Homework 3
Introduction to Data Analysis and Mining
Spring 2018
CSCI-B 365

Yining Wang

March 20, 2018

Directions

Please follow the syllabus guidelines in turning in your homework. I am providing the LaTeX of this document too. This homework is due Friday, March 16, 2018 10:00p.m. **OBSERVE THE TIME**. Absolutely no homework will be accepted after that time. All the work should be your own.

All the work herein is solely mine.

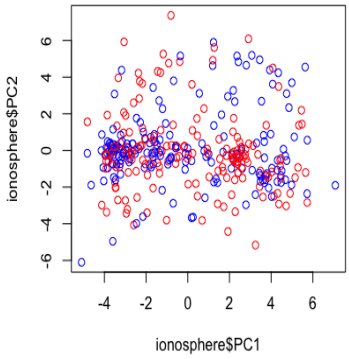
Problem 1 [30 points]

In this question, you will first perform principal component analysis (PCA) over Ionosphere Data Set and then cluster the reduced data using your k-means program (C_k) from previous homework. You are allowed to use R packages for PCA and ignore the class variables (35th variable) while performing PCA. Answer the questions below:

1.1) Perform PCA over Ionosphere data set and make a scatter plot of PC1 and PC2 (the first two principal components). Are PC1 and PC2 linearly correlated?

Listing 1: Sample R Script With Highlighting

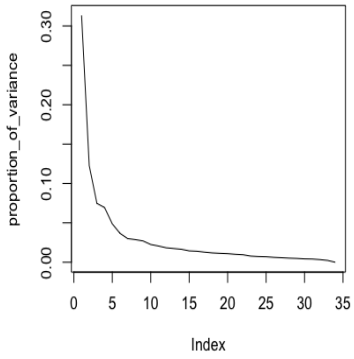
```
#install.packages("data.table")
library(data.table)
#install.packages("curl")
mydata <- fread("https://archive.ics.uci.edu/ml/machine-learning-databases/ionosphere/ionosp
mydata$V2=NULL
pca=prcomp(mydata[, 1:33], scale. = TRUE, center = TRUE)
ionosphere = as.data.table(pca$x)
plot(ionosphere$PC1, ionosphere$PC2, col=c("red", "blue"))
#The scatterplot shows that PC1 and PC2 are not linearly correlated.
```



1.2) There are three methods to pick the set of principle components: (1) In the plot where the curve bends; (2) Add the percentage variance until total 75% is reached $(70 - 90\%)$ (3) Use the components whose variance is at least one. Show the components selected in the Ionosphere data if each of these is used.

Listing 2: Sample R Script With Highlighting

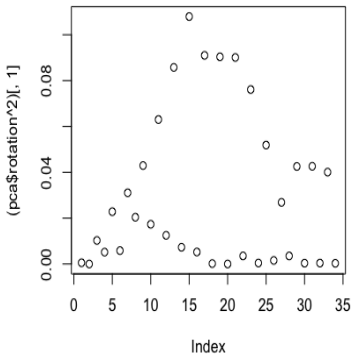
```
#install.packages("data.table")
library(data.table)
#install.packages("curl")
mydata <- fread("https://archive.ics.uci.edu/ml/machine-learning-databases/ionosphere/ionosp
pca=prcomp(mydata[, 1:34], center = TRUE)
ionosphere = as.data.table(pca$x)

############################(1)#####################
variance=pca$sdev^2
proportion_of_variance = variance/sum(variance)
plot(proportion_of_variance, type = "l")
#From the plot, it seems like the first 4 are selected.

############################(2)#####################
which(cumsum(proportion_of_variance) >= 0.75)[1]
#By looking at Cumulative Proportion, the first 9 componenets are selected.


############################(3)#####################
sum(pca$sdev^2 >= 1)
#By looking at Standard deviation, the first 2 components are selected.
```



1.3) Observe the loadings using prcomp() or princomp() functions in R and discuss loadings in PCA?i.e., how are principal components and original variables related?

Listing 3: Sample R Script With Highlighting

```r
#install.packages("data.table")
library(data.table)
#install.packages("curl")
mydata <- fread("https://archive.ics.uci.edu/ml/machine-learning-databases/ionosphere/ionosp
pca=prcomp(mydata[, 1:34], center = TRUE)

plot((pca$rotation^2)[,1])

# Squared factor loadings indicate what percentage of
# the variance in an original variable is
# explained by a factor. From the plot, it seems
# like that all the different variables all
# contribute to the first component.
```



1.4) Perform dimensionality reduction over Ionosphere data set with PCA. Keep 90% of variance after PCA and reduce Ionosphere data set and call this data Δ_R. Cluster Δ_R using your k-means program

from previous assignment and report the total error rates for $k = 2, \ldots, 5$ for 20 runs each. Plots are generally a good way to convey complex ideas quickly, i.e., box plots, whisker plots. Discuss your results, i.e how did PCA affect performance of k-means clustering.

