AEA Continuing Education Program

Machine Learning and Econometrics

Susan Athey, Stanford Univ.
Guido Imbens, Stanford Univ.

January 7-9, 2018
Machine Learning for Economics: An Introduction

SUSAN ATHEY (STANFORD GSB)
# Two Types of Machine Learning

## SUPERVISED

- Independent observations
- Stable environment
- Regression/prediction:
  - $E[Y|X=x]$
- Classification
  - $Pr(Y=y|X=x)$

## UNSUPERVISED

- Collections of units characterized by features
  - Images
  - Documents
  - Individual internet activity history
- Find groups of similar items
Classification

Advances in ML dramatically improve quality of image classification
Classification

Neural nets figure out what features of image are important
Features can be used to classify images
Relies on stability

Given $X_i$, is this a cat?

$$\Pr(Y_i = \text{CAT}|X_i) = .95$$
$$\Pr(Y_i = \text{DOG}|X_i) = .05$$
What’s New About ML?

Flexible, rich, data-driven models

Increase in personalization and precision

Methods to avoid overfitting
Ability to Fit Complex Shapes
Prediction in a Stable Environment

Goal: estimate $\mu(x) = E[Y | X = x]$ and minimize MSE in a new dataset where only $X$ is observed

- MSE: $\frac{1}{n} \sum_i (Y_i - \hat{\mu}(X_i))^2$
- No matter how complex the model, the output, the prediction, is a single number
- Can hold out a test set and evaluate the performance of a model
- Ground truth is observed in a test set
- Only assumptions required: independent observations, and joint distribution of $(Y,X)$ same in test set as in training set

Note: minimizing MSE entails bias-variance tradeoff, and always accept some bias

- Idea: if estimator too sensitive to current dataset, then procedure will be variable across datasets
- Models are very rich, and overfitting is a real concern, so approaches to control overfit necessary

Idea of ML algorithms

- Consider a family of models
- Use the data to select among the models or choose tuning parameters
- Common approach: cross-validation
  - Break data into 10 folds
  - Estimate on 9/10 of data, estimate MSE on last tenth, for each of a grid of tuning parameters
  - Choose the parameters that minimize MSE

ML works well because you can accurately evaluate performance without add’l assumptions

- Your robotic research assistant then tests many models to see what performs best
# What We Say v. What We Do (ML)

<table>
<thead>
<tr>
<th>What we say</th>
<th>What we do</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML = Data Science, statistics</td>
<td>Use predictive models and ignore other considerations</td>
</tr>
<tr>
<td>Is there anything else?</td>
<td>E.g. Causality, equilibrium or feedback effects</td>
</tr>
<tr>
<td>Use language of answering questions or solving problems, e.g. advertising allocation, salesperson prioritization</td>
<td>Wonder/worry about interpretability/reliability/robustness/ adaptability, but have little way to ask algos to optimize for it</td>
</tr>
<tr>
<td>Aesthetic: human analyst does not have to make any choices</td>
<td></td>
</tr>
<tr>
<td>All that matters is prediction</td>
<td></td>
</tr>
</tbody>
</table>
Contrast with Traditional Econometrics

Economists have focused on the case with substantially more observations than covariates (N>>P)

- In-sample MSE is a good approximation to out-of-sample MSE
- OLS is BLUE, and if overfitting is not a problem, then no need to incur bias
- OLS uses all the data and minimizes in-sample MSE

OLS obviously fails due to overfitting when P~N and fails entirely when P>N

- ML methods generally work when P>N

Economists worry about estimating causal effects and identification

- Causal effects
- Counterfactual predictions
- Separating correlation from causality
- Standard errors
- Structural models incorporating behavioral assns

Identification problems can not be evaluated using a hold-out set

- If joint dist’n of observable same in training and test, will get the same results in both

Causal methods sacrifice goodness-of-fit to focus only on variation in data that identifies parameters of interest
# What We Say v. What We Do

## (Econometrics)

### What We Say
- Causal inference and counterfactuals
- God gave us the model
- We report estimated causal effects and appropriate standard errors
- Plus a few additional specifications for robustness

### What we do
- Run OLS or IV regressions
  - Try a lot of functional forms
  - Report standard errors as if we ran only one model
  - Have research assistants run hundreds of regressions and pick a few “representative” ones
- Use complex structural models
  - Make a lot of assumptions without a great way to test them
Key Lessons for Econometrics

Many problems can be decomposed into predictive and causal parts
- Can use off-the-shelf ML for predictive parts

Data-driven model selection
- Tailored to econometric goals
  - Focus on parameters of interest
  - Define correct criterion for model
  - Use data-driven model selection where performance can be evaluated
- While retaining ability to do inference

ML-Inspired Approaches for Robustness

Validation
- ML always has a test set
- Econometrics can consider alternatives
  - Ruiz, Athey and Blei (2017) evaluate on days with unusual prices
  - Athey, Blei, Donnelly and Ruiz (2017) evaluate change in purchases before and after price changes
  - Tech firm applications have many A/B tests and algorithm changes

Other computational approaches for structural models
- Stochastic gradient descent
- Variational Inference (Bayesian models)

See Sendhil Mullainathan et al (JEP, AER) for key lessons about prediction in economics
- See also Athey (Science, 2017)
Empirical Economics in Five Years: My Predictions

<table>
<thead>
<tr>
<th>Regularization/data-driven model selection will be the standard for economic models</th>
<th>Models will explicitly distinguish causal parts and predictive parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction problems better appreciated</td>
<td>Reduced emphasis on sampling variation</td>
</tr>
<tr>
<td>Measurement using ML techniques an important subfield</td>
<td>Model robustness emphasized on equal footing with standard errors</td>
</tr>
<tr>
<td>Textual analysis standard (already many examples)</td>
<td>Models with lots of latent variables</td>
</tr>
</tbody>
</table>
An Introduction to Regularized Regression

Machine Learning and Causal Inference

Susan Athey

Thanks to Sendhil Mullainathan for sharing slides
What we do in Econometrics: 
The Case of Regression

• Specify a model:
  \[ Y_i = f(X_i) + \epsilon_i = X_i \beta + \epsilon_i \]
  
• Data set has observations \( i = 1, \ldots, n \)

• Use OLS regression on the entire dataset to construct an estimate \( \hat{\beta} \)

• Discuss assumptions under which some components of \( \hat{\beta} \) have a causal interpretation

• Consider that \( S_n \) (set of observed units, \( i = 1, \ldots, n \)) is a random sample from a much larger population.

• Construct confidence intervals and test the hypothesis that some components are equal to zero.

• Theorem: OLS is BLUE (Best Linear Unbiased Estimator)
  - Best = lowest-variance
Goals of Prediction and Estimation

• Goal of estimation: unbiasedness

\[ E[\hat{f}] = f \]

• Goal of prediction: loss minimization

\[ L(f) = E_{(x,y)} \ell(f(x), y) \]
\[ \hat{f} \approx \min_{f \in F} L(f) \]

– E.g. \( \ell(f(x), y) = (f(x) - y)^2 \)
– Use the data to pick a function that does well on a new data point
Key assumptions in both cases

• Stationary data generating process
  
  \[ S_n = (y_i, x_i) \text{ iid} \]

• Estimation:
  – Interested in a parameter of that process

• Prediction:
  – Interested in predicting \( y \)
High v. Low Dimensional Analysis

• We have discussed prediction as a high dimensional construct

• Practically that is where it is useful

• But to understand how high dimensional prediction works we must unpack an implicit presumption
  – Presumption: Our known estimation strategies would be great predictors *if they were feasible*
A Simple OLS example

• Suppose we truly live in a linear world

\[ y = \beta_0 + \beta_1 x_1 + \varepsilon \]

\[ \varepsilon \sim \mathcal{N}(0, \sigma^2_\varepsilon) \quad x_1 \sim \mathcal{N}(0, 1) \]

• Write \( x = (1, x) \)

\[ y = \beta x + \varepsilon. \]
OLS seems like a good predictor

\[ L(f^{\text{OLS}}) = E_{(y,x)}(\hat{\beta}' x - y)^2 = (\hat{\beta}_0 - \beta_0)^2 + (\hat{\beta}_1 - \beta_1)^2 + \sigma^2_\varepsilon \]

So wouldn’t we want the \( \hat{\beta} \) with \( E_{S_n}(\hat{\beta}) = \beta \)?

Especially since it is known to be efficient
An Even Simpler Set-up

• Let’s get even lower dimensional

• No variables at all

• Suppose you get the data of the type:

\[ y_i = \mu + \epsilon_i \]

• You would like to estimate the mean
Forming an estimator of the mean

\[ \mu \hat{\mu} = \alpha \bar{y} \]

\[ E[\hat{\mu}] = \alpha \mu \]

- Minimize bias: \( \alpha = 1 \)
- The sample mean is an unbiased estimator
  - Also what you would get from OLS regression on a constant
A prediction problem

• In the same setup, you are given \( n \) data points

• You would like to guess the value of a new data point from the same distribution

• Goal: minimize quadratic loss of prediction
Best Predictor

\[ \hat{\mu} = \alpha \bar{y} \]

\[ E[\hat{\mu}] = \alpha \mu \]

\[ E[\ell(\hat{\mu}, y)] = [(1 - \alpha) \mu]^2 + \frac{1}{n} \alpha^2 \sigma^2 + \sigma^2 \]
The higher alpha the lower the bias

\[ \hat{\mu} = \alpha \bar{y} \]

\[ y_i = \mu + \epsilon_i \]

\[ \begin{align*}
S^1 & \quad \hat{\mu}^1 \\
S^2 & \quad \hat{\mu}^2 \\
S^3 & \quad \hat{\mu}^3 \\
S^4 & \quad \hat{\mu}^4 \\
S^5 & \quad \hat{\mu}^5
\end{align*} \]

The higher alpha the more variable across samples it is
Key problem

• The unbiased estimator has a nice property:

\[ E[\hat{\mu}|\mu] = \mu \]

• But getting that property means large sample to sample variation of estimator

• This sample to sample variation means that in any particular finite sample I’m paying the cost of being off on all my predictions
Intuition

• I see your first test score. What should my prediction of your next test be?
  – Your first test score is an unbiased estimator
  – But it is very variable

• Note: “Bayesian” intuition
  – Even simpler: what was my guess before I saw any information
  – Shrink to that
  – In this example I’m shrinking to zero
But in a way you know this

- As empiricists you already have this intuition
Back to Simple OLS example

• Suppose we truly live in a linear world

\[ y = \beta_0 + \beta_1 x_1 + \varepsilon \]

\[ \varepsilon \sim \mathcal{N}(0, \sigma^2_\varepsilon) \quad x_1 \sim \mathcal{N}(0, 1) \]

• Write \( x = (1, x) \)

\[ y = \beta x + \varepsilon. \]
A Simple Example

• You run a one variable regression and get

\[ \hat{\beta}_0^{OLS} = 0 \pm 0.2 \]
\[ \hat{\beta}_1^{OLS} = 2 \pm 10 \]

• Would you use the OLS coefficients to predict

• Or drop the first variable and use this:

\[ \hat{\beta}_0^{OLS} = \arg \min_{\beta_0} \hat{E}_S n (\beta_0 - y)^2 = \hat{E}_S n y \]
Deciding whether to drop

• Suppose in the (impossible) case we got the true world right.
  – (0,2) are the right coefficients

• Of course OLS does perfectly (by assumption).

• But how would OLS do on new samples…where (0,2) being the generating coefficients?
  – We’re giving OLS a huge leg up here.
OLS Performance

\[ \mathcal{L}_n(\text{OLS}) - \sigma^2 = \]

\[ = E_{(y,x)} E S_n [\beta' x - (\hat{\beta}^{\text{OLS}})' x]^2 \]

\[ = E_{(y,x)} \left[ (\beta' x - (E S_n \hat{\beta}^{\text{OLS}})' x)^2 \right] + \text{Var}_S ((\hat{\beta}^{\text{OLS}})' x) \]

\[ = \text{Var}_S (\hat{\beta}^{0\text{OLS}}) + \text{Var}_S (\hat{\beta}^{1\text{OLS}}) \]
What if we dropped the variable

\[ \mathcal{L}_n(\text{OLS}_0) - \sigma^2_\varepsilon = \]
\[ \mathcal{L}_n(\text{OLS}) - \sigma^2 = \]
\[ = \mathbb{E}(y,x) E_{S_n} [\beta'x - (\hat{\beta}_{\text{OLS}})'x]^2 \]
\[ = \mathbb{E}(y,x) [(\beta'x - (E_{S_n} \hat{\beta}_{\text{OLS}})'x)^2 + \text{Var}_{S_n}((\hat{\beta}_{\text{OLS}})'x)] \]
\[ = \text{Var}_{S_n}(\hat{\beta}_{0\text{OLS}}) + \text{Var}_{S_n}(\hat{\beta}_{1\text{OLS}}) \]

\[ \mathcal{L}_n(\text{OLS}_0) - \sigma^2 = \]
\[ = \mathbb{E}(y,x) E_{S_n} [\beta'x - (\hat{\beta}_{\text{OLS}_0}'x]^2 \]
\[ = \mathbb{E}(y,x) [(\beta'x - (E_{S_n} \hat{\beta}_{\text{OLS}_0})'x)^2 + \text{Var}_{S_n}((\hat{\beta}_{\text{OLS}_0})'x)] \]
\[ = (0 - 2)^2 + \text{Var}_{S_n}(\hat{\beta}_{0\text{OLS}_0}) + \text{Var}_{S_n}(\hat{\beta}_{1\text{OLS}_0}) \]

\[ \mathcal{L}_n(\text{OLS}) - \mathcal{L}_n(\text{OLS}_0) = \frac{\text{Var}(\hat{\beta}_{1\text{OLS}}) - (0 - 2)^2}{\text{variance bias}} \]

Your standard error worry!
Where does your standard error intuition come from?

• You see a standard error

• You think “that variable is not ‘significant’” so you might not want to include it.

• But this is misleading
What parameters could generate this estimate?
What estimates could have been generated?

What parameters could generate this estimate?

Variability of Estimator
Your Standard Error Worry

• For hypothesis testing se tells you whether the coefficient is significant are not

• For prediction it’s telling you how variable an estimator using it really is
Dual purposes of the standard error

• The standard error also tells you that even if you’re right on average:
  – Your estimator will produce a lot of variance
  – And then in those cases you make systematic prediction mistakes.

• Bias variance tradeoff
  – Being right on average on the coefficient is not equal to the best predictor.
The Problem Here

• Prediction quality suffers from:
  – Biased coefficients
  – Variability in *estimated* coefficients
    • Even if the true coefficient is 2, in any sample, we will estimate something else

• OLS is lexicographic
  – First ensure unbiased
  – Amongst unbiased estimators: seek efficiency

• Good predictions must trade these off
Two Variable Example

• Belaboring the point here…
• Assume now that we have two variables
  – As before, both normally distributed unit variance
• Your estimator produces
  \[
  \hat{\beta}_0^{OLS} = 0 \pm 0.2 \\
  \hat{\beta}_1^{OLS} = 2 \pm 10
  \]
What would you do now?

• Logic above suggests you would drop both variables?

• Or keep both variables?

• It really depends on how you feel about the variance (10)?
Calculation

\[
\mathcal{L}_n(\text{OLS}) - \mathcal{L}_n(\text{OLS}_0) = \underbrace{\text{variance}}_{\text{variance}} \underbrace{\text{bias}}_{\text{bias}} \\
= \text{Var}(\hat{\beta}_1^{\text{OLS}}) + \text{Var}(\hat{\beta}_2^{\text{OLS}}) - ((0 - 2)^2 + (0 - 2)^2) \\
+ 2\rho_{12} \text{Cov}(\hat{\beta}_1^{\text{OLS}}, \hat{\beta}_2^{\text{OLS}}) - 2\rho_{12}(0 - 2)^2 \\
\text{covariance variance covariance bias}
\]

\[
\mathcal{L}_n(\text{OLS}) - \mathcal{L}_n(\text{OLS}_0) = \underbrace{\text{variance}}_{\text{variance}} \underbrace{\text{bias}}_{\text{bias}} \\
= \text{Var}(\hat{\beta}_1^{\text{OLS}}) - (0 - 2)^2 \\
\text{variance bias}
\]
Hidden in Bias-Variance Tradeoff

• Covariance is central

• The standard error on several variables can be large, even though together their effect is highly consistent

• For prediction covariance between x matters
In a way this problem is not important

\[ \mathcal{L}_n(\text{OLS}) - \mathcal{L}_n(\text{OLS}_0) = \text{Var}(\hat{\beta}_1^{\text{OLS}}) - (0 - 2)^2 \]

• The variance term diminishes with sample size
  – Prediction-estimation wedge falls off as \( \frac{1}{n} \)

• But variance term increases with “variables”
  – Prediction-estimation rises with \( k \)

• So this is a problem when…
  – Function class high dimensional relative to data \( \frac{k}{n} \)
What this means practically

• In some cases what you already know (estimation) is perfectly fine for prediction
  – This is why ML textbooks teach OLS, etc.
  – They are perfectly useful for the kinds of prediction problems ML tries to solve in low dimensional settings

• But in high dimensional settings…
  – Note: high dimensional does not ONLY mean lots of variables! It can mean rich interactions.
So far…

• All this gives you a flavor of how the prediction task is not mechanically a consequence of the estimation task.

• But it doesn’t really tell you **how** to predict:
  – Bias variance tradeoff is entirely unactionable
  – What’s the bias?
  – What’s the variance?
  – This is not really a tradeoff you can make.

• A different look at the same problem produces a practical insight though.
Back to OLS

\[
\hat{\beta}^{\text{OLS}} = \arg \min_{\beta} \mathbb{E}_{S_n} (\beta' x - y)^2
\]

\[
\beta^\text{prediction} = \arg \min_{\beta} \mathbb{E}_{(y,x)} (\beta' x - y)^2
\]

- The real problem here is minimizing the “wrong” thing: In-sample fit vs out-of-sample fit
Overfit problem

• OLS looks good with the sample you have
  – It’s the best you can do *on this sample*

• Bias-variance improving predictive power is about improving *out of sample* predictive power

• Problem is OLS by construction overfits
  – We overfit in estimation
This problem is exactly why wide data is troubling

• Similarly think of the wide data case

• Why are we worried about having so many variables?

• We’ll fit very well (perfectly if $k > n$) in sample

• But arbitrarily badly out of sample
Understanding overfit

• Let’s consider a general class of algorithms
A General Class of Algorithms

- Let $L(f) = \int_{x,y} \ell(f(x), y)dP(x, y)$ for some loss function $\ell$ (e.g. squared error)
  - Note: $L$ is an unknown function: we don’t know $P$
- Consider algorithms of the form
  \[
  \hat{f}_{A,S_n} = \text{arg min}_{f \in \mathcal{F}_A} \hat{L}_{S_n}(f)
  \]
  - $\hat{L}_{S_n}$ is used here as shorthand for sample mean observations in sample $S_n$ of size $n$
  - OLS is an empirical loss minimizer: it minimizes the sample average over observed data of the loss function

- So empirical loss minimization algorithms are defined by the function class they choose from
- For estimation what we typically do…
  - Show that empirical loss minimizers generate unbiasedness
Empirical Loss minimization

• Leads to unbiasedness/consistency
  – Fit the data you have…
  – In a frequentist world “on average” (across all $S_n$) this will produce the right thing
  – This is usually how we prove consistency/unbiasedness

• Other variants:
  – MLE
Some Notation

• Define

\[ f^* = \arg \min_{f \in \mathcal{F}} L(f) \]  

The best we can do

\[ f^*_A = \arg \min_{f \in \mathcal{F}_A} L(f) \]  

The best in the subset of functions that the algorithm looks at

— Recall: \( L \) is infeasible b/c we don’t know true data-generating process

• Contrast the latter with:

\[ \hat{f}_{A,S_n} = \arg \min_{f \in \mathcal{F}_A} \hat{L}_{S_n}(f) \]  

What the in-sample loss minimizer actually produces given a sample
Performance of Algorithm

- Performance of a predictor
  \[ L(\hat{f}_A, S_n) \]
- Performance of an Algorithm
  \[ \mathcal{L}_n(A) := E_{S_n} L(\hat{f}_A, S_n) \]
  - Algorithm’s expected loss
  - (Suppress \( S_n \) in some of the notation for estimator)
The performance of $A$

$$\mathcal{L}_n(A) = L(f^*) + \underbrace{L(f_A^*) - L(f^*)}_{\text{approximation error}} + \underbrace{E_{S_n}(L(\hat{f}_A) - L(f_A^*))}_{\text{irreducible error}} + \underbrace{E_{S_n}(L(\hat{f}_A) - L(f_A^*))}_{\text{estimation error}}$$

Understanding estimation error:

$$E_{S_n}(\hat{L}(\hat{f}_A) - L(f_A^*)) = E_{S_n}(\hat{L}(\hat{f}_A) - \hat{L}(f_A^*)) + E_{S_n}(L(\hat{f}_A) - \hat{L}(f_A))$$

“Wrong” function looks good in-sample

Algorithm does not see this
Basic Tradeoff

• These two terms go hand in hand:

\[
\mathcal{L}_n(A) = L(f^*) + L(f_A^*) - L(f^*) + E_{S_n}(L(f_A) - L(f_A^*))
\]

irreducible error

approximation error

estimation error

\[
E_{S_n}(L(f_A) - L(f_A^*)) = E_{S_n}(\hat{L}(f_A) - \hat{L}(f_A^*)) + E_{S_n}(L(f_A) - \hat{L}(f_A))
\]

unseen overfit

out-of-sample, ≥0

in-sample, ≤0
Approximation – Overfit Tradeoff

• If we reduce set of \( f \) to reduce possible over-fit:

\[
E_{S_n}(L(\hat{f}_A) - \hat{L}(\hat{f}_A))
\]
unseen overfit

• Then we fit fewer “true functions” and drive up

\[
\text{approximation error} \\
L(f^*_A) - L(f^*)
\]

• Only way to avoid this is if we knew information about \( f^* \) so we could shrink the set
Unobserved overfit

• So the problem of prediction really is managing unobserved overfit

\[
L(\hat{f}_A) = \hat{L}(\hat{f}_A) + (L(\hat{f}_A) - \hat{L}(\hat{f}_A))
\]

unobserved out-of-sample unobserved overfit

• We do well in-sample. But some of that “fit” is overfit.
Return to the original example

\[ OLS \quad OLS_0 \]

Greater Chance  Less Chance
To Overfit      To Overfit

- We drove down overfit by doing a constrained optimization
Basic Tradeoff at the Heart of Machine Learning

• Bigger function classes…
  – The more likely we are to get to the truth (less approximation)
  – The more likely we are to overfit

• So we want to not just minimize in-sample error given a class of functions

• We also want to decide on the class of functions
  – More expressive means less approximation error
  – More expressive means more overfit
Let’s do the same thing here

Unconstrained

\[ \hat{f}_{A, S_n} = \arg \min_{f \in \mathcal{F}_A} \hat{L}_{S_n}(f) \]

But we are worried about

\[ \mathbb{E}_{S_n}(L(\hat{f}_A) - \hat{L}(\hat{f}_A)) \]
unseen overfit

So why not do this instead?

\[ \arg \min_{f \in \mathcal{F}_A} \hat{L}_{S_n}(f) \]

s.t. \( R(f) \leq c \)

Complexity measure: tendency to overfit
Return to the original example

\[ \text{\textit{OLS}} \quad \text{\textit{OLS}_0} \]

- Greater Overfit
- Better approximation
- More Expressive
- \(R(f) \) higher

- Less Overfit
- Worse approximation
- Less Expressive
- \(R(f) \) lower

- Reduce overfit by approximating worse
- Choose less expressive function class
Constrained minimization

• We could do a constrained minimization

• But notice that this is equivalent to:

\[
\hat{f}_{A\lambda, S_n} = \arg\min_{f \in \mathcal{F}_A} \hat{L}_{S_n}(f) + \lambda R(f)
\]

\[
\text{want: } \approx L(f) - \hat{L}(f)
\]

• Complexity measure should capture tendency to overfit
Basic insight

• Data has signal and noise

• More expressive function classes-
  – Allow us to pick up more of the signal
  – But also pick up more of the noise

• So the problem of prediction becomes the problem of choosing expressiveness
Overall Structure

• Create a regularizer that:
  – Measures expressiveness

• Penalize algorithm for choosing more expressive functions
  – Tuning parameter lambda

• Let it weigh this penalty against in-sample fit
Linear Example

• Linear function class $x \mapsto \beta' x$ ($\beta \in \mathbb{R}^{k+1}$)

• Regularized linear regression

$$\hat{\beta}_\lambda^R = \arg \min_{\beta \in \mathbb{R}^{k+1}} \widehat{E}_{S_n} (\beta' x - y)^2 - \lambda R(\beta)$$
Regularizers for Linear Functions

- Linear functions more expressive if use more variables

\[ R(\beta) = \sum_{j=1}^{k} 1_{\beta_j \neq 0} \]

- Can transform coefficients

\[ R(\beta) = \sum_{j=1}^{k} |\beta_j|^p \]
Computationally More Tractable

- **Lasso**
  \[ \mathcal{F}_{1,c} = \{ f_\gamma ; \sum_{j=1}^{k} |\gamma_j| \leq c \} \]

- **Ridge**
  \[ \mathcal{F}_{2,c} = \{ f_\gamma ; \sum_{j=1}^{k} \gamma_j^2 \leq c \} \]
What makes a good regularizer?

