Math642\_HW4\_FyonaSun

Fyona Sun

2/5/2020

## 6.1

We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain p + 1 models, containing 0, 1, 2, . . . , p predictors. Explain your answers: ###(a) Which of the three models with k predictors has the smallest training RSS? The model with k predictors with the smallest training RSS can be obtained from the best subset selection. Since when applying the best subset select, the model with k predictors is the model with the smallest RSS among all the models. On the other hand with the forward stepwise selection or the backward stepwise selection, the model with k predictors start with the best k-1 model and chosing the best k given a fixed k-1 (forward) or in reverse start at the best k+1 and chosing the best single feature to remove resulting in the best model.

### (b) Which of the three models with k predictors has the smallest test RSS?

The training error can be a poor estimate of the test error. In order to select the best model with respect to test error, we need to estimate this test error by either calculating and comparing the , AIC, BIC, and adjusted or performing a cross-validation. ###(c) True or False: i. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identified by forward stepwise selection. True. In the forward stepwise selection, the model with k+1 predictors is obtained by augmenting the predictors in the model with k predictors with one additional feature.

1. The predictors in the k-variable model identified by back- ward stepwise are a subset of the predictors in the (k + 1)- variable model identified by backward stepwise selection.

True. The k variable model contains all but one feature in the k+1 best model, minus the single feature resulting in the smallest gain in RSS.

1. The predictors in the k-variable model identified by back- ward stepwise are a subset of the predictors in the (k + 1)- variable model identified by forward stepwise selection.

False. There is no direct linkage between forward stepwise selection and backward stepwise selection. They could be disjoint sets.

1. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identified by backward stepwise selection.

False. There is no direct linkage between forward stepwise selection and backward stepwise selection. They could be disjoint sets.

1. The predictors in the k-variable model identified by best subset are a subset of the predictors in the (k + 1)-variable model identified by best subset selection.

False. The model with k+1 variable is obtained by selecting among all possible models with k+1 variables. It does not necessarily contain all the predictors selected for the k-variable model.

## 6.2

For parts (a) through (c), indicate which of i. through iv. is correct. Justify your answer. (a) The lasso, relative to least squares, is: i. More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance. ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias. iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance. iv. Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

The statement iii is true. The LASSO is a more restrictive model, so it could reduce overfitting and variance in predictions.

1. Repeat (a) for ridge regression relative to least squares. The statement iii is true. The ridge regression is also a more restrictive model, so it could reduce overfitting and variance in predictions.
2. Repeat (a) for non-linear methods relative to least squares. The statement ii is true. Non-linear methods are more flexible and will give improved prediction accuracy when their increase in variance are less than their decrease in bias.

## 6.3

Suppose we estimate the regression coefficients in a linear regression model by minimizing subject to for a particular value of s.

1. As we increase s from 0, the training RSS will:
2. Increase initially, and then eventually start decreasing in an inverted U shape.
3. Decrease initially, and then eventually start increasing in a U shape.
4. Steadily increase.
5. Steadily decrease.
6. Remain constant. The statement iv is true. As s increases, the l1 penalty increases, creating less restriction. When s is sufficiently large, it will shrink the RSS until it eventually yields the least squares solution.
7. Repeat (a) for test RSS. (c) Repeat (a) for variance.

For the test RSS the statement ii is true. Initially as are forced to 0, the test RSS will improve as the model has less overfitting. However eventually necessary coefficients will be removed from the model, and the test RSS will then increase, making a U shape. For variance, the statement iii is true. The variance will increase as more penalty is placed on the model.

1. Repeat (a) for (squared) bias. The statement iv is true. As we increase s from 0, we are restricting the more and more, and so the model is becoming less and less flexible which provokes a steady increase in bias.
2. Repeat (a) for the irreducible error. The statement v is true. The irreducible error would remain the same.

## 6.8

In this exercise, we will generate simulated data, and will then use this data to perform best subset selection.

