

**School of Information Technology and Electrical Engineering**  
**INFS4203/7203 – Data Mining**

**Tutorials 5 - 6 Clustering (Solutions)**

**Question 1.**

Suppose that the task is to cluster the following eight points (with  $(x,y)$  representing location) into three clusters.

A1: (2, 10), A2: (2, 5), A3: (8, 4),  
B1: (5, 8), B2: (7, 5), B3: (6, 4),  
C1: (1, 2), C2: (4, 9)

The distance function is Euclidean distance. Suppose initially we assign A1, B2, and C1 as the centre of each cluster, respectively. Use the *k-means* algorithm to show:

- a) The three cluster centres after the first round execution, and
- b) The final three clusters.

**ANSWER:**

The given eight points were:

A1: (2, 10)  
A2: (2, 5)  
A3: (8, 4)  
B1: (5, 8)  
B2: (7, 5)  
B3: (6, 4)  
C1: (1, 2)  
C2: (4, 9)

The points A1, B2, and C1 were assumed to be the initial three cluster centres. There are two possible ways of clustering the given points using the k-means algorithm, depending on to which cluster we assign the point B1:

**The first way:**

1.1. Round One:

- Compute for each of the eight points its distance from each cluster centre, and assign it to the cluster to which it is most similar.

=> The clustering after the first round execution is:

Cluster 1: A1, B1, C2  
Cluster 2: B2, A3, B3

Cluster 3: C1, A2

- update the cluster centres:

The new centre of Cluster 1: (3.67, 9)

The new centre of Cluster 2: (7, 4.33)

The new centre of Cluster 3: (1.5, 3.5)

1.2. Round Two:

- No changes

### **The second way:**

2.1. Round One:

- Compute for each of the eight points its distance from each cluster centre, and assign it to the cluster to which it is most similar.

=> The clustering after the first round execution is:

Cluster 1: A1, C2

Cluster 2: B2, A3, **B1**, B3

Cluster 3: C1, A2

- update the cluster centres:

The new centre of Cluster 1: (3, 9.5)

The new centre of Cluster 2: (6.5, 5.25)

The new centre of Cluster 3: (1.5, 3.5)

2.2. Round Two:

- recompute for each of the eight points its distance from each cluster centre, and assign it to the cluster to which it is most similar.

=> The clustering after the second round execution is:

Cluster 1: A1, **B1**, C2

Cluster 2: B2, A3, B3

Cluster 3: C1, A2

- update the cluster centres:

The new centre of Cluster 1: (3.67, 9)

The new centre of Cluster 2: (7, 4.33)

The new centre of Cluster 3: (1.5, 3.5)

2.3. Round Three:

- No changes

**Question 2.** (Read Chapter 8.3 of the textbook, pp515-526)

Consider a set of six objects: {A, B, C, D, E, F}. Let the following be a dissimilarity matrix between these objects.

	A	B	C	D	E	F
A	0.0	1.0	5.0	9.0	10.0	2.0
B	1.0	0.0	3.5	8.0	7.0	5.5
C	5.0	3.5	0.0	3.0	4.0	6.5
D	9.0	8.0	3.0	0.0	0.5	4.5
E	10.0	7.0	4.0	0.5	0.0	2.5
F	2.0	5.5	6.5	4.5	2.5	0.0

Construct agglomerative clustering hierarchies for these objects using

- a) The single linkage,
- b) The complete linkage, and
- c) The average linkage method.

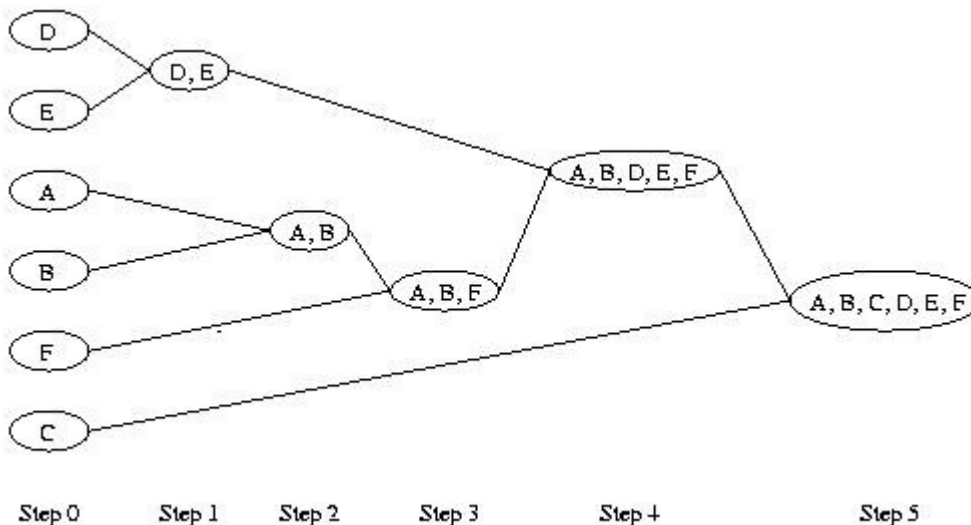
Are the constructed hierarchies similar or different to each other?

