

Machine Learning for Microeconometrics

Part 2: Shrinkage estimators

A. Colin Cameron
Univ.of California - Davis

May 2022

Course Outline

- **1:** Variable selection and cross validation
- **Part 2: Shrinkage methods**
 - ▶ **ridge, lasso, elastic net**
- **3.** ML for causal inference using lasso
 - ▶ OLS with many controls, IV with many instruments
- **4.** Other methods for prediction
 - ▶ nonparametric regression, principal components, splines
 - ▶ neural networks
 - ▶ regression trees, random forests, bagging, boosting
- **5.** More ML for causal inference
 - ▶ ATE with heterogeneous effects and many controls.
- **6.** Classification and unsupervised learning
 - ▶ classification (categorical y) and unsupervised learning (no y).

1. Introduction

- Consider linear regression model with p potential regressors where p is too large.
- Methods that **reduce the model complexity** are
 - ▶ 1. choose a subset of regressors (previous slides)
 - ▶ 2. shrink regression coefficients towards zero (these slides)
 - ▶ 3. reduce the dimension of the regressors
 - ★ principal components analysis (later slides).
- Linear regression may predict well if include interactions and powers as potential regressors.
- And methods can be adapted to alternative loss functions for estimation.
- **Shrinkage** is also called **regularization**
 - ▶ lasso, ridge, elastic net.

Overview

- 1 Introduction
- 2 Shrinkage: Variance-bias trade-off
- 3 Shrinkage methods
 - 1 Ridge regression
 - 2 LASSO
 - 3 Elastic net
 - 4 Asymptotic Properties of Lasso
 - 5 Clustered data
- 4 Generated data
- 5 Prediction using LASSO, ridge and elasticnet
 - 1 Lasso command
 - 2 Lasso linear regression example
 - 3 Lasso postestimation commands example
 - 4 Adaptive lasso
 - 5 Elastic net and ridge regression
 - 6 Shrinkage for logit, probit and Poisson

2. Shrinkage: variance-bias trade-offs

- Consider prediction in the regression model

$$y = f(\mathbf{x}) + u \text{ with } E[u] = 0 \text{ and } u \perp \mathbf{x}.$$

- For out-of-estimation-sample point (y_0, \mathbf{x}_0) the true prediction error

$$E[(y_0 - \hat{f}(\mathbf{x}_0))^2] = \text{Var}[\hat{f}(\mathbf{x}_0)] + \{ \text{Bias}(\hat{f}(\mathbf{x}_0)) \}^2 + \text{Var}(u)$$

- The last term $\text{Var}(u)$ is called irreducible error
 - ▶ we can do nothing about this.
- So need to **minimize sum of variance and bias-squared!**
 - ▶ more flexible models have less bias (good) and more variance (bad).
 - ▶ this trade-off is fundamental to machine learning.

Variance-bias trade-off

- Shrinkage is one method that is biased but the bias may lead to lower squared error loss
 - ▶ first show this for estimation of a parameter θ
 - ▶ then show this for prediction of y .
- The mean squared error of a scalar estimator $\tilde{\theta}$ is

$$\begin{aligned}
 \text{MSE}(\tilde{\theta}) &= E[(\tilde{\theta} - \theta)^2] \\
 &= E[\{(\tilde{\theta} - E[\tilde{\theta}]) + (E[\tilde{\theta}] - \theta)\}^2] \\
 &= E[(\tilde{\theta} - E[\tilde{\theta}])^2] + (E[\tilde{\theta}] - \theta)^2 + 2 \times 0 \\
 &= \text{Var}(\tilde{\theta}) + \text{Bias}^2(\tilde{\theta})
 \end{aligned}$$

- ▶ as the cross product term $2 \times E[(\tilde{\theta} - E[\tilde{\theta}])(E[\tilde{\theta}] - \theta)] =$
 $\text{constant} \times E[(\tilde{\theta} - E[\tilde{\theta}])] = 0.$

Bias can reduce estimator MSE: a shrinkage example

- Suppose scalar estimator $\hat{\theta}$ is unbiased for θ with
 - ▶ $E[\hat{\theta}] = \theta$ and $Var[\hat{\theta}] = v$
 - ▶ So $MSE(\hat{\theta}) = v$.
- Construct the shrinkage estimator $\tilde{\theta} = a\hat{\theta}$ where $0 \leq a \leq 1$.
 - ▶ Bias: $Bias(\tilde{\theta}) = E[\tilde{\theta}] - \theta = a\theta - \theta = (a - 1)\theta$.
 - ▶ Variance: $Var[\tilde{\theta}] = Var[a\hat{\theta}] = a^2 Var(\hat{\theta}) = a^2 v$
 - ▶ So $MSE(\tilde{\theta}) = Var[\tilde{\theta}] + Bias^2(\tilde{\theta}) = a^2 v + (a - 1)^2 \theta^2$.

Shrinkage example continued

- So we have

$$\begin{array}{ll} \text{Unbiased} & \hat{\theta} & \text{MSE}(\hat{\theta}) = v \\ \text{Biased} & \tilde{\theta} = a\hat{\theta} & \text{MSE}(\tilde{\theta}) = a^2v + (a-1)^2\theta^2 \end{array}$$

- Then $\text{MSE}(\tilde{\theta}) < \text{MSE}[\hat{\theta}]$ if $\theta^2 < \frac{1+a}{1-a}v$
 - ▶ e.g. if $\tilde{\theta} = 0.9\hat{\theta}$ then $\tilde{\theta}$ has lower MSE for $\theta^2 < 19v!$
- We will consider
 - ▶ ridge estimator shrinks towards zero
 - ▶ LASSO estimator selects and shrinks towards zero.