• You might think…
  – Bayesian assumptions
  – Example: Ridge

• A good regularizer can build in beliefs

• Those are great and useful when available

• But central force is tendency to overfit

• Example:
  – Even if true world were not sparse or priors were not normal you’d still do this
Summary

• Regularization is one half of the secret sauce

• Gives a single-dimensional way of deciding of capturing expressiveness

\[ \hat{f}_{A\lambda,S_n} = \arg\min_{f \in \mathcal{F}_A} \hat{L}_{S_n}(f) + \lambda R(f) \]

• Still missing ingredient is lambda
Choosing lambda

- How much should we penalize expressiveness?
- How do you make the over-fit approximation tradeoff?
- The tuning problem.
- Use cross-validation
How Does Cross Validation Work?

Tuning Set = 1/5 of Training Set

Tuning Set
Train
Tune
CV-Tuning
CV-Training
Cross-Validation Mechanics

• Loop over cross-validation samples
  – Train a deep tree on CV-training subset

• Loop over penalty parameters $\lambda$
  – Loop over cross-validation samples
    • Prune the tree according to penalty
    • Calculate new MSE of tree
  – Average (over c-v samples) the MSE for this penalty

• Choose the penalty $\lambda^*$ that gives the best average MSE
LASSO c-v Example
Creating Out-of-Sample In Sample

• Major point:
  – Not many assumptions
  – Don’t need to know true model.
  – Don’t need to know much about algorithm

• Minor but important point
  – To get asymptotics right we need to make some regularity assumptions

• Side point (to which we return)
  – We’d like to choose best algorithm for sample size $n$
  – But this will not do that. Why?
Why does this work?

1. Not just because we can split a sample and call it out of sample

   – It’s because the thing we are optimizing is observable (easily estimable)
This is more than a trick

• It illustrates what separates prediction from estimation:
  – I can’t ‘observe’ my prior.
    • Whether the world is truly drawn from a linear model
  – But prediction quality is observable

• Put simply:
  – Validity of predictions are measurable
  – Validity of coefficient estimators require structural knowledge

*This is the essential ingredient to prediction: Prediction quality is an empirical quantity not a theoretical guarantee*
Why does this work?

1. It’s because the thing we are optimizing is **observable**

2. By focusing on prediction quality we have **reduced dimensionality**
To understand this…

• Suppose you tried to use this to choose coefficients
  – Ask which set of coefficients worked well out-of sample.

• Does this work?

• Problem 1: Estimation quality is unobservable
  – Need the same assumptions as algorithm to know whether you “work” out of sample
    • If you just go by fit you are ceding to say you want best predicting model

• Problem 2: No dimensionality reduction.
  – You’ve got as many coefficients as before to search over
\[
\hat{\beta}_\lambda^R = \arg\min_{\beta \in \mathbb{R}^{k+1}} \mathbb{E}_{S_n} (\beta' x - y)^2 + \lambda R(\beta)
\]

<table>
<thead>
<tr>
<th>Method</th>
<th>( R(\beta) )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OLS</strong></td>
<td>0</td>
</tr>
<tr>
<td><strong>Subset selection</strong></td>
<td>( | \beta |<em>0 = \sum</em>{j=1}^{k} 1 \beta_j \neq 0 )</td>
</tr>
<tr>
<td><strong>Lasso</strong></td>
<td>( | \beta |<em>1 = \sum</em>{j=1}^{k} \beta_j )</td>
</tr>
<tr>
<td><strong>Ridge</strong></td>
<td>( | \beta |<em>2^2 = \sum</em>{j=1}^{k} \beta_j^2 )</td>
</tr>
<tr>
<td><strong>Elastic Net</strong></td>
<td>( \alpha | \beta |_1 + (1 - \alpha) | \beta |_2^2 )</td>
</tr>
</tbody>
</table>
Bayesian Interpretation of Ridge

Consider the regression

$$Y_i = \sum_{k=1}^{K} \beta_k \cdot X_{ik} + \varepsilon_i$$

with

$$\varepsilon_i | X_{i1}, \ldots, X_{iK} \sim \mathcal{N}(0, \sigma^2)$$

Suppose we put a prior on the $\beta_k$:

$$\beta_k \sim \mathcal{N}(0, \tau^2)$$

and all the $\beta_k$ independent. Assume $\sigma^2$ is known.
Bayesian Interpretation of Ridge

Then the posterior distribution is proportional to

\[ p(\beta|\text{data}) \propto \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^{N} \left( Y_i - \sum_{k=1}^{K} \beta_k \cdot X_{ik} \right)^2 \right) \prod_{k=1}^{K} \exp \left( -\frac{\beta_k^2}{2\tau^2} \right) \]

\[ = \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^{N} \left( Y_i - \sum_{k=1}^{K} \beta_k \cdot X_{ik} \right)^2 - \sum_{k=1}^{K} \frac{\beta_k^2}{2\tau^2} \right) \]

\[ = \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (Y_i - \beta'X_i)^2 - \frac{\beta'\beta}{2\tau^2} \right) \]
Bayesian Interpretation of Ridge

So, the posterior is normal, and the posterior mean minimizes

$$\sum_{i=1}^{N} (Y_i - \beta'X_i)^2 + \beta' \beta \cdot \frac{\sigma^2}{\tau^2}$$

$$= \sum_{i=1}^{N} (Y_i - \beta'X_i)^2 + \frac{\sigma^2}{\tau^2} \cdot \|\beta\|^2$$

This leads to the posterior mean

$$\left(X'X + I_K \cdot \sigma^2/\tau^2\right)^{-1} X'Y.$$

If the $X'X$ matrix is diagonal, all elements of $\beta$ would be shrunk towards zero by the same fraction. With a non-diagonal matrix the degree of shrinkage varies.
POST-Lasso

• Important distinction:
  – Use LASSO to choose variables
  – Use OLS on these variables

• How should we think about these?
In the orthonormal case, i.e. $X^T X = I = (X^T X)^{-1}$:

$$
\hat{\beta}_j(\lambda_1) = \text{sgn}(\hat{\beta}_j)(|\hat{\beta}_j| - \lambda_1/2)_+
$$

That is, the lasso estimate is related to the OLS estimate via the so-called *soft threshold function* (depicted here for $\lambda=1$).
Why not Hard Thresholding?

Soft Thresholding

\( \hat{\beta}_{LASSO} \)

\( \hat{\beta}_{OLS} \)
\\hat{\beta}_{OLS}?
Orthonormal: \[ \hat{\beta}_{\text{RIDGE}} = \frac{\hat{\beta}_{\text{OLS}}}{1 + \lambda} \]
$\hat{\beta}_{OLS}$?
Can be very misleading
Coefficient on Number of Bedrooms
Coefficient on Number of Bedrooms

Graph showing the coefficient on the number of bedrooms as a function of \( \lambda \) for different methods.
What is this about?

Coefficient on Number of Bedrooms
Prediction Policy

Susan Athey-Machine Learning and Causal Inference
Thanks to Sendhil Mullainathan for sharing his slides
Three Direct Uses of Prediction

1. Policy

2. Testing Whether Theories are Right

3. Testing Theory Completeness
When is Prediction Primary Focus?

- Economics: “allocation of scarce resources”
- An allocation is a decision.
  - Generally, optimizing decisions requires knowing the counterfactual payoffs from alternative decisions.
- Hence: intense focus on causal inference in applied economics
- Examples where prediction plays the dominant role in a decision
  - Decision is obvious given an unknown state
  - Many decisions hinge on a prediction of a future state
Prediction and Decision-Making: Predicting a State Variable

Kleinberg, Ludwig, Mullainathan, and Obermeyer (2015)

• Motivating examples:
  – Will it rain? (Should I take an umbrella?)
  – Which teacher is best? (Hiring, promotion)
  – Unemployment spell length? (Savings)
  – Risk of violation of regulation (Health inspections)
  – Riskiest youth (Targeting interventions)
  – Creditworthiness (Granting loans)

• Empirical applications:
  – Will defendant show up for court? (Should we grant bail?)
  – Will patient die within the year? (Should we replace joints?)
Allocation of Inspections

• **Examples:**
  – Auditors
  – Health inspectors
  – Fire code inspectors
  – Equipment

• **Efficient use of resources:**
  – Inspect highest-risk units
  – *(Assuming you can remedy problem at equal cost for all...)*
Figure 3: Correlation between Yelp Ratings and Hygiene Inspection Scores
Review data consists of reviews on Yelp.com for restaurants in San Francisco, CA from September, 2010 through September, 2013. Hygiene scores consist are from the San Francisco Department of Public Health, for the same timeframe.
Prediction Problem

• Over 750,000 joint replacements every year

• Benefits
  – Improved mobility and reduced pain

• Costs
  – Monetary: $15,000 (roughly)
  – Non-monetary: short-run utility costs as people recover from surgery
Look at death rate in a year
How well are we doing avoiding unnecessary surgery?

Medicare claims data 2010 surgeries for joint replacement

Average death rate is 5%

But is that the right metric for excess joint replacements?

Don’t want average patient

Want marginal patient
   Predictably highest risk patients
Table 2: The Riskiest People Receiving Joint Replacements

<table>
<thead>
<tr>
<th>Predicted Mortality Percentile</th>
<th>Observed Mortality Rate</th>
<th>Total Number Annually</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.562 (.027)</td>
<td>4905</td>
</tr>
<tr>
<td>2</td>
<td>0.530 (.02)</td>
<td>9810</td>
</tr>
<tr>
<td>5</td>
<td>0.456 (.012)</td>
<td>24525</td>
</tr>
<tr>
<td>10</td>
<td>0.345 (.008)</td>
<td>49045</td>
</tr>
<tr>
<td>20</td>
<td>0.228 (.005)</td>
<td>98090</td>
</tr>
<tr>
<td>30</td>
<td>0.165 (.004)</td>
<td>147135</td>
</tr>
<tr>
<td>100</td>
<td>0.057 (.001)</td>
<td>490450</td>
</tr>
</tbody>
</table>

Approach: use ML methods to predict mortality as a function of covariates
- e.g. regularized regression, random forest
- Put individuals into percentiles of mortality risk

A large number of joint replacements going to people who die within the year

Could we just eliminate the ones above a certain risk?
Econometrics of Prediction Policy
Problems

1. Problem: Omitted Payoff Bias
This Unobservable is a Problem

What if those with high Mortality also benefit most?

- Medical History
- Years of Life
- Benefits
- Pain
- Replace Joint
- w
Omitted Payoff Bias

\[ Y = f(X, Z) \]

\[ \Pi = g(X_0, W) \]

\( \text{Cov}(X, Z) \) is not a problem
\( \text{Cov}(X, W) \) is a problem
Econometrics of Prediction Policy

Problems

1. Omitted Payoff Bias
   - Like omitted variable bias but not in $y$
   - Can partially assess on the basis of observables
Table 2: The Riskiest People Receiving Joint Replacements

<table>
<thead>
<tr>
<th>Predicted Mortality Percentile</th>
<th>Observed Mortality Rate</th>
<th>Total Number Annually</th>
<th>PT + Joint Injections</th>
<th>Physician Visits for Osteo.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.562 (.027)</td>
<td>4905</td>
<td>4.4 (.356)</td>
<td>1.4 (.173)</td>
</tr>
<tr>
<td>2</td>
<td>0.530 (.02)</td>
<td>9810</td>
<td>4.0 (.316)</td>
<td>1.8 (.13)</td>
</tr>
<tr>
<td>5</td>
<td>0.456 (.012)</td>
<td>24525</td>
<td>3.9 (.208)</td>
<td>2.0 (.092)</td>
</tr>
<tr>
<td>10</td>
<td>0.345 (.008)</td>
<td>49045</td>
<td>3.8 (.143)</td>
<td>2.1 (.066)</td>
</tr>
<tr>
<td>20</td>
<td>0.228 (.005)</td>
<td>98090</td>
<td>3.9 (.091)</td>
<td>1.8 (.042)</td>
</tr>
<tr>
<td>30</td>
<td>0.165 (.004)</td>
<td>147135</td>
<td>3.8 (.076)</td>
<td>1.9 (.035)</td>
</tr>
<tr>
<td>100</td>
<td>0.057 (.001)</td>
<td>490450</td>
<td>3.9 (.046)</td>
<td>2.1 (.023)</td>
</tr>
</tbody>
</table>

No sign of bias: Highest risk show no signs of greater benefit
Quantifying gain of predicting better

• Allocation problem:
  – Reallocate joints to other eligible patients
• How to estimate the risk of those who didn’t get surgery?
  – Look at those who could get surgery but didn’t
  – Doctors should choose the least risky first
  – So those who don’t receive should be particularly risky.
• Take a conservative approach
  – Compare to median risk in this pool
Table 2: The Riskiest People Receiving Joint Replacements

<table>
<thead>
<tr>
<th>Predicted Mortality Percentile</th>
<th>Observed Mortality Rate</th>
<th>Total Number Annually</th>
<th>Substitute with 50th percentile Eligibles</th>
<th>Annual Savings (in millions)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Futile Procedures Averted</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.562 (.027)</td>
<td>4905</td>
<td>2403</td>
<td>36</td>
</tr>
<tr>
<td>2</td>
<td>0.530 (.02)</td>
<td>9810</td>
<td>4485</td>
<td>67</td>
</tr>
<tr>
<td>5</td>
<td>0.456 (.012)</td>
<td>24525</td>
<td>9398</td>
<td>141</td>
</tr>
<tr>
<td>10</td>
<td>0.345 (.008)</td>
<td>49045</td>
<td>13350</td>
<td>200</td>
</tr>
<tr>
<td>20</td>
<td>0.228 (.005)</td>
<td>98090</td>
<td>15219</td>
<td>228</td>
</tr>
<tr>
<td>30</td>
<td>0.165 (.004)</td>
<td>147135</td>
<td>13548</td>
<td>203</td>
</tr>
<tr>
<td>100</td>
<td>0.057 (.001)</td>
<td>490450</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Assessing the Research Agenda

• Follows economic tradition of using data to improve policy

• In an area of economic interest
  – Similar to a lot of health econ work

• Of course this does not answer all questions of interest
  – Why not?
Another Prediction Policy Problem

• Each year police make over 12 million arrests

• Many detained in jail before trial

• Release vs. detain high stakes
  – Pre-trial detention spells avg. 2-3 months (can be up to 9-12 months)
  – Nearly 750,000 people in jails in US
  – Consequential for jobs, families as well as crime

Kleinberg Lakkaraju Leskovec Ludwig and Mullainathan
Judge’s Problem

• Judge must decide whether to release or not (bail)

• Defendant when out on bail can behave badly:
  – Fail to appear at case
  – Commit a crime

• The judge is making a prediction
Omitted Payoff Bias?

• Bail carefully chosen
  – Unlike other sentencing no other concerns:
    • Retributive justice
  – Family & other considerations low

• Bad use case: Parole Decision
Evaluating the Prediction

• NOT just AUC or Loss

• Use predictions to create a release rule

• What is the release – crime rate tradeoff?

• Note: There’s a problem
Econometrics of Prediction Policy Problems

1. Omitted Payoff Bias

2. “Selective Labels”
   - What do we do with people algorithm releases that judge jails?
   - (Like people who get surgery and didn’t before)
Notes: The figure above shows the results of an algorithm built using 221,876 observations in our NYC training set, applied to the 110,938 observations in our test set (see Figure 1). We report the observed judge's release rate (y-axis) against both the algorithm's predicted crime risk for each observation and the observed crime rate (observed only for those defendants in the test set released by the judges) for 1,000 bins sorted by predicted risk. The coloring shows share observations in each bin with a prior failure to appear. The bottom and back panels show the projection of the figure onto the two dimensional \{predicted crime risk, observed crime rate\} space and the \{predicted crime risk, judge release rate\} space.
Selective Labels Revisted

• What is the crime rate we must use?
  – For released defendants, empirical crime rate
  – For jailed ones, imputed crime rate
• But imputation may be biased…
  – Judge sees factors we don’t
  – Suppose young people have dots on their foreheads
    • Perfectly predictive: judge releases only if no dot
  – In released sample: young people have no crime
    • We would falsely conclude young people have no risk.
    • But this is because the young people with dots are in jail.
  – We would then falsely presume when we release all young people we will do better than judge
• Key problem: unobserved factors seen by judge affect crime rate (judge uses these wisely)
• How to fix?
Is not problem when we look just at released

Key insight: Contraction

Would judges *knowingly* release at 55% risk?

Willing to live with very high crime rates?

Or

Judges mispredicting
Contraction

- Multiple judges with similar caseloads and different lenience

- Strategy: use most lenient judges.
  - Take their released population and ask which of those would you incarcerate to become less lenient
  - Compare to less lenient judges
Contraction

• Requires
  – Judges have similar cases (random assignment)
  – Does not require judges having “similar” rankings

• But does give performance of a different rule
  – “Human constrained” release rule
Contraction and Imputation Compared

![Graph showing the comparison between Ground Truth, Identification through Contraction, and Imputed Crime Rate against Release Rate.](image)
Selective Labels

• In this case does not appear to be a problem

• But generically a problem
  – Extremely common problem – occurs whenever prediction -> decision -> treatment
  – Data generated by previous decisions
Figure 3: Who is Jailed as Judges Become More Stringent?

Notes: The bar in the middle of the figure shows who the most lenient quintile judges jail (top) and release (bottom) in our NYC dataset. The color shading shows the algorithm’s predicted crime risk. The bar at the right shows how the 2nd most lenient quintile judges implicitly select their marginal detainees to get from the most lenient quintile’s release rate down to their own release rate. The arrows show where within the risk distribution of the lenient quintile’s released set the stricter judges are selecting the marginal detainees. The bar at the left shows how the algorithm would select marginal detainees to achieve the same reduction in release rate.
Figure 4: Who do Stricter Judges Jail? Predicted Risk of Marginal Defendants

Notes: This figure shows where each of the quintiles of stricter judges in NYC select their marginal defendants (relative to the most lenient quintile), compared to how the algorithm would select marginal detainees. Within each panel, we divide the sample up into 20 bins by predicted crime risk (shown on the x-axis). The black segment at the top of each bar shows the share of each bin the most lenient quintile judges jail. In the top right-hand panel, we show which defendants the second-most-lenient quintile judges implicitly select to jail so get from the most lenient judge’s release rate down to their own lower release rate (blue), and who they continue to release (white). The left-hand top panel shows whom the algorithm would select instead. Each of the remaining rows shows the same comparison between the judge and algorithm decisions for the other less-lenient judge quintiles.
Figure 8: Simulation of Crime Rate - Release Tradeoff Algorithm Allows

Notes: The curve in the top panel shows the crime rate and release rate combinations that would be possible in NYC if judges were given a risk tool that could re-rank all defendants by their predicted crime risk and recommend them for detention in order of risk. Since we would like a crime rate that can be meaningfully compared across release rates, the y-axis shows the ratio of crimes committed by released defendants to the total number of defendants, not just the number released. The curve shows what gains would be possible relative to actual current judge decisions, assuming perfect compliance with the new tool. The curve in the bottom panel shows the risk level of the marginal person detained at each possible release rate under the algorithmic release rule.
Econometrics of Prediction Policy
Problems

1. Omitted Payoff Bias
2. Selective Labels
3. Restricted Inputs
Restricted Inputs

• Race and gender are not legal to use
  – We do not use them

• But is that enough?
  – Reconstruction problem
  – Optimizing in presence of this additional reconstruction constraint

• Rethinking disparate impact and disparate treatment
### Racist Algorithms?

