Homework 1: Cluster Analysis

Huihang Liu

9/20/2019

1 Preparation

Cluster analysis is used to grouping or segmenting a collection of objects into subsets or "clusters", such that those within each cluster are more closely related to one another than objects assigned to different clusters. In addition, the goal is sometimes to arrange the clusters into a natural hierarchy. Cluster analysis is also used to form descriptive statistics to ascertain whether or not the data consists of a set distinct subgroups, each group representing objects with substantially different properties.

Central to all of the goals of cluster analysis is the notion of the degree of similarity (or dissimilarity) between the individual objects being clustered.

Let's load data.

```
# Import the training and testing data we used
data.train <- read.table('data.txt', header = FALSE)
data.test <- read.table('true_clustering.txt',header = FALSE)$V1
# Calculate the euclidean distance matrix of this data set
dist.data.train <- dist(data.train)
```

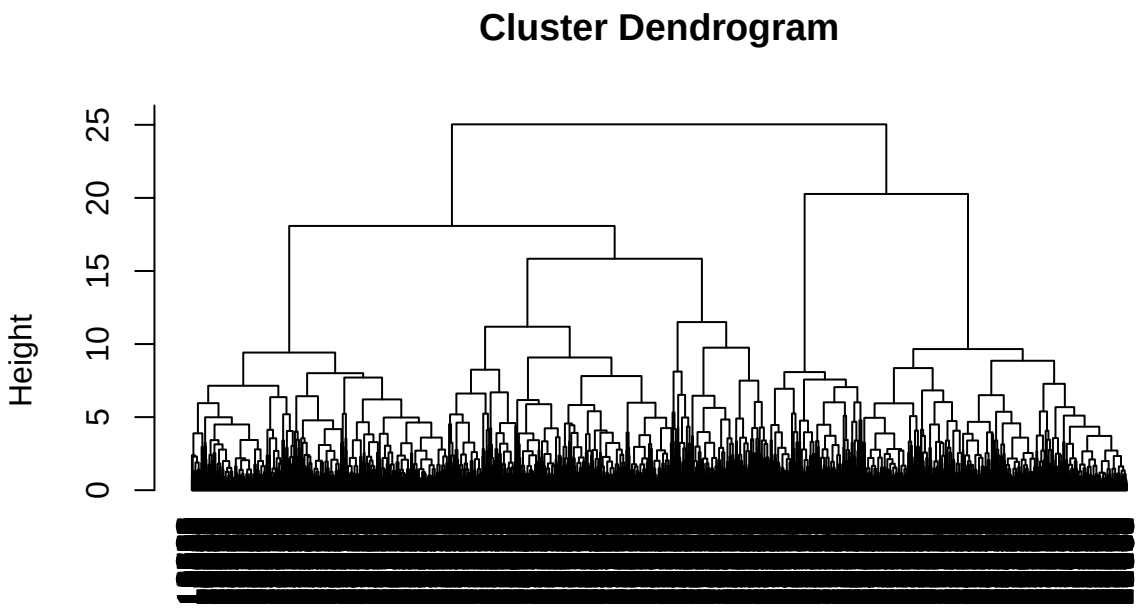
2 Hierarchial Clustering

For the hierarchial clustering methods, the **dendogram** is the main graphical tool for getting insight into a cluster solution. When we use *hclust* or *agnes* to perform a cluster analysis, we can see the dendogram by passing the result of the clustering to the *plot* function.

```
# clusters <- hclust(dist(iris[, 3:4]))
fit.hclust <- hclust(dist.data.train)
```

which generates the following dendrogram.

```
plot(fit.hclust, hang = -1)
```



We can see from the figure that the best choices for total number of clusters are either 4 or 5:

To do this, we can cut off the tree at the desired number of clusters using cutree.

```
cluster.hclust <- cutree(fit.hclust, 5)
```

