Module 4: Recommended Exercises

TMA4268 Statistical Learning V2022

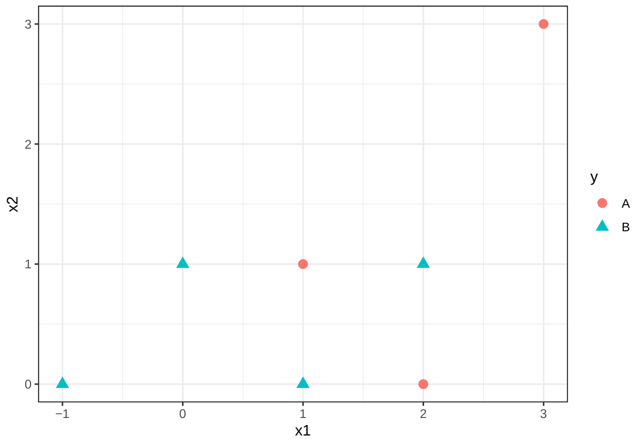
Emma Skarstein, Daesoo Lee, Stefanie Muff
Department of Mathematical Sciences, NTNU

January 31, 2022

Problem 1: KNN (Exercise 2.4.7 in ISL textbook slightly modified)

The table and plot below provides a training data set consisting of seven observations, two predictors and one qualitative response variable.

```
##   x1 x2 y
## 1  3  3 A
## 2  2  0 A
## 3  1  1 A
## 4  0  1 B
## 5 -1  0 B
## 6  2  1 B
## 7  1  0 B
```



We wish to use this data set to make a prediction for Y when $X_1 = 1, X_2 = 2$ using the K-nearest neighbors classification method.

a)

Calculate the Euclidean distance between each observation and the test point, $X_1 = 1, X_2 = 2$.

b)

Use $P(Y = j \mid X = x_0) = \frac{1}{K}\sum_{I\in\mathcal{N}_0} I(Y = j)$ to predict the class of Y when $K = 1$, $K = 4$ and $K = 7$. Why is $K = 7$ a bad choice?

c)

If the Bayes decision boundary in this problem is highly non-linear, would we expect the best value for K to be large or small? Why?

Problem 2: Bank notes and LDA (with calculations)

To distinguish between genuine and fake bank notes measurements of length and diagonal of an image part of the bank notes have been made. For 1000 bank notes (500 genuine and 500 false) this gave the following values for the mean and the covariance matrix (using unbiased estimators), where the first value is the length of the bank note.

Genuine bank notes:

$$\bar{\mathbf{x}}_G = \begin{bmatrix} 214.97 \\ 141.52 \end{bmatrix} \text{ and } \hat{\boldsymbol{\Sigma}}_G = \begin{bmatrix} 0.1502 & 0.0055 \\ 0.0055 & 0.1998 \end{bmatrix}$$

Fake bank notes:

$$\bar{\mathbf{x}}_F = \begin{bmatrix} 214.82 \\ 139.45 \end{bmatrix} \text{ and } \hat{\boldsymbol{\Sigma}}_F = \begin{bmatrix} 0.1240 & 0.0116 \\ 0.0116 & 0.3112 \end{bmatrix}$$

a)

Assume the true covariance matrix for the genuine and fake bank notes are the same. How would you estimate the common covariance matrix?

b)

Explain the assumptions made to use linear discriminant analysis to classify a new observation to be a genuine or a fake bank note. Write down the classification rule for a new observation (make any assumptions you need to make).

c)

Use the method in b) to determine if a bank note with length 214.0 and diagonal 140.4 is genuine or fake. You can use R to perform the matrix calculations.

R-hints:

```
# inv(A)
solve(A)
# transpose of vector
t(v)
# determinant of A
det(A)
# multiply vector and matrix / matrix and matrix
v%*%A
B%*%A
```

d)

What is the difference between LDA and QDA? Use the classification rule for QDA to determine the bank note from c). Do you obtain the same result? You can use R to perform the matrix calculations.

Hint: the following formulas might be useful.

$$A^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$|A| = det(A) = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

Problem 3: Odds (Exercise 4.7.9 in ISL textbook)

This problem has to do with *odds*.

a)

On average, what fraction of people with an odds of 0.37 of defaulting on their credit card payment will in fact default?

b)

Suppose that an individual has a 16% chance of defaulting on her credit card payment. What are the odds that she will default?

Problem 4: Logistic regression (Exercise 4.7.6 in ISL textbook)

Suppose we collect data for a group of students in a statistics class with variables x_1 = hours studied, x_2 = undergrad grade point average (GPA), and Y = receive an A. We fit a logistic regression and produce estimated coefficient, $\hat{\beta}_0 = -6, \hat{\beta}_1 = 0.05, \hat{\beta}_2 = 1$.

a)

Estimate the probability that a student who studies for 40 h and has an undergrad GPA of 3.5 gets an A in the class.

b)

How many hours would the student in part a) need to study to have a 50% probability of getting an A in the class?

Problem 5: Sensitivity, specificity, ROC and AUC

We have a two-class problem, with classes 0=non-disease and 1=disease, and a method $p(x)$ that produces probability of disease for a covariate x. In a population we have investigated N individuals and know the predicted probability of disease $p(x)$ and true disease status for these N.

a)

We choose the rule $p(x) > 0.5$ to classify to disease. Define the sensitivity and the specificity of the test.

b)

Explain how you can construct a reciever operator curve (ROC) for your setting, and why that is a useful thing to do. In particular, why do we want to investigate different cut-offs of the probability of disease?

c)

Assume that we have a competing method $q(x)$ that also produces probability of disease for a covariate x. We get the information that the AUC of the $p(x)$-method is 0.6 and the AUC of the $q(x)$-method is 0.7. What is the definition and interpretation of the AUC? Would you prefer the $p(x)$ or the $q(x)$ method for classification?

