Double machine learning

Magnus Nielsen, SODAS, UCPH

Agenda

- Variable selection
- Post-double-selection
- Double machine learning
- Generalizing DML
- Decisions
- Kitchen sink causality
- Causal model selection

Goal

How to use machine learning methods for causality with unknown nuisance functions

- Variable selection
- Double machine learning

How to perform causal model selection

- When estimating heterogeneous treatment effects

Focus on intuitive understanding of methods and workflow

- Will have to use some math to create a scaffold

A citation mess

There's a lot of different papers building on the same idea

- Many have multiple working papers and sometimes also final publications
  - I try to use final publication if possible
  - Can cause confusion when they cite eachother

Victor Chernozhukov is part of many of these papers

- Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). *Double/debiased machine learning for treatment and structural parameters* is probably the most often cited

What to cite?

Recent work on debiased machine learning by Chernozhukov et al. (2018), Chernozhukov, Newey, and Robins (2018), and Chernozhukov, Newey, and Singh (2018) is partly based on and is also generalized by this paper. The construction of orthogonal moments given here was described in Chernozhukov et al. (2018), which cited this paper for that construction and contains no results from this paper. The asymptotic theory in this paper uses the orthogonal moment construction here to improve on the asymptotic theory of Chernozhukov et al. (2018), as described in Section 6. The doubly robust moment conditions considered in Chernozhukov, Newey, and Robins (2018) and Chernozhukov, Newey, and Singh (2018) were derived in the first version of this paper, Chernozhukov, Escanciano, Ichimura, and Newey (2016), and the asymptotic theory in those other papers uses theory given in this paper. The automatic machine learner given here for the additional unknown functions α generalizes that in Chernozhukov, Newey, and Singh (2018). In addition, Newey and Robins (2017) and Hirshberg and Wager (2019) were concerned with linear functions of a regression that are formulated here. Furthermore, Bonhomme and Weidner (2018) have shown the importance of orthogonal moment functions in specification analysis, Foster and Syrgkanis (2019) in deriving rates of convergence for machine learners, Semenova (2018) for machine learning for partially identified models, and Singh and Sun (2019) for machine learning of complier effects

Source: Chernozhukov et al., 2022

Variable selection

Partially linear model

We consider the following partially linear model

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

With $E[U|X,T] = 0$ and $E[V|X] = 0$

Basic model properties:

- Outcome is confounded by (unknown) nuisance function, $g_0(\cdot)$
- People select into treatment based on observables, with (unknown) propensity function $m_0(\cdot)$

Question

How do you perform functional form/variable selection when using OLS?

Model selection

Classic econometrics:

- Use OLS and include covariates based on theory or inference
  - Canonical functional forms, backward selection, forward selection etc.
- Problems:
  - How to delete covariates systematically?
  - Adjust for multiple hypothesis testing?
  - Does data support it?

Guided selection

Machine learning:

