Analytics 590: Homework #1

Hanjing Wang

```
set.seed(1)

Hitters = read.csv('Hitters.csv')
```

1

(1)

```
library("glmnet")
```

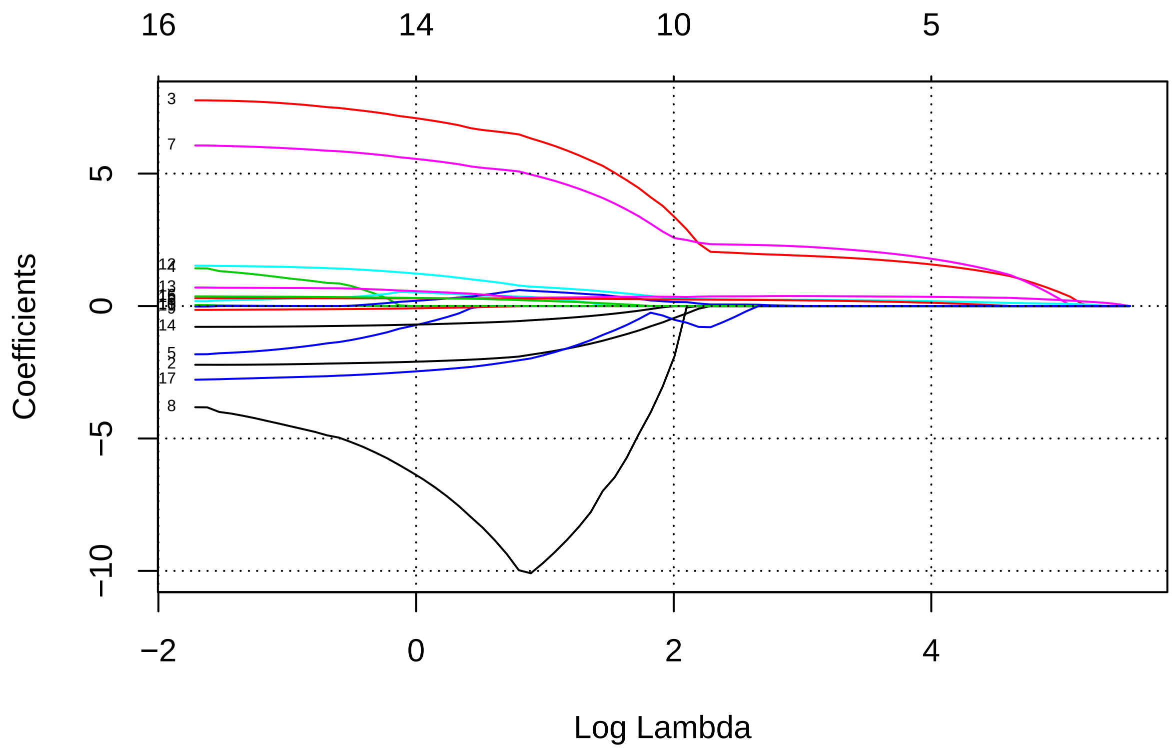
```
## Warning: package 'glmnet' was built under R version 3.4.4
## Loading required package: Matrix
## Loading required package: foreach
## Warning: package 'foreach' was built under R version 3.4.4
## Loaded glmnet 2.0-13
```

```
Hitters = na.omit(Hitters)
```

We use LASSO regression to predict Salary from the other numeric predictors.

```
matrix <- model.matrix(lm(Salary ~ .-X-League-Division-NewLeague, data =Hitters))
fit.lasso = glmnet(matrix, Hitters$Salary, alpha = 1)

# plot the coefficient trajectories as a function of the regularization parameter lambda
plot(fit.lasso, xvar = "lambda", lwd = 1,label=TRUE)
grid(col = 1)
```




```r
# we could also show the coefficients when there are only three variables left
fit.lasso$beta[,5]
```

```
## (Intercept)       AtBat        Hits       HmRun        Runs         RBI
##  0.00000000  0.00000000  0.08064999  0.00000000  0.00000000  0.00000000
##       Walks       Years      CAtBat       CHits      CHmRun       CRuns
##  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000  0.06719193
##        CRBI      CWalks     PutOuts     Assists      Errors
##  0.17823025  0.00000000  0.00000000  0.00000000  0.00000000
```

From the plot, we could see that the final three predictors that remain in the model are Hits, CRuns, CRBI.

