Can LASSO improve causal inference?

Can LASSO improve causal inference?

- How might model selection improve causal inference?

Can LASSO improve causal inference?

- How might model selection improve causal inference?
- Thought experiment:

Can LASSO improve causal inference?

- How might model selection improve causal inference?
- Thought experiment:
  - Methods such as matching and regression rely on unconfoundedness

Can LASSO improve causal inference?

- How might model selection improve causal inference?
- Thought experiment:
  - Methods such as matching and regression rely on unconfoundedness
  - If we have high-dimensional data, we can "control for everything"!

Can LASSO improve causal inference?

- How might model selection improve causal inference?
- Thought experiment:
  - Methods such as matching and regression rely on unconfoundedness
  - If we have high-dimensional data, we can "control for everything"!
  - This would give us a high R^2 and remove any omitted variable bias

Can LASSO improve causal inference?

- How might model selection improve causal inference?
- Thought experiment:
  - Methods such as matching and regression rely on unconfoundedness
  - If we have high-dimensional data, we can "control for everything"!
  - This would give us a high R^2 and remove any omitted variable bias
  - LASSO can potentially select only the most important variables

- The problem with the above thought experiment is that LASSO only predicts

- The problem with the above thought experiment is that LASSO only predicts
- If we took a slightly different sample, it might select different variables

- The problem with the above thought experiment is that LASSO only predicts
- If we took a slightly different sample, it might select different variables
- This is because LASSO doesn't care about inference, it cares only about prediction

- The problem with the above thought experiment is that LASSO only predicts
- If we took a slightly different sample, it might select different variables
- This is because LASSO doesn't care about inference, it cares only about prediction
- Mullainathan & Spiess (2017, JEP) illustrate this in their Figure 2

- The problem with the above thought experiment is that LASSO only predicts
- If we took a slightly different sample, it might select different variables
- This is because LASSO doesn't care about inference, it cares only about prediction
- Mullainathan & Spiess (2017, JEP) illustrate this in their Figure 2
- 2 functions with very different coefficients can produce the exact same prediction

- The problem with the above thought experiment is that LASSO only predicts
- If we took a slightly different sample, it might select different variables
- This is because LASSO doesn't care about inference, it cares only about prediction
- Mullainathan & Spiess (2017, JEP) illustrate this in their Figure 2
- 2 functions with very different coefficients can produce the exact same prediction
- To use ML in econometrics, we need to be more principled about ML's role

- In econometrics, we like our estimators to be CAN (Consistent & Asym Normal)

- In econometrics, we like our estimators to be CAN (Consistent & Asym Normal)
- Suppose we want to estimate a treatment effect θ in a high-dimensional model

- In econometrics, we like our estimators to be CAN (Consistent & Asym Normal)
- Suppose we want to estimate a treatment effect θ in a high-dimensional model

$$Y = D \cdot \theta + g(X) + U, \qquad \mathbb{E}[U|X, D] = 0$$

- In econometrics, we like our estimators to be CAN (Consistent & Asym Normal)
- Suppose we want to estimate a treatment effect θ in a high-dimensional model

$$Y = D \cdot \theta + g(X) + U, \qquad \mathbb{E}[U|X, D] = 0$$

- We might want to use LASSO, ridge, random forest, etc. since X is high-dim.

- In econometrics, we like our estimators to be CAN (Consistent & Asym Normal)
- Suppose we want to estimate a treatment effect θ in a high-dimensional model

$$Y = D \cdot \theta + g(X) + U, \qquad \mathbb{E}[U|X, D] = 0$$

- We might want to use LASSO, ridge, random forest, etc. since X is high-dim.
- This solves the bias/variance tradeoff, but introduces regularization bias into $\hat{\theta}$

Econometrics Journal

Econometrics Journal (2018), volume **21**, pp. C1–C68.
doi: 10.1111/ectj.12097

Double/debiased machine learning for treatment and structural parameters

VICTOR CHERNOZHUKOV[†], DENIS CHETVERIKOV[‡], MERT DEMIRER[†], ESTHER DUFLO[†], CHRISTIAN HANSEN[§], WHITNEY NEWEY[†] AND JAMES ROBINS[¶]

[†]*Massachusetts Institute of Technology, 50 Memorial Drive, Cambridge, MA 02139, USA.*
E-mail: vchern@mit.edu, mdemirer@mit.edu, duflo@mit.edu, wnewey@mit.edu