- Use LASSO to perform covariate selection

A quick recap of LASSO

Short for least absolute shrinkage and selection operator

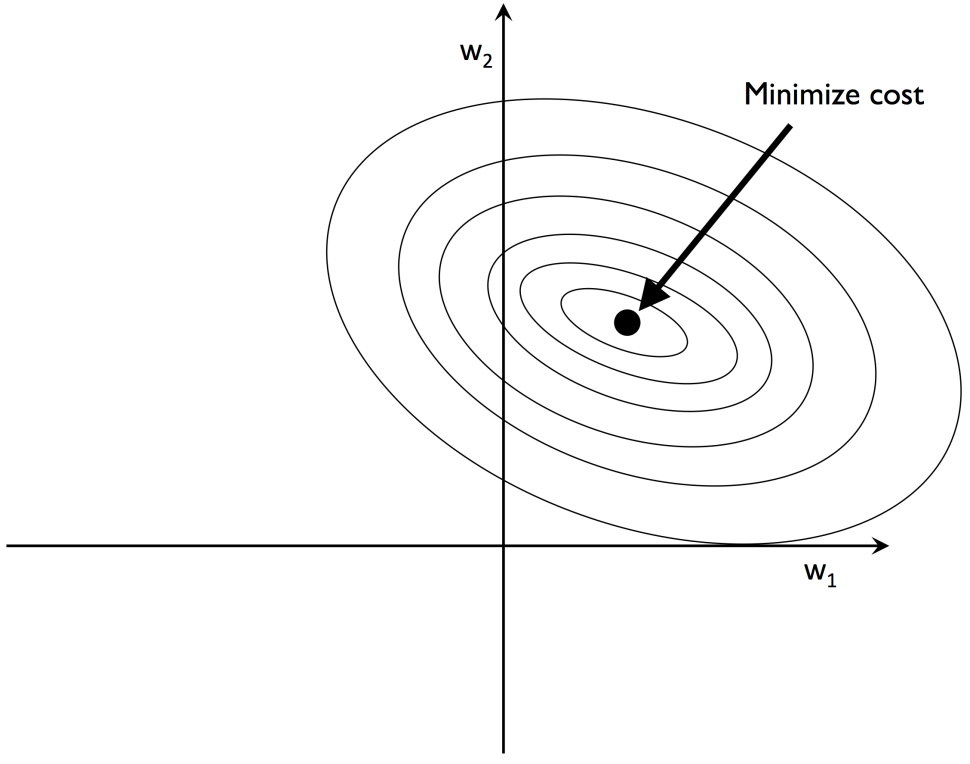
- Introduced by Tibshirani (1996)

We add a term to the minimization problem which penalizes model complexity

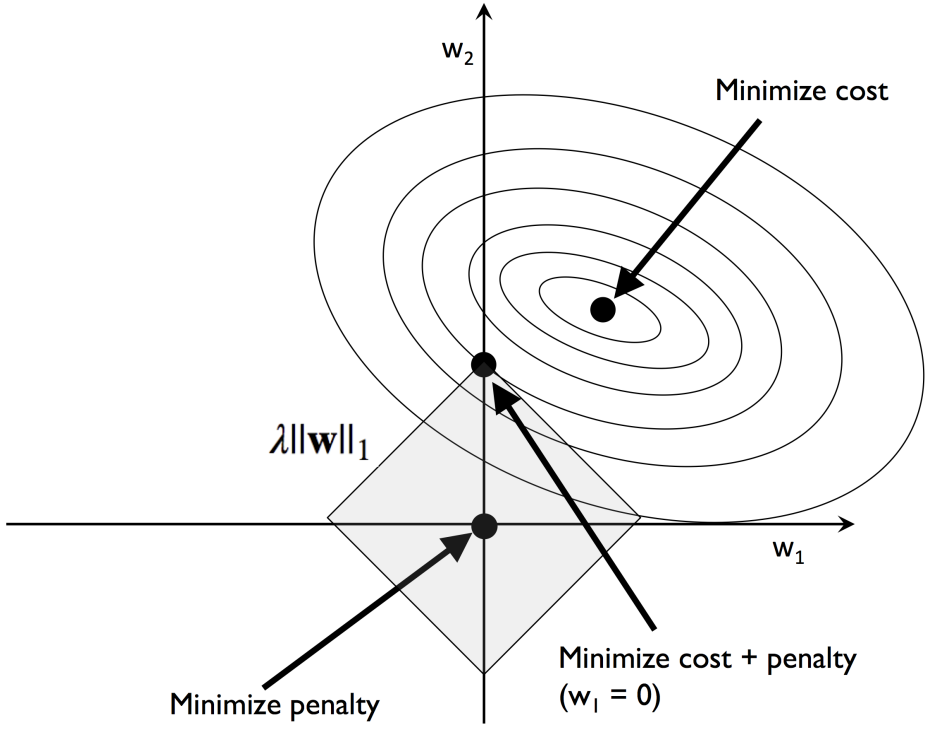
$$\hat{w} = \underset{w}{\operatorname{argmin}} \left\{ \frac{1}{N} ||Y - Xw||_2^2 + \lambda ||w||_1 \right\}, \lambda \geq 0$$

where $|| \cdot ||_1$ is the L1 or Taxicab norm, corresponding to $\sum_{i=1}^{k} |w_i|$

A geometric interpretation



(a) OLS



(b) Lasso

Figure 1: Two-dimensional plots of cost minimization

Source: Raschka & Mirjalili, 2019, ch. 4

A first step

Due to the regularization, all estimates are biased towards zero

- We could exclude the treatment from the regularization

However, some problems remain

Question

What kind of covariates should be included in our regression?