<table>
<thead>
<tr>
<th>Release Rule</th>
<th>Crime Rate</th>
<th>Drop Relative to Judge</th>
<th>Percentage of Jail Population</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>%</td>
<td>Black</td>
</tr>
<tr>
<td>Distribution of Defendants (Base Rate)</td>
<td>.1134</td>
<td>0%</td>
<td>.4877</td>
</tr>
<tr>
<td>Judge</td>
<td>.1134 (.0010)</td>
<td>0%</td>
<td>.573</td>
</tr>
<tr>
<td>Algorithm</td>
<td>.0854 (.0008)</td>
<td>-24.68%</td>
<td>.5984</td>
</tr>
<tr>
<td>Usual Ranking</td>
<td>.0854 (.0008)</td>
<td>-24.64%</td>
<td>.573</td>
</tr>
<tr>
<td>Match Judge on Race</td>
<td>.0855 (.0008)</td>
<td>-24.64%</td>
<td>.5984</td>
</tr>
<tr>
<td>Equal Release Rates for all Races</td>
<td>.0873 (.0008)</td>
<td>-23.02%</td>
<td>.4877</td>
</tr>
<tr>
<td>Match Lower of Base Rate or Judge</td>
<td>.0876 (.0008)</td>
<td>-22.74%</td>
<td>.4877</td>
</tr>
</tbody>
</table>

**Notes:** Table reports the potential gains of the algorithmic release rule relative to the judge at the judge’s release rate with respect to crime reductions and share of the jail population that is black, Hispanic or either black or Hispanic. The first row shows the share of the defendant population overall that is black or Hispanic. The second row shows the results of the observed judge decisions. The third row shows the results of the usual algorithmic re-ranking release rule, which does not use race in predicting defendant risk and makes no post-prediction adjustments to account for race. In the fourth row we adjust the algorithm’s ranking of defendants for detention to ensure that the share of the jail population that is black and Hispanic under the algorithmic release rule are no higher than those under current judge decisions. The next row constraints the algorithmic release rule’s jail population to have no higher share black or Hispanic than that of the general defendant pool, while the final row constrains the algorithm’s jail population to have no higher share black or Hispanic than either the judge decisions or the overall defendant pool.
Econometrics of Prediction Policy

Problems

1. Omitted Payoff Bias
2. Selective Labels
3. Restricted Inputs
4. Response to Decision Rule
Comparing Judges to Themselves

Figure 13: Effect of Detaining Defendants Judges Usually Detain

Notes: This figure compares the change in crime rates and release rates that could be achieved by jailing additional defendants using the algorithm’s predicted crime risk compared to the decisions of stricter judges. The right-most point in the graph represents the release rate of the most lenient quintile of judges, with the crime rate that results. The light dashed line shows the decline in crime (as a percentage of the lenient quintile’s crime rate, shown on the y-axis) that results from randomly selecting additional defendants to detain from within the lenient quintile’s released cases, with the change in release rate relative to the lenient quintile shown on the x-axis. The red curve shows the crime rate / release rate tradeoff that comes from jailing additional defendants within the lenient quintile’s released set in descending order of the algorithm’s predicted crime risk. The additional curve on the graph shows the crime rate / release rate outcomes we would get from jailing additional defendants within the lenient quintile judges’ caseloads in descending order of an algorithm’s predicted probability that the judges jail a given defendant. The four points on the graph show the crime rate / release rate outcomes that are observed for the actual decisions made by the second through fifth most lenient quintile judges, who see similar caseloads on average to those of the most lenient quintile judges.
Why do we beat judges?

- Judges see more than we do
- Perhaps that is the problem

<table>
<thead>
<tr>
<th>Correctly used</th>
<th>Misused</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured</td>
<td>Offence history</td>
</tr>
<tr>
<td>Unmeasured</td>
<td>Private info</td>
</tr>
</tbody>
</table>

- Suggests behavioral economics of salience important here
  - In general, any kind of “noise”
General points here

• Need more ways of comparing human and machine predictions

• Notion of private information called into question
Summary

• Many prediction policy problems

• Raise their own econometric challenges

• Can also provide conceptual insights
Causal Inference for Average Treatment Effects

Professor Susan Athey
Stanford University
Machine Learning and Causal Inference

Spring 2017
The potential outcomes framework

For a set of i.i.d. subjects $i = 1, ..., n$, we observe a tuple $(X_i, Y_i, W_i)$, comprised of

- A feature vector $X_i \in \mathbb{R}^p$,
- A response $Y_i \in \mathbb{R}$, and
- A treatment assignment $W_i \in \{0, 1\}$.

Following the potential outcomes framework (Holland, 1986, Imbens and Rubin, 2015, Rosenbaum and Rubin, 1983, Rubin, 1974), we posit the existence of quantities $Y_i^{(0)}$ and $Y_i^{(1)}$.

- These correspond to the response we would have measured given that the $i$-th subject received treatment ($W_i = 1$) or no treatment ($W_i = 0$).
- **NB:** We only get to see $Y_i = Y_i^{(W_i)}$.
The potential outcomes framework

For a set of i.i.d. subjects $i = 1, \ldots, n$, we observe a tuple $(X_i, Y_i, W_i)$, comprised of

- A feature vector $X_i \in \mathbb{R}^p$,
- A response $Y_i \in \mathbb{R}$, and
- A treatment assignment $W_i \in \{0, 1\}$.

Define the average treatment effect (ATE), the average treatment effect on the treated (ATT)

$$ \tau = \tau^{ATE} = \mathbb{E} \left[ Y^{(1)} - Y^{(0)} \right]; \ \tau^{ATT} = \mathbb{E} \left[ Y^{(1)} - Y^{(0)} \mid W_i = 1 \right]; $$

and, the conditional average treatment effect (CATE)

$$ \tau(x) = \mathbb{E} \left[ Y^{(1)} - Y^{(0)} \mid X = x \right]. $$
The potential outcomes framework

My stomach hurts

Maybe it was the bad tuna I had for lunch

Too bad I don't have a proper control group, now I'll never know
The potential outcomes framework

If we make no further assumptions, it is not possible to estimate ATE, ATT, CATE, and related quantities.

- This is a failure of identification (infinite sample size), not a small sample issue. Unobserved confounders correlated with both the treatment and the outcome make it impossible to separate correlation from causality.

- One way out is to assume that we have measured enough features to achieve unconfoundedness (Rosenbaum and Rubin, 1983)

\[
\{ Y_i^{(0)}, Y_i^{(1)} \} \perp \perp W_i \mid X_i.
\]

- When this assumption + OVERLAP \((e(x) \in (0,1))\) holds, causal effects are identified and can be estimated.
Identification

\[
\mathbb{E}_{Y_i(1)}[Y_i^{(1)}] = \mathbb{E}_{X_i}[\mathbb{E}_{Y_i(1)|X_i}[Y_i^{(1)}|X_i]] \\
= \mathbb{E}_{X_i}[\mathbb{E}_{Y_i(1)|X_i} \left( \frac{Y_i^{(1)} \cdot W_i}{Pr(W_i = 1|X_i)} \right)|X_i] \\
= \mathbb{E}_{X_i}[\mathbb{E}_{Y_i|X_i} \left( \frac{Y_i \cdot W_i}{Pr(W_i = 1|X_i)} \right)|X_i] \\
= \mathbb{E}_{Y_i} \left( \frac{Y_i \cdot W_i}{Pr(W_i = 1|X_i)} \right)
\]

- Argument is analogous for \( \mathbb{E} [Y^0] \), which leads to ATE; and similar arguments allow you to identify CATE as well as the counterfactual effect of any policy assigning units to treatments on the basis of covariates.

- This result suggests a natural estimator: propensity score weighting using the sample analog of the last equation.
The role of overlap

Note that we need $e(x) \in (0, 1)$ to be able to calculate treatment effects for all $x$.

▶ Intuitively, how could you possibly infer $[Y(0)|X_i = x]$ if $e(x) = 1$?

▶ Note that for discrete $x$, the variance of ATE is infinite when $e(x) = 0$.

▶ “Moving the goalposts”: Crump, Hotz, Imbens, Miller (2009) analyze trimming, which entails dropping observations where $e(x)$ is too extreme. Typical approaches entail dropping bottom and top 5% or 10%.

▶ Approaches that don’t directly require propensity score weighting may seem to avoid the need for this, but important to understand role of extrapolation.
The causal inference literature has developed a variety of conventions, broadly referred to as “supplementary analysis,” for assessing credibility of empirical studies. One of the most prevalent conventions is to plot the propensity scores of treated and control groups to assess overlap.

- Idea: for each $q \in (0, 1)$, plot the fraction of observations in the treatment group with $e(x) = q$, and likewise for the control group.

- Even if there is overlap, when there are large imbalances, this is a sign that it may be difficult to get an accurate estimate of the treatment effect.
Example: Athey, Levin and Seira analysis of timber.

- Assignment to first price or open ascending:
  - in ID, randomized for subset of tracts with different probabilities in different geographies;
  - in CA, small v. large sales (with cutoffs varying by geography).

- So $W = 1$ if auction is sealed, and $X$ represents geography, size and year.
Propensity Score Plots: Assessing Overlap in ID

Very few observations with extreme propensity scores
Propensity Score Plots: Assessing Overlap in CA

Untrimmed v. trimmed so that $e(x) \in [.025, .975]$
Suppose small number of realizations of $X_i$. Under unconfoundedness, can analyze these as separate experiments and average up the results. How does conditioning on $X_i$ affect variance of estimator?
Variance of Estimator: Discrete Case

Let $\hat{E}$ denote the sample average, $\nabla$ be the variance, $\pi(x)$ be the proportion of observations with $X_i = x$, and let $e(x)$ be the propensity score ($Pr(W_i = 1|X_i = x)$).

$$\nabla(\hat{E}_{i: X_i = x, W_i = 1}(Y_i)) = \frac{\sigma^2(x)}{n \cdot \pi(x) \cdot e(x)}$$

$$\nabla(\hat{\tau}(x)) = \frac{\sigma^2(x)}{n \cdot \pi(x) \cdot e(x)} + \frac{\sigma^2(x)}{n \cdot \pi(x) \cdot (1 - e(x))}.$$

$$\nabla(\hat{ATE}) = \sum_x \left[ \frac{n(x)}{n} \cdot \frac{\sigma^2(x)}{n(x) \cdot e(x)} + \frac{\sigma^2(x)}{n(x) \cdot (1 - e(x))} \right].$$

$$= \sum_x \frac{\sigma^2(x)}{n} \left[ \frac{1}{e(x)} + \frac{1}{(1 - e(x))} \right].$$
Estimation Methods

The following methods are efficient when the number of covariates is fixed:

- Propensity score weighting
- “Direct” model of the outcome (model of $\mathbb{E} [Y_i | X_i, W_i]$), e.g. using regression
- Propensity-score weighted regression of $Y$ on $X, W$ (doubly robust)

The choice among these methods is widely studied:

- Other popular methods include matching, propensity score matching, propensity score blocking, which are not efficient but often do better in practice.
- Note: Hirano, Imbens, Ridder (2003) establish that more efficient to weight by estimated propensity score than actual.
Regression Case

Suppose that conditional mean function is given by

$$\mu(w, x) = \beta(w) \cdot x.$$  

If we estimate using OLS, then we can estimate the ATE as

$$\widehat{ATE} = \bar{X} \cdot (\hat{\beta}^{(1)} - \hat{\beta}^{(0)})$$

Note that OLS is unbiased and efficient, so the above quantity converges to the true values at rate $\sqrt{n}$:

$$\bar{X} \cdot (\hat{\beta}^{(1)} - \hat{\beta}^{(0)}) - \mu_x \cdot (\beta^{(1)} - \beta^{(0)}) = O_p\left(\frac{1}{\sqrt{n}}\right)$$
Obvious possibility: substitute in the lasso (or ridge, or elastic net) for OLS. But bias is a big problem.
With lasso, for each component $j$:

$$\hat{\beta}_j^{(w)} - \beta_j^{(w)} = O_p\left(\sqrt{\frac{\log(p)}{n}}\right)$$

This adds up across all dimensions, so that we can only guarantee for the ATT:

$$\hat{\text{ATT}} - \text{ATT} = O_p\left(\sqrt{\frac{\log(p)}{n}} \|\bar{X}_1 - \bar{X}_0\|_\infty \cdot \|\beta^{(0)}\|_0\right)$$
Imposing Sparsity: LASSO Crash Course

Assume linear model, and that there are at most a fixed number $k$ of non-zero coefficients: $\|\beta\|_0 \leq k$.

Suppose $X$ satisfies a “restricted eigenvalue” condition: no small group of variables is nearly collinear.

$$\|\hat{\beta} - \beta\|_2 = O_p\left(\sqrt{\frac{k \cdot \log(p)}{n}}\right)$$

$$\|\hat{\beta} - \beta\|_1 = O_p\left(k \sqrt{\frac{\log(p)}{n}}\right)$$

With the “de-biased lasso” (post-LASSO OLS) we can even build confidence intervals on $\hat{\beta}$ if $k << \frac{\sqrt{n}}{\log(p)}$.

Applying to the ATT, where we need to estimate $\bar{X}_1 \cdot \beta^{(0)}$ (the cf outcome for treated observations had they been control instead):

$$\hat{\text{ATT}} - \text{ATT} = O_p\left(\frac{k \|\bar{X}_1 - \bar{X}_0\|_\infty \sqrt{\log(p)}}{n}\right)$$
Improving the Properties of ATE Estimation in High Dimensions: A “Double-Selection” Method

Belloni, Chernozukov, and Hansen (2013) observe that causal inference is not an off-the-shelf prediction problem: confounders might be important if they have a large effect on outcomes OR a large effect on treatment assignment. They propose:

- Run LASSO of $W$ on $X$. Select variables with non-zero coefficients at a selected $\lambda$ (e.g. cross-validation).
- Run a LASSO of $Y$ on $X$. Select variables with non-zero coefficients at a selected $\lambda$ (may be different than first $\lambda$).
- Run a OLS of $Y$ on $W$ and the union of selected variables. (Not as good at purely predicting $Y$ as using only second set.)

Result: under “approximate sparsity” of BOTH propensity and outcome models, and constant treatment effects, estimated ATE is asymptotically normal and estimation is efficient.

Intuition: with enough data, can find the variables relevant for bias. With approximate sparsity and constant treatment effect, there aren’t too many, and OLS will be unbiased.
Single v. Double Selection in BCH Algorithm

Distributions of Studentized Estimators

post-single-selection estimator

post-double-selection estimator
Belloni, Chernozukov, Fernandez-Val and Hansen (2016) (http://arxiv.org/abs/1311.2645, forthcoming *Econometrica*) have a variety of generalizations:

- Applies general approach to IV
- Allows for a continuum of outcome variables
- Observes that nuisance parameters can be estimated generally using ML methods without affecting the convergence rate, subject to orthogonality conditions
- Shows how to use a framework based on orthogonality in moment conditions
Doubly Robust Methods

With small data, a “doubly robust” estimator (though not the typical one, where typically people use inverse propensity score weighted regression) is (with \( \hat{\gamma}_i = \frac{1}{\hat{e}(X_i)} \)):

\[
\hat{\mu}_1^0 = \bar{X}_1 \cdot \hat{\beta}^{(0)} + \mathbb{E}_{i:W_i=0} \hat{\gamma}_i (Y_i - X_i \hat{\beta}^{(0)})
\]

To see why, note that the term in parentheses goes to 0 if we estimate \( \beta^{(0)} \) well, while to show that we get the right answer if we estimate the propensity score well, we rearrange the expression to be

\[
\hat{\mu}_1^0 = (\bar{X}_1 - \mathbb{E}_{i:W_i=0} (\hat{\gamma}_i X_i)) \hat{\beta}^{(0)} + \mathbb{E}_{i:W_i=0} \hat{\gamma}_i Y_i
\]

The first term has expectation 0, and the second term gives the relevant counterfactual, if the propensity score is well-estimated.
Doubly Robust Methods: A High-Dimensional Analog?

\[
\hat{\mu}_1^0 = \bar{X}_1 \cdot \hat{\beta}^{(0)} + \hat{E}_{i:W_i=0}\hat{\gamma}_i(Y_i - X_i\hat{\beta}^{(0)})
\]

How does this relate to the truth?

\[
\hat{\mu}_1^0 - \mu_1^0 = \bar{X}_1 \cdot (\hat{\beta}^{(0)} - \beta^{(0)}) + \hat{E}_{i:W_i=0}\hat{\gamma}_i(\epsilon_i + X_i\beta^{(0)} - X_i\hat{\beta}^{(0)})
= (\bar{X}_1 - \hat{\gamma}'\bar{X}_0) \cdot (\hat{\beta}^{(0)} - \beta^{(0)}) + \hat{E}_{i:W_i=0}\hat{\gamma}_i\epsilon_i
\]

With high dimensions, we could try to estimate \( \hat{\beta} \) and the propensity score with LASSO or post-LASSO rather than OLS. However, this may not be good enough. It is also not clear how to get good estimates of the inverse propensity score weights \( \gamma_i \), in particular if we don’t want to assume that the propensity model is sparse (e.g. if the treatment assignment is a complicated function of confounders).
Small data approach (a la Robinson’s 1988) analyzed a semi-parametric model

- Model $Y_i = \tau W_i + g(X_i) + \epsilon_i$
- Goal: estimate $\tau$
- Approach: residuals on residuals gives $\sqrt{n}$-consistent and asymptotically normal estimator
- Regress $Y_i - \hat{g}(X_i)$ on $W_i - \mathbb{E}[W_i|X_i]$
Double Machine Learning

- Chernozhukov et al (2017):
  - Model $Y_i = \tau W_i + g(X_i) + \epsilon_i$, $\mathbb{E}[W_i|X_i] = h(X_i)$
  - Goal: estimate $\tau$
  - Use a modern machine learning method like random forests to estimate the “nuisance parameters”
  - Regress $Y_i - \hat{g}(X_i)$ on $W_i - \mathbb{E}[W_i|X_i]$
  - If ML method converges at the rate $n^{1/4}$, residuals on residuals gives $\sqrt{n}$-consistent and asymptotically normal estimator
Comparing Straight Regression to Double ML

- Moments used in estimation:
  - Regression: \( \mathbb{E}[(Y_i - W_i \tau - g(X_i)) \cdot W_i] = 0 \)
  - Double ML:
    \( \mathbb{E}[((Y_i - \hat{g}(X_i) - (W_i - \hat{h}(X_i)))\tau) \cdot (W_i - \hat{h}(X_i))] = 0 \)

- Double robustness and orthogonality: Robinson’s result implies that if \( \hat{g}(X_i) \) is consistent, then \( \hat{\tau} \) is the regression coefficient of the residual on residual regression, and even if \( \hat{h} \) is wrong, the orthogonality of the residual of the outcome regression and the residual \( W_i - \hat{h}(X_i) \) still holds.

- Neyman orthogonality: the Double ML moment condition has the property that when evaluated at \( \hat{g} = g \) and \( \hat{h} = h \), small changes in either of them do not change the moment condition. The moment condition is minimized at the truth.

- You are robust to small mistakes in estimation of nuisance parameters, unlike regression approach.
Comparing Straight Regression to Double ML

Application to Ghana Data (Duflo et al, 2017) with 2000 controls

- Study effect of secondary education.
- Try to recover experimental estimates from observational/non-experimental data using **2,000** controls.

**Returns To Secondary School Completion for Males**

<table>
<thead>
<tr>
<th>Outcome</th>
<th>Experimental</th>
<th>Observ.: OLS (5 controls)</th>
<th>Observ.: DML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standardized Score</td>
<td>0.502</td>
<td>0.595</td>
<td>0.486</td>
</tr>
<tr>
<td></td>
<td>(0.205)</td>
<td>(0.069)</td>
<td>(0.066)</td>
</tr>
<tr>
<td>Wage Worker</td>
<td>0.057</td>
<td>0.091</td>
<td><strong>0.082</strong></td>
</tr>
<tr>
<td></td>
<td>(0.109)</td>
<td>(0.036)</td>
<td>(0.037)</td>
</tr>
<tr>
<td>Log Earnings</td>
<td>-0.195</td>
<td><strong>-0.094</strong></td>
<td>-0.064</td>
</tr>
<tr>
<td></td>
<td>(0.245)</td>
<td>(0.087)</td>
<td>(0.088)</td>
</tr>
<tr>
<td>Partner pregnant</td>
<td>-0.089</td>
<td>-0.167</td>
<td><strong>-0.120</strong></td>
</tr>
<tr>
<td></td>
<td>(0.093)</td>
<td>(0.032)</td>
<td>(0.030)</td>
</tr>
</tbody>
</table>
An Efficient Approach with Non-Sparse Propensity

The solution proposed in Athey, Imbens and Wager (2016) for attacking the gap

\[ \hat{\mu}_1^0 - \mu_1^0 = (\bar{X}_1 - \hat{\gamma}'\bar{X}_0) \cdot (\hat{\beta}^{(0)} - \beta^{(0)}) + \hat{E}_{i:W_i=0}\hat{\gamma}_i\epsilon_i \]

is to bound 1st term by selecting \( \gamma_i \)'s using brute force. In particular:

\[ \hat{\gamma} = \arg\min_\gamma \zeta \cdot \|\bar{X}_1 - \gamma'\bar{X}_0\|_\infty + (1 - \zeta)\|\gamma\|_2^2 \]

The parameter \( \zeta \) is a tuning parameter; the paper shows that \( \zeta \) exists such that the \( \gamma \)'s exist to tightly bound the first term above.

With overlap, we can make \( \|\bar{X}_1 - \gamma'\bar{X}_0\|_\infty \) be \( O(\sqrt{\frac{\log(p)}{n}}) \).

**Result:** If the outcome model is sparse, estimate \( \beta \) using LASSO yielding bias of second term \( O_p\left( k\sqrt{\frac{\log(p)}{n}} \right) \), so the bias term is \( O(k\frac{\log(p)}{n}) \), so for \( k \) small enough, the last term involving \( \hat{\gamma}_i\epsilon_i \) dominates, and ATE estimator is \( O\left( \frac{1}{\sqrt{n}} \right) \).
Why Approximately Balancing Beats Propensity Weighting

One question is why the balancing weights perform better than the propensity score weights. To gain intuition, suppose the propensity score has the following logistic form,

\[ e(x) = \frac{\exp(x \cdot \theta)}{1 + \exp(x \cdot \theta)}. \]

After normalization, the inverse propensity score weights satisfy

\[ \gamma_i \propto \exp(x \cdot \theta). \]

The efficient estimator for \( \theta \) is the maximum likelihood estimator,

\[ \hat{\theta}_{ml} = \arg \max_{\theta} \sum_{i=1}^{n} \left\{ W_i X_i \cdot \theta - \ln(1 + \exp(X_i \cdot \theta)) \right\}. \]

An alternative is the method of moments estimator \( \hat{\theta}_{mm} \) that balances the covariates exactly:

\[ \bar{X}_0 = \sum_{\{i: W_i = 0\}} X_i \frac{\exp(X_i \cdot \theta)}{\sum_{\{j: W_j = 0\}} \exp(X_j \cdot \theta)}. \]
An alternative is the method of moments estimator $\hat{\theta}_{mm}$ that balances the covariates exactly:

$$\bar{X}_0 = \sum_{\{i: W_i = 0\}} X_i \frac{\exp(X_i \cdot \theta)}{\sum_{\{j: W_j = 0\}} \exp(X_j \cdot \theta)}.$$

with implied weights $\gamma_i \propto \exp(X_i \cdot \hat{\theta}_{mm})$.

