A Practical Introduction to Variable Selection using R

Shuang Yin, Xiaomeng Li, Jinjian Mu
Statistical Consulting Service
University of Connecticut

11/13/2019

Outline

- Introduction
- Criteria for Model Selection and Stepwise Regression
- Lasso Regression
- Elastic Net

Introduction

Why is variable selection necessary?

- Most models don't deal well with a large number of irrelevant variables
- Collinearity
- Estimates of model fit might be overly optimistic

Models and Basic Setup

- Full Model:

$$\text{target variable} = \text{intercept} + \text{coefficent} \times \text{predictor} + \text{error}$$

- Goal: Select the best subset of predictors
- Total number of all possible subsets: 2^K, if there are K predictors.

Methods

- Subset selection using criteria: AIC, BIC, R^2
- Forward selection
- Backward elimination
- Stepwise regression

Automatic Search Procedures

Principle of Parsimony (Simplicity of description)

if two models fit the data equally well, the simplest (the smaller model) is the best (better) model.

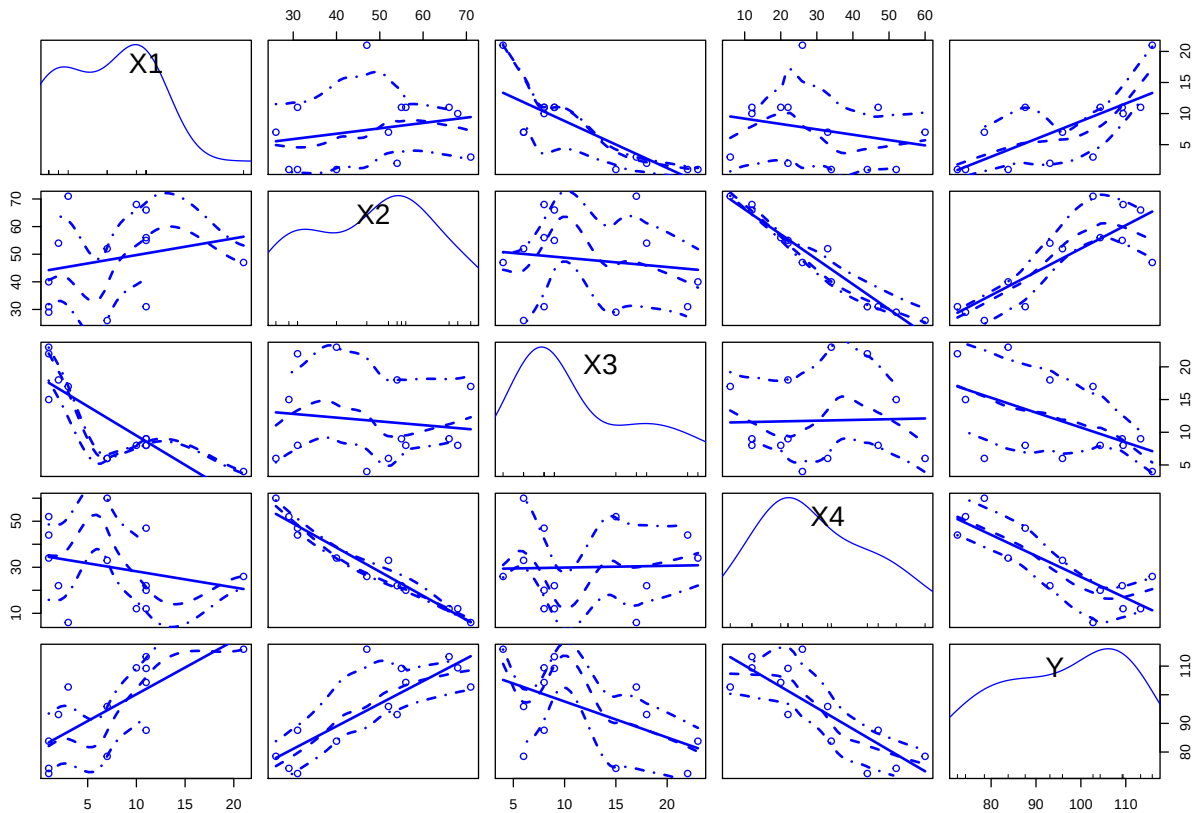
Criteria

- A model with a smaller AIC, BIC
- A model with a larger R^2

Example

There are four predictors, the total number of possible subset models $= 2^4 - 1 = 15$. We use AIC, BIC, R^2, criteria to select the best subset regression model.

```
## Loading required package: carData
```



Output of the example

R-square	AIC	BIC	Variables in Model
0.67	58.85	59.98	X4
0.67	59.18	60.31	X2
0.53	63.52	64.65	X1
0.29	69.07	70.20	X3
0.98	25.42	27.11	X1 X2
0.97	28.74	30.44	X1 X4
0.94	39.85	41.55	X3 X4
0.85	51.04	52.73	X2 X3
0.98	24.97	27.23	X1 X2 X4
0.98	25.01	27.27	X1 X2 X3
0.98	25.73	27.99	X1 X3 X4
0.97	30.58	32.84	X2 X3 X4
0.98	26.94	29.77	X1 X2 X3 X4

Analysis

- The correlation between X_1 and X_3 is -0.824 and the correlation between X_2 and X_4 is -0.973.
- Based on the Principle of Parsimony, we would choose $\{X_1, X_2\}$ to give the best model.