James-Stein estimator

- Suppose $y_i \sim N(\mu_i, 1)$, $i = 1, \dots, n$.
- The MLE is $\hat{\mu}_i = y_i$ with $MSE(\hat{\mu}_i) = 1$.
- The James-Stein estimator is $\tilde{\mu}_i = (1 - c)y_i + c\bar{y}$
 - ▶ where $c = \frac{1}{n-3} \sum_{i=1}^n (y_i - \bar{y})^2$ and $n \geq 4$
 - ▶ this shrinks towards the sample mean \bar{y}
- Then $MSE(\tilde{\mu}_i) < MSE(\hat{\mu}_i)$ for $n \geq 4$!
- This remarkable 1950's/1960's result was a big surprise
 - ▶ an estimator has lower MSE than the maximum likelihood estimator.
- The estimator can be given an empirical Bayes interpretation.

Bias can therefore reduce predictor MSE

- Now consider prediction of $y_0 = \beta x_0 + u$ where $E[u] = 0$
 - ▶ using $\tilde{y}_0 = \tilde{\beta} x_0$ where treat scalar x_0 as fixed.
- Bias: $Bias(\tilde{y}_0) = E[x_0 \tilde{\beta}] - \beta x_0 = x_0 (E[\tilde{\beta}] - \beta) = x_0 Bias(\tilde{\beta})$.
- Variance: $Var[\tilde{y}_0] = Var[x_0 \tilde{\beta}] = x_0^2 Var(\tilde{\beta})$.
- The mean squared error in the scalar regressor case is

$$\begin{aligned}
 MSE(\tilde{y}_0) &= Var(\tilde{y}_0) + Bias^2(\tilde{y}_0) + Var(u) \\
 &= x_0^2 Var(\tilde{\beta}) + (x_0 Bias(\tilde{\beta}))^2 + Var(u) \\
 &= x_0^2 \{ Var(\tilde{\beta}) + Bias^2(\tilde{\beta}) \} + Var(u) \\
 &= x_0^2 MSE(\tilde{\beta}) + Var(u).
 \end{aligned}$$

- So bias in $\tilde{\beta}$ that reduces $MSE(\tilde{\beta})$ also reduces $MSE(\tilde{y}_0)$.

3. Shrinkage Methods

- Shrinkage estimators minimize RSS (residual sum of squares) with a penalty for model size
 - ▶ this shrinks parameter estimates towards zero.
- The extent of shrinkage is determined by a **tuning parameter**
 - ▶ this is determined by cross-validation or penalty such as AIC.
- **Standardize regressors** as ridge, LASSO and elastic net are not invariant to rescaling of regressors
 - ▶ so x_{ij} below is actually $(x_{ij} - \bar{x}_j)/s_j$
 - ▶ and demean y_i so below y_i is actually $y_i - \bar{y}$
 - ▶ \mathbf{x}_i does not include an intercept nor does data matrix \mathbf{X}
 - ▶ we can recover intercept β_0 as $\hat{\beta}_0 = \bar{y}$.
- So work with $y = \mathbf{x}'\boldsymbol{\beta} + \varepsilon = \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \varepsilon$.

3.1 Ridge Regression

- The simplest form of the **ridge estimator** $\hat{\beta}_\lambda$ of β minimizes

$$Q_\lambda(\beta) = \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{x}'_i \beta)^2 + \lambda \sum_{j=1}^p \beta_j^2 = \text{RSS} + \lambda (\|\beta\|_2)^2$$

- where $\lambda \geq 0$ is a tuning parameter to be determined
- $\|\beta\|_2 = \sqrt{\sum_{j=1}^p \beta_j^2}$ is L2 norm.

- Equivalently the ridge estimator minimizes

$$\frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{x}'_i \beta)^2 \text{ subject to } \sum_{j=1}^p \beta_j^2 \leq s.$$

- The ridge estimator is

$$\hat{\beta}_\lambda = (\mathbf{X}'\mathbf{X} + n\lambda\mathbf{I})^{-1} \mathbf{X}'\mathbf{y}.$$

- More generally can weight each β_j

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

Ridge Derivation

- 1. Objective function includes penalty

- ▶ $Q(\boldsymbol{\beta}) = \frac{1}{n}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}'\boldsymbol{\beta}$
- ▶ $\partial Q(\boldsymbol{\beta})/\partial\boldsymbol{\beta} = -\frac{2}{n}\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + 2\lambda\boldsymbol{\beta} = \mathbf{0}$
- ▶ $\Rightarrow \mathbf{X}'\mathbf{X}\boldsymbol{\beta} + \lambda\mathbf{I}\boldsymbol{\beta} = \mathbf{X}'\mathbf{y}$
- ▶ $\Rightarrow \widehat{\boldsymbol{\beta}}_{\lambda} = (\mathbf{X}'\mathbf{X} + n\lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}.$

- 2. Form Lagrangian (multiplier is λ) from objective function and constraint

- ▶ $Q(\boldsymbol{\beta}) = \frac{1}{n}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ and constraint $\boldsymbol{\beta}'\boldsymbol{\beta} \leq s$
- ▶ $L(\boldsymbol{\beta}, \lambda) = \frac{1}{n}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda(\boldsymbol{\beta}'\boldsymbol{\beta} - s)$
- ▶ $\partial L(\boldsymbol{\beta}, \lambda)/\partial\boldsymbol{\beta} = -\frac{2}{n}\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + 2\lambda\boldsymbol{\beta} = \mathbf{0}$
- ▶ $\Rightarrow \widehat{\boldsymbol{\beta}}_{\lambda} = (\mathbf{X}'\mathbf{X} + n\lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$
- ▶ Here $\lambda = \partial L_{opt}(\boldsymbol{\beta}, \lambda, s)/\partial s.$

Ridge Properties

- $\hat{\beta}_\lambda \rightarrow \mathbf{0}$ as $\lambda \rightarrow \infty$ and $\hat{\beta}_\lambda \rightarrow \hat{\beta}_{OLS}$ as $\lambda \rightarrow 0$.
- Ridge best when many predictors important with coeffs of similar size.
- Ridge best when LS has high variance
 - ▶ meaning small changes in training data can lead to large changes in OLS coefficient estimates.
- Algorithms exist to quickly compute $\hat{\beta}_\lambda$ for many values of λ
 - ▶ then choose λ by cross validation.
 - ▶ with search over a decreasing logarithmic grid in λ .

