13.3.1.1.1 k-Means Solved Examples in One-Dimensional Data

Solved Problem 13.1

Apply k-means algorithm in given data for $k = 3$ (that is, 3 clusters). Use $C_1(2)$, $C_2(16)$, and $C_3(38)$ as initial cluster centers. Data: 2, 4, 6, 3, 31, 12, 15, 16, 38, 35, 14, 21, 23, 25, 30.

Solution:

The initial cluster centers are given as $C_1(2)$, $C_2(16)$, and $C_3(38)$. Calculating the distance between each data point and cluster centers, we get the following table.

Data Points	*Distance from $C_1(2)$*	*Distance from $C_2(16)$*	*Distance from $C_3(38)$*
2	$(2-2)^2=0$	$(2-16)^2=196$	$(2-38)^2=1296$
4	$(4-2)^2=4$	$(4-16)^2=144$	$(4-38)^2=1156$
6	$(6-2)^2=16$	$(6-16)^2=100$	$(6-38)^2=1024$
3	$(3-2)^2=1$	$(3-16)^2=169$	$(3-38)^2=1225$
31	$(31-2)^2=841$	$(31-16)^2=225$	$(31-38)^2=49$
12	$(12-2)^2=100$	$(12-16)^2=16$	$(12-38)^2=676$
15	$(15-2)^2=169$	$(15-16)^2=1$	$(15-38)^2=529$
16	$(16-2)^2=196$	$(16-16)^2=0$	$(16-38)^2=484$
38	$(38-2)^2=1296$	$(38-16)^2=484$	$(38-38)^2=0$
35	$(35-2)^2=1089$	$(35-16)^2=361$	$(35-38)^2=9$
14	$(14-2)^2=144$	$(14-16)^2=4$	$(14-38)^2=576$
21	$(21-2)^2=361$	$(21-16)^2=25$	$(21-38)^2=289$
23	$(23-2)^2=441$	$(23-16)^2=49$	$(23-38)^2=225$
25	$(25-2)^2=529$	$(25-16)^2=81$	$(25-38)^2=169$
30	$(30-2)^2=784$	$(30-16)^2=196$	$(30-38)^2=64$

By assigning the data points to the cluster center whose distance from it is minimum of all the cluster centers, we get the following table.

$C_1(2)$	$C_2(16)$	$C_3(38)$
m1 = 2	m2 = 16	m3 = 38
{2, 3, 4, 6}	{12, 14, 15, 16, 21, 23, 25}	{31, 35, 38}
New cluster centers		
m1 = 3.75	m2 = 18	m3 = 34.67

Similarly, using the new cluster centers we can calculate the distance from it and allocate clusters based on minimum distance. It is found that there is no difference in the cluster formed and hence we stop this procedure. The final clustering result is given in the following table.

$C_1(3.75)$	$C_2(18)$	$C_3(34.67)$
m1 = 3.75	m2 = 18	
{2, 3, 4, 6}	{12, 14, 15, 16, 21, 23, 25}	m3 = 34.67
		{31, 35, 38}

Solved Problem 13.2

Apply *k*-means algorithm in given data for $k = 2$. Use $C_1(80)$ and $C_2(250)$ as initial cluster centers. Data: 234, 123, 456, 23, 34, 56, 78, 90, 150, 116, 117, 118, 199.

Solution:

We solve the numerical by following the calculations carried out in solved problem 13.1. The result is presented in the following table.

$C_1(80)$	$C_2(250)$
m1 = 80	
	m2 = 250
{23, 34, 56, 78, 90, 116, 117, 118, 123}	{150, 199, 234, 456}
m1 = 83.9	
	m2 = 259.75
{23, 34, 56, 78, 90, 116, 117, 118, 123}	{150, 199, 234, 456}
m1 = 90.5	
	m2 = 296.3
{23, 34, 56, 78, 90, 116, 117, 118, 123, 150}	{199, 234, 456}
m1 = 90.5	
	m2 = 296.3
{23, 34, 56, 78, 90, 116, 117, 118, 123, 150}	{199, 234, 456}

13.3.1.1.2 *k*-Means Solved Examples in Two-Dimensional Data

Solved Problem 13.3

Apply *k*-means clustering for the datasets given in Table 13.1 for two clusters. Tabulate all the assignments.

Table 13.1 Sample dataset for *k*-means clustering

Sample No.	*X*	*Y*
1	185	72
2	170	56
3	168	60
4	179	68
5	182	72
6	188	77

Solution:

Sample No.	*X*	*Y*	*Assignment*
1	185	72	*C*1
2	170	56	*C*2

Centroid: *C*1 = (185, 72) and *C*2 = (170, 56)
First Iteration:
Distance from *C*1 is Euclidean distance between (185, 72) and (168, 60) = 20.808
Distance from *C*2 is Euclidean distance between (170, 56) and (168, 60) = 4.472
Since *C*2 is closer to (168, 60), the sample belongs to *C*2.