- The only difference between the two sets of weights is that the parameter estimates $\hat{\theta}$ differ.
- The estimator $\hat{\theta}_{mm}$ leads to weights that achieve exact balance on the covariates, in contrast to either the true value $\theta$, or the maximum likelihood estimator $\hat{\theta}_{ml}$.
- The goal of balancing (leading to $\hat{\theta}_{mm}$) is different from the goal of estimating the propensity score (for which $\hat{\theta}_{ml}$ is optimal).
Summarizing the Approximate Residual Balancing Method of Athey, Imbens, Wager (2016)

- Estimate lasso (or elastic net) of $Y$ on $X$ in control group.
- Find “approximately balancing” weights that make the control group look like the treatment group in terms of covariates, while attending to the sum of squares of the weights. With many covariates, balance is not exact.
- Adjust the lasso prediction of the counterfactual outcome for the treatment group (if it had been control) using approximately balancing weights to take a weighted average of the residuals from the lasso model.

Main result: if the model relating outcomes to covariates is sparse, and there is overlap, then this procedure achieves the semi-parametric efficiency bound. No other method is known to do this for non-sparse propensity models. Simulations show that it performs much better than alternatives when propensity is not sparse.
Simulation Experiment

The design $X$ is “clustered.” We study the following settings for $\beta$:

Dense: $\beta \propto (1, 1/\sqrt{2}, ..., 1/\sqrt{p})$,

Harmonic: $\beta \propto (1/10, 1/11, ..., 1/(p + 9))$,

Moderately sparse: $\beta \propto (10, ..., 10, 1, ..., 1, 0, ..., 0)$,

Very sparse: $\beta \propto (1, ..., 1, 0, ..., 0)$. 

$p$
Simulation results, with $n = 300$ and $p = 800$. Approximate residual balancing estimates $\hat{\beta}$ using the elastic net. Inverse propensity residual weighting is like our method, except with $\gamma_i = 1/\hat{e}(X_i)$. We report root-mean-squared error for $\tau$.

**Observation:** Weighting regression residuals works better than weighting the original data; balanced weighting works better inverse-propensity weighting.
## Simulation Experiment

<table>
<thead>
<tr>
<th>$n$</th>
<th>$p$</th>
<th>$\beta_j \propto 1 {j \leq 10}$</th>
<th>$\beta_j \propto 1/j^2$</th>
<th>$\beta_j \propto 1/j$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\eta = 0.25$</td>
<td>$\eta = 0.1$</td>
<td>$\eta = 0.25$</td>
</tr>
<tr>
<td>200</td>
<td>400</td>
<td>0.90</td>
<td>0.84</td>
<td>0.94</td>
</tr>
<tr>
<td>200</td>
<td>800</td>
<td>0.86</td>
<td>0.76</td>
<td>0.92</td>
</tr>
<tr>
<td>200</td>
<td>1600</td>
<td>0.84</td>
<td>0.74</td>
<td>0.93</td>
</tr>
<tr>
<td>400</td>
<td>400</td>
<td>0.94</td>
<td>0.90</td>
<td>0.97</td>
</tr>
<tr>
<td>400</td>
<td>800</td>
<td>0.93</td>
<td>0.91</td>
<td>0.95</td>
</tr>
<tr>
<td>400</td>
<td>1600</td>
<td>0.93</td>
<td>0.88</td>
<td>0.94</td>
</tr>
<tr>
<td>800</td>
<td>400</td>
<td>0.96</td>
<td>0.95</td>
<td>0.98</td>
</tr>
<tr>
<td>800</td>
<td>800</td>
<td>0.96</td>
<td>0.94</td>
<td>0.97</td>
</tr>
<tr>
<td>800</td>
<td>1600</td>
<td>0.95</td>
<td>0.92</td>
<td>0.97</td>
</tr>
</tbody>
</table>

We report coverage of $\tau$ for 95% confidence intervals constructed by approximate residual balancing.
We are in a misspecified linear model; the “main effects” model is 10-sparse and linear.
Simulation Experiment

<table>
<thead>
<tr>
<th></th>
<th>( n )</th>
<th>( p )</th>
<th>100</th>
<th>200</th>
<th>400</th>
<th>800</th>
<th>1600</th>
<th>100</th>
<th>200</th>
<th>400</th>
<th>800</th>
<th>1600</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive</td>
<td>1.72</td>
<td>1.73</td>
<td>1.73</td>
<td>1.72</td>
<td>1.74</td>
<td></td>
<td></td>
<td>1.71</td>
<td>1.70</td>
<td>1.72</td>
<td>1.70</td>
<td>1.72</td>
</tr>
<tr>
<td>Elastic Net</td>
<td>0.44</td>
<td>0.46</td>
<td>0.50</td>
<td>0.51</td>
<td>0.54</td>
<td></td>
<td></td>
<td>0.37</td>
<td>0.39</td>
<td>0.39</td>
<td>0.40</td>
<td>0.42</td>
</tr>
<tr>
<td>Approx. Balance</td>
<td>0.48</td>
<td>0.55</td>
<td>0.61</td>
<td>0.63</td>
<td>0.70</td>
<td></td>
<td></td>
<td>0.24</td>
<td>0.30</td>
<td>0.38</td>
<td>0.40</td>
<td>0.45</td>
</tr>
<tr>
<td>Approx. Resid. Balance</td>
<td>0.24</td>
<td>0.26</td>
<td>0.28</td>
<td>0.29</td>
<td>0.32</td>
<td>0.16</td>
<td>0.17</td>
<td>0.18</td>
<td>0.19</td>
<td>0.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inverse Prop. Weight</td>
<td>1.04</td>
<td>1.07</td>
<td>1.11</td>
<td>1.13</td>
<td>1.18</td>
<td>0.82</td>
<td>0.84</td>
<td>0.88</td>
<td>0.89</td>
<td>0.94</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inv. Prop. Resid. Weight</td>
<td>1.29</td>
<td>1.30</td>
<td>1.31</td>
<td>1.31</td>
<td>1.33</td>
<td>1.25</td>
<td>1.25</td>
<td>1.26</td>
<td>1.25</td>
<td>1.28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Double-Select + OLS</td>
<td>0.28</td>
<td>0.29</td>
<td>0.31</td>
<td>0.31</td>
<td>0.34</td>
<td>0.24</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.26</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Approximate residual balancing estimates \( \hat{\beta} \) using the elastic net. Inverse propensity residual weighting is like our method, except with \( \gamma_i = 1/\hat{e}(X_i) \). We report root-mean-squared error for \( \tau_1 \).
Estimating the Effect of a Welfare-to-Work Program

Data from the California GAIN Program, as in Hotz et al. (2006).

- Program separately randomized in: Riverside, Alameda, Los Angeles, San Diego.
- Outcome: mean earnings over next 3 years.
- We hide county information. Seek to compensate with $p = 93$ controls.
- Full dataset has $n = 19170$. 

![Graph showing coverage and residual balance comparison for different correction methods.]

- Oracle
- Approx. Resid. Balance
- Double Select + OLS
- Lasso Resid. IPW
- No Correction
What are the pros and cons of approximate residual balancing vs. inverse-propensity residual weighting?

**Pros of balancing:**
- Works under weaker assumptions (only overlap).
- Algorithmic transparency.
- ...

**Pros of propensity methods:**
- Potential for double robustness.
- Potential for efficiency under heteroskedasticity.
- Generalizations beyond linearity.
- ...

Closing Thoughts
An Introduction to Regression Trees (CART)

Susan Athey, Stanford University
Machine Learning and Causal Inference
What is the goal of prediction?

- **Machine learning answer:**
  - Smallest mean-squared error in a test set

- **Formally:**
  - Let $S^{te}$ be a test set.
    - Think of this as a random draw of individuals from a population
  - Let $\hat{\mu}(x_i)$ be a candidate (estimated) predictor
  - MSE on test set is:
    $$\frac{1}{|S^{te}|} \sum_{i \in S^{te}} (Y_i - \hat{\mu}(X_i))^2$$
Regression Trees

- Simple method for prediction
  - Partition data into subsets by covariates
  - Predict using average within each subset

- Why are regression trees popular?
  - Easy to understand and explain
  - Businesses often need “segments”
  - Software assigns different algorithms to different segments

- Can completely describe the algorithm and interpretation
Example: Who survived the Titanic?
Regression Trees for Prediction

Data
- Outcomes $Y_i$, attributes $X_i$.
- Support of $X_i$ is $\mathcal{X}$.
- Have training sample with independent obs.
- Want to predict on new sample.

Build a “tree”:
- Partition of $\mathcal{X}$ into “leaves” $\mathcal{X}_j$.
- Predict $Y$ conditional on realization of $X$ in each region $\mathcal{X}_j$ using the sample mean in that region.
- Go through variables and leaves and decide whether and where to split leaves (creating a finer partition) using in-sample goodness of fit criterion.
- Select tree complexity using cross-validation based on prediction quality.
Outcome: Binary ($Y \in \{0, 1\}$)
Two covariates
Goal: Predict $Y$ as a function of $X$
“Classify” units as a function of $X$ according to whether they are more likely to have $Y=0$ or $Y=1$
Regression Trees for Prediction

(1) Tree-building: Use algorithm to partition data according to covariates (adaptive: do this based on the difference in mean outcomes in different potential leaves.)

(II) Estimation/prediction: calculate mean outcomes in each leaf

(III) Use cross-validation to select tree complexity penalty
Tree Building Details

- Impossible to search over all possible partitions, so use a greedy algorithm
- Do until all leaves have less than 2*minsize obs:
  - For each leaf:
    - For each observed value $\bar{x}_j$ of each covariate $x_j$:
      - Consider splitting the leaf into two children according to whether $\bar{x}_j \leq x_j$
      - Make new predictions in each candidate child according to sample mean
      - Calculate the improvement in “fit” (MSE)
    - Select the covariate $j$ and the cutoff value that lead to the greatest improvement in MSE; split the leaf into two child leaves

Observations
- In-sample MSE always improves with additional splits
- What is MSE when each leaf has one observation?
Problem: Tree has been “over-fitted”

- Suppose we fit a tree and pick a particular leaf $\ell$.
  - Do we expect that if we drew a new sample, we would get the same answer?

- More formally:
  - Let $S^{tr}$ be training dataset and $S^{te}$ be an independent test set
  - Let $\hat{\mu}(x_i) = \frac{1}{N_{\ell(x_i)}^{str}} \sum_{i \in \ell(x_i), S^{tr}} Y_i$
  - Is $E_{i \in S^{te}} [Y_i | X_i \in \ell(x_i)] = \hat{\mu}(x_i)$?
What are tradeoffs in tree depth?

- First: note that in-sample MSE doesn’t guide you
  - It always increases with depth

- Tradeoff as you grow tree deeper
  - More personalized predictions
  - More biased estimates
Regression Trees for Prediction: Components

1. Model and Estimation
   A. Model type: Tree structure
   B. Estimator $\hat{Y}_i$: sample mean of $Y_i$ within leaf
   C. Set of candidate estimators $C$: correspond to different specifications of how tree is split

2. Criterion function (for fixed tuning parameter $\lambda$)
   A. In-sample Goodness-of-fit function:
      \[
      Q^{is} = -\text{MSE} = -\frac{1}{N} \sum_{i=1}^{N} (\hat{Y}_i - Y_i)^2
      \]
   A. Structure and use of criterion
      i. Criterion: $Q^{crit} = Q^{is} - \lambda \times \#\text{leaves}$
      ii. Select member of set of candidate estimators that maximizes $Q^{crit}$, given $\lambda$

3. Cross-validation approach
   A. Approach: Cross-validation on grid of tuning parameters. Select tuning parameter $\lambda$ with highest Out-of-sample Goodness-of-Fit $Q^{os}$.
   B. Out-of-sample Goodness-of-fit function: $Q^{os} = -\text{MSE}$
How Does Cross Validation Work?

Tuning Set = 1/5 of Training Set
Cross-Validation Mechanics

- Loop over cross-validation samples
  - Train a deep tree on CV-training subset

- Loop over penalty parameters $\lambda$
  - Loop over cross-validation samples
    - Prune the tree according to penalty
    - Calculate new MSE of tree
  - Average (over c-v samples) the MSE for this penalty

- Choose the penalty $\lambda^*$ that gives the best average MSE
Choosing the penalty parameter

![Graph showing the relationship between size of tree and cross-validation relative error for different cp values.](image-url)
Some example code

## Regression tree:
## rpart(formula = linear, data = processed.scaled.train, method = "anova",
##     y = TRUE, control = rpart.control(cp = 1e-04, minsplit = 30))
##
## Variables actually used in tree construction:
## [1] bach_orhigher    city
## [3] employ_20to64    g2000
## [5] g2002             hh_size
## [7] highschool       median_age
## [9] median_income    noise1
## [15] noise2           noise3
## [17] noise4           noise5
## [19] noise6           noise7
## [21] noise8           noise9
## [27] percent_black    percent_hispanicorlatino
## [29] percent_male    percent_white
## [31] sex              totalpopulation_estimate
## [33] W                yob
## Root node error: $3866.8/18000 = 0.21482$
##
## n= 18000
##
##| CP | nsplit | rel error | xerror  | xstd   |
##|----|--------|-----------|---------|--------|
##| 1  | 0.01831622 | 0          | 1.00000 | 1.00020 | 0.0060337 |
##| 2  | 0.01200939   | 1          | 0.98168 | 0.98201 | 0.0061607 |
##| 3  | 0.00903665   | 2          | 0.96967 | 0.97013 | 0.0061355 |
##| 4  | 0.00555973   | 3          | 0.96064 | 0.96125 | 0.0062722 |
##| 5  | 0.00296112   | 4          | 0.95508 | 0.95571 | 0.0061583 |
##| 6  | 0.00274262   | 5          | 0.95212 | 0.95495 | 0.0062149 |
##| 7  | 0.00267924   | 6          | 0.94937 | 0.95394 | 0.0062370 |
##| 8  | 0.00190289   | 7          | 0.94670 | 0.95150 | 0.0062622 |
##| 9  | 0.00183424   | 8          | 0.94479 | 0.95162 | 0.0063299 |
##| 10 | 0.00181651   | 9          | 0.94296 | 0.95154 | 0.0063322 |
##| 44 | 0.00066122   | 64         | 0.89338 | 0.98640 | 0.0074692 |
##| 45 | 0.00064984   | 67         | 0.89135 | 0.99433 | 0.0076063 |
##| 46 | 0.00064533   | 68         | 0.89070 | 0.99997 | 0.0077120 |
##| 47 | 0.00063905   | 71         | 0.88876 | 1.00373 | 0.0077753 |
##| 48 | 0.00063765   | 72         | 0.88813 | 1.00493 | 0.0078130 |
##| 49 | 0.00063654   | 78         | 0.88429 | 1.00529 | 0.0078222 |
##| 50 | 0.00063212   | 85         | 0.87957 | 1.00727 | 0.0078509 |
##| 51 | 0.00063205   | 86         | 0.87893 | 1.00815 | 0.0078690 |
##| 52 | 0.00062566   | 94         | 0.87385 | 1.00952 | 0.0078949 |
##| 53 | 0.00062404   | 96         | 0.87260 | 1.01128 | 0.0079362 |
##| 54 | 0.00062352   | 99         | 0.87073 | 1.01200 | 0.0079494 |
##| 55 | 0.00061992   | 102        | 0.86886 | 1.01396 | 0.0079794 |
##| 56 | 0.00061970   | 103        | 0.86824 | 1.01481 | 0.0079986 |
##| 57 | 0.00061887   | 105        | 0.86700 | 1.01494 | 0.0080002 |
##| 58 | 0.00061518   | 112        | 0.86228 | 1.01661 | 0.0080294 |
Pruning Code

```r
op.index <- which.min(linear.singletree$cptable[, "xerror"])
cp.vals <- linear.singletree$cptable[, "CP"]
treepruned.linearsingle <- prune(linear.singletree, cp = cp.vals[op.index])
```
A Basic Policy Problem

- Every transfer program in the world must determine...
  - Who is eligible for the transfer
- Typical goal of redistributive programs
  - Transfer to neediest
- But identifying the neediest is easier said than done

Thanks to Sendhil Mullainathan for providing this worked out example....
## Typical Poverty Scorecard

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Value</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. How many members does the household have?</td>
<td>A. Five or more</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B. Four</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C. Three</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D. Two</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td></td>
<td>E. One</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2. Do any household members ages 5 to 18 go to private school or private</td>
<td>A. No</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>pre-school?</td>
<td>B. Yes</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C. No members ages 5 to 18</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>3. How many years of schooling has the female head/spouse completed?</td>
<td>A. Three or less</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B. Four to eleven</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C. Twelve or more</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D. No female head/spouse</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>4. How many household members work as employees with a written contract,</td>
<td>A. None</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>as civil servants for the government, or in the military?</td>
<td>B. One</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C. Two or more</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>5. In their main occupation, how many household members are managers,</td>
<td>A. None</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>administrators, professionals in the arts and sciences, mid-level</td>
<td>B. One</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>technicians, or clerks?</td>
<td>C. Two or more</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>6. How many rooms does the residence have?</td>
<td>A. One to four</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B. Five</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C. Six</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D. Seven</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>E. Eight or more</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>7. How does the household dispose of sewage?</td>
<td>A. Ditch, other, or no bathroom</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B. Simple hole, or directly into river, lake, or</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ocean</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C. Septic tank not connected to public sewage/rainwater system</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D. Septic tank connected to public sewage/rainwater system</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>E. Direct connection to public sewage/rainwater system</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>8. Does the household have a refrigerator?</td>
<td>A. No</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B. Yes, with one door</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C. Yes, with two doors</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>9. Does the household have a washing machine?</td>
<td>A. No</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B. Yes</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>10. Does the household have a cellular or land-line telephone?</td>
<td>A. None</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B. Cellular but not land-line</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C. Land-line but not cellular</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D. Both</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>PPI Score</td>
<td>Total Below the $2.50/Day/2005 PPP Line</td>
<td>Total Above the $2.50/Day/2005 PPP Line</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------------------</td>
<td>-----------------------------------------</td>
<td></td>
</tr>
<tr>
<td>0-4</td>
<td>81.8%</td>
<td>18.2%</td>
<td></td>
</tr>
<tr>
<td>5-9</td>
<td>77.8%</td>
<td>22.2%</td>
<td></td>
</tr>
<tr>
<td>10-14</td>
<td>66.1%</td>
<td>33.9%</td>
<td></td>
</tr>
<tr>
<td>15-19</td>
<td>49.0%</td>
<td>51.0%</td>
<td></td>
</tr>
<tr>
<td>20-24</td>
<td>37.2%</td>
<td>62.8%</td>
<td></td>
</tr>
<tr>
<td>25-29</td>
<td>23.9%</td>
<td>76.1%</td>
<td></td>
</tr>
<tr>
<td>30-34</td>
<td>15.4%</td>
<td>84.6%</td>
<td></td>
</tr>
<tr>
<td>35-39</td>
<td>8.6%</td>
<td>91.4%</td>
<td></td>
</tr>
<tr>
<td>40-44</td>
<td>5.2%</td>
<td>94.8%</td>
<td></td>
</tr>
<tr>
<td>45-49</td>
<td>3.2%</td>
<td>96.8%</td>
<td></td>
</tr>
<tr>
<td>50-54</td>
<td>2.1%</td>
<td>97.9%</td>
<td></td>
</tr>
<tr>
<td>55-59</td>
<td>1.2%</td>
<td>98.8%</td>
<td></td>
</tr>
<tr>
<td>60-64</td>
<td>1.2%</td>
<td>98.8%</td>
<td></td>
</tr>
<tr>
<td>65-69</td>
<td>0.4%</td>
<td>99.6%</td>
<td></td>
</tr>
<tr>
<td>70-74</td>
<td>0.6%</td>
<td>99.4%</td>
<td></td>
</tr>
<tr>
<td>75-79</td>
<td>0.0%</td>
<td>100.0%</td>
<td></td>
</tr>
<tr>
<td>80-84</td>
<td>0.0%</td>
<td>100.0%</td>
<td></td>
</tr>
<tr>
<td>85-89</td>
<td>0.0%</td>
<td>100.0%</td>
<td></td>
</tr>
<tr>
<td>90-94</td>
<td>0.0%</td>
<td>100.0%</td>
<td></td>
</tr>
<tr>
<td>95-100</td>
<td>0.0%</td>
<td>100.0%</td>
<td></td>
</tr>
</tbody>
</table>
Can we do better?

- This component of targeting is a pure prediction problem
- We fundamentally care about getting best predictive accuracy
- Let’s use this example to illustrate the mechanics of prediction
Brazilian Data

- The data:
  - 44,787 data points
  - 53 variables
  - Not very wide?
- Median
  - Annual consumption (in dollars): 3918
  - 348.85 monthly income
- 6 percent below 1.90 poverty line
- 14 percent below the 3.10 poverty line
Consumption
log (consumption)
$2,246

50\textsuperscript{th} Percentile
25^{th} Percentile

$1,377
$997
$\log_{10}\text{consumption} = 9.12 + 0.175 \cdot \text{rooms} - 0.227 \cdot \text{residents}$
log_consumption = 9.12 + 0.175 \cdot \text{rooms} - 0.227 \cdot \text{residents}
Two Variable Tree
\[
\text{log\_consumption} = 9.12 + 0.175 \cdot \text{rooms} + -0.227 \cdot \text{residents}
\]
28,573 data points to Fit with

<table>
<thead>
<tr>
<th>1</th>
<th>8061</th>
<th>8061</th>
<th>8061</th>
<th>8061</th>
<th>8061</th>
<th>8061</th>
</tr>
</thead>
</table>

Set of Trees

Fit trees on 4/5 of the data
Fit a tree for every level of split size
28,573 data points to Fit with

REPEAT leaving each fold out
Overfit Dominates
Why are we tuning on number of splits?
Questions and Observations

- How do we choose hold-out set size?
- How to choose the # of folds?
- What to tune on? (regularizer)
What are these standard errors?
Questions and Observations

- How do we choose hold-out set size?
- How to choose the # of folds?
- What to tune on? (regularizer)
- Which tuning parameter to choose from cross-validation?
Tuning Parameter Choice

- Minimum?

- One standard error “rule” (rule of thumb)
  - Which direction?
Output

- Which of these many trees do we output?

- Even after choosing lambda we have as many trees as folds...

- Estimate one tree on full data using chosen cut size

- Key point: Cross validation is just for choosing tuning parameter
  - Just for deciding how complex a model to choose
Questions and Observations

- How do we choose hold-out set size?
- How to choose the # of folds?
- What to tune on? (regularizer)
- Which tuning parameter to choose from cross-validation?
- Is there a problem tuning on subsets and then outputting fitted value on full set?
Let's look at the predictions. Notice something?
WHY?
What does the tree look like?
What else can we look at to get a sense of what the predictions are?
Variable Importance

Empirical loss by noising up $x$ minus Empirical loss

- number_of_bathrooms
- number_of_rooms
- wall_type
- electric_water_heater
- number_of_residents
- garbage_disposal
- sewage_type
- mail_distribution_service
- street_pavement

importance
How to describe model

- Large discussion of “interpretability”
  - Will return to this

- But one implication is that the prediction function itself becomes a new $y$ variable to analyze.