More on Ridge

- Also called Tikhonov regularization.
- Hoerl & Kennard (1970) proposed ridge as a way to reduce MSE of $\hat{\beta}$.
- We can write ridge as $\hat{\beta}_\lambda = (\mathbf{X}'\mathbf{X} + n\lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{X}\hat{\beta}_{OLS}$
 - ▶ so shrinkage of OLS toward zero.
- For scalar regressor and no intercept $\hat{\beta}_\lambda = a\hat{\beta}_{OLS}$ where $a = \frac{\sum_i x_i^2}{\sum_i x_i^2 + n\lambda}$
 - ▶ like earlier example of $\tilde{\beta} = a\hat{\beta}$.
- Ridge is the posterior mean for $\mathbf{y} \sim N(\mathbf{X}\beta, \sigma^2\mathbf{I})$ with prior $\beta \sim N(0, \gamma^2\mathbf{I})$
 - ▶ though γ is a specified prior parameter whereas λ is data-determined.
- Ridge is estimator in model $\mathbf{y} \sim (\mathbf{X}\beta, \sigma^2\mathbf{I})$ with stochastic constraints $\beta \sim (0, \gamma^2\mathbf{I})$.

3.2 LASSO (Least Absolute Shrinkage And Selection Operator)

- The **LASSO estimator** $\hat{\beta}_\lambda$ of β minimizes

$$Q_\lambda(\beta) = \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{x}'_i \beta)^2 + \lambda \sum_{j=1}^p |\beta_j| = \text{RSS} + \lambda \|\beta\|_1$$

- where $\lambda \geq 0$ is a tuning parameter to be determined
 - $\|\beta\|_1 = \sum_{j=1}^p |\beta_j|$ is L1 norm.
- Equivalently the LASSO estimator minimizes

$$\frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{x}'_i \beta)^2 \text{ subject to } \sum_{j=1}^p |\beta_j| \leq s.$$

- Features
 - best when a few regressors have $\beta_j \neq 0$ and most $\beta_j = 0$
 - leads to a more interpretable model than ridge.
- More generally $Q_\lambda(\beta) = \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{x}'_i \beta)^2 + \lambda \sum_{j=1}^p \kappa_j |\beta_j|$.

LASSO versus Ridge (key figure from ISL)

- LASSO is likely to set some coefficients to zero.

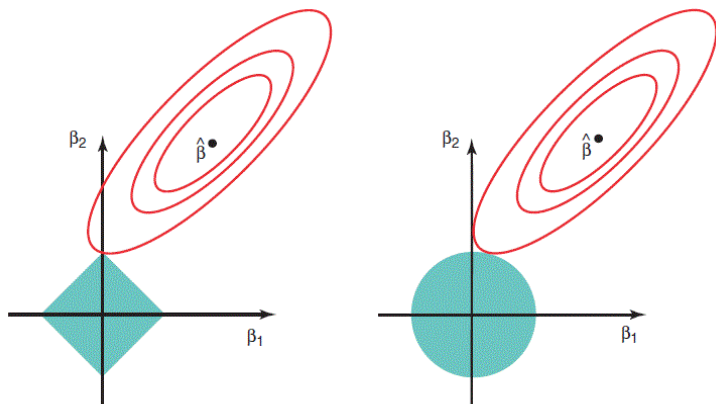


FIGURE 6.7. Contours of the error and constraint functions for the lasso (left) and ridge regression (right). The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \leq s$ and $\beta_1^2 + \beta_2^2 \leq s$, while the red ellipses are the contours of the RSS.

LASSO versus Ridge

- Consider simple case where $n = p$ and $\mathbf{X} = \mathbf{I}_n$ (identity matrix).
- OLS: $\hat{\boldsymbol{\beta}}^{OLS} = (\mathbf{I}'\mathbf{I})^{-1}\mathbf{I}'\mathbf{y} = \mathbf{y}$ so $\hat{\beta}_j^{OLS} = y_j$
- Ridge shrinks all β_j 's towards zero

$$\hat{\boldsymbol{\beta}}^R = (\mathbf{I}'\mathbf{I} + \lambda\mathbf{I})^{-1}\mathbf{I}'\mathbf{y} = \mathbf{y}/(1 + \lambda)$$

$$\hat{\beta}_j^R = y_j/(1 + \lambda)$$

- LASSO shrinks some β_j 's towards 0 and sets others = 0

$$\hat{\beta}_j^L = \begin{cases} y_j - \lambda/2 & \text{if } y_j > \lambda/2 \\ y_j + \lambda/2 & \text{if } y_j < -\lambda/2 \\ 0 & \text{if } |y_j| \leq \lambda/2 \end{cases}$$

- Aside: Best subset of size M in this example

$$\hat{\boldsymbol{\beta}}^{BS} = \hat{\boldsymbol{\beta}} \times \mathbf{1}[|\hat{\beta}_j| \geq |\hat{\beta}_{(M)}|]$$

where $\hat{\beta}_{(M)}$ is the M^{th} largest OLS coefficient.

Computation of LASSO estimator

- Most common is a coordinate wise descent algorithm
 - ▶ also called a shooting algorithm due to Fu (1998)
 - ▶ exploits the special structure in the nondifferentiable part of the LASSO objective function that makes convergence possible.
- The algorithm for given λ (λ is later chosen by CV)
 - ▶ denote $\boldsymbol{\beta} = (\beta_j, \boldsymbol{\beta}^{-j})$ and define $S_j(\beta_j, \boldsymbol{\beta}^{-j}) = \partial \text{RSS} / \partial \beta_j$
 - ▶ start with $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}_{OLS}$
 - ▶ at step m for each $j = 1, \dots, p$ let $S_0 = S_j(0, \hat{\boldsymbol{\beta}}^{-j})$ and set

$$\hat{\beta}_j = \begin{cases} \frac{\lambda - S_0}{2\mathbf{x}'_j \mathbf{x}_j} & \text{if } S_0 > \lambda \\ \frac{-\lambda - S_0}{2\mathbf{x}'_j \mathbf{x}_j} & \text{if } S_0 < -\lambda \\ 0 & \text{if } -\lambda \leq S_0 \leq \lambda \end{cases}$$

- ▶ form new $\hat{\boldsymbol{\beta}}_m = [\hat{\beta}_1 \cdots \hat{\beta}_p]$ after updating all $\hat{\beta}_j$.
- Alternatively LASSO is a minor adaptation of least angle regression
 - ▶ so estimate using the forward-stagewise algorithm for LAR.

