

Ridge Lasso Elastic-net - Demo

Comparisons of model coefficient estimate

STAT 32950-24620

Spring 2025 (wk8)

1 / 20

Regularized linear regressions

Comparisons of **Ridge, Lasso, and Elastic-net**

Goal: Parameter estimation in linear regression

$$Y = \beta_0 + \mathbf{X}\beta + \epsilon$$

```
library(MASS)      # use lm.ridge
library(glmnet)
```

2 / 20

Estimators

$$\hat{\beta}_{LS} = \operatorname{argmin}_{\beta} \{ \|Y - \beta_0 - \mathbf{X}\beta\|_2^2 \}$$

$$\hat{\beta}_{Ridge} = \operatorname{argmin}_{\beta} \{ \|Y - \beta_0 - \mathbf{X}\beta\|_2^2 + \lambda_{ridge} \|\beta\|_2^2 \}$$

$$\hat{\beta}_{Lasso} = \operatorname{argmin}_{\beta} \{ \|Y - \beta_0 - \mathbf{X}\beta\|_2^2 + \lambda_{lasso} \|\beta\|_1 \}$$

$$\hat{\beta}_{E-net} = \operatorname{argmin}_{\beta} \{ \|Y - \beta_0 - \mathbf{X}\beta\|_2^2 + \lambda [\alpha \|\beta\|_2^2 + (1 - \alpha) \|\beta\|_1] \}$$

3 / 20

Example 1: Explanatory variables X

$p = 30$ input X variables

$n = 50$ observations

```
set.seed(246329)
n=50
p1=10 # "signals"
p2=20 # "noise"
X = matrix(rnorm(n*(p1+p2)), n, p1+p2)
dim(X)

## [1] 50 30
```

4 / 20

Example 1: Response variable Y

10 signal, 20 uniform noise input variables

```
c1 = 0.5 + (runif(10))/2 # "signals" 0.5 + U(0,1)/2
c2 = (runif(20))*3/10    # "noise"  U(0,1)*3/10
```

```
Y = X[,1:10]%*%c1 + X[,11:30]%*%c2 + rnorm(n)
```

To fit the model

$$Y = \beta_o + \mathbf{X}\beta + \epsilon$$

In expanded form,

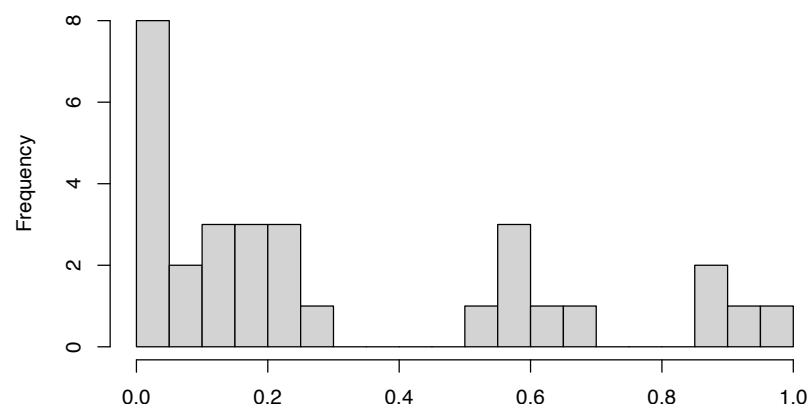
$$Y = \beta_o + \beta_1 X_1 + \dots + \beta_{10} X_{10} + \beta_{11} X_{11} + \dots + \beta_{30} X_{30} + \epsilon$$

5 / 20

True β values in $Y = \beta_o + \mathbf{X}\beta + \epsilon$

```
#range(c1)
#range(c2)
hist(c(c1,c2),nclass=15)
```

Histogram of $c(c1, c2)$



6 / 20

Least Squares for $Y = \beta_o + \mathbf{X}\beta + \epsilon$

```
Y = X[,1:10]%*%c1 + X[,11:30]%*%c2 + rnorm(n)
LSfit = lm(Y~X)
round(LSfit$coeff,2)
```

