4.6 The Lasso

The Lasso

The Lasso

The lasso is a shrinkage method like ridge regression, with subtle but significant differences. The lasso estimate is defined by the following modification of the ridge regression optimisation problem:

$$\mathrm{RSS}(\lambda) = \|\boldsymbol{y} - a_0\mathbf{1} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 = \sum_{i=1}^{n}(y_i - a_0 - x_i^\top\boldsymbol{\beta})^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

Again, an equivalent formulation of the problem is to minimise the RSS subject to

$$\sum_{j=1}^{p}|\beta_j| \leq s.$$

The same considerations on the scaling of predictors as for ridge regression apply to the lasso model.

The difference between ridge and lasso lies entirely in the form of the penalty term, which uses the L^1-norm instead of the (Euclidean) L^2-norm; see Figure 1.

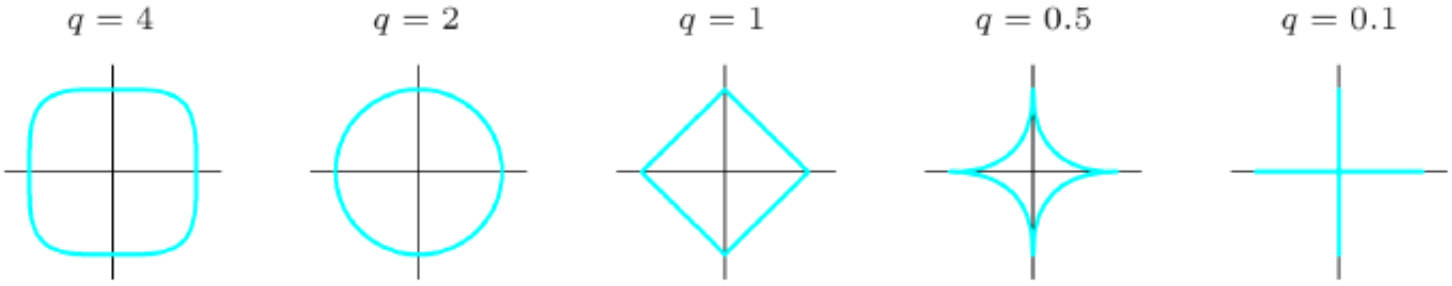


Figure 4.6.1: The contours $\|x\|_q = 1$, for different values of q. Note that $q = 1$ is the only value for which the L^q-norm is both convex and shows cusps.