More on LASSO

- LASSO is due to Tibshirani (1999).
- Can weight each β_j differently
 - ▶ implemented in Stata `lasso` commands.
- Can specify some variables to be always included.
- The group lasso allows to include regressors as groups (e.g. race dummies as a group)
 - ▶ with L groups minimize over β

$$\frac{1}{n} \sum_{i=1}^n \left(y_i - \sum_{l=1}^L \mathbf{x}_i' \beta_l \right)^2 + \lambda \sum_{l=1}^L \sqrt{p_l} \left(\sum_{j=1}^{p_l} |\beta_{lj}| \right).$$

- There are other extensions such as adaptive LASSO - LASSO is popular.
- Giannone, Lenza and Primiceri (2021) find that sparse models predict poorly in several standard economic applications.

3.3 Elastic net

- Elastic net combines ridge regression and LASSO with objective function

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

- ▶ ridge penalty λ averages correlated variables
 - ▶ LASSO penalty α leads to sparsity.
- For elastic net
 - ▶ Ridge is special case $\alpha = 0$
 - ▶ LASSO is special case $\alpha = 1$.
- K-fold cross validation is used with default $K = 10$
 - ▶ set seed for replicability.

3.4 Asymptotic Properties of Lasso

- **A model selection method is consistent** if asymptotically it correctly selects the correct model from a selection of candidate models
 - ▶ selecting on basis of minimum BIC is consistent.
- **A model selection method is conservative** if asymptotically it always selects a model that nests the correct model
 - ▶ selecting a model on the basis of minimum AIC is conservative.
 - ▶ Hannes Leeb and Benedikt M. Pötscher (2005), “Model Selection and Inference”, *Econometric Theory*, 21-59.
- A statistical model selection and estimation method is said to have an **oracle property** if it leads to consistent model selection and a subsequent estimator that is asymptotically equivalent to the estimator that could be obtained if the true model was known so that model selection was unnecessary.

Asymptotic properties of Lasso

- The LASSO is a consistent model selection procedure
 - ▶ but does not have oracle property due to bias.
- The oracle property is an asymptotic property
 - ▶ not useful in finite sample settings that economists encounter
 - ★ our models do not fit perfectly
 - ▶ and gives rates for a penalty parameters but not finite sample value.
- Lasso estimates have complicated finite sample distribution.
 - ▶ cannot perform standard inference on LASSO or post_LASSO
 - ▶ instead add some model structure
 - ★ e.g. partial linear model.

3.5 Clustered Data

- Now consider clustered data
 - ▶ by clustered data I mean “clustered errors”
 - ▶ data are grouped with correlated observations within group and uncorrelated across groups
 - ▶ examples are panel data and grouping by independent regions.
- Notation: y_{ig} is outcome for individual i in cluster g , $i = 1, \dots, N_g$, $g = 1, \dots, G$.
- Here focus on lasso as the machine learning method.

Lasso with Clustered Data

- We want to generalize $Q_\lambda(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{x}'_i \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^p |\beta_j|$.
- **Method 1.** With clustered data one can continue with this in which case equal weight is given to each observation

$$Q_\lambda(\boldsymbol{\beta}) = \frac{1}{G} \sum_{g=1}^G \sum_{i=1}^{N_g} (y_{ig} - \mathbf{x}'_{ig} \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^p |\beta_j|$$

- **Method 2.** Alternatively one can give equal weight to each cluster

$$Q_\lambda(\boldsymbol{\beta}) = \frac{1}{G} \sum_{g=1}^G \frac{1}{N_g} \sum_{i=1}^{N_g} (y_{ig} - \mathbf{x}'_{ig} \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^p |\beta_j|$$

- Stata 17 option `cluster(clustervar)` of the `lasso` command does method 2.
- Which is best?
 - ▶ If data are independent within cluster then 1.?
 - ▶ If data are perfectly correlated within cluster then 2.?
 - ▶ And a big difference if cluster sizes are greatly unbalanced.

Key papers

- Peter Bickel, Ya'acov Ritov and Alexandre Tsybakov (2009), “Simultaneous Analysis of Lasso and Dantzig Selector”, *The Annals of Statistics*, 1705-1732.
 - ▶ In reading this just consider Lasso (ignore Dantzig Selector).
 - ▶ lays out the typical assumptions well
 - ★ including sparsity and $y_i = f(\mathbf{z}_i) + u_i$, u_i i.i.d. $N(0, \sigma^2)$
 - ▶ lays out finite sample bounds for prediction loss
 - ★ key are assumptions on the eigenvalues of the Gram matrix $\mathbf{X}'\mathbf{X}$
 - ★ similar to restrictions on the correlations among regressors.
- Alex Belloni and Victor Chernozhukov (2013), “Least Squares after Model Selection in High-Dimensional Sparse Models”, *Bernoulli*, 521-547.
 - ▶ builds on the previous paper
 - ▶ harder to read, so read Bickel et al first
 - ▶ OLS after LASSO is better than prediction from LASSO estimates.

