STATS415hw6

Name: Yunguo Cai Uniqname: cyunguo Lab: 003

3/2/2018

1. Best option: (3)

As we increase s from 0, the number of variables included in the model will steadily increase because more are incorporated in the model.

1. Best option: (4)

As we increase s from 0, the training RSS will steadily decreases because the model becomes more flexible with the increasing s and thus is constrained and will be more and more close to the least squares estimate.

1. Best option: (2)

As we increase s from 0, the test RSS will deacrease intially, and then eventually start increasing because is firstly constrained close to 0 for overfitting, resulting in decrease and coefficients are then removed from the model with the increasing of s, resulting in increase.

1. Best option: (3)

As we increase s from 0, the variance of will steadily increase because more are incorporated in the model, which increases the variance.

1. Best option: (4)

As we increase s from 0, the squared bias of will steadily decrease because the model is highly biased when s = 0 and then the bias is decreased. The coefficients will increase to their least squares estimates and the model is becoming more and more flexible which provokes a steady decrease in bias.

2.(a)

library(ISLR)  
library(glmnet)

## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-13

library(boot)  
library(leaps)  
library(SignifReg)  
set.seed(23456)  
data("College")

# Randomly pick observations from the data for the test data  
test\_id = sample(1:nrow(College),size=floor(0.30\*length(1:nrow(College))))  
College\_train <- College[-test\_id, ]  
College\_test <- College[test\_id, ]

fit\_linear <- lm(Apps ~ ., data = College\_train)  
mse = function(model, y, data) {  
 yhat = predict(model, data)  
 mean((y - yhat)^2)  
}  
training\_err\_linear = mse(fit\_linear, College\_train$Apps, College\_train)  
training\_err\_linear

## [1] 993164.6

test\_err\_linear = mse(fit\_linear, College\_test$Apps, College\_test)  
test\_err\_linear

## [1] 1300431

The training error is 993164.6, and the test error is 1300431.

#forward selection  
regfit\_fwd = SignifReg(Apps~., data = College\_train, alpha = 0.05, direction = "forward",   
 criterion = "p-value", correction = "FDR")  
summary(regfit\_fwd)

##   
## Call:  
## lm(formula = reg, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5647.2 -445.2 -28.0 320.9 6877.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 532.67828 275.39506 1.934 0.053610 .   
## Accept 1.66367 0.04262 39.038 < 2e-16 \*\*\*  
## Top10perc 45.25051 6.41865 7.050 5.54e-12 \*\*\*  
## Enroll -0.74356 0.11826 -6.287 6.70e-10 \*\*\*  
## PrivateYes -785.61216 134.10855 -5.858 8.18e-09 \*\*\*  
## Expend 0.04746 0.01236 3.841 0.000137 \*\*\*  
## PhD -10.54103 3.50697 -3.006 0.002773 \*\*   
## Top25perc -11.86217 5.00593 -2.370 0.018159 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1029 on 536 degrees of freedom  
## Multiple R-squared: 0.933, Adjusted R-squared: 0.9322   
## F-statistic: 1067 on 7 and 536 DF, p-value: < 2.2e-16

training\_err\_fwd = mse(regfit\_fwd, College\_train$Apps, College\_train)  
training\_err\_fwd

## [1] 1043037

test\_err\_fwd = mse(regfit\_fwd, College\_test$Apps, College\_test)  
test\_err\_fwd

## [1] 1334782

#backward selection  
regfit\_bwd = SignifReg(Apps~., data = College\_train, alpha = 0.05, direction = "backward",   
 criterion = "p-value", correction = "FDR")  
summary(regfit\_bwd)

##   
## Call:  
## lm(formula = reg, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5589.8 -440.1 -1.4 315.3 6658.0   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 182.17566 203.56862 0.895 0.37124   
## PrivateYes -403.41491 146.86794 -2.747 0.00622 \*\*   
## Accept 1.69288 0.04301 39.361 < 2e-16 \*\*\*  
## Enroll -0.83323 0.11921 -6.990 8.21e-12 \*\*\*  
## Top10perc 45.82197 6.35006 7.216 1.84e-12 \*\*\*  
## Top25perc -12.12395 4.91888 -2.465 0.01402 \*   
## Outstate -0.08479 0.01876 -4.519 7.65e-06 \*\*\*  
## Expend 0.06782 0.01346 5.038 6.45e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1018 on 536 degrees of freedom  
## Multiple R-squared: 0.9344, Adjusted R-squared: 0.9335   
## F-statistic: 1091 on 7 and 536 DF, p-value: < 2.2e-16

training\_err\_bwd = mse(regfit\_bwd, College\_train$Apps, College\_train)  
training\_err\_bwd

## [1] 1021693

test\_err\_bwd = mse(regfit\_bwd, College\_test$Apps, College\_test)  
test\_err\_bwd

## [1] 1355206

By forward selection, the variables PrivateYes, Accept, Enroll, Top10perc, Top25perc, PhD and Expend are recommended to include in the final model. The train error is 1043037, and the test error is 1334782.  
By backward selection, the variables PrivateYes, Accept, Enroll, Top10perc, Top25perc, Outstate and Expend are recommended to include in the final model. The train error is 1021693, and the test error is 1355206.

