One last time through the hot New Jersey night…

Claire Jellison

10/7/2019

R Markdown

```
install.packages("glmnet")
```

```
## Installing package into '/home/clajelli/R/x86_64-pc-linux-gnu-library/3.5'
## (as 'lib' is unspecified)
```

```
d <- read.csv("http://andrewpbray.github.io/data/crime-train.csv")
library(ggplot2)
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-18
```

```
library(leaps)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
x=model.matrix(ViolentCrimesPerPop ~.,d)[,-1]
y=d$ViolentCrimesPerPop
```

The Ridge Regression

```
grid=10^seq(10,-2,length=100)
ridge.mod=glmnet(x,y,alpha=0,lambda=grid)
dim(coef(ridge.mod))
```

```
## [1] 2144  100
```

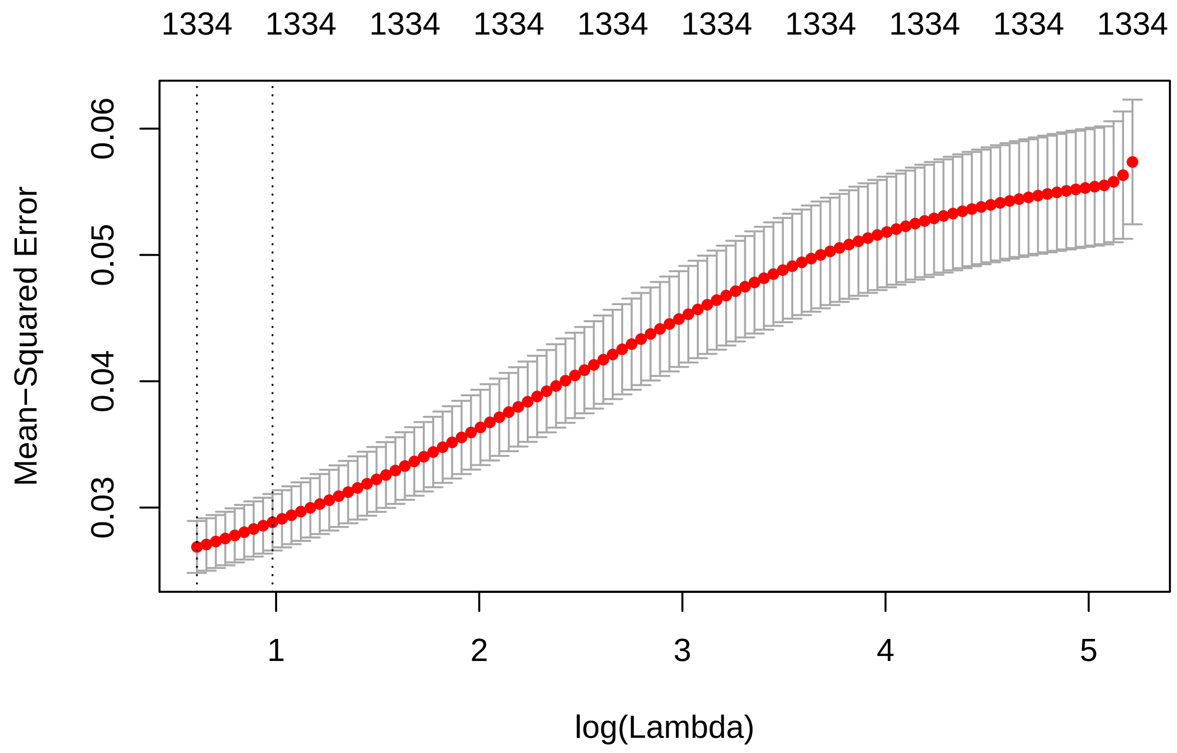
```
predict(ridge.mod,s=50,type="coefficients")[1:10,]
```

```
##   (Intercept)         state       county1     county105      county11
##  2.390485e-01 -1.283392e-05 -2.183260e-04 -8.442478e-04 -8.164369e-04
##     county113     county119     county123     county129      county13
## -7.205606e-04 -9.652844e-05 -7.763388e-04 -9.153132e-04  3.961756e-04
```

```
set.seed (1)
train=sample(1:nrow(x), nrow(x)/2) #splitting the data into test and training
test=(-train)
y.test=y[test]
y.train = y[train]
ridge.mod=glmnet(x[train,],y[train],alpha=0,lambda=grid, thresh =1e-12)
ridge.predtrain=predict(ridge.mod,s=4,newx=x[train,])
ridge.predtest=predict(ridge.mod,s=4,newx=x[test,])
mean((ridge.predtrain-y.train)^2) # looking at the mse
```

```
## [1] 0.0222726
```

```
set.seed (1)
cv.out=cv.glmnet(x[train ,],y[train],alpha=0)
plot(cv.out)
```



```
bestlamridge=cv.out$lambda.min
bestlamridge
```

```
## [1] 1.841108
```

