Machine Learning for Economists: Part 5 – [Causal] Inference

Michal Andrle
International Monetary Fund

PRELIMINARY

Washington, D.C.,
November, 2019
Disclaimer #1:

The views expressed herein are those of the authors and should not be attributed to the International Monetary Fund, its Executive Board, or its management.
Context

Machine Learning methods are very popular for **prediction**

Increasingly, there are ways of using machine learning for (causal) statistical **inference**

“**Pioneers:**”
V. Chernozhukov, A. Belloni, S. Athey, C. Hansen, L.W. Mackey, V. Syrgkanis, S. Wager, G. Imbens, …
Breiman ponders the state of statistics and sees two cultures... 

- One culture assumes to know the model that supposedly generated the data, tests hypothesis... 
- The other culture uses algorithmic models and treats the data-generating process as unknown... 

Breiman argues that commitment to the first culture: 
“has lead to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems.”
“Just Do It” Approach

Machine-Learning/Computer Science community focuses on solving problems...

Solve the problem first, worry about all the theory later

Would you rather be “roughly right” or “precisely wrong”?

What use is a linear stylized model with well-understood properties if it fails to solve the problem at hand...?

WORSE:
Classical inference gets often abused by un-disciplined specification searches, and undisciplined data mining...
See Leamer (1979): Specification Searches
Machine Learning and [Causal] Inference

Trying not to mix the *statistical* and *causal* issues...

- **Causality** – *why* and *what* is the causal target parameters
  - What is the motivation, what identifies the causal parameters of interest
  - What (not) to condition on (front-door, back-door criteria,...)
  - DAGs, Endogeneity, instruments, diff-in-diff, Neyman-Rubin’s model, SUTVA, ...
  - Most often *untestable* assumptions

- **Statistics** – *how* to estimate the identified target parameters the ‘best way’
  - Parametric or non-parametric, functional forms, ...
  - Model selection, variable selection, especially when \( p \gg N \)
  - Efficient estimation, asymptotic properties, Neyman orthogonality, ...

Once the ‘causal identification’ is done, focus on the statistical details... (causality ≠ inference)
WARNING: Causality – Are you DAGing it?

**DAG** – Directed Acyclical Graph[s] (J. Pearl (2000))

(A) :)

(B) Should you control for $Z$?

No! Bad covariate, inducing bias due to ‘back-door criterion’.
ML Hopes & Issues

Hopes & Benefits:
▶ ML methods are very flexible, allow for nonlinearities, . . .
▶ ML algorithms can deal well with large number of variables
▶ ML is focused on disciplined model selection

Issues & Opportunities:
▶ post-selection inference
▶ little-to-none formal results for popular ML algorithms (asymptotics, efficiency, . . .)
▶ rapid progress on statistical inference with ML

For policy analysis both prediction and inference are important. . .
‘Post-Selection’ Inference...

Classical statistical theory ignores model selection in assessment of estimation accuracy...

Most statistical theory assumes the model selection is not adaptive, not using the data at hand... (which makes learning from data really hard!)

Not accounting for the selection process is bad datamining, and the inference can be overly optimistic!

‘Replicability crisis’ in science...
Ioanidis (2005): “...most published research findings are false.”
ML is about GOOD data mining, a disciplined learning from data.

**Regularization** – adaptive model selection

By knowing the process of model selection, it is easier to account for it in the **inference stage**

It matters a lot **what is the subject of inference** – parameters, or more aggregate ‘effects’ (ATE, etc.)

In some (sub)-models, not all parameters exist!

**What do you do?**

(conditonal coverage?, simultaneous coverage, controlling for family-wise error rate?)
Post-Selection Inference using **Sample Splitting**


“... the idea of splitting a sample in two and then developing the hypothesis on the basis of one part and testing it on the remainder may perhaps be said to be one of the most seriously neglected ideas in statistics”

**Sample Splitting:**

1. Split the IID sample $S$ into $S_A$ and $S_B$
2. Explore and select models using $S_A$
3. Given the selected model, carry out inference using $S_B$

No cheating! ... dates back to Cox (1975), Stone (1974), etc.)
Model Selection using Sample Splitting
Sometimes*, the whole learning process can be bootstrapped!