2.1 Number of Clusters: Elbow method

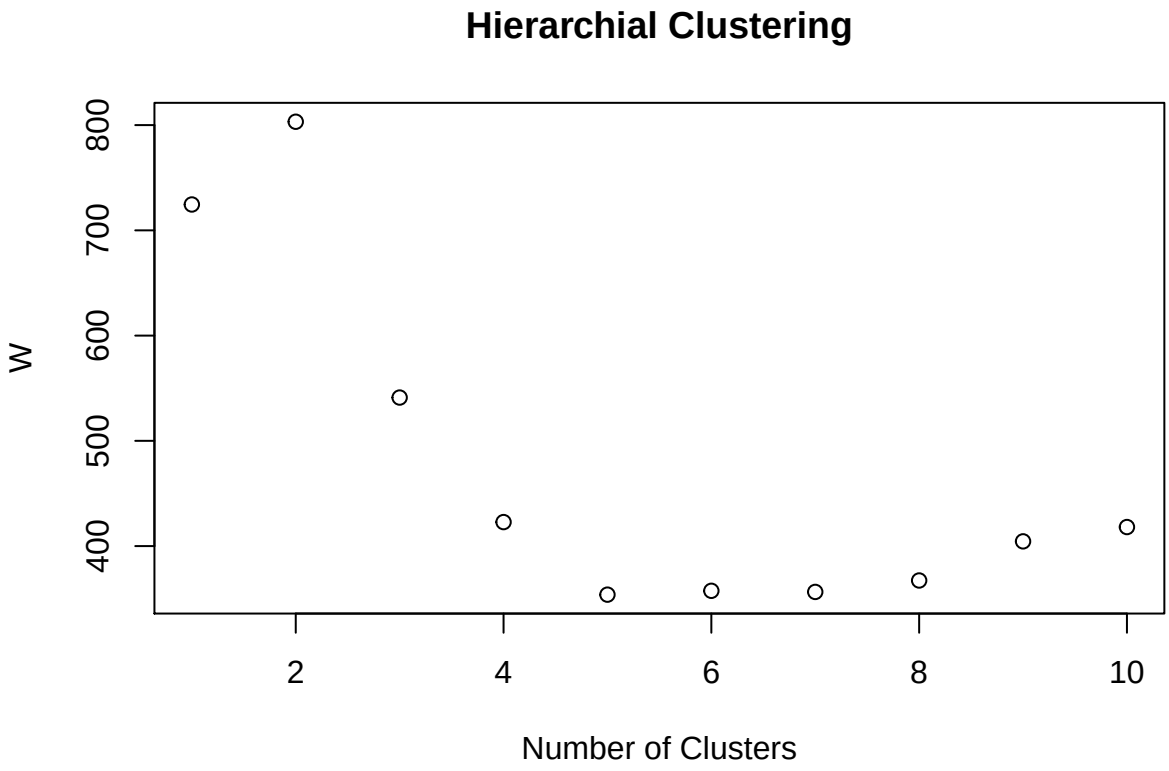
The objective function:

$$W(K) = \sum_{k=1}^{K} \sum_{i \in C_k} \|X_i - M_{C_k}\|^2 \tag{1}$$

We know that $W(K)$ cannot serve as a criterion to choose K. While, elbow method who chooses a k such that the direction of $W(k)$ being changing gives a intuitive solution.

```
x <- data.matrix(data.train)
W <- rep(0,10)
for (K in 1:10){
  cluster.hclust.K <- cutree(fit.hclust, K)
  data.k.dist <- 0
  for (k in 1:K){
    data.k <- x[which(cluster.hclust.K == k),]
    data.k.centroid <- data.matrix(apply(data.k, 2, mean))
    data.k.dist <- data.k.dist + norm(data.k-matrix(rep(data.k.centroid,dim(data.k)[1]), ncol = 3, byrow
  }
  W[K] <- data.k.dist
```

```
}
plot(W, xlab="Number of Clusters", main="Hierarchial Clustering")
```



From the above figure, we know that $K = 5$ is a good choice.

2.2 Evaluating Result

Now, let us compare it with the original species.

```
table(cluster.hclust, data.test)
```

```
##                data.test
## cluster.hclust    1    2    3    4    5
##              1    0    0    0 3051    1
##              2    0    0    0   11 3591
##              3    1 1338    2    0    0
##              4    0    0 1270    0    0
##              5 3505   10    0    0    0
```

Now, let's evaluate the result of cluster analysis by Classification Error Rate

```
classError(cluster.hclust, data.test)$errorRate
```

```
## [1] 0.001956182
```

3 K-means Clustering

For **kmeans**, we can directly use *kmeans function* defined in *r base*.

Note that k is the number of clusters should be determined before running the kmeans.

Before clustering, we'll use Gap Statistic to determine k.

3.1 Gap Statistic

gap statistic is supported by the *cluster* package using function **clusGap**.

```
clusGap(data.train, kmeans, K.max = 10)
```

```
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 639000)

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Clustering Gap statistic ["clusGap"] from call:
## clusGap(x = data.train, FUNcluster = kmeans, K.max = 10)
## B=100 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
##  --> Number of clusters (method 'firstSEmax', SE.factor=1): 5
##            logW    E.logW       gap      SE.sim
##  [1,] 10.344612 10.656816 0.3122040 0.002474801
##  [2,]  9.950233 10.427041 0.4768076 0.018792793
##  [3,]  9.676278 10.247080 0.5708016 0.003358214
##  [4,]  9.279786 10.093887 0.8141010 0.002368877
##  [5,]  9.048170 10.029928 0.9817583 0.004190997
##  [6,]  8.995861  9.971133 0.9752716 0.005539008
##  [7,]  8.953587  9.920649 0.9670619 0.003961268
##  [8,]  8.936610  9.876522 0.9399124 0.008047106
##  [9,]  8.907313  9.838528 0.9312148 0.008031037
## [10,]  8.906797  9.807238 0.9004415 0.004695090
```

clusGap function returns the optimal number of clusters based on the gap statistic in the output, but it spends a lot of time compared with other K determining method. And its result is unstable and even incorrect among different trials.

