Analytics 512: Additional Homework Problems

```
set.seed(134)
```

Chapter 2

Problem 1

The built in data set **cars** gives the speed in mph and stopping distance in feet for 50 cars. The data were recorded in the 1920s.

a) Make a scatterplot of stopping distance against speed and describe the relation.

b) Fit a linear regression model to the data, plot the regression line in the scatterplot, and compute the RMS of the residuals.

c) Find the Bayes predictor (based on the data) of stopping distance is a function of speed, add the predictions to the data frame, and plot the predictions in the scatterplot.

d) Find the RMS of the prediction errors for the Bayes predictor.

Problem 2

You are given a data frame with numerical columns x and y. Each value of x appears exactly once.

a) What is the Bayes predictor of y is a function of x? Explain your answer.

b) What can you say about the Bayes predictor of x as a function of y? Explain your answer.

Problem 3

Look at exercise 2.4.10 and work through parts a and b before working on this problem.

We want to examine whether the variable **nox** can be predicted from the variables **rad** and **chas**.

a) Explain what these variables mean. Do you think it is reasonable to expect that **nox** can be predicted from the other two variables?

b) How many observations are there for each combination of the variables **rad** and **chas**?

c) Set up the Bayes predictor for **nox** as a function of **rad** and **chas**. The result should be a data frame with three columns.

d) Use the Bayes predictor to predict **nox** from **rad** and **chas** for all operations in the data set and compute the root mean square (RMS) error of these predictions.

Problem 4

*Refer to the class discussion of 1/15 or 1/16, specifically implementation of a k-nearest neighbor method for classifying digits. Use the **MNIST** data that are available in Canvas.* Here is the code that was used in class.

```r
load("../Data/mnist_all.RData")

# predict a digit from the MNIST training set from the most frequent digit
# among the 100 closest neighbors in the training set

myclosest = function(mydigit){
  digit.dist = function(j){
    return(sqrt(mean((test$x[mydigit,] - train$x[j,])^2) ) )
  }
  mnist.distances = sapply(1:60000,FUN = digit.dist)
  myclosest = head(order(mnist.distances),100)
  mytable <- table(train$y[myclosest])
  myindex = which.max(mytable)
  return(as.numeric(names(mytable[myindex])))
}

# Try it. Prediction and actual value of digit 234 in the test set

c(test$y[234], myclosest(234))
```

```
## [1] 8 8
```

a) Modify the function **myclosest()** so that it uses exactly **k** neighbors instead of 100 to classify a test digit. The new function should have two arguments, namely **mydigit** and **k**.

b) Demonstrate the modified function by trying to classify a test digit of your choice. Find a value of k such that the classification is correct and another value of $k < 1000$ such that the classification of the same test digit is incorrect.

Problem 5

*Refer to the class discussion of 1/15 or 1/16, specifically implementation of a k-nearest neighbor method for classifying digits. Use the **MNIST** data that are available in Canvas.*

a) Modify the function **myclosest()** so that it uses exactly **k** neighbors instead of 100 to classify a test digit. The new function should have two arguments, namely **mydigit** and **k**.

b) If $k = 1$, misclassification happens quite frequently. Demonstrate this by finding the smallest index j larger than K such that the test digit j is not classified correctly, where K is 300 times the day in your birth date (e.g. $K = 4500$ if you were born on May 15). Also make a plot of this digit.

c) For the test digit that you identified in (b), find an approximate range of ks such that nearest neighbor classification with k digits gives the right answer. The range should be in multiples of 10.

Problem 6

This problem uses the Shiny app at https://keeganhines.shinyapps.io/bias_variance/ . Before working on this problem, load the app, read the explanation, play with the slider and the "Generate New Data" button, and answer the questions at the bottom of the page ("Check your understanding") for yourself or discuss them with others.

Model complexity = degree of the polynomial that is being fitted.

a) Make 10 different simulations with model complexity = 1. Compute the average Residual SSE and find the approximate range of the highest order coefficient for these 10 simulations. This is a measure for the baseline variance for a low complexity model.

b) Make 10 different simulations with model complexity = 10. Compute the average Residual SSE. Which coefficient has the largest range in this case? What is that range? This is a measure for the variance for a high complexity model.

c) How do your results illustrate the bias - variance trade-off? The answer should be a short paragraph.

d) For which model complexity between 1 and 15 do you typically obtain a curve which is most similar and overall close to the unknown curve that is to be estimated? Try multiple simulation for several different model complexities, summarize what you see, and explain your answer. Pictures or numerical results are not required.

Chapter 3

Problem 7

Use the following **R** code to make artificial data for a problem with $n = 100$ observations and $p = 3$ predictors.

```
set.seed(1091)
n <- 100
p <- 3
X <- matrix(runif(n*p), ncol = p)
beta <- c(2,.5,-3)
sigma = 1
y <- 1 + X%*%beta + sigma*rnorm(n)
mydf.6 <- as.data.frame(X)
names(mydf.6) <- c("x1", "x2", "x3")
mydf.6$y <- y
```

Construct the linear model $y \sim x_1 + x_2 + x_3$. Then compute SST and SSE and make a histogram the standardized residuals, using only the output of the **summary()** function.

Problem 8

A linear model of the form $y \sim \beta_0 + \beta_1 + \sum + \beta_p x_p$ has been fitted for n observations and the residuals $r_i = y_i - \hat{y}_i$ have been obtained. Now you want to fit a linear model with the same predictors to the residuals, i.e. $r \sim \gamma_0 + \gamma_1 x_1 + \cdots + \gamma_p x_p$. Prove that $\hat{\gamma}_0 = \hat{\gamma}_1 = \cdots = \hat{\gamma}_p = 0$. *Use the hat matrix.*

Problem 9

Consider the built-in data set **cars** (see problem 1). Fit a linear regression model to the data. Then use **R** to make 90% prediction intervals for cars traveling at speeds of 6mph and of 21 mph. Comment on the result.

Problem 10

Consider the built-in data set **cars** (see problem 1). Fit a linear regression model to the data. What are the three observations with the largest standardized residuals (in magnitude)? What are their leverages? Where are the three observation with the largest leverage? What are their standardized residuals? *Use the **R** function **influence.lm**.*

Problem 11

The data for this exercise are in the UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/index.php. We shall use the concrete compressive strength data which may be found here:

http://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength

Each row of the data set contains measurements for a single high performance concrete sample. Each column contains measurements of a physical property. The goal is to predict compressive strength (column 9) from the other quantities.

a) Import the data into your **R** workspace and change all variable names to something simpler.

b) Find the ranges of all predictors.

c) Fit a linear model that uses all predictors. What is R^2? What is the residual standard error? How does that compare to the range of the response? Which predictors are statistically significant?

d) Which observation has the largest leverage? For this observation, some of the variables are at or near the extremes of their range. Identify these variables. *Use* ***influence.lm()*** *to find the leverages.*

e) Fit a linear model that does not use the two least significant predictors. How does the adjusted R^2 change? List some of the major changes in the new model.

Version of 1/27/19