HW0

Sooeun Oh

September 16, 2018

Problem 1.1

Use LASSO regression to predict Salary from the other numeric predictors (you should omit the categorical predictors). Create a visualization of the coefficient trajectories. Comment on which are the final three predictors that remain in the model. Use cross-validation to find the optimal value of the regularization penalty. How many predictors are left in that model?

Load *ISLR* library for *Hitters* dataset. Also, load corresponding libraries to fit the ridge and lasso regression models, and to plot the coeffcient trajectories.

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.4.1
```

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.4.1
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 3.4.1
```

```
## Loaded glmnet 2.0-10
```

```
library(plotmo)
```

```
## Warning: package 'plotmo' was built under R version 3.4.4
```

```
## Loading required package: plotrix
```

```
## Warning: package 'plotrix' was built under R version 3.4.1
```

```
## Loading required package: TeachingDemos
```

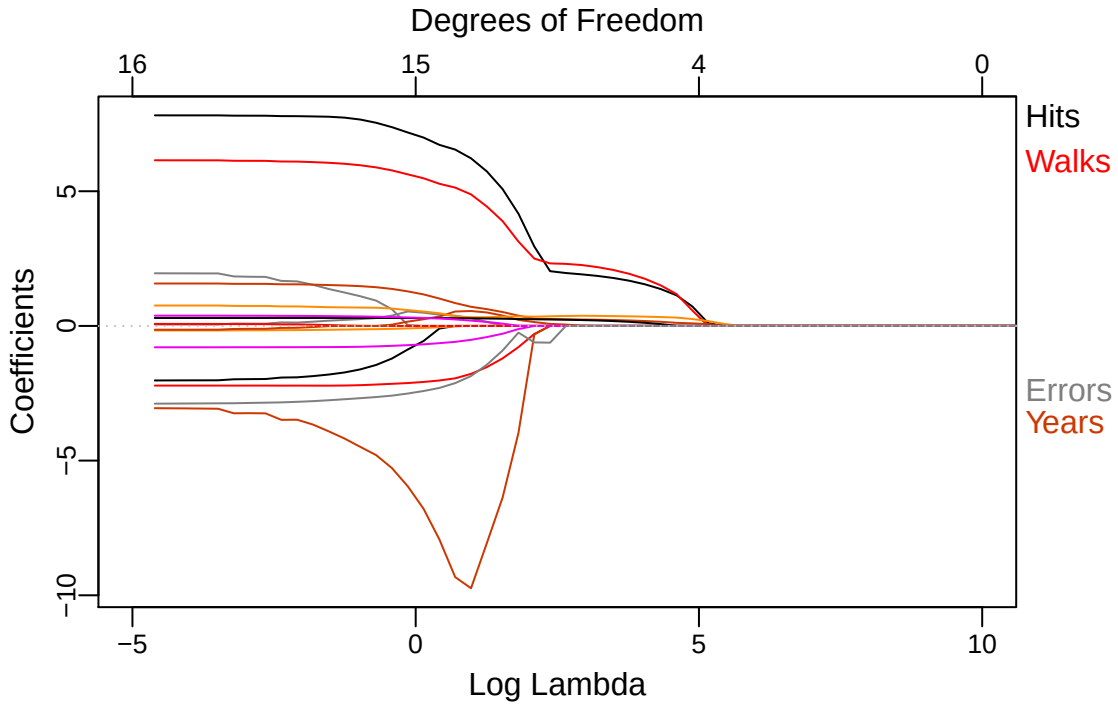
```
## Warning: package 'TeachingDemos' was built under R version 3.4.4
```

First, we remove categorical variables such as *LeagueN*, *NewLeagueN*, and *DivisionW* so that we have only 16 predictors left. Then apply **na.omit()** to get rid of any rows with missing Salary values (NaN).

```
Hitters = Hitters[,unlist(lapply(Hitters,is.numeric))]
Hitters = na.omit(Hitters)
x = model.matrix(Salary~.,Hitters)[,-1]
y = Hitters$Salary
```

Fit the lasso regression model with **alpha=1** and *Salary* as the target variable.

```
grid = 10^seq(10,-2,length=100)
lasso.mod = glmnet(x,y,alpha=1,lambda=grid)
plot_glmnet(lasso.mod,xvar="lambda",xlim=c(-5,10),label=4)
```



As we can see from the above plot of coefficient trajectories, the final three predictors in the model are *Hits*, *Walks*, and *Years*. Now, apply cross-validation to find the optimal value of the regularization penalty, λ. Set a random seed for the reproducibility:

```
set.seed(123)
train = sample(1:nrow(x),nrow(x)/2)
cv.out = cv.glmnet(x[train,],y[train],alpha=1)
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 23.03503
```

```
lasso.coef = predict(lasso.mod,type="coefficients",s=bestlam)[1:17,]
lasso.coef[lasso.coef!=0]
```

```
##   (Intercept)          Hits         Walks        CHmRun         CRuns
## -23.947419179   1.872608393   2.204889919   0.002744444   0.232569501
##          CRBI       PutOuts
##   0.373043286   0.205842880
```

