13) Ridge Regression and Least Absolute Shrinkage and Selection Operator (LASSO)

Vitor Kamada

February 2019

Reference

Tables, Graphics, and Figures from:

1) An Introduction to Statistical Learning

James et al. (2017): Ch 6.2, and 6.6

2) The Elements of Statistical Learning

Hastie et al. (2017): Ch 3.3, and 3.4

Stamey et al. (1989)

Y: log of Prostate-Specific Antigen

lcavol: log cancer volume

lweight: log prostate weight

lbph: log of the amount of benign prostatic hyperplasia

svi: seminal vesicle invasion

lcp: log of capsular penetration

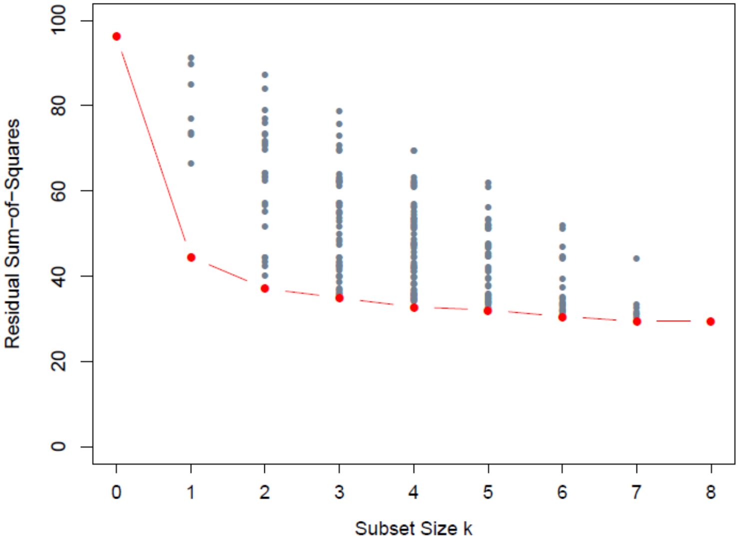
gleason: Gleason score

pgg45: Gleason scores 4 or 5

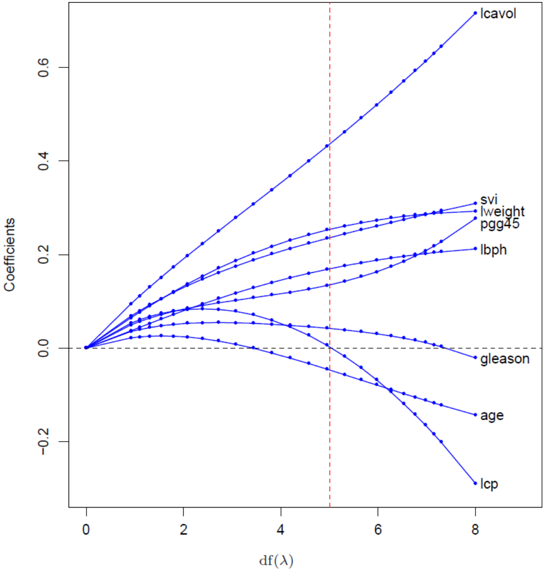
Tenfold Cross-Validation

Term	LS	Best Subset	Ridge	Lasso
Intercept	2.465	2.477	2.452	2.468
lcavol	0.680	0.740	0.420	0.533
lweight	0.263	0.316	0.238	0.169
age	−0.141		−0.046	
lbph	0.210		0.162	0.002
svi	0.305		0.227	0.094
lcp	−0.288		0.000	
gleason	−0.021		0.040	
pgg45	0.267		0.133	
Test Error	0.521	0.492	0.492	0.479
Std Error	0.179	0.143	0.165	0.164

Best-Subset Selection (Prostate Cancer)



