STA 6703 SML HW6

# CH6. Exercise 1

## 1.a

The best subset selection has the smallest training RSS. This is because the best subset selects the best model from all possible models, while the forward stepwise and backward stepwise methods only iteratively add/drop one predictor based on the previous step’s model without attempting all possibilities. Therefore, the best subset selection performs better on the training data set.

## 1.b

Any of the three selections may have the smallest test RSS. Suppose . If the best one-variable model is but the best two-variable contains and , then the forward stepwise will fail to select the best two-variable model because the two-variable model contains , resulting in a large test RSS. Similarly, in backward stepwise selection, it will fail to select the best one-variable model because the model either contains or without containing . In this case, the best subset selection will have the smallest test RSS. However, if forward stepwise or backward stepwise gets the best possible model, then they will get the smallest test RSS.

## 1.c TRUE or FALSE

1. **TRUE** This is the selection step in the forward stepwise method. It conducts the (k+1)-variable model selection based on the selected k-variable model.
2. **TRUE** When conducting the backward stepwise selection, we select the best k-variable model from the (k+1) predictors.
3. **FALSE** The predictors in the best (k+1)-variable model selected by the forward stepwise method may differ from the ones selected by the backward stepwise method. Thus, the predictors in the k-variable model identified by backward stepwise may not be a subset of the predictors in the (k+1)-variable model identified by forward stepwise selection.
4. **FALSE** Similar reasons as we described in iii.
5. **FALSE** The best subset selection (k+1)-variable model by conducting all possible models, which separates from selection k-variable model.

# CH6. Exercise 2

## 2.a Lasso

iii is correct. The lasso performs variable selection, which forces some of the coefficient estimates to be zero by increasing the tunning parameter . Thus, as the increases, the model becomes less flexible. As increases, the prediction accuracy increases when the estimated bias increases less than the decrease in variance.

## 2.b Ridge regression

iii is correct. Though ridge regression cannot force some of the coefficient estimates exactly to be zero, it still aims to minimize the coefficients. Similarly, as we descibed in 2.a, as the increases, the prediction accuracy increases when the estimated bias increases less than the decrease in variance.

## 2.c Non-linear methods

ii is correct. Non-linear methods aim to add some quadrative and/or cubic terms to the model to increase the model flexibility and thus decrease the estimated bias. Hence, the prediction accuracy increases when its increase in variance is less than the decrease in bias.

# CH6. Exercise 4

## 4.a The training RSS will

steadily increase. As increases, the coefficient estimates tend to zero, which is similar to the number of variables decreases, and thus the training RSS will increase due to the less flexibility.

## 4.b The test RSS will

decrease initially, and then eventually start increasing in a U shape. As increases, the test RSS will decrease initially because the variance decreases larger than the increase in bias. However, when the decrease in variance is smaller than the increase in bias, the test RSS will change to increase.

## 4.c Variance will

steadily decrease. Similarly, as we described in 4.a, the variance decreases due to the less flexibility of the model.

## 4.d Squared bias will

steadily increase. Similarly, as we described in 4.a, as the increases, the model becomes less flexible and thus increases more noise. Hence, it causes an increase in the squared bias.

## 4.e The irreducible error will

remains constant. The irreducible error is random, which is irrelate to the .

# CH6. Exercise 8

## 8.a

n=100  
set.seed(0)  
X <- rnorm(n)  
noise <- rnorm(n)

## 8.b

beta0 = 1  
beta1 = 1.2  
beta2 = 2  
beta3 = 3  
  
Y <- beta0 + beta1 \* X + beta2 \* X^2 + beta3 \* X^3 + noise

## 8.c

#best subset selection  
data <- data.frame(cbind(X,Y))  
modelfit <- regsubsets(Y ~ poly(X, degree = 10, raw = T), data = data, nvmax = 10)  
results <- summary(modelfit)  
  
which.min(results$cp)

## [1] 3

which.min(results$bic)