1. Use the rnorm() function to generate a predictor X of length n = 100, as well as a noise vector of length n = 100.

set.seed(1)  
x <- rnorm(100)  
eps <- rnorm(100)

1. Generate a response vector Y of length n = 100 according to the model

b0 <- 2  
b1 <- 3  
b2 <- -1  
b3 <- -2  
y <- b0 + b1 \* x + b2 \* x^2 + b3 \* x^3 + eps

1. Use the regsubsets() function to perform best subset selection in order to choose the best model containing the predictors

library(leaps)  
library(tidyverse)

## ── Attaching packages ────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.2.1 ✔ purrr 0.3.2  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.8  
## ✔ tidyr 0.8.2 ✔ stringr 1.4.0  
## ✔ readr 1.3.1 ✔ forcats 0.3.0

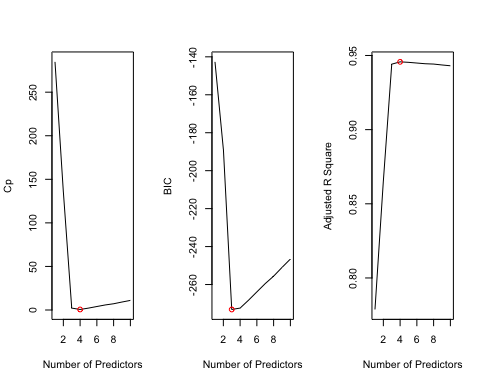
## Warning: package 'ggplot2' was built under R version 3.5.2

## Warning: package 'purrr' was built under R version 3.5.2

## Warning: package 'stringr' was built under R version 3.5.2

## ── Conflicts ───────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

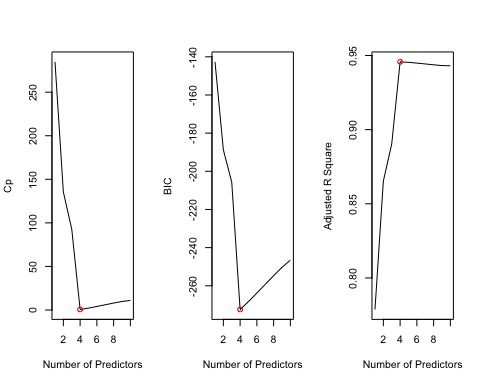
data <- data.frame(y = y, x = x)  
fit <- regsubsets(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) + I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data, nvmax = 10)  
  
fit.summary <- summary(fit)  
  
par(mfrow=c(1,3))  
plot(1:10, fit.summary$cp, xlab="Number of Predictors", ylab="Cp", type="l")  
cp.min=min(fit.summary$cp)  
points(c(1:10)[fit.summary$cp==cp.min], cp.min, pch=1, col="red")  
plot(1:10, fit.summary$bic, xlab="Number of Predictors", ylab="BIC", type="l")  
bic.min=min(fit.summary$bic)  
points(c(1:10)[fit.summary$bic==bic.min], bic.min, pch=1, col="red")  
plot(1:10, fit.summary$adjr2,xlab="Number of Predictors", ylab="Adjusted R Square", type="l")  
adjr2.max=max(fit.summary$adjr2)  
points(c(1:10)[fit.summary$adjr2==adjr2.max], adjr2.max, pch=1, col="red")



The best model selected by contains 4 predictors, . The best model selected by BIC contains 3 predictors, . The best model selected by adjusted contains 4 predictors,

1. Repeat (c), using forward stepwise selection and also using backwards stepwise selection. How does your answer compare to the results in (c)?

##forward stepwise  
fit.frd <- regsubsets(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) + I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data, nvmax = 19, method="forward")  
fit.summary <- summary(fit.frd)  
par(mfrow=c(1,3))  
plot(1:10, fit.summary$cp, xlab="Number of Predictors", ylab="Cp", type="l")  
cp.min=min(fit.summary$cp)  
points(c(1:10)[fit.summary$cp==cp.min], cp.min, pch=1, col="red")  
plot(1:10, fit.summary$bic, xlab="Number of Predictors", ylab="BIC", type="l")  
bic.min=min(fit.summary$bic)  
points(c(1:10)[fit.summary$bic==bic.min], bic.min, pch=1, col="red")  
plot(1:10, fit.summary$adjr2,xlab="Number of Predictors", ylab="Adjusted R Square", type="l")  
adjr2.max=max(fit.summary$adjr2)  
points(c(1:10)[fit.summary$adjr2==adjr2.max], adjr2.max, pch=1, col="red")