Ridge Coefficients for the Prostate Cancer



Ridge Regression

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2}}$$

$$\frac{\|\hat{\beta}_\lambda^R\|_2}{\|\hat{\beta}\|_2}$$

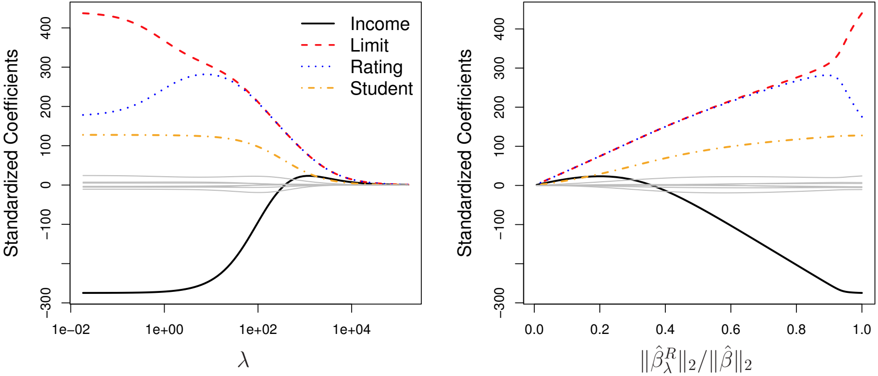
$$\|\beta\|_2 = \sqrt{\sum_{j=1}^{p}\beta_j^2}$$

Ridge Regression - Matrix Form

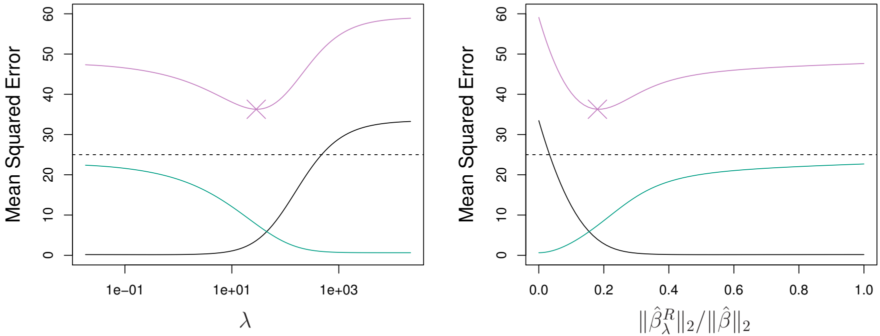
$$RSS(\lambda) = (y - X\beta)^T (y - X\beta)^T + \lambda \beta^T \beta$$

$$\hat{\beta}^{ridge} = (X^T X + \lambda I)^{-1} X^T y$$

Credit Data Set



Ridge: Squared Bias (Black), Variance (Green), and Test Mean Squared Error (Pink)



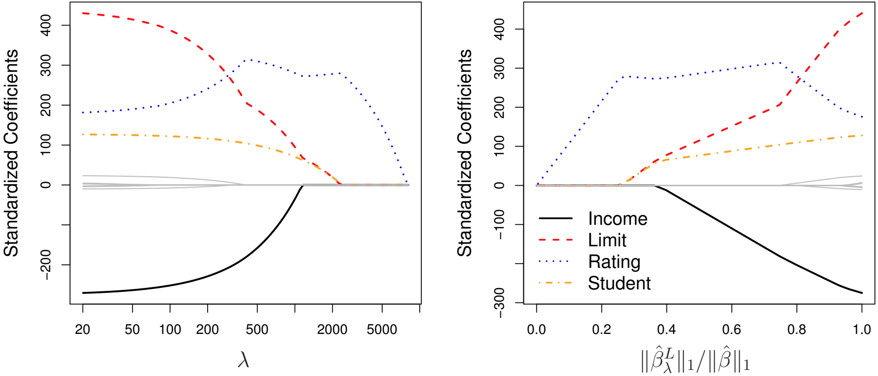
Least Absolute Shrinkage and Selection Operator (LASSO)

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

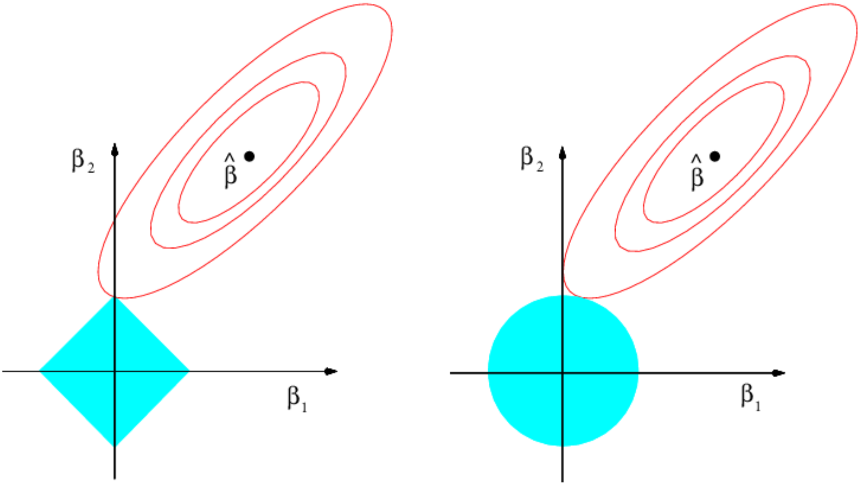
$$\frac{||\hat{\beta}_\lambda^L||_1}{||\hat{\beta}||_1}$$

