Machine Learning Methods: A Brief Overview

A. Colin Cameron
U.C.-Davis

.

Presented at U.C.-Davis

June 7 2019

A ten minute summary: 1. Prediction

- Think of machine learning as potentially better nonparametric regression.
- We wish to predict y given $\mathbf{x}$ using fitted function $\widehat{f}(\mathbf{x})$.
- We could use various nonparametric methods
  - kernel regression such as local linear, nearest neighbors, sieves
  - but these perform poorly if $\mathbf{x}$ is high dimensional
    - ★ the curse of dimensionality.
- Machine learning uses different algorithms that may predict better
  - including lasso, neural networks, deep nets and random forests.
  - these require setting tuning parameter(s)
    - ★ just as e.g. kernel regression requires setting bandwidths.
- Machine learning focuses purely on prediction
  - sometimes useful in microeconomics applications
  - e.g. predict one-year survival following hip transplant operation.

A ten minute summary: 2. Inference for Economics

- But much empirical microeconomics emphasizes estimating a partial effect.
- In principle can perturb an x to get $\Delta\widehat{f}(\mathbf{x})$
  - but very black box especially if $\widehat{f}(\mathbf{x})$ is very nonlinear
  - and statistical inference is a problem.
- Instead economists impose more structure.
- A leading example is the partially linear model
  - estimate β in the model $y = \beta x_1 + g(\mathbf{x}_2) + u$.
- A second leading example is an average treatment effect
  - rather than an individual treatment effect.

A ten minute summary: 3. Orthogonalization

- A standard semiparametric estimator is the Robinson (1988) differencing estimator in the partially linear model
  - $y = \beta x_1 + g(\mathbf{x}_2) + u$
  - β is OLS estimate in model $y - \widehat{m}_y = \beta(x_1 - \widehat{m}_{x_1}) +$ error.
  - where use kernel regression of y on $\mathbf{x}_2$ for $\widehat{m}_y$ and x on $\mathbf{x}_2$ for $\widehat{m}_{x_1}$.
- Remarkable result
  - the asymptotic distribution of β at the second stage
  - is not affected by the first step estimation of $\widehat{m}_y$ and $\widehat{m}_{x_1}$
  - an example of using an "orthogonal moment condition".
- This generalizes
  - use a machine learner for $\widehat{m}_y$ and $\widehat{m}_{x_1}$ instead of kernel regression
  - and apply to other settings with an "orthogonal moment condition"
    - ★ e.g. ATE, ATET and LATE where x_1 is a binary treatment.
  - this is a big, big deal.

A ten minute summary: 4. Other contributions of machine learning

- Estimators overfit the sample at hand
  - e.g. chasing outliers
  - so use out-of-sample prediction as criteria
    - ★ in particular k-fold cross-validation
  - or use penalties such as AIC, BIC.
- Biased estimators can outperform unbiased estimators
  - e.g. shrinkage estimators such as LASSO and ridge.
- Data carpentry that creates y and $\mathbf{x}$
  - web scraping, text mining, digitizing images, SQL.

Overview

1. Terminology
2. Model selection - especially cross-validation.
3. Variance-bias trade-off and shrinkage (LASSO and Ridge)
4. Dimension reduction (principal components)
5. Nonparametric and semiparametric regression
6. Flexible regression (splines, sieves, neural networks,...)
7. Regression trees and random forests
8. Classification (support vector machines)
9. Unsupervised learning (cluster analysis)
10. Prediction for economics
11. LASSO for causal homogeneous effects
12. Heterogeneous treatment effects
13. Double / debiased machine learning
14. Conclusions

1. Terminology

- The term **machine learning** is used because the machine (computer) figures out from data the model $\widehat{f}(\mathbf{x})$
  - compared to a modeler who e.g. specifies $\mathbf{x}$ and $y = \mathbf{x}'\boldsymbol{\beta} + u$.
- The data may be big or small
  - typically dim($\mathbf{x}$) is large but n can be small or large.
- **Supervised learning = Regression**
  - We have both outcome y and regressors (or **features**) $\mathbf{x}$
  - 1. **Regression**: y is continuous
  - 2. **Classification**: y is categorical.
- **Unsupervised learning**
  - We have no outcome y - only several $\mathbf{x}$
  - 3. **Cluster Analysis**: e.g. determine five types of individuals given many psychometric measures.
- Focus on 1. as this is most used by economists.

Terminology (continued)

- Consider two types of data sets
  - 1. **training data set** (or **estimation sample)**
    - ★ used to fit a model.
  - 2. **test data set** (or **hold-out sample** or **validation set**)
    - ★ additional data used to determine how good is the model fit
    - ★ a test observation $(\mathbf{x}_0, y_0)$ is a previously unseen observation.

2. Model selection

- Can choose $x's$ by
  - start from smallest and build
  - start from largest and prune
  - best subsets: find best model of given size and then choose best size.
- Traditionally use statistical significance $(p < 0.05)$
  - but pre-testing changes the distribution of $\widehat{\beta}$.
- Machine learners instead use predictive ability
  - typically mean squared error MSE $= \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2$.

Overfitting

- Problem: models "overfit" within sample.
  - e.g. $\widehat{\mathbf{u}} = (\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_{OLS}) = (\mathbf{I} - \mathbf{M})\mathbf{u}$ where $\mathbf{M} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}$
    - ★ so $|\widehat{u}_i| < |u_i|$ on average.
- Two solutions:
  - penalize for overfitting e.g. $\bar{R}^2$, AIC, BIC, Mallows Cp
  - use out-of-estimation sample prediction (cross-validation)
    - ★ new to econometrics
    - ★ can apply to other loss functions and not just MSE.

K-fold cross-validation is standard method

- K-fold cross-validation
  - split data into K mutually exclusive folds of roughly equal size
  - for $j = 1, ..., K$ fit using all folds but fold j and predict on fold j
  - standard choices are $K = 5$ and $K = 10$.
- The following shows case $K = 5$

	Fit on folds	Test on fold
$j = 1$	2,3,4,5	$1 \rightarrow \text{MSE}_{(1)}$
$j = 2$	1,3,4,5	$2 \rightarrow \text{MSE}_{(2)}$
$j = 3$	1,2,4,5	$3 \rightarrow \text{MSE}_{(3)}$
$j = 4$	1,2,3,5	$4 \rightarrow \text{MSE}_{(4)}$
$j = 5$	1,2,3,4	$5 \rightarrow \text{MSE}_{(5)}$

- The K-fold CV estimate is

$$\text{CV}_K = \frac{1}{K}\sum_{j=1}^{K} \text{MSE}_{(j)}, \text{ where } \text{MSE}_{(j)} \text{ is the MSE for fold } j.$$

3. Bias-Variance Trade-off and Shrinkage Estimation

- The goal is minimize MSE = Variance + Bias-squared.
- More flexible models have
  - less bias (good) and more variance (bad).
  - this trade-off is fundamental to machine learning.
- Shrinkage reduces variance and may offset increased bias.
  - e.g. $\widehat{\beta} = 0$ has reduced variance to zero.

Shrinkage Methods: Ridge

- Shrinkage estimators minimize RSS (residual sum of squares) with a penalty for model size
  - this shrinks parameter estimates towards zero.
- The ridge estimator $\widehat{\boldsymbol{\beta}}_\lambda$ of $\boldsymbol{\beta}$ minimizes

$$Q_\lambda(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \mathbf{x}_i'\boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = RSS + \lambda \sum_{j=1}^{p} \beta_j^2$$

  - where $\lambda \geq 0$ is a tuning parameter to be determined
- This yields $\widehat{\boldsymbol{\beta}}_\lambda = (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$
  - $\widehat{\boldsymbol{\beta}}_\lambda \rightarrow \widehat{\boldsymbol{\beta}}_{OLS}$ as $\lambda \rightarrow 0$ and $\widehat{\boldsymbol{\beta}}_\lambda \rightarrow \mathbf{0}$ as $\lambda \rightarrow \infty$.
- Typically first standardize $x's$ to have mean zero and variance 1.

