STA567 HW2

Lina Lee

9/16/2019

Problem 1

As the model flexibility increases, $\hat{f}(X)$ get close to $f(X)$.

a) $Bias[\hat{f}(X)]$
   We can write $Bias[\hat{f}(X)]$ as $E[\hat{f}(X) - f(X)]$. As complexity increases, $\hat{f}(X)$ get close to $f(X)$. So, $E\left[\hat{f}(X) - f(X)\right]$ goes to zero.

b) $Var[\hat{f}(X)]$
   The variance of predicted value does not depend on the model flexibility.
   $Var[\hat{f}(X)]$ $=\sigma^2 X[X'X]^{-1}X'$ Therefore, $Var[\hat{f}(X)]$ does not change even though the model flexibility increase.

c) $Var[\epsilon]$
   As a model flexibility increases, $Var[\epsilon]$ is constant.

d) Training MSE
   Training MSE decreases as complexity increase.

e) Test MSE
   Test MSE decrease as the complexity of a model increases until the complexity reaches a certain point, after that, the test MSE increases.

Load libraries

```
library(tidyverse)

library(FNN)
library(car)

library(caret)
```

Problem 2

```r
knnData <- data.frame(X1 = c(21,19,16,21,21,17),
                      X2 = c(25,43,36,47,42,37),
                      Y = c(4,7,5,7,8,5))
knnData

##   X1 X2 Y
## 1 21 25 4
## 2 19 43 7
## 3 16 36 5
## 4 21 47 7
## 5 21 42 8
## 6 17 37 5
```

b) Euclidean distance

```r
mydistance <- function(x11,x12,x21,x22){
  sqrt((x11-x21)^2 + (x12-x22)^2)
}
knnData$mydistance<-mydistance(knnData$X1,knnData$X2,20,45)
```

Answer: 2.2360, 2.2361, 3.1623, 8.5440, 9.8489, 20.0250 for each point.

b) What are the indexes for the k=3 nearest neighbors to the new data point? What about the indexes for the k=4 nearest neighbors?

```r
knnData<-knnData[order(knnData$mydistance)[1:6],]

print(head(knnData,n=3))

##   X1 X2 Y mydistance
## 2 19 43 7   2.236068
## 4 21 47 7   2.236068
## 5 21 42 8   3.162278

print(head(knnData,n=4))

##   X1 X2 Y mydistance
## 2 19 43 7   2.236068
## 4 21 47 7   2.236068
## 5 21 42 8   3.162278
## 6 17 37 5   8.544004
```

```r
mean(knnData$Y[order(knnData$mydistance)[1:3]])

## [1] 7.333333

mean(knnData$Y[order(knnData$mydistance)[1:4]])

## [1] 6.75
```

k	Indices of neighbors from training data	Predicted Y
3	2, 4, 5	7.33
4	2, 4, 5, 6	6.75

Problems 3

Read Data from book website

```r
college <- read.csv("http://faculty.marshall.usc.edu/gareth-james/ISL/College
.csv")

college[college$X=="Miami University at Oxford", ]

##                              X Private Apps Accept Enroll Top10perc
## 366 Miami University at Oxford      No 9239   7788   3290        35
##     Top25perc F.Undergrad P.Undergrad Outstate Room.Board Books Personal
## 366        39       13606         807     8856       3960   500     1382
##     PhD Terminal S.F.Ratio perc.alumni Expend Grad.Rate
## 366  81       89      17.6          20   7846        85

row.names(college) <- college$X
college <- college[ , -1]


college$Private <- as.factor(college$Private)
```

