Name : B.M.G.G.K. Rajapaksha
Index No. : S14210

CS4104 – Data Analytics

Assignment - Clustering

(1)

(i) Discuss the characteristics of the data sets for the suitability of the following inter-cluster dissimilarities used in agglomerative hierarchical clustering algorithms.

(a). Single link
(b). Complete link
(c). Average Link

Answer

At the beginning, agglomerative hierarchical clustering algorithms take all observation as an individual cluster. Iteratively it merges clusters until all the data points are merged into a single cluster. Clusters are merged based on the distance between them and different type of linkages are used to determine these pairwise distances. Single linkage takes the minimum distance between the objects of the two clusters as the distance between the two clusters while complete linkage uses maximum distance. Average linkage takes the average of all the distances between objects of the two clusters.

Since there are several linkages, applicability of these are highly depend on the dataset which is going to be analysed. In general, classification is better as the original number of clusters are increased in the considered dataset. Single link can perform well on non-globular data, especially for long chain type cluster containing datasets. But performs poorly in the presence of noise in a given dataset. Single linkage also tends to perform poorly with clusters that are truly elliptical. It also cannot handle the datasets that contains clusters of varying density.

Both average and complete linkage perform well on cleanly separated globular clusters containing data, but have mixed results with otherwise. Average linkage method also performs best with the unequal cluster sizes while complete linkage performs well with the equal cluster sizes. Complete linkage method reacts badly when outliers are introduced into the simulated data. Complete linkage can perform well also with the mix population data which has randomly chosen means and variances. But average linkage relatively gives poor results with these types of datasets.

When analysing the functional data, if the goal of the analysis of the functional data is to identify a few outlying clusters (a few very small clusters) and one large cluster, average linkage is the best. Single linkage and complete linkage are generally not suitable for these type of datasets.

(ii) Use the agglomerative hierarchical clustering algorithm with the **Manhattan distance** function for the following data set with two variables, x and y shown in Table 1 to obtain the best hierarchy of clusters for the above three inter-cluster dissimilarities.

Table 1

objects	X	Y
1	2	9
2	2	5
3	8	4
4	5	8
5	7	5
6	6	4
7	1	2
8	4	9
9	7	3
10	2	4
11	8	5

a.) Draw separate Dendrograms to show the hierarchies obtained for **each** inter-cluster dissimilarity given above. Note that you are required to show all the necessary calculation in determining the best pair of clusters at each iteration.

Answer - Using agglomerative hierarchical clustering algorithms and tools in R

Step 1 - Create and scale the dataset

```
# Create the dataset
dataSet = data.frame(X=c(2,2,8,5,7,6,1,4,7,2,8),Y=c(9,5,4,8,5,4,2,9,3,4,5))


# Scale the dataset
dataSet = as.data.frame(scale(dataSet))
dataSet
```

Output of the scaled dataset

Objects	Parameters	
	X	Y
1	-1.0294761	1.5725095
2	-1.0294761	-0.1150617
3	1.2353713	-0.5369545
4	0.1029476	1.1506167
5	0.8578967	-0.1150617
6	0.4804222	-0.5369545
7	-1.4069506	-1.3807400
8	-0.2745270	1.5725095
9	0.8578967	-0.9588473
10	-1.0294761	-0.5369545
11	1.2353713	-0.1150617

Step 2 - Create the initial distance matrix using Manhattan distance

```
distanceM = dist(dataSet, method = "manhattan")
distanceM
```

Output of the initial distance matrix

	1	2	3	4	5	6	7	8	9	10
2	1.687571									
3	4.374311	2.686740								
4	1.554317	2.398102	2.819995							
5	3.574944	1.887373	0.799367	2.020628						
6	3.619362	1.931791	0.754949	2.065046	0.799367					
7	3.330724	1.643153	3.486108	4.041255	3.530526	2.731158				
8	0.754949	2.442520	3.619362	0.799367	2.819995	2.864413	4.085673			
9	4.418730	2.731158	0.799367	2.864413	0.843786	0.799367	2.686740	3.663780		
10	2.109464	0.421893	2.264847	2.819995	2.309266	1.509898	1.221260	2.864413	2.309266	
11	3.952419	2.264847	0.421893	2.398102	0.377475	1.176842	3.908000	3.197469	1.221260	2.686740

