasp what kind of ML tools might be useful;
- Someone coming from Stats or CS into Economics;

Who is this for?

- Students starting their empirical research agendas;
- Fellow empirical researchers trying to grasp what kind of ML tools might be useful;
- Someone coming from Stats or CS into Economics;

Who is this not for?

- Super sophisticated economists already deploying these techniques everywhere;
- Hardcore econometricians looking for ~~dark magic~~ ultra fancy theorems and proofs;

Who is this for?

- Students starting their empirical research agendas;
- Fellow empirical researchers trying to grasp what kind of ML tools might be useful;
- Someone coming from Stats or CS into Economics;

Who is this not for?

- Super sophisticated economists already deploying these techniques everywhere;
- Hardcore econometricians looking for ~~dark magic~~ ultra fancy theorems and proofs;

DISCLAIMER: These are **my** own views, based on **my** experience, and **my** own readings.
Other people will disagree.



- | | | |
|---|---|----------|
| 1. What is ML, anyway? | } | Today |
| 2. Causality in High Dimensions | | Tomorrow |
| 3. (Seriously) Heterogeneous Partial Effects | } | Tomorrow |
| 4. Solving Large-Scale General Equilibrium Models | | Tomorrow |



- 1. What is ML, anyway?
 - 2. Causality in High Dimensions
 - 3. (Seriously) Heterogeneous Partial Effects
 - 4. Solving Large-Scale General Equilibrium Models
- }
- Today
- }
- Tomorrow

Please bring questions at any time!

Questions?

A General Framework

What is *Machine Learning*?

- Different fields = different definitions: CS, Stats, Operations Research, ...
- Many types: Supervised, Unsupervised, Reinforcement Learning, ...
- More buzzwords = better consulting gigs! 😎

What is *Machine Learning*?

- Different fields = different definitions: CS, Stats, Operations Research, ...
- Many types: Supervised, Unsupervised, Reinforcement Learning, ...
- More buzzwords = better consulting gigs! 😎
- Today and tomorrow: **Supervised Learning**;

What is *Machine Learning*?

- Different fields = different definitions: CS, Stats, Operations Research, ...
- Many types: Supervised, Unsupervised, Reinforcement Learning, ...
- More buzzwords = better consulting gigs! 
- Today and tomorrow: **Supervised Learning**;
- I will be brave enough and provide the one I think is really useful for Economists:

What is *Machine Learning*?

- Different fields = different definitions: CS, Stats, Operations Research, ...
- Many types: Supervised, Unsupervised, Reinforcement Learning, ...
- More buzzwords = better consulting gigs! 😊
- Today and tomorrow: **Supervised Learning**;
- I will be brave enough and provide the one I think is really useful for Economists:

(Supervised) **Machine Learning** is a set of tools that enable computationally-feasible data-driven search over high-dimensional functional spaces.

A General Framework

$$y = f(\mathbf{x}) + \varepsilon$$

- $y \in \mathbb{R}^k$ is some "target" or "outcome";
- $\mathbf{x} \in \mathbb{R}^p$ is a vector of "features", or "predictors", or "covariates";
- $f : \mathbb{R}^p \rightarrow \mathbb{R}^k$ is some unknown function;
- ε is some unobserved noise because the world is messy;

A General Framework

$$y = f(\mathbf{x}) + \varepsilon$$

- $y \in \mathbb{R}^k$ is some "target" or "outcome";
- $\mathbf{x} \in \mathbb{R}^p$ is a vector of "features", or "predictors", or "covariates";
- $f : \mathbb{R}^p \rightarrow \mathbb{R}^k$ is some unknown function;
- ε is some unobserved noise because the world is messy;

Question: given a function space \mathcal{F} , how to find $\hat{f} \in \mathcal{F}$ that approximates f well?

A General Framework

$$y = f(\mathbf{x}) + \varepsilon$$

- $y \in \mathbb{R}^k$ is some "target" or "outcome";
- $\mathbf{x} \in \mathbb{R}^p$ is a vector of "features", or "predictors", or "covariates";
- $f : \mathbb{R}^p \rightarrow \mathbb{R}^k$ is some unknown function;
- ε is some unobserved noise because the world is messy;

Question: given a function space \mathcal{F} , how to find $\hat{f} \in \mathcal{F}$ that approximates f well?

- Collect data $\{(y_1, \mathbf{x}_1), \dots, (y_n, \mathbf{x}_n)\}$;
- Define some notion of "approximates well" \implies (a loss function);
- Be explicit about \mathcal{F} ;
- Be explicit about your optimization mechanism;

You are already doing ML!

Consider an outcome y_i , and a set of covariates \mathbf{x}_i for $i = 1, \dots, n$:

$$y_i = \alpha + \mathbf{x}'_i \boldsymbol{\beta} + \varepsilon_i$$

- This is a linear regression model;
- The function space \mathcal{F} is the set of all affine functions of the treatment and covariates;
- The loss function is the MSE: $\mathcal{L}(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$;
- OLS: minimize a convex loss function over the space of parameters;

You are already doing ML!

