Hw0-590-dong

xiaoman dong

9/18/2018

Question 1.1

1.1.1. Create a visualization of the coefficient trajectories

```
set.seed(123)
library(glmnet)
```

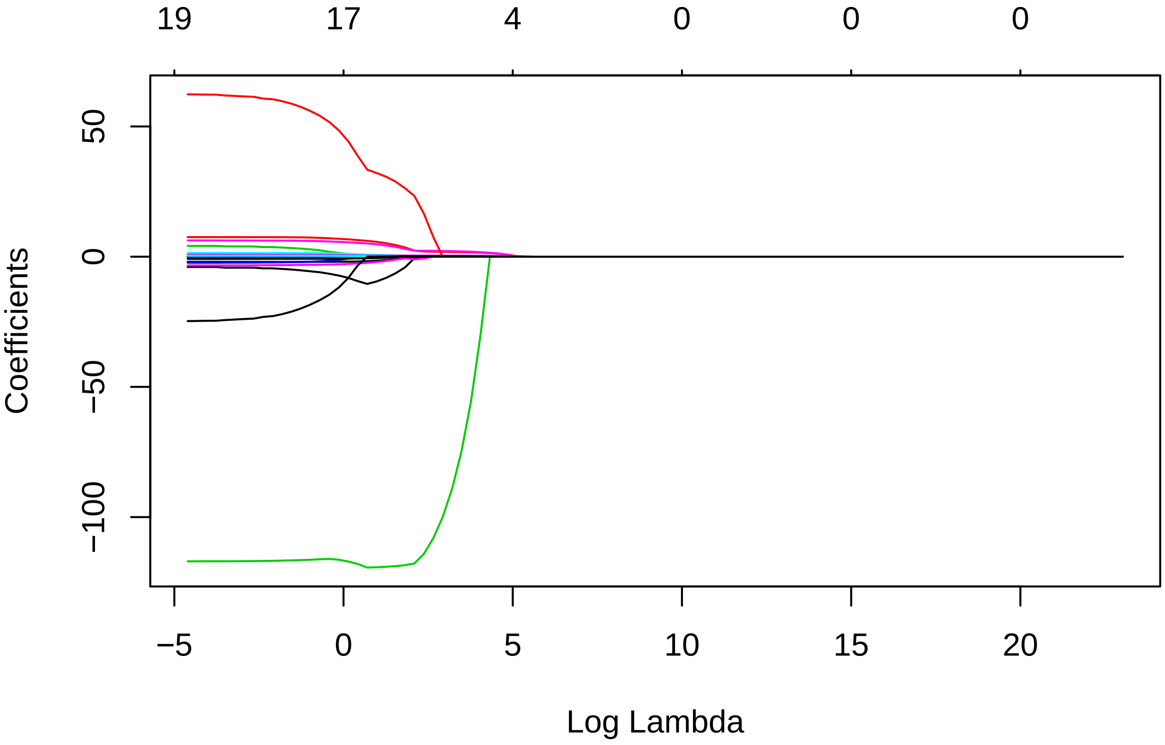
```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-13
```

```
library(ISLR)
#Omit the missing value in salary
Hitters=na.omit(Hitters)
#Transform categorical data into dummy variables
x=model.matrix(Salary~.,Hitters)[,-1]
y=Hitters$Salary

grid=10^seq(10,-2,length=100)
lasso.fit=glmnet(x,y,alpha=1,lambda=grid)
plot(lasso.fit,xvar='lambda')
```



Fro, plot above we see that when log(lambda)=-5, all 19 variables are in the model, however when log(lambda)=5, only 4 variables are retained.

1.1.2. Comment on which are the final three predictors that remain in the model

```
set.seed(123)
cv.lasso=cv.glmnet(x,y)
coef(cv.lasso)
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept) 193.74263858
## AtBat         .
## Hits          1.21471320
## HmRun         .
## Runs          .
## RBI           .
## Walks         1.28957902
## Years         .
## CAtBat        .
## CHits         .
## CHmRun        .
## CRuns         0.12923755
## CRBI          0.31515925
## CWalks        .
## LeagueN       .
## DivisionW     .
## PutOuts       0.02533305
## Assists       .
## Errors        .
## NewLeagueN    .
```

```
#sum(abs(coef(cv.lasso)))
```

From the result above, we see that three predictors still remain are hits, Walks, and CRBI. Even though coefficient of CRuns is not 0, but is quite close to 0, and we can omit the predictor.