Shrinkage Methods: LASSO

- Instead of squared penalty use absolute penalty.
- The Least Absolute Shrinkage and Selection (LASSO) estimator $\widehat{\boldsymbol{\beta}}_\lambda$ of $\boldsymbol{\beta}$ minimizes

$$Q_\lambda(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \mathbf{x}_i'\boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j| = RSS + \lambda \sum_{j=1}^{p} |\beta_j|$$

  - where $\lambda \geq 0$ is a tuning parameter to be determined.
- No closed form solution
  - sets some $\beta's$ to zero and shrinks others towards zero
  - hence name.

LASSO versus Ridge (key figure from ISL)

- LASSO is likely to set some coefficients to zero.

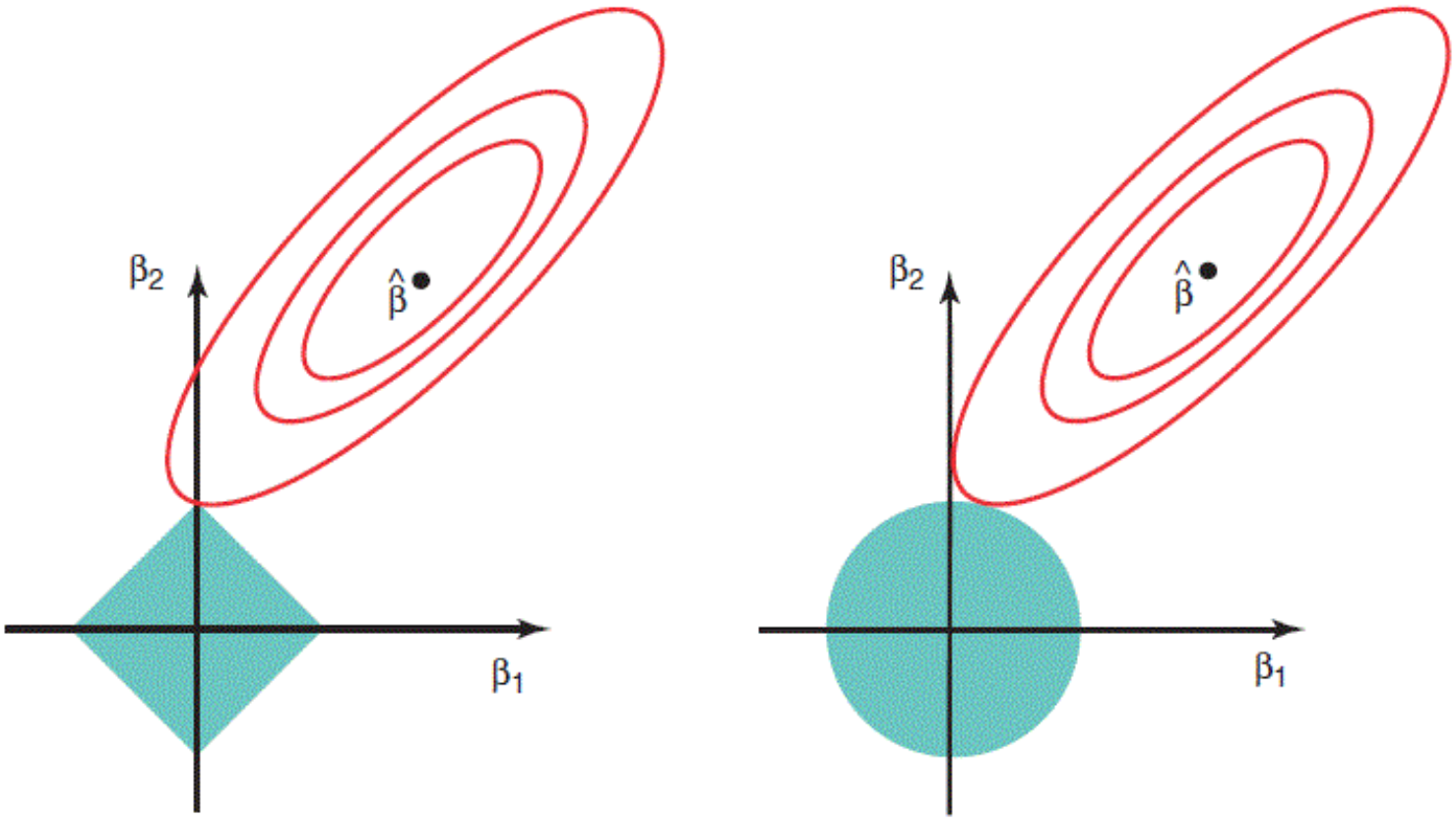


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

4. Dimension Reduction

- Reduce from p regressors to $M < p$ linear combinations of regressors.
- Principal components (or factor analysis) is standard method
  - The first principal component has the largest sample variance among all normalized linear combinations of the columns of $n \times p$ data matrix $\mathbf{X}$
- Considers only $\mathbf{x}$ without considering y
  - but still generally does good job of explaining y.

5. Nonparametric regression and semiparametric regression

- Nonparametric regression is the most flexible approach.
- Nonparametric regression methods for $f(\mathbf{x}_0) = \mathrm{E}[y|\mathbf{x} = \mathbf{x}_0]$ borrow from observations near to $\mathbf{x}_0$
  - ***k*-nearest neighbors**
    - ★ average y_i for the k observations with $\mathbf{x}_i$ closest to $\mathbf{x}_0$.
  - **kernel-weighted local regression**
    - ★ use a weighted average of y_i with weights declining as $||\mathbf{x}_i - \mathbf{x}_0||$ increases.
- But are not practical for high $p = \dim(\mathbf{x})$
  - due to the curse of dimensionality
  - e.g. if 10 bins in one dimension need 10^2 bins in two dimensions, ...

Semiparametric regression

- Semiparametric models provide some structure to reduce the nonparametric component from many dimensions to fewer dimensions (often one).
  - Econometricians focus on
    - ★ partially linear models $y = f(\mathbf{x}, \mathbf{z}) + u = \mathbf{x}'\boldsymbol{\beta} + g(\mathbf{z}) + u$
    - ★ single-index models $(y = g(\mathbf{x}'\boldsymbol{\beta}))$.
  - Statisticians use
    - ★ generalized additive models and project pursuit regression.
- Later we will work with partially linear models.