Data analysis with R

For the following problems, you should check out and learn how to use the following R functions: `glm()` (`stats` library), `lda()`, `qda()` (`MASS` library), `knn()` (`class` library), `roc()` and `auc()` (`pROC` library).

Problem 6 (Exercise 4.7.10 in ISL textbook - modified)

This question should be answered using the `Weekly` data set, which is part of the `ISLR` package. This data is similar in nature to the `Smarket` data from this chapter's lab, except that it contains $1,089$ weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

a)

Produce numerical and graphical summaries of the `Weekly` data. Do there appear to be any patterns? **R-hint**: Load the data as follows:

```r
data("Weekly")
```

b)

Use the full data set to perform a logistic regression with `Direction` as the response and the five lag variables plus `Volume` as predictors. Use the `summary()` function to print the results. Which of these predictors appears to be of interest?
R-hints: You should use the `glm()` function with the argument `family="binomial"` to make a logistic regression model.

c)

Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.
R-hints: insert the name of your model for `yourGlmModel` in the code below to get the predicted probabilities for "Up", the classified direction and the confusion matrix.

```
glm.probs_Weekly = predict(yourGlmModel, type="response")
glm.preds_Weekly = ifelse(glm.probs_Weekly > 0.5, "Up", "Down")
table(glm.preds_Weekly, Weekly$Direction)
```

d)

Now fit the logistic regression model using a training data period from 1990 to 2008, with `Lag2` as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).
R-hints: use the following code to divide into test and train set. For predicting the direction of the test set, use `newdata = Weekly_test` in the `predict()` function.

```
Weekly_trainID = (Weekly$Year < 2009)
Weekly_train = Weekly[Weekly_trainID,]
Weekly_test = Weekly[!Weekly_trainID,]
```

e)

Repeat d) using LDA.

f)

Repeat d) using QDA.

R-hints: plug in you variables in the following code to perform lda and qda (just replacing `lda` with `qda`).

```
library(MASS)
lda.Weekly = lda(Response~pred1, data=youTrainData)
lda.Weekly_pred = predict(yourModel, newdata=YourTestData)$class
lda.Weekly_prob = predict(yourModel, newdata=YourTestData)$posterior
table(lda.Weekly_pred, YourTestData$Direction)
```

g)

Repeat d) using KNN with $K = 1$.

R-hints: plug in you variables in the following code to perform KNN. The argument `prob=T` will provide the probabilities for the classified direction (which you will need later). When there are ties (same amount of Up and Down for the nearest neighbors), the `knn` function picks a class at random. We use the `set.seed()` function such that we don't get different answers for each time we run the code.

```
library(class)
knn.train = as.matrix(YourTrainData$Lag2)
knn.test = as.matrix(YourTestData$Lag2)

set.seed(123)
yourKNNmodel = knn(train = knn.train, test = knn.test, cl = YourTrainData$Direction, k=YourValueOfK, pr
table(yourKNNmodel, YourTestData$Direction)
```

h)

Use the following code to find the best value of K. Report the confusion matrix and overall fraction of correct predictions for this value of K.

```
#knn error:
K=30
knn.error = rep(NA,K)

set.seed(234)
for(k in 1:K){
  knn.pred = knn(train = knn.train, test = knn.test, cl = Weekly_train$Direction, k=k)
  knn.error[k] = mean(knn.pred != Weekly_test$Direction)
}
knn.error.df = data.frame(k=1:K, error = knn.error)
ggplot(knn.error.df, aes(x=k, y=error))+geom_point(col="blue")+geom_line(linetype="dotted")
```

i)

Which of these methods appear to provide the best results on this data?

j)

Plot the ROC curves and calculate the AUC for the four methods (using your the best choice for KNN). What can you say about the fit of these models?

R-hints:

- For KNN you can use `knn(...,prob=TRUE)` to get the probability for the classified direction. Note that we want $P(Direction = Up)$ when plotting the ROC-curve, so we need to modify the probabilties returned from the `knn` function.

```
#get the probabilities for the classified class
yourKNNProbs = attributes(yourKNNmodel)$prob

# since we want the probability for Up, we need to take 1-p for the elements that gives probability for
down= which(yourKNNmodel == "Down")
yourKNNProbs[down] = 1-yourKNNProbs[down]
```

- Use the following code to produce ROC-curves:

```
#install.packages("plotROC")
#install.packages("pROC")
library(pROC)
library(plotROC)

yourRoc = roc(response = Weekly_test$Direction, predictor = yourModelsPredictedProb, direction = "<")
#you can use this function for all your methods and plot them using plot(yourRoc)

#or use ggplot2
dat = data.frame(Direction = Weekly_test$Direction, glm = yourGlmProbs,
                 lda = yourLDAProbs[,2], qda = yourQDAProbs[,2], knn = yourKNNProbs)
dat_long = melt_roc(dat, "Direction", c("glm", "lda", "qda", "knn"))
ggplot(dat_long, aes(d = D, m = M, color = name)) + geom_roc(n.cuts = F) +
  xlab("1-Specificity") + ylab("Sensitivity")
#glm is very similar to lda, so the roc-curve for glm is not shown.


#AUC: yourAUC = auc(yourRoc)
```