[‡]*University of California Los Angeles, 315 Portola Plaza, Los Angeles, CA 90095, USA.*
E-mail: chetverikov@econ.ucla.edu

[§]*University of Chicago, 5807 S. Woodlawn Ave., Chicago, IL 60637, USA.*
E-mail: chansen1@chicagobooth.edu

[¶]*Harvard University, 677 Huntington Avenue, Boston, MA 02115, USA.*
E-mail: robins@hsph.harvard.edu

First version received: October 2016; final version accepted: June 2017

Summary We revisit the classic semi-parametric problem of inference on a low-dimensional parameter θ_0 in the presence of high-dimensional nuisance parameters η_0. We depart from the classical setting by allowing for η_0 to be so high-dimensional that the traditional assumptions (e.g. Donsker properties) that limit complexity of the parameter space for this object break down. To estimate η_0, we consider the use of statistical or machine learning (ML) methods, which are particularly well suited to estimation in modern, very high-dimensional cases. ML methods perform well by employing regularization to reduce variance and trading off regularization bias with overfitting in practice. However, both regularization bias and overfitting in estimating η_0 cause a heavy bias in estimators of θ_0 that are obtained by naively plugging ML estimators of η_0 into estimating equations for θ_0. This bias results in the naive estimator failing to be $N^{-1/2}$ consistent, where N is the sample size. We show that the impact of regularization bias and overfitting on estimation of the parameter of interest θ_0 can be removed by using two simple, yet critical, ingredients: (1) using Neyman-orthogonal moments/scores that have reduced sensitivity with respect to nuisance parameters to estimate θ_0; (2) making use of cross-fitting, which provides an efficient form of data-splitting. We call the resulting set of methods double or debiased ML (DML). We verify that DML delivers point estimators that concentrate in an $N^{-1/2}$-neighbourhood of the true parameter values and are approximately unbiased and normally distributed, which allows construction of valid confidence statements. The generic statistical theory of DML is elementary and simultaneously relies on only weak theoretical requirements, which will admit the use of a broad array of modern ML methods for estimating the nuisance parameters, such as random forests, lasso, ridge, deep neural nets, boosted trees, and various hybrids and ensembles of these methods. We illustrate the general theory by applying it to provide theoretical properties of the following: DML applied to learn the main regression parameter in a partially linear regression model; DML applied to learn the coefficient on an endogenous variable in a partially linear instrumental variables model; DML applied to learn the average treatment effect and the average treatment effect on the treated under unconfoundedness; DML applied

- How do we solve the regularization bias problem? Add another equation

- How do we solve the regularization bias problem? Add another equation
- Consider outcome and selection equations, respectively

- How do we solve the regularization bias problem? Add another equation

- Consider outcome and selection equations, respectively

$$\begin{aligned} Y &= D \cdot \theta + g(X) + U, & \mathbb{E}[U|X, D] &= 0 \\ D &= m(X) + V, & \mathbb{E}[V|X] &= 0 \end{aligned}$$

- How do we solve the regularization bias problem? Add another equation

- Consider outcome and selection equations, respectively

$$\begin{aligned} Y &= D \cdot \theta + g(X) + U, & \mathbb{E}[U|X, D] &= 0 \\ D &= m(X) + V, & \mathbb{E}[V|X] &= 0 \end{aligned}$$

- We include the second equation to orthogonalize D

- How do we solve the regularization bias problem? Add another equation
- Consider outcome and selection equations, respectively

$$\begin{aligned} Y &= D \cdot \theta + g(X) + U, & \mathbb{E}[U|X, D] &= 0 \\ D &= m(X) + V, & \mathbb{E}[V|X] &= 0 \end{aligned}$$

- We include the second equation to orthogonalize D
- We also need to split our sample to be able to estimate this

- How do we solve the regularization bias problem? Add another equation
- Consider outcome and selection equations, respectively

$$\begin{aligned} Y &= D \cdot \theta + g(X) + U, & \mathbb{E}[U|X, D] &= 0 \\ D &= m(X) + V, & \mathbb{E}[V|X] &= 0 \end{aligned}$$

- We include the second equation to orthogonalize D
- We also need to split our sample to be able to estimate this
- Use ML to estimate both $g(\cdot)$ and $m(\cdot)$ (hence "double" ML)