**Non-Parametric Bootstrap** applied to the whole pipeline of training the model (and parameters) and choosing the model (hyper-parameters, e.g. $\lambda$, $C_p$, lag-length, . . . )

The bootstrap for ‘effects’ (ATE, means, etc.) is more feasible than for the coefficients (e.g. with sparsity not all are always defined. . .)

* conditions & terms apply ;)
Post-Selection Inference & Bootstrap
Bradley Efron (Estimation and Accuracy after Model Selection, JASA, July 1, 2014)

<table>
<thead>
<tr>
<th>Regression model</th>
<th>$m$</th>
<th>$C_p(m) - 80,000$</th>
<th>(Bootstrap %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>2</td>
<td>1132</td>
<td>(19%)</td>
</tr>
<tr>
<td>Quadratic</td>
<td>3</td>
<td>1412</td>
<td>(12%)</td>
</tr>
<tr>
<td>Cubic</td>
<td>4</td>
<td><strong>667</strong></td>
<td>(34%)</td>
</tr>
<tr>
<td>Quartic</td>
<td>5</td>
<td>1591</td>
<td>(8%)</td>
</tr>
<tr>
<td>Quintic</td>
<td>6</td>
<td>1811</td>
<td>(21%)</td>
</tr>
<tr>
<td>Sextic</td>
<td>7</td>
<td>2758</td>
<td>(6%)</td>
</tr>
</tbody>
</table>

**Figure 3:** $B = 4000$ bootstrap replications $\hat{\mu}_1$ of the $C_p$-OLS regression estimate for Subject 1. The original estimate $t(y) = \hat{\mu}_1$ is 2.71, exceeding 76% of the replications. Bootstrap standard deviation (2.4) equals 8.02. Triangles indicate 2.5th and 97.5th percentiles of the histogram.
Post-LASSO Inference

An obvious elephant in the room. . .

Lasso is an adaptive model selection algorithm, aimed at prediction accuracy (fit).

Intuitive BUT WRONG!:

1. Train the model using LASSO with CV-validated $\lambda$ [YES]
2. Estimate post-lasso regressions with selected variables to de-bias the coefficient estimates [YES]
3. Carry out standard statistical inference [NO!]

What’s the way OUT?
Large literature with many special cases. . .

Post-LASSO Inference

Actually, with LASSO, what is the inference about?

(A) Is the inference about the coefficients?
... the trouble is in some models not all coefs exist!

1. Examining everything, \( H_{0,j}^* : \beta_j^* = 0 \), conditional on all variables

2. Inference based on a sub-model, \( \mathcal{M} \)

(B) Or, is the inference about a statistic that is ALWAYS part of the model?

► mean prediction, ...
► average treatment effect (ATE), etc.
► ...

\[\]
A few suggestions...

- **Sample splitting...**

- ‘**In Defense of the Indefensible**’
  
  (Zhao, Shojaie, Witten, 2017) The naive two-step approach shouldn’t work... but ‘can’ yield confidence intervals with asympt. correct coverage, as well as OK p-vals; and there’s reason for that...

- **Various form of bootstrap**
  

- **Exact Post-selection inference** conditioned on the selection event
  
  Lee, Sun, Sun, Taylor (2016)

  When the variable is easily selected, the intervals are essentially the OLS intervals, but when a variable is barely selected, things are bad and intervals very wide...

- **Go Bayesian!**
High-Dimensional Inference & Treatment Effects...

Different focus here – **ONE** parameter of interest, the rest are *nuisance* params...
Non-Technical Introduction

**Parametric Variable Selection & Post-Selection Inference**
Given $K$ variables, $\{X_1, X_2, \ldots, X_k\}$, select ‘optimally’ only $R$ components for a set $\Omega$

$$y_i = \alpha_1 \times \text{Treatment}_i + \sum_{r \in \Omega} \beta_r X_r$$  \hspace{1cm} (1)

**Semi-Parametric Estimation & Variable Selection**

$$y_i = \theta \times \text{Treatment}_i + f(X_i),$$  \hspace{1cm} (2)

with $f(.)$ unknown and $X$ high-dimensional. . .

