

# Treatment-Effects Estimation Using Lasso

Di Liu

Stata

# Table of Contents

1 Motivation

2 Setup

3 `telasso`

4 Double machine learning

5 Postestimation

# Motivation: Estimating **ATE** with many controls

## Example

- We want to estimate the effect of eligibility of a 401(k) on net financial assets (Chernozhukov et al., 2018).
- Conditioning on income and other variables, the access to a 401(k) can be seen as randomly assigned (Poterba and Venti, 1994, Poterba et al. 1995).

## **More vs. Fewer variables**

- On the one hand, we think a simple specification may not be adequate to control for the related confounders. So we need **more** variables or **flexible** models.
- On the other hand, flexible models decrease the power to learn about the treatment effects. So we need **fewer** variables or **simple** models.

```
. webuse assets, clear  
(Excerpt from Chernozhukov and Hansen (2004))
```

```
. describe
```

```
Contains data from https://www.stata-press.com/data/r17/assets.dta
```

```
Observations:      9,913      Excerpt from Chernozhukov and  
                        Hansen (2004)  
Variables:          10      15 Jun 2020 14:15  
                        (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
assets	float	%9.0g		Net total financial assets
age	byte	%9.0g		Age
income	float	%9.0g		Household income
educ	byte	%9.0g		Years of education
pension	byte	%16.0g	lbpen	Pension benefits
married	byte	%11.0g	lbmar	Marital status
twoearn	byte	%9.0g	lbyes	Two-earner household
e401k	byte	%12.0g	lbe401	401(k) eligibility
ira	byte	%9.0g	lbyes	IRA participation
ownhome	byte	%9.0g	lbyes	Homeowner

```
Sorted by: e401k
```

● **outcome:** assets

**treatment:** e401k

## Set controls

```
. //---- orthogonal polynomial ----//  
.   
. orthpoly age, degree(6) generate(_orth_age*)  
. orthpoly income, degree(8) generate(_orth_inc*)  
. orthpoly educ, degree(4) generate(_orth_educ*)  
.   
. //---- define controls -----//  
.   
. global cvars _orth*  
. global fvars pension married twoearn ira ownhome  
. global controls $cvars i.($fvars) c.($cvars)#i.($fvars) ///  
>      i.($fvars)#i.($fvars)
```

- There are **248** controls and **9913** observations.

# Including all the controls?