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

```
## [1] 0.02371262
```

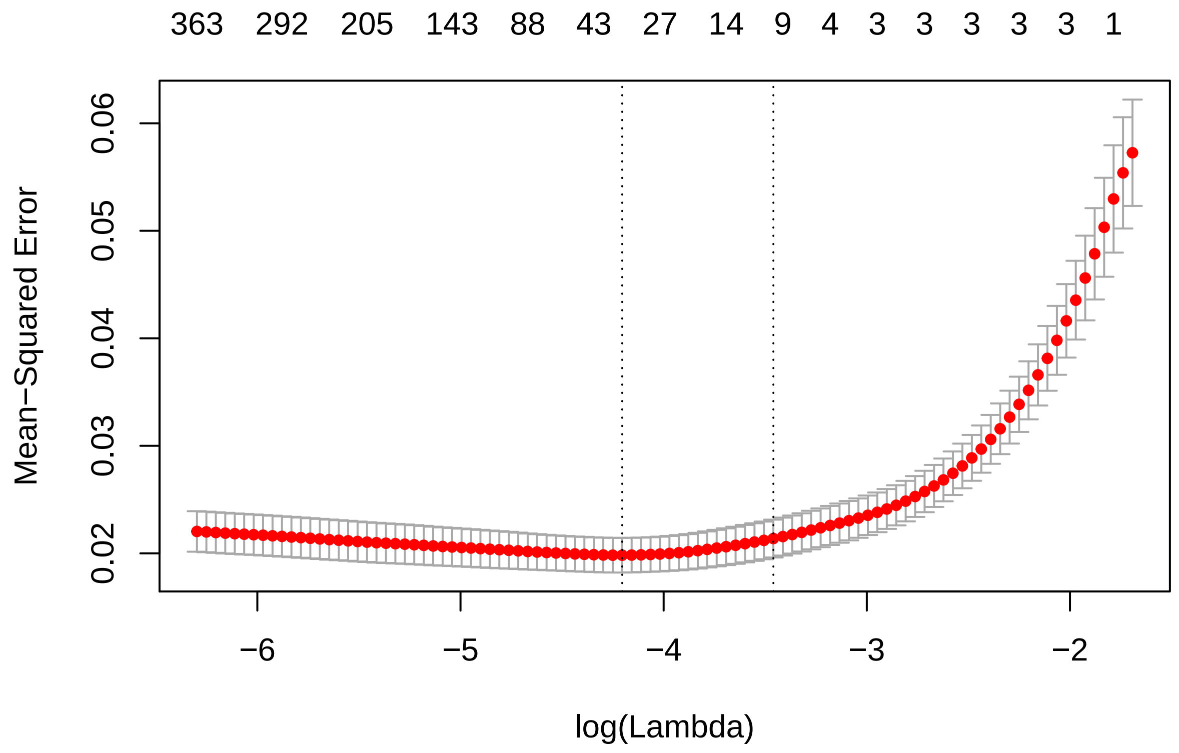
The Lasso Regression

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

```
## [1] 2144  100
```

```
#plot(lasso.mod)
```

```
set.seed (1)
cv.out.lasso=cv.glmnet(x[train ,],y[train],alpha=1)
plot(cv.out.lasso)
```



```
bestlamlasso <- cv.out.lasso$lambda.min
lasso.pred=predict(lasso.mod,s=bestlamlasso ,newx=x[test,])
mean((lasso.pred-y.test)^2)
```

```
## [1] 0.02051799
```

```
out=glmnet(x,y,alpha=1,lambda=grid)
lasso.coef=predict(out,type="coefficients",s=bestlamlasso)
lasso.coef[lasso.coef !=0]
```

```
## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient

##  [1]  3.504442e-01 -6.180800e-04  1.211083e-02  7.466622e-02  6.968229e-02
##  [6]  5.341522e-08  1.061181e-01  6.565264e-02  8.800262e-02 -1.036881e-01
## [11]  4.104646e-02  7.249591e-04  2.447381e-04  1.240517e-01  1.034673e-01
## [16]  2.194634e-02  2.230008e-02  9.585600e-02  3.368520e-01  6.037136e-02
## [21]  2.666842e-04 -1.508495e-01  8.698143e-03  1.355150e-01 -2.278340e-01
## [26] -9.990305e-03  1.654399e-03  2.588708e-01  7.779776e-02  4.433791e-02
## [31]  4.884449e-02  9.935481e-02  3.312907e-03  2.291077e-02  4.742481e-04
## [36]  4.826155e-06  6.040245e-09  6.103411e-02  4.068585e-02  7.614600e-08
## [41]  1.348410e-02  6.374494e-08  7.023654e-03  3.813223e-03  2.065502e-02
```

```
y.train = y[train]
```

```
lasso_best <- glmnet(x[train,], y[train], alpha = 1, lambda = bestlamlasso)
predtrain <- predict(lasso_best, s = bestlamlasso, newx = x[train,])
```

```r
pred <- predict(lasso_best, s = bestlamlasso, newx = x[test,])
final <- cbind(y[test], pred)
head(final)
```

```
##                  1
## 1 0.15 0.1529721
## 2 0.26 0.4717808
## 3 0.28 0.1873142
## 4 0.34 0.5517771
## 5 0.60 0.2823333
## 6 0.83 0.3015998
```

```r
mean((predtrain-y.train)^2)
```

```
## [1] 0.01388612
```

Exercise 1

How many variables were selected by the LASSO?

The lasso selected 45 variables.

Exercise 2

What are the training MSEs for ridge and LASSO using the optimal value of λ ?

The MSE for the ridge was 0.0222726 and the MSE for the lasso was 0.01388612.