Given this, what's the problem with using the modified LASSO for variable selection?

Confounders

Still problematic

- We may omit potentially relevant variables
  - LASSO excludes possible variables if little predictive power w.r.t. Y
  - Excluded variables may still have predictive power w.r.t. T
    - Confounders with little (but non-zero) predictive power w.r.t. Y may be excluded

Post-double-selection

Fixing the LASSO

A simple solution suggested by Belloni et al. (2014a) is to use a post-double-selection method to correct for bias:

- Step 1: estimate two LASSO models
  - a. Regress Y on X
  - b. Regress T on X
- Step 2: run OLS using only variables that were kept in either LASSO

Also sometimes known as double-LASSO

Why?

This can be motivated in roughly two ways:

- Researcher covariate selection is funky
  - P-hacking, subjectivity etc.
- We are assuming sparsity

Hyperparameter selection

Usually done with cross validation

- Great out of sample predictive performance

Some issues

- Tends to include too many variables
- No formal theory

Theoretical hyperparameters

There also exist theoretically justified hyperparameters

- Good at variable selection
- Formal theory

One issue:

- Not great out of sample predictive performance
  - Hence never used in predictive modelling

Luckily, we're just interested in covariate selection

The formula

$$\lambda = 2.2\sigma_r\sqrt{N}\Phi^{-1}\left(1 - \frac{\alpha}{2K \cdot \ln(N)}\right)$$

Where

- σ_r is the standard deviation of the residuals (homoskedastic)
- Φ^{-1} is the inverse CDF of the normal distribution
- K and N are amount of covariates and sample size
- C and α are constants
  - Usually 1.1 and 0.1

Taken from Urminsky et al. (2016) appendix

- A non-technical introduction to post-double-selection

Residuals?

We must estimate the standard deviation of the residuals

- As such, we need residuals

This is done in an iterative way

- Get starter residuals by subtracting mean
- Estimate standard deviation
- Use formula
  - Repeat x amount of times or until convergence

In Urminsky et al. (2016) $x = 100$

Inference

Formally, we need to assume sparsity to make valid inference (Belloni et al., 2014b)

- This is also what we implicitly do when we make low-dimensional functional forms
- As an example, LASSO could return $K > N$, which won't work with OLS

However, LASSO can handle $K > N$

- We can perform variable selection even if there are more variables than observations!

More variables than observations?

You might find high-dimensional problems irrelevant

- Not so fast

Method can be used for datasets with few observations

- We generally have many controls through Statistics Denmark

Alternatively, you just don’t know the functional form

- But willing to assume linear functional form
- Create polynomial features

Technical regressors

Consider the second, third or higher order polynomial expansion of a regular amount of variables

- Just amount of interactions quickly explodes
  - $\frac{n!}{k!(n-k)!}$ at each level
- Sparsity probably justified
- Sometimes called technical regressors

Could also be other transformations of covariates

Data driven selection

This method only does data driven selection of covariates

- As such, it may leave out variables which you want to include a priori

The formal theory in Belloni et al. (2014b) allows for inclusion of these

- If you want an additional covariate in, just add it to the set

Instrumental variables

The same idea can be applied to instrumental variables

- Introduced in Belloni et al. (2012)

A contender for how to approach many weak instruments problem

- We need to assume sparsity

Implementations

There are a couple of packages

- `hdm` in R
- `pdslasso` in Stata

Hyperparameter selection that is robust to non-Gaussian and heteroskedastic errors exist (Belloni et al., 2012)

