

Lecture 17:  
Regularization/Shrinkage Methods  
Big Data and Machine Learning for Applied Economics  
Econ 4676

Ignacio Sarmiento-Barbieri

Universidad de los Andes

October 12, 2021

# Agenda

- 1 Recap: Model Selection
- 2 Regularization
  - Lasso
  - Ridge
- 3 Family of penalized regressions
- 4 More predictors than observations ( $k > n$ )
- 5 Elastic Net
- 6 Review & Next Steps
- 7 Further Readings
- 8 Demo Regularization

# Model Selection

- ▶ ML we care about prediction out of sample

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{k-1} X_{k-1} + \beta_k X_k + u \quad (1)$$

- ▶ Trade off Bias/Variance: more "complex" models have less bias but more variance
- ▶ Overfit: complex models predict very well inside a sample but "bad" outside
- ▶ Choose the right complexity level
- ▶ Big innovation accept some bias to decrease variance → improves out of sample prediction

# Model Selection

- ▶ Given the models, we choose the one that “best” predicts out of sample

$$M_0 \rightarrow y = \beta_0 + u \quad (2)$$

$$M_1 \rightarrow y = \beta_0 + \beta_j X_j + u \quad (3)$$

$$\vdots \quad (4)$$

$$M_k \rightarrow y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + u \quad (5)$$

- ▶ How we bet the “best”:

- 1 Choose based on some measure of sample fit, but penalized for complexity, e.g.

$$BIC = SSR - \frac{1}{2}p \log(n)$$

- 2 Use resampling techniques (create “a fake” out of sample) use prediction error

# Model Selection

- Where do these  $k$  models come from? If I have  $k$  variables how do I search among the possible models

$$M_0 \rightarrow y = \beta_0 + u \quad (6)$$

$$M_1 \rightarrow y = \beta_0 + \beta_j X_j + u \quad (7)$$

$$\vdots \quad (8)$$

$$M_k \rightarrow y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + u \quad (9)$$

- Best subset selection
- Forward Stepwise Selection
- Backward Stepwise Selection

# Regularization

- ▶ What about if I do the following?
- ▶ Get the most general model I can

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{k-1} X_{k-1} + \beta_k X_k + u \quad (10)$$

# Regularization

- ▶ What about if I do the following?
- ▶ Get the most general model I can

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{k-1} X_{k-1} + \beta_k X_k + u \quad (10)$$

- ▶ Run model, get coefficients and p-values

# Regularization

- ▶ What about if I do the following?
- ▶ Get the most general model I can

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{k-1} X_{k-1} + \beta_k X_k + u \quad (10)$$

- ▶ Run model, get coefficients and p-values
- ▶ Take out those with p-values below a certain  $\alpha$



# Regularization

- ▶ What about if I do the following?
- ▶ Get the most general model I can

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{k-1} X_{k-1} + \beta_k X_k + u \quad (10)$$

- ▶ Run model, get coefficients and p-values
- ▶ Take out those with p-values below a certain  $\alpha$
- ▶ Why is this a bad idea?

# Regularization

- ▶ What about if I do the following?
- ▶ Get the most general model I can

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{k-1} X_{k-1} + \beta_k X_k + u \quad (10)$$

- ▶ Run model, get coefficients and p-values
  - ▶ Take out those with p-values below a certain  $\alpha$
  - ▶ Why is this a bad idea?
- 
- ▶ Backward selection approximates that idea (not exactly) and does a better job than the previous point.

# Regularization

- ▶ 'Crossing out' variables / coefficients is an extreme way to 'shrink' them.
- ▶ Lasso: a formal and algorithmic way of accomplishing this task.
- ▶ The strategy involves penalizing complexity so as to depart from optimality and stabilize the system
- ▶ The key of modern statistics is regularization

# Lasso

- ▶ For  $\lambda \geq 0$  given, consider minimizing the following objective function

$$\min_{\beta} L(\beta) = \sum_{i=1}^n (y_i - x_i' \beta)^2 + \lambda \sum_{s=2}^p |\beta_s| \quad (11)$$

- ▶ Note:
  - ▶ First coef. constant

# Lasso

- ▶ For  $\lambda \geq 0$  given, consider minimizing the following objective function

$$\min_{\beta} L(\beta) = \sum_{i=1}^n (y_i - x_i' \beta)^2 + \lambda \sum_{s=2}^p |\beta_s| \quad (11)$$

- ▶ Note:
  - ▶ First coef. constant
  - ▶  $\lambda = 0$ ?