Exercise 3

If the MSEs differed, why do you think one is higher than the other in this setting?

This means that the MSE was lower for the lasso than for the ridge. This does not seem that surprising since the ridge keeps all the predictors in the model and thus might result in a higher MSE than the lasso which has fewer predictors, especially since it seems like in this scenario there are a lot of predictors relative to the number of observations.

Chapter 6 Exercises

2 a

The lasso, relative to least squares, is:

i. More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

False, the lasso is less flexible than least squares.

ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

False, the lasso is less flexible.

iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

True, the lasso is less flexible because it applies a penalty, and it results in greater bias and less variance. However, the tradeoff can be worth it especially with data that has high variance and it can actually improve MSE.

iv. Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

False, the lasso decreases the variance and increases the bias.

2 b

The ridge, relative to least squares, is:

i. More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

False, less flexible especially as penalty increases.

ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

False, less flexible.

iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

This is true, the penalty imposed makes the model less flexible and results in less variance but can be more biased. It will have better prediction accuracy when the increase in bias is smaller than the decrease in variance because the MSE will likely be improved and it will preform better on test data.

iv. Less flexible and hence will give improved prediction accu- racy when its increase in variance is less than its decrease in bias.

False, the tradoff is opposite.

3

i. Increase initially, and then eventually start decreasing in an inverted U shape.
ii. Decrease initially, and then eventually start increasing in a U shape.
iii. Steadily increase.
iv. Steadily decrease.
v. Remain constant.