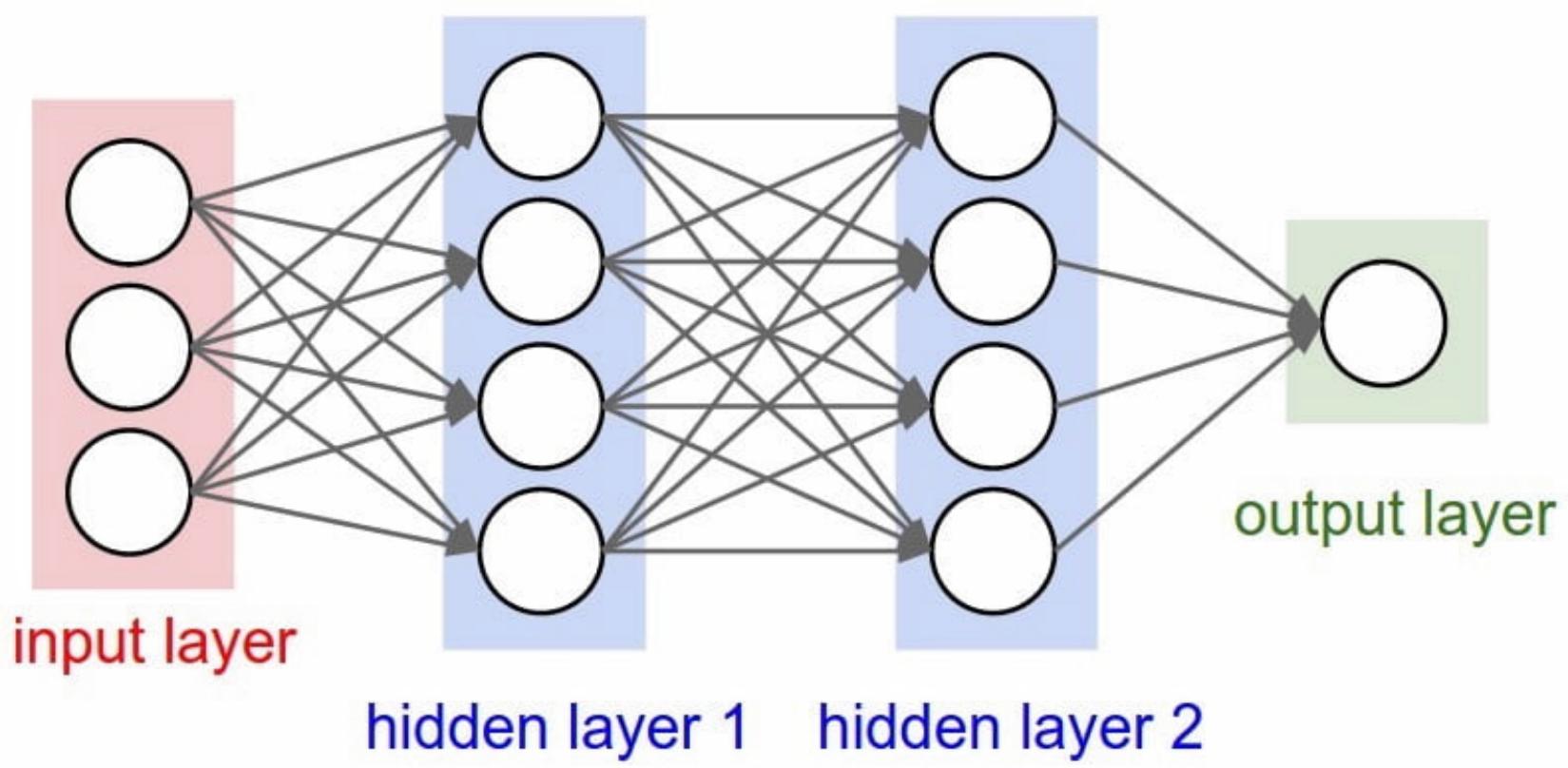
6. Flexible Regression

- Basis function models
  - scalar case: $y_i = \beta_0 + \beta_1 b_1(x_i) + \cdots + \beta_K(x_i) + \varepsilon_i$
    - ★ where $b_1(\cdot), ..., b_K(\cdot)$ are basis functions that are fixed and known.
  - global polynomial regression
  - splines: step functions, regression splines, smoothing splines
  - wavelets
  - polynomial is global while the others break range of x into pieces.
- Other methods
  - neural networks.

Neural Networks

- A neural network involves a series of nested logit regressions.
- A single hidden layer neural network explaining y by $\mathbf{x}$ has
  - y depends on $\mathbf{z}'s$ (a hidden layer)
  - $\mathbf{z}'s$ depend on $\mathbf{x}'s$.
- A neural network with two hidden layers explaining y by $\mathbf{x}$ has
  - y depends on $\mathbf{w}'s$ (a hidden layer)
  - $\mathbf{w}'s$ depend on $\mathbf{z}'s$ (a hidden layer)
  - $\mathbf{z}'s$ depend on $\mathbf{x}'s$.
- Neural nets are good for prediction
  - especially in speech recognition (Google Translate), image recognition, ...
  - but require much tuning and very difficult (impossible) to interpret
  - and basis for deep nets and deep learning.

Neural Network Example

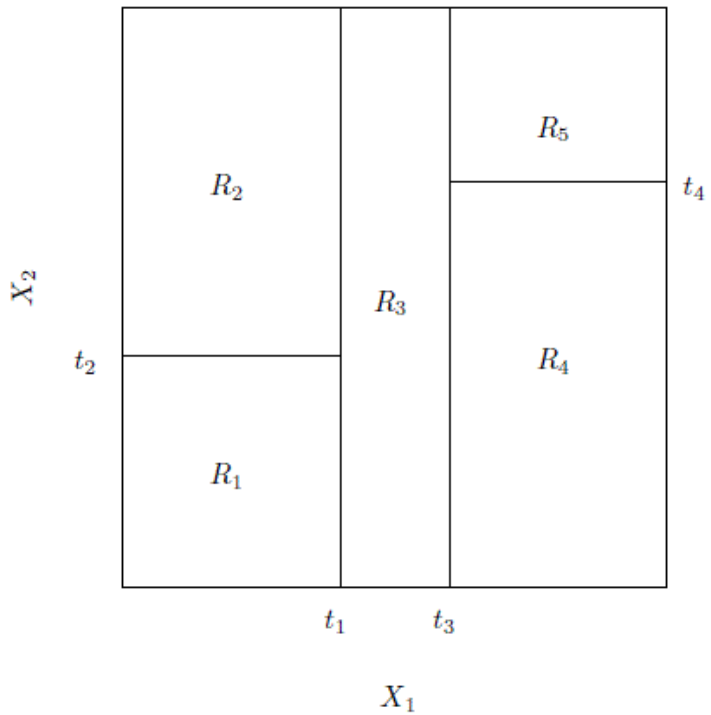


7. Regression Trees and Random Forests

- Regression trees sequentially split regressors $\mathbf{x}$ into regions that best predict y.
- Sequentially split $\mathbf{x}'s$ into rectangular regions in way that reduces RSS
  - then $\widehat{y}_i$ is the average of $y's$ in the region that $\mathbf{x}_i$ falls in
  - with J blocks RSS$= \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \bar{y}_{R_j})^2$.
- Simplest case is a single x
  - split at x^* that minimizes $\sum_{i:x_i \leq x^*} (y_i - \bar{y}_{R_1})^2 + \sum_{i:x_i > x^*} (y_i - \bar{y}_{R_1})^2$
    - ★ where $\bar{y}_{R_1}$ is average of y_i for $i : x_i \leq x^*$
    - ★ and $\bar{y}_{R_2}$ is average of y_i for $i : x_i > x^*$.
  - second split is then best split within R_1 and R_2
  - then predicted $y's$ are a step function of x.

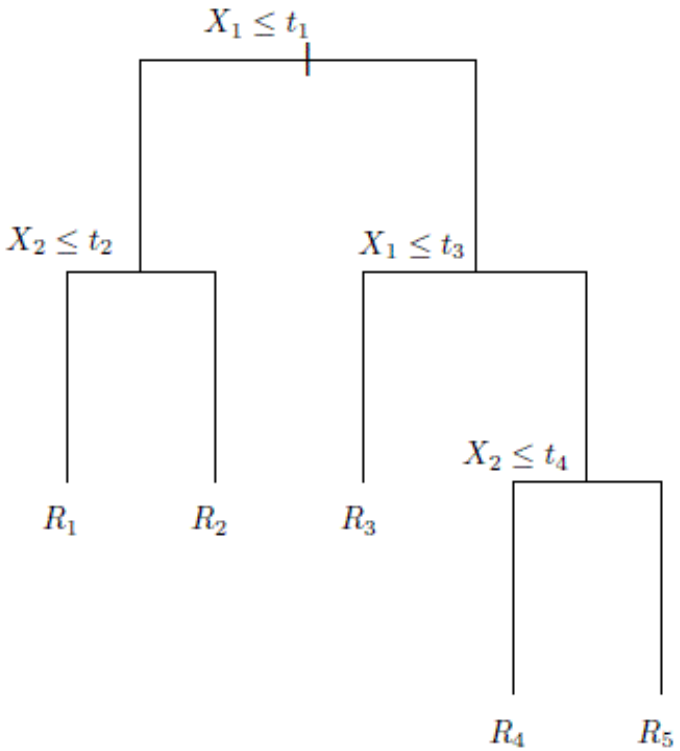
Tree example from ISL page 308

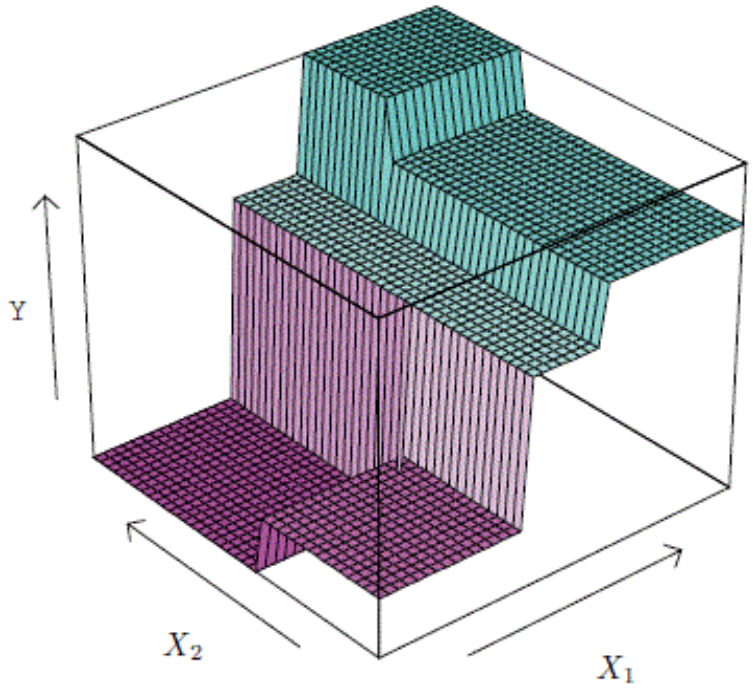
- (1) split X1 in two;
  (2) split the lowest X1 values on the basis of X2 into R1 and R2;
  (3) split the highest X1 values into two regions (R3 and R4/R5);
  (4) split the highest X1 values on the basis of X2 into R4 and R5.