Are some of them more reasonable than the others?

**ANSWER:**

Assuming that we have the given set of six objects and the given dissimilarity matrix, we get the following agglomerative clustering hierarchies:

- a) Using the *single linkage* method, where  $d(C_i, C_j) = \min \{ d(x,y) \mid x \text{ in } C_i, y \text{ in } C_j \}$ .  
We get the hierarchy:



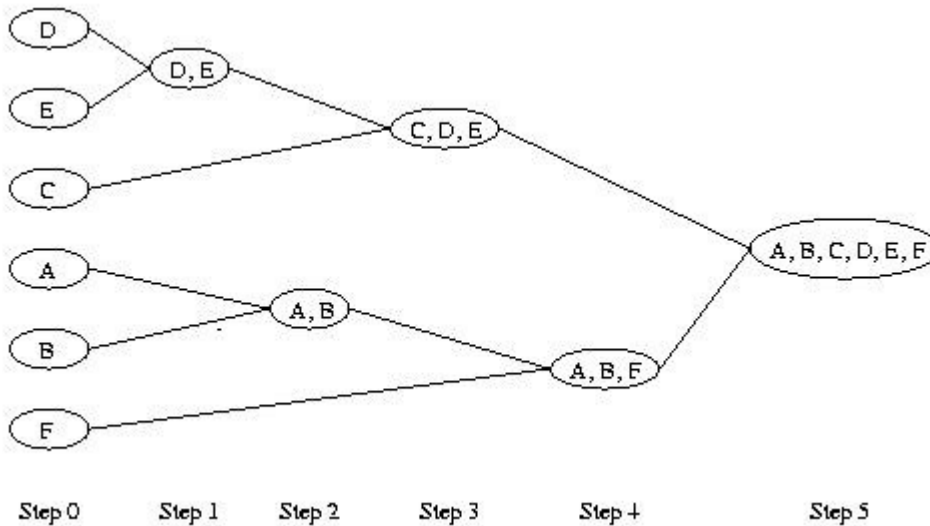
The following marked (red text) entries show the min values selected during the computations.

	A	B	C	D	E	F
A	0.0	1.0 (Step 2)	5.0	9.0	10.0	2.0 (Step 3)
B	1.0	0.0	3.5	8.0	7.0	5.5
C	5.0	3.5	0.0	3.0	4.0	6.5
D	9.0	8.0	3.0	0.0	0.5 (Step 1)	4.5
E	10.0	7.0	4.0	0.5	0.0	2.5 (Step 4)
F	2.0	5.5	6.5	4.5	2.5	0.0

b) Using the complete linkage method, where

$$d(C_i, C_j) = \max \{ d(x,y) \mid x \text{ in } C_i, y \text{ in } C_j \}$$

We get the hierarchy:



The following marked (red text) entries show the min values selected during the computations.

	A	B	C	D	E	F
A	0.0	1.0	5.0	9.0	10.0	2.0
B	1.0 (Step 2)	0.0	3.5	8.0	7.0	5.5
C	5.0	3.5	0.0	3.0	4.0	6.5
D	9.0	8.0	3.0	0.0	0.5 (Step 1)	4.5
E	10.0	7.0	4.0 (Step 3)	0.5 (Step 1)	0.0	2.5
F	2.0	5.5 (Step 4)	6.5	4.5	2.5	0.0

Actually a more detailed process of this approach can be illustrated as:

**Step 0:** Take every point as an individual cluster.

**Step 1:** Calculate maximum distances between clusters. At this step the distances of all pairs of points are calculated. Then find out the minimum distance to merge (D & E) (i.e.,  $d(D,E) = 0.5$ )

**Step 2:** Calculate maximum distances between clusters. At this step the distances of following clusters are calculated:

The maximum distance between clusters of DE and C is 4, denoted as  $DE:C = 4$ .

Similarly, others can be found as:

$$DE:A = 10$$

$$DE:B = 8$$

$$DE:F=4.5$$

$$C:A=5$$

$$C:B=3.5$$

$$C:F=6.5$$

$$A:B=1.0$$

$$A:F=2.0$$

$$B:F=5.5$$

The minimum of the above maximum distances is  $d(A,B) = 1.0$ . So, the points AB are merged.