Listing 4: Sample R Script With Highlighting

```
#install.packages("data.table")
library(data.table)
#install.packages("curl")
mydata <- fread("https://archive.ics.uci.edu/ml/machine-learning-databases/ionosphere/ionosp

############Kmeans from HW3###########
#the distance function
distance <- function(cent, dp){
  d = 0
  for(i in 1:length(cent)){
    d = d + (cent[i] - dp[i])^2
  }
  return(sqrt(d))
}
scalar_product <- function(cents_old, cents_new){
  sp = 0
  k = nrow(cents_new)
  for (i in 1: k){
    sp = sp + distance(cents_old[i,], cents_new[i,])
  }
  return(sp / k)
}
# the last column of input dataset must be class
kmeans <- function(input_dataset, d=distance, k, tau){
  dataset = input_dataset[, -ncol(input_dataset)]
  classes = input_dataset[, ncol(input_dataset)]
  dimension = ncol(dataset)
  dataset_min = min(dataset)
  dataset_max = max(dataset)
  #initialization of centroids
  centroids = matrix(runif(k*dimension, min = dataset_min,
                           max = dataset_max),
                     nrow = k, ncol = dimension)
  iteration = 0
  repeat{
    iteration = iteration+1
    print(iteration)
    #First row in every matrix is not used
    B <- lapply(1:k, function(x) matrix(rep(0,dimension), nrow=1,
                                        ncol=dimension))
    goods_and_bads=matrix(rep(0,2*k), nrow=k, ncol=2)

    #Assign data point to nearest centroid
    for (j in 1: nrow(dataset)){
      dp = dataset[j,]
      min = d(centroids[1,], dp)
      nearest_centroid = 1
      for (t in 2 : k){
        dist = d(centroids[t,], dp)
        if(dist < min) {
          min = dist
          nearest_centroid = t
```

```
        }
      }
      B[[nearest_centroid]] = rbind(B[[nearest_centroid]], dp)
      class = classes[j]
      #print(class)
      if(class == "g"){
        goods_and_bads[nearest_centroid,1] =
          goods_and_bads[nearest_centroid,1] + 1

      }else if(class == "b"){
        goods_and_bads[nearest_centroid,2] =
          goods_and_bads[nearest_centroid, 2] +1
      }
    }
    #error rate
    #goods_and_bads; k*2 matrix
    error_rate = 0
    for(i in 1:k){
      total = goods_and_bads[i,1]+goods_and_bads[i,2]
      if (goods_and_bads[i,1] > goods_and_bads[i,2]){
        error_rate = error_rate + (goods_and_bads[i,1]/total)
      }else{
        error_rate = error_rate + (goods_and_bads[i,2]/total)
      }
    }


    #Get size of centroid & Update centroid with average
    new_centroids = matrix(rep(0, k*dimension), nrow = k, ncol = dimension)
    for (j in 1:k){
      if (nrow(B[[j]]) == 1) {
        new_centroids[j,] = runif(dimension, min=dataset_min, max=dataset_max)
      } else if (nrow(B[[j]]) == 2){
        new_centroids[j,] = B[[j]][2,]
      } else {
        new_centroids[j,] = colMeans(B[[j]][-1,])
      }
    }
  return(error_rate)
  }
}
##############################################################################



pca=prcomp(mydata[, 1:34], center = TRUE)
plot((pca$rotation^2)[,1])
pca_data = as.data.table(pca$x)
variance=pca$sdev^2
proportion_of_variance = variance/sum(variance)
t = which(cumsum(proportion_of_variance) >= 0.9)[1]
##18
class = mydata[,35]
dataset_R=pca_data[,1:(t+1)]
dataset_R[,t+1] = mydata[,35]

kmeans(dataset_R, distance, 2, 1)
#With PCA, it seems like that only the first 18 componenets are picked.
```

```
#From 18 to 33, dimensionality is decreased.
#The clustering increases the accuracy and decreases the error rates.
```

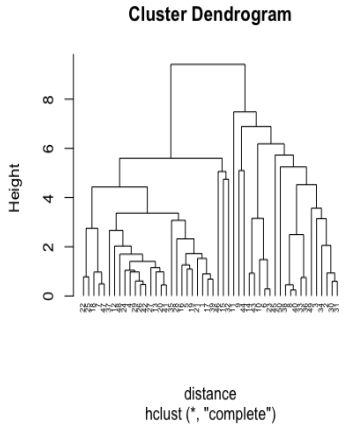
Problem 2 [30 points]

Randomly choose 50 points from Ionosphere data set (call this data set I_{50}) and perform hierarchical clustering. You are allowed to use R packages for this question. (Ignore the class variable while performing hierarchical clustering.)