Double machine learning

Back to the partially linear model

We're returning to the partially linear model

$$\begin{aligned} Y =& T\theta_0 + g_0(X) + U \\ T =& m_0(X) + V \end{aligned}$$

With $E[U|X,T] = 0$ and $E[V|X] = 0$

We will be going through the main ideas of Chernozhukov et al. (2018)

- The intuitive version

First idea: Sample splitting

We use data splitting to get two samples

- Main part I^m, auxiliary part I^a, each with size n

In the auxiliary sample, I^a, we estimate $g(\cdot)$

- We can use any arbitrary machine learning method
- Learn the confounding function

Estimator

In the main sample we estimate the parameter of interest

$$\hat{\theta}_0 = \frac{\frac{1}{\sqrt{n}} \sum_{i \in I^m} T_i (Y_i - \hat{g}_0(X_i))}{\frac{1}{n} \sum_{i \in I^m} T_i^2}$$

However, this estimator generally does not converge to the true value

- Not usable

Decomposing the error

We can decompose the error into two parts

$$\sqrt{n}(\theta - \hat{\theta}_0) = \frac{\frac{1}{\sqrt{n}} \sum_{i \in I^m} T_i U_i}{\frac{1}{n} \sum_{i \in I^m} T_i^2} + \frac{\frac{1}{\sqrt{n}} \sum_{i \in I^m} T_i (g_0 - \hat{g}_0(X_i))}{\frac{1}{n} \sum_{i \in I^m} T_i^2}$$

The first part converges under mild conditions, the second does not

- What could be the cause of this?
  - Think of what we do when learning $\hat{g}_0$?

Regularization bias

It can be shown that this is due to regularization bias

- We curb overfitting when learning $\hat{g}$
  - This is done to control the bias-variance trade-off
  - Necessary for informative learning in complex and high-dimensional settings
- Estimator will have bias term that asymptotically diverges and is not centered
  - Converges too slowly

Orthogonalization

Suppose we also estimate $\hat{m}_0(\cdot)$ on the auxiliary sample I^a

- Calculate residuals $\hat{V} = T - \hat{m}_0(X)$ for $X \in I^m$

We can then utilize the following estimator

$$\check{\theta}_0 = \frac{\frac{1}{n}\sum_{i \in I^m} \hat{V}_i (y_i - \hat{g}_0(X_i))}{\frac{1}{n}\sum_{i \in I^m} \hat{V}_i T_i}$$

Decomposing

This can be decomposed into three terms, a, b and c

- a is once again well behaved
- b concerns regularization bias
- c concerns overfitting bias

The term b

The term b now depends on the product of the estimation errors in $\hat{m}_0$ and $\hat{g}_0$

$$b = \frac{\frac{1}{\sqrt{n}} \sum_{i \in I^m} [\hat{m}_0(X_i) - m_0(X_i)][\hat{g}_0(X_i) - g_0(X_i)]}{E[V^2]}$$

The moment is Neyman orthogonal

- Locally insensitive to the value of the nuisance parameters
  - We can use noisy estimates
  - Hence why it is sometimes called orthogonal machine learning
- Even though both estimates converge slowly, the product still converges

The term c

The term c relates to overfitting

- Errors in DGP and estimation errors are related
  - Model overfits to the noise
- Traditionally handled in semi-parametric analysis by assuming limited complexity
  - This rules out settings with machine learning methods, as they are too complex

By utilizing sample splitting, errors in DGP and estimation errors are unrelated

- Weak assumptions needed, dependent on application

Sample splitting

Like in the last session, we don't like estimating treatment effects in just one part of the sample

- We rotate the data and repeat the process
  - Also supports K-fold data splitting
    - More data for estimating m_0 and g_0
    - Should be done in practice

Partialling out

There are estimators that take this into account exist

- Also in the setup we've described

We're going to switch to a slightly different estimator based on Robinson (1988)

- Difference between equation 4.3 and 4.4 in Chernozhukov et al. (2018)

Generalizing

Consider the DGP

$$Y = \theta(X) \cdot T + g(X, W) + \epsilon$$
$$T = f(X, W) + \eta$$

where

$$E[\epsilon | X, W] = 0$$
$$E[\eta | X, W] = 0$$
$$E[\epsilon \cdot \eta | X, W] = 0$$

Question: What's the difference between X and W?