**Step 3:** The following maximum distances between the clusters are calculated:

$$DE:C=4$$

$$DE:AB=10$$

$$DE:F=4.5$$

$$C:AB = 5$$

$$C:F=5.5$$

$$AB:F=5.5$$

At this step, the minimum value of all maximum values is  $d(DE,C) = 4$ . So, the points of (DE) & C are merged.

**Step 4:** The following maximum distances between the clusters are calculated:

$$CDE:AB=10$$

$$CDE:F=6.5$$

$$AB:F=5.5$$

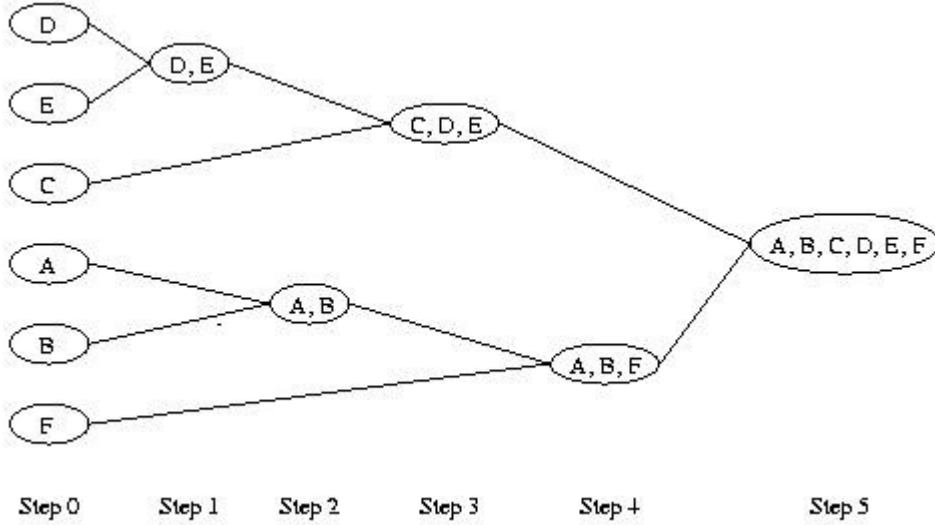
At this step, the minimum value of all maximum values is  $d(AB,F) = 5.5$ . So, points of (AB) & F are merged.

**Step 5:** All points are merged.

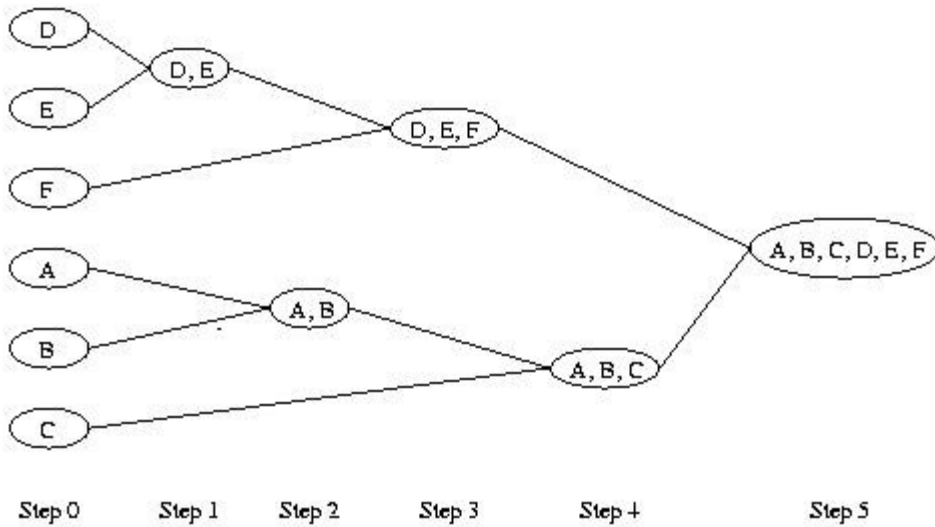
c) Using the average linkage method, where

$$d(C_i, C_j) = \text{avg} \{ d(x,y) \mid x \text{ in } C_i, y \text{ in } C_j \}.$$

We get the hierarchy:



Or the hierarchy:



Depending on the clustering choice we make in Step 3.

NB. The dimensionality of the points in this question is not mentioned. It is not necessarily in 2 or 3 dimensions. The only assumption is that these points are in the Euclidian space.