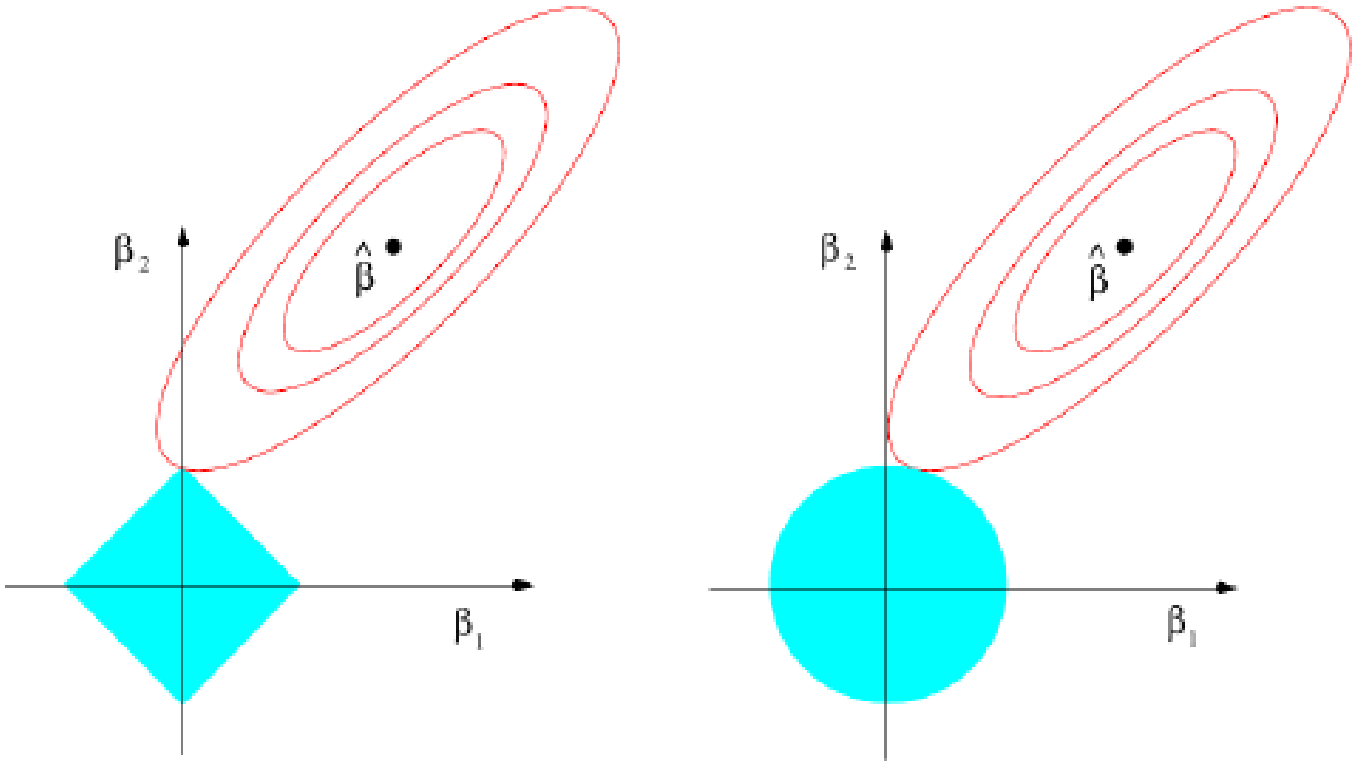


Figure 4.6.2: Restricted parameter set (blue) and contours of the least squares error function (red). The lasso (left) shrinks β_1 to 0 (exactly), whereas ridge regression (right) does not.

Comments:

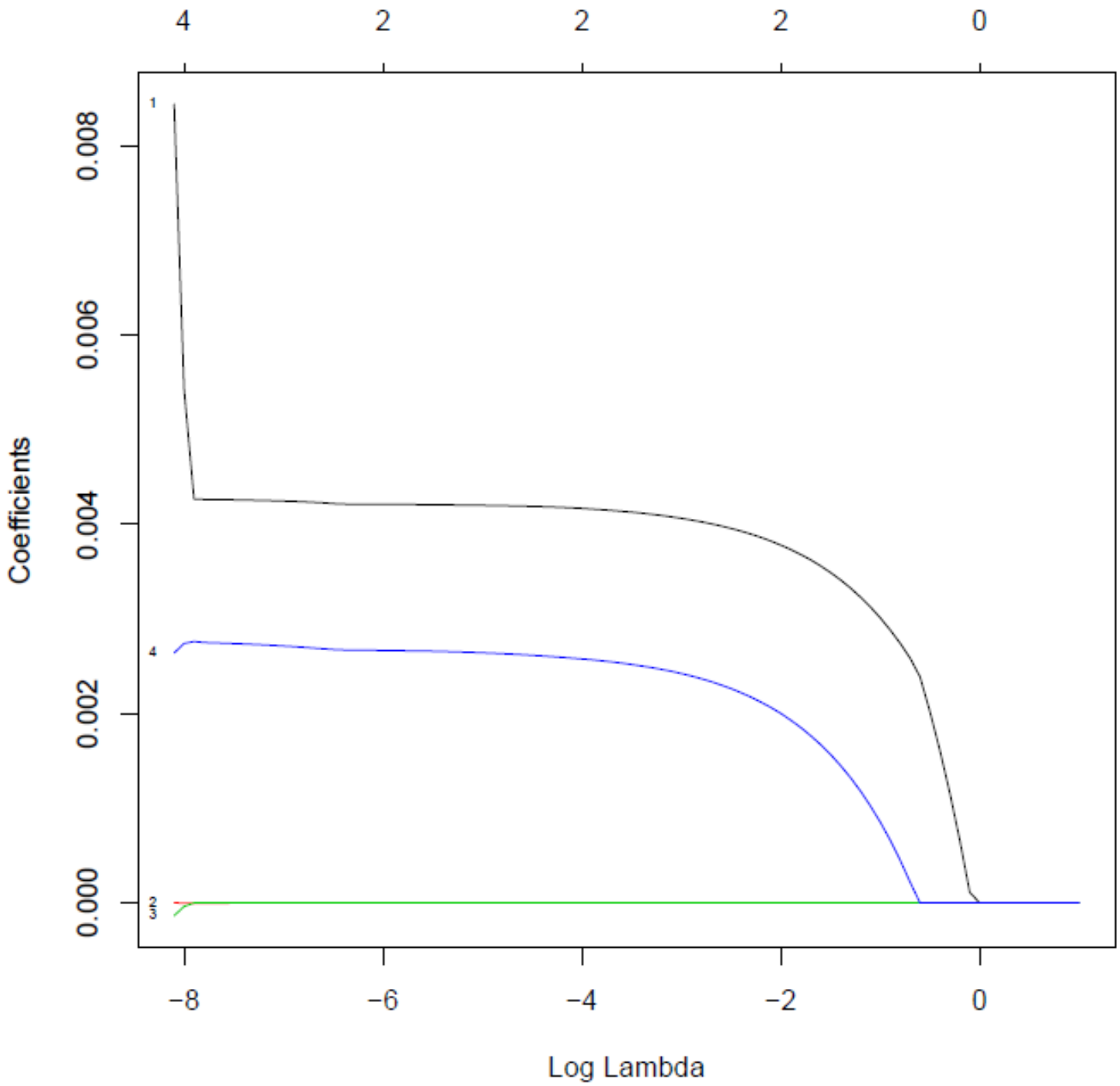
- $\hat{\beta}$ is the least squares estimate.
- If s is sufficiently large, the constraint regions would contain the LS estimates (corresponding to the case $\lambda = 0$)
- The *red ellipses* represent the regions of constant RSS: going far from $\hat{\beta}$ would increase RSS.
- Lasso and ridge estimates are given by *the first point at which an ellipse touch the constraint region*.
- Since lasso, differently from ridge, has corners on the axes, the ellipses are more likely to touch the constraint region on an axis.