(a) As we increase s from 0, the training RSS will:

iii. As s increases the coefficients do not have to be as shrunken and thus the training RSS can be reduced further with the increased flexibility.

(b) Repeat (a) for test RSS.

ii. Whe s is low the model is super inflexible so increasing s some will lower the training and test RSS however increasing it by a lot could make it overfit the training data resulting in a higher RSS for the test data.

(c) Repeat (a) for variance.

iii. A low s means that the coefficients are pretty restricted so the variance is pretty small, as s increases the variance will also increase.

(d) Repeat (a) for (squared) bias.

iv. As s increases the model will become less biased since it is more flexible and the coefficients can be larger.

(e) Repeat (a) for the irreducible error.

v. The irreducible error is inevitable and the model cannot acount for it.

4

i. Increase initially, and then eventually start decreasing in an inverted U shape.

ii. Decrease initially, and then eventually start increasing in a U shape.

iii. Steadily increase.

iv. Steadily decrease.

v. Remain constant.

(a) As we increase lambda from 0, the training RSS will:

iii. As the model becomes less flexible and the coefficients are penalized more heavily the training RSS will go up.

(b) Repeat (a) for test RSS.

ii. This depends somewhat on the bias variance tradoff. When the penalty is 0 it is basically just a linear model. As it becomes less flexible, the bias increases but variance decreases so depending on the slopes the test RSS might decrease then increase in a U shape.

(c) Repeat (a) for variance.

iv.As lambda increases this leads to decreased variance.

(d) Repeat (a) for (squared) bias.

iii. As lambda increases this leads to increased bias since the coefficients are shrunken.

(e) Repeat (a) for the irreducible error.

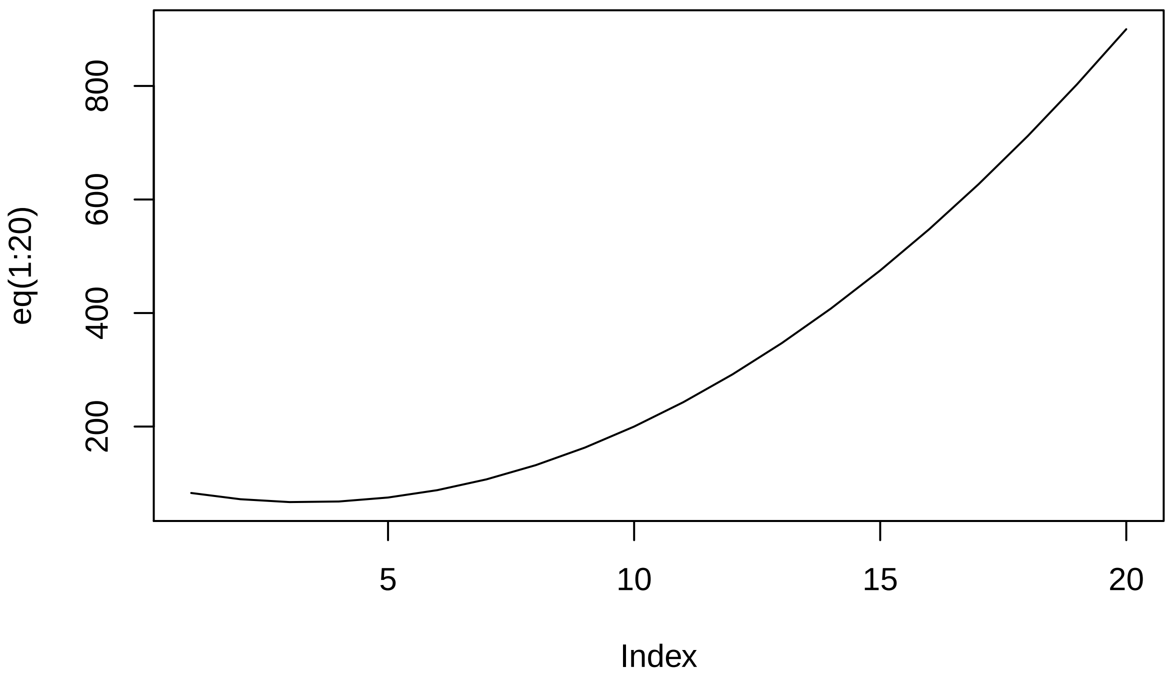
v. The irreducible error is inevitable and not something the model can capture.