1.1.3. Use cross-validation to find the optimal value of the regularization penalty

```
set.seed(123)
#split with train and test data
train=sample(1:nrow(x),nrow(x)/2)
test=(-train)
y.test=y[test]



#Apply cv lasso regression to Hitters
Hitters_lasso=cv.glmnet(x[train,],y[train],alpha=1)
#plot results
#plot(Hitters_lasso)
min(Hitters_lasso$cvm)
```

```
## [1] 145385.3
```

```
#lambda for this min mse
Hitters_lasso$lambda.min
```

```
## [1] 25.28092
```

```
#test with cv
cv.out=cv.glmnet(x[train,],y[train], alpha=1)
lasso.pred=predict(lasso.fit,s=cv.out$lambda.min,newx=x[test,])
r=mean((lasso.pred-y.test)^2)
print(paste('The optimal regularization of lasso regression is', round(Hitters_lasso$lambda.min)))
```

```
## [1] "The optimal regularization of lasso regression is 25"
```

```
print(paste('The MSE of lasso regression is', round(r)))
```

```
## [1] "The MSE of lasso regression is 82522"
```

1.1.4. How many predictors are left in that model? 4 predictors, hits, walks, cruns, and crbi.

——————————————Question 1.2 Repeat with Ridge Regression—————————————————-

1.2.1 Visualize the coefficient trajectories

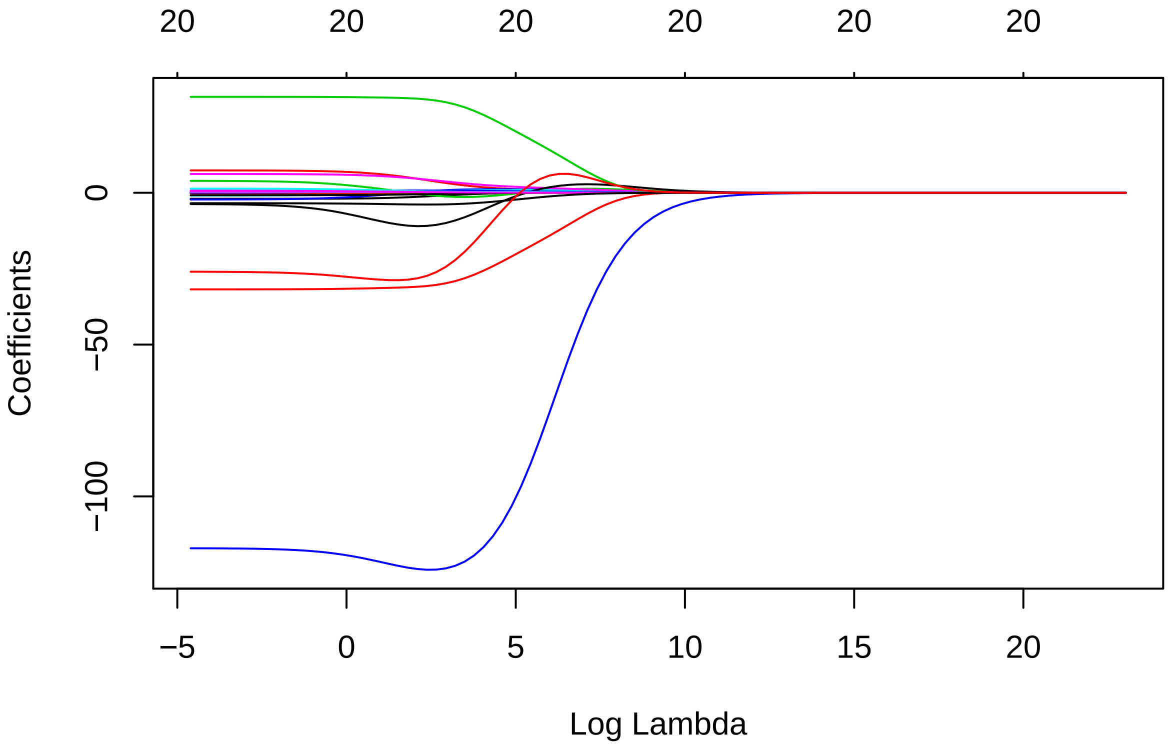
```
set.seed(123)
library(glmnet)
library(ISLR)
#Omit the missing value in salary
Hitters=na.omit(Hitters)
with(Hitters,sum(is.na(Salary)))
```

```
## [1] 0
```

```
#Transform categorical data into dummy variables
x=model.matrix(Salary~.-1,data=Hitters)
y=Hitters$Salary

#Set with ridge regression
grid=10^seq(10,-2,length=100)
ridge.fit=glmnet(x,y,alpha=0,lambda=grid)

#Visualization coefficient trajectory
plot(ridge.fit,xvar='lambda')
```