## [1] 3

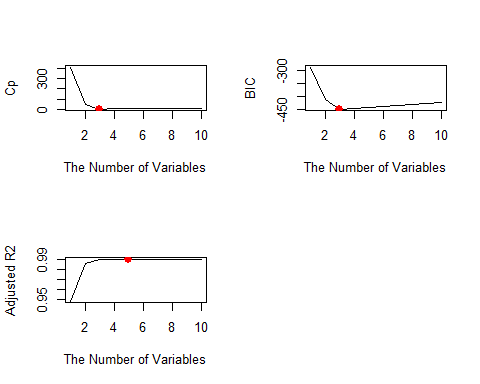
which.max(results$adjr2)

## [1] 5

#difference in the adjusted R2  
results$adjr2[which.min(results$cp)] - results$adjr2[which.max(results$adjr2)]

## [1] -0.0001257791

As the plots of , , and adjusted show, the model with the lowest and is the three-variable model. The model with the maximum adjusted is the five-variable model but the difference between the three-variable model and the five-variable model is small. Therefore, we conclude that the three-variable model with , , and is the best model.



**Table. The coefficients of the best model**

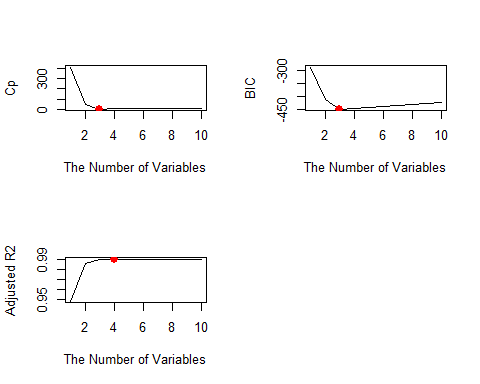
|  | Est. coefficient |
| --- | --- |
| Intercept | 0.9660167 |
| X | 1.2301865 |
| X\_square | 1.9706439 |
| X\_cubic | 3.0503899 |

## 8.d

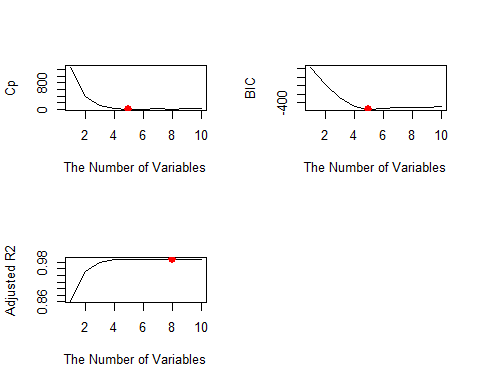
#forward stepwise selection  
modelfit\_fwd <- regsubsets(Y ~ poly(X, degree = 10, raw = T), data = data, nvmax = 10,  
 method = "forward")  
results\_fwd <- summary(modelfit\_fwd)  
  
#backward stepwise selection  
modelfit\_bwd <- regsubsets(Y ~ poly(X, degree = 10, raw = T), data = data, nvmax = 10,  
 method = "backward")  
results\_bwd <- summary(modelfit\_bwd)

Compared to the answer in 8.c, our conclusion in the forward stepwise is the same as the best subset selection. The best model is the three-variable model with , , and , which is identical to the best subset selection. However, the backward stepwise selection has different conclusion, which concludes that the best model is five-variable model with predictors , , , , and .