Forward Selection Procedure

step 0	Specify an entry level of significance α, start with no predictor variable in the model (only the intercept)
step 1	Fit the model with the predictor one by one and keep the one with the smallest p-value
step 2	Given the predictor from step 1 in the model, fit with the rest predictors one by one and keep the one with the smallest p-value
step 3	Repeat this procedure until we go through all the predictors

Disadvantage: Once a variable enters the model, it will never be able to get out (even if its presence in the model becomes unnecessary).

Backward Elimination Procedure

step 0	Specify an elimination level of significance, start with all predictors in the model
step 1	Check the p-values of the model with deleting one variable out one by one, and eliminate the one with the largest p-value
step 2	Given the rest of variables in the model, check the p-value of the model with one variable deleted one by one, also eliminate the one with the largest p-value
step 3	Repeat this procedure until we go through all the predictors

Disadvantage: Once a variable is removed from the model, it can never get back into the model again.

Stepwise Regression

- Stepwise regression is a combination of Forward Selection and Backward Elimination
- Specify a significance level for entry and also a significance level for elimination
- Specify starting from Forward or Backward regression.

Implementation in R

```
model <- lm(Y~X1 + X2 + X3 + X4, data= hald)
step(model, direction = c("backward"), k=1)
```

Output

```
## Start:  AIC=21.94
## Y ~ X1 + X2 + X3 + X4
##
##        Df Sum of Sq    RSS    AIC
## - X3    1    0.1091 47.973 20.974
## - X4    1    0.2470 48.111 21.011
## - X2    1    2.9725 50.836 21.728
## <none>               47.864 21.944
## - X1    1   25.9509 73.815 26.576
##
## Step:  AIC=20.97
## Y ~ X1 + X2 + X4
##
##        Df Sum of Sq    RSS    AIC
## <none>               47.97 20.974
## - X4    1      9.93  57.90 22.420
## - X2    1     26.79  74.76 25.742
## - X1    1    820.91 868.88 57.629



##
## Call:
## lm(formula = Y ~ X1 + X2 + X4, data = hald)
##
## Coefficients:
## (Intercept)           X1           X2           X4
##     71.6483       1.4519       0.4161      -0.2365
```

Some Guidlines

Advantages	The ability to manage large amounts of potential predictor variables, and to choose the best subset of variables.
	The order in which variables are removed or added can provide valuable information about the quality of the predictor
Disadvantages	Stepwise regression is unreliable, if another sample is taken from the same population, the variables selected may be very different.
	Some other issues: biased statistis, estimates and in the presence of collinearity.