## (Intercept)	X1	X2	X3
## -0.18	0.35	0.92	0.28
## X6	X7	X8	X9
## 0.68	0.87	0.83	0.28
## X12	X13	X14	X15
## 0.03	0.14	-0.24	0.35
## X18	X19	X20	X21
## 0.12	-0.12	0.19	0.23
## X24	X25	X26	X27
## -0.35	-0.11	0.22	0.47
## X30			
## 0.29			

7 / 20

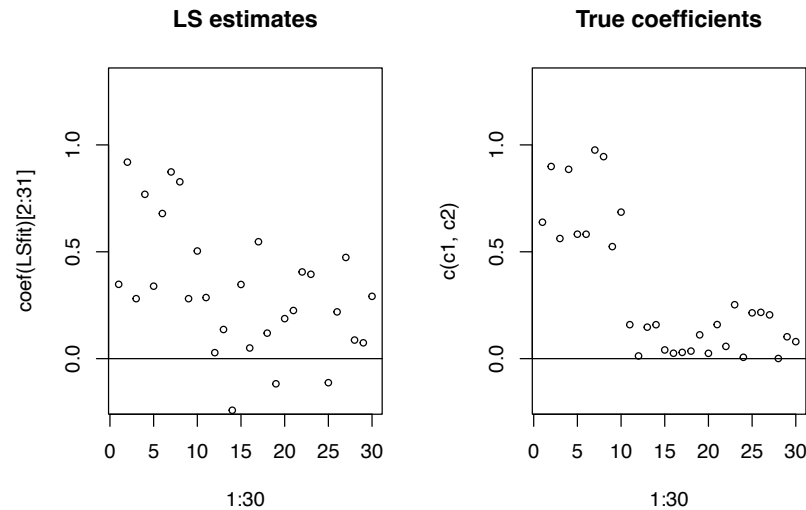
Compare model fitted $\hat{\beta}$ with true β (code)

```
par(mfrow=c(1,2)) # Get LS beta
plot(1:30,coef(LSfit)[2:31],cex=.7,ylim=c(-.2,1.3),
     main="LS estimates");abline(h=0)
plot(1:30,c(c1,c2),cex=.7,ylim=c(-.2,1.3),
     main="True coefficients");abline(h=0)

par(mfrow=c(1,2)) # Get Ridge beta
plot(1:30,coef(lm.ridge(Y~X,lambda=25))[2:31],cex=.7,
     ylim=c(-.2,1.3),main="Ridge estimates")
abline(h=0)
plot(1:30,c(c1,c2),cex=.7,ylim=c(-.2,1.3),
     main="True coefficients");abline(h=0)
```