Then, We use cross-validation to find the optimal value of the regularization penality.

```r
cv.lasso <- cv.glmnet(matrix, Hitters$Salary, alpha = 1)
bestlambda<- cv.lasso$lambda.min
bestlambda
```

```
## [1] 2.935124
```

Finally, we use the bestlambda to fit the model.

```r
fit.lasso1 = glmnet(matrix, Hitters$Salary, alpha = 1, lambda = bestlambda)
fit.lasso1$beta
```

```
## 17 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept)   .
## AtBat       -1.7099436
## Hits         6.0823207
```

```
## HmRun         .
## Runs          .
## RBI            0.2602043
## Walks          4.7542167
## Years         -9.0357077
## CAtBat         .
## CHits          .
## CHmRun         0.5745583
## CRuns          0.6973902
## CRBI           0.3033389
## CWalks        -0.4924500
## PutOuts        0.2804673
## Assists        0.1839622
## Errors        -1.7085130
```

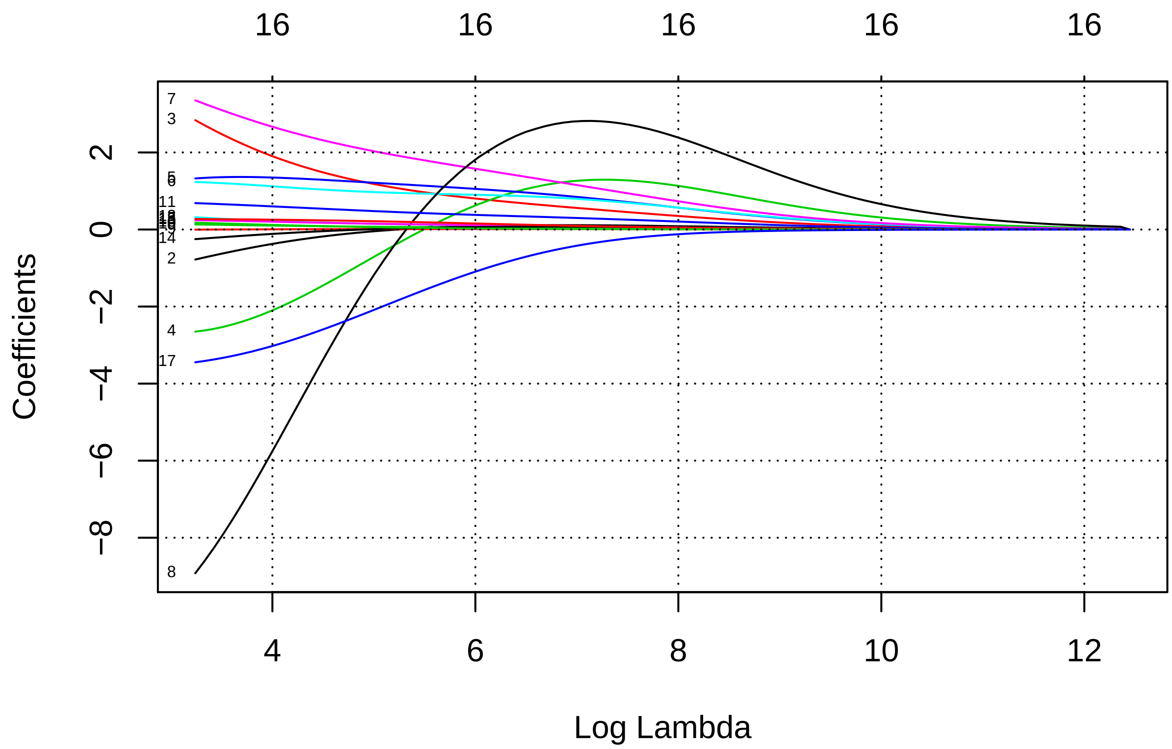
We could see that 12 predictors are left in that model.

(2)

Repeat with Ridge Regression.

```
matrix1 <- model.matrix(lm(Salary ~ .-X-League-Division-NewLeague, data =Hitters))
fit.ridge = glmnet(matrix1, Hitters$Salary, alpha = 0)

# plot the coefficient trajectories as a function of the regularization parameter lambda
plot(fit.ridge, xvar = "lambda", lwd = 1,label=TRUE)
grid(col = 1)
```




```r
cv.ridge <- cv.glmnet(matrix1, Hitters$Salary, alpha = 0)
bestlambda<- cv.ridge$lambda.min
bestlambda
```

```
## [1] 28.01718
```

```r
# the coefficients
fit.ridge1 = glmnet(matrix, Hitters$Salary, alpha = 0, lambda = bestlambda)
fit.ridge1$beta
```

```
## 17 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept)   .
## AtBat       -0.723504487
## Hits         2.704625078
## HmRun       -2.595387303
## Runs         1.351260127
## RBI          1.226757322
## Walks        3.254158746
## Years       -8.485767019
## CAtBat      -0.001692903
## CHits        0.125773172
## CHmRun       0.661490023
## CRuns        0.297444435
## CRBI         0.242310378
## CWalks      -0.228174794
## PutOuts      0.270173394
```

```
## Assists       0.169600493
## Errors       -3.413387246
```

2

The bias-variance trade-off is that in a series of models, the models which have a lower bias for the parameter estimation will have a higher variance, and vice visa. The regularization will shrink the coefficient estimates towards zero. Shrinking the coefficient estimate will lead to a substantial reduction in the variance of the predictions, at the expense of a slight increase in bias so that the overal prediction accruracy will be improved. For example, in 1.1, the lasso regression model using the best lambda generating from the cross validation will have a sightly higher bias than the model using all the predictors but also will have a much lower variance. Compared the model where there are only three predictors left, the model with the best lambda will have a lower bias but higher variance.