- How do we solve the regularization bias problem? Add another equation
- Consider outcome and selection equations, respectively

$$\begin{aligned} Y &= D \cdot \theta + g(X) + U, & \mathbb{E}[U|X, D] &= 0 \\ D &= m(X) + V, & \mathbb{E}[V|X] &= 0 \end{aligned}$$

- We include the second equation to orthogonalize D
- We also need to split our sample to be able to estimate this
- Use ML to estimate both $g(\cdot)$ and $m(\cdot)$ (hence "double" ML)
- Instead of using D, we use $\hat{V} = D - \hat{m}(X)$

- How do we solve the regularization bias problem? Add another equation
- Consider outcome and selection equations, respectively

$$\begin{aligned} Y &= D \cdot \theta + g(X) + U, & \mathbb{E}[U|X, D] &= 0 \\ D &= m(X) + V, & \mathbb{E}[V|X] &= 0 \end{aligned}$$

- We include the second equation to orthogonalize D
- We also need to split our sample to be able to estimate this
- Use ML to estimate both $g(\cdot)$ and $m(\cdot)$ (hence "double" ML)
- Instead of using D, we use $\hat{V} = D - \hat{m}(X)$
- This idea is related to the concept of control functions

Steps for Double ML

Steps for Double ML

1. Divide the sample in half (or K folds); call one subsample I^C and the other I

Steps for Double ML

1. Divide the sample in half (or K folds); call one subsample I^C and the other I
2. Estimate $\hat{V} = D - \hat{m}(X)$ in I^C

Steps for Double ML

1. Divide the sample in half (or K folds); call one subsample I^C and the other I
2. Estimate $\hat{V} = D - \hat{m}(X)$ in I^C
3. Estimate $\hat{U} = Y - \hat{g}(X)$ in I^C

Steps for Double ML

1. Divide the sample in half (or K folds); call one subsample I^C and the other I
2. Estimate $\hat{V} = D - \hat{m}(X)$ in I^C
3. Estimate $\hat{U} = Y - \hat{g}(X)$ in I^C
4. Estimate $\check{\theta} = \left(\hat{V}'D\right)^{-1} \hat{V}'\hat{U}$ in I (cf. biased $\hat{\theta} = (D'D)^{-1} D'\hat{U}$)

Steps for Double ML

1. Divide the sample in half (or K folds); call one subsample I^C and the other I
2. Estimate $\hat{V} = D - \hat{m}(X)$ in I^C
3. Estimate $\hat{U} = Y - \hat{g}(X)$ in I^C
4. Estimate $\check{\theta} = \left(\hat{V}'D\right)^{-1} \hat{V}'\hat{U}$ in I (cf. biased $\hat{\theta} = (D'D)^{-1} D'\hat{U}$)
5. Repeat steps 1-3, but switch I^C and I (this is known as cross-fitting)

Steps for Double ML

1. Divide the sample in half (or K folds); call one subsample I^C and the other I
2. Estimate $\hat{V} = D - \hat{m}(X)$ in I^C
3. Estimate $\hat{U} = Y - \hat{g}(X)$ in I^C
4. Estimate $\check{\theta} = \left(\hat{V}'D\right)^{-1} \hat{V}'\hat{U}$ in I (cf. biased $\hat{\theta} = (D'D)^{-1} D'\hat{U}$)
5. Repeat steps 1-3, but switch I^C and I (this is known as cross-fitting)
6. $\check{\theta}_{cf} = \frac{1}{2}\check{\theta}(I^C, I) + \frac{1}{2}\check{\theta}(I, I^C)$

Steps for Double ML

1. Divide the sample in half (or K folds); call one subsample I^C and the other I
2. Estimate $\hat{V} = D - \hat{m}(X)$ in I^C
3. Estimate $\hat{U} = Y - \hat{g}(X)$ in I^C
4. Estimate $\check{\theta} = \left(\hat{V}'D\right)^{-1} \hat{V}'\hat{U}$ in I (cf. biased $\hat{\theta} = (D'D)^{-1} D'\hat{U}$)
5. Repeat steps 1-3, but switch I^C and I (this is known as cross-fitting)
6. $\check{\theta}_{cf} = \frac{1}{2}\check{\theta}(I^C, I) + \frac{1}{2}\check{\theta}(I, I^C)$

Now $\check{\theta}$ will be (approximately) unbiased and asymptotically efficient