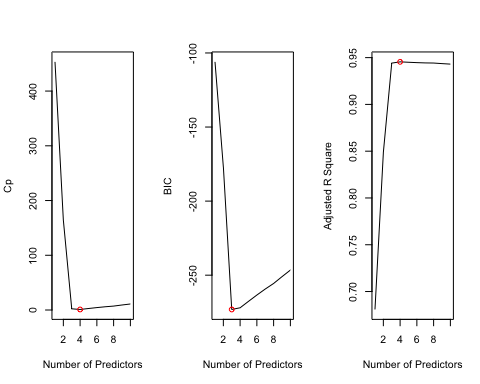


fit.summary

## Subset selection object  
## Call: regsubsets.formula(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) +   
## I(x^6) + I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data,   
## nvmax = 19, method = "forward")  
## 10 Variables (and intercept)  
## Forced in Forced out  
## x FALSE FALSE  
## I(x^2) FALSE FALSE  
## I(x^3) FALSE FALSE  
## I(x^4) FALSE FALSE  
## I(x^5) FALSE FALSE  
## I(x^6) FALSE FALSE  
## I(x^7) FALSE FALSE  
## I(x^8) FALSE FALSE  
## I(x^9) FALSE FALSE  
## I(x^10) FALSE FALSE  
## 1 subsets of each size up to 10  
## Selection Algorithm: forward  
## x I(x^2) I(x^3) I(x^4) I(x^5) I(x^6) I(x^7) I(x^8) I(x^9)  
## 1 ( 1 ) " " " " " " " " "\*" " " " " " " " "   
## 2 ( 1 ) " " "\*" " " " " "\*" " " " " " " " "   
## 3 ( 1 ) "\*" "\*" " " " " "\*" " " " " " " " "   
## 4 ( 1 ) "\*" "\*" "\*" " " "\*" " " " " " " " "   
## 5 ( 1 ) "\*" "\*" "\*" " " "\*" "\*" " " " " " "   
## 6 ( 1 ) "\*" "\*" "\*" " " "\*" "\*" " " " " "\*"   
## 7 ( 1 ) "\*" "\*" "\*" " " "\*" "\*" "\*" " " "\*"   
## 8 ( 1 ) "\*" "\*" "\*" " " "\*" "\*" "\*" "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*" " " "\*" "\*" "\*" "\*" "\*"   
## 10 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## I(x^10)  
## 1 ( 1 ) " "   
## 2 ( 1 ) " "   
## 3 ( 1 ) " "   
## 4 ( 1 ) " "   
## 5 ( 1 ) " "   
## 6 ( 1 ) " "   
## 7 ( 1 ) " "   
## 8 ( 1 ) " "   
## 9 ( 1 ) "\*"   
## 10 ( 1 ) "\*"

The best model selected by contains 4 predictors, . The best model selected by BIC contains 4 predictors, . The best model selected by adjusted contains 4 predictors,

##backward stepwise  
fit.bwd <- regsubsets(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) + I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data, nvmax = 19, method="backward")  
fit.summary <- summary(fit.bwd)  
par(mfrow=c(1,3))  
plot(1:10, fit.summary$cp, xlab="Number of Predictors", ylab="Cp", type="l")  
cp.min=min(fit.summary$cp)  
points(c(1:10)[fit.summary$cp==cp.min], cp.min, pch=1, col="red")  
plot(1:10, fit.summary$bic, xlab="Number of Predictors", ylab="BIC", type="l")  
bic.min=min(fit.summary$bic)  
points(c(1:10)[fit.summary$bic==bic.min], bic.min, pch=1, col="red")  
plot(1:10, fit.summary$adjr2,xlab="Number of Predictors", ylab="Adjusted R Square", type="l")  
adjr2.max=max(fit.summary$adjr2)  
points(c(1:10)[fit.summary$adjr2==adjr2.max], adjr2.max, pch=1, col="red")