Consider an outcome y_i , and a set of covariates \mathbf{x}_i for $i = 1, \dots, n$:

$$y_i = \alpha + \mathbf{x}'_i \boldsymbol{\beta} + \varepsilon_i$$

- This is a linear regression model;
- The function space \mathcal{F} is the set of all affine functions of the treatment and covariates;
- The loss function is the MSE: $\mathcal{L}(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$;
- OLS: minimize a convex loss function over the space of parameters;

Conclusion: Linear regression is a (very simple) ML method! But there is so much more...

Hold on... Isn't Machine Learning just Non-Parametric Estimation?

- The general framework I used could be used in a non-parametric estimation class...
- Why do we need new tools? We already have the good and old kernel regression!

Hold on... Isn't Machine Learning just Non-Parametric Estimation?

- The general framework I used could be used in a non-parametric estimation class...
- Why do we need new tools? We already have the good and old kernel regression!
- Well... there is the curse of dimensionality! If $p \approx 6$, you are already in trouble!

Hold on... Isn't Machine Learning just Non-Parametric Estimation?

- The general framework I used could be used in a non-parametric estimation class...
- Why do we need new tools? We already have the good and old kernel regression!
- Well... there is the curse of dimensionality! If $p \approx 6$, you are already in trouble!

OLS

- Leverages linearity (strong!);
- Easy to compute and interpret;

Fully Non-Parametric Methods

- Extreme flexibility;
- Super data hungry!

Hold on... Isn't Machine Learning just Non-Parametric Estimation?

- The general framework I used could be used in a non-parametric estimation class...
- Why do we need new tools? We already have the good and old kernel regression!
- Well... there is the curse of dimensionality! If $p \approx 6$, you are already in trouble!

OLS

- Leverages linearity (strong!);
- Easy to compute and interpret;

Fully Non-Parametric Methods

- Extreme flexibility;
- Super data hungry!

Machine Learning = a *compromise*: richer parametrizations while still computationally feasible in high dimensions.

Questions?

Causality in High Dimensions

Preliminaries

- No estimator will lead to causality by itself – only careful design will;

Preliminaries

- No estimator will lead to causality by itself – only careful design will;
- ML methods were *not* created to tackle causality problems;
- See Varian (2014), Mullainathan and Spiess (2017), Athey and Imbens (2019), and Masini et al. (2023);

- No estimator will lead to causality by itself – only careful design will;
- ML methods were *not* created to tackle causality problems;
- See Varian (2014), Mullainathan and Spiess (2017), Athey and Imbens (2019), and Masini et al. (2023);
- Kleinberg et al. (2015): many policy-relevant questions are prediction problems!

- No estimator will lead to causality by itself – only careful design will;
- ML methods were *not* created to tackle causality problems;
- See Varian (2014), Mullainathan and Spiess (2017), Athey and Imbens (2019), and Masini et al. (2023);
- Kleinberg et al. (2015): many policy-relevant questions are prediction problems!
- Belloni , Chernozhukov, Hansen and co-authors took it even further:
 - Computing the propensity score *is* forecasting!
 - The first-stage regression in an IV context *is* forecasting!

Treatment Effects in High Dimensions

Suppose you're interested in the treatment effect $\theta_0 \in \mathbb{R}$:

$$y_i = d_i\theta_0 + \mathbf{x}'_i\boldsymbol{\beta} + \varepsilon_i$$

- $y_i \in \mathbb{R}$ is an outcome;
- $d_i \in \mathbb{R}$ is a treatment;
- $\mathbf{x}_i \in \mathbb{R}^p$ is a vector of available covariates;
- ε_i is some unobserved noise with $\mathbb{E}[\varepsilon_i \mid d_i, \mathbf{x}_i] = 0$;

Treatment Effects in High Dimensions

Suppose you're interested in the treatment effect $\theta_0 \in \mathbb{R}$:

$$y_i = d_i\theta_0 + \mathbf{x}'_i \boldsymbol{\beta} + \varepsilon_i$$

- $y_i \in \mathbb{R}$ is an outcome;
- $d_i \in \mathbb{R}$ is a treatment;
- $\mathbf{x}_i \in \mathbb{R}^p$ is a vector of available covariates;
- ε_i is some unobserved noise with $\mathbb{E}[\varepsilon_i | d_i, \mathbf{x}_i] = 0$;
- You have an i.i.d. sample $\{y_i, d_i, \mathbf{x}_i\}_{i=1}^n$ and we allow for $p \gg n$;