As illustrated for $p = 2$ parameters in Figure 2, the lasso formulation of the restricted parameter space favours values for β which lie at a cusp, meaning that one or more coordinates of β are set to 0. A similar statement holds in higher dimensions. The **lasso hence both selects and shrinks the coefficients** β.

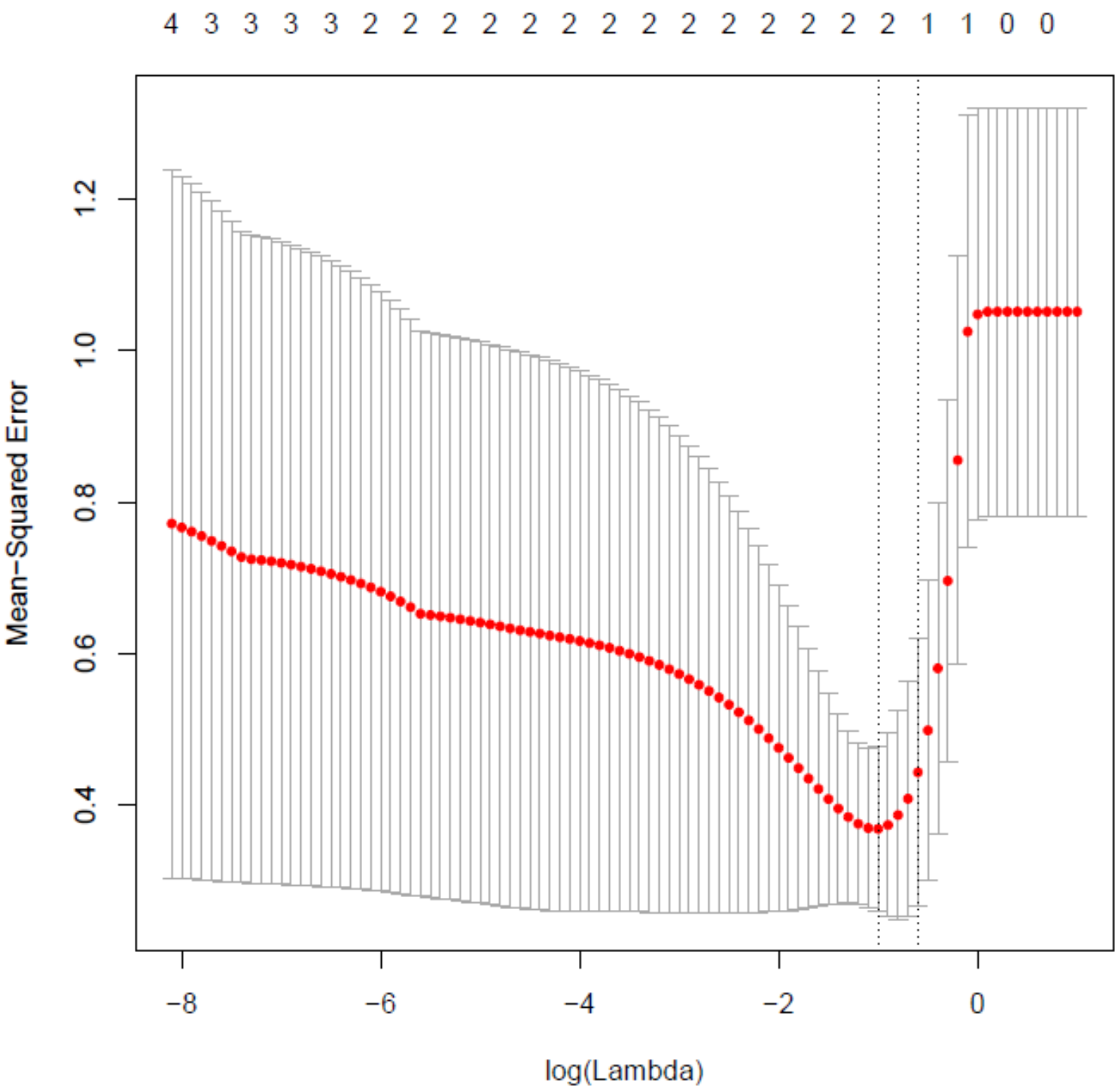
Example: Lasso Fit in R

Setting $\texttt{alpha} = 1$ in $glmnet()$ fits a lasso model.

```
require(bestglm)
require(glmnet)

data(manpower)
x=as.matrix(manpower[,1:4])
y=log(manpower$Hours)
lambda=exp(seq(-8.1,1,0.1))
fit2=glmnet(x,y,alpha=1,lambda=lambda)

set.seed(1) # For reproducibility
cv.fit2=cv.glmnet(x,y,alpha=1,nfolds=5,lambda=lambda)
plot(fit2,xvar="lambda",label=T)
```




```
require(bestglm)
require(glmnet)

data(manpower)
x=as.matrix(manpower[,1:4])
y=log(manpower$Hours)
lambda=exp(seq(-8.1,1,0.1))
fit2=glmnet(x,y,alpha=1,lambda=lambda)

set.seed(1) # For reproducibility
cv.fit2=cv.glmnet(x,y,alpha=1,nfolds=5,lambda=lambda)
plot(cv.fit2)
bestlam=cv.fit2$lambda.min
bestlam
```


4 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 0 0
Mean−Squared Error
0.4
0.6
0.8
1.0
1.2
−8
−6
−4
−2
0
log(Lambda)

Example: Lasso vs. Ridge Regression

In this example we will compare the predictive power of the Lasso and ridge regression using the **`Hitters`** dataset.

We will first define the model matrix and the response variable.

```
library(ISLR)
Hitters=na.omit(Hitters)
x=model.matrix(Salary~., Hitters)[,-1]
y=Hitters$Salary
```

Next, we will split the samples into a training set and a test set to estimate the test error of ridge regression and the lasso.

We perform cross-validation to choose λ for ridge regression.

```
require(bestglm)
require(glmnet)
library(ISLR)
Hitters=na.omit(Hitters)
x=model.matrix(Salary~., Hitters)[,-1]
y=Hitters$Salary
lambda=exp(seq(-8.1,1,0.1))

set.seed(1)
train=sample(1:nrow(x),nrow(x)/2)
test=(-train)
y.test=y[test]

set.seed(1)
cv.out=cv.glmnet(x[train,], y[train], alpha=0)
bestlam=cv.out$lambda.min
bestlam

## From R3.6.0 or later
## [1] 326.0828
## Before R 3.6.0
## [1] 211.7416
```

We will now determine what is the test MSE associated with the chosen λ.

```
require(bestglm)
require(glmnet)
library(ISLR)
Hitters=na.omit(Hitters)
x=model.matrix(Salary~., Hitters)[,-1]
y=Hitters$Salary
```

```
lambda=exp(seq(-8.1,1,0.1))

set.seed(1)
train=sample(1:nrow(x),nrow(x)/2)
test=(-train)
y.test=y[test]

set.seed(1)
cv.out=cv.glmnet(x[train,], y[train], alpha=0)
bestlam=cv.out$lambda.min

ridge.mod=glmnet(x[train,], y[train], alpha=0)
ridge.pred=predict(ridge.mod, s=bestlam, newx=x[test,])
mean((ridge.pred-y.test)^2)
```

```
[1] 139863.2
```