Sample No.	*X*	*Y*	*Assignment*
1	185	72	*C*1
2	170	56	*C*2
3	168	60	*C*2
4	179	68	
5	182	72	
6	188	77	

Similarly,

1. Distance from *C*1 for (179, 68) = 7.21
   Distance from *C*2 for (179, 68) = 15
   Since *C*1 is closer to (179, 68), the sample belongs to *C*1.
2. Distance from *C*1 for (182, 72) = 3
   Distance from *C*2 for (182, 72) = 20
   Since *C*1 is closer to (182, 72), the sample belongs to *C*1.
3. Distance from *C*1 for (188, 77) = 5.83
   Distance from *C*2 for (188, 77) = 27.66
   Since *C*1 is closer to (188, 77), the sample belongs to *C*1.

Sample No.	*X*	*Y*	*Assignment*
1	185	72	*C*1
2	170	56	*C*2
3	168	60	*C*2
4	179	68	*C*1
5	182	72	*C*1
6	188	77	*C*1

The new centroid for $C1$ is

$$\left(\frac{185+179+182+188}{4}, \frac{72+68+72+77}{4}\right) = (183.5, 73)$$

The new centroid for $C2$ is

$$\left(\frac{170+168}{2}, \frac{56+60}{2}\right) = (169, 58)$$

Second Iteration:

Distance from $C1$ is Euclidean distance between (183.5, 73) and (168, 60) = 20.2
Distance from $C2$ is Euclidean distance between (169, 58) and (168, 60) = 2.24
Since $C2$ is closer to (168, 60), the sample belongs to $C2$.
Similarly,

1. Distance from $C1$ for (179, 68) = 6.73
   Distance from $C2$ for (179, 68) = 14.14
   Since $C1$ is closer to (179, 68), the sample belongs to $C1$.
2. Distance from $C1$ for (182, 72) = 1.80
   Distance from $C2$ for (182, 72) = 19.10
   Since $C1$ is closer to (182, 72), the sample belongs to $C1$.
3. Distance from $C1$ for (188, 77) = 6.02
   Distance from $C2$ for (188, 77) = 26.87
   Since $C1$ is closer to (188, 77), the sample belongs to $C1$.

Sample No.	*X*	*Y*	*Assignment*
1	185	72	$C1$
2	170	56	$C2$
3	168	60	$C2$
4	179	68	$C1$
5	182	72	$C1$
6	188	77	$C1$

After the second iteration, the assignment has not changed and hence the algorithm is stopped and the points are clustered.

13.3.2 k-Medoids

The k-medoids algorithm is a clustering algorithm very similar to the k-means algorithm. Both k-means and k-medoids algorithms are partitional and try to minimize the distance between points and cluster center. In contrast to the k-means algorithm, k-medoids chooses data points as centers and uses Manhattan distance to define the distance between cluster centers and data points. This technique clusters the dataset of n objects into k clusters, where the number of clusters k is known in prior. It is more robust to noise and outliers as compared to k-means because it minimizes a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances. A medoid is defined as an object of a cluster whose average dissimilarity to all the objects in the cluster is minimal.

The Manhattan distance between two vectors in an n-dimensional real vector space is given by Eq. (13.2). It is used in computing the distance between a data point and its cluster center.

$$d_1(p,q) = \|p-q\|_1 = \sum_{i=1}^{n} |p_i - q_i| \tag{13.2}$$

The most common algorithm in k-medoid clustering is Partitioning Around Medoids (PAM) algorithm. PAM uses a greedy search which is faster than the exhaustive search and may not find the optimum solution. It works as follows:

1. Initialize: select k of the n data points as the medoids.
2. Associate each data point to the closest medoid.
3. While the cost of the configuration decreases: For each medoid m and for each non-medoid data point o:
   - Swap m and o, recompute the cost (sum of distances of points to their medoid).
   - If the total cost of the configuration increased in the previous step, undo the swap.

Solved Problem 13.4

Cluster the following dataset of 6 objects into two clusters, that is, $k = 2$.

X1	2	6
X2	3	4
X3	3	8
X4	4	2
X5	6	2
X6	6	4

Solution:

Step 1: Two observations c1 = X2 = (3, 4) and c2 = X6 = (6, 4) are randomly selected as medoids (cluster centers).

Step 2: Manhattan distances are calculated to each center to associate each data object to its nearest medoid.