(a) A brief description of your model fitting procedure

Regression subset test.

```r
set.seed(09162019)
test_index <- sample(1:777, 277)
test_data <- college[test_index,]
train_data <- college[-test_index,]
```

```
library(leaps)


n <- length(test_data$Grad.Rate)
subset <- regsubsets(Grad.Rate ~.,
                     method="exhaustive", nbest=1, data=test_data)
cbind(summary(subset)$outmat, round(summary(subset)$rsq, 3),
      round(summary(subset)$adjr2, 3), round(summary(subset)$cp, 1), round(sq
rt(summary(subset)$rss/(n-c(rep(1:7,rep(2,7)),8)-1)), 4))

## Warning in summary(subset)$rss/(n - c(rep(1:7, rep(2, 7)), 8) - 1): longer
## object length is not a multiple of shorter object length

## Warning in cbind(summary(subset)$outmat, round(summary(subset)$rsq, 3), :
## number of rows of result is not a multiple of vector length (arg 5)

##          PrivateYes Apps Accept Enroll Top10perc Top25perc F.Undergrad
## 1  ( 1 ) " "        " "  " "    " "    " "       " "       " "
## 2  ( 1 ) " "        " "  " "    " "    " "       "*"       " "
## 3  ( 1 ) " "        " "  " "    " "    " "       "*"       " "
## 4  ( 1 ) " "        " "  " "    " "    " "       "*"       " "
## 5  ( 1 ) "*"        " "  " "    " "    " "       "*"       " "
## 6  ( 1 ) " "        " "  "*"    " "    " "       "*"       "*"
## 7  ( 1 ) "*"        " "  "*"    " "    " "       "*"       "*"
## 8  ( 1 ) "*"        "*"  " "    " "    " "       "*"       "*"
##          P.Undergrad Outstate Room.Board Books Personal PhD Terminal
## 1  ( 1 ) " "         "*"      " "        " "   " "      " " " "
## 2  ( 1 ) " "         "*"      " "        " "   " "      " " " "
## 3  ( 1 ) " "         "*"      " "        " "   " "      " " " "
## 4  ( 1 ) " "         "*"      " "        " "   " "      " " " "
## 5  ( 1 ) " "         "*"      " "        " "   " "      " " " "
## 6  ( 1 ) " "         "*"      " "        " "   " "      " " " "
## 7  ( 1 ) " "         "*"      " "        " "   " "      " " " "
## 8  ( 1 ) " "         "*"      " "        " "   " "      " " " "
##          S.F.Ratio perc.alumni Expend
## 1  ( 1 ) " "       " "         " "    "0.351" "0.348" "45.6" "13.6955"
## 2  ( 1 ) " "       " "         " "    "0.39"  "0.386" "28.2" "13.2707"
## 3  ( 1 ) "*"       " "         " "    "0.41"  "0.404" "20.2" "13.0724"
## 4  ( 1 ) "*"       "*"         " "    "0.431" "0.422" "12.4" "12.847"
## 5  ( 1 ) "*"       "*"         " "    "0.434" "0.424" "12.5" "12.8283"
## 6  ( 1 ) "*"       "*"         " "    "0.453" "0.441" "5.2"  "12.6103"
## 7  ( 1 ) "*"       "*"         " "    "0.458" "0.444" "5"    "12.5819"
## 8  ( 1 ) "*"       "*"         "*"    "0.462" "0.445" "5.2"  "12.5397"
```

Description: The 7th model, 8th, and 9th model has the highest adj-R^2, which is larger than 0.44. However, the model 9 p= 9 but Cp is 7.8, which does not satisfy Cp criterion. So, I will exclude this model. The model 8 has higher adj-R^2 than the model 7. both models satisfy p<Cp criterion. The RSS of the model 8 is lower than the model 7.

AIC/BIC approach

```
model.7<-lm(Grad.Rate~Apps+Top25perc+P.Undergrad+Outstate+Room.Board+perc.alu
mni+Expend,data=test_data)
k <- 7
n*log(sum(residuals(model.7)^2))-n*log(n)+2*(k+1)

## [1] 1425.147

n*log(sum(residuals(model.7)^2))-n*log(n)+log(n)*(k+1)

## [1] 1454.139
```

AIC= 1425.147, BIC=1454.139

```
model.8<-lm(Grad.Rate~Apps+Top25perc+P.Undergrad+Outstate+Room.Board+Personal
+perc.alumni+Expend,data=test_data)
k<-8
n*log(sum(residuals(model.7)^2))-n*log(n)+2*(k+1)

## [1] 1427.147

n*log(sum(residuals(model.7)^2))-n*log(n)+log(n)*(k+1)

## [1] 1459.763
```

