Assignment 3: Generalized Additive Models and Unsupervised Learning

BUAD 5082 Section 1 – Spring 2018

---

1. Objectives

The purpose of this assignment is to exercise recently developed skills in GAMS and Unsupervised Learning.

2. What You Will Need

- Access to a Windows computer with a recent version of R installed.

3. What You Will Hand In

Submit your script as Assignment3.R via Blackboard – Assignment3.

4. Due Date

Monday April 9th, 2018 just before midnight.

5. Note on Collaboration

This is a Category C individual assignment.

6. Preliminaries:

To get set up for the lab, follow these steps:

1. As the first statement in your script file, enter rm(list=ls())
2. Your response to each question in the assignment should begin with the following three comment lines, where *n* is the question number:

   ####################
   #### Question *m*
   ####################

3. I should be able to run your script on my computer without errors or interruptions. For this to happen, you must:
   a. Avoid entering file path information…my files will be located in a different location that yours, and so your code will fail on my machine. Instead, always refer only to files in your working directory.
   b. Include require() or library() commands for all required packages
   c. Do not include install.packages() functions (so exclude entirely or comment them out)
   d. Do not use functions like file.choose(), fix(),edit(), or q()
4. Do not create console output other than what is asked of you explicitly. For example, in your final script, remove any statements that you used to verify the contents or structure of data that were not requested.

7. Tasks:

Question 1: PCA (33%)

In Section 10.2.3, a formula for calculating PVE was given in Equation 10.8. We also saw that the PVE can be obtained using the sdev output of the prcomp() function.

On the USArrests data, calculate PVE in two ways:

(a) Using the sdev output of the prcomp() function, as was done in Section 10.2.3.

(b) By applying Equation 10.8 directly. That is, use the prcomp() function to compute the principal component loadings. Then, use those loadings in Equation 10.8 to obtain the PVE.

These two approaches should give the same results.

Hint: You will only obtain the same results in (a) and (b) if the same data is used in both cases. For instance, if in (a) you performed prcomp() using centered and scaled variables, then you must center and scale the variables before applying Equation 10.3 in (b).

Question 2: Clustering (33%)

The text mentions that the use of correlation-based distance and Euclidean distance as dissimilarity measures for hierarchical clustering. It turns out that these two measures are almost equivalent.

For example, if each observation has been centered to have mean zero and standard deviation one, and if we let r_{ij} denote the correlation between the i^{th} and j^{th} observations, then the quantity $1 - r_{ij}$ is proportional to the squared Euclidean distance between the i^{th} and j^{th} observations.

On the USArrests data, show that this proportionality holds.

Hints:

1. The Euclidean distance can be calculated using the dist() function, and correlations can be calculated using the cor() function.
2. The lower.tri() function may come in handy.

Question 3: PCA and Clustering (34%)

In this question, you will generate simulated data, and then perform PCA and *K*-means clustering on the data.

a) Generate a simulated data set called x.values with 20 observations in each of three classes (i.e. 60 observations total), and 50 variables using the following code:

```
n <- 20 # the number of samples per class
p <- 50 # the number of variables
x1 <- matrix(rnorm(n*p), nrow=n, ncol=p)
x2 <- matrix(rnorm(n*p), nrow=n, ncol=p)
x3 <- matrix(rnorm(n*p), nrow=n, ncol=p)
for(i in 1:n){
      x1[i,] <- x1[i,] + rep(1, p)
      x2[i,] <- x2[i,] + rep(-1, p)
      x3[i,] <- x3[i,] + c(rep(+1, p / 2), rep(-1, p / 2))
}
x.values <- rbind(x1, x2, x3)
```

b) Create an vector named y.values that represents class labels for the observations in x.values using the following code:

```
y.values <- c(rep(1,n), rep(2,n), rep(3,n)) # the "true" class labels of the points.
```

c) Perform PCA on the 60 observations and plot the first two principal component score vectors. Use a different color to indicate the observations in each of the three classes.

d) Perform *K*-means clustering of the observations with $K = 3$. How well do the clusters that you obtained in *K*-means clustering compare to the true class labels?

Hint: You can use the table() function in R to compare the true class labels to the class labels obtained by clustering. Be careful how you interpret the results: K-means clustering will

arbitrarily number the clusters, so you cannot simply check whether the true class labels and clustering labels are the same.

e) Perform K-means clustering with K = 2. Describe your results.

f) Now perform *K*-means clustering with $K = 4$, and describe your results.

g) Now perform *K*-means clustering with $K = 3$ on the first two principal component score vectors, rather than on the raw data. That is, perform K-means clustering on the 60×2 matrix of which the first column is the first principal component score vector, and the second column is the second principal component score vector. Comment on the results.

h) Using the scale() function, perform *K*-means clustering with $K = 3$ on the data after scaling each variable to have standard deviation one. How do these results compare to those obtained in c)? Explain.