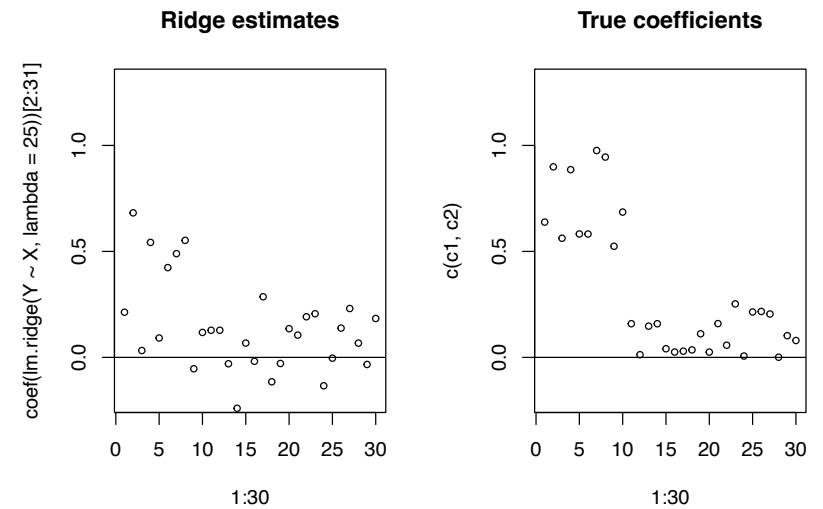
8 / 20

$\hat{\beta}_{LS}$ values for $Y = \beta_o + \mathbf{X}\beta + \epsilon$



9 / 20

$\hat{\beta}_{Ridge}$ values for $Y = \beta_o + \mathbf{X}\beta + \epsilon$



10 / 20

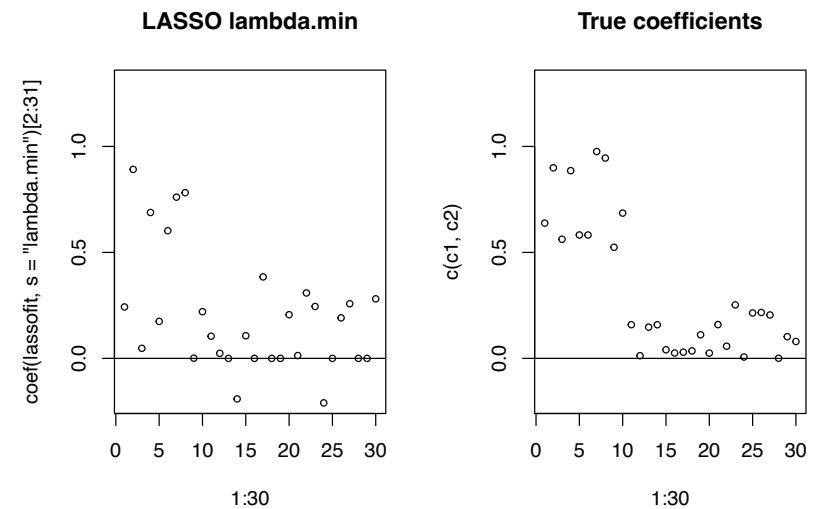
$\hat{\beta}_{Lasso}$ values for $Y = \beta_o + \mathbf{X}\beta + \epsilon$

```
lassofit = cv.glmnet(X,Y)
par(mfrow=c(1,2)) # Get LASSO beta, lambda-min
plot(1:30,coef(lassofit,s="lambda.min")[2:31],cex=.7,
     ylim=c(-.2,1.3),main="LASSO lambda.min");
abline(h=0)
plot(1:30,c(c1,c2),cex=.7,ylim=c(-.2,1.3),
     main="True coefficients");abline(h=0)
```

```
lassofit = cv.glmnet(X,Y)
par(mfrow=c(1,2)) # Get LASSO beta, lambda-1se
plot(1:30,coef(lassofit,s="lambda.1se")[2:31],cex=.7,
     ylim=c(-.2,1.3),main="LASSO lambda.1se");
abline(h=0)
plot(1:30,c(c1,c2),cex=.7,ylim=c(-.2,1.3),
     main="True coefficients");abline(h=0)
```