Tree example from ISL (continued)

- The left figure gives the tree.
- The right figure shows the predicted values of y.





Improvements to regression trees

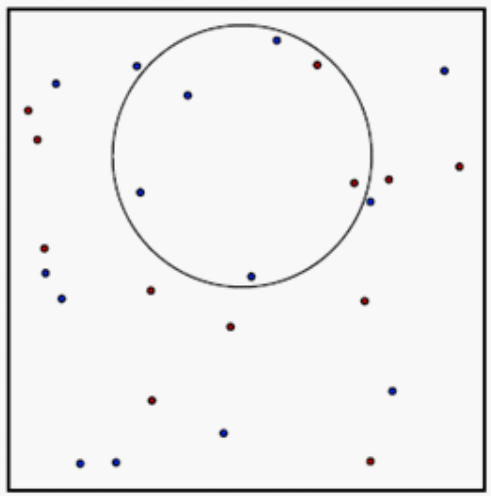
- Regression trees are easy to understand if there are few regressors.
- But they do not predict as well as methods given so far
  - due to high variance (e.g. split data in two then can get quite different trees).
- Better methods are
  - bagging
    - ★ bootstrap aggregating averages regression trees over many samples
  - random forests
    - ★ averages regression trees over many sub-samples
  - boosting
    - ★ trees build on preceding trees (fit residuals not y).

Random Forests

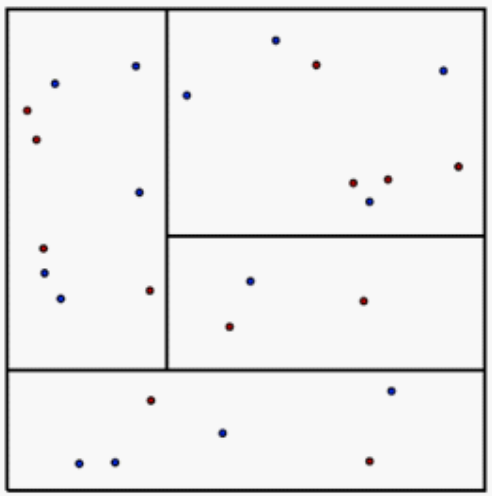
- If we bootstrap resample with replacement (bagging) the B estimates are correlated
  - e.g. if a regressor is important it will appear near the top of the tree in each bootstrap sample.
  - the trees look similar from one resample to the next.
- Random forests get bootstrap resamples (like bagging)
  - but within each bootstrap sample use only a random sample of $m < p$ predictors in deciding each split.
  - usually $m \simeq \sqrt{p}$
  - this reduces correlation across bootstrap resamples.
- Random forests are related to kernel and k-nearest neighbors
  - as use a weighted average of nearby observations
  - but with a data-driven way of determining which nearby observations get weight
  - see Lin and Jeon (JASA, 2006).
  - Susan Athey and coauthors are big on random forests.

Tree as alternative to k-NN or kernel regression

- Figure from Athey and Imbens (2019), "Machine Learning Methods Economists should Know About"
  - axes are x_1 and x_2
  - note that tree used explanation of y in determining neighbors
  - tree may not do so well near boundaries of region
    - ★ random forests form many trees so not always at boundary.



Euclidean neighborhood, for KNN matching.



Tree-based neighborhood.

8. Classification

- $y's$ are now categorical e.g. binary.
- Interest lies in predicting y using $\widehat{y}$ (classification)
  - whereas economist typically want $\widehat{\Pr}[y = j|\mathbf{x}]$
  - use number misclassified as loss function (not MSE).
- Some methods choose category with highest $\widehat{\Pr}[y = j|\mathbf{x}]$
  - logit, k-nearest neighbors, discriminant analysis
- Support vector machines skip $\widehat{\Pr}[y = j|\mathbf{x}]$ and directly get $\widehat{y}$
  - can do better.

ISL Figure 9.9: Support Vector Machine

- Example with $y = 1$ blue and $y = 0$ red
  - a linear (logit or linear discriminant analysis) or quadratic classifier (quadratic DA) won't work whereas SVM does.

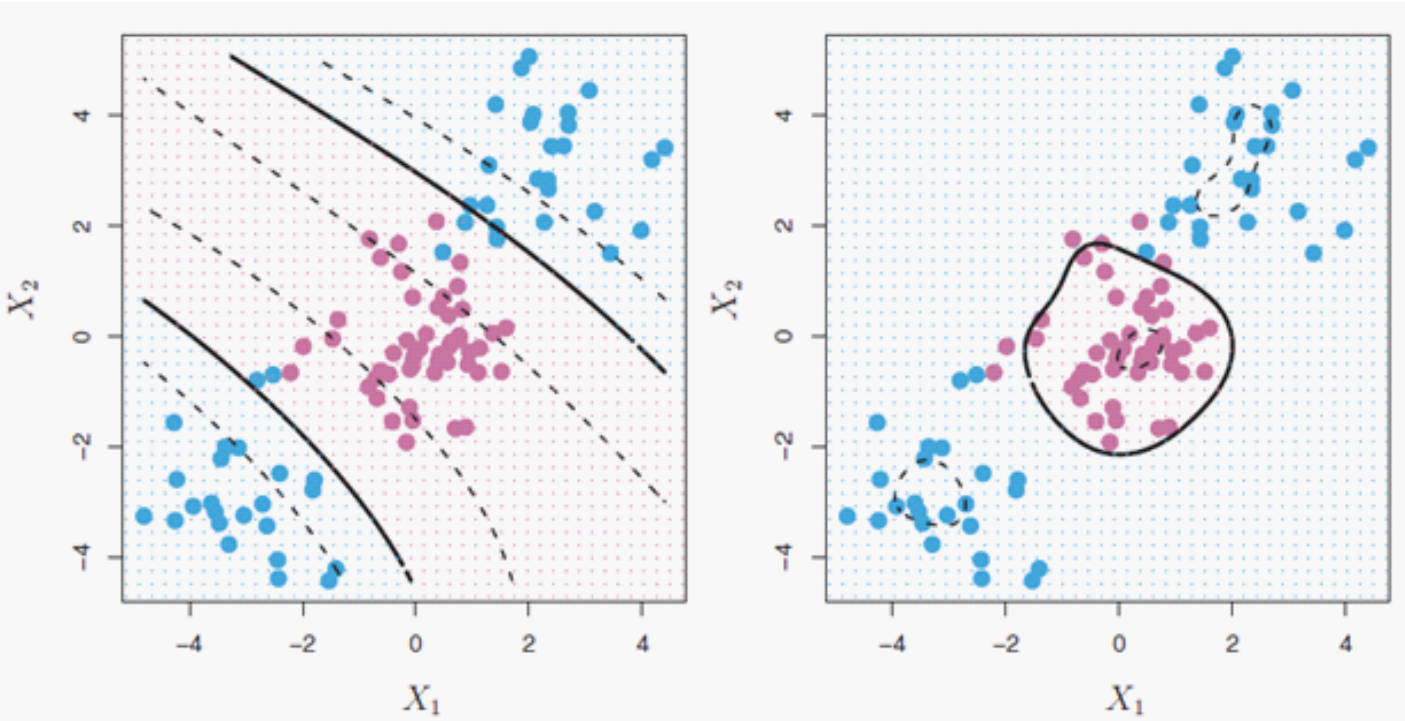


FIGURE 9.9. Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

9. Unsupervised Learning: cluster analysis

- Challenging area: no y, only $\mathbf{x}$.
- Example is determining several types of individual based on responses to many psychological questions.
- Principal components analysis
  - already presented earlier.
- Clustering Methods
  - k-means clustering.
  - hierarchical clustering.