Manhattan distance calculation

$$\boldsymbol{d(i,j) = |X_{i1} - X_{j1}| + |X_{i2} - X_{j2}| \ldots + |X_{ip} - X_{jp}|}$$

Example calculation using scaled X and Y values:

Object	Parameters	
	X	Y
1	-1.0294761	1.5725095
2	-1.0294761	-0.1150617
3	1.2353713	-0.5369545
4	0.1029476	1.1506167
5	0.8578967	-0.1150617
6	0.4804222	-0.5369545
7	-1.4069506	-1.3807400
8	-0.2745270	1.5725095
9	0.8578967	-0.9588473
10	-1.0294761	-0.5369545
11	1.2353713	-0.1150617

Manhattan distance between object 1 and object 2

= |(-1.0294761)-(-1.0294761)|+|1.5725095-(-0.1150617)|

= 0+1.6875712

= 1.6875712

	1	2	3	4	5	6	7	8	9	10
2	1.687571									
3	4.374311	2.686740								
4	1.554317	2.398102	2.819995							
5	3.574944	1.887373	0.799367	2.020628						
6	3.619362	1.931791	0.754949	2.065046	0.799367					
7	3.330724	1.643153	3.486108	4.041255	3.530526	2.731158				
8	0.754949	2.442520	3.619362	0.799367	2.819995	2.864413	4.085673			
9	4.418730	2.731158	0.799367	2.864413	0.843786	0.799367	2.686740	3.663780		
10	2.109464	0.421893	2.264847	2.819995	2.309266	1.509898	1.221260	2.864413	2.309266	
11	3.952419	2.264847	0.421893	2.398102	0.377475	1.176842	3.908000	3.197469	1.221260	2.686740

Step 3 - Look for the least distance and merge those into a cluster

	1	2	3	4	5	6	7	8	9	10
2	1.687571									
3	4.374311	2.686740								
4	1.554317	2.398102	2.819995							
5	3.574944	1.887373	0.799367	2.020628						
6	3.619362	1.931791	0.754949	2.065046	0.799367					
7	3.330724	1.643153	3.486108	4.041255	3.530526	2.731158				
8	0.754949	2.442520	3.619362	0.799367	2.819995	2.864413	4.085673			
9	4.418730	2.731158	0.799367	2.864413	0.843786	0.799367	2.686740	3.663780		
10	2.109464	0.421893	2.264847	2.819995	2.309266	1.509898	1.221260	2.864413	2.309266	
11	3.952419	2.264847	0.421893	2.398102	0.377475	1.176842	3.908000	3.197469	1.221260	2.686740

Step 4 - Re-compute the distance matrix after forming a cluster

i) Using single linkage method

$$\boldsymbol{d(C_{(ij)}, C_k) = \min\{d(C_i, C_k), d(C_j, C_k)}$$

Example calculation using (5, 11) cluster and object 1;

$$d(C_{(5,11)}, C_1) = \min\{d(C_5, C_1), d(C_{11}, C_1)\} = \min(3.574944,\ 3.9524186) = 3.574944$$

```
# Fit distance matrix into the "single" method
fitSingle = hclust(distanceM, method = "single")

# Find the height of the every merging after each iteration
fitSingle[["height"]]

##  [1] 0.3774746 0.4218928 0.4218928 0.7549491 0.7549491 0.7993674 0.7993674
##  [8] 1.2212601 1.5098983 1.6875712
```

`height` - A set of n-1 real values (non-decreasing for ultrametric trees). The clustering height: that is, the value of the criterion associated with the clustering method for the particular agglomeration. Here are 10 ascending ordered heights respect to the 10 iterations and each of them represent the minimal distance between the merged clusters in every iteration.