- Is any of this stable? What would a confidence interval look like?
Questions and Observations

- How do we choose hold-out set size?
- How to choose the # of folds?
- What to tune on? (regularizer)
- Which tuning parameter to choose from cross-validation?

- Is there a problem tuning on subsets and then outputting fitted value on full set?
- What is stable/robust about the estimated function?
# Measuring Performance

<table>
<thead>
<tr>
<th>Predicted condition</th>
<th>Total population</th>
<th>Predicted Condition positive</th>
<th>Predicted Condition negative</th>
<th>Prevalence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>True positive</td>
<td>False Negative (Type II error)</td>
<td>( \frac{\text{( \Sigma ) Condition positive}}{\text{( \Sigma ) Total population}} )</td>
</tr>
<tr>
<td>True condition</td>
<td></td>
<td>True positive</td>
<td>False Negative (Type II error)</td>
<td>( \frac{\text{( \Sigma ) True positive}}{\text{( \Sigma ) Condition positive}} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>False Positive (Type I error)</td>
<td>True negative</td>
<td>( \frac{\text{( \Sigma ) False positive}}{\text{( \Sigma ) Condition negative}} )</td>
</tr>
<tr>
<td>False condition</td>
<td></td>
<td>False Positive (Type I error)</td>
<td>True negative</td>
<td>( \frac{\text{( \Sigma ) True negative}}{\text{( \Sigma ) Condition negative}} )</td>
</tr>
</tbody>
</table>

**Accuracy (ACC):**\( \frac{\text{\( \Sigma \) True positive}}{\text{\( \Sigma \) Total population}} \)

**Precision:**\( \frac{\text{\( \Sigma \) True positive}}{\text{\( \Sigma \) Test outcome positive}} \)

**False discovery rate (FDR):**\( \frac{\text{\( \Sigma \) False positive}}{\text{\( \Sigma \) Test outcome positive}} \)

**False omission rate (FOR):**\( \frac{\text{\( \Sigma \) False negative}}{\text{\( \Sigma \) Test outcome negative}} \)

**Positive predictive value (FPV):**\( \frac{\text{\( \Sigma \) True positive}}{\text{\( \Sigma \) Test outcome positive}} \)

**Negative predictive value (NPV):**\( \frac{\text{\( \Sigma \) True negative}}{\text{\( \Sigma \) Test outcome negative}} \)

**False positive rate (FPR), Fall-out:**\( \frac{\text{\( \Sigma \) False positive}}{\text{\( \Sigma \) Condition positive}} \)

**True negative rate (TNR), Specificity (SPC):**\( \frac{\text{\( \Sigma \) True negative}}{\text{\( \Sigma \) Condition negative}} \)

**True positive rate (TPR), Sensitivity, Recall:**\( \frac{\text{\( \Sigma \) True positive}}{\text{\( \Sigma \) Condition positive}} \)

**False negative rate (FNR), Miss rate:**\( \frac{\text{\( \Sigma \) False negative}}{\text{\( \Sigma \) Condition positive}} \)

**Positive likelihood ratio (LR+):**\( \frac{\text{TPR}}{\text{FPR}} \)

**Diagnostic odds ratio (DOR):**\( \frac{\text{LR+}}{\text{LR-}} \)

**Negative likelihood ratio (LR-):**\( \frac{\text{FNR}}{\text{TNR}} \)
**TPR = TP/P = TP/(TP + FN)**

**SPC = TN/N = TN/(TN + FP)**
Measuring Performance

- Area Under Curve: Typical measure of performance

- What do you think of this measure?
What fraction of the poor do we reach?
Measuring Performance

- AUC: Typical measure of performance

- What do you think of this measure?

- Getting the domain specific meaningful performance measure
  - Magnitudes
  - Need point of comparison
What fraction of the poor do we reach?

Confidence Intervals?
This is what we want from econometric theorems

- How do we choose hold-out set size?
- How to choose the # of folds?
- What to tune on? (regularizer)
- Which tuning parameter to choose from cross-validation?
- Is there a problem tuning on subsets and then outputting fitted value on full set?
- What is stable/robust about the estimated function?
- How do we form standard errors on performance?
Summary

- Regression trees easy to understand and interpret
- Tradeoff between personalized versus inaccurate predictions
- Cross-validation is a tool to figure out the best balance in a particular dataset
  - E.g. if truth is complex, may want to go deeper
- CART is ad hoc, but works well in practice
  - Loses to OLS/logit if true model is linear
  - Good at finding lots of complex interactions
Heterogeneous Treatment Effects and Parameter Estimation with Causal Forests and Gradient Forests

Susan Athey
Stanford University

Machine Learning and Econometrics

See Wager and Athey (forthcoming, JASA) and Athey, Tibshirani and Wager
https://arxiv.org/abs/1610.01271
Treatment Effect Heterogeneity

Heterogeneous Treatment Effects

- Insight about mechanisms
- Designing policies, selecting groups for application/eligibility
- Personalized policies

Literature: Many Covariates

- See Wager and Athey (2015) and Athey and Imbens (2016) for ML-based analyses and many references on treatment effect heterogeneity
- Imai and Ratkovic (2013) analyze treatment effect heterogeneity with LASSO
- Targeted ML (van der Laan, 2006) can be used as a semi-parametric approach to estimating treatment effect heterogeneity
ML Methods for Causal Inference: Treatment Effect Heterogeneity

- ML methods perform well in practice, but many do not have well established statistical properties (see Chen and White (1999) for early analysis of neural nets).
- Unlike prediction, ground truth for causal parameters not directly observed.
- Need valid confidence intervals for many applications (AB testing, drug trials); challenges include adaptive model selection and multiple testing.
- Different possible questions of interest, e.g.:
  - Identifying subgroups (Athey and Imbens, 2016)
  - Testing for heterogeneity across all covariates (List, Shaikh, and Xu, 2016)
  - Robustness to model specification (Athey and Imbens, 2015)
  - **Personalized estimates** (Wager and Athey, 2015; Taddy et al 2014; others)
The potential outcomes framework

For a set of i.i.d. subjects $i = 1, ..., n$, we observe a tuple $(X_i, Y_i, W_i)$, comprised of

- A feature vector $X_i \in \mathbb{R}^p$,
- A response $Y_i \in \mathbb{R}$, and
- A treatment assignment $W_i \in \{0, 1\}$.

Following the potential outcomes framework (Holland, 1986, Imbens and Rubin, 2015, Rosenbaum and Rubin, 1983, Rubin, 1974), we posit the existence of quantities $Y_i^{(0)}$ and $Y_i^{(1)}$.

- These correspond to the response we would have measured given that the $i$-th subject received treatment ($W_i = 1$) or no treatment ($W_i = 0$).
The potential outcomes framework

For a set of i.i.d. subjects $i = 1, \ldots, n$, we observe a tuple $(X_i, Y_i, W_i)$, comprised of

- A feature vector $X_i \in \mathbb{R}^p$,
- A response $Y_i \in \mathbb{R}$, and
- A treatment assignment $W_i \in \{0, 1\}$.

Goal is to estimate the conditional average treatment effect

$$
\tau (x) = \mathbb{E} \left[ Y^{(1)} - Y^{(0)} \mid X = x \right].
$$

**NB:** In experiments, we only get to see $Y_i = Y_i^{(W_i)}$. 
The potential outcomes framework

If we make no further assumptions, estimating $\tau(x)$ is not possible.

- Literature often assumes **unconfoundedness** (Rosenbaum and Rubin, 1983)
  \[
  \{ Y_i^{(0)}, Y_i^{(1)} \} \perp\!\!\!\!\perp W_i \mid X_i.
  \]

- When this assumption holds, methods based on matching or propensity score estimation are usually consistent.
Baseline method: \(k\)-NN matching

Consider the \(k\)-NN matching estimator for \(\tau(x)\):

\[
\hat{\tau}(x) = \frac{1}{k} \sum_{S_1(x)} Y_i - \frac{1}{k} \sum_{S_0(x)} Y_i,
\]

where \(S_{0/1}(x)\) is the set of \(k\)-nearest cases/controls to \(x\). This is consistent given unconfoundedness and regularity conditions.

- **Pro:** Transparent asymptotics and good, robust performance when \(p\) is small.
- **Con:** Acute curse of dimensionality, even when \(p = 20\) and \(n = 20k\).

**NB:** Kernels have similar qualitative issues as \(k\)-NN.
Adaptive nearest neighbor matching

**Random forests** are a popular heuristic for adaptive nearest neighbors estimation introduced by Breiman (2001).

- **Pro:** Excellent empirical track record.
- **Con:** Often used as a black box, without statistical discussion.

There has been considerable interest in using forest-like methods for treatment effect estimation, but without formal theory.

- Green and Kern (2012) and Hill (2011) have considered using Bayesian forest algorithms (BART, Chipman et al., 2010).
- Several authors have also studied related tree-based methods: Athey and Imbens (2016), Su et al. (2009), Taddy et al. (2014), Wang and Rudin (2015), Zeilis et al. (2008), ...

Wager and Athey (2015) provide the first formal results allowing random forest to be used for provably valid asymptotic inference.
Making \(k\)-NN matching adaptive

Athey and Imbens (2016) introduce causal tree: defines neighborhoods for matching based on recursive partitioning (Breiman, Friedman, Olshen, and Stone, 1984), advocate sample splitting (w/ modified splitting rule) to get assumption-free confidence intervals for treatment effects in each leaf.

Euclidean neighborhood, for \(k\)-NN matching.

Tree-based neighborhood.
Suppose we have a training set \( \{(X_i, Y_i, W_i)\}_{i=1}^n \), a test point \( x \), and a tree predictor

\[
\hat{\tau}(x) = T(x; \{(X_i, Y_i, W_i)\}_{i=1}^n).
\]

**Random forest idea:** build and average many different trees \( T^* \):

\[
\hat{\tau}(x) = \frac{1}{B} \sum_{b=1}^B T^*_b(x; \{(X_i, Y_i, W_i)\}_{i=1}^n).
\]
Suppose we have a training set $\{(X_i, Y_i, W_i)\}_{i=1}^n$, a test point $x$, and a tree predictor

$$\hat{\tau}(x) = T(x; \{(X_i, Y_i, W_i)\}_{i=1}^n).$$

**Random forest idea:** build and average many different trees $T^*$:

$$\hat{\tau}(x) = \frac{1}{B} \sum_{b=1}^B T_b^*(x; \{(X_i, Y_i, W_i)\}_{i=1}^n).$$

We turn $T$ into $T^*$ by:

- **Bagging / subsampling the training set** (Breiman, 1996); this helps smooth over discontinuities (Bühlmann and Yu, 2002).
- **Selecting the splitting variable** at each step from $m$ out of $p$ randomly drawn features (Amit and Geman, 1997).
Statistical inference with regression forests

**Honest trees** do not use the same data to select partition (splits) and make predictions. Ex: Split-sample trees, propensity trees.

**Theorem.** (Wager and Athey, 2015) Regression forests are asymptotically **Gaussian and centered**,

\[ \frac{\hat{\mu}_n(x) - \mu(x)}{\sigma_n(x)} \Rightarrow \mathcal{N}(0, 1), \quad \sigma^2_n(x) \to_p 0, \]

given the following assumptions (+ technical conditions):

1. **Honesty.** Individual trees are honest.

2. **Subsampling.** Individual trees are built on random subsamples of size \( s \asymp n^\beta \), where \( \beta_{\text{min}} < \beta < 1 \).

3. **Continuous features.** The features \( X_i \) have a density that is bounded away from 0 and \( \infty \).

4. **Lipschitz response.** The conditional mean function \( \mu(x) = \mathbb{E} [ Y \mid X = x ] \) is Lipschitz continuous.
Athey and Imbens (2016), Wager and Athey (2015) highlight the perils of adaptive estimation for confidence intervals, tradeoff between MSE and coverage for trees but not forests.
Proof idea

Use the shorthand $Z_i = (X_i, Y_i)$ for training examples.

- The regression forest prediction is
  \[
  \hat{\mu} := \hat{\mu}(Z_1, ..., Z_n).
  \]

- The Hájek projection of the regression forest is
  \[
  \hat{\mu} = \mathbb{E}[\hat{\mu}] + \sum_{i=1}^{n} \left( \mathbb{E}[\hat{\mu} | Z_i] - \mathbb{E}[\hat{\mu}] \right).
  \]

- Classical statistics (Hoeffding, Hájek) tells us that
  \[
  \text{Var} [\hat{\mu}] \leq \text{Var} [\hat{\mu}], \quad \text{and that} \quad \lim_{n \to \infty} \frac{\text{Var} [\hat{\mu}]}{\text{Var} [\hat{\mu}]} = 1
  \]
  implies asymptotic normality.
Proof idea

Now, let $\hat{\mu}_b^*(x)$ denote the estimate for $\hat{\mu}(x)$ given by a single regression tree, and let $\check{\mu}_b^*$ be its Hájek projection,

- Using the **adaptive nearest neighbors** framework of Lin and Jeon (2006), we show that

$$\frac{\text{Var} [\check{\mu}_b^*]}{\text{Var} [\hat{\mu}_b^*]} \gtrsim \log^{-p}(s).$$

- As a consequence of the **ANOVA decomposition** of Efron and Stein (1981), the full forest gets

$$\frac{\text{Var} [\hat{\mu}]}{\text{Var} [\hat{\mu}]} \geq 1 - \frac{s}{n} \frac{\text{Var} [\hat{\mu}_b^*]}{\text{Var} [\check{\mu}_b^*]},$$

thus yielding the **asymptotic normality** result for $s \asymp n^\beta$ for any $0 < \beta < 1$.

- For **centering**, we bound the bias by requiring $\beta > \beta_{\text{min}}$. 
Variance estimation for regression forests

We estimate the variance of the regression forest using the **infinitesimal jackknife for random forests** (Wager, Hastie, and Efron, 2014). For each of the $b = 1, \ldots, B$ trees comprising the forest, define

- The estimated response as $\hat{\mu}^*_b(x)$, and
- The number of times the $i$-th observation was used as $N^{*}_{bi}$.

Then, defining $\text{Cov}^*_*$ as the covariance taken with respect to all the trees comprising the forest, we set

$$\hat{\sigma}^2 = \frac{n-1}{n} \left( \frac{n}{n-s} \right)^2 \sum_{i=1}^{n} \text{Cov}^*_* [\hat{\mu}^*_b(x), N^{*}_{bi}]^2.$$ 

**Theorem.** (Wager and Athey, 2015) Given the same conditions as used for asymptotic normality, the infinitesimal jackknife for regression forests is consistent:

$$\hat{\sigma}^2_n(x) / \sigma^2_n(x) \to_p 1.$$
Causal forest example

We have \( n = 20k \) observations whose features are distributed as
\( X \sim U([-1, 1]^p) \) with \( p = 6 \); treatment assignment is random. All
the signal is concentrated along two features.

The plots below depict \( \hat{\tau}(x) \) for 10k random test examples,
projected into the 2 signal dimensions.

true effect \( \tau(x) \)  causal forest  \( k\)-NN estimate

available at github: susanathey/causalTree
Causal forest example

We have $n = 20k$ observations whose features are distributed as $X \sim U([-1, 1]^p)$ with $p = 20$; treatment assignment is random. All the signal is concentrated along two features.

The plots below depict $\hat{\tau}(x)$ for 10k random test examples, projected into the 2 signal dimensions.

Causal forest example

The causal forest dominates $k$-NN for both bias and variance. With $p = 20$, the relative mean-squared error (MSE) for $\tau$ is

$$\frac{\text{MSE for } k\text{-NN (tuned on test set)}}{\text{MSE for forest (heuristically tuned)}} = 19.2.$$ 

causal forest  \hspace{2cm} k$\text{-NN estimate$

For $p = 6$, the corresponding MSE ratio for $\tau$ is 2.2.
Application: General Social Survey

The General Social Survey is an extensive survey, collected since 1972, that seeks to measure demographics, political views, social attitudes, etc. of the U.S. population.

Of particular interest to us is a randomized experiment, for which we have data between 1986 and 2010.

▶ **Question A:** Are we spending too much, too little, or about the right amount on welfare?

▶ **Question B:** Are we spending too much, too little, or about the right amount on assistance to the poor?

**Treatment effect:** how much less likely are people to answer too much to question B than to question A.

▶ We want to understand how the treatment effect depends on covariates: political views, income, age, hours worked, ...

**NB:** This dataset has also been analyzed by Green and Kern (2012) using Bayesian additive regression trees (Chipman, George, and McCulloch, 2010).
Application: General Social Survey

A causal forest analysis uncovers strong treatment heterogeneity \((n = 28,686, p = 12)\).
Much recent literature bringing ML methods to causal inference focus on single binary treatment in environment with unconfoundedness.

Economic models often have more complex estimation approaches.

Athey, Tibshirani, and Wager (2016) tackle general GMM case:
  - Quantile regression
  - Instrumental Variables
  - Panel regression
  - Consumer choice
  - Euler equations
  - Survival analysis
Average Treatment Effects with IV or Unconfoundedness

- In a series of influential papers, Belloni, Chernozhukov, Hansen, et al generalized LASSO methods to average treatment effect estimation through instrumental variables models, unconfoundedness, and also moment-based methods.

- See also Athey, Imbens and Wager (2016) combine regularized regression and high-dimensional covariate balancing for average treatment effect estimation; and references therein on more recent papers on ATE estimation in high dimensions.
Forests for GMM Parameter Heterogeneity

- Local GMM/ML uses kernel weighting to estimate personalized model for each individual, weighting nearby observations more.
  - Problem: curse of dimensionality
- We propose forest methods to determine what dimensions matter for “nearby” metric, reducing curse of dimensionality.
  - Estimate model for each point using “forest-based” weights: the fraction of trees in which an observation appears in the same leaf as the target
- We derive splitting rules optimized for objective
- Computational trick:
  - Use approximation to gradient to construct pseudo-outcomes
  - Then apply a splitting rule inspired by regression trees to these pseudo-outcomes
Related Work

(Semi-parametric) local maximum likelihood/GMM

▶ Local ML (Hastie and Tibshirani, 1987) weights nearby observations; e.g. local linear regression. See Loader, C. (1999); also Hastie and Tibshirani (1990) on GAM; see also Newey (1994)

▶ Lewbel (2006) asymptotic prop of kernel-based local GMM

▶ Other approaches include Sieve: Chen (2007) reviews

Score-based test statistics for parameter heterogeneity

▶ Andrews (1993), Hansen (1992), and many others, e.g. structural breaks, using scores of estimating equations

▶ Zeiles et al (2008) apply this literature to split points, when estimating models in the leaves of a single tree.

Splitting rules

▶ CART: MSE of predictions for regression, Gini impurity for classification, survival (see Bouhamad et al (2011))

▶ Statistical tests, multiple testing corrections: Su et al (2009)

▶ Causal trees/forests: adaptive v. honest est. (Athey and Imbens, 2016); propensity forests (Wager and Athey, 2015)
Solving estimating equations with random forests

We have $i = 1, \ldots, n$ i.i.d. samples, each of which has an observable quantity $O_i$, and a set of auxiliary covariates $X_i$.

Examples:

- Non-parametric regression: $O_i = \{Y_i\}$.
- Treatment effect estimation: $O_i = \{Y_i, W_i\}$.
- Instrumental variables regression: $O_i = \{Y_i, W_i, Z_i\}$.

Our parameter of interest, $\theta(x)$, is characterized by an estimating equation:

$$\mathbb{E} \left[ \psi_{\theta(x), \nu(x)}(O_i) \mid X_i = x \right] = 0 \text{ for all } x \in \mathcal{X},$$

where $\nu(x)$ is an optional nuisance parameter.
The GMM Setup: Examples

Our parameter of interest, $\theta(x)$, is characterized by

$$\mathbb{E} \left[ \psi_{\theta(x)}(O_i) \mid X_i = x \right] = 0 \quad \text{for all} \quad x \in \mathcal{X},$$

where $\nu(x)$ is an optional nuisance parameter.

▶ **Quantile regression**, where $\theta(x) = F_x^{-1}(q)$ for $q \in (0, 1)$:

$$\psi_{\theta(x)}(Y_i) = q \mathbf{1}(\{Y_i > \theta(x)\}) - (1 - q) \mathbf{1}(\{Y_i \leq \theta(x)\})$$

▶ **IV regression**, with treatment assignment $W$ and instrument $Z$. We care about the treatment effect $\tau(x)$:

$$\psi_{\tau(x), \mu(x)} = \begin{pmatrix} Z_i(Y_i - W_i \tau(x) - \mu(x)) \\ Y_i - W_i \tau(x) - \mu(x) \end{pmatrix}.$$
Solving heterogeneous estimating equations

The classical approach is to rely on local solutions (Fan and Gijbels, 1996; Hastie and Tibshirani, 1990; Loader, 1999).

$$\sum_{i=1}^{n} \alpha(x; X_i) \psi_{\hat{\theta}(x), \hat{\nu}(x)} (O_i) = 0,$$

where the weights $\alpha(x; X_i)$ are obtained from, e.g., a kernel.

We use random forests to get good data-adaptive weights. Has potential to be help mitigate the curse of dimensionality.

- Building many trees with small leaves, then solving the estimating equation in each leaf, and finally averaging the results is a bad idea. Quantile and IV regression are badly biased in very small samples.
- Using RF as an “adaptive kernel” protects against this effect.
The random forest kernel

Forests induce a kernel via averaging tree-based neighborhoods. This idea was used by Meinshausen (2006) for quantile regression.
Solving estimating equations with random forests

We want to use an estimator of the form

\[ \sum_{i=1}^{n} \alpha(x; X_i) \psi(\hat{\theta}(x), \hat{\nu}(x)) (O_i) = 0, \]

where the weights \( \alpha(x; X_i) \) are from a random forest.

**Key Challenges:**

- How do we grow trees that yield an **expressive** yet **stable** neighborhood function \( \alpha(\cdot; X_i) \)?
- We do not have access to “**prediction error**” for \( \theta(x) \), so how should we **optimize splitting**?
- How should we account for **nuisance parameters**?
- Split evaluation rules need to be **computationally efficient**, as they will be run many times for each split in each tree.
Step #1: Conceptual motivation

Following CART (Breiman et al., 1984), we use greedy splits. Each split directly seeks to improve the fit as much as possible.

- For regression trees, in large samples, the best split is that which increases the heterogeneity of the predictions the most.
- The same fact also holds locally for estimating equations.

We split a parent node \( P \) into two children \( C_1 \) and \( C_2 \). In large samples and with no computational constraints, we would like to maximize

\[
\Delta (C_1, C_2) = n_{C_1} n_{C_2} (\hat{\theta}_{C_1} - \hat{\theta}_{C_2})^2,
\]

where \( \hat{\theta}_{C_1}, \hat{\theta}_{C_2} \) solve the estimating equation in the children.
Step #2: Practical realization

Computationally, solving the estimating equation in each possible child to get $\hat{\theta}_C$ and $\hat{\nu}_C$ can be prohibitively expensive.