$$||\beta||_1 = \Sigma|\beta_j|$$

The Standardized Lasso Coefficients

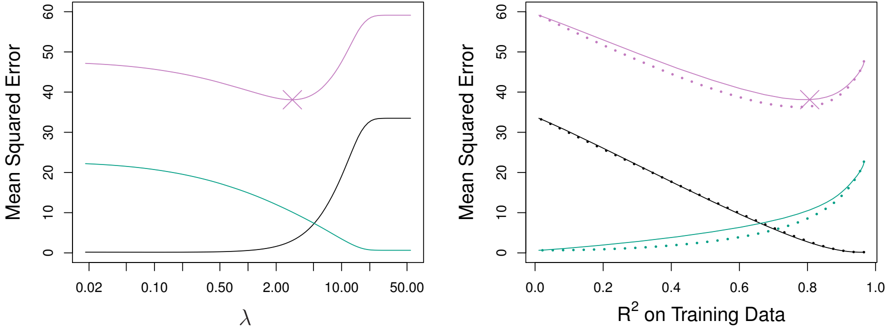


$|\beta_1| + |\beta_2| \leq s$ and $\beta_1^2 + \beta_2^2 \leq s$

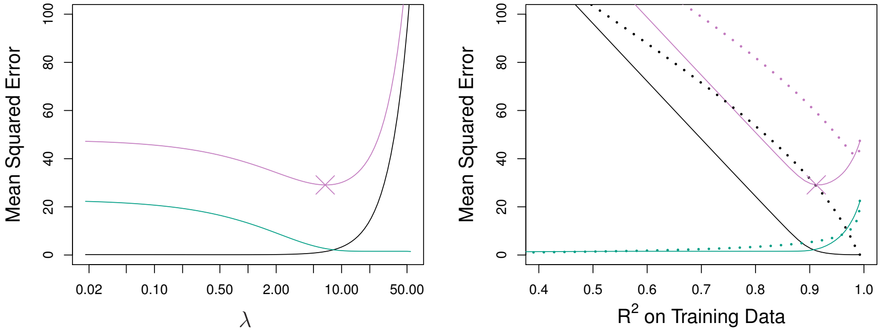


45 Xs related to Y: Lasso (Solid) vs Ridge (Dotted)

Squared Bias (Black), Variance (Green), and Test MSE (Pink)



Squared Bias (Black), Variance (Green), and Test MSE (Pink)



$n = p$ and X a Diagonal Matrix with 1's

$$\sum_{j=1}^{p} (y_j - \beta_j)^2$$

$$\hat{\beta}_j = y_j$$

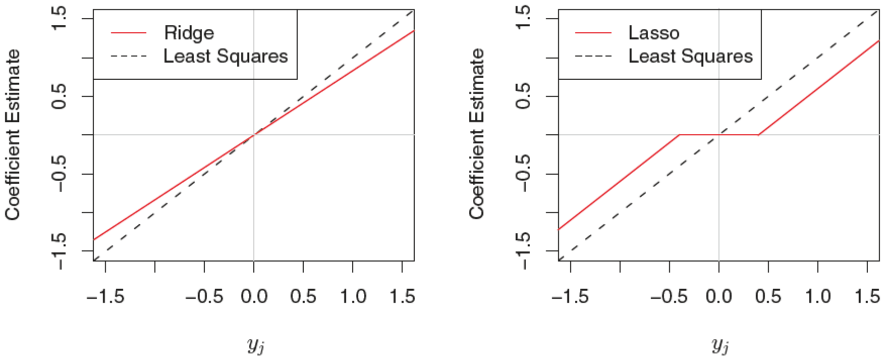
$$\sum_{j=1}^{p} (y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