Finally, we refit our regression model on the full dataset, using the value of λ chosen by cross-validation and examine the coefficient estimates.

```
require(bestglm)
require(glmnet)
library(ISLR)
Hitters=na.omit(Hitters)
x=model.matrix(Salary~., Hitters)[,-1]
y=Hitters$Salary
lambda=exp(seq(-8.1,1,0.1))

set.seed(1)
train=sample(1:nrow(x),nrow(x)/2)
test=(-train)
y.test=y[test]

set.seed(1)
cv.out=cv.glmnet(x[train,], y[train], alpha=0)
bestlam=cv.out$lambda.min

out=glmnet(x,y,alpha=0)
predict(out, type="coefficients",s=bestlam)[1:20,]
```

```
(Intercept)        AtBat         Hits        HmRun         Runs          RBI
15.44383135   0.07715547   0.85911581   0.60103107   1.06369007   0.87936105
      Walks        Years       CAtBat        CHits       CHmRun        CRuns
 1.62444616   1.35254780   0.01134999   0.05746654   0.40680157   0.11456224
       CRBI       CWalks      LeagueN    DivisionW      PutOuts      Assists
 0.12116504   0.05299202  22.09143189 -79.04032637   0.16619903   0.02941950
     Errors   NewLeagueN
-1.36092945   9.12487767
```

As expected, none of the coefficients is zero. Ridge regression does not perform variable selection.

We will now perform Lasso regression by using the same steps as in the case of ridge regression.

```
require(bestglm)
require(glmnet)
library(ISLR)
Hitters=na.omit(Hitters)
x=model.matrix(Salary~., Hitters)[,-1]
y=Hitters$Salary
lambda=exp(seq(-8.1,1,0.1))

set.seed(1)
train=sample(1:nrow(x),nrow(x)/2)
test=(-train)
y.test=y[test]

set.seed(1)
cv.out=cv.glmnet(x[train,], y[train], alpha=1)
bestlam=cv.out$lambda.min
bestlam

## [1] 9.286955
## [1] 16.78016  before R 3.6.0

lasso.mod=glmnet(x[train,], y[train], alpha=1)
lasso.pred=predict(lasso.mod, s=bestlam, newx=x[test,])
mean((lasso.pred-y.test)^2)

## [1] 143668.8
## [1] 100838.2 before R 3.6.0
```

This is slightly higher from the ridge regression test MSE. However, Lasso regression has an advantage over ridge regression in that the resulting coefficients are sparse.

```
require(bestglm)
require(glmnet)
library(ISLR)
Hitters=na.omit(Hitters)
x=model.matrix(Salary~., Hitters)[,-1]
y=Hitters$Salary
lambda=exp(seq(-8.1,1,0.1))

set.seed(1)
train=sample(1:nrow(x),nrow(x)/2)
test=(-train)
y.test=y[test]

set.seed(1)
cv.out=cv.glmnet(x[train,], y[train], alpha=1)
bestlam=cv.out$lambda.min

out=glmnet(x,y,alpha=1)
predict(out, type="coefficients",s=bestlam)[1:20,]
```

```
(Intercept)         AtBat          Hits         HmRun          Runs
  -3.04787648    0.00000000    2.02551572    0.00000000    0.00000000
```

```
          RBI         Walks         Years        CAtBat         CHits
   0.00000000    2.26853781    0.00000000    0.00000000    0.00000000
       CHmRun         CRuns          CRBI        CWalks       LeagueN
   0.01647106    0.21177390    0.41944632    0.00000000   20.48456543
    DivisionW       PutOuts       Assists        Errors    NewLeagueN
-116.59062078    0.23718459    0.00000000   -0.94739923    0.00000000
```

Here we can see that 10 out of 19 coefficient are exactly zero (We ignore the intercept as it is not included in the penalty term). So the lasso model with λ chosen by cross-validation contains only 9 variables.

Additional Activity

Question *Submitted Feb 7th 2024 at 4:11:40 pm*

Select all that applies. The difference between ridge regression and Lasso lies in:

- [x] the form of the penalty term in the optimisation problem;
- [x] the lasso regression shrinks parameters to zero whereas ridge regression does not;
- [ ] the lasso regression cannot be used for variable selection while the ridge regression can.