4. Prediction using LASSO: Stata lasso command

- `lasso model depvar [(alwaysvars)] othervars, options`
- Model is
 - ▶ linear, logit, probit or poisson
- `folds(#)`
- penalty parameter λ
 - ▶ cross validation (`selection(cv)`) sets all $\kappa_j = 1$
 - ▶ adaptive cv (`selection(adaptive cv)`) κ_j can vary
 - ▶ AIC (`bic`)
 - ▶ plug-in (`selection(plugin)`) for non-prediction applications

Postestimation commands

- lasso command focuses on finding λ
- Following commands give more info
 - ▶ `lassoknots`
 - ▶ `lassoselect`
 - ▶ `cvplot`
 - ▶ `coefpath`
 - ▶ `lassoinfo`
 - ▶ `lassocoeff`
 - ▶ `lassogof`

4.2 LASSO linear regression example

- Generated data example: $n = 40$, $p = 3$.
- Three correlated regressors.

$$\blacktriangleright \begin{bmatrix} x_{1j} \\ x_{2j} \\ x_{3j} \end{bmatrix} \sim N \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{bmatrix} \right)$$

- But only x_1 determines y
 - $y = 2 + x_{1j} + u_j$ where $u_j \sim N(0, 3^2)$.
- Same generate data as in part 1 slides.

```
. * Summarize data
. summarize
```

Variable	Obs	Mean	Std. Dev.	Min	Max
x1	40	.3337951	.8986718	-1.099225	2.754746
x2	40	.1257017	.9422221	-2.081086	2.770161
x3	40	.0712341	1.034616	-1.676141	2.931045
y	40	3.107987	3.400129	-3.542646	10.60979

```
. correlate
(obs=40)
```

	x1	x2	x3	y
x1	1.0000			
x2	0.5077	1.0000		
x3	0.4281	0.2786	1.0000	
y	0.4740	0.3370	0.2046	1.0000

Aside: Demeaning data

- Stata commands such as `lasso` do this automatically
 - ▶ but for completeness following code demeans.

```
. * Standardize regressors and demean y
. foreach var of varlist x1 x2 x3 {
2.     qui egen z`var' = std(`var')
3.     }

. quietly summarize y

. quietly generate ydemeaned = y - r(mean)

. summarize ydemeaned z*
```

Variable	Obs	Mean	Std. Dev.	Min	Max
ydemeaned	40	-1.71e-08	3.400129	-6.650633	7.501798
zx1	40	2.05e-09	1	-1.594598	2.693921
zx2	40	2.79e-10	1	-2.34211	2.80662
zx3	40	2.79e-09	1	-1.688912	2.764129

- The original variables x_1 to x_3 had standard deviations 0.89867, 0.94222 and 1.03462
 - ▶ means differ from zero due to single precision rounding error.

Demeaning data better

- Aside: Use double precision

```
. * Standardize regressors and demean y
. foreach var of varlist x1 x2 x3 {
  2.     qui egen double z`var' = std(`var')
  3. }

. qui summarize y

. qui generate double ydemeaned = y - r(mean)

. summarize ydemeaned z*
```

Variable	Obs	Mean	Std. Dev.	Min	Max
ydemeaned	40	-3.33e-17	3.400129	-6.650633	7.501798
zx1	40	2.63e-17	1	-1.594598	2.693921
zx2	40	2.62e-17	1	-2.34211	2.80662
zx3	40	-2.98e-17	1	-1.688912	2.764129

- Stata does internal calculations in double precision but default is to save variables in single precision.

LASSO linear regression example: lasso command

- Apply to generated data example: $n = 40$, $K = 5$, $p = 3$.
 - ▶ **set the seed !**
 - ▶ First regressor selected when $\lambda = 1.450138$

```
. * Lasso linear using 5-fold cross validation
. lasso linear y x1 x2 x3, selection(cv) folds(5) rseed(10101)

5-fold cross-validation with 100 lambdas ...
Grid value 1:      lambda = 1.591525   no. of nonzero coef. =      0
Folds: 1...5     CVF = 11.85738
Grid value 2:      lambda = 1.450138   no. of nonzero coef. =      1
Folds: 1...5     CVF = 11.60145
Grid value 3:      lambda = 1.321312   no. of nonzero coef. =      1
Folds: 1...5     CVF = 11.2296
Grid value 4:      lambda = 1.20393    no. of nonzero coef. =      1
Folds: 1...5     CVF = 10.87719
Grid value 5:      lambda = 1.096976    no. of nonzero coef. =      1
Folds: 1...5     CVF = 10.60149
Grid value 6:      lambda = .9995238   no. of nonzero coef. =      1
Folds: 1...5     CVF = 10.38463
Grid value 7:      lambda = .9107289   no. of nonzero coef. =      1
Folds: 1...5     CVF = 10.20522
Grid value 8:      lambda = .8298222   no. of nonzero coef. =      1
Folds: 1...5     CVF = 10.05685
```

lasso command (continued)

- Second regressor included when $\lambda = 0.6277301$

```

Grid value 9:      lambda = .7561031    no. of nonzero coef. =    1
Folds: 1...5    CVF = 9.934201
Grid value 10:     lambda = .688933    no. of nonzero coef. =    1
Folds: 1...5    CVF = 9.829713
Grid value 11:     lambda = .6277301    no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.739804
Grid value 12:     lambda = .5719643    no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.666469
Grid value 13:     lambda = .5211525    no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.606777
Grid value 14:     lambda = .4748548    no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.562824
Grid value 15:     lambda = .43267      no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.525748
Grid value 16:     lambda = .3942328    no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.493472
Grid value 17:     lambda = .3592102    no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.460115
Grid value 18:     lambda = .327299    no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.43311
Grid value 19:     lambda = .2982226    no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.411316

```

lasso command (continued)

- Minimum CV of 9.393523 with two regressors.

```

Grid value 20:    lambda = .2717294    no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.393794
Grid value 21:    lambda = .2475897    no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.393523
Grid value 22:    lambda = .2255945    no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.40661
Grid value 23:    lambda = .2055533    no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.420332
Grid value 24:    lambda = .1872925    no. of nonzero coef. =    2
Folds: 1...5    CVF = 9.434326
... cross-validation complete ... minimum found

```

- Default grid search is a decreasing logarithmic grid of 100 values
 - $\lambda_j = \lambda_1 \times 10^{-4(j-1)/99}$, $j = 2, \dots, 100$
 - $\lambda_1 = 1.591525$ is the smallest value at which no values are selected.

