HWO Lufei Wang

Load Data

hitters = read.csv("/Users/lufeiwang/Desktop/Hitters.csv")

1.1

# Get rid of Categorical  
hitters = hitters[,!names(hitters) %in% c('League', 'Division', 'X', 'NewLeague')]  
hitters = hitters[!is.na(hitters$Salary),]  
x = model.matrix (Salary ~.,hitters )[,-1]  
y = hitters$Salary

library(glmnet)

## Warning: package 'glmnet' was built under R version 3.4.2

## Loading required package: Matrix

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 3.4.3

## Loaded glmnet 2.0-13

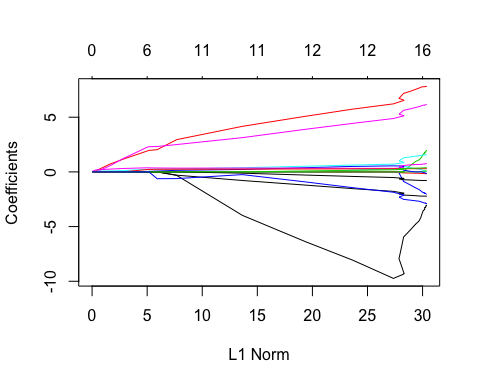
library(foreach)  
  
# Get a list of different lambdas  
grid =10^ seq(10,-2, length =100)  
# use cv to get best lambda  
cv = cv.glmnet(x, y, alpha = 1)  
bestlam = cv$lambda.min  
bestlam

## [1] 15.66387

# build model  
lasso = glmnet(x, y, alpha = 1, lambda = grid)  
coefs = predict(lasso, type = "coefficients", s = bestlam)[1:17,]  
coefs

## (Intercept) AtBat Hits HmRun Runs   
## -46.95779760 0.00000000 1.94754410 0.00000000 0.00000000   
## RBI Walks Years CAtBat CHits   
## 0.00000000 2.28950688 0.00000000 0.00000000 0.00000000   
## CHmRun CRuns CRBI CWalks PutOuts   
## 0.03699302 0.24336137 0.37168299 0.00000000 0.22521285   
## Assists Errors   
## 0.00000000 0.00000000

# plot the trajectory  
lasso.plot = glmnet(x, y, alpha = 1, lambda = grid)  
plot(lasso.plot)



# The final three predictors left in the model are Years, Wlaks, and Hits  
  
# In the optimal model, there are 14 predictors left

1.2

Ridge Regression

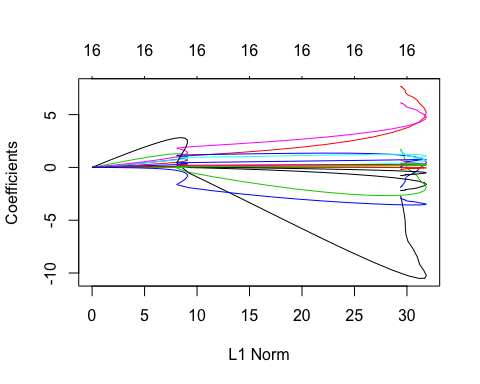
set.seed(1)  
ridge = glmnet(x, y, lambda = grid, alpha = 0)  
  
# Use cv to find best lambda  
cv = cv.glmnet(x, y, alpha = 0)  
bestlam =cv$lambda.min  
bestlam

## [1] 345.4089

out = glmnet(x, y, alpha = 0)  
  
# optimal coefficients  
coefs = predict(out ,type="coefficients",s=bestlam )[1:17 ,]  
coefs

## (Intercept) AtBat Hits HmRun Runs   
## -11.12084142 0.07553021 0.84946719 0.45045601 1.08086922   
## RBI Walks Years CAtBat CHits   
## 0.91139955 1.64361719 1.48526972 0.01126421 0.05664590   
## CHmRun CRuns CRBI CWalks PutOuts   
## 0.39266195 0.11453779 0.11828758 0.05930088 0.16582517   
## Assists Errors   
## 0.03079812 -1.23276330

# Plot the coefficients trajectory  
plot(ridge)



Question 2:

Regularization penalize linear model for being to complex, thus reduce the possibility of overfitting of the model - it reduces variance and improve bias.

In ridge regression in the last question, we can see that some coefficients have been reduced to values that are close to zero, which reduced the importance of those variables so thet have less effect on the output, so the model it's getting simpler, because there are less variables that have large effects on the regression output. For lasso regression, it is ultimately a variable selection process. From the lasso result in the last question, some of the coefficients are zero, so the variables are not in the model anymore, so the model is getting less complex. It can reduce variance and improve bias.

Linear models and decision tree have high bias and low variance

models like high-degree polynomial regression, neural networks, and ensemble methods.