2.1) Using hierarchical clustering with complete linkage and Euclidean distance cluster I_{50}. Give the dendrogram.

Listing 5: Sample R Script With Highlighting

```
#install.packages("data.table")
library(data.table)
#install.packages("curl")
mydata <- fread("https://archive.ics.uci.edu/ml/machine-learning-databases/ionosphere/ionosp
rows_50 = sample(1:nrow(mydata), 50)
I50 = mydata[rows_50,]

distance <- dist(I50[,1:34], method = "euclidean")
hieclust = hclust(distance, method = "complete", members = NULL)
plot(hieclust, cex = 0.5, hang = -1)
```



2.2) Cut the dendrogram at a height that results in two distinct clusters. Calculate the error-rate.

Listing 6: Sample R Script With Highlighting

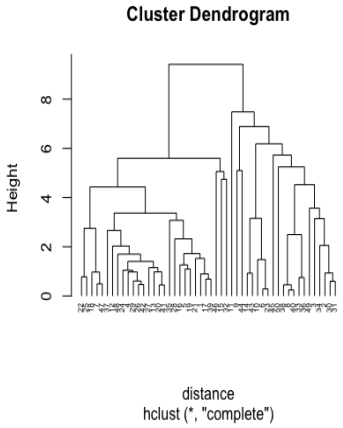
```
#install.packages("data.table")
library(data.table)
#install.packages("curl")
mydata <- fread("https://archive.ics.uci.edu/ml/machine-learning-databases/ionosphere/ionosp
rows_50 = sample(1:nrow(mydata), 50)
I50 = mydata[rows_50,]

distance <- dist(I50[,1:34], method = "euclidean")
hieclust = hclust(distance, method = "complete", members = NULL)
clusterCut <- cutree(hieclust, k=2)
#table(clusterCut)
```

```r
#error rate

res = table(clusterCut, I50$V35)
#print(res)
error_rates = 0
for (i in c(1:2)){
  error_rates = error_rates + min(res[i,])/sum(res[i,])
}
print(error_rates)
#The error rate is 0.3958333.
```

2.3) First, perform PCA on I_{50} (Keep 90% of variance). Then hierarchically cluster the reduced data using complete linkage and Euclidean distance. Report the dendrogram.

Listing 7: Sample R Script With Highlighting

```r
#install.packages("data.table")
library(data.table)
#install.packages("curl")
mydata <- fread("https://archive.ics.uci.edu/ml/machine-learning-databases/ionosphere/ionosp
rows_50 = sample(1:nrow(mydata), 50)
I50 = mydata[rows_50,]


#pca
pca=prcomp(I50[, 1:34], center = TRUE)
pca_data = as.data.table(pca$x)

variance=pca$sdev^2
proportion_of_variance = variance/sum(variance)
t = which(cumsum(proportion_of_variance) >= 0.9)[1]

#hierarchically cluster the reduced data using complete linkage and Euclidean distance
dataset_R=pca_data[,1:t]
distance <- dist(dataset_R[,1:t], method = "euclidean")
hieclust = hclust(distance, method = "complete", members = NULL)

plot(hieclust, cex = 0.5, hang=-1)
```



2.4) Cut the dendrogram at a height that results in two distinct clusters. Give the error-rate. Discuss your findings, i.e., how did PCA affect hierarchical clustering results?