To avoid this problem, we use a gradient-based approximation. The same idea underlies gradient boosting (Friedman, 2001).

$$\hat{\theta}_C \approx \tilde{\theta}_C := \hat{\theta}_P - \frac{1}{|\{i : X_i \in C\}|} \sum_{i : X_i \in C} \xi^\top A_P^{-1} \psi_{\hat{\theta}_P, \hat{\nu}_P} (O_i),$$

$$A_P = \frac{1}{|\{i : X_i \in P\}|} \sum_{i : X_i \in P} \nabla \psi_{\hat{\theta}_P, \hat{\nu}_P} (O_i),$$

where $\hat{\theta}_P$ and $\hat{\nu}_P$ are obtained by solving the estimating equation once in the parent node, and $\xi$ is a vector that picks out the $\theta$-coordinate from the $(\theta, \nu)$ vector.
Step #2: Practical realization

In practice, this idea leads to a **split-relabel** algorithm:

1. **Relabel step:** Start by computing pseudo-outcomes

   \[ \tilde{\theta}_i = -\xi^\top A_P^{-1} \psi_{\hat{\theta}_P, \hat{\nu}_P} (O_i) \in \mathbb{R}. \]

2. **Split step:** Apply a CART-style regression split to the \( \tilde{Y}_i \).

This procedure has several advantages, including the following:

- **Computationally,** the most demanding part of growing a tree is in scanning over all possible splits. Here, we reduce to a regression split that can be efficiently implemented.

- **Statistically,** we only have to solve the estimating equation once. This reduces the risk of hitting a numerically unstable leaf—which can be a risk with methods like IV.

- From an **engineering** perspective, we can write a single, optimized split-step algorithm, and then use it everywhere.
Step #3: Variance correction

Conceptually, we saw that—in large samples—we want splits that maximize the heterogeneity of the $\hat{\theta}(X_i)$. In small samples, we need to account for **sampling variance**.

We need to penalize for the following two sources of variance.

- **Our plug-in estimates** for the heterogeneity of $\hat{\theta}(X_i)$ will be **overly optimistic** about the large-sample parameter heterogeneity. We need to correct for this kind of over-fitting.

- We **anticipate “honest” estimation**, and want to avoid leaves where the **estimating equation is unstable**. For example, with IV regression, we want to avoid leaves with an unusually weak 1st-stage coefficient.

This is a generalization of the analysis of Athey and Imbens (2016) for treatment effect estimation.
Gradient forests

Our label-and-regress splitting rules can be used to grow an ensemble of trees that yield a forest kernel. We call the resulting procedure a **gradient forest**.

- Regression forests are a special case of gradient forests with a squared-error loss.

Available as an R-package, gradientForest, built on top of the ranger package for random forests (Wright and Ziegler, 2015).
Asymptotic normality of gradient forests

**Theorem.** (Athey, Tibshirani and Wager, 2016) Given regularity of both the estimating equation and the data-generating distribution, gradient forests are **consistent** and **asymptotically normal**:

\[
\frac{\hat{\theta}_n(x) - \theta(x)}{\sigma_n(x)} \Rightarrow N(0, 1), \quad \sigma_n^2 \to 0.
\]

**Proof sketch.**

- Influence functions: Hampel (1974); also parallels to use in Newey (1994).
- Influence function heuristic motivates approximating gradient forests with a class of regression forests.
- Analyze the approximating regression forests using Wager and Athey (2015)
- Use coupling result to derive conclusions about gradient forests.
Asymptotic normality of gradient forests: Proof details

- Influence function heuristic motivates approximating gradient forests with a class of regression forests. Start as if we knew true parameter value in calculating influence fn:
  - Let $\tilde{\theta}^*_i(x)$ denote the influence function of the $i$-th observation with respect to the true parameter value $\theta(x)$:
    \[
    \tilde{Y}^*_i(x) = -\xi^\top V(x)^{-1} \psi_{\theta(x), \nu(x)}(O_i)
    \]
  - Pseudo-forest predictions: $\tilde{\theta}^*(x) = \theta(x) + \sum_{i=1}^n \alpha_i \tilde{\theta}^*_i(x)$.

- Apply Wager and Athey (2015) to this. Key points: $\tilde{\theta}^*(x)$ is linear function, so we can write it as an average of tree predictions, with trees built on subsamples. Thus it is U-statistic; can use the ANOVA decomposition.

- Coupling result: conclusions about gradient forests.

Suppose that the gradient forest estimator $\hat{\theta}(x)$ is consistent for $\theta(x)$. Then $\hat{\theta}(x)$ and $\tilde{\theta}^*(x)$ are coupled,

\[
\tilde{\theta}^*(x) - \hat{\theta}(x) = o_P \left( \left\| \sum_{i=1}^n \alpha_i(x) \psi_{\theta(x), \nu(x)}(O_i) \right\|_2 \right). \tag{1}
\]
In quantile regression, we want to estimate the $q$-th quantile of the conditional distribution of $Y$ given $X$, namely $\theta(x) = F_x^{-1}(q)$.

- Meinshausen (2006) used the random forest kernel for quantile regression. However, he used standard CART regression splitting instead of a tailored splitting rule.
- In our split-relabel paradigm, quantile splits reduce to classification splits ($\hat{\theta}_P$ is the $q$-th quantile of the parent):
  \[ \tilde{Y}_i = 1\{Y_i > \hat{\theta}_P\}. \]
- To estimate many quantiles, we do multi-class classification.
The above examples show quantile estimates at \( q = 0.1, 0.5, 0.9 \), on Gaussian data with \( n = 2,000 \) and \( p = 40 \). The package `quantregForest` implements the method of Meinshausen (2006).
Simulation example: Instrumental variables

We want to estimate **heterogeneous treatment effects** with endogenous treatment assignment: $Y_i$ is the treatment, $W_i$ is the treatment assignment, and $Z_i$ is an instrument satisfying:

$$\{ Y_i(w) \}_{w \in W} \perp Z_i \mid X_i.$$  

- Our **split-relabel** formalism tells us to use pseudo-outcomes

  $$\tilde{\tau}_i = (Z_i - \bar{Z}_p) \left( (Y_i - \bar{Y}_p) - \hat{\tau}_P \left( W_i - \bar{W}_p \right) \right),$$

  where $\hat{\tau}_P$ is the IV solution in the parent, and $\bar{Y}_p$, $\bar{W}_p$, $\bar{Z}_p$ are averages over the parent.

- This is just IV regression residuals projected onto the instruments.
Simulation example: Instrumental variables

Using IV forests is important

We have spurious correlations:

- OLS for $Y$ on $W$ given $X$ has two jumps, at $X_1 = -1/3$ and at $X_1 = 1/3$.
- The causal effect $\tau(X)$ only has a jump at $X_1 = -1/3$.
- $n = 10,000$, $p = 20$.

The response function is

$$Y_i = (2W_i - 1) \mathbf{1}(\{X_{1,i} > -1/3\}) + (3A_i - 1.5) \mathbf{1}(\{X_{1,i} > 1/3\}) + 2\varepsilon_i.$$ 

$A_i$ is correlated with $W_i$. 

$\mathbf{1}(\cdot)$ denotes an indicator function.
Using IV splits is important

We have useless correlations:

- The joint distribution of $(W_i, Y_i)$ is independent of the covariates $X_i$.
- But: the causal effect $\tau(X)$ has a jump at $X_1 = 0$.
- $n = 5,000$, $p = 20$.

The response function is

$$Y_i = 2 \cdot 1\{X_{1,i} \leq 0\} A_i + 1\{X_{1,i} > 0\} W_i + 1(1 + 0.73 \cdot 1\{X_{1,i} > 0\}) \varepsilon_i.$$ 

$A_i$ is correlated with $W_i$. 

Empirical Application: Family Size

Angrist and Evans (1998) study the effect of family size on women’s labor market outcomes. Understanding heterogeneity can guide policy.

- Outcomes: participation, female income, hours worked, etc.
- Treatment: more than two kids
- Instrument: first two kids same sex
- First stage effect of same sex on more than two kids: .06
- Reduced form effect of same sex on probability of work, income: .008, $132
- LATE estimates of effect of kids on probability of work, income: .133, $2200
Treatment Effects: Magnitude of Decline

Effect on Participation  
Baseline Probability of Working

![Graph showing the effect on participation and baseline probability of working against father's income.](image-url)
Treatment Effects: Magnitude of Decline

Effect on Participation

Effect relative to Baseline
Treatment Effects: Magnitude of Decline

Effect on Earnings

Baseline Earnings

- Father's Income [$1,000/year]
- Mother's Baseline Income [$1,000/year]

Graphs showing the effect of age on earnings and baseline earnings.
Robustness

Professor Susan Athey
ML and Causal Inference
Robustness of Causal Estimates
Athey and Imbens (AER P&P, 2015)

• General nonlinear models/estimation methods
• Causal effect is defined as a function of model parameters
  • Simple case with binary treatment, effect is $\tau_i = Y_i(1) - Y_i(0)$
• Consider other variables/features as “attributes”
• Proposed metric for robustness:
  • Use a series of “tree” models to partition the sample by attributes
    • Simple case: take each attribute one by one
  • Re-estimate model within each partition
  • For each tree, calculate overall sample average effect as a weighted average of effects within each partition
  • This yields a set of sample average effects
  • Propose the standard deviation of effects as robustness measure
Robustness of Causal Estimates
Athey and Imbens (AER P&P, 2015)

• Four Applications:
  • Randomly assigned training program
  • Treated individuals with artificial control group from census data (Lalonde)
  • Lottery data (Imbens, Rubin & Sacerdote (2001))
  • Regression of earnings on education from NLSY

• Findings
  • Robustness measure better for randomized experiments, worse in observational studies
Lalonde data

<table>
<thead>
<tr>
<th>Variable</th>
<th>Exper $\hat{\theta}_B$</th>
<th>s.e.</th>
<th>Non-exper $\hat{\theta}_B$</th>
<th>s.e.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Model</td>
<td>1.67</td>
<td>(0.67)</td>
<td>1.07</td>
<td>(0.63)</td>
</tr>
<tr>
<td>$\hat{\sigma}_\theta$</td>
<td>[0.13]</td>
<td></td>
<td>[2.13]</td>
<td></td>
</tr>
<tr>
<td>Split on treatment</td>
<td>est. 1.58</td>
<td>22.8</td>
<td>est. -4.26</td>
<td>45.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>1.55</td>
<td>10.1</td>
<td>1.97</td>
<td>144.5</td>
</tr>
<tr>
<td>black</td>
<td>1.71</td>
<td>11.4</td>
<td>1.38</td>
<td>26.2</td>
</tr>
<tr>
<td>hispanic</td>
<td>1.61</td>
<td>7.2</td>
<td>1.54</td>
<td>59.7</td>
</tr>
<tr>
<td>married</td>
<td>1.87</td>
<td>11.0</td>
<td>1.06</td>
<td>10.4</td>
</tr>
<tr>
<td>education</td>
<td>1.77</td>
<td>14.7</td>
<td>1.25</td>
<td>74.5</td>
</tr>
<tr>
<td>nodegree</td>
<td>1.33</td>
<td>18.6</td>
<td>1.77</td>
<td>46.1</td>
</tr>
<tr>
<td>re74</td>
<td>1.64</td>
<td>11.2</td>
<td>0.58</td>
<td>71.0</td>
</tr>
<tr>
<td>re75</td>
<td>1.63</td>
<td>8.9</td>
<td>-0.88</td>
<td>94.1</td>
</tr>
<tr>
<td>u74</td>
<td>1.64</td>
<td>11.2</td>
<td>-2.44</td>
<td>88.6</td>
</tr>
<tr>
<td>u75</td>
<td>1.71</td>
<td>7.0</td>
<td>-0.83</td>
<td>81.9</td>
</tr>
</tbody>
</table>
Comparing Robustness

<table>
<thead>
<tr>
<th></th>
<th>Exper</th>
<th>Non-exp</th>
<th>IRS</th>
<th>NLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Est</td>
<td>1.67</td>
<td>1.07</td>
<td>-0.44</td>
<td>0.059</td>
</tr>
<tr>
<td>(s.e.)</td>
<td>(0.67)</td>
<td>(0.63)</td>
<td>(0.012)</td>
<td>(0.010)</td>
</tr>
<tr>
<td>$\sigma_\theta$</td>
<td>[0.13]</td>
<td>[2.13]</td>
<td>[0.10]</td>
<td>[0.004]</td>
</tr>
<tr>
<td>ratio</td>
<td>0.20</td>
<td>3.38</td>
<td>0.83</td>
<td>0.40</td>
</tr>
</tbody>
</table>
Robustness Metrics: Desiderata

• Invariant to:
  • Scaling of explanatory variables
  • Transformations of vector of explanatory variables
  • Adding irrelevant variables

• Each member model must be somehow distinct to create variance, yet we want to allow lots of interactions
  • Need to add lots of rich but different models

• Well-grounded way to weight models
  • This paper had equal weighting
Robustness Metrics: Work In Progress

Std Deviation versus Worst-Case

• Desire for set of alternative models that grows richer
  • New additions are similar to previous ones, lower std dev

• Standard dev metric:
  • Need to weight models to put more weight on distinct alternative models

• “Worst-case” or “bounds”:
  • Find the lowest and highest parameter estimates from a set of models
  • Ok to add more models that are similar to existing ones.
  • But worst-case is very sensitive to outliers—how do you rule out “bad” models?

Theoretical underpinnings

✦ Subjective versus objective uncertainty
  • Subjective uncertainty: correct model
  • Objective uncertainty: distribution of model estimates given correct model

✦ What are the preferences of the “decision-maker” who values robustness?
  • “Variational preferences”
    • “Worst-case” in set of possible beliefs, allow for a “cost” of beliefs that captures beliefs that are “less likely.” (see Strzalecki, 2011)

  • Our approach for exog. covariate case:
    • Convex cost to models that perform poorly out of sample from a predictive perspective

✦ Good model
  • Low obj. uncertainty: tightly estimated
  • Other models that predict outcomes well also yield similar parameter estimates
Policy Estimation

Susan Athey
Stanford University

See Athey and Wager (2017)
Efficient Policy Estimation

- Learning optimal policy assignment and estimating treatment effect heterogeneity closely related
- ML literature proposed variety of methods (Langford et al; Swaminathan and Joachims; in econometrics, Kitagawa and Tetenov, Hirano and Porter, Manski)
- Estimating the value of a personalized policy closely related to estimating average treatment effect (comparing treat all policy to treat none policy)
- Lots of econometric theory about how to estimate average treatment effects efficiently (achieve semi-parametric efficiency bound)
- Athey and Wager (2017): prove that bounds on regret (gap between optimal policy and estimated policy) can be tightened using an algorithm consistent with econometric theory
  - Theory provides guidance for algorithm choice
Setup and Approach

Setup

- Policy $\pi : \mathcal{X} \rightarrow \{\pm 1\}$
- Given a class of policies $\Pi$, the optimal policy $\pi^*$ and the regret $R(\pi)$ of any other policy are respectively defined as

  $$\pi^* = \arg\max_{\pi \in \Pi} \{ \mathbb{E} [Y_i(\pi(X_i))] \} \quad (1)$$

  $$R(\pi) = \mathbb{E} [Y_i(\pi^*(X_i))] - \mathbb{E} [Y_i(\pi(X_i))] . \quad (2)$$

- Goal: estimate a policy $\pi$ that minimizes regret $R(\pi)$.

Approach: Estimate $Q(\pi)$, choose policy to minimize $\hat{Q}(\pi)$:

  $$Q(\pi) = \mathbb{E} [Y_i(\pi(X_i))] - \frac{1}{2} \mathbb{E} [Y_i(-1) + Y_i(+1)] \quad (3)$$

  $$\hat{\pi} = \arg\max_{\pi \in \Pi} \{ \hat{Q}(\pi) \} , \quad (4)$$
Alternative Approaches

\[ Q(\pi) = \mathbb{E} [Y_i (\pi (X_i))] - \frac{1}{2} \mathbb{E} [Y_i(-1) + Y_i(+1)] \]  

Methods for estimating ATE or ATT can also be used to estimate the effect of any policy—simply define treatment as following policy, and control as the reverse.

Kitagawa and Tetenov (forthcoming, Econometrica):

- Estimate \( \hat{Q}(\pi) \) using inverse propensity weighting.
- Suppose that \( Y_i \leq M \) uniformly bounded, overlap satisfied with \( \eta \leq e(x) \leq 1 - \eta \), \( \Pi \) is Vapnik-Chervonenkis Class of dimension \( \text{VC}(\Pi) \)
  - Roughly, VC dimension is the maximum integer \( D \) such that some data set of cardinality \( D \) can be “shattered” by a policy in \( \Pi \). If tree is limited to \( D \) leaves, its VC dimension is \( D \).
- Then, \( \hat{\pi} \) satisfies the regret bound

\[ R (\hat{\pi}) = \mathcal{O}_P \left( \frac{M}{\eta} \sqrt{\frac{\text{VC}(\Pi)}{n}} \right). \]
Alternative Approaches

\[ Q(\pi) = \frac{1}{2} (\mathbb{E} [Y_i(\pi(X_i))] - \mathbb{E} [Y_i(-\pi(X_i))]) \] (7)

\[ \hat{Q}(\pi) = \frac{1}{n} \sum_{i=1}^{n} \pi(X_i) \hat{\Gamma}_i \] (8)

What is \( \hat{\Gamma}_i \)?

Zhao (2015) assume a randomized controlled trial and use
\( \hat{\Gamma}_i = W_i Y_i / \mathbb{P}[W_i = 1] \), while Kitagawa and Tetenov (forthcoming) uses inverse-propensity weighting
\( \hat{\Gamma}_i = W_i Y_i / \hat{e}_{W_i}(X_i) \). In an attempt to stabilize the weights, Beygelzimer et al (2009) introduce an “offset”

\[ \hat{\Gamma}_i = \frac{W_i}{\hat{e}_{W_i}(X_i)} \left( Y_i - \frac{\max \{ Y_i \} + \min \{ Y_i \}}{2} \right), \]

while Zhao et al (2015) go further and advocate

\[ \hat{\Gamma}_i = \frac{W_i}{\hat{e}_{W_i}(X_i)} \left( Y_i - \frac{\hat{\mu}_{+1}(X_i) + \hat{\mu}_{-1}(X_i)}{2} \right). \]
Alternative Approaches

Class of estimators:

\[
\hat{Q}(\pi) = \frac{1}{n} \sum_{i=1}^{n} \pi (X_i) \hat{\Gamma}_i \tag{9}
\]

What is \( \hat{\Gamma}_i \)? Athey/Wager:

\[
\hat{\Gamma}_i := \hat{\mu}_{+1}^{(-k(i))}(X_i) - \hat{\mu}_{-1}^{(-k(i))}(X_i) + W_i \frac{Y_i - \hat{\mu}_{W_i}^{(-k(i))}(X_i)}{\hat{e}_{W_i}^{(-k(i))}(X_i)}, \tag{10}
\]

where \( k(i) \in \{1, \ldots, K\} \) denotes the fold containing the \( i \)-th obs.
Optimizing $Q$ and Estimating Impact

Class of estimators:

$$\hat{Q}(\pi) = \frac{1}{n} \sum_{i=1}^{n} \pi(X_i) \hat{\Gamma}_i$$  \hspace{1cm} (11)

Optimize $\hat{Q}(\pi)$: use a classifier with labels $\text{sign}(\hat{\Gamma}_i)$ and weights $|\hat{\Gamma}_i|$. This step is treated as a constrained optimization problem, not estimation.

Issue: extending to multiple classes.

Evaluating benefit of policy: Re-estimate policy using leave one out only at second stage; or leave one out for whole routine. Assign using re-estimated policy. Do this at all 10 folds; use this overall assignment as a unified policy. Then, use ATE methods to estimate difference between estimated treatment effect for group assigned to treatment, and the estimated treatment effect for group assigned to control. That is overall estimated benefit of the policy.
Let $V(\pi)$ denote the semiparametrically efficient variance for estimating $Q(\pi)$.

Let $V_* := V(\pi^*)$ denote the semiparametrically efficient variance for evaluating $\pi^*$

Let $V_{\text{max}}$ denote a sharp bound on the worst case efficient variance $\sup_{\pi} V(\pi)$ for any policy $\pi$. Results:

Given policy class $\Pi$ with VC dimension $\text{VC}(\Pi)$, proposed learning rule yields policy $\hat{\pi}$ with regret bounded by

$$R(\hat{\pi}) = \mathcal{O}_P \left( \sqrt{V_* \log \left( \frac{V_{\text{max}}}{V_*} \right) \frac{\text{VC}(\Pi)}{n}} \right). \quad (12)$$

We also develop regret bounds for non-parametric policy classes $\Pi$ with a bounded entropy integral, such as finite-depth decision trees.
**Illustration**

![Heat map depicting average over 200 simulations.](image)

**Inverse-propensity weighting**  
**Double machine learning**

**Figure:** Results from two attempts at learning a policy $\pi$ by counterfactual risk minimization over depth-2 decision tree, with $\hat{Q}$-estimators obtained by inverse-propensity weighting and double machine learning. Dashed blue line denotes the optimal decision rule (treat in the upper-right corner, do not treat elsewhere); solid black lines denote learned policies $\pi$ (treat in the shaded regions, not elsewhere). Heat map depicts average over 200 simulations.
Conclusions

Contributions from causal inference and econometrics literature:

▶ Identification and estimation of causal effects
▶ Classical theory to yield asymptotically normal and centered confidence intervals
▶ Semiparametric efficiency theory

Contributions from ML:

▶ Practical, high performance algorithms for personalized prediction and policy estimation

Putting them together:

▶ Practical, high performance algorithms
▶ Causal effects with valid confidence intervals
▶ Consistent with insights from efficiency theory
Deep Learning and Neural Nets

American Economic Association,
Continuing Education Program 2018
Machine Learning and Econometrics, Lecture 7

Guido Imbens
Outline

1.
Scalar outcome $y_i$, $p$-dimensional vector of covariates/inputs/features $x_i$, $l$-th element equal to $x_{il}$.

We focus on two cases here:

- **(Regression)** Outcome $y_i$ is ordered, possibly continuous, e.g., expenditure, earnings.

- **(Classification)** Outcome $y_i$ is element of finite unordered set, e.g., choice of item.
I. In the regression problem we are interested in predicting the value of $y_i$, with a focus on out-of-sample prediction.

Given a data set, $(x_i, y_i)$, $i = 1, \ldots, N$, we split it into two parts, a training sample, say the first $N_{\text{train}}$ observations, and a test sample, the last $N_{\text{test}}$ observations, we are interested in using the training sample (but not the test sample) to choose a function $f : X \mapsto \mathbb{R}$ that minimizes the average squared error in the test sample:

$$
\frac{1}{N_{\text{test}}} \sum_{i=N_{\text{train}}+1}^{N} \left( y_i - f(x_i) \right)^2
$$
• From a conventional econometric/statistical perspective, if we think that the original sample is a random sample from a large population, so we can think of the \((x_i, y_i)\) as realizations of random variables \((X_i, Y_i)\), then the function that minimizes the objective is the conditional expectation of the outcome given the covariates/features:

\[
f(x) = \mathbb{E}[Y_i | X_i = x]
\]

In that case we can think of the problem as just using the training sample to get the optimal (in terms of expected squared error) estimator for this conditional expectation.
There are two conceptual points to note with regard to that perspective:

• We do not care separately about bias and variance, just about expected squared error. In the same vein, we do not necessarily care about inference.