**Question 3.** (From the Textbook Exercise 8-3)

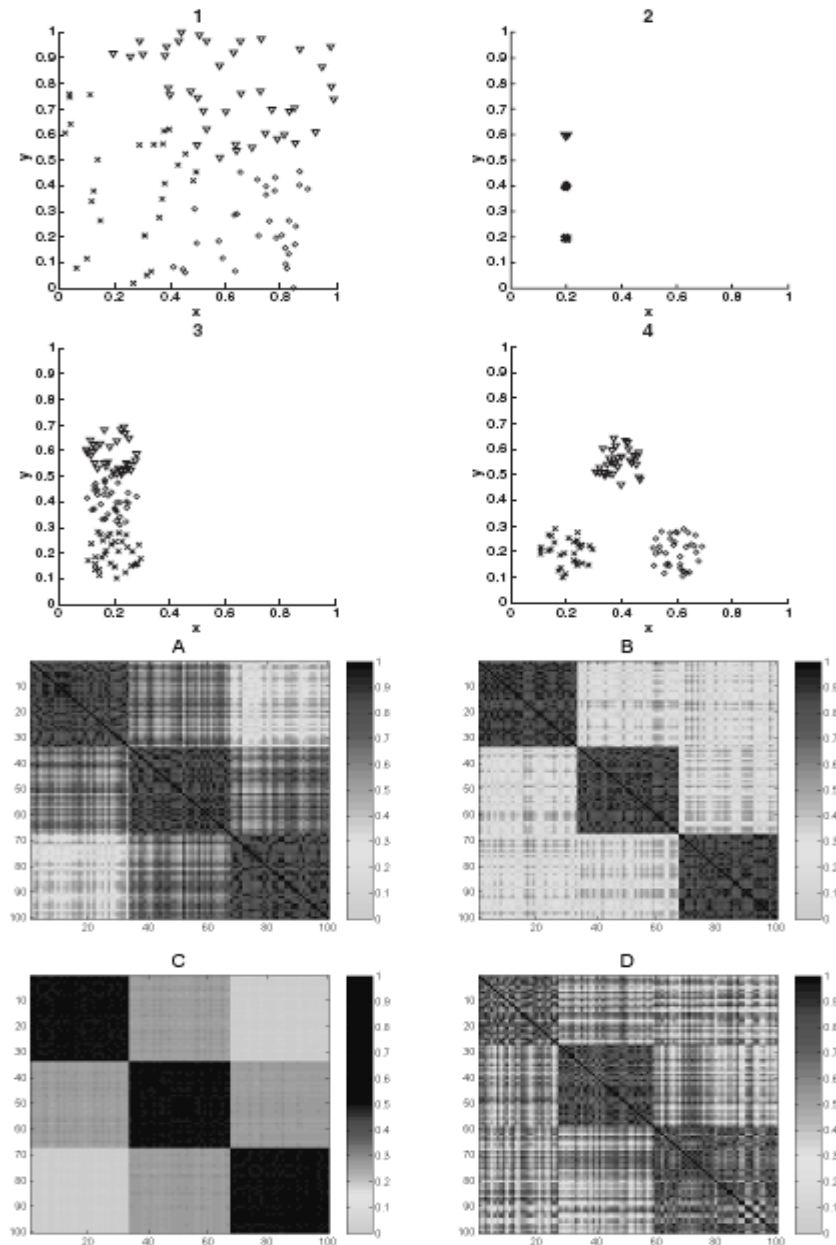
Many partitional clustering algorithms that automatically determine the number of clusters claim that this is an advantage. List two situations in which this is not the case.

**ANSWER:**

- **When there is hierarchical structure in the data**  
Most algorithms that automatically determine the number of clusters are partitioned, and thus, ignore the possibility of sub-clusters.
- **When the clustering is for data utility**  
If a certain reduction in data size is needed, then it is necessary to specify how many clusters (cluster centroids) are produced.

**Question 4.** (From the Textbook Exercise 8-32)

In Figure 8.9, match the similarity matrices, which are sorted according to cluster labels, with the sets of points. Differences in shading and marker shape distinguish between clusters, and each set of points contains 100 points and three clusters. In the set of points labelled 2, there are three very tight, equal-sized clusters.



**Figure 8.9.** Points and similarity matrices for Exercise 32.

ANSWER.

1-D, 2-C, 3-A, 4-B. Note that every matrix has 100 x 100 entries. The darker that the entry is, the more similar (i.e., closer to each other) that two points are. The picture is symmetric because the similarity function as the Euclidian distance is symmetric. A line (horizontal or vertical) indicates 100 similarity results computed for a single point to all other points. So the anti-diameter dark line in the picture shows that every point is mostly similar to itself (i.e., in the darkest colour).