lasso command (continued)

- Final results on optimal λ .

```
Lasso linear model                No. of obs      =      40
                                  No. of covariates =      3
Selection: Cross-validation       No. of CV folds =      5
```

ID	Description	lambda	No. of nonzero coef.	Out-of- sample R-squared	CV mean prediction error
1	first lambda	1.591525	0	-0.0519	11.85738
20	lambda before	.2717294	2	0.1666	9.393794
* 21	selected lambda	.2475897	2	0.1666	9.393523
22	lambda after	.2255945	2	0.1655	9.40661
24	last lambda	.1872925	2	0.1630	9.434326

* lambda selected by cross-validation.

lassoknots command

- Lists values of λ at which variables are added and removed
 - ▶ here first x_1 and then x_2 are added.

```
. * List the values of lambda at which variables are added or removed
. lassoknots
```

ID	lambda	No. of nonzero coef.	CV mean pred. error	Variables (A)dded, (R)emoved, or left (U)nchanged
2	1.450138	1	11.60145	A x1
11	.6277301	2	9.739804	A x2
* 21	.2475897	2	9.393523	U
24	.1872925	2	9.434326	U

* lambda selected by cross-validation.

cvplot command

- Plot value of CV_5 against λ on log scale
 - ▶ simply command `cvplot`
- Plot of how coefficients change with λ
 - ▶ command `coefpath`

lassoinfo command

- Provides a summary of the LASSO.

```
. * Provide a summary of the lasso
. lassoinfo
```

```
Estimate: active
Command: lasso
```

Depvar	Model	Selection method	Selection criterion	lambda	No. of selected variables
y	linear	cv	CV min.	.2475897	2

lassocof command

- Provides three different sets of coefficient estimates
- **1.** Standardized coefficients (default) are those directly from lasso.
- **2.** Penalized coefficients are the preceding ones rescaled so that the standardization of variables is removed
 - ▶ i.e. can interpret in terms of the original data.
- **3.** Post-selection coefficients are obtained by OLS of y on the selected regressors (here x_1 and x_2).
 - ▶ often called post-lasso estimates.

Standardized coefficients for standardized regressors

- We have

```
. * Lasso coefficients for the standardized regressors  
. lassocoeff, display(coef, standardized)
```

	active
x1	1.206056
x2	.2715635
_cons	0

Legend:

- b - base level
- e - empty cell
- o - omitted

Unstandardized coefficients for original regressors

- Recall d.g.p. $y = 2 + 1 \times x_1 + 0 \times x_2 + 0 \times x_3 + u$.

```
. * Lasso coefficients for the unstandardized regressors  
. lassocoeff, display(coef, penalized) nolegend
```

	active
x1	1.35914
x2	.2918877
_cons	2.617622

Post-selection (post-LASSO)

- OLS on the selected and unstandardized regressors
 - ▶ same as `regress y x1 x2`

```
. * Post-selection estimated coefficients for the unstandardized regressors  
. lassocoef, display(coef, postselection) nolegend
```

	active
x1	1.544198
x2	.4683922
_cons	2.533663

lassogof command

- Goodness-of-fit

- ▶ as expected post-lasso OLS fits better than lasso

```
. * Goodness-of-fit with penalized coefficients and postselection coefficients
. lassogof, penalized
```

Penalized coefficients

MSE	R-squared	Obs
8.679274	0.2300	40

```
. lassogof, postselection
```

Postselection coefficients

MSE	R-squared	Obs
8.597958	0.2372	40

Adaptive Lasso

- Method that usually leads to fewer variables than basic lasso.
 - First lasso as usual with $\kappa_j = 1$
 - Second exclude x_j with $\hat{\beta}_j = 0$ and for remainder set $\kappa_j = 1/|\hat{\beta}_j|^\delta$ with default $\delta = 1$.
- Here only x_1 is selected.

```
. * Lasso linear using 5-fold adaptive cross validation
. qui lasso linear y x1 x2 x3, selection(adaptive) folds(5) rseed(10101)

. lassoknots
```

ID	lambda	No. of nonzero coef.	CV mean pred. error	Variables (A)dded, (R)emoved, or left (U)nchanged
26	3.945214	1	11.60145	A x1
* 52	.3512089	1	9.160539	U
57	.2205694	2	9.210699	A x2
95	.0064297	2	9.172378	U

* lambda selected by cross-validation in final adaptive step.

4.5 Elasticnet and Ridge Regression

- Elastic net combines ridge regression and LASSO with objective function

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

- ▶ ridge penalty λ averages correlated variables
 - ▶ LASSO penalty α leads to sparsity.
- For elastic net
 - ▶ Ridge is special case $\alpha = 0$
 - ▶ LASSO is special case $\alpha = 1$.

Ridge Regression

- Standardized ridge (OLS estimates were 1.555, 0.471 and -0.026.)

```
. * Ridge estimation using the elasticnet command and selected results
. qui elasticnet linear y x1 x2 x3, alpha(0) rseed(10101) folds(5)

. lassoknots
```

alpha	ID	lambda	No. of nonzero coef.	CV mean pred. error	Variables (A)dded, (R)emoved, or left (U)nchanged
0.000	1	1591.525	3	11.9595	A x1 x2 x3
	* 93	.3052401	3	9.54017	U
	100	.1591525	3	9.566065	U

* alpha and lambda selected by cross-validation.

```
. lassocoeff, display(coef, penalized) nolegend
```

	active
x1	1.139476
x2	.4865453
x3	.0958546
_cons	2.659647

Elastic net

- Default is λ 100 point logarithmic grid and $\alpha = 0.5, 0.7, 1.0$
 - ▶ here $\alpha = 1.0$ (lasso) so narrow grid to 0.90, 0.95, 1.0
 - ▶ optimal $\alpha = 0.95$, $\lambda = 0.2717$, and x1 and x2 selected.

```
. * Elastic net estimation and selected results
. qui elasticnet linear y x1 x2 x3, alpha(0.9(0.05)1) rseed(10101) folds(5)

. lassoknots
```

alpha	ID	lambda	No. of nonzero coef.	CV mean pred. error	Variables (A)dded, (R)emoved, or left (U)nchanged
1.000	4	1.450138	1	11.60145	A x1
	13	.6277301	2	9.739804	A x2
	26	.1872925	2	9.434326	U
0.950	29	1.591525	1	11.73019	A x1
	38	.688933	2	9.81611	A x2
	* 48	.2717294	2	9.3884	U
	51	.2055533	2	9.425887	U
0.900	53	1.675289	1	11.74015	A x1
	62	.7561031	2	9.900317	A x2
	76	.2055533	2	9.431641	U

* alpha and lambda selected by cross-validation.