# Lasso

- ▶ For  $\lambda \geq 0$  given, consider minimizing the following objective function

$$\min_{\beta} L(\beta) = \sum_{i=1}^n (y_i - x_i' \beta)^2 + \lambda \sum_{s=2}^p |\beta_s| \quad (11)$$

- ▶ Note:
  - ▶ First coef. constant
  - ▶  $\lambda = 0$ ?
  - ▶  $\lambda = \infty$ ?

$$\min_{\beta} L(\beta) = \sum_{i=1}^n (y_i - x_i' \beta)^2 + \lambda \sum_{s=2}^p |\beta_s| \quad (12)$$

- ▶ LASSO's free lunch: automatically chooses which variables go in ( $\beta_s \neq 0$ ) and which are out ( $\beta_s = 0$ )
- ▶ Why? Coefficients that don't go in are corner solutions
- ▶  $L(\beta)$  is non differentiable

# Corner Solutions

## ► Lasso Intuition

$$\min_{\beta} L(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| \quad (13)$$

- Only one predictor, i.e., one coefficient.
- For  $\lambda = 0$

$$\hat{\beta}_{OLS} = \frac{\sum_i x_i y_i}{\sum_i x_i^2} \quad (14)$$

- If we standardize predictor  $\sum x_i^2 = 1$ ?



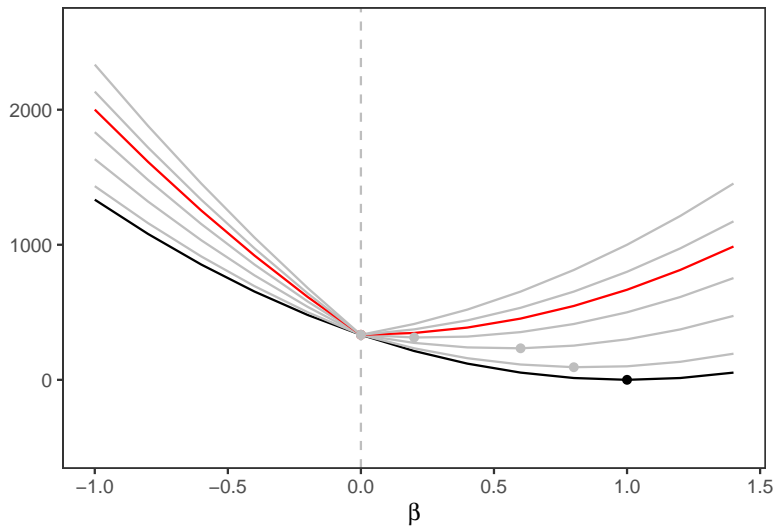
# Corner Solutions

$$\min_{\beta} L(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| \quad (15)$$

- ▶ Non differentiable at  $\beta = 0$
- ▶ Differentiable otherwise  $\beta \neq 0$

$$L(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| = \begin{cases} \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda \beta & \text{if } \beta > 0 \\ \sum_{i=1}^n (y_i - x_i \beta)^2 - \lambda \beta & \text{if } \beta < 0 \end{cases} \quad (16)$$

# Corner Solutions



## Corner Solutions

$$\begin{aligned}\frac{dL(\beta)^+}{d\beta} &= -2 \sum y_i x_i + 2\beta \sum x_i^2 + \lambda \\ &= -2 \sum y_i x_i + \beta + \lambda\end{aligned}\tag{17}$$

$$\frac{dL(0)^+}{d\beta} = -2 \sum y_i x_i + \lambda$$

then, if

$$\lambda \geq 2 \sum y_i x_i\tag{18}$$

we have  $\hat{\beta}_{lasso} = 0$

# Corner Solutions

- If  $\lambda < 2 \sum y_i x_i$  we have an interior solution

$$-2 \sum y_i x_i + \hat{\beta}_{lasso} + \lambda = 0 \quad (19)$$

$$\hat{\beta}_{lasso} = \sum y_i x_i - \frac{\lambda}{2} \quad (20)$$

$$\hat{\beta}_{lasso} = \hat{\beta}_{OLS} - \frac{\lambda}{2} \quad (21)$$

- We get shrinkage towards 0

# Corner Solutions

$$\min_{\beta} L(\beta) = \sum_{i=1}^n (y_i - x_i \beta)^2 + \lambda |\beta| \quad (22)$$

$$\hat{\beta}_{lasso} = \begin{cases} 0 & \text{si } \lambda \geq 2 \sum y_i x_i \\ \hat{\beta}_{OLS} - \frac{\lambda}{2} & \text{si } \lambda < 2 \sum y_i x_i \end{cases} \quad (23)$$