ISL Figure 10.5

- Data is $(x_1.x_2)$ with $K = 2, 3$ and 4 clusters identified.

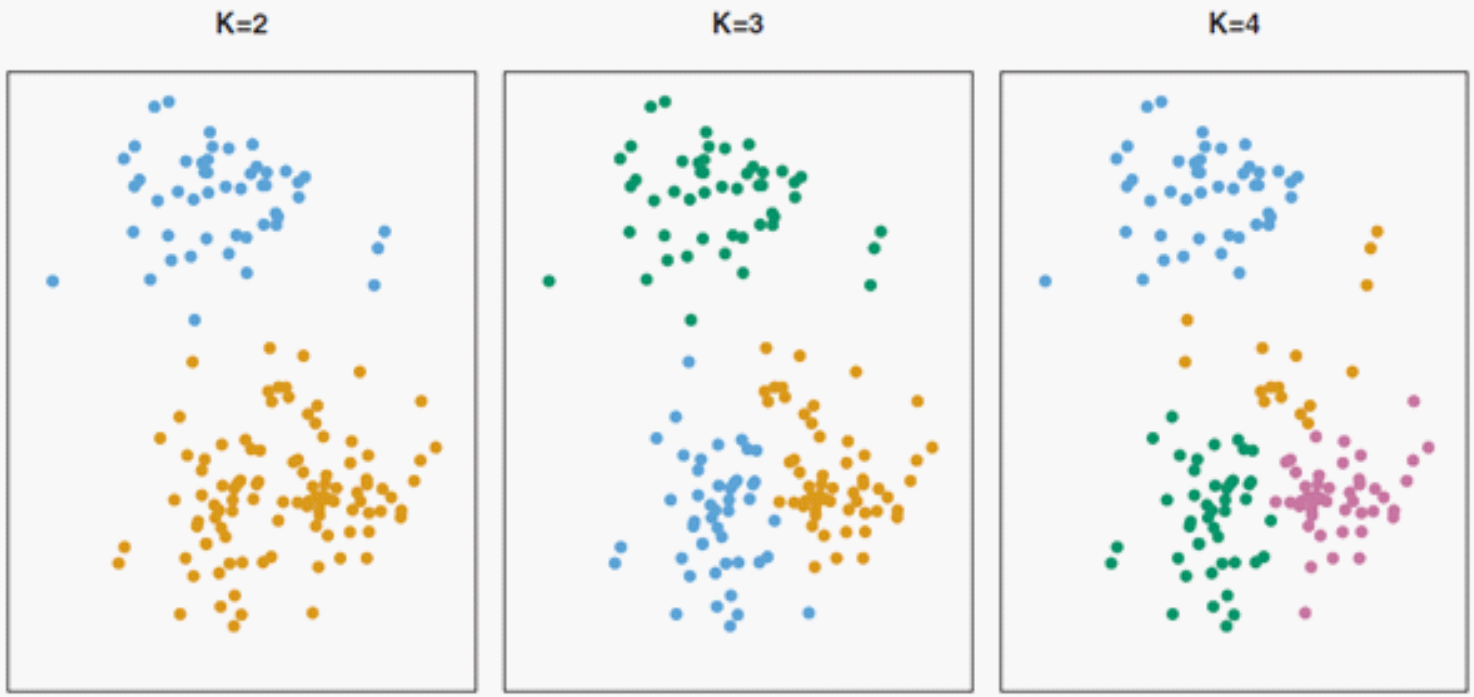


FIGURE 10.5. *A simulated data set with 150 observations in two-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. Note that*

10. Prediction for Economics: Mullainathan and Spiess

- Microeconometrics focuses on estimation of β or of partial effects.
- But in some cases we are directly interested in predicting y
  - probability of one-year survival following hip transplant operation
    - ★ if low then do not have the operation.
  - probability of re-offending
    - ★ if low then grant parole to prisoner.
- Mullainathan and Spiess (2017)
  - consider prediction of housing prices
  - detail how to do this using machine learning methods
  - and then summarize many recent economics ML applications.
- So summarize Mullainathan and Spiess (2017).

Summary of Machine Learning Algorithms

Table 2

Some Machine Learning Algorithms

Function class $\mathcal{F}$ (and its parametrization)	*Regularizer $R(f)$*
Global/parametric predictors	
Linear $\beta'x$ (and generalizations)	Subset selection $\|\|\beta\|\|_0 = \sum_{j=1}^{k} 1_{\beta_j \neq 0}$
	LASSO $\|\|\beta\|\|_1 = \sum_{j=1}^{k} \|\beta_j\|$
	Ridge $\|\|\beta\|\|_2^2 = \sum_{j=1}^{k} \beta_j^2$
	Elastic net $\alpha\|\|\beta\|\|_1 + (1-\alpha)\|\|\beta\|\|_2^2$
Local/nonparametric predictors	
Decision/regression trees	Depth, number of nodes/leaves, minimal leaf size, information gain at splits
Random forest (linear combination of trees)	Number of trees, number of variables used in each tree, size of bootstrap sample, complexity of trees (see above)
Nearest neighbors	Number of neighbors
Kernel regression	Kernel bandwidth

Table 2 (continued)

Mixed predictors	
Deep learning, neural nets, convolutional neural networks	Number of levels, number of neurons per level, connectivity between neurons
Splines	Number of knots, order
Combined predictors	
Bagging: unweighted average of predictors from bootstrap draws	Number of draws, size of bootstrap samples (and individual regularization parameters)
Boosting: linear combination of predictions of residual	Learning rate, number of iterations (and individual regularization parameters)
Ensemble: weighted combination of different predictors	Ensemble weights (and individual regularization parameters)

Example: Predict housing prices

- y is log house price in U.S. 2011
  - $n = 51,808$ is sample size
  - $p = 150$ is number of potential regressors.
- Predict using
  - OLS (using all regressors)
  - regression tree
  - LASSO
  - random forest
  - ensemble: an optimal weighted average of the above methods.
- 1. Train model on 10,000 observations using 8-fold CV.
- 2. Fit preferred model on these 10,000 observations.
- 3. Predict on remaining 41,808 observations
  - and do 500 bootstraps to get 95% CI for R^2.

- Random forest (and subsequent ensemble) does best out of sample.