**Non-Parametric**
Consider a problem

\[ Y = \theta \times D + f(X) + U, \]  

with \( E[U|X, D] = 0. \)

- \( Y \) — outcome variable of interest
- \( D \) — policy or treatment variable
- \( \theta \) — target parameter of interest
- \( X \) — high-dimensional covariates (confounders)
- \( f(.) \) — unknown, complicated function

Confounders \( X \) important because

\[ D = m(X) + V, \quad E[V|X] = 0. \]

How to use modern ML techniques to estimate \( f(.) \) and carry out inference about low-dimensional \( \theta \)?
Rewrite the model in expectations, conditioned on $X$, i.e.

$$E[Y|X] = \theta \times E[D|X] + f(X) + E[U|X], \quad (5)$$

and subtract from the original problem

$$Y = \theta \times D + f(X) + U, \quad (6)$$

to get

$$R_Y = \theta \times R_D + \hat{U}. \quad (7)$$

The estimate of $\theta$ is obtained from regressing $R_Y$ on $R_D$. 

\[ \text{Orthogonal/Double Machine Learning} \]
Orthogonal/Double Machine Learning

The regression of $R_Y$ on $R_D$ is *infeasible*...

Use modern **machine-learning tools** (random forests, neural nets, boosting, Lasso, ...) to learn mappings functions $h(.)$ and $m(.)$

$\hat{R}_Y \equiv (Y - E[Y|X]) \equiv Y - \hat{h}(X)$

$\hat{R}_D \equiv (Y - E[D|X]) \equiv Y - \hat{m}(X)$

Can we estimate $\theta$ as

$$\hat{\theta} = (\hat{D}'\hat{R}_D)^{-1}\hat{R}'\hat{R}_Y? \quad (7)$$

**The issue** is that $\hat{h}(.)$ and $\hat{m}(.)$ were obtained using the whole sample, risking over-fitting and complicating inference...
Sample Splitting & Cross-Fitting

Sample splitting is one of the most **understated** methods in statistics...

1. Split the sample in two parts, $S_1$ and $S_2$...
2. Use $S_1$ to train and select the models
3. Use $S_2$ to carry out inference using the models

Sample splitting decreases sample size and thus **lowers** power.

**Cross-fitting:** Do multiple splits and average the estimates...

Of course, related to cross-validation...
Given the model

\[ Y = \theta \times D + f(X) + U, \quad (8) \]

1. Randomly split the sample in two parts, \( S_1 \) and \( S_2 \).
2. Using the sample \( S_1 \), estimate \( E[Y|X] = \hat{h}(X) \) and \( E[D|X] = \hat{m}(X) \).
3. Using the sample \( S_2 \) compute projection errors \( \hat{R}_Y = Y - \hat{h}(X) \) and \( \hat{R}_D = D - \hat{m}(X) \).
4. Compute the estimate \( \hat{\theta}_1 \) by regressing \( \hat{R}_Y \) on \( \hat{R}_D \).
5. Flip the roles of samples \( S_1 \) and \( S_2 \), to estimate \( \hat{\theta}_2 \).
6. Estimate \( \hat{\theta} = \frac{1}{2} \hat{\theta}_1 + \frac{1}{2} \hat{\theta}_2 \). Now, \( \sqrt{N}(\hat{\theta} - \theta) \sim N(0, \Sigma) \).
Orthogonal ML – Monte Carlo Simulation

Inspired by the blog by Gabriel Vasconselos...

Data-Generating Process to test:

\[ y_i = D_i \theta + \cos^2(x'_i \gamma) + u_i \] \hspace{1cm} (9)

\[ D_i = \sin(x'_i \gamma) + \cos(x'_i \gamma) + v_i \] \hspace{1cm} (10)

with \( u, v \sim N(0, 1) \), \( \theta = 0.5 \) and \( \gamma_k = 1/k \).

Comparing three estimators:

- OLS
- ‘Naive’ ML, see Chernozhukov et al (2016) for details
- Double-ML with cross-fitting, using Random Forests for regression

Note: for the OLS estimation the “safer” benchmark is \( y \sim d + x \)
Orthogonal ML – Monte Carlo Simulation

Graph showing the frequency distribution of different estimators:
- Blue line: Double ML + cross fitting
- Green line: OLS
- Red line: ‘Naive’ ML

The x-axis represents \( \hat{\theta} \) values ranging from -0.25 to 1.50, and the y-axis represents the frequency ranging from 0.0 to 2.5.
Orthogonal ML – Examples
Orthogonal ML – Why Does It Work?