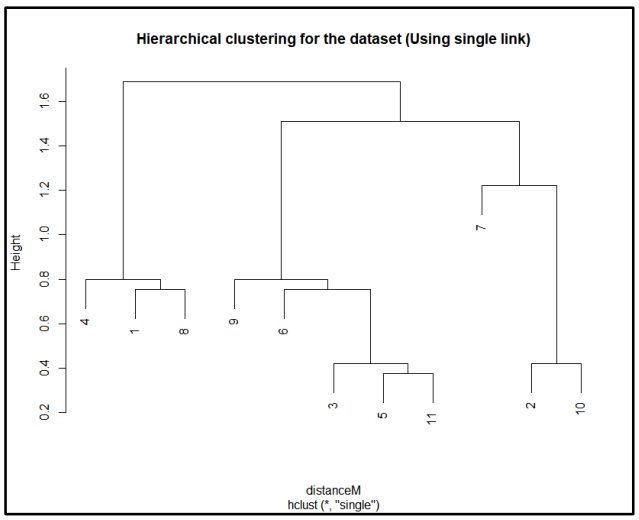
```
# Find the merging pattern in every iteration
fitSingle[["merge"]]

##       [,1] [,2]
##  [1,]   -5  -11
##  [2,]   -2  -10
##  [3,]   -3    1
##  [4,]   -1   -8
##  [5,]   -6    3
##  [6,]   -4    4
##  [7,]   -9    5
##  [8,]   -7    2
##  [9,]    7    8
## [10,]    6    9
```

`merge` - An n-1 by 2 matrix. Row i of merge describes the merging of clusters at step i of the clustering. If an element j in the row is negative, then observation -j was merged at this stage. If j is positive then the merge was with the cluster formed at the (earlier) stage j of the algorithm. Thus negative entries in merge indicate agglomerations of singletons, and positive entries indicate agglomerations of non-singletons.

```
# Find the object order of the clustering
fitSingle[["order"]]

##  [1]  4  1  8  9  6  3  5 11  7  2 10
```

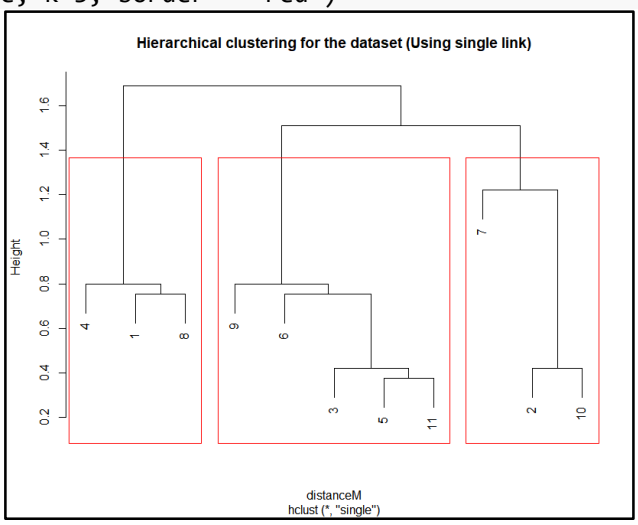
`order` - A vector giving the permutation of the original observations suitable for plotting, in the sense that a cluster plot using this ordering and matrix merge will not have crossings of the branches.

```
# Plot the dendrogram
plot(fitSingle, main = "Hierarchical clustering for the dataset (Using single link
)")
```



The largest differences of heights are in the dendrogram occurs 2 clusters before the final combination. The optimal number of clusters is thus 3.

```
rect.hclust(fitSingle, k=3, border = "red")
```



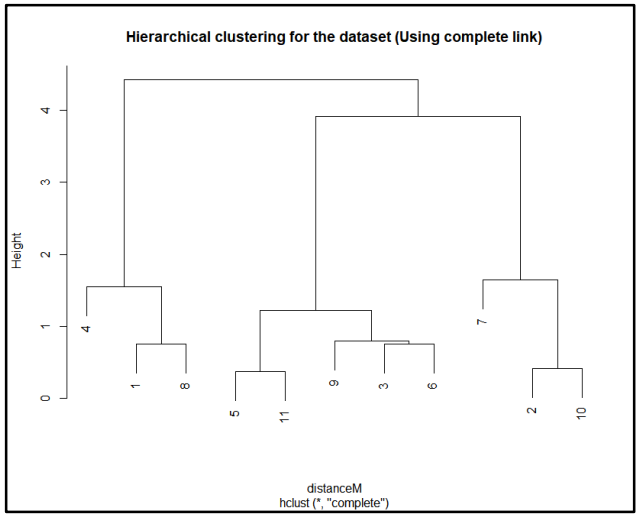
ii) Using complete linkage method

$$\boldsymbol{d(C_{(ij)}, C_k) = \max\{d(C_i, C_k), d(C_j, C_k)}$$

Example calculation using (5, 11) cluster and object 1;