• We do not need to think about where the sample came from (e.g., the infinite superpopulation). The problem of finding a good predictor is well-defined irrespective of that.
II. In the **classification** problem we are interested in classifying observations on the basis of their covariate/feature values $x_i$ in terms of the outcome $y_i$. Suppose $y_i \in \{1, \ldots, J\}$.

Again we wish to come up with a function $f : \mathbb{X} : \{1, \ldots, J\}$ that minimizes the mis-classification rate in the test sample:

$$
\frac{1}{N_{\text{test}}} \sum_{i=N_{\text{train}}+1}^{N} 1_{y_i \neq f(x_i)}
$$
Suppose again that the sample is a random sample from an infinite population, and that in the population the fraction of units with $Y_i = j$ in the subpopulation of units with $X_i = x$ is

$$p_j(x) = \text{pr}(Y_i|X_i = x)$$

In that case the solution is

$$f(x) = j \text{ iff } p_j(x) = \max_{k=1}^{J} p_k(x)$$

Note that we are not actually focused here on estimating $p_j(x)$, just the index of the choice with the maximum probability.

- Knowing $p_j(\cdot)$ solves the problem, but it is not necessary to estimate $p_j(\cdot)$. 
Consider the classical example of the classification problem, the zipcode problem:

Given an image of a digit, with the accompanying label (from zero to nine), classify it as a 0-9.

- Thinking of this as a problem of estimating a conditional probability is not necessarily natural: it is not clear what the conditional probabilities mean. Although, from a Bayesian perspective one might improve things by taking into account the other digits in a handwritten zipcode.
Suppose that $J = 2$, so we are in a setting with a binary outcome, where the regression and classification problem are the same in the standard econometric/statistics approach.

Suppose that the sample is a random sample from a large population, with the conditional probability that $Y_i = 2$ equal to $p_2(x) = 1 - p_1(x)$.

Viewing the problem as a regression problem the solution is

$$f(x) = 1 + p_2(x) \quad \text{with } f(x) \in [1, 2]$$

The solution to the classification problem is slightly different:

$$f(x) = 1 + 1_{p_2(x) > 1/2} \quad \text{with } f(x) \in \{1, 2\}$$
Back to the Regression Problem

Linear Model for $f(\cdot)$:

$$f(x) = \omega_0 + \sum_{j=1}^{p} \omega_j x_j$$

Minimize sum of squared deviations over “weights” (parameters) $\omega_j$:

$$\min_{\omega_0, \ldots, \omega_p} \sum_{i=1}^{N_{\text{train}}} \left( Y_i - \omega_0 - \sum_{j=1}^{p} \omega_j X_{ij} \right)^2$$

This is not very flexible, but has well-understood properties.
Let’s make this more flexible:

Single index model:

\[ f(x) = g \left( \sum_{j=1}^{p} \omega_j x_j \right) \]

Estimate both weights \( \omega_j \) and transformation \( g(\cdot) \)
Additive model:

\[ f(x) = \sum_{j=1}^{p} g_j(x_j) \]

Estimate the \( p \) covariate-specific transformations \( g_j(\cdot) \).

Advantage over nonparametric kernel regression is the improved rate of convergence (rate of convergence does not depend on number of covariates, whereas for kernel regression the rate of convergence slows down the larger the dimension of \( x_i, p, \) is).
Projection Pursuit (hybrid of single index model and additive models):

\[ f(x) = \sum_{l=1}^{L} g_l \left( \sum_{j=1}^{p} \omega_{lj} x_j \right) \]

Estimate both the \( L \) transformations \( g_l(\cdot) \) and the \( L \times p \) weights \( \omega_{lj} \).
Neural Net with single hidden layer:

\[ f(x) = \omega_0^{(2)} + \sum_{l=1}^{L} \omega_l^{(2)} g \left( \sum_{j=1}^{p} \omega_{lj}^{(1)} x_j \right) \]

Fix transformation \( g(\cdot) \) and estimate only the weights \( \omega_{lj}^{(1)} \) and \( \omega_l^{(2)} \) (but in both layers).

- Increased flexibility in one place, at the expense of flexibility in other places.

- Computational advantages especially when generalized to multiple hidden layers.
General neural net with $K$ hidden layers, one observed input layer, one observed output layer, $K+2$ layers total.

Need additional notation to keep things tractable. Superscripts in parentheses denote layers, from (1) to $(K+2)$.

$p_1 = p$ observed inputs (dimension of $x$). Define $\alpha_l^{(1)} = z_l^{(1)} = x_l$ for $l = 1, \ldots, p_1$.

First hidden layer: $p_2$ hidden elements $z_1^{(2)}, \ldots, z_{p_2}^{(2)}$

$$z_l^{(2)} = \omega_{l0}^{(1)} + \sum_{j=1}^{p_1} \omega_{lj}^{(1)} \alpha_j^{(1)} = \omega_{l0}^{(1)} + \sum_{j=1}^{p_1} \omega_{lj}^{(1)} x_j, \quad l = 1, \ldots, p_2$$

$$\alpha_l^{(2)} = g\left(z_l^{(2)}\right) \quad \text{(with transformation $g(\cdot)$ known)}$$
Hidden layer $k$, with $p_k$ hidden elements $z_1^{(k)}, \ldots, z_{p_k}^{(k)}$, for $k = 2, \ldots, K + 1$

$$z_l^{(k)} = \omega_{l0}^{(k-1)} + \sum_{j=1}^{p_k-1} \omega_{lj}^{(k-1)} \alpha_j^{(k-1)}, \quad \alpha_l^{(k)} = g\left(z_l^{(k)}\right)$$

Final output layer (layer $K + 2$) with $p_{K+2} = 1$:

$$z_1^{(K+2)} = \omega_{10}^{(K+1)} + \sum_{j=1}^{p_{K+1}} \omega_{1j}^{(K+1)} \alpha_j^{(K+1)}$$

$$f(x; \omega) = \alpha_1^{(K+2)} = g^{(K+2)}\left(z_1^{(K+2)}\right) = z_1^{(K+2)}$$
Choices for $g(\cdot)$ (pre-specified, \textbf{not} chosen by formal data-driven optimization):

1. sigmoid: $g(a) = (1 + \exp(-a))^{-1}$

2. tanh: $g(a) = (\exp(a) - \exp(-a))/ (\exp(a) + \exp(-a))$

3. rectified linear $g(a) = a \times 1_{a>0}$

4. leaky rectified linear $g(a) = a \times 1_{a>0} + \gamma \times 1_{a<0}$
Lost of complexity allowed for, but comes with lots of choices.

Not easy to use out-of-the-box, but very successful in complex settings.

Computationally tricky because of multi-modality.

- can approximate smooth functions accurately (universal approximator) with many layers and many hidden units.
Interpretation

We can think of the layers up to the last one as constructing regressors: \( z^{(K+1)} = h(\omega, x) \)

Alternative is to choose functions of regressors, e.g., polynomials \( z_{ij} = x_{i1} \times x_{i4} \times x_{i7}^2 \).

In what sense is this better? Is this a statement about the type of functions we encounter?
Multiple layers versus multiple hidden units

“We observe that shallow models [models with few layers] in this context overfit at around 20 millions parameters while deep ones can benefit from having over 60 million. This suggests that using a deep model expresses a useful preference over the space of functions the model can learn.” *Deep Learning*, Goodfellow, Bengio, and Courville, 2016, MIT Press, p. 197.
Link to standard parametric methods

The sequential setup implies a (complicated) par function

\[ f(x; \omega) \]

We could simply take this as parametric model for \( \mathbb{E}[Y_i | X_i = x] \).

- Then we can estimate \( \omega \) by minimizing, over all \( \omega_{l_j}^{(k)} \), \( j = 1, \ldots, p_k, \ l = 1, \ldots, p_{k+1}, \ k = 1, \ldots, K + 1 \), using nonlinear optimization methods (e.g., Newton-Raphson):

\[
\sum_{i=1}^{N} \left( y_i - f(x_i; \omega) \right)^2
\]

- Linearize to do standard inference for nonlinear least squares and get asymptotic normality for \( \omega_{l_j}^{(k)} \) and predicted values given model specification.
That is not what this literature is about:

- Computationally messy, multiple modes, too many parameters to calculate second derivatives.

- Too many parameters, with the model too non-linear, so that asymptotic normality is unlikely to be accurate approximation.
Instead, first add regularization:

Minimize, over all $\omega^{(k)}_{lj}$, $j = 1, \ldots, p_k$, $l = 1, \ldots, p_{k+1}$, $k = 1, \ldots, K + 1$:

$$\sum_{i=1}^{N} \left( y_i - f(x_i; \omega) \right)^2 + \text{penalty term}$$

where the penalty term is something like a LASSO penalty, or similar:

$$\lambda \times \sum_{k=1}^{K} \sum_{j=1}^{p_k} \sum_{l=1}^{p_{k-1}} \left( \omega^{(k)}_{jl} \right)^2$$

Choose $\lambda$ through cross-validation.
Estimating the Parameters of a Neural Network: Back-propagation Algorithm

Define objective function (initially without regularization)

\[ J_i(\omega, x, y) = \left( y_i - f(x_i; \omega) \right)^2 \]

\[ J(\omega, x, y) = \sum_{i=1}^{N} J_i(\omega, x_i, y_i) = \sum_{i=1}^{N} \left( y_i - f(x_i; \omega) \right)^2 \]

We wish to minimize this over \( \omega \).

• How do we calculate first derivatives (at least approximately)?

• Forget about second cross derivatives!
The backpropagation algorithm calculates the derivatives of $J_i(\omega, x_i, y_i)$ with respect to all $\omega_{li}^{(k)}$.

You start at the input layer and calculate the hidden elements $z_{lj}^{(k)}$, one layer at a time.

Then after calculating all the current values of the hidden elements, we calculate the derivatives, starting with the derivatives for the $\omega^{(k)}$ in the last, output layer, working backwards to the first layer.

Key is that the $\omega$ only enter in one layer each.
Recall layer $k$,

$$z_l^{(k)} = \omega_l^{(k-1)} + \sum_{j=1}^{p_{k-1}} \omega_{lj}^{(k-1)} g(k) \left( z_l^{(k-1)} \right)$$

or in vector notation:

$$\mathbf{z}^{(k)} = h^{(k)} \left( \omega^{(k-1)}, \mathbf{z}^{(k-1)} \right)$$

- This function $h^{(k)}(\cdot)$ does not depend on $\omega$ beyond $\omega^{(k-1)}$.

Hence we can write the function $f(\cdot)$ recursively:

$$f(x; \omega) = h^{(K+2)} \left( \omega^{(K+1)}, h^{(K+1)} \left( \omega^{(K)}, \ldots, h^{(2)} \left( \omega^{(1)}, x \right) \ldots \right) \right)$$
Now start at the last layer where

\[ h^{(K+2)}(z^{(K+2)}) = z^{(K+2)}. \]

Define for observation \( i \):

\[ \delta_i^{K+2} = \frac{\partial}{\partial z_i^{(K+2)}} \left( y_i - h^{(K+2)} \left( z_i^{(K+2)} \right) \right)^2 \]

\[ = -2 \left( y_i - h \left( z_i^{(K+2)} \right) \right) h^{(K+2)}' \left( z_i^{(K+2)} \right) \]

\[ = -2 \left( y_i - z_i^{(K+2)} \right) \]

(this is just the scaled residual).
Down the layers:

Define $\delta^{(k)}_{li}$ in terms of $\delta^{(k+1)}_{li}$ and $z^{(k)}_{li}$ as a linear combination of the $\delta^{(k+1)}_{li}$:

$$
\delta^{(k)}_{li} = \left( \sum_{j=1}^{p_k} \omega^{(k)}_{jl} \delta^{(k+1)}_{li} \right) g' \left( z^{(k)}_{li} \right)
$$

Then the derivatives are

$$
\frac{\partial}{\partial \omega^{(k)}_{lj}} J_i(\omega, x_i, y_i) = g \left( z^{(k)}_{li} \right) \delta^{(k+1)}_{li}
$$
Given the derivatives, iterate over

\[ \omega_{m+1} = \omega_m - \alpha \times \frac{\partial}{\partial \omega} J(\omega, x, y) \]

\( \alpha \) is the "learning rate" often set at 0.01.

This is still computationally (unnecessarily) demanding:

- given that we dont do optimal line search, we do not need the exact derivative.
• often **stochastic gradient descent**: Instead of calculating

\[
\frac{\partial}{\partial \omega} J(\omega, x, y)
\]

**estimate** the derivative:

\[
\frac{\sum_{i=1}^{N} R_i \frac{\partial}{\partial \omega} J(\omega, x_i, y_i)}{\sum_{i=1}^{N} R_i}
\]

for a random selection of units \((R_i \in \{0, 1\})\) because it is faster.
Regularization to avoid overfitting:

• add penalty term, \( \lambda \sum_{jlk} \left( \omega_{jl}^{(k)} \right)^2 \), or \( \lambda \sum_{jlk} |\omega_{jl}^{(k)}| \)

• early stopping rule: monitor average error in test sample, and stop iterating when average test error deteriorates.
Classification

American Economic Association,
Continuing Education Program 2018
Machine Learning and Econometrics, Lecture 8

Guido Imbens
Outline

1. Classification
2. Relation to Discrete Choice Models
3. Classification Trees
4. Neural Networks for Classification
5. Support Vector Machines
A second canonical problem in supervised learning is classification.

The setting is again one where we observe for a large number of units, say $N$, a $p$-vector of covariates $x_i$, and an outcome $y_i$. The outcome is now a class, an unordered discrete variable, $y_i \in \{1, \ldots, M\}$. $M$ may be 2, or fairly large.

The goal is a function $f : \mathbb{X} \mapsto \{1, \ldots, M\}$ that will allow us to assign new units to one of the classes.

- This is like discrete choice problems, but often we are not interested in estimating the probability of $y_i = m$ given $x_i = x$, just the classification itself.
The objective is typically to minimize for new cases (out-of-sample) the miss-classification rate.

\[
\frac{1}{N_{\text{test}}} \sum_{i=N_{\text{train}}+1}^{N_{\text{test}}} 1_{y_i \neq f(x_i)}
\]
Consider the two-class (binary $y_i$) case.

Suppose that the sample is a random sample from an infinite population, where:

$$\text{pr}(y_i = 1|x_i = x) = \frac{\exp(x'\beta)}{1 + \exp(x'\beta)}$$

Then the optimal classification is

$$\hat{f}(x) = 1 \left\{ \frac{\exp(x'\beta)}{1 + \exp(x'\beta)} \geq 0.5 \right\}$$

or equivalently

$$\hat{f}(x) = 1 \left\{ x'\beta \geq 0 \right\}$$
The standard thing to do in discrete choice analysis would be to estimate $\beta$ as

$$\hat{\beta} = \arg \max L(\beta)$$

where

$$L(\beta) = \sum_{i=1}^{n} y_i \exp(x'_{i}\beta) - \ln \left( 1 + \exp(x'\beta) \right)$$

and then classify as

$$\hat{f}(x) = 1 \left\{ x'\hat{\beta} \geq 0 \right\}$$

Outliers can be very influential here.
The machine learning methods take a different approach. These differences include

1. the objective function,

2. complexity of the models,

3. regularization

We will look at each of these.
\( \beta \) is estimated by focusing on (in- and out-of-sample) classification accuracy. For in-sample classification accuracy, we could consider a maximum score type objective function:

\[
\hat{\beta} = \arg \max \sum_{i=1}^{N} \left| y_i - 1 \left\{ x' \beta \geq 0 \right\} \right|
\]

This of course is not necessarily an improvement. Given the true value for \( \beta \) using \( x' \beta > 0 \) as a classifier is optimal, and the maximum likelihood estimator is better than the estimator based on classification error.

- The classification rate does not give the optimal weight to different classification errors if we know the model is logistic. But, it is robust because it does not rely on a model.
With many covariates we can also regularize this estimator (or the maximum likelihood estimator) using an $L_1$ penalty term:

$$\hat{\beta} = \arg \max_{\beta} \sum_{i=1}^{N} \left| y_i - 1 \left\{ x_i' \beta \geq 0 \right\} \right| + \lambda \sum_{j=1}^{p} |\beta_p|$$

For logistic regression this goes back to Tibshirani’s original LASSO paper:

$$\hat{\beta} = \arg \max_{\beta} \sum_{i=1}^{n} \left\{ y_i \exp(x_i' \beta) - \ln \left( 1 + \exp(x_i' \beta) \right) \right\} + \lambda \sum_{j=1}^{p} |\beta_j|$$

We could also use other penalties, e.g., ridge, or elastic net.
Many difference in complexity and type of models. Methods and models are very close to regression versions. Slight differences in implementation and tuning parameters.

Here, we consider

1. trees

2. neural networks

3. support vector machines
Classification Trees

The basic algorithm has the same structure as in the regression case.

We start with a single leaf. We then decide on a covariate/feature $k = 1, \ldots, p$, to split on, and a threshold $c$, so that we split the single leaf into two leaves, depending on whether

$$x_{ik} \leq c$$

or not.

The key question is how to compare the different potential splits. In regression problems we use the sum of squared residuals within each potential leave for this. Here we consider alternatives.
Now define an **impurity** function as a function of the shares of different classes in a leaf:

\[ \phi(\pi_1, \ldots, \pi_M) \]

satisfying

\[ \phi(\pi_1, \ldots, \pi_M) \leq (1/M, 1/M, \ldots, 1/M) \]

\[ \phi(\pi_1, \ldots, \pi_M) \geq (1, 0, \ldots, 0) \]

(and permutations thereof)
Now consider a tree $T$ with leaves $t = 1, \ldots, T$. Let

$$\pi_{m|t} \text{ and } \pi_t$$

be the share of class $m$ in leave $t$ and the share of leave $t$ respectively.

Then the impurity of tree $T$ is the average of the impurities of the leaves, weighted by their shares:

$$I(T) = \sum_{t=1}^{T} p_t \phi(\pi_{1|t}, \ldots, \pi_{M|t})$$

This replaces the average squared residual as the objective function to grow/compare/prune trees.
Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.

\[
\phi(p_1, \ldots, p_M) = \sum_{m=1}^{M} p_m (1 - p_m) = 1 - \sum_{m=1}^{M} p_m^2
\]

Information criterion

\[
\phi(p_1, \ldots, p_M) = - \sum_{m=1}^{M} p_m \ln(p_m)
\]
Note that we do \textbf{not} use the within-sample classification error here as the criterion. That would amount to using

\[ \phi(p_1, \ldots, p_M) = 1 - \max_m p_m \]
After the first split, we consider each of the two leaves separately. For each leaf we again determine the optimal covariate to split on, and the optimal threshold.

Then we compare the impurity of the optimal split based on splitting the first leaf, and the impurity of the optimal split based on splitting the second leaf, and choose the one that most improves the impurity.

We keep doing this till we satisfy some stopping criterion.
Typically: stop if the impurity plus some penalty (e.g., linear in the number of leaves) is minimized:

\[ I(\mathcal{T}) + \lambda |\mathcal{T}| \]

The value of the penalty term, \( \lambda \), is determined by out-of-sample cross-validation.
To evaluate trees, and to compare them out of sample, we typically do use the fraction of miss-classified units in a test sample:

\[ \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} 1 \{ y_i \neq f(x_i) \} \]

without any weighting.
Neural Networks for Classification

Recall the output layer in the regression version, where we had a single scalar outcome $y$:

$$z^{(K+2)} = \omega_0^{(K+1)} + \sum_{j=1}^{p_{K+1}} \omega_j^{(K+1)} \alpha_j^{K+1},$$

$$f(x) = g^{(K+2)}(z^{(K+2)}) = z^{(K+2)}$$

$$= \omega_0^{(K+1)} + \sum_{j=1}^{p_{K+1}} \omega_j^{(K+1)} \alpha_j^{K+1},$$
Now modify this by defining the outcome to be the vector $\tilde{y}$

$$\tilde{y}_m = 1\{y = m\}$$

Then introduce an $M$-component vector $z^{(K+2)}$

$$z_m^{(K+2)} = \omega_{m0}^{(K+1)} + \sum_{j=1}^{p_{K+1}} \omega_{mj}^{(K+1)} \alpha_j^{K+1},$$

and transformation (note that this involves all elements of $z^{(K+2)}$, unlike the transformations we considered before)

$$g_m^{(K+2)} (z^{(K+2)}) = \frac{\exp \left( z_m^{(K+2)} \right)}{\sum_{l=1}^{M} \exp \left( z_l^{(K+2)} \right)}$$

Now we can use all the machinery from the deep neural networks developed for the regression case.
Support Vector Machines

Consider the case with two classes, \( y_i \in \{-1, 1\} \).

Define a hyperplane by

\[
\{x | x' \beta + \beta_0 = 0\}
\]

with \( \|\beta\| = 1 \)

The hyperplane generates a classification rule:

\[
f(x) = \text{sign}(\beta_0 + x' \beta)
\]

Now if we are very lucky, or have lots of covariates, we may find a hyperplane that lead to a classification rule that has no in-sample classification errors, so that for all \( i \).

\[
y_i f(x_i) > 0 \quad (y_i = 1, f(x_i) > 0) \text{ or } (y_i = -1, f(x_i) < 0)
\]
In that case there are typically many such hyperplanes. We can look for the one that has the biggest distance between the two classes:

$$\max_{\beta_0, \beta, \|\beta\| = 1} M \text{ subject to } y_i(\beta_0 + x_i^T \beta) \geq M \quad i = 1, \ldots, N$$

This is equivalent to

$$\min_{\beta_0, \beta} \|\beta\| \text{ subject to } y_i(\beta_0 + x_i^T \beta) \geq 1 \quad i = 1, \ldots, N$$

This hyperplane depends only on a few units - outside of those you can move the points a little bit without changing the value of $\beta$.

Note that in this perfect separation case, the logit model is not estimable (parameters go to infinity).
Now it is rare that we can find a hyperplane that completely separates the classes. More generally we solve

$$\min_{\beta_0,\beta,\epsilon_1,\ldots,\epsilon_N} \|\beta\| \quad \text{subject to } y_i(\beta_0 + x_i^T\beta) \geq 1 - \epsilon_i$$

and $\epsilon_i \geq 0, \ i = 1, \ldots, N \sum_{i=1}^{N} \epsilon_i \leq K$

Or we can write this as ($C$ is cost tuning parameter)

$$\min_{\beta_0,\beta,\epsilon_1,\ldots,\epsilon_N} \|\beta\| + C \sum_{i=1}^{N} \epsilon_i$$

subject to $y_i(\beta_0 + x_i^T\beta) \geq 1 - \epsilon_i, \ \epsilon_i \geq 0, \ i = 1, \ldots, N$
So far this is very much tied to the linear representation of the classifier.