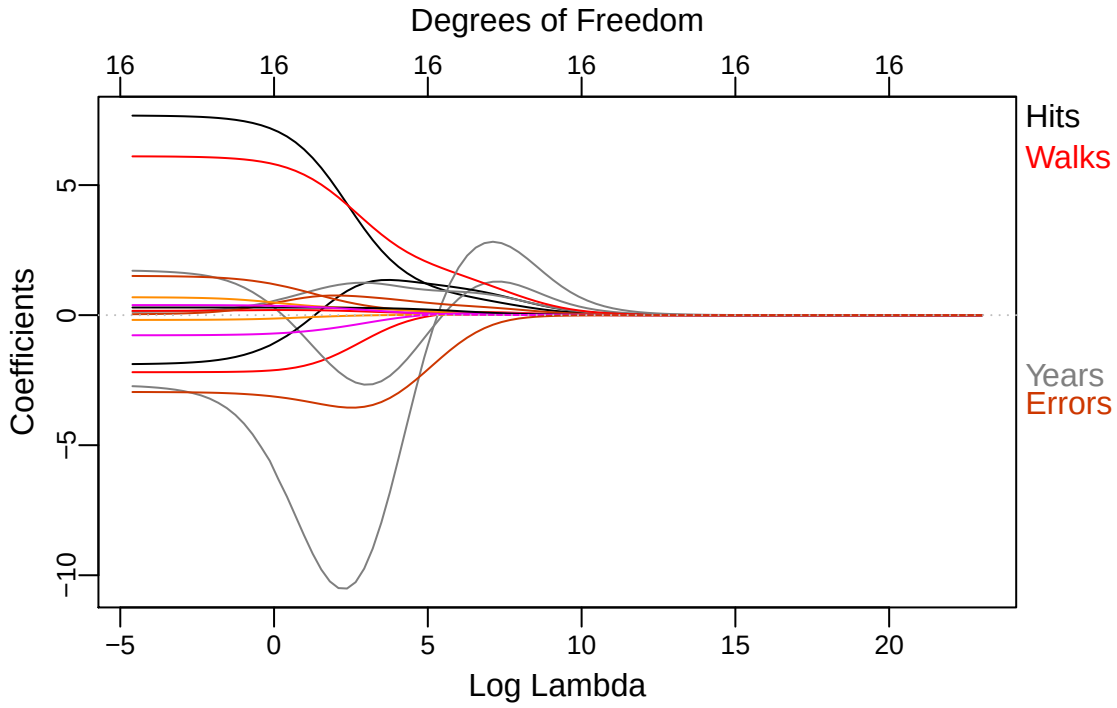
Thus, the optimal value of λ is about 23.035 and the total of 6 most important predictors are left in that model.

Problem 1.2

Repeat with Ridge Regression. Visualize coeffecient trajectories. Use cross-validation to find the optimal value of the regularization penalty.

Repeat the process from the previous problem, this time with **alpha=0**.

```
ridge.mod = glmnet(x,y,alpha=0,lambda=grid)
plot_glmnet(ridge.mod,xvar="lambda",label=4)
```



From the above plot of coefficient trajectories, we see that the final three predictors in the model are *Hits*, *Walks*, and *Errors*, instead of *Years*.

```
set.seed(123)
train = sample(1:nrow(x),nrow(x)/2)
cv.out = cv.glmnet(x[train,],y[train],alpha=0)
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 114.6459
```

```
ridge.coef = predict(ridge.mod,type="coefficients",s=bestlam)[1:17,]
ridge.coef[ridge.coef!=0]
```

```
##  (Intercept)        AtBat         Hits        HmRun         Runs
## -23.05138800  -0.11175651   1.33585701  -1.10377288   1.25124887
##          RBI        Walks        Years       CAtBat        CHits
##   1.00079306   2.17409491  -2.29985425   0.00921885   0.07799774
##       CHmRun        CRuns         CRBI       CWalks      PutOuts
##   0.50542229   0.16411290   0.16514255  -0.01591957   0.22777164
##      Assists       Errors
##   0.07438258  -2.36298628
```

In this case, the optimal value of λ is about 114.646 and the total of 16 predictors (i.e. all the variables of the

dataset) are left in that model as expected.

Problem 2

Explain in your own words the bias-variance tradeoff. What role does regularization play in this tradeoff? Make reference to your findings in number (1) to describe models of high/low bias and variance.

The bias-variance tradeoff is the typical relationship between bias and variance of any model and it's called "tradeoff" because it is difficult to obtain a method that yields both low variance and low squared bias at the same time. Instead, it tends to have high variance when the bias is low and high bias when the variance is low.

Regularization plays a significant role in this tradeoff – it helps to choose an appropriate midpoint of the bias-variance relationship without overfitting or underfitting the dataset by imposing the penalty with the shrinkage parameter λ. For example, we saw that none of the coefficients reached zero and included all predictors in the model when ridge regression was used. This is because ridge regression does not perform variable selection. In other words, the model became more flexible and thus, has higher variance and lower bias. On the other hand, lasso yielded sparse model with half number of predictors left (i.e. less variable, but more biased).