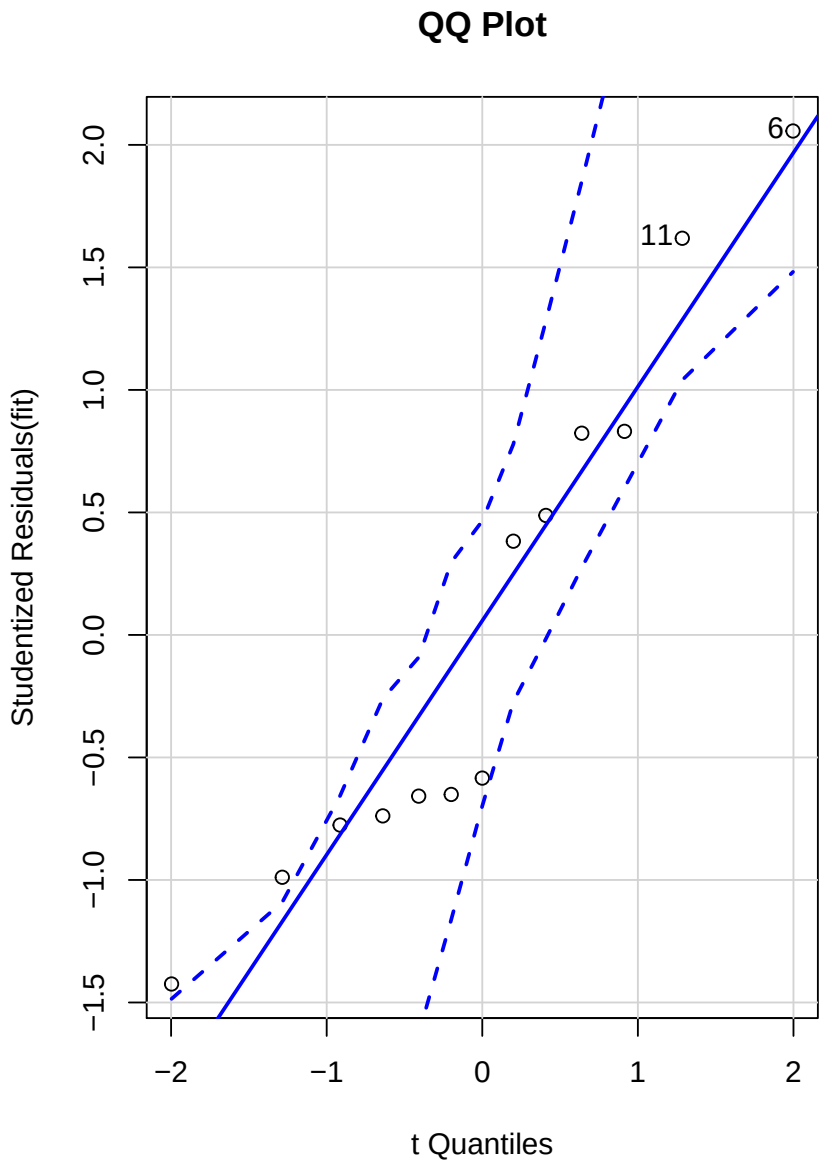
Whatever technique is used, the final model(s) selected should be assessed via residual diagnostics.

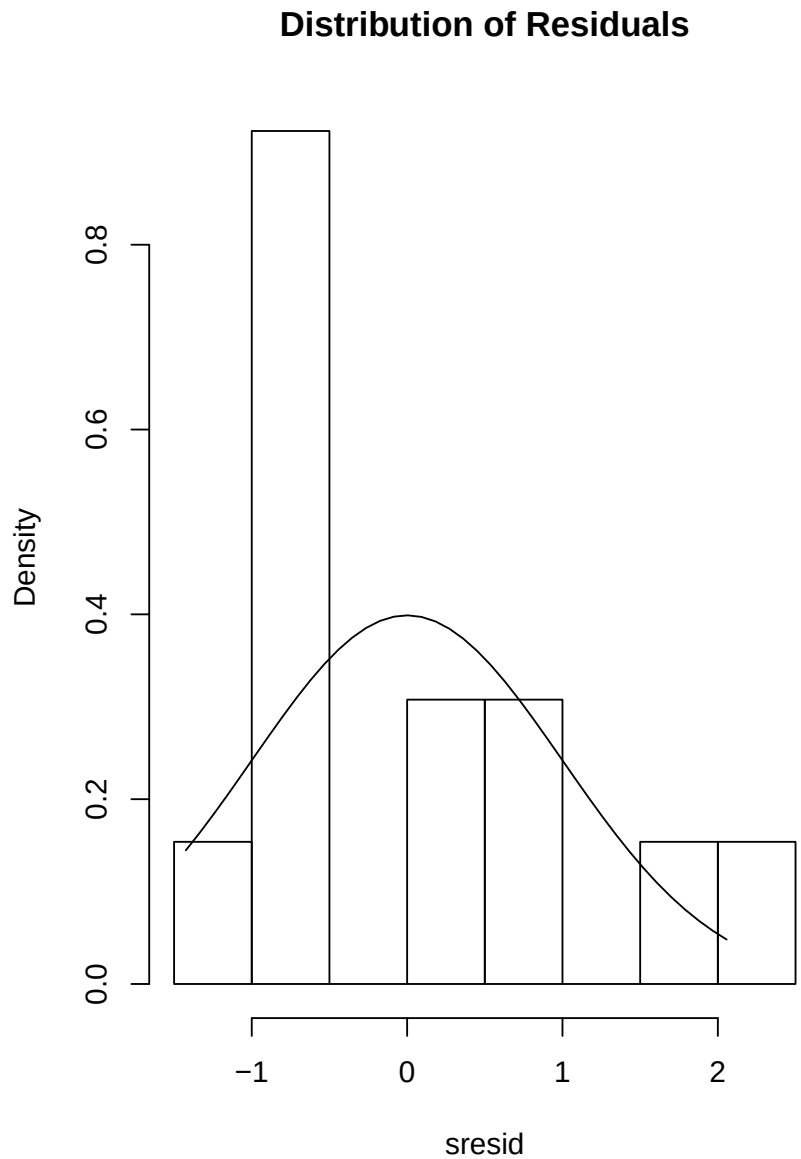
Regression Assumptions

- Linearity
- Independence
- Normally distribution
- Homogeneity of variance of the error

