Hierarchical Clustering

Q1. Briefly describe the basic idea behind hierarchical clustering.

[Answer]

Hierarchical clustering sequentially partitions a data set with a given distance measure. In this sequential partition process, an algorithm constructs nested partitions layer by layer via grouping objects into a tree of clusters solely based on the distance measure without the need to know the number of clusters in advance.

There are two strategies to a tree of clusters; i.e., top-down and bottom-up. The top-down strategy assumes that all objects in a data set is initially in a single cluster and then the cluster is recurrently divided into smaller and smaller clusters until a stopping condition is met. In contrast, the bottom-up strategy, well known as agglomerative algorithm, assumes that all objects in a data set are atomic clusters of a single element. Then all atomic clusters merge to become larger and larger clusters until a stopping condition is met.

Q2. Summarise the main differences between hierarchical clustering with partitioning clustering, e.g., K-means clustering.

[Answer]

First of all, hierarchical clustering is a sequential partitioning process, which results in a hierarchical nested cluster structure, while partitioning clustering is an iterative partitioning process, which leads to a flat mutually exclusive cluster structure. Thus, the membership of an object or cluster is fixed in hierarchical clustering once the cluster that it belongs to is formed during the clustering process whilst the membership could change constantly in partitioning clustering.

Next, hierarchical clustering does not need prior knowledge on the number of clusters behind a data set, whilst partitioning clustering must be given the number of clusters in advance.

Finally, hierarchical clustering provides a generic clustering technique regardless of data types, whilst partitioning clustering requires that data are summarised by a set of representative entities, e.g., centroids in K-means clustering. For instance, there is no natural centroids underlying categorical data and therefore K-means clustering is not applicable to such data without modification. In contrast, hierarchical clustering needs a distance measure only and is applicable to many data types including categorical data.

Q3. Given a one-dimensional data set {2, 4, 5, 9, 10}, it has been divided into two clusters {1, 2, 3} and {4, 5}, use single, complete and average links with Euclidean distance to calculating the distances between them, respectively.

[Answer]

In order to calculate the distance between two clusters, we need to know the distance between any two objects in different clusters. Therefore, we first calculate the distance with Euclidean distance to find the distance matrix of this data set as follows:

$$\begin{bmatrix} 0 & 2 & 3 & 7 & 8 \\ 2 & 0 & 1 & 4 & 5 \\ 3 & 1 & 0 & 4 & 5 \\ 7 & 4 & 4 & 0 & 1 \\ 8 & 5 & 5 & 1 & 0 \end{bmatrix}$$

For the single link, we need to find the shortest distance; i.e.,
d({1, 2, 3}, {4, 5}) = min{ d(1,4), d(1, 5), d(2,4), d(2,5), d(3,4), d(3,5)}
= min{ 7, 8, 4, 5, 4, 5}
= 4

For the complete link, we need to find the longest distance; i.e.,
d({1, 2, 3}, {4, 5}) = max{ d(1,4), d(1, 5), d(2,4), d(2,5), d(3,4), d(3,5)}
= max{ 7, 8, 4, 5, 4, 5}
= 8

For the average link, we need to find the averaging distance; i.e.,
d({1, 2, 3}, {4, 5}) = [d(1,4) + d(1, 5) + d(2,4) + d(2,5) + d(3,4) + d(3,5)}]/6
= [7 + 8 + 4 + 5 +4 + 5]/6
= 5.5

Q4. Given a one-dimensional data set {1, 5, 8, 10, 2}, use the agglomerative clustering algorithms with the complete link with Euclidean distance to establish a hierarchical grouping relationship. By using the maximal lifetime as the cutting threshold, how many clusters are there? What is their membership in each cluster?

[Answer]

In order to use the agglomerative algorithm, we need to calculate the distance matrix.

$$\begin{bmatrix} 0 & 4 & 7 & 9 & 1 \\ 4 & 0 & 3 & 5 & 3 \\ 7 & 3 & 0 & 2 & 6 \\ 9 & 5 & 2 & 0 & 8 \\ 1 & 3 & 6 & 8 & 0 \end{bmatrix}$$

From the distance matrix, we can find that the distance between points 1 and 5 is smallest. Therefore, we merge them together with their distance as the threshold. Then, we update the distance matrix by using the cluster {1, 5} . Using the complete link, we can re-calculate the distance between this cluster and other points

d(2, {1,5}) = max{ d(2,1), d(2,5) } = max {4, 3} = 4
d(3, {1,5}) = max{ d(3,1), d(3,5) } = max {7, 6} = 7
d(4, {1,5}) = max{ d(4,1), d(4,5) } = max {9, 8} = 9

Let the 1st column (row) denote the distances between this cluster and other points, we have the following distance matrix:

$$\begin{bmatrix} 0 & 4 & 7 & 9 \\ 4 & 0 & 3 & 5 \\ 7 & 3 & 0 & 2 \\ 9 & 5 & 2 & 0 \end{bmatrix}$$

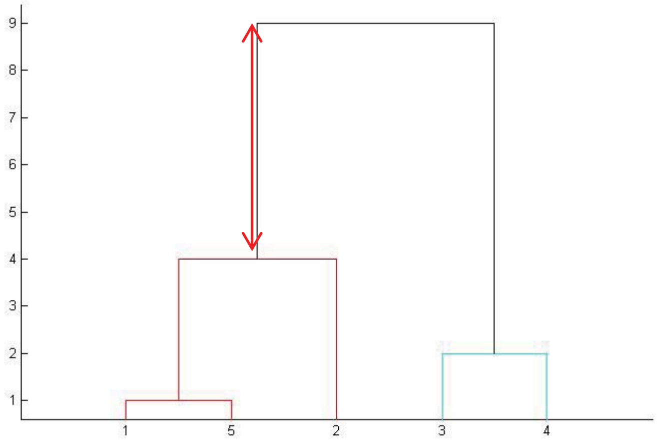
From the above distance matrix, we can see the distance between points 3 and 4 is smallest. Hence, they merge together to form a cluster {3, 4}. Using the complete link, we have the distance between different points/clusters as follows:
d({1,5}, {3, 4}) = max{ d({1,5}, 3), d({1,5}, 4) } = max{ 7, 9} = 9
d(2, {3,4}) = max{ d(2,3), d(2,4) } = max {3, 5} = 5
Thus, we can update the distance matrix, where row 2 corresponds to point 2, rows 1 and 3 correspond to clusters {1, 5} and {3, 4}, as follows:

$$\begin{bmatrix} 0 & 4 & 9 \\ 4 & 0 & 5 \\ 9 & 5 & 0 \end{bmatrix}$$

Following the same procedure, we merge point 2 with the cluster {1, 5} to form {1, 2, 5} and update the distance matrix as follows:

$$\begin{bmatrix} 0 & 5 \\ 5 & 0 \end{bmatrix}$$

After increasing the distance threshold to 5, all clusters would merge.
Based on all above distance matrices, we draw the dendrogram tree as follows:



Based on the dendrogram tree, we can get the lifetime for different clusters as follows: L_2 = 5, L_3 = 2, L_4 = 1. When we use the maximal lifetime L_2, as shown in the above diagram, to cut the dendrogram tree, we can achieve two clusters {1, 2, 5} and {3, 4}, as coloured by red and green in the dendrogram tree.