6

We will now explore (6.12) and (6.13) further.

(a) Consider (6.12) with p = 1. For some choice of y1 and lambda > 0, plot (6.12) as a function of beta1. Your plot should confirm that (6.12) is solved by (6.14).

```r
y1 = 10
lam = 2
eq = function(b){(y1 - b)^2 + lam*(b)^2}
plot(eq(1:20), type='l')
```



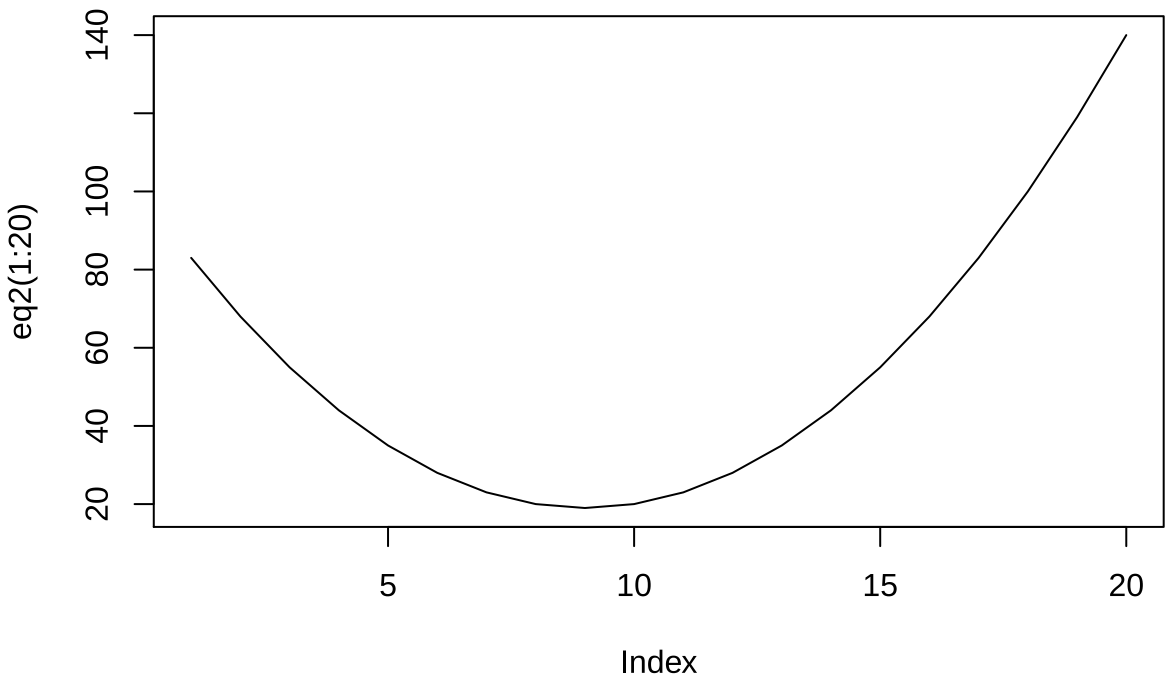
```r
bmin = 10/(1 + 2)
bmin
```

```
## [1] 3.333333
```

Checking that it the graph is solved by the other equation looks about right.

(b) Consider (6.13) with p = 1. For some choice of y1 and lambda > 0, plot (6.13) as a function of beta1. Your plot should confirm that (6.13) is solved by (6.15).

```r
eq2 = function(b){(y1 - b)^2 + lam*abs(b)}
plot(eq2(1:20), type='l')
```



```r
bmin2 = 10 - lam/2
bmin2
```

```
## [1] 9
```

Checking that it the graph is solved by the other equation looks about right.