Rewriting

We can subtract $E[Y|X,W]$ and get:

$$Y - E[Y|X,W] = \theta(X) \cdot (T - E[T|X,W]) + \epsilon$$

Where we use that

$$E[Y|X,W] = \tau(X) \cdot E[T|X,W] + g(X,W)$$

We can estimate the nuisance functions $E[Y|X,W]$ and $E[T|X,W]$

- Non-parametric prediction problem
- Double machine learning

Compute residuals

We estimate the nuisance functions (using data splitting) and calculate residuals

$$\tilde{Y} = Y - E[Y|X, W]$$
$$\tilde{T} = Y - E[T|X, W]$$

The residuals are related by the equation

$$\tilde{Y} = \theta(X) \cdot \tilde{T} + \epsilon$$

Partialling out estimator

The estimator based on the Robinson's (1988) partialling out approach is then

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} E_n \left[(\tilde{Y} - \theta(X) \cdot \tilde{T})^2 \right]$$

For some model class Θ, i.e. a constant average treatment effect in Chernozhukov et al. (2018)

- Equation 4.4 in Chernozhukov et al. (2018)

Summing up

We perform two steps of machine learning and regress residuals

- Predict T based on X, W
- Predict Y based on X, W
- Calculate residuals
- Regress them on each other, subject to some functional form

Where prediction of T and Y utilize data splitting

In practice

When coding this up, this is what actually happens:

- Find best model and hyperparameters for predicting Y and T
- Give these to an estimator with functional form assumption you want
- (Possibly evaluate model if heterogeneous treatment effects)

Decisions

Worries

You need to worry about

- Predictive performance in first double machine learning stage
- What functional form to assume

And whether there's only selection on observables

- Some methods support IV

Predictive performance

The predictive performance in the two predictive models can be assessed as usual

- Hold out data (same amount of folds as estimator)

Any model you can think of

- Linear/logistic models, including regularized regression
- Tree based models, including random forests & boosted trees
- Neural networks

Or an ensemble of all these

- Good that you learned a lot about prediction!

Tuning beforehand?

You can perform hyperparameter selection using all the data

- Alternatively, perform hyperparameter selection just within auxiliary sample

Better hyperparameter selection results in better nuisance estimates

- More precise causal inference
  - As such, this should be done

Not a problem as long as relatively few hyperparameters are tuned, see references here

Repeated nuisance estimation

You can also increase precision by estimating noise multiple times

- Advocated by Duflo in this presentation, slide 65
- Choose median or mean for final regression
  - Median more robust to outliers

Linear functional forms

ATE in Chernozhukov et al. (2018)

- Also considers instrumental variables

Linear and high-dimensional linear in Semenova et al. (2017)

- High-dimensional linear needs to assume sparsity
  - Utilizes a debiased LASSO from Van de Geer et al. (2014)
  - Could also be used with many technical regressors

Non-parametric functional form

Non-parametric in Athey et al. (2019)

- Also considers instrumental variables
- Relatively low-dimensional data

Non-parametric in Oprescu et al. (2019)

- Can provably handle sparse high-dimensional data

Both use the bootstrap of little bags for inference

Implementations

Implemented in `econml`

- `LinearDML`
- `SparseLinearDML`
- `CausalForestDML`
- `DMLOrthoForest`

Also has other implementations, but `LinearDML`, `SparseLinearDML` and `CausalForestDML` most cited papers

- Proxy for usability?

Some alternative packages

In R: `doubleml`

- Supports clusters
  - See Chiang et al. (2022) for theory
- Chernozhukov on author list
- Also a Python package
  - EconML has $\approx$ 60 times more downloads in a week
    - More estimators & functionality

In Stata: `ddml` & `pystacked`

- `pystacked` uses stacked `sklearn` models
- Same guys behind `pdslasso` and other LASSO implementations in Stata

Doubly robust variants

For categorical treatments, one can also use doubly robust methods

- Predict Y based on both X, W and T, not just X, W
- Otherwise procedure basically the same
  - Last step uses an augmented inverse probability weighted estimator (AIPW)
  - See here for more information

Doubly robust or not?