Magic...;

A few of key ingredients (intuitively):

- The implied moment conditions (and the influence function) are not “too” sensitive to “small” mistakes in estimating nuisance functions $h(.)$ and $m(.)$

- Not every moment condition necessarily satisfies this ‘Neyman Orthogonality’

- Sample splitting further helps to lower assumptions needed about smoothness of $h(.)$ and $m(.)$

- The FWL “residualization” is essentially the same as in standard semi-parametric literature. BUT the inference works for quite different reasons with ML plugin estimators, which have more complex properties...
Double ML – General version
Consider having sample size $N$ for

$$Y_i = \theta \times D_i + X_i'\alpha + \varepsilon_i$$  \hspace{1cm} (11)$$

where

$Y_i$ – outcome  
$\theta$ – [causal] coefficient of interest  
$D_i$ – treatment, exogenous variable  
$X_i$ – “large” vector of covariates (confounders), $P \gg N$  
$\alpha$ – ‘nuisance’ parameters

With $P \gg N$, we must work on **variable selection**, standard OLS won’t do it...

What pops into your mind? **LASSO!**
Double-Selection Approach to Lasso (2)

Bad Approach:
Running standard Lasso, while forcing \( D \) to be always selected, will work poorly.

- Lasso does not necessarily always selects all the ‘correct’ variables
- Lasso targets prediction and will omit variables that are highly correlated with \( D \), as they are not needed for prediction...
- This may result into severe omitted variables bias.
- In general, bootstrap won’t help...

GOOD Approach: Double-Selection Lasso
GOOD Approach: Double-Selection Lasso

1. Run first-stage selection using Lasso:

\[ Y_i = X_i' \beta_1 + \nu_{1,i} \]  

(12)

2. Run second-stage selection using Lasso:

\[ D_i = X_i' \beta_2 + \nu_{2,i} \]  

(13)

3. Run OLS with the union of variables selected in Stage 1 and Stage 2: \( \tilde{X}_i \in S | (\beta_{1,i} \neq 0 \text{ or } \beta_{2,i} \neq 0) \):

\[ Y_i = \alpha D_i + \tilde{X}_i' \theta + \varepsilon \]  

(14)
**Double-Selection Lasso: Simulation Experiment**

[Follows Chernozhukov et al. 2016]

\[ Y_i = D_i \theta + X_i' \alpha_1 + u_i \]  \hspace{1cm} (15)

\[ D_i = X_i' \alpha_2 + v_i \]  \hspace{1cm} (16)

\[
\begin{align*}
X & \text{ is } [N_{\text{obs}} \times k] \\
\text{The model is “approximately sparse”, i.e.} & \\
\alpha_{1,j} & = 1/j^2, \alpha_{2} = 0.7 \alpha_{1} \quad j = 1, \ldots, k \\
\text{The covariates are correlated and Normally distributed} & \\
x & \sim N(0, \Sigma), \Sigma_{kj} = (1/2)^{|j-k|}
\end{align*}
\]
Double-Selection Lasso: Simulation Experiment
($\rho >> N$)
When $p < N$ the OLS is still feasible.

Of course, as degrees of freedom decline, the variance of the estimator of $\theta$ increases dramatically...

Variable/model selection using post double-selection is vastly superior...
Double-Selection Lasso: Simulation Experiment
($p < N$)
Double-Selection Lasso: Instrumental Variables (IV)
Chernozhukov, Hansen, and Spindler (2015)

Consider the following linear IV model:

\[ Y_i = \alpha D_i + X'_i \beta + \varepsilon_i, \] \hspace{1cm} (17)
\[ D_i = X'_i \gamma + Z'_i \delta + u_i, \] \hspace{1cm} (18)
\[ Z_i = \Pi X_i + \zeta_i \] \hspace{1cm} (19)

where

- \( Y_i \) – outcome for \( i \)-th observation
- \( D_i \) – endogenous variable of interest
- \( X_i \) – potentially high-dim. vector of covariates
- \( Z_i \) – potentially high-dim. vector of instruments
- \( \alpha \) – parameter of interest

Assume the dimension of full parameter vector is much larger than estimate with available sample size...
Double-Selection Lasso: Instrumental Variables (IV)

Using simple substitutions, the system

\[ Y_i = \alpha D_i + X_i' \beta + \varepsilon_i, \quad (20) \]
\[ D_i = X_i' \gamma + Z_i' \delta + u_i, \quad (21) \]
\[ Z_i = \Pi X_i + \zeta_i \quad (22) \]

can be expressed as a reduced form

\[ Y_i = X_i' \theta + r_i^Y, \quad (23) \]
\[ D_i = X_i' \psi + r_i^D, \quad (24) \]

which depends only on \( X_i \).