4.6 Comparison of Shrinkage Estimators

- Compare OLS, Lasso, ridge, elastic net.

```
. * Estimate various models and store results
. qui regress y x1 x2 x3

. estimates store OLS

. qui lasso linear y x1 x2 x3, selection(cv) folds(5) rseed(10101)

. estimates store LASCV

. qui lasso linear y x1 x2 x3, selection(adaptive) folds(5) rseed(10101)

. estimates store LASADAPT

. qui lasso linear y x1 x2 x3, selection(plugin) folds(5)

. estimates store LASPLUG

. qui elasticnet linear y x1 x2 x3, alpha(0) selection(cv) folds(5) rseed(10101)

. estimates store RIDGECV

. qui elasticnet linear y x1 x2 x3, alpha(0.9(0.05)1) rseed(10101) folds(5)

. estimates store ELASTIC
```

Comparison of Shrinkage Estimators

- Most select x_1 and x_2 , adaptive lasso only x_1 , ridge all three.

```
. * Compare in-sample fit and selected coefficients of various models
. lassogof OLS LASCV LASADAPT LASPLUG RIDGECV ELASTIC
```

Penalized coefficients

Name	MSE	R-squared	Obs
OLS	8.597403	0.2373	40
LASCV	8.679274	0.2300	40
LASADAPT	8.755573	0.2232	40
LASPLUG	10.27195	0.0887	40
RIDGECV	8.70562	0.2277	40
ELASTIC	8.693386	0.2288	40

```
. lassocoeff OLS LASCV LASADAPT LASPLUG RIDGECV ELASTIC, display(coef) nolegend
```

	OLS	LASCV	LASADAPT	LASPLUG	RIDGECV	ELASTIC
x1	1.555582	1.206056	1.462431	.3533654	1.011134	1.179972
x2	.4707111	.2715635			.452667	.2705777
x3	-.0256025				.0979251	
_cons	2.531396	0	0	0	0	0

4.7 Shrinkage for logit, probit and poisson

- More generally can apply shrinkage to other objective functions.
- For logit, probit and poisson replace the squared residual by the squared deviance residual
 - ▶ deviance residual is used for generalized linear models
- Consider lasso $\hat{\beta}_\lambda$ of β minimizes

$$Q_\lambda(\beta) = \sum_{i=1}^n q(y_i, \mathbf{x}_i, \beta) + \lambda \sum_{j=1}^p |\beta_j|$$

- Logit: $q(y_i, \mathbf{x}_i, \beta) = \{2[y_i \ln \Lambda(\mathbf{x}'_i \beta) + (1 - y_i) \ln(1 - \Lambda(\mathbf{x}'_i \beta))]\}^2$
- Probit: $q(y_i, \mathbf{x}_i, \beta) = \{2[y_i \ln \Phi(\mathbf{x}'_i \beta) + (1 - y_i) \ln(1 - \Phi(\mathbf{x}'_i \beta))]\}^2$
- Poisson: $q(y_i, \mathbf{x}_i, \beta) = \{2[y_i \mathbf{x}'_i \beta - \exp(\mathbf{x}'_i \beta) - v_i]\}^2$
 - ▶ $v_i = 0$ if $y_i = 0$ and $v_i = y_i \ln y_i$ otherwise.

Lasso shrinkage for logit example

- Create $dy = y > 3$

- ▶ only x1 selected

```
. * Lasso for logit example
. qui generate dy = y > 3

. qui lasso logit dy x1 x2 x3, rseed(10101) folds(5)

. lassoknots
```

ID	lambda	No. of nonzero coef.	CV mean deviance	Variables (A)dded, (R)emoved, or left (U)nchanged
2	.2065674	1	1.407613	A x1
* 24	.0266792	1	1.192646	U
26	.0221495	2	1.192865	A x2
30	.0152668	3	1.194545	A x3
31	.0139106	3	1.195055	U

* lambda selected by cross-validation.

Other user-written Stata commands for LASSO

- User-written command `lassoshooting` (Christian Hansen)
 - ▶ uses the coordinate descent (called lasso shooting) algorithm of Fu (1998)
 - ▶ with theoretical or user-choice of λ (no cross validation)
 - ▶ superseded by the `lassopack` package.
- `Lassopack` package of Ahrens, Hansen and Schaffer (2020)
 - ▶ `cvlasso` for λ chosen by K-fold cross-validation and h-step ahead rolling cross-validation for cross-section, panel and time-series data
 - ▶ `rlasso` for theory-driven ('rigorous') penalization for the lasso and square-root lasso for cross-section and panel data
 - ▶ `lasso2` for information criteria choice of λ
 - ▶ **now supplanted by Stata's commands.**
- User-written command `lars` (Mander)
 - ▶ `lars ydemeaned zx1 zx2 zx3 zx4, a(lasso)`
 - ▶ at each step minimizes Mallows C_p

5. Prediction for Economics

- Microeconometrics focuses on estimation of β or of partial effects (later).
- 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, JPE)
 - ▶ consider prediction of housing prices
 - ▶ detail how to do this using machine learning methods
 - ▶ and then summarize many recent economics ML applications.

5.1 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 (and not post-LASSO OLS)
 - ▶ 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
 - ▶ training sample is $n = 10,000$ and holdout sample is $n = 41,808$.