AIC= 1427.147, BIC=1459.763

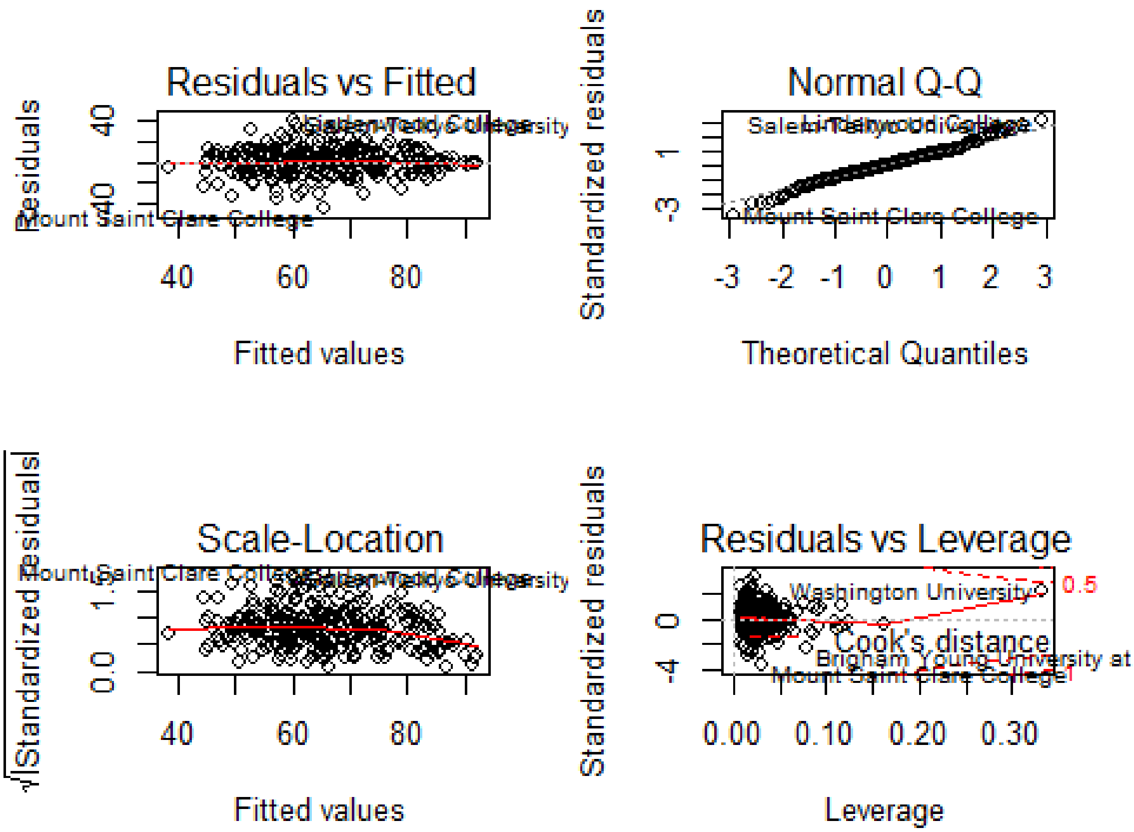
Multicollinearity check

```
vif(model.7)

##        Apps   Top25perc P.Undergrad    Outstate  Room.Board perc.alumni
##    1.783686    1.864878    1.644925    3.465395    1.915278    1.626626
##      Expend
##    2.456803

vif(model.8)

##        Apps   Top25perc P.Undergrad    Outstate  Room.Board    Personal
##    1.785698    1.864879    1.698155    3.526551    1.921673    1.197711
## perc.alumni      Expend
##    1.638530    2.479809
```

Description continued: VIF of both models is not that big. Therefore, I will choose model 8, although AIC/BIC of it is a little bit larger than model 8. The results of the steps are attached below.