Diagnostic Plots

```
## [1]  6 11
```





Ridge and Lasso

Advantages of Ridge Regression

- Differentiate "important" from "less important" predictors and avoids overfitting
- Can handle datasets with more predictors than observations
- Handle multicollinearity problem

Penalty parameter λ

In ridge regression, we minimize

sum of squared residuals$+\lambda\times$sum of squared estimated parameters

- $\lambda \longrightarrow 0, \quad \hat{\beta}_{ridge} \longrightarrow \hat{\beta}_{OLS}$
- $\lambda \longrightarrow \infty, \quad \hat{\beta}_{ridge} \longrightarrow 0$

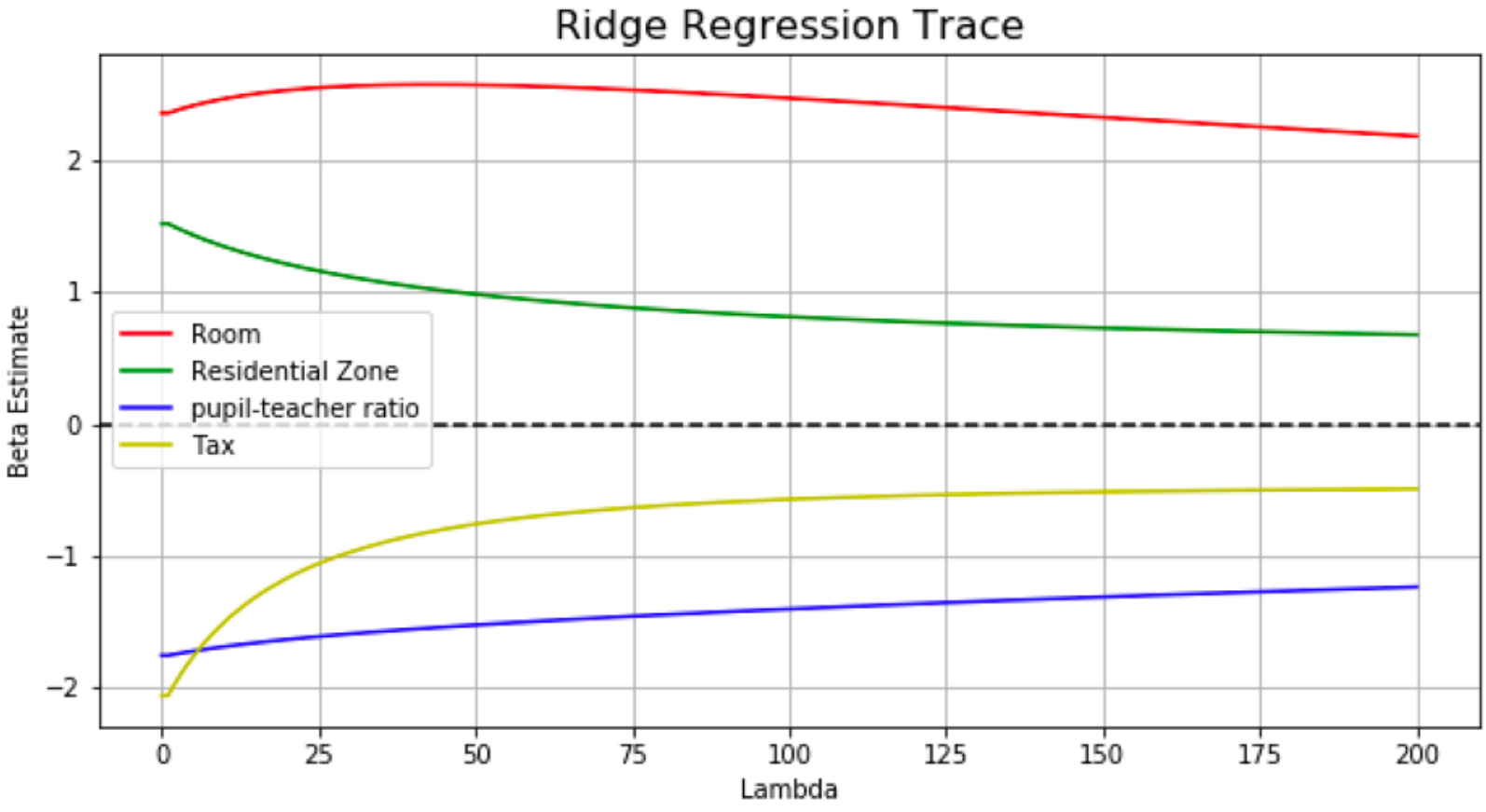
Example

Fit the model

$$\hat{\text{house price}} = \hat{\beta}_1 \text{Room} + \hat{\beta}_2 \text{Residential Zone} + \hat{\beta}_3 \text{pupil-teacher ratio}$$

$$+ \hat{\beta}_4 \text{tax} + \lambda(\beta_1^2 + \beta_2^2 + \beta_3^2 + \beta_4^2)$$