Table 1

Performance of Different Algorithms in Predicting House Values

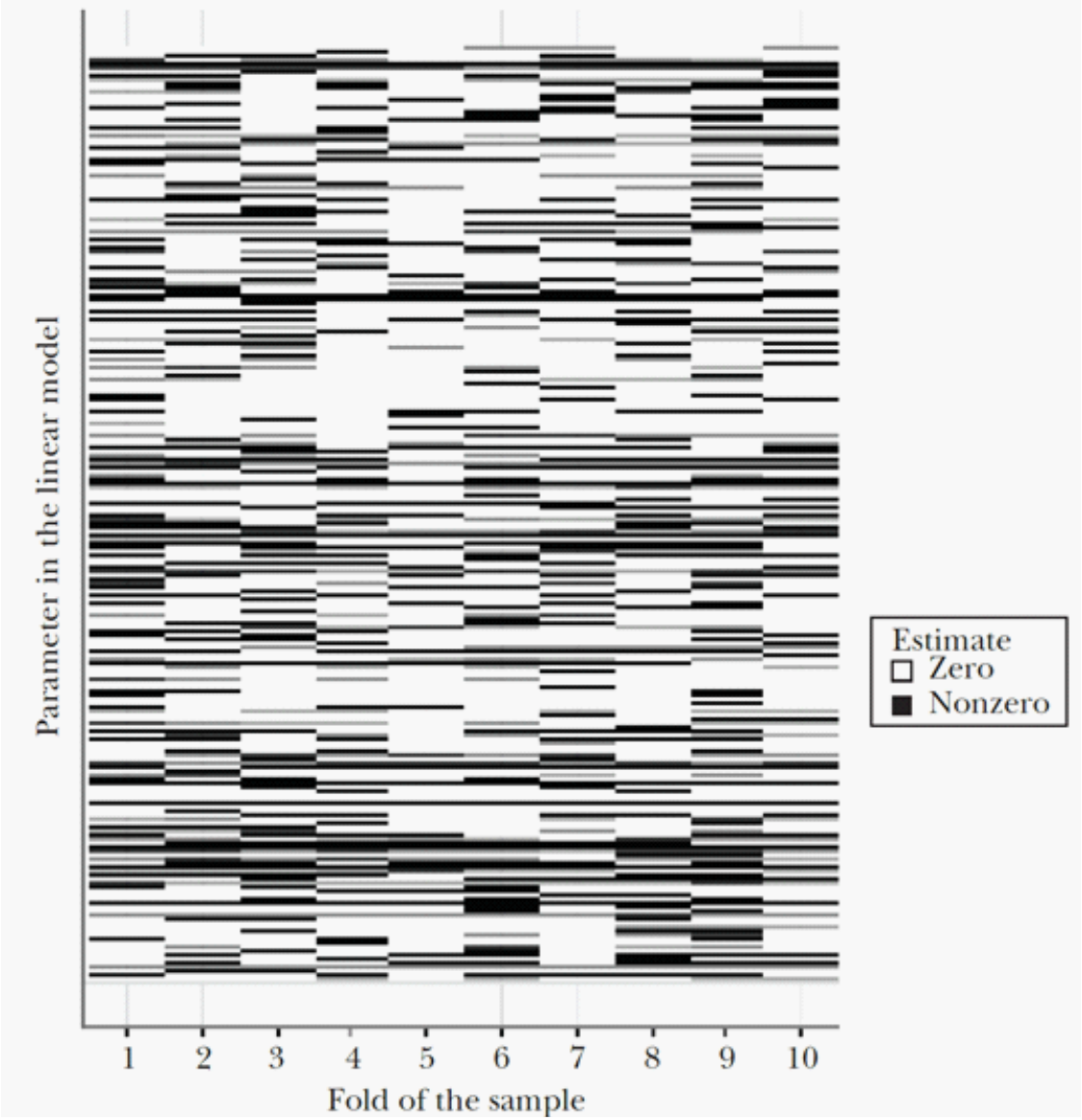
	Prediction performance (R^2)		*Relative improvement over ordinary least squares by quintile of house value*				
Method	*Training sample*	*Hold-out sample*	1st	2nd	3rd	4th	5th
Ordinary least squares	47.3%	41.7% [39.7%, 43.7%]	–	–	–	–	–
Regression tree tuned by depth	39.6%	34.5% [32.6%, 36.5%]	−11.5%	10.8%	6.4%	−14.6%	−31.8%
LASSO	46.0%	43.3% [41.5%, 45.2%]	1.3%	11.9%	13.1%	10.1%	−1.9%
Random forest	85.1%	45.5% [43.6%, 47.5%]	3.5%	23.6%	27.0%	17.8%	−0.5%
Ensemble	80.4%	45.9% [44.0%, 47.9%]	4.5%	16.0%	17.9%	14.2%	7.6%

Further details

- Downloadable appendix to the paper gives more details and R code.
- 1. Divide into training and hold-out sample.
- 2. On the training sample do 8-fold cross-validation to get tuning parameter(s) such as λ.
  - If e.g. two tuning parameters then do two-dimensional grid search.
- 3. The prediction function $\widehat{f}(x)$ is estimated using the entire sample with optimal λ.
- 4. Now apply this $\widehat{f}(x)$ to the hold-out sample and can compute R^2 and MSE.
- 5. A 95% CI for R^2 can be obtained by bootstrapping hold-out sample.
- Ensemble weights are obtained by 8-fold CV in the training sample.

LASSO

- LASSO does not pick the "correct" regressors
  - it just gets the correct $\widehat{f}(x)$ especially when regressors are correlated with each other.
- Diagram on next slide shows which of the 150 variables are included in separate models for 10 subsamples
  - there are many variables that appear sometimes but not at other times
    - ★ appearing sometimes in white and sometimes in black.


Parameter in the linear model
1
2
3
4
5
6
7
8
9
10
Fold of the sample
Estimate
Zero
Nonzero

Some Thoughts on ML Prediction

- Clearly there are many decisions to make in implementation
  - how are features converted into x's
  - tuning parameter values
  - which ML method to use
  - even more with an ensemble forecast.
- For commercial use this may not matter
  - all that matters is that predict well enough.
- But for published research we want reproducibility
  - At the very least document exactly what you did
  - provide all code (and data if it is publicly available)
  - keep this in mind at the time you are doing the project.
- For public policy we prefer some understanding of the black box
  - this may be impossible.

11. LASSO for causal homogeneous effects

- Here the basic model is $y = \beta x_1 + g(\mathbf{x}_2) + u$.
- Good choice of controls $g(\mathbf{x}_2)$ makes the unconfoundedness assumption that $Cov(x_1, u) = 0$ more plausible so can give β a plausible interpretation.
- Suppose $g(\mathbf{x}_2) \simeq \mathbf{w}'\gamma$
  - the $\mathbf{w}$ are various transformations of the $\mathbf{x}_2$ variables
  - so powers, interactions, logs, ...
  - this allows for nonlinearity in $g(\mathbf{x}_2)$.
- There are now many potential $\mathbf{w}$
  - assume only a few matter (most have $\gamma = 0$) - a sparsity assumption.
  - use LASSO to pick these.
- Double selection method
  - LASSO of y on $\mathbf{w}$ picks subset $\mathbf{w}_y$ of the $\mathbf{w}$ variables
  - LASSO of x_1 on $\mathbf{w}$ picks subset $\mathbf{w}_{x_1}$ of the $\mathbf{w}$ variables.
- $\widehat{\beta}$ obtained from OLS of y on x_1 and the union of $\mathbf{w}_y$ and $\mathbf{w}_{x_1}$
  - can use the usual asymptotic theory for $\widehat{\beta}$!

Key LASSO References

- Belloni, Chernozhukov and Hansen and coauthors have many papers.
- The following is accessible with three applications
  - Belloni, Chernozhukov and Hansen (2014), "High-dimensional methods and inference on structural and treatment effects," *Journal of Economic Perspectives*, Spring, 29-50
    - ★ has the preceding example and a many IV example.
- This gives more detail on LASSO methods as well as on Stata commands in the Stata add-on package lassopack
  - Ahrens, Hansen and Schaffer (2019), "lassopack: Model selection and prediction with regularized regression in Stata," arXiv:1901.05397.

Lassopack

- Consider a variant of LASSO with variable weights
  - useful for extension to heteroskedastic and clustered errors.
- The LASSO estimator $\widehat{\boldsymbol{\beta}}_\lambda$ of $\boldsymbol{\beta}$ minimizes

$$Q_\lambda(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i'\boldsymbol{\beta})^2 + \frac{\lambda}{n}\sum_{j=1}^{p}\psi_j|\beta_j|$$

  - where y_i and x_{ij} are demeaned so $\bar{y} = 0$ and $\bar{x}_j = 0$
  - and $\lambda \geq 0$ is a tuning parameter to be determined.
- Weights vary with errors homoskedastic, heteroskedastic or clustered.
- Tuning parameter λ determined in three different ways
  - `cvlasso` uses K-fold cross-validation
  - `lasso2` uses goodness-of-fit (AIC, BIC, AICC, EBIC)
  - `rlasso` uses user-specified value "theory-driven" or "rigorous"
    - ★ defaults are $c = 1.1$ and $\gamma = 0.1/\log(n)$
- `pdslasso` handles the above model $y = \beta x_1 + g(\mathbf{x}_2) + u$.
- `ivlasso` allows x_1 to be endogenous with potentially many instruments z.

Caution

- The LASSO methods are easy to estimate using the `lassopack` program
  - they'll be (blindly) used a lot.
- However in any application
  - is the underlying assumption of sparsity reasonable?
  - has the asymptotic theory kicked in?
  - are the default values of c and γ reasonable?