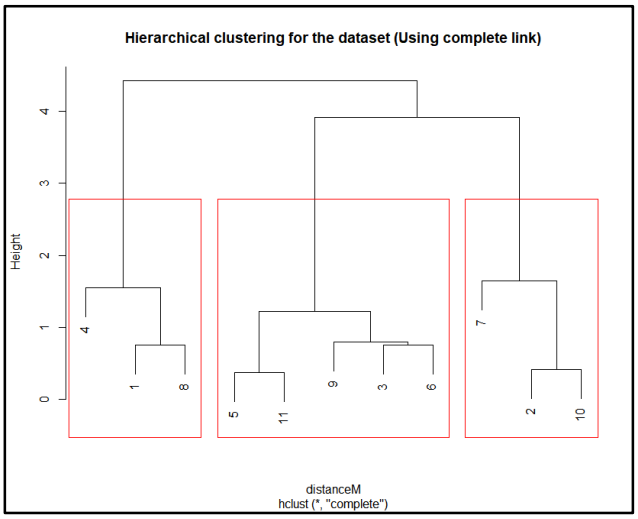
$$d(C_{(5,11)}, C_1) = \max\{d(C_5, C_1), d(C_{11}, C_1)\} = \max(3.574944,\ 3.9524186) = 3.9524186$$

```
# Fit distance matrix into the "complete" method
fitComplete = hclust(distanceM, method = "complete")

# Find the height of the every merging after each iteration
fitComplete[["height"]]

##  [1] 0.3774746 0.4218928 0.7549491 0.7549491 0.7993674 1.2212601 1.5543165
##  [8] 1.6431529 3.9080003 4.4187296

# Find the merging pattern in every iteration
fitComplete[["merge"]]

##       [,1] [,2]
##  [1,]   -5  -11
##  [2,]   -2  -10
##  [3,]   -1   -8
##  [4,]   -3   -6
##  [5,]   -9    4
##  [6,]    1    5
##  [7,]   -4    3
##  [8,]   -7    2
##  [9,]    6    8
## [10,]    7    9

# Find the object order of the clustering
fitComplete[["order"]]

##  [1]  4  1  8  5 11  9  3  6  7  2 10

# Plot the dendrogram
plot(fitComplete, main = "Hierarchical clustering for the dataset (Using complete
link)")
```



The largest differences of heights are in the dendrogram occurs 2 clusters before the final combination. The optimal number of clusters is thus 3.

```
rect.hclust(fitComplete, k=3, border = "red")
```



iii) Using average linkage method

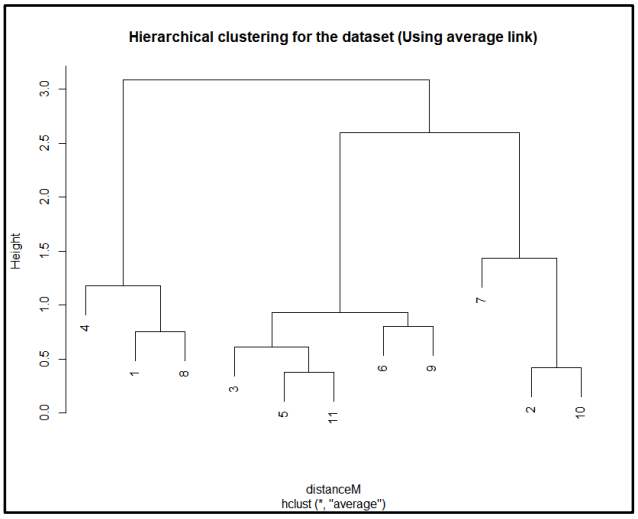
$$d(C_{(ij)}, C_k) = \frac{d(C_i, C_k) + d(C_j, C_k)}{No.\ of\ distances}$$