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

Ridge and Lasso Regression

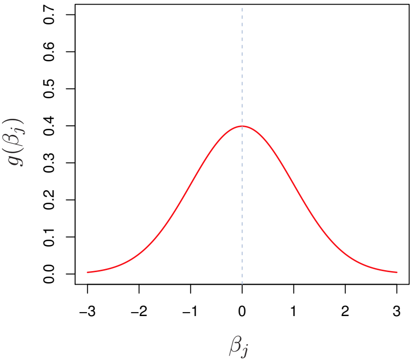
$$\sum_{j=1}^{p} (y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

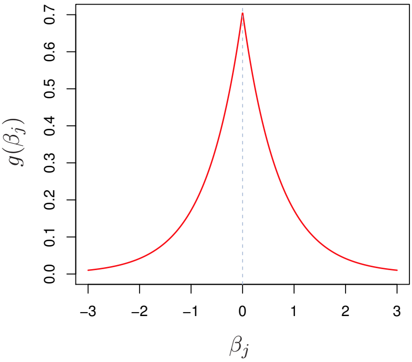
$$\hat{\beta}_j^R = y_j/(1+\lambda)$$



Gaussian Prior vs Double-Exponential Prior

$$p(\beta|X,Y) \propto f(Y|X,\beta)p(\beta|X)$$





library(glmnet)

```
x=model.matrix(Salary~.,Hitters)[,-1]
y=Hitters$Salary
grid=10^seq(10,-2,length=100)
ridge.mod=glmnet(x,y,alpha=0,lambda=grid)
dim(coef(ridge.mod))
```

20 rows (one for each $Xs + \beta_0$)

100 columns (λ)

ridge.mod$lambda[50]

$$\lambda = 11497.57$$

coef(ridge.mod)[,50]

```
 (Intercept)        AtBat         Hits        HmRun
407.356050200  0.036957182  0.138180344  0.524629976
        Runs          RBI        Walks        Years
 0.230701523  0.239841459  0.289618741  1.107702929
      CAtBat        CHits       CHmRun        CRuns
 0.003131815  0.011653637  0.087545670  0.023379882
        CRBI       CWalks      LeagueN    DivisionW
 0.024138320  0.025015421  0.085028114 -6.215440973
     PutOuts      Assists       Errors   NewLeagueN
 0.016482577  0.002612988 -0.020502690  0.301433531
```

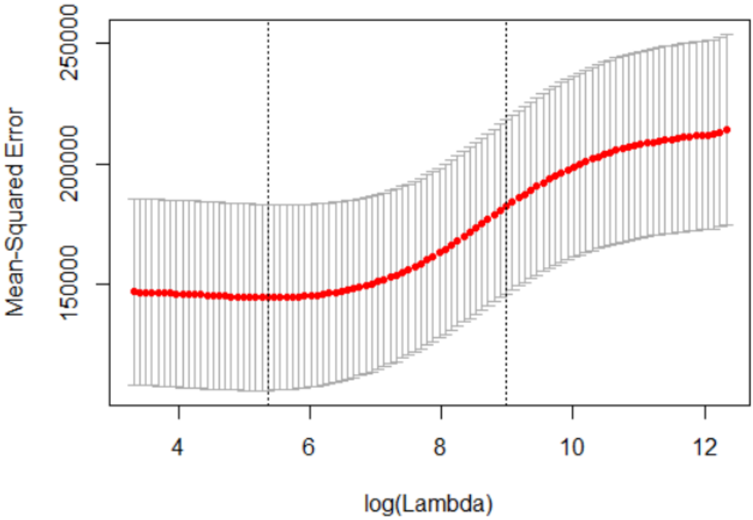
set.seed(1)

```
train=sample(1:nrow(x), nrow(x)/2)
test=(-train); y.test=y[test]
set.seed(1)
# 10 fold cross-validation
cv.out=cv.glmnet(x[train,],y[train],alpha=0)
bestlam=cv.out$lambda.min; bestlam
```

211.74

plot(cv.out)

$$\exp(5.356) = 211.74$$



MSE: Ridge ($\lambda = 211.74$) vs OLS vs Intercept

```
ridge.pred=predict(ridge.mod,s=bestlam, newx=x[test,])
mean((ridge.pred-y.test)^2)
```

96015.51

```
ridge.predOLS=predict(ridge.mod,s=0, newx=x[test,])
mean((ridge.predOLS-y.test)^2)
```