Ridge Trace Plot



Ridge Trace Plot

- Variables in red and blue line are more important, since they do not shrink a lot over iteration.
- The green and yellow lines represent the less important variables, since they go to zero remarkably

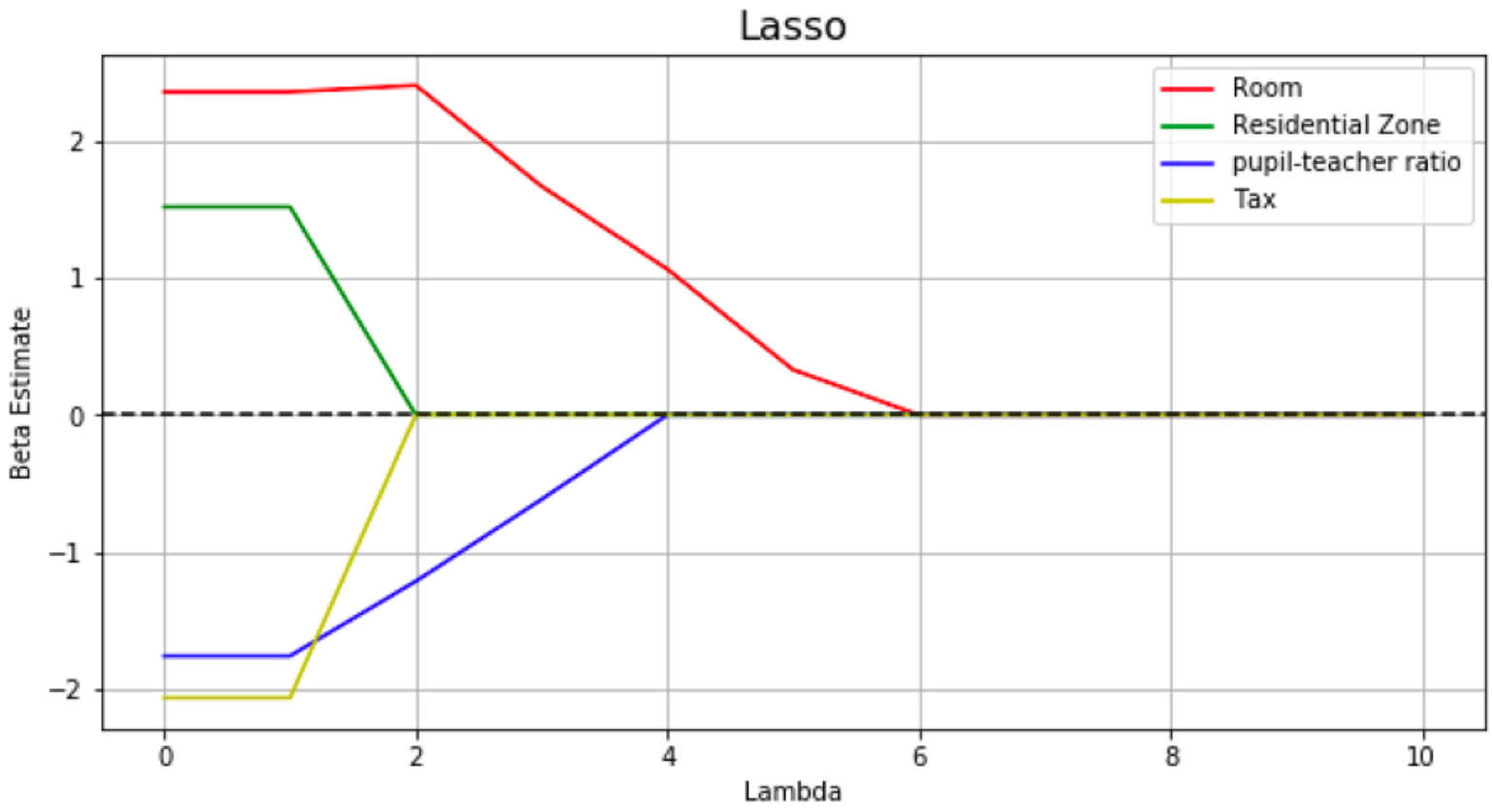
LASSO

- Similar conceptually to ridge regression
- Penalize the sum of absolute values of estimated paramters
- Coefficients can be exactly zero under lasso

Advantages of LASSO

- Avoid overfitting
- It is fast in terms of inference and fitting
- Can do variable selection.