Example calculation using (5, 11) cluster and object 1;

$$d(C_{(5,11)}, C_1) = \frac{d(C_5, C_1) + d(C_{11}, C_1)}{2} = \frac{(3.574944 + 3.9524186)}{2} = 3.7636813$$

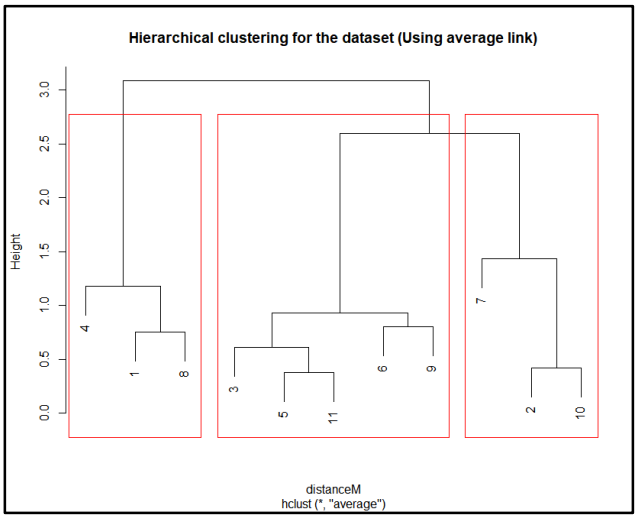
```
# Fit distance matrix into the "average" method
fitAverage = hclust(distanceM, method = "average")

# Find the height of the every merging after each iteration
fitAverage[["height"]]

##  [1] 0.3774746 0.4218928 0.6106301 0.7549491 0.7993674 0.9325952 1.1768419
##  [8] 1.4322065 2.5949639 3.0855286

# Find the merging pattern in every iteration
fitAverage[["merge"]]

##       [,1] [,2]
##  [1,]   -5  -11
##  [2,]   -2  -10
##  [3,]   -3    1
##  [4,]   -1   -8
##  [5,]   -6   -9
##  [6,]    3    5
##  [7,]   -4    4
##  [8,]   -7    2
##  [9,]    6    8
## [10,]    7    9
```

```
# Find the object order of the clustering
fitAverage[["order"]]

##  [1]  4  1  8  3  5 11  6  9  7  2 10

# Plot the dendrogram
plot(fitAverage, main = "Hierarchical clustering for the dataset (Using average li
nk)")
```



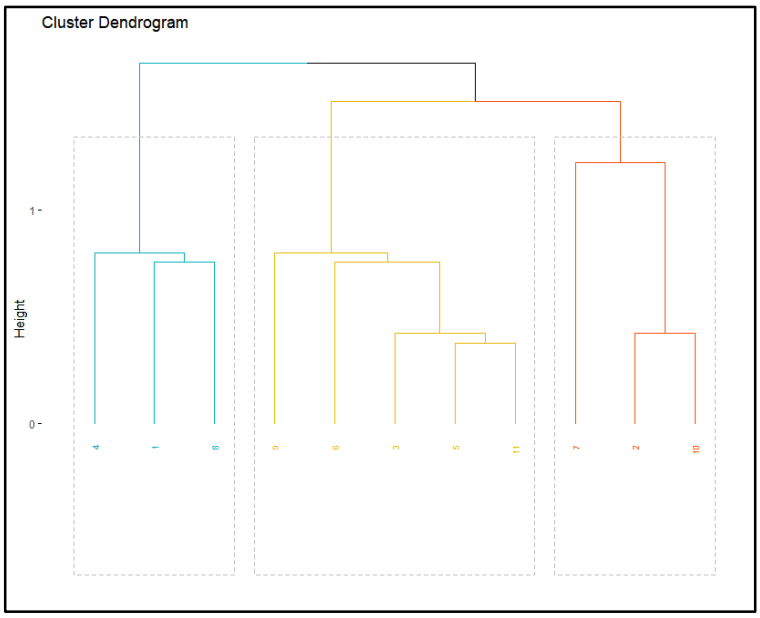
The largest differences of heights are in the dendrogram occurs 2 clusters before the final combination. The optimal number of clusters is thus 3.

```
rect.hclust(fitAverage, k=3, border = "red")
```