(b) A residual plot and statement about model fit.

```r
model.7<-lm(Grad.Rate~Apps+Top25perc+P.Undergrad+Outstate+Room.Board+perc.alu
mni+Expend,data=test_data)

par(mfrow=c(2,2))
plot(model.7)
```



Explanation: In the Residual vs Fitted plot, residuals clustered around the horizontal center line although they are spread over all Fitted value. However, it doesn't have a clear pattern, so it does not have a main problem. It suggests that the model fits the data well.

(c) Calculate and report the R^(2) for your model.

```r
summary(model.7)$r.squared

## [1] 0.4352662

summary(model.7)$adj.r.squared

## [1] 0.4205705
```

Answer: R^2 is 0.44

(d) Calculate and report the training and test MSE for the model.

```
mod.train<-lm(Grad.Rate~Apps+Top25perc+P.Undergrad+Outstate+Room.Board+Person
al+perc.alumni+Expend,data=train_data)
train_data$pred <- predict(mod.train, train_data)
traintMSE <- with(train_data, mean((Grad.Rate-pred)^2))
traintMSE

## [1] 159.5712

mod.test<-lm(Grad.Rate~Apps+Top25perc+P.Undergrad+Outstate+Room.Board+Persona
l+perc.alumni+Expend,data=test_data)
test_data$pred <- predict(model.8, test_data)

testMSE <- with(test_data, mean((Grad.Rate-pred)^2))
testMSE

## [1] 161.1459
```

Answer: Training MSE is 159.5712, and Test MSE: 161.1459

Problems 4

Standardize all numeric input variables for kNN regression of Grad.Rate

```
college[,-c(1,18)] <- as.data.frame(scale(college[,-c(1,18)]))
set.seed(09162019)
test_index <- sample(1:777, 277)
knnreg_test_data <- college[test_index,]
knnreg_train_data <- college[-test_index,]


knnreg1<-knnreg(Grad.Rate~Apps+Accept+Enroll+Top10perc+Top25perc+F.Undergrad+
P.Undergrad+Outstate+Room.Board+Books
                +Personal+PhD+Terminal+S.F.Ratio+perc.alumni+Expend,data=knnr
eg_train_data,k=1)
knnreg_test_data$pred1 <- predict(knnreg1, knnreg_test_data)
knnreg_train_data$pred1 <- predict(knnreg1, knnreg_train_data)

testMSE1 <- with(knnreg_test_data, mean((Grad.Rate-pred1)^2))
trainMSE1 <- with(knnreg_train_data, mean((Grad.Rate-pred1)^2))
testMSE1

## [1] 370.1588

trainMSE1

## [1] 0

knnreg2<-knnreg(Grad.Rate~Apps+Accept+Enroll+Top10perc+Top25perc+F.Undergrad+
P.Undergrad+Outstate+Room.Board+Books
```

```
                +Personal+PhD+Terminal+S.F.Ratio+perc.alumni+Expend
                 ,data=knnreg_train_data, k=20)

knnreg_test_data$pred2 <- predict(knnreg2, knnreg_test_data)
knnreg_train_data$pred2 <- predict(knnreg2, knnreg_train_data)


testMSE2 <- with(knnreg_test_data, mean((Grad.Rate-pred2)^2))
trainMSE2 <- with(knnreg_train_data, mean((Grad.Rate-pred2)^2))
testMSE2

## [1] 185.6359

trainMSE2

## [1] 155.0933

knnreg3<-knnreg(Grad.Rate~Apps+Accept+Enroll+Top10perc+Top25perc+F.Undergrad+
P.Undergrad+Outstate+Room.Board+Books+Personal+PhD+Terminal+S.F.Ratio+perc.al
umni+Expend,data=knnreg_train_data,k=50)
knnreg_test_data$pred3 <- predict(knnreg3, knnreg_test_data)
knnreg_train_data$pred3 <- predict(knnreg3, knnreg_train_data)

testMSE3 <- with(knnreg_test_data, mean((Grad.Rate-pred3)^2))
trainMSE3 <- with(knnreg_train_data, mean((Grad.Rate-pred3)^2))
testMSE3

## [1] 183.0455

trainMSE3

## [1] 170.4048
```

k	Training MSE	Test MSE
1	0	370.1588
20	155.0933	185.6359
50	170.4048	183.0445

Explanation: The model at k=50 has the lowest Test MSE, so it is the best model. Training MLE will be always best when k=1.