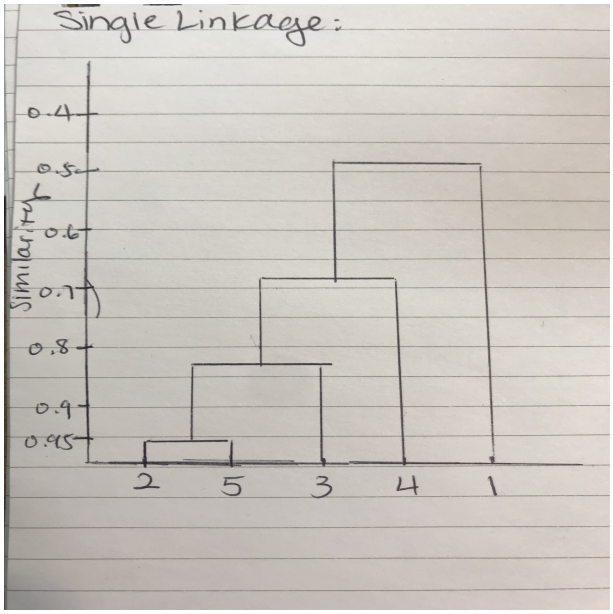
Listing 8: Sample R Script With Highlighting

```
#install.packages("data.table")
library(data.table)
#install.packages("curl")
mydata <- fread("https://archive.ics.uci.edu/ml/machine-learning-databases/ionosphere/ionosp
rows_50 = sample(1:nrow(mydata), 50)
I50 = mydata[rows_50,]

pca=prcomp(I50[, 1:34], center = TRUE)
pca_data = as.data.table(pca$x)
variance=pca$sdev^2
proportion_of_variance = variance/sum(variance)
t = which(cumsum(proportion_of_variance) >= 0.9)[1]
dataset_R=pca_data[,1:t]
distance <- dist(dataset_R[,1:t], method = "euclidean")
hieclust = hclust(distance, method = "complete", members = NULL)

#Cut the dendrogram
clusterCut <- cutree(hieclust, k=2)

res = table(clusterCut, I50$V35)
#print(res)

#error rates
error_rates = 0
for (i in c(1:2)){
  error_rates = error_rates + min(res[i,])/sum(res[i,])
}
print(error_rates)
#The error rate is 0.6883469.

#In general, with the PCA, reduction of dimensionality should
#result in decrease of the error rate.
#However, in this case, orginally there's 50x34 data, which
#is not a large dataset and with dimensional reduction, there's
#even less data. Therefore, the result could be worse.
```

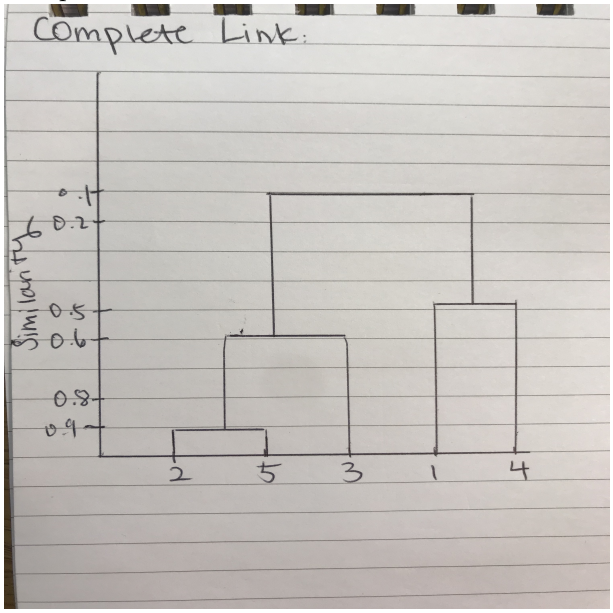
Problem 3 [20 points]

From textbook, Chapter 8 exercises 16, 18 and 30 (Pages 563-566)

1. 16.
   Single Link:



Complete Link:



2. 18.
   Neither Ward's method nor bisecting K-means represent a local minimum because unlike the regular K-means, none of them has stepwise refinement that would produce a local minimum nor a global minimum. A stepwise refinement have to be added to Ward's method or to bisecting K-means in order for it to produce a local minimum.
   On the other hand, ordinary K-means is able to produce a local minimum. However, it might not be able to produce a global minimum.

3. 30. (a)
   A set of term clusters defined by the top terms in a document cluster might overlap and some terms might be hidden inside the cluster by the top terms.

On the other hand, the word clusters found by clustering the terms with K-means contains all the terms without overlapping.

(b)
Term clustering could be used to define clusters of documents by taking the documents are have the highest frequency of containing terms in the clusters, for a term cluster.

Extra Credit [10 points]

From textbook, Chapter 8 exercise 12 (Page 562).
(a).
Advantages:
The leader algorithm most of the times is computationally faster since it only compares centroids once.
Also, the leader algorithm is more consistent since it produces the same set of clusters every time.
Disadvantages:
Compare to K-means, the leader algorithm produces less quality clusters because it does not have sum of squared residuals to measure from.
Also, in leader algorithm, the number of clusters resulted from the program cannot be set directly.
(b).
We can improve the algorithm by making sure the program would finish within a finite time once leader is selected. Watch out during randomized approaches because sometimes it can take infinite time.
Also, the distance between points calculated from sampling can be used in better quality ways.
With one run, allow more than one thresholds.

What to Turn-in

Submit a .zip file that includes the files below. Name the .zip file as "usename-section number", i.e., hakurban-B365.

- The *tex and *pdf of the written answers to this document.
- *Rfiles for:
  - R code for problem 1 ("pca1.R").
  - R code for problem 2 ("hierarchical2.R").