```
. teffects aipw (assets $controls) (e401k $controls)
Note: tmodel mlogit initial estimates did not converge; the model may not be
      identified
treatment 0 has 2 propensity scores less than 1.00e-05
treatment 1 has 5 propensity scores less than 1.00e-05
treatment overlap assumption has been violated; use option osample() to
identify the overlap violators
r(498);
```

- Including too **many controls** will **violate** the **overlap** assumption!
- In practice, to avoid the conflicts, researchers usually do some sort of model selection, but they **conduct inference** as if there is **no model selection** or assuming the **selected model is correct!**
  - ▶ It's mostly dangerous! Very! (Leeb and Pötscher 2005, 2008)

# Table of Contents

1 Motivation

**2 Setup**

3 `telasso`

4 Double machine learning

5 Postestimation

# ATE and ATET in a potential outcome framework

## Model

$$y = g_0(\tau, \mathbf{x}) + u, \quad \mathbb{E}[u|\mathbf{x}, \tau] = 0$$
$$\tau = m_0(\mathbf{z}) + v, \quad \mathbb{E}[v|\mathbf{z}] = 0$$

where  $y$  is the outcome variable,  $\tau$  is the binary treatment variable,  $\mathbf{x}$  are covariates,  $g_0(\tau, \mathbf{x})$  is the potential outcome, and  $m_0(\mathbf{z})$  is the probability of getting treatment.

## Objective

$$\mathbf{ATE} = \mathbb{E}(g_0(1, \mathbf{x}) - g_0(0, \mathbf{x}))$$
$$\mathbf{ATET} = \mathbb{E}(g_0(1, \mathbf{x}) - g_0(0, \mathbf{x}) | \tau = 1)$$



## Advantages about the model

$$y = g_0(\tau, \mathbf{x}) + u, \quad \mathbb{E}[u|\mathbf{x}, \tau] = 0$$
$$\tau = m_0(\mathbf{z}) + v, \quad \mathbb{E}[v|\mathbf{z}] = 0$$

- The treatment effect is **heterogeneous**, so it varies across observations.
- The treatment effect can be **interactive** with the controls.
- The functions  $g_0(\tau, \mathbf{x})$  and  $m_0(\mathbf{z})$  are **semiparametric**.
  - ▶ We **know the functional form** of  $g_0(\cdot)$  and  $m_0(\cdot)$  (**linear, logit, probit, and poisson**).
  - ▶  $\mathbf{x}$  and  $\mathbf{z}$  can be regarded as a set of basis functions, and we **do not know which terms should go into the model**.

# Conflicts between the CI and overlap assumptions

To identify ATE, we need three key assumptions:

- **Conditional independence:**  $\mathbb{E}(y_\tau | \mathbf{x}, \tau) = \mathbb{E}(y_\tau | \mathbf{x})$ . Dependent on a set of control variables, the potential outcome is independent of the treatment assignment.
- **Overlap:**  $m_0(\mathbf{z}) > 0$ . There is always a positive probability that any given unit is treated or untreated.
- **I.I.D.:** identically independent distributed observations.

## Conflicts

- The more covariates we have, the easier the CI assumption is satisfied.
- Certain specific values of covariates may not be observed in some treatment groups, which means **the violation of the overlap assumption**.

# Honestly solve the conflicts

In practice, to avoid conflicts, researchers usually do some sort of model selection, but they **conduct inference as if there is no model selection or assuming the selected model is correct!**

- It's mostly dangerous! Very! (Leeb and Pötscher 2005, 2008).

## **Model selection and inference**

- We need to **select variables that matter** to outcome and treatment. We do not need them all!
- The inference should **be robust to model-selection mistakes**. We admit that we make the model selection and that we may select wrong variables.

# Table of Contents

1 Motivation

2 Setup

3 `telasso`

4 Double machine learning

5 Postestimation

## Treatment effects + lassos

To estimate ATE, we use the following moment condition in Chernozhukov et al. (2018).

$$ATE = \mathbb{E} \left( g(1, \mathbf{x}) + \frac{\tau (y - g(1, \mathbf{x}))}{m(\mathbf{z})} \right) - \mathbb{E} \left( g(0, \mathbf{x}) + \frac{(1 - \tau) (y - g(0, \mathbf{x}))}{1 - m(\mathbf{z})} \right)$$

- We use **lasso-type techniques to predict**  $g(1, \mathbf{x})$ ,  $g(0, \mathbf{x})$ , and  $m(\mathbf{x})$ .
- It is just a **machine-learning** version of **teffects aipw** (augmented inverse-probability weighting).
- It is **doubly-robust**; i.e., either the outcome or treatment model can be misspecified.
- It is **Neyman orthogonal**; i.e., it is robust to model selection mistakes.

# Intuition

## **Resolve the conflicts between CI and overlap**

- Although the CI assumption expects many variables, we only need the covariates that matter for the outcome.
- If the final selected model is simple or approximately sparse, the overlap assumption is more plausible to be satisfied.

## **Guard against machine-learning mistakes**

- The AIPW moment condition happens to be immune to small machine-learning mistakes.
- In contrast, RA (regression adjustment) and IPW (inverse-probability weighting) are not robust to machine-learning mistakes.

## Example: ATE

```
. telasso (assets $controls) (e401k $controls)
Estimating lasso for outcome assets if e401k = 0 using plugin method ...
Estimating lasso for outcome assets if e401k = 1 using plugin method ...
Estimating lasso for treatment e401k using plugin method ...
Estimating ATE ...
Treatment-effects lasso estimation      Number of observations      =      9,913
Outcome model:  linear                  Number of controls         =      248
Treatment model: logit                  Number of selected controls =      29
```

assets	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE e401k (Eligible vs Not elig..)	8408.417	1259.405	6.68	0.000	5940.029	10876.81
POmean e401k Not eligi..	13958.04	874.6395	15.96	0.000	12243.78	15672.31

- On average, being eligible for a 401(k) will increase financial assets by \$8408.

## Example: ATET