Example


Lasso
Room
Residential Zone
pupil-teacher ratio
Tax
Beta Estimate
Lambda
2
1
0
−1
−2
0
2
4
6
8
10

Variable Selection Results

λ	Selected variables
0	Room, Residential Zone, Pupil-teacher Ratio, Tax
3	Room, Pupil-teacher Ratio
5	Room

Comparison of Ridge and Lasso

	Ridge	**Lasso**
Common	• Avoid overfitting • Differentiate "important" from "less important" predictors • Can handle datasets with more predictors than observations	
	• handle multicollinearity problem	• Can do feature selection

Elastic Net

Elastic Net

Elastic net

- Combines the penalties of ridge regression and LASSO

$$\hat{\beta} = \arg\min_{\beta} \mathit{LossFunction} + \lambda_2 ||\beta||^2 + \lambda_1 ||\beta||_1$$

$$\hat{\beta} = \arg\min_{\beta} \mathit{LossFunction} + \lambda \left(\frac{1-\alpha}{2} ||\beta||^2 + \alpha ||\beta||_1 \right)$$

Why elastic net

LASSO

- Selects at most n variables if $p > n$
- Fails to do grouped selection

Ridge regression

- Can't do variable selection

Why elastic net

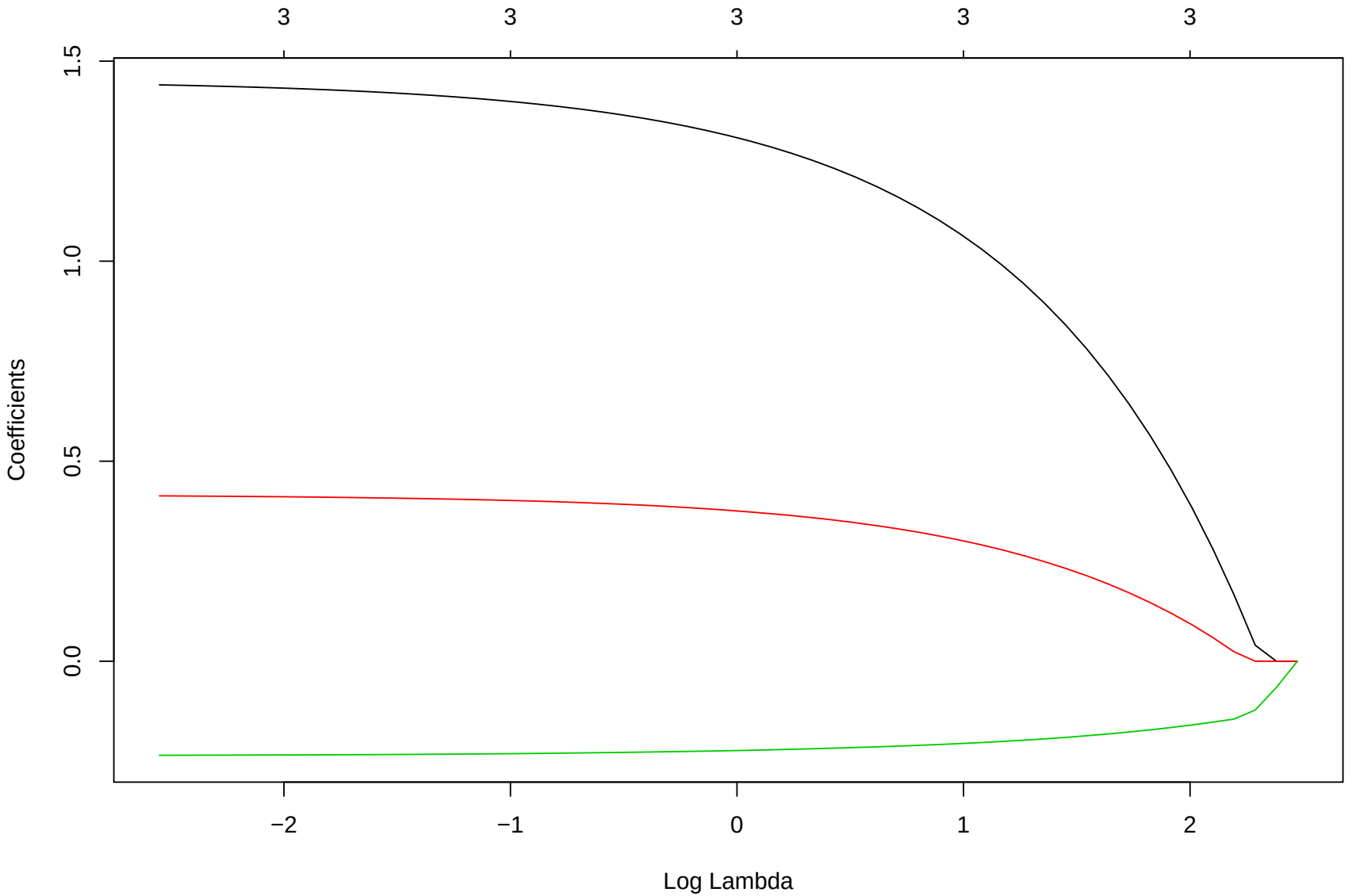
- Removes the limitation on the number of selected variables
- Can deal with group effects
- Stabilizes the regularization path