We can of course add functions of the original $x$'s to make the model arbitrarily flexible.

It turns out we can do that in a particularly useful way using kernels of the type we use in kernel regression.

First we rewrite the solution for the classifier in terms of the dual problem.
Specifically, we can characterize the solution for the classifier as

$$\max_{0 \leq \alpha_i \leq C} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} x_i^\top x_{i'} \quad \text{s.t.} \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

and

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i x^\top x_i + \hat{\beta}_0$$

The $\alpha_i$ are the Lagrange multipliers for the restrictions $y_i(\beta_0 + x_i^\top \beta) \geq 1 - \epsilon_i$

$C$ is the critical tuning parameter, typically chosen through cross-validation.
This can be generalized by replacing $x^\top x_i$ by a kernel $K(\cdot, \cdot)$ evaluated at $x$ and $x_i$, so that

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + \beta_0$$

For example, we can use an exponential kernel

$$K(x, x') = \exp(-\gamma \|x - x'\|)$$

This leads to very flexible surfaces to separate the classes.
Matrix Completion Methods
and Causal Panel Models

American Economic Association,
Continuing Education Program 2018
Machine Learning and Econometrics, Lecture 7-8

Guido Imbens
Motivating Example I

- California’s anti-smoking legislation (Proposition 99) took effect in 1989.

- What is the causal effect of the legislation on smoking rates in California in 1989?

- We observe smoking rates in California in 1989 given the legislation. We need to impute the counterfactual smoking rates in California in 1989 had the legislation not been enacted.

- We have data in the absence of smoking legislation in California prior to 1989, and for other states both before and after 1989. (and other variables, but not of essence)
Motivating Example II

In US, on any day, 1 in 25 patients suffers at least one Hospital Acquired Infection (HAI).

• Hospital acquired infections cause 75,000 deaths per year, cost 35 billion dollars per year

• 13 states have adopted a reporting policy (at different times during 2000-2010) that requires hospitals to report HAIs to the state.

What is (average) causal effect of reporting policy on deaths or costs?
Set Up: we observe (in addition to covariates):

\[
Y = \begin{pmatrix}
Y_{11} & Y_{12} & Y_{13} & \ldots & Y_{1T} \\
Y_{21} & Y_{22} & Y_{23} & \ldots & Y_{2T} \\
Y_{31} & Y_{32} & Y_{33} & \ldots & Y_{3T} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
Y_{N1} & Y_{N2} & Y_{N3} & \ldots & Y_{NT}
\end{pmatrix}
\]  

(realized outcome).

\[
W = \begin{pmatrix}
1 & 1 & 0 & \ldots & 1 \\
0 & 0 & 1 & \ldots & 0 \\
1 & 0 & 1 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & 0 & 1 & \ldots & 0
\end{pmatrix}
\]  

(binary treatment).

- rows of \( Y \) and \( W \) correspond to physical units, columns correspond to time periods.
In terms of potential outcome matrices $Y(0)$ and $Y(1)$:

$$Y(0) = \begin{pmatrix} \_ & \_ & \checkmark & \ldots & \_ \\ \checkmark & \checkmark & \_ & \ldots & \checkmark \\ \_ & \checkmark & \_ & \ldots & \checkmark \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \_ & \checkmark & \_ & \ldots & \checkmark \end{pmatrix} \quad Y(1) = \begin{pmatrix} \checkmark & \checkmark & \_ & \ldots & \checkmark \\ \_ & \_ & \checkmark & \ldots & \_ \\ \checkmark & \_ & \checkmark & \ldots & \_ \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \_ & \_ & \_ & \ldots & \_ \end{pmatrix}. $$

$Y_{it}(0)$ observed iff $W_{it} = 0$, $Y_{it}(1)$ observed iff $W_{it} = 1$.

In order to estimate the average treatment effect for the treated, (or other average, e.g., overall average effect)

$$\tau = \frac{\sum_{i,t} W_{it}(Y_{it}(1) - Y_{it}(0))}{\sum_{i,t} W_{it}},$$

We need to **impute** the missing potential outcomes in $Y(0)$ (and in $Y(1)$ for other estimands).
Focus on problem of imputing missing in $N \times T$ matrix $Y = Y(0)$

$$Y_{N \times T} = \begin{pmatrix}
? & ? & \checkmark & \ldots & ? \\
\checkmark & \checkmark & ? & \ldots & \checkmark \\
? & \checkmark & ? & \ldots & \checkmark \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
? & \checkmark & ? & \ldots & \checkmark
\end{pmatrix}$$

$\emptyset$ and $\mathcal{M}$ are sets of indices $(i, t)$ with $Y_{i,t}$ observed and missing, with cardinalities $|\emptyset|$ and $|\mathcal{M}|$. Covariates, time-specific, unit-specific, time/unit-specific.

• Now the problem is a **Matrix Completion Problem**.
At times we focus on the special case with $W_{it} = 1$ iff $(i, t) = (N, T)$, so that $|\varnothing| = N \times T - 1$ and $|\mathcal{M}| = 1$:

$$
\mathbf{Y} = \begin{pmatrix}
Y_{11} & Y_{12} & Y_{13} & \ldots & Y_{1T} \\
Y_{21} & Y_{22} & Y_{23} & \ldots & Y_{2T} \\
Y_{31} & Y_{32} & Y_{33} & \ldots & Y_{3T} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
Y_{N1} & Y_{N2} & Y_{N3} & \ldots & ?
\end{pmatrix} \quad \text{(realized outcome)}.
$$

$$
\mathbf{W} = \begin{pmatrix}
1 & 1 & 1 & \ldots & 1 \\
1 & 1 & 1 & \ldots & 1 \\
1 & 1 & 1 & \ldots & 1 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & 1 & 1 & \ldots & 0
\end{pmatrix} \quad \text{(binary treatment)}.
$$

Here single entry $Y_{NT}$ needs to be imputed.
But also important **Staggered Adoption** (e.g., adoption of technology, Athey and Stern, 1998)

− Here each unit is characterized by an adoption date $T_i \in \{1, \ldots, T, \infty\}$ that is the first date they are exposed to the treatment.

− Once exposed a unit will subsequently be exposed, $W_{it+1} \geq W_{it}$.

$$Y_{N \times T} = \begin{pmatrix}
\checkmark \checkmark \checkmark \checkmark \ldots \checkmark \\
\checkmark \checkmark \checkmark \checkmark \ldots ? \\
\checkmark \checkmark \checkmark \checkmark \ldots ? \\
\checkmark \checkmark ? ? \ldots ? \\
\checkmark \checkmark ? ? \ldots ? \\
\vdots \vdots \vdots \vdots \ddots \vdots \\
\checkmark ? ? ? \ldots ? \end{pmatrix}
$$

(never adopter) (late adopter) (medium adopter) (early adopter)
Netflix Problem

- $N \approx 500,000$ (individuals), raises computational issues
- Large $T \approx 20,000$ (movies),
- General missing data pattern,
- Fraction of observed data is close to zero, $|\emptyset| \ll N \times T$

$$Y_{N \times T} = \begin{pmatrix}
\checkmark & ? & ? & ? & \checkmark & ? & \ldots & \checkmark \\
\checkmark & ? & ? & ? & ? & ? & \ldots & \checkmark \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\end{pmatrix}$$
How is this problem treated in the various literatures?

1. Program evaluation literature under unconfoundedness / selection-on-observables (Imbens-Rubin 2015)

2. Synthetic Control literature (Abadie-Diamond-Hainmueller, JASA 2010)

3. Differences-In-Differences Literature (Bertrand-Dulfo-Mullainathan 2003) and Econometric Panel Data literature (Bai 2003, 2009; Bai and Ng 2002)

1. Program Evaluation Literature under unconfoundedness / selection-on-observables (Imbens-Rubin 2015)

Focuses on special case:

- **Thin** Matrix ($N$ large, $T$ small),

- $Y_{iT}$ is missing for some $i$ ($N_t$ ”treated units”), and no missing entries for other units ($N_c = N - N_t$ “control units”).

- $Y_{it}$ is always observed for $t < T$.

\[
Y_{N \times T} = \begin{pmatrix}
\checkmark & \checkmark & \checkmark \\
\checkmark & \checkmark & \checkmark \\
\checkmark & \checkmark & ? \\
\checkmark & \checkmark & ? \\
\vdots & \vdots & \vdots \\
\checkmark & \checkmark & ?
\end{pmatrix}
\]

\[
W_{N \times T} = \begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 0 \\
1 & 1 & 0 \\
1 & 1 & 1 \\
1 & 1 & 0
\end{pmatrix}
\]
Parametric version: **Horizontal** Regression

**Specification:**

\[ Y_{iT} = \beta_0 + \sum_{t=1}^{T-1} \beta_t Y_{it} + \varepsilon_i \]

estimated on \( N_c \) control units, with \( T \) regressors.

**Prediction for** \( N_t \) treated units:

\[ \hat{Y}_{iT} = \hat{\beta}_0 + \sum_{t=1}^{T-1} \hat{\beta}_t Y_{Nt} \]

- Nonparametric version: for each treated unit \( i \) find a control unit \( j \) with \( Y_{it} \approx Y_{jt}, \ t < T \).

- If \( N \) large relative to \( T_0 \) use regularized regression (lasso, ridge, elastic net)
2. **Synthetic Control Literature** (Abadie-Diamond-Hainmueller, JASA 2010)

Focuses on special case:

- **Fat** Matrix ($N$ small, $T$ large)

- $Y_{Nt}$ is missing for $t \geq T_0$ and no missing entries for other units.

\[
Y_{N \times T} = \begin{pmatrix}
\checkmark & \checkmark & \checkmark & \ldots & \checkmark \\
\checkmark & \checkmark & \checkmark & \ldots & \checkmark \\
\checkmark & \checkmark & ? & \ldots & ?
\end{pmatrix}
\quad W_{N \times T} = \begin{pmatrix}
1 & 1 & 1 & \ldots & 1 \\
1 & 1 & 1 & \ldots & 1 \\
1 & 1 & 0 & \ldots & 0
\end{pmatrix}
\]
Parametric version: \textbf{Vertical Regression}:

\[ Y_{Nt} = \alpha_0 + \sum_{i=1}^{N-1} \alpha_i Y_{it} + \eta_t \]

estimated on $T_0$ pre-treatment periods, with $N$ regressors (in ADH with restrictions $\alpha_0 = 0$, $\alpha_i \geq 0$).

Prediction for $T - T_0$ treated periods:

\[ \hat{Y}_{Nt} = \hat{\alpha}_0 + \sum_{i=1}^{N-1} \hat{\alpha}_i Y_{it} \]

- Nonparametric version: for each post-treatment period $t$ find a pre-treatment period $s$ with $Y_{is} \approx Y_{it}$, $i = 1, \ldots, N - 1$.

- If $T$ large relative to $N_c$ use regularized regression (lasso, ridge, elastic net), following Doudchenko and Imbens (2016).
3. Differences-In-Differences Literature and Econometric Panel Data literature (Bertrand-Dulfo-Mullainathan, 2003; Bai 2003, 2009; Bai-Ng 2002)

Note: we model only $Y = Y(0)$ here, because $Y_{it}(1)$ is observed for the relevant units/time-periods.

Models developed here for complete-data matrices:

**DID:**

$$Y_{it} = \delta_i + \gamma_t + \varepsilon_{it}$$

**Generalized Fixed Effects** (Bai 2003, 2009)

$$Y_{it} = \sum_{r=1}^{R} \delta_{ir} \gamma_{rt} + \varepsilon_{it}$$

Estimate $\delta$ and $\gamma$ by least squares and use to impute missing values.

− Focus on setting with many missing entries: \( \frac{|\emptyset|}{|\mathcal{M}|} \approx 0 \).
  E.g., netflix problem, with units corresponding to individuals, and time periods corresponding to movie titles, or image recovery from limited information, with \( i \) and \( t \) corresponding to different dimensions.

− Focus on computational feasibility, with both \( N \) and \( T \) large.

− Focus on randomly missing entries: \( \mathbf{W} \perp \perp \mathbf{Y} \).

Like interactive fixed effect, focus on low-rank structure underlying observations.
Set Up

General Case: \( N >> T, \ N << T, \ N \approx T, \) possibly stable patterns over time, possibly stable patterns within units.

Set up without covariates, connecting to the interactive fixed effect literature (Bai, 2003), and the matrix completion literature (Cands, and Recht, 2009):

\[
Y_{N \times T} = L_{N \times T} + \varepsilon_{N \times T}
\]

• Key assumption:

\[
W_{N \times T} \perp \perp \varepsilon_{N \times T} \quad \text{(but } W \text{ may depend on } L)\]

• Staggered entry, \( W_{it+1} \geq W_{it} \)

• \( L \) has low rank relative to \( N \) and \( T \).
More general case, with unit-specific $P$-component covariate $X_i$, time-specific $Q$-component covariate $Z_t$, and unit-time-specific covariate $V_{it}$:

$$Y_{it} = L_{it} + \sum_{p=1}^{P} \sum_{q=1}^{Q} X_{ip} H_{pq} Z_{qt} + \gamma_i + \delta_t + V_{it} \beta + \varepsilon_{it}$$

- We do not necessarily need the fixed effects $\gamma_i$ and $\delta_t$, these can be subsumed into $L$. It is convenient to include the fixed effects given that we regularize $L$. 
Too many parameters (especially $N \times T$ matrix $L$), so we need regularization:

We shrink $L$ and $H$ towards zero.

For $H$ we use Lasso-type element-wise $\ell_1$ norm: defined as $\|H\|_{1,e} = \sum_{p=1}^{P} \sum_{q=1}^{Q} |H_{pq}|$. 
How do we regularize $L$?

$$L_{N \times T} = S_{N \times N} \Sigma_{N \times T} R_{T \times T}$$

$S, R$ unitary, $\Sigma$ is rectangular diagonal with entries $\sigma_i(L)$ that are the singular values. Rank of $L$ is number of non-zero $\sigma_i(L)$.

$$\|L\|_F^2 = \sum_{i,t} |L_{it}|^2 = \sum_{j=1}^{\min(N,T)} \sigma_i^2(L) \quad \text{(Frobenius, like ridge)}$$

$$\implies \|L\|_* = \sum_{j=1}^{\min(N,T)} \sigma_i(L) \quad \text{(nuclear norm, like LASSO)}$$

$$\|L\|_R = \sum_{j=1}^{\min(N,T)} 1_{\sigma_i(L) > 0} \quad \text{(Rank, like subset selection)}$$
Frobenius norm imputes missing values as 0.

Rank norm is computationally not feasible for general missing data patterns.

The preferred Nuclear norm leads to low-rank matrix but is computationally feasible.
Our proposed method: regularize using using nuclear norm:

\[
\min_{L} \frac{1}{|\mathcal{O}|} \sum_{(i,t) \in \mathcal{O}} (Y_{it} - L_{it})^2 + \lambda_L \|L\|_*
\]

For the general case we estimate \(H, L, \delta, \gamma, \) and \(\beta\) as

\[
\min_{H,L,\delta,\gamma} \frac{1}{|\mathcal{O}|} \sum_{(i,t) \in \mathcal{O}} \left( Y_{it} - L_{it} - \sum_{p=1}^{P} \sum_{q=1}^{Q} X_{ip} H_{pq} Z_{qt} - \gamma_i - \delta_t - V_{it} \beta \right)^2
\]

\[+ \lambda_L \|L\|_* + \lambda_H \|H\|_{1,e}\]

We choose \(\lambda_L\) and \(\lambda_H\) through cross-validation.
**Algorithm** (Mazumder, Hastie, & Tibshirani 2010)

Given any $N \times T$ matrix $A$, define the two $N \times T$ matrices $P_\emptyset(A)$ and $P_{\perp \emptyset}(A)$ with typical elements:

$$P_\emptyset(A)_{it} = \begin{cases} A_{it} & \text{if } (i, t) \in \emptyset, \\ 0 & \text{if } (i, t) \notin \emptyset, \end{cases}$$

and

$$P_{\perp \emptyset}(A)_{it} = \begin{cases} 0 & \text{if } (i, t) \in \emptyset, \\ A_{it} & \text{if } (i, t) \notin \emptyset. \end{cases}$$
Let $A = S \Sigma R^\top$ be the Singular Value Decomposition for $A$, with $\sigma_1(A), \ldots, \sigma_{\min(N,T)}(A)$, denoting the singular values.

Then define the matrix shrinkage operator

$$\text{shrink}_\lambda(A) = S \tilde{\Sigma} R^\top,$$

where $\tilde{\Sigma}$ is equal to $\Sigma$ with the $i$-th singular value $\sigma_i(A)$ replaced by $\max(\sigma_i(A) - \lambda, 0)$. 
Given these definitions, the algorithm proceeds as follows.

− Start with the initial choice $L_1(\lambda) = P_\emptyset(Y)$, with zeros for the missing values.

− Then for $k = 1, 2, \ldots$, define,

$$L_{k+1}(\lambda) = \text{shrink}_\lambda \left\{ P_\emptyset(Y) + P_\emptyset^\perp(L_k(\lambda)) \right\},$$

until the sequence $\{L_k(\lambda)\}_{k \geq 1}$ converges.

− The limiting matrix $L^*$ is our estimator for the regularization parameter $\lambda$, denoted by $\hat{L}(\lambda, \emptyset)$.
Results I (for case without covariates, and just $L$, $\emptyset$ is sufficiently random, and $\varepsilon_{it} = Y_{it} - L_{it}$ are iid with variance $\sigma^2$.

$$\|Y\|_F = \sqrt{\sum_{i,t} Y_{i,t}^2}$$ is Frobenius norm. $\|Y\|_\infty = \max_{i,t} |Y_{i,t}|$.

Let $Y^*$ be the matrix including all the missing values (e.g., $Y(1)$.

The estimated matrix $\hat{L}$ is close to $L^*$ in the following sense:

$$\frac{\|\hat{L} - L\|_F}{\|L\|_F} \leq C \max \left( \sigma, \frac{\|L\|_\infty}{\|L\|_F} \right) \frac{\text{rank}(L)(N + T) \ln(N + T)}{|\emptyset|}$$

In many cases the number of observed entries $|\emptyset|$ is of order $N \times T$ so if $\text{rank}(L) \ll (N + T)$ the error goes to 0 as $N + T$ grows.
Adaptive Properties of Matrix Regression

Suppose $N$ is large, $T = 2$, $W_{i1} = 0$, many control units (treatment effect setting)

– In that case the natural imputation is $L_{i2} = Y_{i1} \times \rho$, where $\rho$ is the within unit correlation between periods for the control units.

– The program-evaluation / horizontal-regression approach would lead to $L_{i2} = Y_{i1} \times \rho$

– The synthetic-control / vertical-regression approach would lead to $L_{i2} = 0$.

– How does the matrix completion estimator impute the missing values in this case?
Suppose for control units

\[
\begin{pmatrix}
Y_{i1} \\
Y_{i2}
\end{pmatrix}
\sim \mathcal{N}\left(\begin{pmatrix}
0 \\
0
\end{pmatrix}, \begin{pmatrix}
1 & \rho \\
\rho & 1
\end{pmatrix}\right),
\]

and suppose the pairs \((Y_{i1}, Y_{i2})\) are jointly independent.

Then for small \(\lambda\), for the treated units, the imputed value is

\[
\hat{L}_{i2} = Y_{i1} \times \frac{\rho}{1 + \sqrt{(1 - \rho^2)}}
\]

The regularization leads to a small amount of shrinkage towards zero relative to optimal imputation.

**Matrix Completion** method adapts well to shape of matrix and correlation structure.
Illustrations

• We take complete matrices $Y_{N \times T}$.

• We pretend some entries are missing.

• We use different estimators to impute the “missing” entries and compare them to actual values, and calculate the mean-squared-error.
We compare three estimators:

1. Matrix Completion Nuclear Norm, MC-NNM

2. Horizontal Regr with Elastic Net Regularization, EN-H (Program evaluation type regression)

3. Vertical Regr with Elastic Net Regularization, EN-V (Synthetic Control type regression)
Illustrations I: Stock Market Data

We use daily returns for 2453 stocks over 10 years (3082 days). We create sub-samples by looking at the first $T$ daily returns of $N$ randomly sampled stocks for pairs of $(N, T)$ such that $N \times T = 4900$, ranging from fat to thin: $(N, T) = (10, 490), \ldots, (70, 70), \ldots, (490, 10)$.

Given the sample, we pretend that half the stocks are treated at the mid point over time, so that 25% of the entries in the matrix are missing.

\[
Y_{N \times T} = \begin{pmatrix}
✓ & ✓ & ✓ & ✓ & \ldots & ✓ \\
✓ & ✓ & ✓ & ✓ & \ldots & ✓ \\
✓ & ✓ & ✓ & ✓ & \ldots & ✓ \\
✓ & ✓ & ✓ & ✓ & \ldots & ✓ \\
✓ & ✓ & ✓ & ? & \ldots & ? \\
✓ & ✓ & ✓ & ? & \ldots & ? \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
✓ & ✓ & ✓ & ? & \ldots & ? \\
\end{pmatrix}
\]
N\times T = 4900   Fraction Missing = 0.25

Average RMSE

Method
EN
EN−T

MC−NNM

NxT = 4900   Fraction Missing = 0.25

Average RMSE

Method
EN
EN−T
MC−NNM

log(N)
Illustrations II: California Smoking Rate Data

The outcome here is per capita smoking rates by state, for 38 states, 31 years.

We compare both simultaneous adoption and staggered adoption.
Illustration I: California Smoking Example

\((N = 38, T = 31)\)
Generalizations I:

- Allow for propensity score weighting to focus on fit where it matters:

Model propensity score $E_{it} = pr(W_{it} = 1|X_i, Z_t, V_{it})$, $E$ is $N \times T$ matrix with typical element $E_{it}$

Possibly using matrix completion:

$$\hat{E} = \arg \min_E \frac{1}{NT} \sum_{(i,t)} (W_{it} - E_{it})^2 + \lambda_L \|E\|_*$$

and then

$$\min_L \frac{1}{|\emptyset|} \sum_{(i,t) \in \emptyset} \frac{\hat{E}_{it}}{1 - \hat{E}_{it}} (Y_{it} - L_{it})^2 + \lambda_L \|L\|_*$$
Generalizations II:

- Take account of time series correlation in $\varepsilon_{it} = Y_{it} - L_{it}$

Modify objective function from logarithm of Gaussian likelihood based on independence to have autoregressive structure.
References


Abadie, Alberto, and Jeremy L’Hour “A Penalized Synthetic Control Estimator for Disaggregated Data”

Athey, Susan, Guido W. Imbens, and Stefan Wager. Efficient inference of average treatment effects in high dimensions via


Imbens, G., and D. Rubin *Causal Inference* Cambridge University Press.