Table 1

Performance of Different Algorithms in Predicting House Values

Method	Prediction performance (R^2)		Relative improvement over ordinary least squares by quintile of house value				
	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%

Details

- Downloadable appendix to the paper gives more details and R code.
- 1. Divide into training ($n = 10,000$) and hold-out sample ($n = 41,808$).
- 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 $\hat{f}(x)$ is estimated using the entire training sample ($n = 10,000$) with optimal λ .
- 4. Now apply this $\hat{f}(x)$ to the hold-out sample compute $MSE = \frac{1}{41808} \sum_i (y_i - \hat{f}(x_i))^2$
 hence compute $R^2 = 1 - \frac{\sum_i (y_i - \hat{f}(x_i))^2}{\sum_i (y_i - \bar{y})^2} = 1 - \frac{MSE}{\frac{1}{41808} (y_i - \bar{y})^2}$.
- 5. A 95% CI for R^2 is obtained by bootstrapping the hold-out sample.

Ensemble Weights

- Ensemble weights are similar to portfolio diversification.

- Example: $X_1 \sim (\mu, \sigma^2)$ independent of $X_2 \sim (\mu, \sigma^2)$

then

$$\text{Var}[(X_1 + X_2)/2] = \frac{1}{4}\{\text{Var}[X_1] + \text{Var}[X_2]\} = \frac{\sigma^2}{2} < \text{Var}[X_1] = \sigma^2.$$

- ▶ benefit is less the more correlated are X_1 and X_2 .

- So consider a linear combination of predictions.

- For each ML method create 10,000 predictions in the training sample as follows

- ▶ for each of the eight folds estimate (using the optimal tuning parameter(s)) using seven folds and predict on the remaining fold
- ▶ this gives $(10,000 \times 1)$ vectors $\hat{\mathbf{y}}_{OLS}$, $\hat{\mathbf{y}}_{REGTREE}$, $\hat{\mathbf{y}}_{LASSO}$, $\hat{\mathbf{y}}_{RF}$.

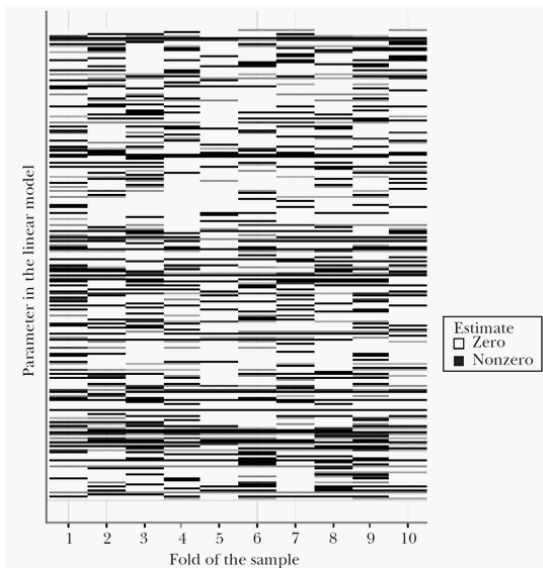
- The ensemble weights are the $\hat{\alpha}'$ s from the OLS regression in the training sample

$$y_i = \alpha_0 + \alpha_1 \hat{y}_{OLS,i} + \alpha_2 \hat{y}_{REGTREE,i} + \alpha_3 \hat{y}_{LASSO,i} + \alpha_4 \hat{y}_{RF,i} + u_i.$$

- These ensemble weights are also used in the holdout sample exercise.

Further Details

- LASSO does not pick the “correct” regressors
 - ▶ it just gets the correct $\hat{f}(x)$ 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.



6. Some R Commands

- These are from *An Introduction to Statistical Learning: with Applications in R*. **There may be better commands.**
- Basic regression
 - ▶ OLS is `lm.fit`
 - ▶ cross-validation for OLS uses `cv.glm()`
 - ▶ bootstrap uses `boot()` function in `boot` library
- Variable selection
 - ▶ best subset, forward stepwise and backward stepwise: `regsubsets()` in `leaps` library
- Penalized regression
 - ▶ ridge regression: `glmnet(,alpha=0)` function in `glmnet` library
 - ▶ lasso: `glmnet(,alpha=1)` function in `glmnet` library
 - ▶ CV to get lambda for ridge/lasso: `cv.glmnet()` in `glmnet` library

7. References

- ISL2: Gareth James, Daniela Witten, Trevor Hastie and Robert Tibsharani (2013), An Introduction to Statistical Learning: with Applications in R, Second edition, Springer.
 - ▶ free legal pdf at <https://www.statlearning.com/>
- 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 copy <http://www.springer.com/gp/products/books/mycopy>
- EH: Bradley Efron and Trevor Hastie (2016), Computer Age Statistical Inference: Algorithms, Evidence and Data Science, Cambridge University Press.
- Chapter 28.3 and 28.4 of “Machine Learning for prediction and inference” in A. Colin Cameron and Pravin K. Trivedi (2022), Microeconometrics using Stata, Second edition, forthcoming.

References (continued)

- Sendhil Mullainathan and J. Spiess (2017): “Machine Learning: An Applied Econometric Approach”, *Journal of Economic Perspectives*, Spring, 87-106.
- Hannes Leeb and Benedikt M. Pötscher (2005), “Model selection and Inference”, *Econometric Theory*, 21-59.
- Peter Bickel, Ya’acov Ritov and Alexandre Tsybakov (2009), “Simultaneous Analysis of Lasso and Dantzig Selector”, *The Annals of Statistics*, 1705-1732.
- Alex Belloni and Victor Chernozhukov (2013), “Least Squares after Model Selection in High-Dimensional Sparse Models”, *Bernoulli*, 521-547.
- Achim Ahrens, Christian B. Hansen, Mark E. Schaffer (2020), “lassopack: Model selection and prediction with regularized regression in Stata”, *The Stata Journal*, 20, 176-235 (also ArXiv:1901.05397).
- Domenico Giannone, Michele Lenza, Giorgio E. Primiceri (2021), “Economic Predictions with Big Data: The Illusion of Sparsity,”
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3835164.