Example 1

```
library(glmnet)
dat <- read.csv("hald.csv")
mod1 <- glmnet(as.matrix(dat[,1:4]), dat$Y,
               family = "gaussian",
               alpha = 1,
               standardize = TRUE)
```

Example 1

```
plot(mod1, xvar = "lambda")
```



Example 1

```
search <- NULL
for(i in seq(0, 1, 0.05)){
  cv <- cv.glmnet(as.matrix(dat[,1:4]), dat$Y,
                  nfolds = 13, family = "gaussian",
                  alpha = i, standardize = TRUE)
  search <- rbind(search,
    data.frame(cvm = cv$cvm[cv$lambda == cv$lambda.1se],
               lambda.1se = cv$lambda.1se, alpha = i))
}
cv.optim <- search[search$cvm == min(search$cvm), ]
cv.optim
```

```
##         cvm lambda.1se alpha
## 5 8.637488    1.57649   0.2
```

Example 1

```
mod2 <- glmnet(as.matrix(dat[,1:4]), dat$Y,
               family = "gaussian",
               lambda = cv.optim$lambda.1se,
               alpha = cv.optim$alpha,
               standardize = TRUE)
coef(mod2)
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 85.6131282
## X1           1.1148588
## X2           0.2895057
## X3          -0.2183468
## X4          -0.3293218
```

Example 2

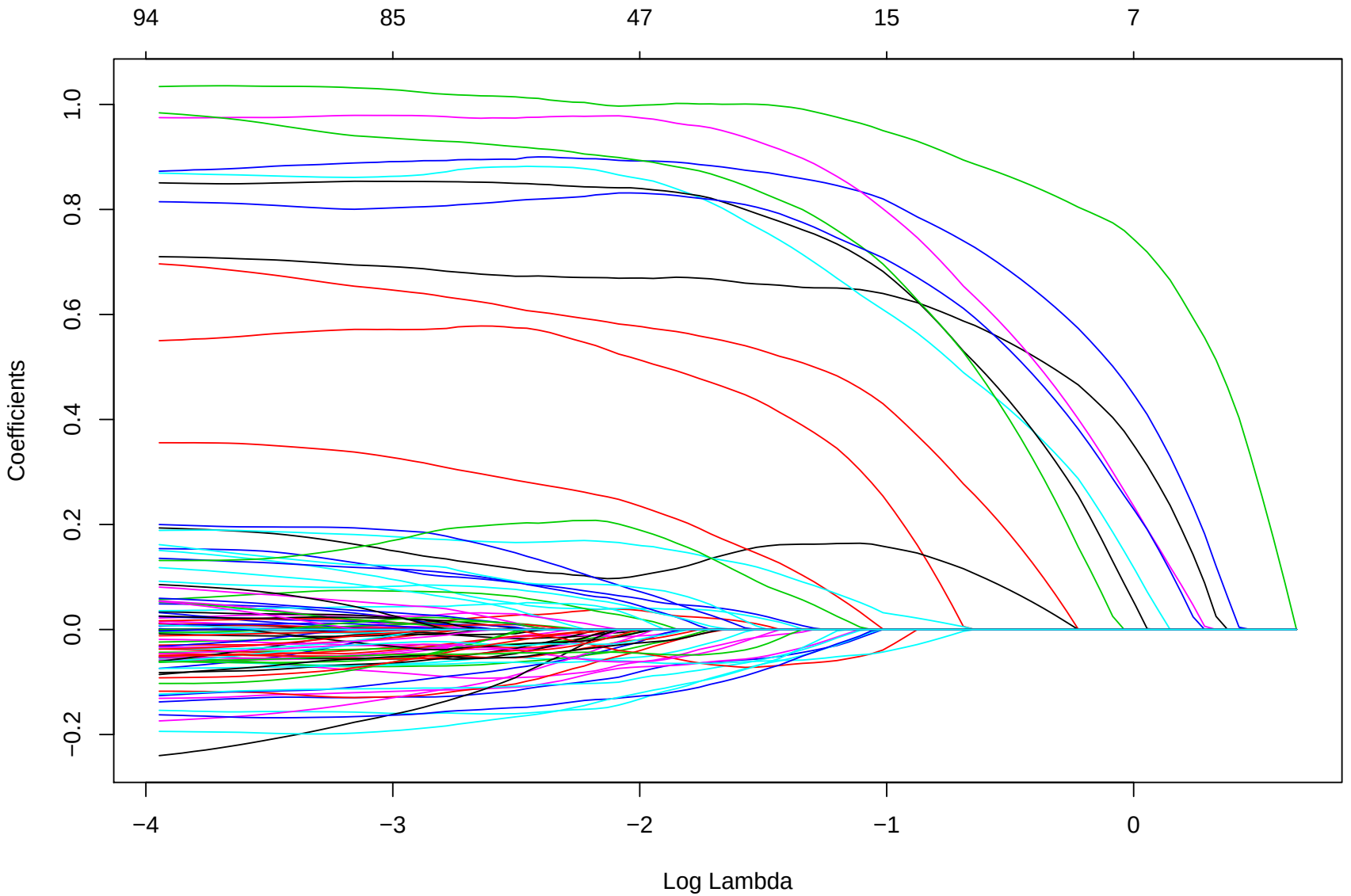
Generate data:

- 100 observations, 500 predictors, $p > n$
- Example in real life: metabolites data

```
set.seed(20191113)
n <- 100
p <- 500
x <- matrix(rnorm(n * p), nrow = n, ncol = p)
y <- apply(x[, 1:10], 1, sum) + rnorm(n)
```

Example 2

```
fit1 <- glmnet(x, y)
plot(fit1, xvar = "lambda")
```



Example 2

```
search <- NULL
for(i in seq(0, 1, 0.05)){
  cv <- cv.glmnet(x, y, nfolds = 10,
                  family = "gaussian", alpha = i)
  search <- rbind(search,
    data.frame(cvm = cv$cvm[cv$lambda == cv$lambda.1se],
               lambda.1se = cv$lambda.1se, alpha = i))
}
cv.optim <- search[search$cvm == min(search$cvm), ]
cv.optim
```

```
##          cvm lambda.1se alpha
## 13 2.412281  0.3005463   0.6
```

Example 2

```
fit2 <- glmnet(x, y,
               family = "gaussian",
               lambda = cv.optim$lambda.1se,
               alpha = cv.optim$alpha)
```

Example 2

```
coef(fit2)
```

```
## 501 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)  0.2406510170
## V1           0.6490770984
## V2           0.4155387383
## V3           0.9763024455
## V4           0.8547272676
## V5           0.7776405867
## V6           0.9138035033
## V7           0.7669011284
## V8           0.5237985022
## V9           0.8320473937
## V10          0.7852401032
## V11          .
## V12          .
## V13          .
## V14          .
## V15          .
## V16          .
## V17          .
## V18          .
## V19          .
## V20          .
## V21          .
## V22          .
## V23          .
## V24          .
## V25          .
## V26          .
## V27          .
## V28         -0.0354343658
## V29          .
```

Example 2

Among 500 variables, 38 variables are selected into the model.

```
out <- coef(fit2)
out@i
```

```
##  [1]   0   1   2   3   4   5   6   7   8   9  10  28  32  58  93 118 119
## [18] 123 145 173 178 228 237 245 257 258 274 287 288 312 319 367 372 409
## [35] 412 439 481 484 494
```

```
out@x
```

```
##  [1]  0.2406510170  0.6490770984  0.4155387383  0.9763024455  0.8547272676
##  [6]  0.7776405867  0.9138035033  0.7669011284  0.5237985022  0.8320473937
## [11]  0.7852401032 -0.0354343658  0.0099676973  0.0402904090  0.0228600030
## [16] -0.0733270989 -0.0098655825  0.1505796171  0.1884936394  0.1573570476
## [21] -0.0767757508 -0.0117450076 -0.0074926533  0.0393149682 -0.0007993162
## [26] -0.0306790655 -0.1005391946  0.0525005181 -0.0721526252 -0.0628331302
## [31] -0.0047535578  0.0415721052  0.0586725872 -0.0724380682 -0.0442021877
## [36]  0.1589291683 -0.1083150609 -0.0962665312 -0.0464256708
```

Summary

- Stepwise selection, also forward selection and backward selection, is simple and intuitive, but has many limitations
- Ridge regression can deal with multicollinearity but can not do variable selection
- Lasso can do variable selection, especially when $p > n$
- Elatic net is a combination of ridge regression and Lasso, and can do variable selection in more general situations