114723.6

```
mean((mean(y[train])-y.test)^2)
```

193253.1

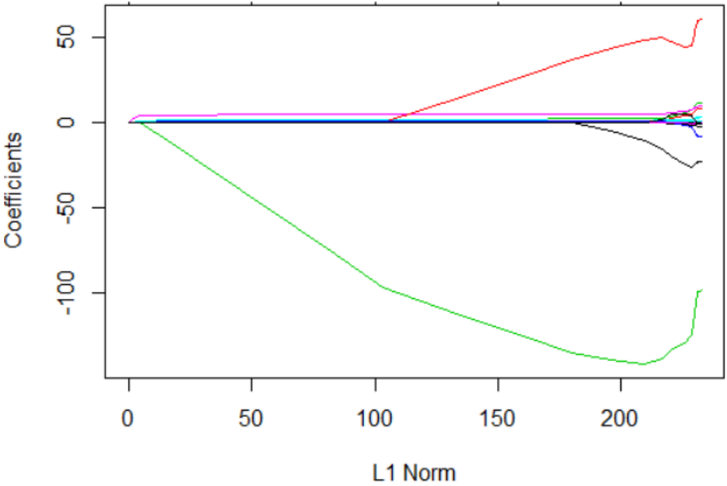
out=glmnet(x,y,alpha=0)

predict(out,type="coefficients",s=bestlam)[1:20,]

```
  (Intercept)         AtBat          Hits         HmRun
   9.88487157    0.03143991    1.00882875    0.13927624
         Runs           RBI         Walks         Years
   1.11320781    0.87318990    1.80410229    0.13074381
       CAtBat         CHits        CHmRun         CRuns
   0.01113978    0.06489843    0.45158546    0.12900049
         CRBI        CWalks       LeagueN     DivisionW
   0.13737712    0.02908572   27.18227535  -91.63411299
      PutOuts       Assists        Errors    NewLeagueN
   0.19149252    0.04254536   -1.81244470    7.21208390
```

lasso.mod=glmnet(x[train,],y[train], alpha=1,lambda=grid)

plot(lasso.mod)



set.seed(1)

cv.out=cv.glmnet(x[train,], y[train],alpha=1)

bestlam=cv.out$lambda.min; bestlam

16.78

log(bestlam)

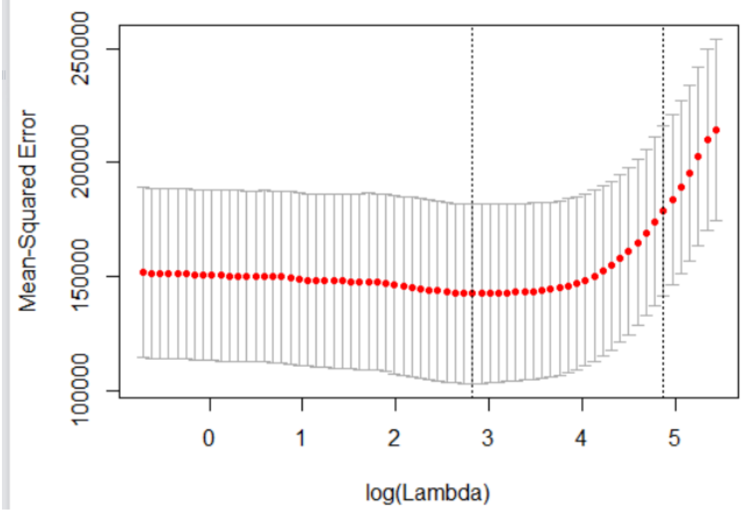
2.82

lasso.pred=predict(lasso.mod,s=bestlam,newx=x[test,])

mean((lasso.pred-y.test)^2)

100743.4

plot(cv.out)



Mean-Squared Error
100000
150000
200000
250000
0
1
2
3
4
5
log(Lambda)

out=glmnet(x,y,alpha=1,lambda=grid)

lasso.coef=predict(out, type="coefficients", s=bestlam)[1:20,]

```
(Intercept)        AtBat          Hits        HmRun
 18.5394844    0.0000000     1.8735390    0.0000000
       Runs          RBI         Walks        Years
  0.0000000    0.0000000     2.2178444    0.0000000
     CAtBat        CHits        CHmRun        CRuns
  0.0000000    0.0000000     0.0000000    0.2071252
       CRBI       CWalks       LeagueN    DivisionW
  0.4130132    0.0000000     3.2666677 -103.4845458
    PutOuts      Assists        Errors   NewLeagueN
  0.2204284    0.0000000     0.0000000    0.0000000
```