**Forward stepwise selection outcomes**



**Backward stepwise selection outcomes**



**Table. The coefficients of the best model in forward stepwise selection**

|  | Est. coefficient |
| --- | --- |
| Intercept | 0.9660167 |
| X | 1.2301865 |
| X\_square | 1.9706439 |
| X\_cubic | 3.0503899 |

**Table. The coefficients of the best model in backward stepwise selection**

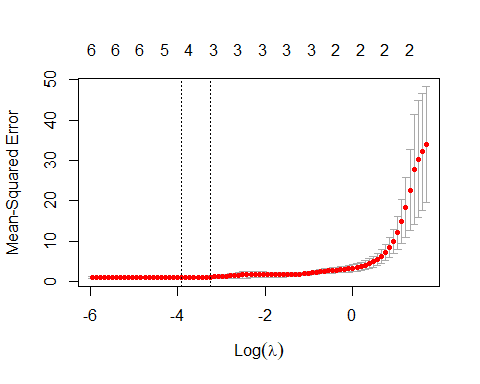
|  | Est. coefficient |
| --- | --- |
| Intercept | 1.0171276 |
| X | 2.3147556 |
| X\_square | 1.9052995 |
| X\_5 | 2.2217950 |
| X\_7 | -0.5606459 |
| X\_9 | 0.0453176 |

## 8.e

#lasso  
X\_matrix <- poly(X, degree = 10, raw = TRUE)  
#10-fold cross-validation  
set.seed(0)  
train\_index <- sample(1:n, size = n/2)  
  
cv.out <- cv.glmnet(X\_matrix[train\_index, ], Y[train\_index],  
 alpha = 1, nfolds = 10)  
  
#test MSE is  
lasso\_predict <- predict(cv.out, s = cv.out$lambda.min, newx = X\_matrix[-train\_index, ])  
  
mean((lasso\_predict - Y[-train\_index])^2)

## [1] 1.526109

**Plot of cross-validation errors**



The optimal value of is

## [1] 0.02

**Table. Estimated coefficients by lasso**

modelfit\_lasso <- glmnet(X\_matrix, Y, alpha = 1)  
results\_lasso <- predict(modelfit\_lasso, type = "coefficients", s = cv.out$lambda.min)  
results\_lasso

## 11 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 1.08421070  
## 1 1.23920700  
## 2 1.69850220  
## 3 3.01195952  
## 4 0.05602966  
## 5 .   
## 6 .   
## 7 .   
## 8 .   
## 9 .   
## 10 .

Lasso selects the best model, which is the four-variable model with predictors , , , and . The conclusion is different from the outcomes in the best subset, forward stepwise and backward stepwise selections.

## 8.f

#new generation  
beta7 = 1.1  
Y <- beta0 + beta7\*X^7+noise  
data2 <- data.frame(cbind(Y, X))

#the best subset selection  
new\_modelfit <- regsubsets(Y ~ poly(X, degree = 10, raw = TRUE), data = data2)  
  
new\_results <- summary(new\_modelfit)  
  
which.min(new\_results$cp)

## [1] 1

which.min(new\_results$bic)

## [1] 1

which.max(new\_results$adjr2)

## [1] 4

new\_results$adjr2[which.min(new\_results$cp)] - new\_results$adjr2[which.max(new\_results$adjr2)]

## [1] -2.888794e-06

#The difference in the adjusted R2 between   
 #the one-variable model and the four-variable model is small  
  
#best model  
t <- data.frame(coef(new\_modelfit, which.min(new\_results$cp)))  
colnames(t) <- "Est. coefficient"  
rownames(t) <- c("Intercept", "X7")  
kable(t)

|  | Est. coefficient |
| --- | --- |
| Intercept | 0.9432264 |
| X7 | 1.1015262 |

#lasso  
cv.out <- cv.glmnet(X\_matrix[train\_index,], Y[train\_index], alpha = 1)  
new\_modelfit\_lasso <- glmnet(X\_matrix, Y, alpha = 1)  
lasso.coef <- predict(new\_modelfit\_lasso, type = "coefficients", s = cv.out$lambda.min)  
lasso.coef

## 11 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 1.18523374  
## 1 .   
## 2 .   
## 3 .   
## 4 .   
## 5 0.06724044  
## 6 .   
## 7 1.05757773  
## 8 .   
## 9 .   
## 10 .

For the best subset selection, the best model is the one-variable model with .

For the lasso, the best model is the two-variable model with and .

In this data set, the best subset selection performs better in estimates than the lasso.