12. Heterogeneous treatment effects

- Consider a binary treatment, so $x_1 = d \in \{0, 1\}$
- The preceding partially linear model $y = \beta d + \mathbf{x}_2'\delta + u$
  - restricts the same response β for each individual
  - requires that $E[u|d, \mathbf{x}_2] = 0$ for unconfoundedness.
- The heterogeneous effects approach is more flexible
  - different responses for different individuals
  - and unconfoundness assumptions may be more reasonable.

Heterogeneous effects model

- Consider a binary treatment $d \in \{0, 1\}$
  - for some individuals we observe y only when $d = 1$ (treated)
  - for others we observe y only when $d = 0$ (untreated or control).
- Denote potential outcomes $y^{(1)}$ if $d = 1$ and $y^{(0)}$ if $d = 0$
  - for a given individual we observe only one of $y_i^{(1)}$ and $y_i^{(0)}$.
- The goal is to estimate the average treatment effect
  - ATE$= E[y_i^{(1)} - y_i^{(0)}]$
- The key assumption is the conditional independence assumption
  - $d_i \perp \{y_i^{(0)}, y_i^{(1)}\} | \mathbf{x}_i$.
  - conditional on $\mathbf{x}$, treatment is independent of the potential outcome
  - a good choice of $\mathbf{x}$ makes this assumption more reasonable.

ATE estimates

- ATE can be estimated in several ways
  - regression adjustment models $E[y^{(1)}|\mathbf{x}]$ and $E[y^{(0)}|\mathbf{x}]$
    - ⋆ then compute the average difference in predicted values
  - propensity score matching models $\Pr[d = 1|\mathbf{x}]$
    - ⋆ then compare $y^{(0)}$ and $y^{(1)}$ for people with similar propensity score
  - doubly-robust methods combine the two.
- Machine learning can help in getting good models for $E[y^{(1)}|\mathbf{x}]$ and $E[y^{(0)}|\mathbf{x}]$ and $\Pr[d = 1|\mathbf{x}]$
  - to date the LASSO is used.

Heterogeneous Effects using Random Forests

- Here the goal is to obtain the treatment effect at a given level of $\mathbf{x}$
  - and not just the overall average (ATE)
  - e.g. useful for customized treatment.
- Random forests predict very well
  - Susan Athey's research emphasizes random forests.
- Stefan Wager and Susan Athey (2018), "Estimation and Inference of Heterogeneous Treatment Effects using Random Forests," JASA, 1228-1242.
- Standard binary treatment and heterogeneous effects with unconfoundness assumption
  - use random forests to determine the controls.
  - proves asymptotic normality and gives point-wise confidence intervals
    - ★ This is a big theoretical contribution.

Heterogeneous Effects using Random Forests (continued)

- Let L denote a specific leaf in tree b.
- $\tau(\mathbf{x}) = E[y^{(1)} - y^{(0)}|\mathbf{x}]$ in a single regression tree b is estimated by

$$\begin{aligned}\widehat{\tau}_b(\mathbf{x}) &= \frac{1}{\#\{i:d_i=1,\mathbf{x}_i\in L\}}\sum_{i:d_i=1,\mathbf{x}_i\in L} y_i - \frac{1}{\#\{i:d_i=0,\mathbf{x}_i\in L\}}\sum_{i:d_i=0,\mathbf{x}_i\in L} y_i \\ &= \bar{y}_1 \text{ in leaf } L - \bar{y}_0 \text{ in leaf } L.\end{aligned}$$

- Then a random forest with sub-sample size s gives B trees with

$$\begin{aligned}\widehat{\tau}_b(\mathbf{x}) &= \frac{1}{B}\sum_{b=1}^{B}\widehat{\tau}_b(\mathbf{x}) \\ \widehat{Var}[\widehat{\tau}_b(\mathbf{x})] &= \frac{n-1}{n}\left(\frac{n}{n-2}\right)^2\sum_{i=1}^{n} Cov(\widehat{\tau}_b(\mathbf{x}), d_{ib})\end{aligned}$$

  - where $d_{ib} = 1$ if i^{th} observation in tree b and 0 otherwise
  - and the covariance is taken over all B trees.
- Key is that a tree is honest.
- A tree is honest if for each training observation i it only uses y_i to
  - either estimate $\widehat{\tau}(\mathbf{x})$within leaf
  - or to decide where to place the splits
  - but not both.

13. Double or Debiased Machine Learning

- Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey and Robins (2018), "Double/debiased machine learning for treatment and structural parameters," *The Econometrics Journal*.
- Interest lies in estimation of key parameter(s) controlling for high-dimensional nuisance parameters.
- There are two components to double ML or debiased ML and subsequent inference
  - work with **orthogonalized** moment conditions to estimate parameter(s) of interest.
  - use **sample splitting** (cross fitting) to remove bias induced by overfitting.
- This yields asymptotic normal distribution for parameters of interest
  - where a variety of ML methods can be used
    - ★ random forests, lasso, ridge, deep neural nets, boosted trees, ensembles.
- Can apply to partial linear model, ATE and ATET under unconfoundedness, LATE in an IV setting.

Orthogonalization defined

- Define $\boldsymbol{\beta}$ as parameters of interest and $\boldsymbol{\eta}$ as nuisance parameters.
- Estimate $\widehat{\boldsymbol{\beta}}$ is obtained following first step estimate $\widehat{\boldsymbol{\eta}}$ of $\boldsymbol{\eta}$
  - First stage: $\widehat{\boldsymbol{\eta}}$ solves $\sum_{i=1}^{n} \omega(\mathbf{w}_i, \boldsymbol{\eta}) = \mathbf{0}$ on 90% (say) of sample
  - Second stage: $\widehat{\boldsymbol{\beta}}$ solves $\sum_{i=1}^{n} \psi(\mathbf{w}_i, \boldsymbol{\beta}, \widehat{\boldsymbol{\eta}}) = \mathbf{0}$ on the other 10%.
- The distribution of $\widehat{\boldsymbol{\beta}}$ is usually affected by the noise due to estimating $\boldsymbol{\eta}$
  - e.g. Heckman's two-step estimator in selection models.
- But this is not always the case
  - e.g. the asymptotic distribution of feasible GLS is not affected by first-stage estimation of variance model parameters to get $\widehat{\Omega}$.
- Result: The distribution of $\widehat{\boldsymbol{\beta}}$ is unaffected by first-step estimation of $\boldsymbol{\eta}$ if the function $\psi(\cdot)$ satisfies
  - $E[\partial \psi(\mathbf{w}_i, \boldsymbol{\beta}, \boldsymbol{\eta}) / \partial \boldsymbol{\eta}] = \mathbf{0}$; see next slide.
- So choose functions $\psi(\cdot)$ that satisfy the orthogonalization condition

$$E[\partial \psi(\mathbf{w}_i, \boldsymbol{\beta}, \boldsymbol{\eta}) / \partial \boldsymbol{\eta}] = \mathbf{0}.$$

Orthogonalization (continued)

- Why does this work?

$$\begin{aligned}
&\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\psi(\mathbf{w}_i,\widehat{\boldsymbol{\beta}},\widehat{\boldsymbol{\eta}}) \\
= &\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\psi(\mathbf{w}_i,\boldsymbol{\beta}_0,\boldsymbol{\eta}_0)+\frac{1}{n}\sum_{i=1}^{n}\left.\frac{\partial\psi(\mathbf{w}_i,\boldsymbol{\beta},\boldsymbol{\eta})}{\partial\boldsymbol{\beta}'}\right|_{\boldsymbol{\beta}_0,\boldsymbol{\eta}_0}\times\sqrt{n}(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_0) \\
&+\frac{1}{n}\sum_{i=1}^{n}\left.\frac{\partial\psi(\mathbf{w}_i,\boldsymbol{\beta},\boldsymbol{\eta})}{\partial\boldsymbol{\eta}'}\right|_{\boldsymbol{\beta}_0,\boldsymbol{\eta}_0}\times\sqrt{n}(\widehat{\boldsymbol{\eta}}-\boldsymbol{\eta}_0)
\end{aligned}$$

- By a law of large numbers $\frac{1}{n}\sum_{i=1}^{n}\left.\frac{\partial\psi(\mathbf{w}_i,\boldsymbol{\beta},\boldsymbol{\eta})}{\partial\boldsymbol{\eta}}\right|_{\boldsymbol{\beta}_0,\boldsymbol{\eta}_0}$ converges to its expected value which is zero if $E[\partial\psi(\mathbf{w}_i,\boldsymbol{\beta},\boldsymbol{\eta})/\partial\boldsymbol{\eta}]=\mathbf{0}$.
- So the term involving $\widehat{\boldsymbol{\eta}}$ drops out.
- For more detail see e.g. Cameron and Trivedi (2005, p.201).