```
. telasso (assets $controls) (e401k $controls), atet
Estimating lasso for outcome assets if e401k = 0 using plugin method ...
Estimating lasso for outcome assets if e401k = 1 using plugin method ...
Estimating lasso for treatment e401k using plugin method ...
Estimating ATET ...
Treatment-effects lasso estimation      Number of observations      =      9,913
Outcome model:  linear                  Number of controls         =      248
Treatment model: logit                  Number of selected controls =      29
```

assets	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATET e401k (Eligible vs Not elig..)	11027.94	1750.394	6.30	0.000	7597.23	14458.65
POmean e401k Not eligi..	19319.45	1402.546	13.77	0.000	16570.51	22068.39

- On average, among the people who are actually eligible for a 401(k), being eligible will increase financial assets by \$11027.



# Example: Control individual lasso

```
. telasso (assets $controls, lasso(0, select(bic)) ) (e401k $controls)
Estimating lasso for outcome assets if e401k = 0 using BIC ...
Estimating lasso for outcome assets if e401k = 1 using plugin method ...
Estimating lasso for treatment e401k using plugin method ...
Estimating ATE ...
```

```
Treatment-effects lasso estimation      Number of observations      =      9,913
Outcome model: linear                   Number of controls         =      248
Treatment model: logit                   Number of selected controls =      44
```

assets	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE e401k (Eligible vs Not elig..)	8206.566	1241.276	6.61	0.000	5773.71	10639.42
POmean e401k Not eligi..	14159.9	859.9154	16.47	0.000	12474.49	15845.3

```
. estimates store bic
```

# Table of Contents

1 Motivation

2 Setup

3 `telasso`

4 Double machine learning

5 Postestimation

# Double machine learning

Double machine learning means cross-fitting + resampling.

## Why do we need them?

- **Cross-fitting** relaxes the requirements in the sparsity assumption.
  - ▶ **Without cross-fitting**, the sparsity assumption requires

$$s_g^2 + s_m^2 \ll N$$

where  $s_g$  and  $s_m$  are the number of actual terms in the outcome and treatment models, respectively.

- ▶ **With cross-fitting**, the sparsity assumption requires

$$s_g * s_m \ll N$$

- **Resampling** reduces the randomness in cross-fitting.

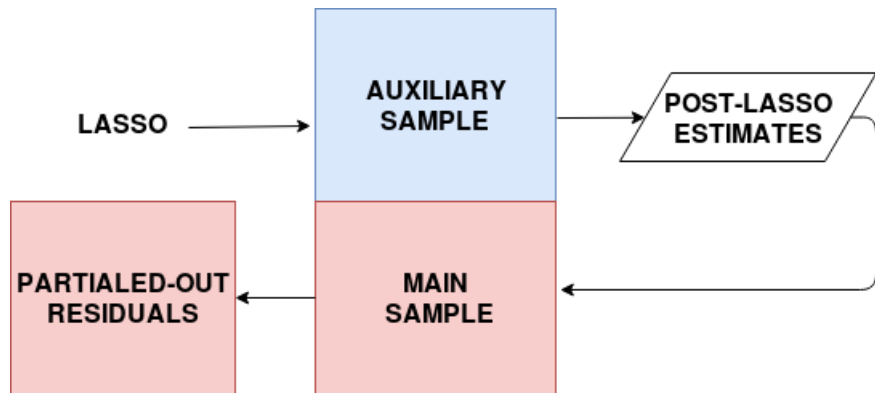
# Basic idea of double machine learning

$$ATE = \mathbb{E} \left( g(1, \mathbf{x}) + \frac{\tau (y - g(1, \mathbf{x}))}{m(\mathbf{z})} \right) - \mathbb{E} \left( g(0, \mathbf{x}) + \frac{(1 - \tau) (y - g(0, \mathbf{x}))}{1 - m(\mathbf{z})} \right)$$

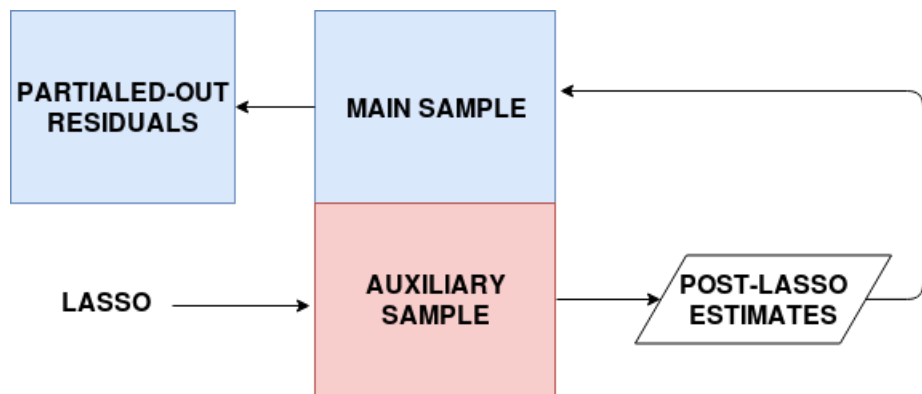
## Basic idea

- 1 Split sample into **auxiliary** part and **main** part
- 2 All the **machine-learning techniques** are applied to the **auxiliary sample**
- 3 All the **post-lasso residuals** are obtained from the **main sample**
- 4 **Switch the role of auxiliary sample and main sample**, and do steps 2 and 3 again
- 5 Solving the moment equation using the full sample

## 2-fold cross-fitting (I)



## 2-fold cross-fitting (II)



## cross-fitting