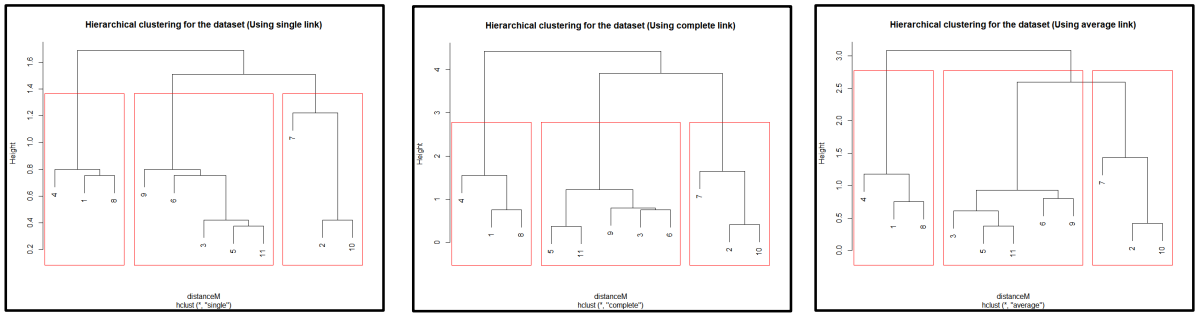


Step 5 - Enhanced visualization of the dendrogram

```
library(ggplot2)
library(factoextra)

fviz_dend(fitSingle, k=3, cex = 0.5, k_colors = c("#00AFBB", "#E7B800", "#FC4E07")
, color_labels_by_k = TRUE, rect = TRUE)
```



Comparison of the dendrograms produced by the three methods



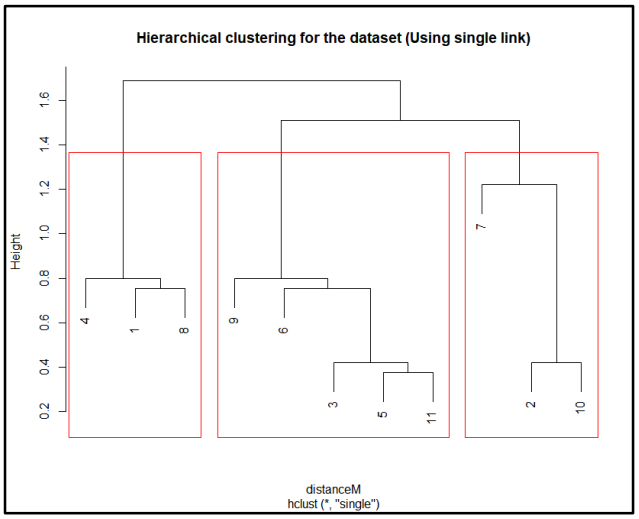
Three major clusters are formed by each three methods and the objects belong to the each cluster are similar in the three methods. Even though, distance values among clusters which were formed by three methods are quite different due to the different merging calculations of the three methods. When consider the topologies of the three graphs, both single link method and the average link method gave the identical topologies. In complete link method, clustering of the 5, 11, 9, 3 and 6 objects are completely different to the other two methods.

(b) How do you use the Dendrogram to compare the similarities of clusters at different levels of hierarchies?

Answer

Dendogram is used to show the hierarchical relationship between objects and usually created as an output from hierarchical clustering. Generally it gives the summary of the distance matrix. The height of the links that joins two objects or clusters together reflects the Euclidian or Manhattan distance between these two. These distances are relatively proportional to the similarity between the two objects. Shorter heights reflects the lower distances, and hence high similarity between two objects/clusters and vice versa.

In this question, according to the single link method, distance between the (((2,10),7), ((((11,5),3),6),9)) is 1.5098983 and the distance between the (((((11,5),3),6),9)), ((8,1),4)) clusters is 1.6875712. Since 1.5098983 < 1.6875712, first two clusters are closely related compared to the second two clusters.



References

Saraçli, S., Doğan, N. & Doğan, İ. Comparison of hierarchical cluster analysis methods by cophenetic correlation. *J Inequal Appl* **2013**, 203 (2013). https://doi.org/10.1186/1029-242X-2013-203

<https://www.molmine.com/help/algorithms/linkage/.htm>, Accsessed on 2022.09.17