Treatment Effects in High Dimensions

Suppose you're interested in the treatment effect $\theta_0 \in \mathbb{R}$:

$$y_i = d_i \theta_0 + \mathbf{x}'_i \boldsymbol{\beta} + \varepsilon_i$$

- $y_i \in \mathbb{R}$ is an outcome;
- $d_i \in \mathbb{R}$ is a treatment;
- $\mathbf{x}_i \in \mathbb{R}^p$ is a vector of available covariates;
- ε_i is some unobserved noise with $\mathbb{E}[\varepsilon_i | d_i, \mathbf{x}_i] = 0$;
- You have an i.i.d. sample $\{y_i, d_i, \mathbf{x}_i\}_{i=1}^n$ and we allow for $p \gg n$;
- **Goal:** estimate θ_0 and get a confidence interval;

Treatment Effects in High Dimensions

Suppose you're interested in the treatment effect $\theta_0 \in \mathbb{R}$:

$$y_i = d_i\theta_0 + \mathbf{x}'_i \boldsymbol{\beta} + \varepsilon_i$$

- $y_i \in \mathbb{R}$ is an outcome;
- $d_i \in \mathbb{R}$ is a treatment;
- $\mathbf{x}_i \in \mathbb{R}^p$ is a vector of available covariates;
- ε_i is some unobserved noise with $\mathbb{E}[\varepsilon_i | d_i, \mathbf{x}_i] = 0$;
- You have an i.i.d. sample $\{y_i, d_i, \mathbf{x}_i\}_{i=1}^n$ and we allow for $p \gg n$;
- **Goal:** estimate θ_0 and get a confidence interval;

Question: what will happen if you try OLS here?

Treatment Effects in High Dimensions

- Let's say you believe only a few β_j 's are $\neq 0 \implies$ "*sparsity*" in β ;
- But you do not know which ones!

Treatment Effects in High Dimensions

- Let's say you believe only a few β_j 's are $\neq 0 \implies$ "*sparsity*" in β ;
- But you do not know which ones!
- What about using your economic intuition to select a subset of controls?

Treatment Effects in High Dimensions

- Let's say you believe only a few β_j 's are $\neq 0 \implies$ "*sparsity*" in β ;
- But you do not know which ones!
- What about using your economic intuition to select a subset of controls?
- Applying Econ theory is always a good idea, but:
 - You might not get a meaningful reduction with theory alone;
 - Your referee might not agree with your choices;
 - You might get lost in a sea of robustness checks...

Treatment Effects in High Dimensions

- Let's say you believe only a few β_j 's are $\neq 0 \implies$ "*sparsity*" in β ;
- But you do not know which ones!
- What about using your economic intuition to select a subset of controls?
- Applying Econ theory is always a good idea, but:
 - You might not get a meaningful reduction with theory alone;
 - Your referee might not agree with your choices;
 - You might get lost in a sea of robustness checks...
- Good news: ML researchers devoted a lot of attention to *sparse regressions*!

Welcome to SBE, Mr. LASSO

The Least Absolute Shrinkage and Selection Operator (LASSO) estimator solves:

$$\hat{\boldsymbol{\delta}} \equiv \arg \min_{\boldsymbol{\delta} \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{w}'_i \boldsymbol{\delta})^2 + \lambda \sum_{j=1}^p |\delta_j| \right\}$$

- $\lambda \geq 0$ is a tuning parameter that controls the amount of penalization (“*regularization*”);
- \mathbf{w}_i is a general vector of regressors of size p ;

Welcome to SBE, Mr. LASSO

The Least Absolute Shrinkage and Selection Operator (LASSO) estimator solves:

$$\hat{\boldsymbol{\delta}} \equiv \arg \min_{\boldsymbol{\delta} \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{w}'_i \boldsymbol{\delta})^2 + \lambda \sum_{j=1}^p |\delta_j| \right\}$$

- $\lambda \geq 0$ is a tuning parameter that controls the amount of penalization (“*regularization*”);
- \mathbf{w}_i is a general vector of regressors of size p ;
- The ℓ_1 penalty $\sum_{j=1}^p |\delta_j|$ induces sparsity in $\hat{\boldsymbol{\delta}}$;

Welcome to SBE, Mr. LASSO

The Least Absolute Shrinkage and Selection Operator (LASSO) estimator solves:

$$\hat{\boldsymbol{\delta}} \equiv \arg \min_{\boldsymbol{\delta} \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{w}'_i \boldsymbol{\delta})^2 + \lambda \sum_{j=1}^p |\delta_j| \right\}$$

- $\lambda \geq 0$ is a tuning parameter that controls the amount of penalization (“*regularization*”);
- \mathbf{w}_i is a general vector of regressors of size p ;
- The ℓ_1 penalty $\sum_{j=1}^p |\delta_j|$ induces sparsity in $\hat{\boldsymbol{\delta}}$;
- If $\lambda = 0$, we get OLS; if $\lambda \rightarrow \infty$, we get $\hat{\boldsymbol{\delta}} = \mathbf{0}$;

Welcome to SBE, Mr. LASSO

The Least Absolute Shrinkage and Selection Operator (LASSO) estimator solves:

$$\hat{\boldsymbol{\delta}} \equiv \arg \min_{\boldsymbol{\delta} \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{w}'_i \boldsymbol{\delta})^2 + \lambda \sum_{j=1}^p |\delta_j| \right\}$$

- $\lambda \geq 0$ is a tuning parameter that controls the amount of penalization (“*regularization*”);
- \mathbf{w}_i is a general vector of regressors of size p ;
- The ℓ_1 penalty $\sum_{j=1}^p |\delta_j|$ induces sparsity in $\hat{\boldsymbol{\delta}}$;
- If $\lambda = 0$, we get OLS; if $\lambda \rightarrow \infty$, we get $\hat{\boldsymbol{\delta}} = \mathbf{0}$;
- For intermediate values of λ , some $\hat{\delta}_j$'s will be exactly zero!