Post-lasso double selection strategy that ‘immunizes’ estimation from selection errors exists...
Double-Selection Lasso: Instrumental Variables (IV)

Algorithm:

1. Do post-lasso regression of $D_i$ on $X_i, Z_i$ to get $\hat{\gamma}$ and $\hat{\delta}$.

2. Do post-lasso regression of $Y_i$ on $X_i$ to get $\hat{\theta}$.

3. Define $\hat{D}_i := X_i' \hat{\gamma} + Z_i' \hat{\delta}$ and run post-lasso of $\hat{D}_i$ on $X_i$, getting $\hat{\psi}$.

4. Define

$$\hat{r}_i^Y := Y_i - X_i' \hat{\theta},$$  \hspace{1cm} (25) \\
$$\hat{r}_i^D := D_i - X_i' \hat{\psi},$$  \hspace{1cm} (26) \\
$$\hat{\nu}_i^D := (X_i' \hat{\gamma} + Z_i' \hat{\delta}) - X_i' \hat{\psi}. \hspace{1cm} (27)$$

5. Estimate $\hat{\alpha}$ by standard IV regression of $\hat{r}_i^Y$ on $\hat{r}_i^D$, using $\hat{\nu}_i$ as an instrument. Standard inference applies...
Double-Selection Lasso: Instrumental Variables (IV)

Why this works?

It works because the implied moment condition for the last-stage IV problem

$$E[(\hat{r}_i^Y - \hat{r}_i^D \alpha)\hat{\nu}] = 0 \quad (28)$$

is not overly sensitive (‘immunized’) against lasso selection mistakes, and imperfect estimation of $\gamma, \delta, \psi, \beta$.

Definition of $\hat{\nu}_i^D := (X'_i\hat{\gamma} + Z'_i\hat{\delta}) - X'_i\hat{\psi}$ is very important.

Chernozhukov, Hansen, and Spindler (2015) illustrate that moment condition

$$E[(\hat{r}_i^Y - \hat{r}_i^D \alpha)\hat{D}_i] \equiv E[(\hat{r}_i^Y - \hat{r}_i^D \alpha)(X'_i\hat{\gamma} + Z'_i\hat{\delta})] = 0 \quad (29)$$

is not ‘robust’ to selection errors...
Double-Selection Lasso: Simulation Experiment
Digression: Neyman-Rubin Potential-Outcomes Model

Let’s consider a **binary treatment** $D \in \{0, 1\}$.

**Fundamental Problem of Causal Inference:**

We never observe the effects of treatment and non-treatment for the given individual (or unit)...

**Potential Outcomes:**

- $y_i(0)$ – outcome if unit $i$ **not treated**, $D = 0$
- $y_i(1)$ – outcome if unit $i$ **treated**, $D = 1$

We observe either $y_i(0)$ or $y_i(1)$ but never both. We observe one outcome, the other being **counter-factual**.
Digression: Neyman-Rubin Potential-Outcomes Model

**Treatment effect:** \( \tau = y_i(1) - y_i(0) \)

It is impossible to learn the treatment effect using the observed data without additional assumptions...

The assumptions differ for:

(a) **Randomized controlled trials (RCT)**
   (treatment \( D \) is random, independent of outcome and control variables...)

(b) **Observational studies**
   (treatment \( D \) may depend on control variables, \( X \)...)

Digression: Neyman-Rubin Potential-Outcomes Model

Unconfoundedness Assumption
Treatment assignment unconfounded when treatment is independent of potential outcomes, after conditioning on controls (confounders)

Intuitively, one searches for treated unit $j$ that’s “as similar as possible” to a non-treated unit $i$ and compare their outcomes...