# Ridge

- ▶ For  $\lambda \geq 0$  given, consider minimizing the following objective function

$$\min_{\beta} R(\beta) = \sum_{i=1}^n (y_i - x_i' \beta)^2 + \lambda \sum_{s=2}^p (\beta_s)^2 \quad (24)$$

- ▶ Note:
  - ▶ Intuition is similar to Lasso, however the problem is completely different

# Ridge

Simple case 1 predictor

$$\min_{\beta} R(\beta) = \sum_{i=1}^n (y_i - x_i' \beta)^2 + \lambda (\beta_s)^2 \quad (25)$$

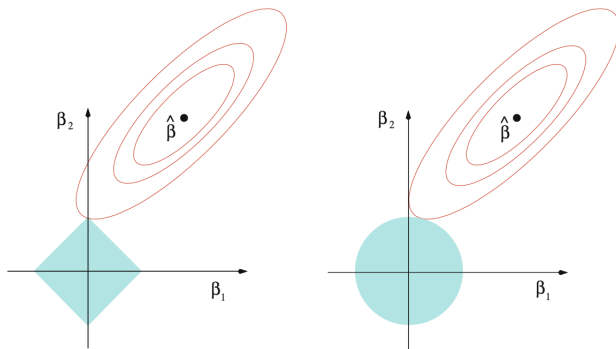
FOC

$$-2 \sum_{i=1}^n y_i x_i + 2\beta + 2\lambda \beta_s = 0 \quad (26)$$

$$\begin{aligned} \hat{\beta}_{ridge} &= \frac{\sum_{i=1}^n y_i x_i}{(1 + \lambda)} \\ &= \frac{\beta_{OLS}}{(1 + \lambda)} \end{aligned} \quad (27)$$

- ▶ Solution is *always* interior (unlike Lasso)
- ▶ Solutions is "shrunk"

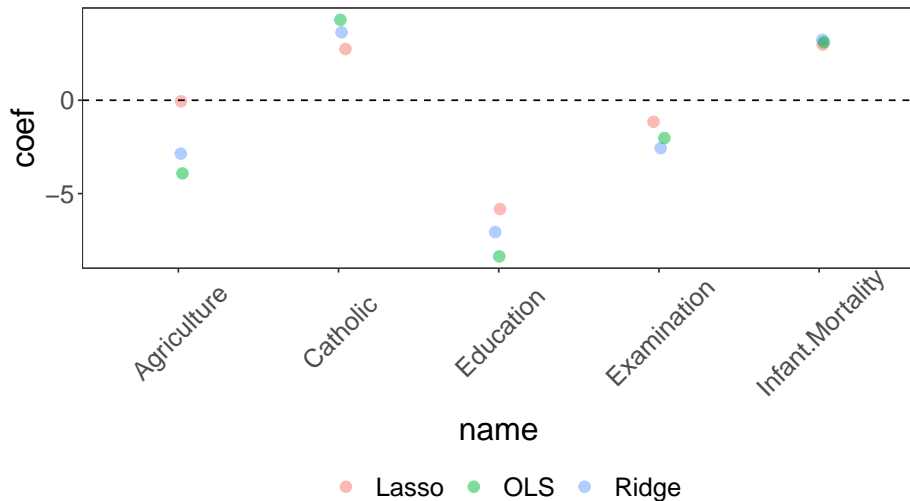
# Lasso and Ridge Intuition



**FIGURE 3.11.** Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions  $|\beta_1| + |\beta_2| \leq t$  and  $\beta_1^2 + \beta_2^2 \leq t^2$ , respectively, while the red ellipses are the contours of the least squares error function.



# Lasso and Ridge Example



# Technical comments

- ▶ We showed that Ridge is biased, but you have to show in the PS that  $\lambda$   $MSE(\beta_{ridge}) < MSE(\beta_{OLS})$
- ▶ Not possible to derive an exact result for Lasso, but Ridge works similarly
- ▶ Lasso shrinks everything towards zero, Ridge not quite so
- ▶ Application wise:
  - ▶ Standardize the data
  - ▶ Selection of  $\lambda$ ?