Welcome to SBE, Mr. LASSO

The Least Absolute Shrinkage and Selection Operator (LASSO) estimator solves:

$$\hat{\boldsymbol{\delta}} \equiv \arg \min_{\boldsymbol{\delta} \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{w}'_i \boldsymbol{\delta})^2 + \lambda \sum_{j=1}^p |\delta_j| \right\}$$

- $\lambda \geq 0$ is a tuning parameter that controls the amount of penalization (“*regularization*”);
- \mathbf{w}_i is a general vector of regressors of size p ;
- The ℓ_1 penalty $\sum_{j=1}^p |\delta_j|$ induces sparsity in $\hat{\boldsymbol{\delta}}$;
- If $\lambda = 0$, we get OLS; if $\lambda \rightarrow \infty$, we get $\hat{\boldsymbol{\delta}} = \mathbf{0}$;
- For intermediate values of λ , some $\hat{\delta}_j$'s will be exactly zero!
- $\hat{\boldsymbol{\delta}}$ gives up unbiasedness for much lower variance;
- This problem is still feasible if $p \gg n$ and it is convex \implies fast computation;

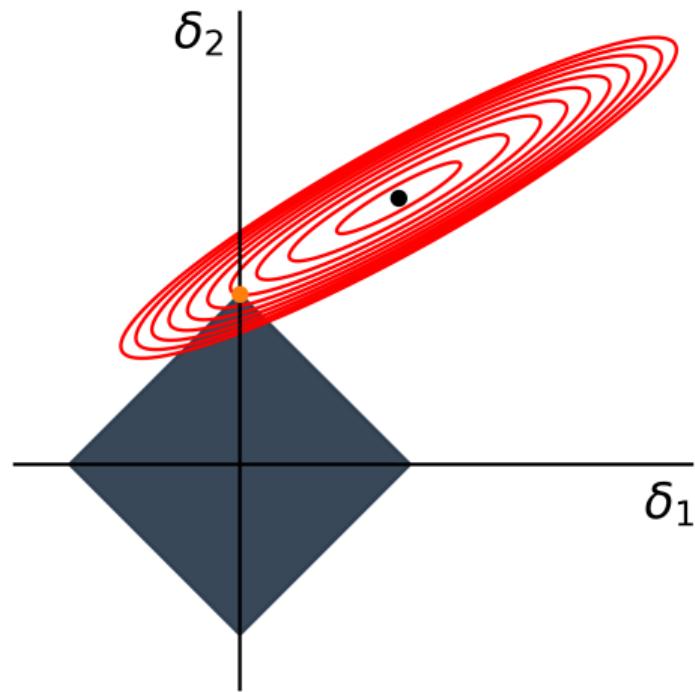
The Geometry of LASSO

For $c > 0$, consider the following:

$$\tilde{\boldsymbol{\delta}} \equiv \arg \min_{\boldsymbol{\delta} \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{w}'_i \boldsymbol{\delta})^2 \right\}$$

$$\text{subject to } \sum_{j=1}^p |\delta_j| \leq c$$

- Think about the Lagrangian of this problem!
- For every λ , there is a c such that $\hat{\boldsymbol{\delta}} = \tilde{\boldsymbol{\delta}}$;



Exploring Options

Recall our treatment effects model:

$$y_i = d_i \theta_0 + \mathbf{x}'_i \boldsymbol{\beta} + \varepsilon_i$$

- **Approach 1:** run LASSO of y_i on d_i and \mathbf{x}_i . Is this a good idea?

Exploring Options

Recall our treatment effects model:

$$y_i = d_i \theta_0 + \mathbf{x}'_i \boldsymbol{\beta} + \varepsilon_i$$

- **Approach 1:** run LASSO of y_i on d_i and \mathbf{x}_i . Is this a good idea?
- $\hat{\theta}_0$ will be severely biased and might even be set to zero!

Exploring Options

Recall our treatment effects model:

$$y_i = d_i \theta_0 + \mathbf{x}'_i \boldsymbol{\beta} + \varepsilon_i$$

- **Approach 1:** run LASSO of y_i on d_i and \mathbf{x}_i . Is this a good idea?
- $\hat{\theta}_0$ will be severely biased and might even be set to zero!
- **Approach 2:** run LASSO but with no penalty on θ_0 . Is this a good idea?