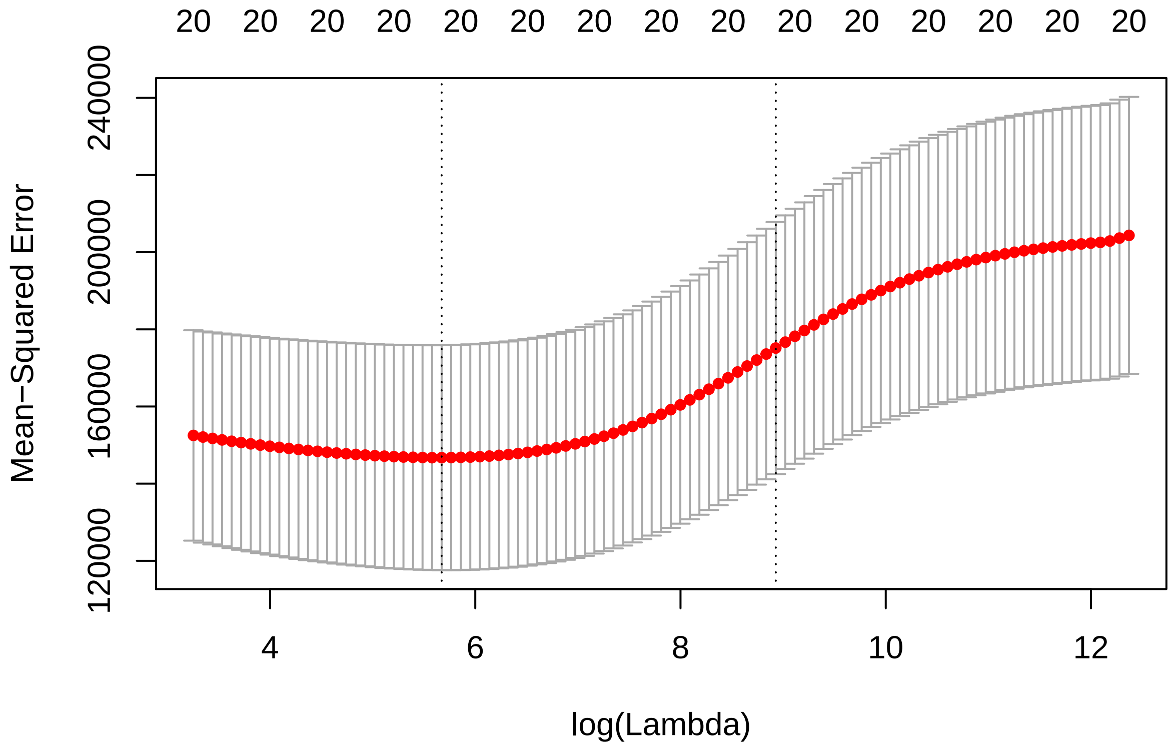


1.2.2

Use cross-validation to find the optimal value of the regularization penalty

```
set.seed(123)
#split with train and test data
train=sample(1:nrow(x),nrow(x)/2)
test=(-train)
y.test=y[test]

#use cross-validation to find parameter lambda
cv.out=cv.glmnet(x[train,], y[train], alpha=0)
plot(cv.out)
```



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

```
## [1] 290.6692
```

```
ridge.mod=glmnet(x[train,],y[train],alpha=0,lambda=grid, thresh=1e-12)
ridge.pred=predict(ridge.mod,s=bestlambda, newx=x[test,])
re=mean((ridge.pred - y.test)^2)
re
```

```
## [1] 105456.7
```

```
print(paste('The optimal regularization of ridge regression is', round(bestlambda)))
```

```
## [1] "The optimal regularization of ridge regression is 291"
```

```
print(paste('The MSE of ridge regressin is', round(re)))
```

```
## [1] "The MSE of ridge regressin is 105457"
```

Question2.1

Explain in your own words the bias-variance tradeoff.

A larger bias variance is preferred when data are noisy to achieve a better control of variance, whereas bias can be decreased as more data become available, therefore the variance decreases.

Question2.2

What role does regularization play in this tradeoff? In general, regularization refers to avoid of overfitting. It provides limitation on least square error. In terms of the lasso regression, it helps to select number of variables that retained in the model. Also, it helps to balance the bias and variance tradeoff.

Question2.3

Make reference to your findings in number (1) to describe models of high/low bias and variance.

From the result in 1.1.3 and 1.2.2, we see that the mse of both regression model are pretty similar by cross-validation. However, lasso performs better over ridge regression that the coefficient estimator is more sparse. It helps to reduce the number of predictors with non coefficient to nearly 3.

When comparing with ridge regression, as least squares has large variance, lasso would decrease the variance to get a more succinct result. It has the variable selections,therefore the model of lasso regression is easier to explain than ridge regression. In general, ridge regression trade off bias for variance, whereas lasso trade off variance for bias