fit.summary

## Subset selection object  
## Call: regsubsets.formula(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) +   
## I(x^6) + I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data,   
## nvmax = 19, method = "backward")  
## 10 Variables (and intercept)  
## Forced in Forced out  
## x FALSE FALSE  
## I(x^2) FALSE FALSE  
## I(x^3) FALSE FALSE  
## I(x^4) FALSE FALSE  
## I(x^5) FALSE FALSE  
## I(x^6) FALSE FALSE  
## I(x^7) FALSE FALSE  
## I(x^8) FALSE FALSE  
## I(x^9) FALSE FALSE  
## I(x^10) FALSE FALSE  
## 1 subsets of each size up to 10  
## Selection Algorithm: backward  
## x I(x^2) I(x^3) I(x^4) I(x^5) I(x^6) I(x^7) I(x^8) I(x^9)  
## 1 ( 1 ) " " " " "\*" " " " " " " " " " " " "   
## 2 ( 1 ) "\*" " " "\*" " " " " " " " " " " " "   
## 3 ( 1 ) "\*" "\*" "\*" " " " " " " " " " " " "   
## 4 ( 1 ) "\*" "\*" "\*" " " " " " " " " " " "\*"   
## 5 ( 1 ) "\*" "\*" "\*" " " " " " " " " "\*" "\*"   
## 6 ( 1 ) "\*" "\*" "\*" " " " " " " " " "\*" "\*"   
## 7 ( 1 ) "\*" "\*" "\*" " " " " "\*" " " "\*" "\*"   
## 8 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" " " "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"   
## 10 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## I(x^10)  
## 1 ( 1 ) " "   
## 2 ( 1 ) " "   
## 3 ( 1 ) " "   
## 4 ( 1 ) " "   
## 5 ( 1 ) " "   
## 6 ( 1 ) "\*"   
## 7 ( 1 ) "\*"   
## 8 ( 1 ) "\*"   
## 9 ( 1 ) "\*"   
## 10 ( 1 ) "\*"

The best model selected by contains 4 predictors, . The best model selected by BIC contains e predictors, . The best model selected by adjusted contains 4 predictors,

1. Now fit a lasso model to the simulated data. Use cross-validation to select the optimal value of . Create plots of the cross-validation error as a function of . Report the resulting coefficient estimates, and discuss the results obtained.

library(glmnet)

## Warning: package 'glmnet' was built under R version 3.5.2

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':  
##   
## expand

## Loading required package: foreach

##   
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':  
##   
## accumulate, when

## Loaded glmnet 2.0-18

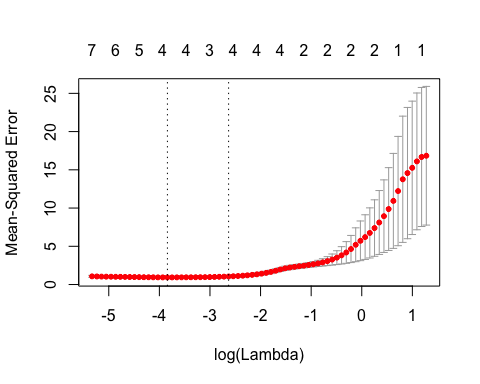
x=cbind(x,x^2,x^3,x^4,x^5,x^6,x^7,x^8,x^9,x^10)  
y=y  
  
##Cross-validation  
lasso.cv = cv.glmnet(x,y, alpha=1)  
lasso.cv$lambda.min

## [1] 0.02146947

lasso.cv$lambda.1se

## [1] 0.07195712

plot(lasso.cv)