```
. telasso (assets $controls) (e401k $controls), xfolds(5) rseed(123)
Cross-fit fold 1 of 5 ...
Estimating lasso for outcome assets if e401k = 0 using plugin method ...
Estimating lasso for outcome assets if e401k = 1 using plugin method ...
Estimating lasso for treatment e401k using plugin method ...
... output omitted ...
Treatment-effects lasso estimation      Number of observations      =      9,913
                                         Number of controls         =       248
                                         Number of selected controls =        43
Outcome model:      linear              Number of folds in cross-fit =         5
Treatment model:    logit                Number of resamples        =         1
```

assets	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE e401k (Eligible vs Not elig..)	8244.876	1521.009	5.42	0.000	5263.754	11226
POmean e401k Not eligi..	14271.34	921.0897	15.49	0.000	12466.03	16076.64

- Option `xfold(5)` specifies to use 5-folds cross-fitting. The default is `xfold(10)`.

## cross-fitting + resampling

```
. telasso (assets $controls) (e401k $controls), xfolds(5) resample(3) rseed(123  
> )
```

```
Resample 1 of 3 ...
```

```
Cross-fit fold 1 of 5 ...
```

```
Estimating lasso for outcome assets if e401k = 0 using plugin method ...
```

```
... output omitted ...
```

```
Treatment-effects lasso estimation      Number of observations      =      9,913  
                                         Number of controls         =       248  
                                         Number of selected controls =       47  
Outcome model:      linear              Number of folds in cross-fit =       5  
Treatment model:    logit                Number of resamples        =       3
```

assets	Coefficient	Robust std. err.	z	P> z	[95% conf. interval]	
ATE e401k (Eligible vs Not elig..)	8132.74	1434.918	5.67	0.000	5320.353	10945.13
POmean e401k Not eligi..	14175.17	907.9799	15.61	0.000	12395.56	15954.78

- Option `xfold(5)` specifies to use 5-folds cross-fitting.
- Option `resample(3)` specifies to use 3 resampling.



# Table of Contents

1 Motivation

2 Setup

3 `telasso`

4 Double machine learning

**5 Postestimation**

# Postestimation

The following postestimation commands are of special interest after `telasso`:

Command	Description
<code>teoverlap</code>	overlap plots
<code>tebalance</code>	check balance of covariates
<code>coefpath</code>	plot path of coefficients
<code>cvplot</code>	plot cross-validation function
<code>bicplot</code>	plot BIC function
<code>lassocoeff</code>	display selected coefficients
<code>lassoinfo</code>	display information about lasso estimation results
<code>lassoknots</code>	knot table of coefficient selection and measure of it
<code>lassoselect</code>	select alternative $\lambda^*$

## Refer to a specific lasso result within `telasso`

**Question:** Suppose that we want to use `lassoselect` to modify one of the lasso results within `telasso`. How do we refer to a specific lasso result?

- To refer to the lasso for the **outcome** model with **treatment level = 1**

```
lassoselect id = 4, for(assets) tlevel(1)
```

- To refer to the lasso for the **outcome** model with **treatment level = 0**