Exploring Options

Recall our treatment effects model:

$$y_i = d_i \theta_0 + \mathbf{x}'_i \boldsymbol{\beta} + \varepsilon_i$$

- **Approach 1:** run LASSO of y_i on d_i and \mathbf{x}_i . Is this a good idea?
• $\hat{\theta}_0$ will be severely biased and might even be set to zero!
- **Approach 2:** run LASSO but with no penalty on θ_0 . Is this a good idea?
• Still terrible! $\hat{\theta}_0$ will still be biased!

Exploring Options

Recall our treatment effects model:

$$y_i = d_i \theta_0 + \mathbf{x}'_i \boldsymbol{\beta} + \varepsilon_i$$

- **Approach 1:** run LASSO of y_i on d_i and \mathbf{x}_i . Is this a good idea?
• $\hat{\theta}_0$ will be severely biased and might even be set to zero!
- **Approach 2:** run LASSO but with no penalty on θ_0 . Is this a good idea?
• Still terrible! $\hat{\theta}_0$ will still be biased!
- **Approach 3:** do Approach 2, but then run OLS on the selected x_i and d_i . Is this good?

Exploring Options

Recall our treatment effects model:

$$y_i = d_i \theta_0 + \mathbf{x}'_i \boldsymbol{\beta} + \varepsilon_i$$

- **Approach 1:** run LASSO of y_i on d_i and \mathbf{x}_i . Is this a good idea?
• $\hat{\theta}_0$ will be severely biased and might even be set to zero!
- **Approach 2:** run LASSO but with no penalty on θ_0 . Is this a good idea?
• Still terrible! $\hat{\theta}_0$ will still be biased!
- **Approach 3:** do Approach 2, but then run OLS on the selected x_i and d_i . Is this good?
• [Leeb and Pötscher \(2008a\)](#) and [Leeb and Pötscher \(2008b\)](#): terrible idea again!
• Main problem: *omitted variable bias* if some relevant controls are not selected!
• If some x_j is correlated with d_i and affects y_i , omitting it biases $\hat{\theta}_0$!

Something That Finally Works!

Belloni et al. (2014a) thought about how d_i and \mathbf{x}_i interact:

$$d_i = \mathbf{x}'_i \gamma + u_i, \quad \mathbb{E}[u_i | \mathbf{x}_i] = 0$$

- What if γ is also sparse, i.e., only a few x_j 's affect d_i ?
- Can we find a small subset of \mathbf{x}_i that *predicts* treatment well?

Something That Finally Works!

Belloni et al. (2014a) thought about how d_i and \mathbf{x}_i interact:

$$d_i = \mathbf{x}'_i \gamma + u_i, \quad \mathbb{E}[u_i | \mathbf{x}_i] = 0$$

- What if γ is also sparse, i.e., only a few x_j 's affect d_i ?
- Can we find a small subset of \mathbf{x}_i that *predicts* treatment well?

They proposed the **Double LASSO** procedure:

1. Run LASSO of y_i on \mathbf{x}_i to select controls \hat{S}_y ;
2. Run LASSO of d_i on \mathbf{x}_i to select controls \hat{S}_d ;
3. Run OLS of y_i on d_i and \mathbf{x}_i with $x_j \in \hat{S}_y \cup \hat{S}_d$;

A Really Cool Result

Belloni et al. (2014b) provide conditions under which:

$$\sqrt{n}(\hat{\theta}_0 - \theta_0) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$$

where σ^2 is complicated but consistently estimated.

A Really Cool Result

Belloni et al. (2014b) provide conditions under which:

$$\sqrt{n}(\hat{\theta}_0 - \theta_0) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$$

where σ^2 is complicated but consistently estimated.

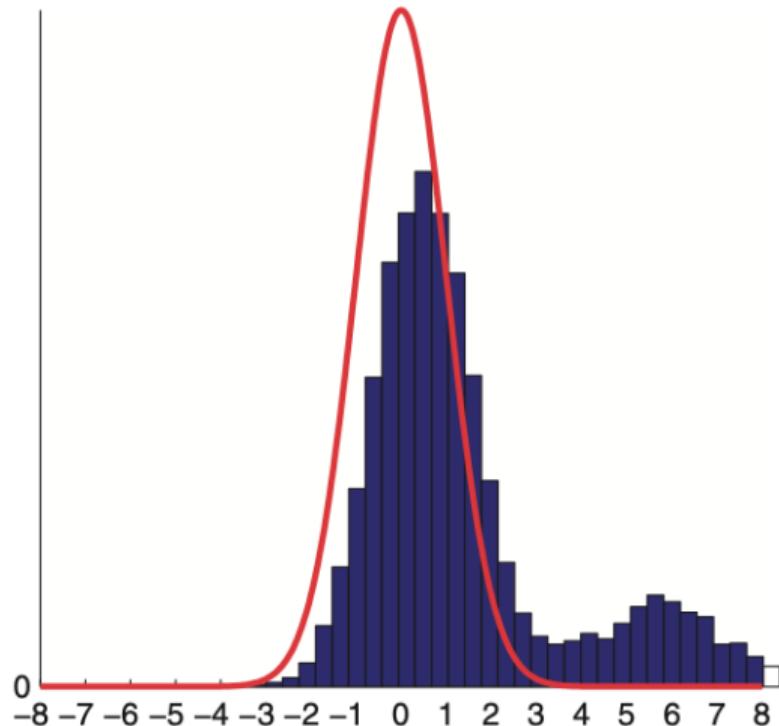
The impressive stuff:

- This convergence is uniform over a large class of DGPs;
- Convergence still happens at the rate \sqrt{n} , even if $p \gg n$!
- Under homoskedasticity, it attains semi-parametric efficiency!
- Construct confidence intervals in the usual ways;

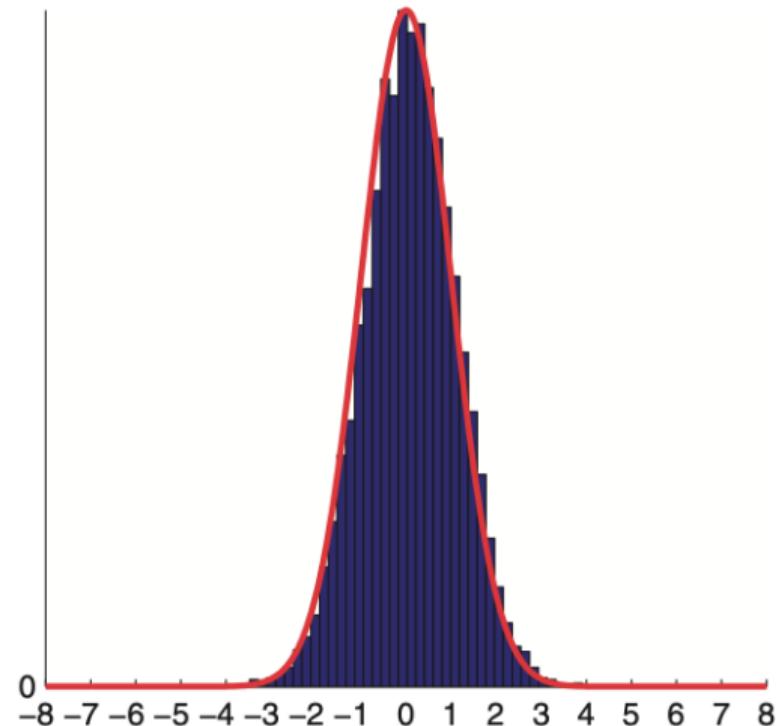
Key assumption: sparse representation;

Some Monte-Carlo Reassurance

post-single-selection estimator



post-double-selection estimator



Questions?

Limitations and Generalizations

Limitations

- What if we want to allow for non-linearities?
- What if we want to use other ML methods?
- What if sparsity is not a good assumption?
- What if treatment has heterogenous effects?
- What if outcomes are function-valued?

Generalization

Belloni et al. (2017) and Chernozhukov et al. (2018) generalize all of this:

$$y_i = g_0(d_i, \mathbf{x}_i) + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i | d_i, \mathbf{x}_i] = 0$$

$$d_i = m_0(\mathbf{x}_i) + u_i, \quad \mathbb{E}[u_i | \mathbf{x}_i] = 0$$

- $g_0(\cdot)$ and $m_0(\cdot)$ are unknown (possibly non-linear) functions;
- You can use several different ML method to estimate $g_0(\cdot)$ and $m_0(\cdot)$;
- Sparsity is not necessary anymore;

Generalization

Belloni et al. (2017) and Chernozhukov et al. (2018) generalize all of this:

$$y_i = g_0(d_i, \mathbf{x}_i) + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i | d_i, \mathbf{x}_i] = 0$$

$$d_i = m_0(\mathbf{x}_i) + u_i, \quad \mathbb{E}[u_i | \mathbf{x}_i] = 0$$

- $g_0(\cdot)$ and $m_0(\cdot)$ are unknown (possibly non-linear) functions;
- You can use several different ML method to estimate $g_0(\cdot)$ and $m_0(\cdot)$;
- Sparsity is not necessary anymore;
- Secret sauce I: Neyman Orthogonal Scores ψ

$$\mathbb{E} \left[\psi(\text{data}, \underbrace{\text{param of interest}}_{\equiv \theta_0}, \underbrace{\text{nuisance params}}_{\equiv \eta_0}) \right] = 0, \quad \frac{\partial}{\partial \eta} \mathbb{E} [\psi(\text{data}, \theta_0, \eta)] \Big|_{\eta=\eta_0} = 0$$