3.2 Run kmeans and evaluate the result

```
fit.kmeans <- kmeans(data.train, centers = 5)
```

```
classError(fit.kmeans$cluster, data.test)$errorRate
```

```
## [1] 0.003051643
```

4 PAM Clustering

we can use **pam** function to get PAM clustering with a preassigned number of clusters K.

```
fit.pam <- pam(data.train, k = 5, metric = "euclidean")
```

```
classError(fit.pam$clustering, data.test)$errorRate
```

```
## [1] 0.002973396
```

Before clustering, we'll use Average Silhouette Width to determine k.

4.1 Average Silhouette Width

The silhouette width of i is

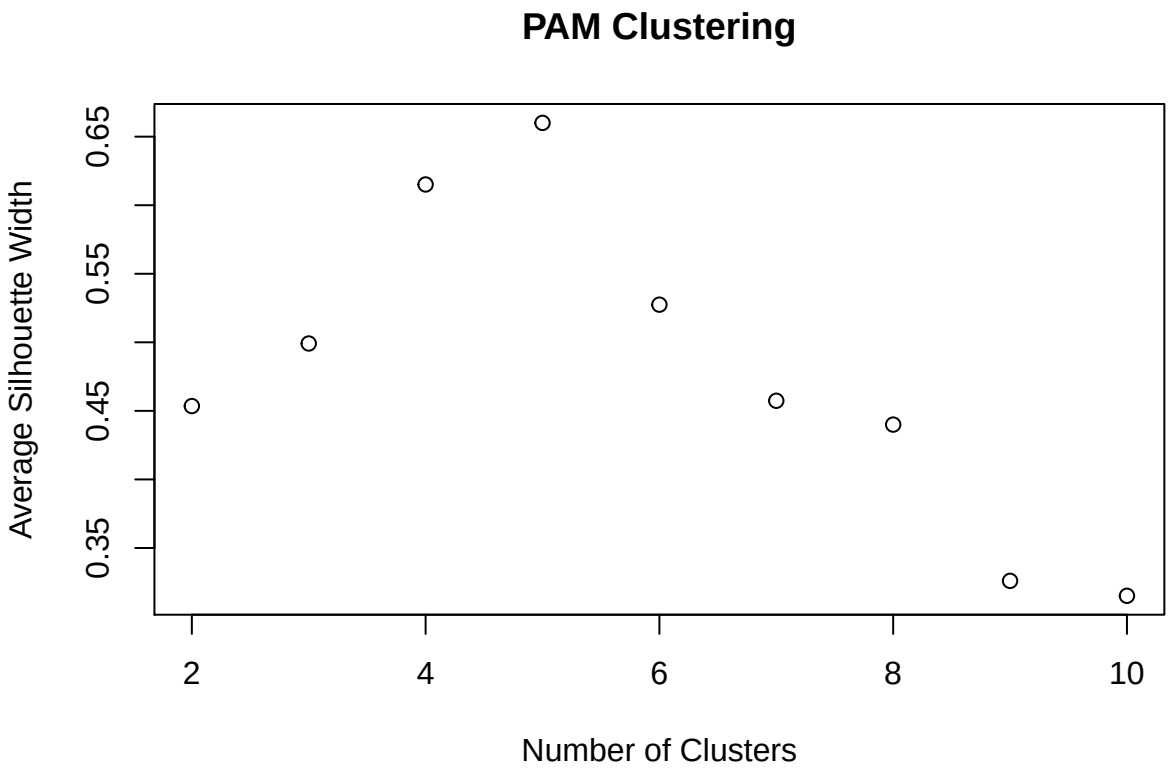
$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{2}$$

where $a(i)$ and $b(i)$ are average distance of i to all other data points in cluster k and minimum average distance of i to all other clusters, respectively.

The Average Silhouette Width is defined as $\frac{1}{n}\sum_{i=1}^{n} s(i)$.

```
pam.sil <- rep(0, 10)
for (K in 2:10){
  fit.pam.K <- pam(data.train, k=K, metric="euclidean")
  pam.sil[K] <- summary(silhouette(fit.pam.K$clustering, dist.data.train))$avg.width
}

plot(2:10, pam.sil[2:10], xlab="Number of Clusters", ylab="Average Silhouette Width", main="PAM Cluster
```



Choosing K to maximize the average silhouette width, we have $K = 5$, which is the same with the result given by elbow method.

5 Summary

In this experiment, I explore three clustering method: **Hierarchial Clustering**, **KMean** and **PAM**. They generate similar clustering result when input a simulated data with 12780 observations and 3 variables. Because of the large amount of clean data, the clustering error rate of **Hierarchial Clustering** and**PAM** are small. While **KMeans** generates a relatively poor results.

Hence, I recommend using **Hierarchial Clustering** and **PAM**.

What's more, I compare three K determining method: **Elbow method**, **Gap Statistic** and **Average Silhouette Width**. All of them selete true K with different computation efficiency. **Elbow method** is the most efficient method, because it requires only some basic matrix calculation. The latter one is **Average Silhouette Width** who requires additional comparison of extreme values, but it's still a efficient method. While the most unstable and computation expensive method is **Gap**, because it generates uniform random variables and runs PCA iteratively. Sometimes, **Gap** gives a wrong result because it doesnot converge in iterations.

So, among those K determining method, I recommend **Elbow method** and **Average Silhouette Width**.