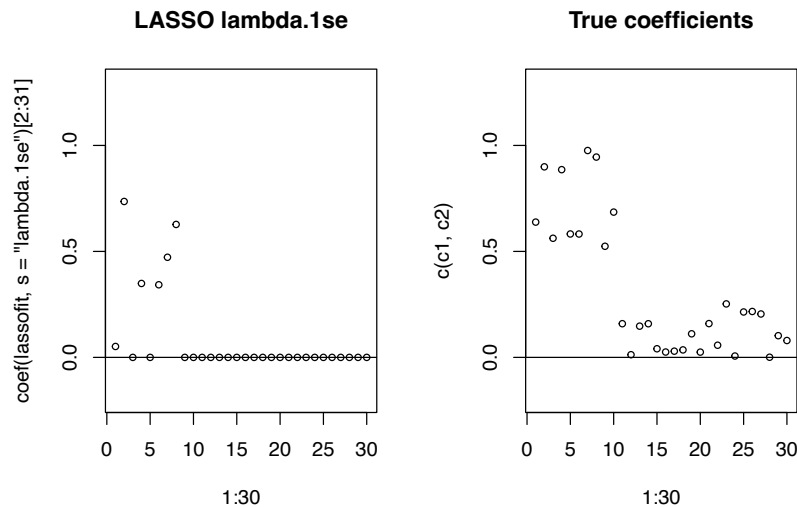
11 / 20

$\hat{\beta}_{Lasso}$ values when $\lambda = \text{lambda.min}$



12 / 20

$\hat{\beta}_{Lasso}$ values when $\lambda = \text{lambda.1se}$



13 / 20

Model parameter estimates comparison

$$Y = \beta_o + \mathbf{X}\beta + \epsilon$$

We compare value and range of the true coefficients

$$\beta = (\beta_1, \dots, \beta_{30})$$

with estimated $\hat{\beta}$ values

$$\hat{\beta} = (\hat{\beta}_1, \dots, \hat{\beta}_{30})$$

Using the method

- Least Squares
- Ridge
- Lasso
- Elastic-net

14 / 20

Visualize model parameter estimates $\hat{\beta}$

Fit linear regression model $Y = \beta_o + \mathbf{X}\beta + \epsilon$ by

- Least Squares
- Ridge
- Lasso
- Elastic-net

```
netfit = cv.glmnet(X,Y,alpha=0.5)
plot(rep(1,30),c(c1,c2),cex=.5,col=c(rep(2,10),rep(1,20)),
     xlim=c(0.5,6.5),ylim=c(-.5,1.5),xlab="",ylab="")
abline(h=0,lty=2)
```

15 / 20

(Code)

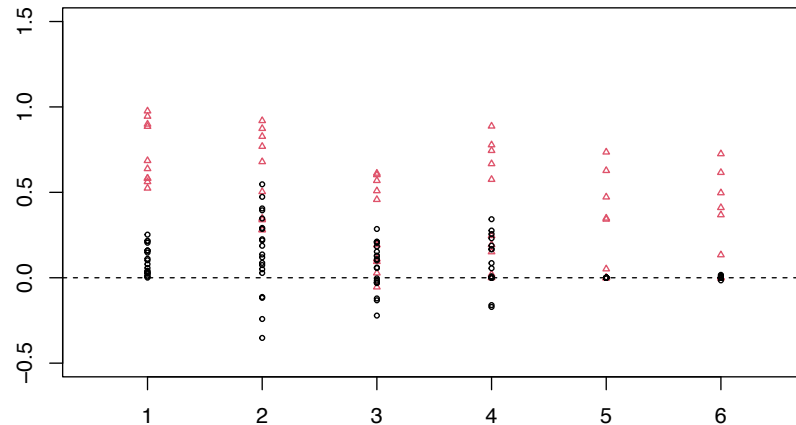
```
points(rep(2,30),lm(Y~X)$coef[2:31],cex=.5,
       col=c(rep(2,10),rep(1,20)), pch=c(rep(2,10),rep(1,20)))
points(rep(3,30),lm.ridge(Y~X,lambda=25)$coef,cex=.5,
       col=c(rep(2,10),rep(1,20)), pch=c(rep(2,10),rep(1,20)))
points(rep(4,30),coef(lassofit,s="lambda.min")[2:31],cex=.5,
       col=c(rep(2,10),rep(1,20)), pch=c(rep(2,10),rep(1,20)))
points(rep(5,30),coef(lassofit,s="lambda.1se")[2:31],cex=.5,
       col=c(rep(2,10),rep(1,20)), pch=c(rep(2,10),rep(1,20)))
points(rep(6,30),coef(netfit,s="lambda.1se")[2:31],cex=.5,
       col=c(rep(2,10),rep(1,20)), pch=c(rep(2,10),rep(1,20)))
title("True coefficients, LS, Ridge,
      Lasso.min, Lasso.1se, Net.1se")
```

16 / 20

Input variables: **Signal** (triangles) + uniform **Noise**

β , $\hat{\beta}_{LS}$, $\hat{\beta}_{Ridge}$, $\hat{\beta}_{Lasso.min}$, $\hat{\beta}_{Lasso.1se}$, $\hat{\beta}_{E-net}$