Pros

- If wrong functional form, get best linear projection
- Slightly stronger robustness guarantees

Cons

- Requires categorical treatment
- Generally higher variance
  - Especially if weak overlap

Kitchen sink causality

What controls to include

Flexible methods

- Both post-double-selection and double machine learning

Tempted to join everything available on observation

- Let the model select important variables
- Keeps all covariates with predictive power

Question

What’s the problem with the aforementioned idea?

Good and bad controls

As always, we should not include ´bad controls´ (Angrist & Pischke, 2009)

- Double machine learning is still sensitive to it, see e.g. Hünermund et al. (2021)

Perform variable selection and double machine learning only with good and neutral controls

Causal graphs

Good and bad controls are relatively vague

Graphs create a way of organizing thoughts about DGP's

- Can be combined with a structural approach
  e.g. Hünermund & Bareinboim (2019)
- Different graphs imply different conditional distributions
  - Can create graph selection procedures, but hidden confounders still a problem

There's a crash course in Cinelli et al. (2020) and a discussion of the use of graph approaches versus potential outcome approaches in Imbens (2020)

Causal model selection

Many CATE estimators

We have many different CATE estimators

- All based on same ‘important’ assumptions
  - Unconfoundedness

Sometimes we might prefer one method a priori

- Linear models are more interpretable

Sometimes we are just interested in personalized estimates

- We want the most accurate model
  - Hence causal model selection

Supervised model selection

Usually we utilize that we observe the ground truth Y for model selection

- Maximize out of sample performance
  - Minimize expected loss, $E[l(\hat{Y}, Y)]$
  - $l(\cdot)$ could be mean squared error, accuracy etc.

The counterpart in causality would be $E[l(\hat{\tau}(x), \tau)]$

- Sadly, we do not observe the ground truth of causality

Question

How could one evaluate CATE's?

Hint: Remember the partialling out regression

$$\tilde{Y} = \theta(X) \cdot \tilde{T} + \epsilon$$

An alternative

Nie & Wager (2021) propose an alternative method to evaluate CATE estimates, $\hat{\tau}$

Rewriting the partialling out regression

$$\tilde{Y} = \theta(X) \cdot \tilde{T} + \epsilon$$
$$\tilde{Y} = \hat{\tau}\tilde{T} + \epsilon$$

Our treatment effects should explain the residual $\tilde{Y}$

- $E[\epsilon|X, W] = 0]$

The loss function

With a slight reformulation of eq. 4 in the paper, we get

$$\hat{L}_n\left[\hat{\tau}(\cdot)\right] = \frac{1}{n}\sum_{i=1}^{n}\left[\tilde{Y} - \hat{\tau}\tilde{T}\right]^2$$

Should use out of sample residuals and CATE estimates

- Split data into bins
- Estimate residuals $\tilde{Y}, \tilde{T}$ in each bin, training on all others
  - Just like double machine learning
- Then evaluate loss function

Alternatively, use held-out data

Causal model selection

Can evaluate estimates from any model

- We now have causal model evaluation
  - This method most consistently selects high-performing model (Schuler et al., 2018)
- Implementations exist
  - How `grf` tunes model hyperparameters with `tune`
  - `econml` has a `score` function

Can also create models

Nie & Wager (2021) also motivate a two step modelling process

- Estimate nuisance functions
- Select parameter estimates that minimizes loss function
  - Estimate $\hat{\tau}$ using a model, i.e. kernel regression

Called the R-learner due to it's close link to Robinson (1988) and the focus on residuals

- `KernelDML` in `econml`
- No confidence intervals

Heterogeneity or not?