regfit.full = regsubsets(Apps~., data = College\_train, nvmax = 18)  
reg.summary = summary(regfit.full)  
#AIC  
id\_AIC = which.min(reg.summary$cp)  
coef(regfit.full, id\_AIC)

## (Intercept) PrivateYes Accept Enroll Top10perc   
## 92.77034574 -525.34275680 1.67339267 -0.78553299 46.44504933   
## Top25perc Outstate Room.Board PhD Expend   
## -11.82100842 -0.10008892 0.11938762 -7.40829931 0.06759232   
## Grad.Rate   
## 5.00261295

names(coef(regfit.full, id\_AIC))

## [1] "(Intercept)" "PrivateYes" "Accept" "Enroll" "Top10perc"   
## [6] "Top25perc" "Outstate" "Room.Board" "PhD" "Expend"   
## [11] "Grad.Rate"

model\_AIC = lm(Apps ~ Private+Accept+Enroll+Top10perc+Top25perc+Outstate+Room.Board+PhD+Expend  
 +Grad.Rate, data = College\_train)  
training\_err\_AIC = mse(model\_AIC, College\_train$Apps, College\_train)  
training\_err\_AIC

## [1] 1001215

test\_err\_AIC = mse(model\_AIC, College\_test$Apps, College\_test)  
test\_err\_AIC

## [1] 1282321

#BIC  
id\_BIC = which.min(reg.summary$bic)  
coef(regfit.full, id\_BIC)

## (Intercept) PrivateYes Accept Enroll Top10perc   
## -163.96646146 -386.22739630 1.68324007 -0.82769819 32.90344332   
## Outstate Expend   
## -0.08889235 0.07474331

names(coef(regfit.full, id\_BIC))

## [1] "(Intercept)" "PrivateYes" "Accept" "Enroll" "Top10perc"   
## [6] "Outstate" "Expend"

model\_BIC = lm(Apps ~ Private+Accept+Enroll+Top10perc+Outstate+Expend, data = College\_train)  
training\_err\_BIC = mse(model\_BIC, College\_train$Apps, College\_train)  
training\_err\_BIC

## [1] 1033273

test\_err\_BIC = mse(model\_BIC, College\_test$Apps, College\_test)  
test\_err\_BIC

## [1] 1380054

By AIC criterian, the variables PrivateYes, Accept, Enroll, Top10perc, Top25perc, Outstate, Room.Board, PhD, Expend, and Grad.Rate are recommended to include in the final model. The train error is 1001215, and the test error is 1282321.

By BIC criterian, the variables PrivateYes, Accept, Enroll, Top10perc, Outstate, and Expend are recommended to include in the final model. The train error is 1033273, and the test error is 1380054.

X = model.matrix(Apps~., College\_train)[, -1]  
y = College\_train$Apps  
grid = 10^seq(10, -2, length = 100)  
ridge.mod = glmnet(X, y, alpha = 0, lambda = grid)  
cv.out\_ridge = cv.glmnet(X, y, alpha = 0)  
minlam\_ridge = cv.out\_ridge$lambda.min  
minlam\_ridge

## [1] 411.4072

ridge.pred\_train = predict(ridge.mod, s = minlam\_ridge, newx = X)  
training\_err\_ridge = mean((ridge.pred\_train - y)^2)  
training\_err\_ridge

## [1] 1384811

ridge.pred\_test = predict(ridge.mod, s = minlam\_ridge, newx = model.matrix(Apps~., College\_test)[, -1])  
test\_err\_ridge = mean((ridge.pred\_test - College\_test$Apps)^2)  
test\_err\_ridge

## [1] 1223126

The value of chosen by smallest cross-validation error is 411.4072. The train error is 1384811, and the test error is 1223126.

set.seed(23456)  
lasso.mod = glmnet(X, y, alpha = 1, lambda = grid)  
cv.out\_lasso = cv.glmnet(X, y, alpha = 1)  
minlam\_lasso = cv.out\_lasso$lambda.min  
minlam\_lasso

## [1] 3.495947

lasso.pred\_train = predict(lasso.mod, s = minlam\_lasso, newx = X)  
training\_err\_lasso = mean((lasso.pred\_train - y)^2)  
training\_err\_lasso

## [1] 994152.8

lasso.pred\_test = predict(lasso.mod, s = minlam\_lasso, newx = model.matrix(Apps~., College\_test)[, -1])  
test\_err\_lasso = mean((lasso.pred\_test - College\_test$Apps)^2)  
test\_err\_lasso

## [1] 1292839

The value of chosen by smallest cross-validation error is 3.495947. The train error is 994152.8, and the test error is 1292839.

1. The test errors of different methods range from 1223126 to 1380054, with mean 1309823 and standard deviation 52065. We can predict the number of college applications received most accurately by ridge regression. For prediction, I recommend the ridge regression method because it has the smallest test error from the prediction model. For interpretation, I recommend the AIC criterion method because it uses fewer variables than the other with similar test errors.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | OLS | Forward | Backward | AIC | BIC | Ridge | Lasso |
| Train error | 993165 | 1043037 | 1021693 | 1001215 | 1033273 | 1384811 | 994153 |
| Test error | 1300431 | 1334782 | 1355206 | 1282321 | 1380054 | 1223126 | 1292839 |
| Nv | 17 | 7 | 7 | 10 | 6 | 17 | 17 |