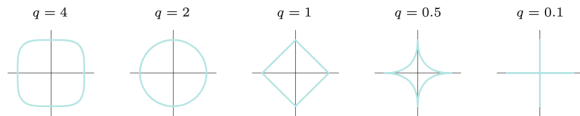
# Technical comments: $\lambda$ Selection

- ▶ Selection of  $\lambda$ ?
- ▶ Use CV
  - ▶ Choose a grid of  $\lambda$  values, and compute the CV error for each value
  - ▶ Select the  $\lambda^*$  that minimizes the prediction error
  - ▶ Estimate using all the observations and the selected  $\lambda^*$

# Family of penalized regressions

## ► Family of penalized regressions

$$\min_{\beta} R(\beta) = \sum_{i=1}^n (y_i - x_i' \beta)^2 + \lambda \sum_{s=2}^p |\beta_s|^q \quad (28)$$



**FIGURE 3.12.** Contours of constant value of  $\sum_j |\beta_j|^q$  for given values of  $q$ .

# More predictors than observations ( $k > n$ )

- ▶ Objective 1: Accuracy
  - ▶ Minimize prediction error (in one step)  $\rightarrow$  Ridge, Lasso
- ▶ Objective 2: Dimensionality
  - ▶ Reduce the predictor space  $\rightarrow$  Lasso's free lunch

# More predictors than observations ( $k > n$ )

- ▶ Objective 1: Accuracy
  - ▶ Minimize prediction error (in one step)  $\rightarrow$  Ridge, Lasso
- ▶ Objective 2: Dimensionality
  - ▶ Reduce the predictor space  $\rightarrow$  Lasso's free lunch
- ▶ What happens when we have more predictors than observations ( $k > n$ )?
  - ▶ OLS fails
  - ▶ Ridge and Lasso to the rescue?

## OLS when $k > n$

- ▶ Rank? Max number of rows or columns that are linearly independent
  - ▶ Implies  $\text{rank}(X_{k \times n}) \leq \min(k, n)$
- ▶ MCO we need  $\text{rank}(X_{k \times n}) = k \implies k \leq n$
- ▶ If  $\text{rank}(X_{k \times n}) = k$  then  $\text{rank}(X'X) = k$
- ▶ If  $k > n$ , then  $\text{rank}(X'X) \leq n < k$  then  $(X'X)$  cannot be inverted
- ▶ Ridge and Lasso work when  $k \geq n$

## Ridge when $k > n$

$$\min_{\beta} R(\beta) = \sum_{i=1}^n (y_i - x_i' \beta)^2 + \lambda \sum_{s=1}^p (\beta_s)^2 \quad (29)$$

- ▶ Solution  $\rightarrow$  data augmentation
- ▶ Intuition: Ridge “adds”  $k$  additional points.
- ▶ Allows us to “deal” with  $k \geq n$



## Lasso when $k > n$

- ▶ Lasso works fine in this case
- ▶ However, there are some issues to keep in mind
  - ▶ When  $k > n$  chooses at most  $n$  variables
  - ▶ When we have a group of highly correlated variables,
    - ▶ Lasso chooses only one. Makes it unstable for prediction. (Doesn't happen to Ridge)
    - ▶ Ridge shrinks the coefficients of correlated variables toward each other. This makes Ridge “work” better than Lasso. “Work” in terms of prediction error

# Naive Elastic Net

► Combination? Elastic Net

$$\begin{aligned} \min_{\beta} EL(\beta) &= \sum_{i=1}^n (y_i - x_i' \beta)^2 + \lambda_1 \sum_{s=2}^p |\beta_s| + \lambda_2 \sum_{s=2}^p \beta_s^2 \\ &= \sum_{i=1}^n (y_i - x_i' \beta)^2 + \lambda \sum_{s=2}^p ((1 - \alpha) \beta_s^2 + \alpha |\beta_s|) \end{aligned} \quad (30)$$

# Naive Elastic Net

- ▶ Elastic net: happy medium.
  - ▶ Good job at prediction and selecting variables

$$\min_{\beta} NEL(\beta) = \sum_{i=1}^n (y_i - x_i' \beta)^2 + \lambda_1 \sum_{s=2}^p |\beta_s| + \lambda_2 \sum_{s=2}^p \beta_s^2 \quad (31)$$

- ▶ Mixes Ridge and Lasso
- ▶ Lasso selects predictors
- ▶ Strict convexity part of the penalty (ridge) solves the grouping instability problem
- ▶ H.W.:  $\beta_{OLS} > 0$  one predictor standardized

$$\hat{\beta}_{naive EN} = \frac{\left(\hat{\beta}_{OLS} - \frac{\lambda_1}{2}\right)_+}{1 + \lambda_2} \quad (32)$$