One could also evaluate against a constant average treatment effect

- How much better are we at explaining residuals compared to a constant average treatment effect

Implemented in the `RScorer` in `econml`, see here

- If score is negative, heterogeneous treatment effects are worse than constant average treatment effect
  - Possibly overfitting during training

A quick summation

The very short version

- Decide whether to use doubly robust estimator or not
  - Treatment type and how good is overlap
- Create models for T and Y based on X, W (and T if doubly robust)
  - Choose best hyperparameters before DML using cross-validation
- Choose a functional form (an `econml` function)
  - Alternatively, perform causal model selection

Different estimators

`econml` has a table of different estimators

- Concise summary of estimators with different assumptions & treatment types
- Find the papers behind the estimators and read those!

`econml` also has a GitHub with example notebooks

Some additional possibilities

Meta-learners also estimate CATE's

- Cannot deliver confidence intervals
- See Künzel et al. (2019)

Bayesian additive regression trees (BART) also estimate CATE's

- It's Bayesian
- See Chipman et al. (2010)

References

Post-double-selection papers

Belloni, A., Chen, D., Chernozhukov, V., & Hansen, C. (2012). Sparse models and methods for optimal instruments with an application to eminent domain. Econometrica, 80(6), 2369-2429.

Belloni, A., Chernozhukov, V., & Hansen, C. (2014a). High-dimensional methods and inference on structural and treatment effects. Journal of Economic Perspectives, 28(2), 29-50.

Belloni, A., Chernozhukov, V., & Hansen, C. (2014b). Inference on treatment effects after selection among high-dimensional controls. The Review of Economic Studies, 81(2), 608-650.

Urminsky, O., Hansen, C., & Chernozhukov, V. (2016). Using double-lasso regression for principled variable selection. Available at SSRN 2733374.

Double machine learning I

Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters.

Chernozhukov, V., Escanciano, J. C., Ichimura, H., Newey, W. K., & Robins, J. M. (2022). Locally robust semiparametric estimation. Econometrica, 90(4), 1501-1535.

Chiang, H. D., Kato, K., Ma, Y., & Sasaki, Y. (2022). Multiway cluster robust double/debiased machine learning. Journal of Business & Economic Statistics, 40(3), 1046-1056.

Nie, X., & Wager, S. (2021). Quasi-oracle estimation of heterogeneous treatment effects. Biometrika, 108(2), 299-319.

Oprescu, M., Syrgkanis, V., & Wu, Z. S. (2019, May). Orthogonal random forest for causal inference. In International Conference on Machine Learning (pp. 4932-4941). PMLR.

Robinson, P. M. (1988). Root-N-consistent semiparametric regression. Econometrica: Journal of the Econometric Society, 931-954.

Double machine learning II

Schuler, A., Baiocchi, M., Tibshirani, R., & Shah, N. (2018). A comparison of methods for model selection when estimating individual treatment effects. arXiv preprint arXiv:1804.05146.

Semenova, V., Goldman, M., Chernozhukov, V., & Taddy, M. (2017). Estimation and inference on heterogeneous treatment effects in high-dimensional dynamic panels. arXiv preprint arXiv:1712.09988.

Miscellaneous I

Angrist, J. D., & Pischke, J. S. (2009). Mostly harmless econometrics: An empiricist's companion. Princeton university press.

Cinelli, C., Forney, A., & Pearl, J. (2020). A crash course in good and bad controls. Sociological Methods & Research, 0049124121099552.

Chipman, H. A., George, E. I., & McCulloch, R. E. (2010). BART: Bayesian additive regression trees.

Hünermund, P., & Bareinboim, E. (2019). Causal inference and data fusion in econometrics. arXiv preprint arXiv:1912.09104.

Hünermund, P., Louw, B., & Caspi, I. (2021). Double Machine Learning and Automated Confounder Selection–A Cautionary Tale. arXiv preprint arXiv:2108.11294.

Imbens, G. W. (2020). Potential outcome and directed acyclic graph approaches to causality: Relevance for empirical practice in economics. Journal of Economic Literature, 58(4), 1129-1179.

Miscellaneous II

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. Proceedings of the national academy of sciences, 116(10), 4156-4165.

Raschka, S., & Mirjalili, V. (2019). Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. Packt Publishing Ltd.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.

Thanks for attending the course

To the exercises!