Problems 5

Standardize all numeric input variables for kNN classification of Private (Yes/No)

```
college[,-1] <- as.data.frame(scale(college[,-1]))
set.seed(09162019)
test_index <- sample(1:777, 277)
knn_test_data <- college[test_index,]
knn_train_data <- college[-test_index,]



pknn1<- knn3(Private~Apps+Accept+Enroll+Top10perc+Top25perc+F.Undergrad+P.Und
ergrad+Outstate+Room.Board+Books
                   +Personal+PhD+Terminal+S.F.Ratio+perc.alumni+Expend+Grad.R
ate
                   ,knn_train_data,k=1)

test_misclass1<-mean(predict(pknn1,knn_test_data, type="class") != knn_test_d
ata$Private)
test_misclass1

## [1] 0.0866426

train_misclass1<-mean(predict(pknn1,knn_train_data, type="class") != knn_trai
n_data$Private)
train_misclass1

## [1] 0

pknn20 <- knn3(Private~Apps+Accept+Enroll+Top10perc+Top25perc+F.Undergrad+P.U
ndergrad+Outstate+Room.Board+Books
               +Personal+PhD+Terminal+S.F.Ratio+perc.alumni+Expend+Grad.Rate,
data=knn_train_data,k=20)
test_misclass20<-mean(predict(pknn20,knn_test_data, type="class") != knn_test
_data$Private)
test_misclass20

## [1] 0.0830

train_misclass20<-mean(predict(pknn20,knn_train_data, type="class") != knn_tr
ain_data$Private)
train_misclass20

## [1] 0.068

pknn50 <- knn3(Private~Apps+Accept+Enroll+Top10perc+Top25perc+F.Undergrad+P.U
ndergrad+Outstate+Room.Board+Books
```

```
                 +Personal+PhD+Terminal+S.F.Ratio+perc.alumni+Expend+Grad.Rate,
data=knn_train_data,k=50)


test_misclass50<-mean(predict(pknn50,knn_test_data, type="class") != knn_test
_data$Private)
test_misclass50

## [1] 0.1046

train_misclass50<-mean(predict(pknn50,knn_train_data, type="class") != knn_tr
ain_data$Private)
train_misclass50

## [1] 0.076
```

k	Training misclassification error rate	Test misclassification error rate
1	0	0.0866
20	0.068	0.083
50	0.076	0.1046

Explanation: The training misclassification error rate will always increase. The Test misclassification error rate is the smallest at K=20. Therefore, K=20 is the best model to predict college type for a new school.

6) logistic regression misspecification rate

```
knn_train_data$binary <- ifelse(knn_train_data$Private=="Yes", 1, 0)
knn_test_data$binary <- ifelse(knn_test_data$Private=="Yes", 1, 0)


logitmod <- glm(binary~Apps+Accept+Enroll+Top10perc+Top25perc+F.Undergrad+P.U
ndergrad+Outstate+Room.Board+Books
                +Personal+PhD+Terminal+S.F.Ratio+perc.alumni+Expend+Grad.Rate
,
                family = binomial(link=logit),
                data=knn_train_data)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
# Model accuracy
misrate1<-mean(ifelse(predict(logitmod,knn_train_data, type="response") > 0.5
, "Yes", "No") != knn_train_data$Private)
misrate1

## [1] 0.04

misrate2<-mean(ifelse(predict(logitmod,knn_test_data, type="response") > 0.5,
 "Yes", "No") != knn_test_data$Private)
misrate2

## [1] 0.06859206
```

Explanation: The training misclassification error rate is 0.04, and the test misclassification error rates for this model is 0.0686. The test misclassification error rate for this model is smaller than the rate of KNN classification model at K=20. Therefore, this model is better than KNN classifiers.