Conditional Average Treatment Effect (CATE)

$$CATE(x) = \tau(x) = E[Y|D = 1, X = x] - E[Y|D = 0, X = x]$$

$$= E[Y(1)|D = 1, X = x] - E[Y(0)|D = 0, X = x]$$

$$= E[Y(1)|X = x] - E[Y(0)|X = x] \quad (30)$$

Average Treatment Effect (ATE)

$$ATE = E[CATE(x)] \quad (31)$$
Let $p = \Pr(D_i = 1)$ be the marginal treatment probability, and let

$$e(x) = \Pr(D_i = 1 | X_i = x)$$

be the **propensity score** (conditional treatment prob.)

One option for using ML is to estimate flexible predictive models for the propensity score... [low-hanging fruit] for propensity-score matching.
Causal Forests
Stefan Wager and Susan Athey (2018)

Using random forest to estimate heterogeneous treatment effects...  
\[ \tau(x) = E[y_i(1) - y_i(0) | X_i = x]. \]  
(33)

The point is to allocate “similar as possible” units, treated and non-treated, to the trees’ leafs and read-off the treatment effect within the leaf...  

Trees are very good at creating sets of ‘similar’ units, with the similarity being defined with respect to outcome or probability of treatment...  

**Causal trees** make sure each leaf has at least \( k \) observations from both treated and non-treated groups
Having observed data \((y_i, x_i, D_i)\), we want to estimate

\[
\tau(x) = E[y_i(1) - y_i(0)|X_i = x].
\]  

(34)

Given a tree with leaves \(L_k, k = 1, \ldots, K\) we estimate CATE(x) as difference of average outcome of treated and non-treated units within the leaf where the unit with control features \(x\) is a member of.

\[
\hat{\tau}(x) = \text{avg}(y_i)_{\{i: D_i = 1, x_i \in L\}} - \text{avg}(y_i)_{\{i: D_i = 0, x_i \in L\}}
\]  

(35)
Causal Forests

Random forests can be viewed as ‘adaptive nearest neighbors’

**What is the difference to k-NN then?**

Trees and forests do not measure ‘closeness’ just based on $X$, but mainly by their effects on outcome (prob. of treatment, say...). Closeness is defined as being member of the same leaf of a decision tree...

This allows the implicit functional form defining closeness to reflecting the strength of the information (signal)

TODO: Generalized Random Forests and ”weigted adaptive NN”
Honest Trees
In order to get valid inference, Wager and Athey (2018) require trees to be ‘honest’

- Honesty is related to over-fitting.
- Honesty means that the treatment effect is computed using data $y_i$ that were NOT used for training the model.
- Honesty can be achieved by **sample splitting** (sample splitting again!)
Athey and Wager propose two types of causal trees

- **Double-Sample Trees**
  1. Split sample in halves, $S_A$ and $S_B$.
  2. Use sample $S_A$ to train the splits for trees.
  3. Use sample $S_B$ for within-leaf estimation, given the tree structure.

- **Propensity Trees** (Also Wang et al. 2015)
  1. Train a classification tree, with $X_i$ predicting treatment, $D_i$.
  2. Each leaf must have at least $k$ observations on treated and non-treated units, each.
  3. Estimate $\tau(x)$ within the leaf containing $x$.

Propensity trees training uses no information on outcome, $Y_i$. 
Causal forests use a splitting criterion that maximizes the **variance** of treatment effects, $\hat{\tau}(x)$.

Wager and Athey (2018) motivate this by an analogy of regression trees, where splits are done to minimize the mean-square error.

Finding the split that minimizes the mean-square error amounts to maximizing the variance of the prediction.

$$
\sum_{i \in L} (\hat{\mu}(x_i) - Y_i)^2 = \sum_{i \in L} (Y_{i}^{obs})^2 - \sum_{i \in L} \hat{\mu}(x_i)^2
$$
Causal Forests and Trees
Implementation:

Implementation in R, Python, or Matlab straightforward...

Ahrens et al. (2018) implement rich lasso toolbox in **Stata**:

- post double-selection lasso (pdslasso)
- elastic net, ridge, lasso, cross-validation, ...
- ...
References


- Instrumental Variables Estimation with Very Many Instruments and Controls, 2015


Thank you for your patience...