# Elastic Net

- ▶ Elastic Net: rescaled version
- ▶ Double Shrinkage introduces “too” much bias, *final* version “corrects” for this

$$\hat{\beta}_{EN} = \frac{1}{\sqrt{1 + \lambda_2}} \hat{\beta}_{naive EN} \quad (33)$$

- ▶ Careful sometimes software asks.
- ▶ How to choose  $(\lambda_1, \lambda_2)$ ?  $\rightarrow$  Bidimensional Crossvalidation
- ▶ Zou, H. & Hastie, T. (2005)

# Review & Next Steps

- ▶ Today:
  - ▶ Regularization
    - ▶ Lasso
    - ▶ Ridge
    - ▶ Elastic Net
- ▶ Next class: Lasso for Causal Inference.
- ▶ PS3 Friday and remember to work on your proposals

## Further Readings

- ▶ Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.
- ▶ James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
- ▶ Kuhn, M. (2012). The caret package. R Foundation for Statistical Computing, Vienna, Austria. <https://topepo.github.io/caret/index.html>
- ▶ Taddy, M. (2019). Business data science: Combining machine learning and economics to optimize, automate, and accelerate business decisions. McGraw Hill Professional.

# Regularization Demo

```
#Load the required packages  
library("dplyr") #for data wrangling  
library("caret") #ML  
  
data(swiss) #loads the data set  
  
set.seed(123) #set the seed for replication purposes  
str(swiss) #compact display
```

```
## 'data.frame':    47 obs. of  6 variables:  
## $ Fertility      : num  80.2 83.1 92.5 85.8 76.9 76.1 83.8 92.4 82.4 82.9 ...  
## $ Agriculture    : num  17 45.1 39.7 36.5 43.5 35.3 70.2 67.8 53.3 45.2 ...  
## $ Examination    : int   15 6 5 12 17 9 16 14 12 16 ...  
## $ Education       : int   12 9 5 7 15 7 7 8 7 13 ...  
## $ Catholic        : num   9.96 84.84 93.4 33.77 5.16 ...  
## $ Infant.Mortality: num   22.2 22.2 20.2 20.3 20.6 26.6 23.6 24.9 21 24.4 ...
```

# Regularization Demo

```
ols <- train(Fertility ~ .,    # model to fit
             data = swiss,
             trControl = trainControl(method = "cv", number = 10),    # Method: crossvalidation,
             method = "lm")      # specifying regression model

ols
```

```
## Linear Regression
##
## 47 samples
## 5 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 42, 42, 42, 42, 42, 44, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##  7.424916  0.6922072  6.31218
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```



# Regularization Demo

```
lambda <- 10^seq(-2, 3, length = 100)
lasso <- train(
  Fertility ~., data = swiss, method = "glmnet",
  trControl = trainControl("cv", number = 10),
  tuneGrid = expand.grid(alpha = 1, lambda=lambda), preProcess = c("center", "scale")
)

lasso
```

```
## glmnet
##
## 47 samples
## 5 predictor
##
## Pre-processing: centered (5), scaled (5)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 43, 43, 43, 42, 42, 41, ...
## Resampling results across tuning parameters:
##
## ...
##
## Tuning parameter 'alpha' was held constant at a value of 1
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 1 and lambda = 0.02009233.
```

# Regularization Demo

```
ridge <- train(
  Fertility ~., data = swiss, method = "glmnet",
  trControl = trainControl("cv", number = 10),
  tuneGrid = expand.grid(alpha = 0, lambda = lambda), preProcess = c("center", "scale")
)
ridge
```

```
## glmnet
##
## 47 samples
## 5 predictor
##
## Pre-processing: centered (5), scaled (5)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 42, 42, 43, 44, 42, 42, ...
## Resampling results across tuning parameters:
##
## ...
##
## Tuning parameter 'alpha' was held constant at a value of 0
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0 and lambda = 0.7390722.
```

# Regularization Demo

```
##  
## Call:  
## summary.resamples(object = ., metric = "RMSE")  
##  
## Models: ridge, lasso  
## Number of resamples: 10  
##  
## RMSE
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
## ridge	2.615430	4.674108	7.627190	6.923531	8.939798	10.55026	0
## lasso	3.205868	5.553161	5.961622	7.324069	8.587818	13.46074	0