Generalization

Belloni et al. (2017) and Chernozhukov et al. (2018) generalize all of this:

$$y_i = g_0(d_i, \mathbf{x}_i) + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i | d_i, \mathbf{x}_i] = 0$$

$$d_i = m_0(\mathbf{x}_i) + u_i, \quad \mathbb{E}[u_i | \mathbf{x}_i] = 0$$

- $g_0(\cdot)$ and $m_0(\cdot)$ are unknown (possibly non-linear) functions;
- You can use several different ML method to estimate $g_0(\cdot)$ and $m_0(\cdot)$;
- Sparsity is not necessary anymore;
- Secret sauce I: Neyman Orthogonal Scores ψ

$$\mathbb{E} \left[\psi(\text{data}, \underbrace{\text{param of interest}}_{\equiv \theta_0}, \underbrace{\text{nuisance params}}_{\equiv \eta_0}) \right] = 0, \quad \frac{\partial}{\partial \eta} \mathbb{E} [\psi(\text{data}, \theta_0, \eta)] \Big|_{\eta=\eta_0} = 0$$

- Secret sauce II: cross-fitting \implies efficiency vs strict assumptions;

Generalization

Belloni et al. (2017) and Chernozhukov et al. (2018) generalize all of this:

$$y_i = g_0(d_i, \mathbf{x}_i) + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i | d_i, \mathbf{x}_i] = 0$$

$$d_i = m_0(\mathbf{x}_i) + u_i, \quad \mathbb{E}[u_i | \mathbf{x}_i] = 0$$

- $g_0(\cdot)$ and $m_0(\cdot)$ are unknown (possibly non-linear) functions;
- You can use several different ML method to estimate $g_0(\cdot)$ and $m_0(\cdot)$;
- Sparsity is not necessary anymore;
- Secret sauce I: Neyman Orthogonal Scores ψ

$$\mathbb{E} \left[\psi(\text{data}, \underbrace{\text{param of interest}}_{\equiv \theta_0}, \underbrace{\text{nuisance params}}_{\equiv \eta_0}) \right] = 0, \quad \frac{\partial}{\partial \eta} \mathbb{E} [\psi(\text{data}, \theta_0, \eta)] \Big|_{\eta=\eta_0} = 0$$

- Secret sauce II: cross-fitting \implies efficiency vs strict assumptions;
- Independence across i is essential;

A Concrete Example (A Partially Linear Model)

$$y_i = d_i\theta_0 + g_0(\mathbf{x}_i) + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i \mid d_i, \mathbf{x}_i] = 0$$

$$d_i = m_0(\mathbf{x}_i) + u_i, \quad \mathbb{E}[u_i \mid \mathbf{x}_i] = 0$$

A Concrete Example (A Partially Linear Model)

$$y_i = d_i \theta_0 + g_0(\mathbf{x}_i) + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i | d_i, \mathbf{x}_i] = 0$$

$$d_i = m_0(\mathbf{x}_i) + u_i, \quad \mathbb{E}[u_i | \mathbf{x}_i] = 0$$

Steps:

- Divide the data into two folds;
- On fold 1, estimate $\hat{g}_0(\mathbf{x}_i)$ and $\hat{m}_0(\mathbf{x}_i)$ using ML methods;
- On fold 2, compute residuals:

$$\hat{\varepsilon}_i = y_i - \hat{g}_0(\mathbf{x}_i)$$

$$\hat{u}_i = d_i - \hat{m}_0(\mathbf{x}_i)$$

- Regress $\hat{\varepsilon}_i$ on \hat{u}_i to get $\hat{\theta}_0$;

A Concrete Example (A Partially Linear Model)

$$y_i = d_i\theta_0 + g_0(\mathbf{x}_i) + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i | d_i, \mathbf{x}_i] = 0$$

$$d_i = m_0(\mathbf{x}_i) + u_i, \quad \mathbb{E}[u_i | \mathbf{x}_i] = 0$$

Steps:

- Divide the data into two folds;
- On fold 1, estimate $\hat{g}_0(\mathbf{x}_i)$ and $\hat{m}_0(\mathbf{x}_i)$ using ML methods;
- On fold 2, compute residuals:

$$\hat{\varepsilon}_i = y_i - \hat{g}_0(\mathbf{x}_i)$$

$$\hat{u}_i = d_i - \hat{m}_0(\mathbf{x}_i)$$

- Regress $\hat{\varepsilon}_i$ on \hat{u}_i to get $\hat{\theta}_0$;
- Repeat switching folds and average $\hat{\theta}_0$'s;

A Concrete Example (A Partially Linear Model)

$$y_i = d_i \theta_0 + g_0(\mathbf{x}_i) + \varepsilon_i, \quad \mathbb{E}[\varepsilon_i | d_i, \mathbf{x}_i] = 0$$

$$d_i = m_0(\mathbf{x}_i) + u_i, \quad \mathbb{E}[u_i | \mathbf{x}_i] = 0$$

Steps:

- Divide the data into two folds;
- On fold 1, estimate $\hat{g}_0(\mathbf{x}_i)$ and $\hat{m}_0(\mathbf{x}_i)$ using ML methods;
- On fold 2, compute residuals:

$$\hat{\varepsilon}_i = y_i - \hat{g}_0(\mathbf{x}_i)$$

$$\hat{u}_i = d_i - \hat{m}_0(\mathbf{x}_i)$$

- Regress $\hat{\varepsilon}_i$ on \hat{u}_i to get $\hat{\theta}_0$;
- Repeat switching folds and average $\hat{\theta}_0$'s;
- In practice you can use K folds!
- See [Chernozhukov et al. \(2017\)](#) for a practical guide!