```
lassoselect id = 10, for(assets) tlevel(0)
```

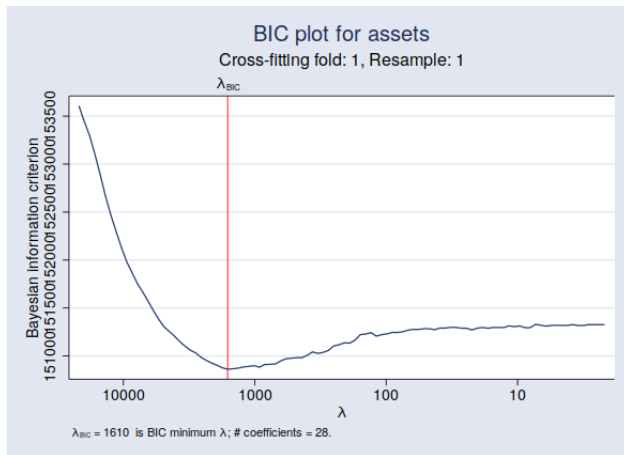
- To refer to the lasso for the **treatment** model

```
lassoselect id = 10, for(e401k)
```

The same philosophy applies to `coefpath`, `cvplot`, `bicplot`, `lassocoeff`, `lassoknots`, and `lassoselect`.

# Sensitivity analysis: bicplot

- . estimates restore bic  
(results bic are active now)
- . bicplot, for(assets) tlevel(0)



# Sensitivity analysis: lassoknots and lassoselect

```
. lassoknots, display(bic nonzero) for(assets) tlevel(0)
```

ID	lambda	No. of nonzero coef.	BIC
2	19843.24	1	153444.4
⋮			
28	1766.475	27	150876.5
28	1766.475	27	150876.5
* 29	1609.546	28	150861.7
30	1466.559	31	150866.8
31	1336.274	34	150872.8
32	1217.563	38	150886
33	1109.398	41	150891.6
⋮			

\* lambda selected by Bayesian information criterion.

```
. lassoselect id = 32, for(assets) tlevel(0)
```

```
ID = 32  lambda = 1217.563 selected
```

# Sensitivity analysis: reestimate

```
. telasso, reestimate
```

```
Estimating lasso for outcome assets if e401k = 0 using BIC ...
```

```
Estimating lasso for outcome assets if e401k = 1 using plugin method ...
```

```
Estimating lasso for treatment e401k using plugin method ...
```

```
Estimating ATE ...
```

```
Treatment-effects lasso estimation      Number of observations      =      9,913
```

```
Outcome model:  linear                  Number of controls         =      248
```

```
Treatment model: logit                  Number of selected controls =      52
```

		Robust				
	assets	Coefficient	std. err.	z	P> z	[95% conf. interval]
ATE						
	e401k					
	(Eligible					
	vs					
	Not elig..)	8291.822	1233.814	6.72	0.000	5873.59 10710.05
POmean						
	e401k					
	Not eligi..	14074.64	852.9615	16.50	0.000	12402.87 15746.41

# Sensitivity analysis: compare results

```
. estimates table . bic, se
```

Variable	Active	bic
ATE e401k (Eligible vs Not elig..)	8291.8222 1233.8144	8206.5656 1241.2759
POmean e401k Not eligi..	14074.639 852.96149	14159.896 859.91542

Legend: b/se

# Summary

- Estimate treatment effects with **high-dimensional controls**
- **Flexible** model specification
  - ▶ Outcome: linear, logit, probit, Poisson
  - ▶ Treatment: logit, probit
- Different measures of treatment effects: **ATE, ATET, POMs**
- Double **robustness** + Neyman **orthogonality**
- Double machine learning: **cross-fitting** and **resampling**



## References

- Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins. 2018. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal* 21(1): C1–C68.
- Leeb, H., and B. M. Pötscher. 2005. Model selection and inference: Facts and fiction. *Econometric Theory* 21(1): 21–59.
- . 2008. Sparse estimators and the oracle property, or the return of Hodges' estimator. *Journal of Econometrics* 142(1): 201–211.
- Poterba, J. M., and S. F. Venti. 1994. 401 (k) plans and tax-deferred saving. In *Studies in the Economics of Aging*, 105–142. University of Chicago Press.
- Poterba, J. M., S. F. Venti, and D. A. Wise. 1995. Do 401 (k) contributions crowd out other personal saving? *Journal of Public Economics* 58(1): 1–32.