Orthogonalization in partially linear model

- Recall OLS of $y_i = \beta x_i + u_i$
  - solves sample moment condition $\sum_i x_i u_i = \sum_i x_i(y_i - \beta x_i) = 0$
  - with underlying population moment condition $E[x(y_i - \beta x)] = 0$.
- Partial linear model $y = \beta x_1 + g(\mathbf{x}_2) + u$.
- Robinson estimator is OLS in
  - $(y - E[y|\mathbf{x}_2]) = \beta(x_1 - E[x_1|\mathbf{x}_2]) + error$.
- So solve population moment condition $E[\psi(\cdot)] = 0$ where
  - $\psi(\cdot) = (x_1 - E[x_1|\mathbf{x}_2])\{y - E[y|\mathbf{x}_2] - \beta(x_1 - E[x_1|\mathbf{x}_2])\}$.
- Define $\eta_1 = E[x_1|\mathbf{x}_2]$ and $\eta_2 = E[y|\mathbf{x}_2]$, so
  - $\psi(w, \beta, \boldsymbol{\eta}) = (x_1 - \eta_1))\{y - \eta_2 - \beta(x_1 - \eta_1)\}$
- This satisfies the orthogonalization condition
  - $E[\partial\psi(\mathbf{w}, \beta, \boldsymbol{\eta})/\partial\eta_1] = E[2(x_1 - \eta_1)\beta] = 0$ as $\eta_1 = E[x_1|\mathbf{x}_2]$
  - $E[\partial\psi(\mathbf{w}, \beta, \boldsymbol{\eta})/\partial\eta_2] = E[-(x_1 - \eta_1)] = 0$ as $\eta_1 = E[x_1|\mathbf{x}_2]$.

Orthogonalization for doubly robust ATE (continued)

- Doubly-robust ATE solves $E[\psi(w, \tau, \boldsymbol{\eta})] = 0$ where
  - $\psi(w, \tau, \boldsymbol{\eta}) = \frac{\mathbf{1}[d=1](y-\eta_1)}{\eta_3} + \eta_1 - \frac{\mathbf{1}[d=0](y-\eta_2)}{1-\eta_3} - \eta_2 + \tau$
  - $\eta_1 = \mu_1(\mathbf{x}) = E[y_1|\mathbf{x}]$, $\eta_2 = \mu_0(\mathbf{x}) = E[y_0|\mathbf{x}]$, $\eta_3 = \Pr[y = 1|\mathbf{x}]$.
- This satisfies the orthogonalization condition
  - $E[\partial\psi(\mathbf{w}, \tau, \boldsymbol{\eta})/\partial\eta_1] = E[-\frac{\mathbf{1}[d=1])}{\eta_3} + 1] = 0$
    - ★ as $E[\mathbf{1}[d = 1]] = p(\mathbf{x}) = \eta_3$
  - $E[\partial\psi(\mathbf{w}, \tau, \boldsymbol{\eta})/\partial\eta_2] = E[\frac{\mathbf{1}[d=0])}{1-\eta_3} - 1] = 0$
    - ★ as $E[\mathbf{1}[d = 0]] = 1 - p(\mathbf{x}) = 1 - \eta_3$
  - $E[\partial\psi(\mathbf{w}, \tau, \boldsymbol{\eta})/\partial\eta_2] = E[-\frac{\mathbf{1}[d=1](y-\eta_1)}{\eta_3^2} - \frac{\mathbf{1}[d=0](y-\eta_2)}{(1-\eta_3)^2}] = 0 - 0 = 0$
    - ★ as $E[\mathbf{1}[d = 1](y - \eta_1)] = E[y_1|\mathbf{x}] - \eta_1 = 0$
    - ★ and $E[\mathbf{1}[d = 0](y - \eta_0)] = E[y_0|\mathbf{x}] - \eta_0 = 0$.

14. Conclusions

- Guard against overfitting
  - use K-fold cross validation or penalty measures such as AIC.
- Biased estimators can be better predictors
  - shrinkage towards zero such as Ridge and LASSO.
- For flexible models popular choices are
  - neural nets
  - random forests.
- Though what method is best varies with the application
  - and best are ensemble forecasts that combine different methods.
- Machine learning methods can outperform nonparametric and semiparametric methods
  - so wherever econometricians use nonparametric and semiparametric regression in higher dimensional models it may be useful to use ML methods
  - though the underlying theory still relies on assumptions such as sparsity.

15. References

- Undergraduate / Masters level book
  - **ISL:** Gareth James, Daniela Witten, Trevor Hastie and Robert Tibsharani (2013), *An Introduction to Statistical Learning: with Applications in R, Springer*.
  - free legal pdf at http://www-bcf.usc.edu/~gareth/ISL/
  - $25 hardcopy via http://www.springer.com/gp/products/books/mycopy
- Masters / PhD level book
  - **ESL:** Trevor Hastie, Robert Tibsharani and Jerome Friedman (2009), *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer.
  - free legal pdf at http://statweb.stanford.edu/~tibs/ElemStatLearn/index.html
  - $25 hardcopy via http://www.springer.com/gp/products/books/mycopy

References (continued)

- A recent book is
  - EH: Bradley Efron and Trevor Hastie (2016), *Computer Age Statistical Inference: Algorithms, Evidence and Data Science*, Cambridge University Press.
- Interesting book: Cathy O'Neil (2016), *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*.
- My website has some material (including these slides)
  - http://cameron.econ.ucdavis.edu/e240f/machinelearning.html

References (continued)

- Achim Ahrens, Christian Hansen, Mark Schaffer (2019), "lassopack: Model selection and prediction with regularized regression in Stata," arXiv:1901.05397
- Susan Athey (2018), "The Impact of Machine Learning on Economics". http://www.nber.org/chapters/c14009.pdf
- Susan Athey and Guido Imbens (2019), "Machine Learning Methods Economists Should Know About."
- Alex Belloni, Victor Chernozhukov and Christian Hansen (2011), "Inference Methods for High-Dimensional Sparse Econometric Models," *Advances in Economics and Econometrics*, ES World Congress 2010, ArXiv 2011.
- Alex Belloni, D. Chen, Victor Chernozhukov and Christian Hansen (2012), "Sparse Models and Methods for Optimal Instruments with an Application to Eminent Domain", *Econometrica*, Vol. 80, 2369-2429.

References (continued)

- Alex Belloni, Victor Chernozhukov and Christian Hansen (2014), "High-dimensional methods and inference on structural and treatment effects," *Journal of Economic Perspectives*, Spring, 29-50.
- Alex Belloni, Victor Chernozhukov, Ivan Fernandez-Val and Christian Hansen (2017), "Program Evaluation and Causal Inference with High-Dimensional Data," *Econometrica*, 233-299.
- Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey and James Robins (2018), "Double/debiased machine learning for treatment and structural parameters," *The Econometrics Journal*, 21, C1-C68.
- Max Farrell (2015), "Robust Estimation of Average Treatment Effect with Possibly more Covariates than Observations", *Journal of Econometrics*, 189, 1-23.
- Max Farrell, Tengyuan Liang and Sanjog Misra (2018), "Deep Neural Networks for Estimation and Inference: Application to Causal Effects and Other Semiparametric Estimands," arXiv:1809.09953v2.

References (continued)

- Jon Kleinberg, H. Lakkaraju, Jure Leskovec, Jens Ludwig, Sendhil Mullainathan (2018), "Human decisions and Machine Predictions", *Quarterly Journal of Economics*, 237-293.
- Sendhil Mullainathan and J. Spiess: "Machine Learning: Am Applied Econometric Approach", *Journal of Economic Perspectives*, Spring 2017, 87-106.
- Hal Varian (2014), "Big Data: New Tricks for Econometrics", *Journal of Economic Perspectives*, Spring, 3-28.
- Stefan Wager and Susan Athey (2018), "Estimation and Inference of Heterogeneous Treatment Effects using Random Forests," JASA, 1228-1242.