Where do we go now?

Some open problems:

- Weak identification, in special in the IV context (see [Scheidegger et al. \(2025\)](#));
- Time series \implies it's impossible to do cross-fitting (see [Lewis and Syrgkanis \(2021\)](#));
- Panel data \implies usual estimators leverage linearity (see [Chernozhukov et al. \(2021\)](#) and [Clarke and Polselli \(2025\)](#));

Where do we go now?

Some open problems:

- Weak identification, in special in the IV context (see Scheidegger et al. (2025));
- Time series \implies it's impossible to do cross-fitting (see Lewis and Syrgkanis (2021));
- Panel data \implies usual estimators leverage linearity (see Chernozhukov et al. (2021) and Clarke and Polselli (2025));

Good news! Plenty of dissertation topics!

Questions?

Thank you!
See you tomorrow, stay tuned!



Appendix and References

References

-  Athey, Susan and Guido W. Imbens (Aug. 2019). "Machine Learning Methods That Economists Should Know About". en. In: *Annual Review of Economics* 11.1, pp. 685–725. ISSN: 1941-1383, 1941-1391. DOI: [10.1146/annurev-economics-080217-053433](https://doi.org/10.1146/annurev-economics-080217-053433). URL: <https://www.annualreviews.org/doi/10.1146/annurev-economics-080217-053433> (visited on 12/03/2025).
-  Belloni, A., V. Chernozhukov, and C. Hansen (Apr. 2014a). "Inference on Treatment Effects after Selection among High-Dimensional Controls". en. In: *The Review of Economic Studies* 81.2, pp. 608–650. ISSN: 0034-6527, 1467-937X. DOI: [10.1093/restud/rdt044](https://doi.org/10.1093/restud/rdt044). URL: <https://academic.oup.com/restud/article-lookup/doi/10.1093/restud/rdt044> (visited on 12/03/2025).
-  Belloni, Alexandre, Victor Chernozhukov, Iván Fernandez-Val, and Christian Hansen (2017). "Program Evaluation and Causal Inference With High-Dimensional Data". en. In: *Econometrica* 85.1, pp. 233–298. ISSN: 0012-9682. DOI: [10.3982/ECTA12723](https://doi.org/10.3982/ECTA12723). URL: <https://www.econometricsociety.org/doi/10.3982/ECTA12723> (visited on 12/04/2025).
-  Belloni, Alexandre, Victor Chernozhukov, and Christian Hansen (May 2014b). "High-Dimensional Methods and Inference on Structural and Treatment Effects". en. In: *Journal of Economic Perspectives* 28.2, pp. 29–50. ISSN: 0895-3309. DOI: [10.1257/jep.28.2.29](https://doi.org/10.1257/jep.28.2.29). URL: <https://pubs.aeaweb.org/doi/10.1257/jep.28.2.29> (visited on 12/03/2025).
-  Chernozhukov, Victor, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, and Whitney Newey (May 2017). "Double/Debiased/Neyman Machine Learning of Treatment Effects". en. In: *American Economic Review* 107.5, pp. 261–265. ISSN: 0002-8282. DOI: [10.1257/aer.p20171038](https://doi.org/10.1257/aer.p20171038). URL: <https://pubs.aeaweb.org/doi/10.1257/aer.p20171038> (visited on 12/02/2025).
-  Chernozhukov, Victor, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins (Feb. 2018). "Double/debiased machine learning for treatment and structural parameters". en. In: *The Econometrics Journal* 21.1, pp. C1–C68. ISSN: 1368-4221, 1368-423X. DOI: [10.1111/ectj.12097](https://doi.org/10.1111/ectj.12097). URL: <https://academic.oup.com/ectj/article/21/1/C1/5056401> (visited on 12/03/2025).
-  Chernozhukov, Victor, Wolfgang Karl Härdle, Chen Huang, and Weining Wang (June 2021). "LASSO-driven inference in time and space". In: *The Annals of Statistics* 49.3. ISSN: 0090-5364. DOI: [10.1214/20-aos2019](https://doi.org/10.1214/20-aos2019). URL: <http://dx.doi.org/10.1214/20-AOS2019>.
-  Clarke, Paul S and Annalivia Polselli (Apr. 2025). "Double machine learning for static panel models with fixed effects". In: *Econometrics Journal*. ISSN: 1368-423X. DOI: [10.1093/ectj/utaf011](https://doi.org/10.1093/ectj/utaf011). URL: <http://dx.doi.org/10.1093/ectj/utaf011>.