### Refit the model  
lasso.mod=glmnet(x,y,alpha=1, lambda=lasso.cv$lambda.min)  
coef(lasso.mod)[,1]

## (Intercept) x   
## 2.0843824946 2.8766388187 -1.1683609866 -1.9571360979 0.0000000000   
##   
## 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000   
##   
## 0.0001296694

lasso.cv$lambda.min

## [1] 0.02146947

lasso.cv$lambda.1se

## [1] 0.07195712

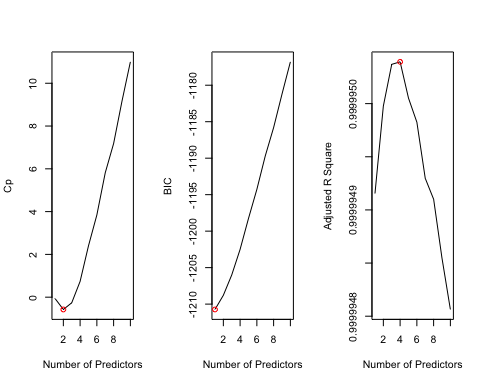
coef(lasso.mod)[,1]

## (Intercept) x   
## 2.0843824946 2.8766388187 -1.1683609866 -1.9571360979 0.0000000000   
##   
## 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000   
##   
## 0.0001296694

The red dots indicate the cross-validation curve. The upper and lower standard deviation curves along the sequence of values. Two selected values are indicated by the vertical dotted line. One is selected by minimizing cv error and the other is one standard devation of the minimum cv errorthe. The values are 98.97694 and 108.6271 respectively.With the value of giving the minimum cv error, the Lasso shrinks the majority predictors to zero, and only leaves and .

1. Now generate a response vector Y according to the model and and perform best subset selection and the lasso. Discuss the results obtained.

set.seed(1)  
x <- rnorm(100)  
eps <- rnorm(100)  
b7 <- 7  
y <- b0 + b7 \* x^7 + eps  
  
data.full <- data.frame(y = y, x = x)  
fit.best <- regsubsets(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) + I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data.full, nvmax = 10)  
fit.summary <- summary(fit.best)  
par(mfrow=c(1,3))  
plot(1:10, fit.summary$cp, xlab="Number of Predictors", ylab="Cp", type="l")  
cp.min=min(fit.summary$cp)  
points(c(1:10)[fit.summary$cp==cp.min], cp.min, pch=1, col="red")  
plot(1:10, fit.summary$bic, xlab="Number of Predictors", ylab="BIC", type="l")  
bic.min=min(fit.summary$bic)  
points(c(1:10)[fit.summary$bic==bic.min], bic.min, pch=1, col="red")  
plot(1:10, fit.summary$adjr2,xlab="Number of Predictors", ylab="Adjusted R Square", type="l")  
adjr2.max=max(fit.summary$adjr2)  
points(c(1:10)[fit.summary$adjr2==adjr2.max], adjr2.max, pch=1, col="red")



coef(fit.best, 1)

## (Intercept) I(x^7)   
## 1.95894 7.00077

#best subset selection with BIC picks the best 1-variable model Y= 1.95894+7.00077X^7

The model picks X^7 with a coeff is pretty much closer to 7.

#lasso  
xmat <- model.matrix(y ~ x + I(x^2) + I(x^3) + I(x^4) + I(x^5) + I(x^6) + I(x^7) + I(x^8) + I(x^9) + I(x^10), data = data.full)[, -1]  
cv.lasso <- cv.glmnet(xmat, y, alpha = 1)  
bestlam <- cv.lasso$lambda.min  
bestlam

## [1] 12.36884

fit.lasso <- glmnet(xmat, y, alpha = 1)  
predict(fit.lasso, s = bestlam, type = "coefficients")[1:11, ]

## (Intercept) x I(x^2) I(x^3) I(x^4) I(x^5)   
## 2.820215 0.000000 0.000000 0.000000 0.000000 0.000000   
## I(x^6) I(x^7) I(x^8) I(x^9) I(x^10)   
## 0.000000 6.796694 0.000000 0.000000 0.000000

#the LASSO picks the best 1-variable model with Y=2.820215+6.796694 X^7

The model picks X^7 with a coeff 6.796694, which is a little bit off but still can estimate the y.