True coefficients, LS, Ridge, Lasso.min, Lasso.1se, Net.1se



17 / 20

Example 2: Signal + white noise

X: 10 Signal + 20 normal noise input variables

```
n=50; p1=10; p2=20
```

```
set.seed(19)
```

```
X = matrix(rnorm(n*(p1+p2)),n, p1+p2)
```

```
dim(X)
```

```
## [1] 50 30
```

```
c1 = 0.5 + runif(10)/2
```

"signals"

```
c2 = (rnorm(20))*3/40
```

"noise" ~ $N(0, \sigma^2)$

```
Y = X[,1:10]%%c1 + X[,11:30]%%c2 + rnorm(n)
```

```
lassofit = cv.glmnet(X,Y); par(mfrow=c(1,2))
```

$$Y = \beta_o + \mathbf{X}\beta + \epsilon$$

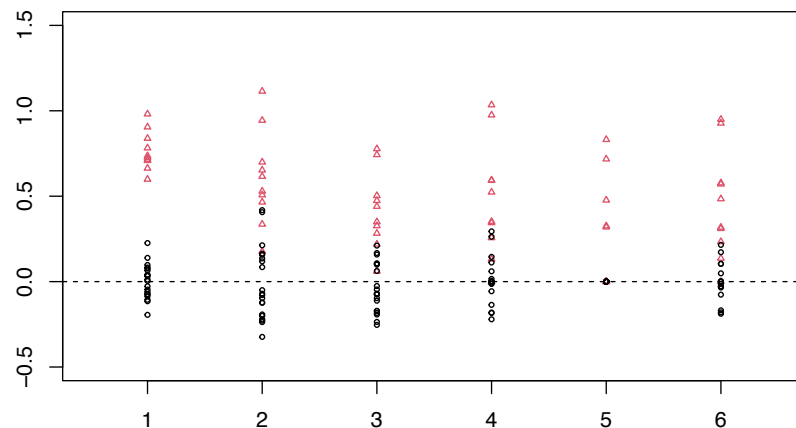
18 / 20

Example 2: Parameter estimates $\hat{\beta}$ comparison

Input variables: **Signal** (triangles) + **Noise** (normal)

β , $\hat{\beta}_{LS}$, $\hat{\beta}_{Ridge}$, $\hat{\beta}_{Lasso.min}$, $\hat{\beta}_{Lasso.1se}$, $\hat{\beta}_{E-net}$

True coefficients, LS, Ridge, Lasso.min, Lasso.1se, Net.1se



19 / 20

(Code)

```
netfit = cv.glmnet(X,Y,alpha=0.5)
```

```
plot(rep(1,30),c(c1,c2),cex=.5,col=c(rep(2,10),rep(1,20)),  
     xlim=c(0.5,6.5),ylim=c(-.5,1.5),xlab="",ylab="")
```

```
abline(h=0,lty=2)
```

```
points(rep(2,30),lm(Y~X)$coef[2:31],cex=.5,
```

```
      col=c(rep(2,10),rep(1,20)), pch=c(rep(2,10),rep(1,20)))
```

```
points(rep(3,30),lm.ridge(Y~X,lambda=25)$coef,cex=.5,
```

```
      col=c(rep(2,10),rep(1,20)), pch=c(rep(2,10),rep(1,20)))
```

```
points(rep(4,30),coef(lassofit,s="lambda.min")[2:31],cex=.5,
```

```
      col=c(rep(2,10),rep(1,20)), pch=c(rep(2,10),rep(1,20)))
```

```
points(rep(5,30),coef(lassofit,s="lambda.1se")[2:31],cex=.5,
```

```
      col=c(rep(2,10),rep(1,20)), pch=c(rep(2,10),rep(1,20)))
```

```
points(rep(6,30),coef(netfit,s="lambda.1se")[2:31],cex=.5,
```

```
      col=c(rep(2,10),rep(1,20)), pch=c(rep(2,10),rep(1,20)))
```

```
title("True coefficients, LS, Ridge,  
      Lasso.min, Lasso.1se, Net.1se")
```

20 / 20