Data Object		*Distance To*	
Sample	*Point*	*c1 = (3, 4)*	*c2 = (6, 4)*
X1	(2, 6)	3	6
X2	(3, 4)	0	3
X3	(3, 8)	4	7
X4	(4, 2)	3	4
X5	(6, 2)	5	2
X6	(6, 4)	3	0
Cost		10	2

Step 3: We select one of the non-medoids O′. Let us assume O′ = (6, 2). So now the medoids are c1(3, 4) and O′(6, 2). If c1 and O′ are the new medoids. We calculate the total cost involved.

Data Object		*Distance To*	
Sample	*Point*	*c1 = (3, 4)*	*c2 = (6, 2)*
X1	(2, 6)	3	8
X2	(3, 4)	0	5
X3	(3, 8)	4	9
X4	(4, 2)	3	2
X5	(6, 2)	5	0
X6	(6, 4)	3	2
Cost		7	4

So cost of swapping medoid from c2 to O′ is 11. Since the cost is less, this is considered as a better cluster assignment. Here swapping is done as the cost is less.

Step 4: We select another non-medoid O′. Let us assume O′ = (4, 2). So now the medoids are c1(3, 4) and O′(4, 2). If c1 and O′ are new medoids, we calculate the total cost involved.

Data Object		*Distance To*	
Sample	*Point*	*c1 = (3, 4)*	*c2 = (4, 2)*
X1	(2, 6)	3	6
X2	(3, 4)	0	3
X3	(3, 8)	4	7
X4	(4, 2)	3	0
X5	(6, 2)	5	2
X6	(6, 4)	3	4
Cost		7	8

So cost of swapping medoid from c2 to O′ is 15. Since the cost is more, this cluster assignment is not considered and the swapping is not done.

Thus, we try other non-medoids points to get minimum cost. The assignment with minimum cost is considered the best. For some applications, *k*-medoids show better results than *k*-means. The most time-consuming part of the *k*-medoids algorithm is the calculation of the distances between objects. The distances matrix can be computed in advance to speed-up the process.

13.4 Hierarchical Methods

The hierarchical agglomerative clustering methods are most commonly used. The construction of a hierarchical agglomerative classification can be achieved by the following general algorithm.

1. Find the two closest objects and merge them into a cluster.
2. Find and merge the next two closest points, where a point is either an individual object or a cluster of objects.
3. If more than one cluster remains, return to step 2.

13.4.1 Agglomerative Algorithms

Agglomerative algorithm follows a bottom-up strategy, treating each object from its own cluster and iteratively merging clusters until a single cluster is formed or a terminal condition is satisfied. According to some similarity measure, the merging is done by choosing the closest clusters first. A dendrogram, which is a tree like structure, is used to represent hierarchical clustering. Individual objects are represented by leaf nodes and the clusters are represented by root nodes. A representation of a dendrogram is shown in Fig. 13.3.

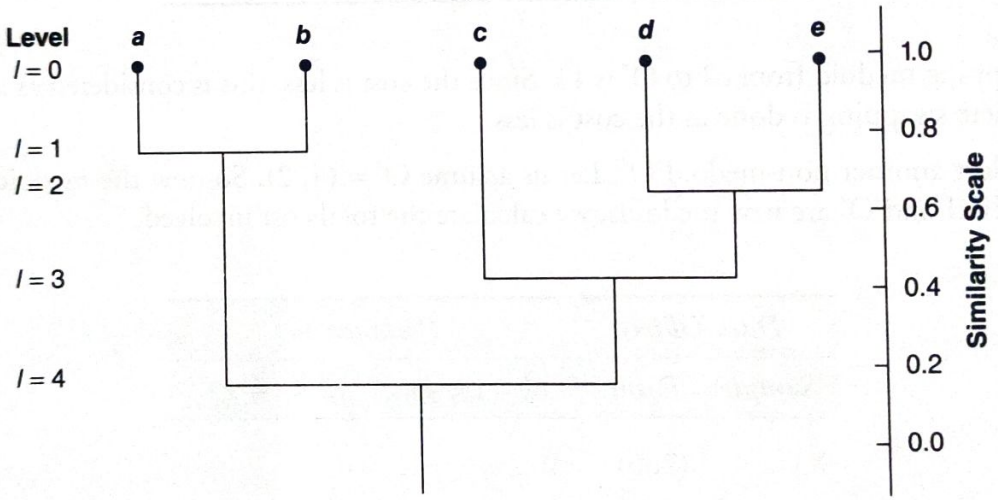

Figure 13.3 Dendogram.

13.4.1.1 Distance Measures

One of the major factors in clustering is the metric that is used to measure the distance between two clusters, where each cluster is generally a set of objects. The distance between two objects or points p and p_1 are computed using Eqs. (13.3) to (13.6). Let C_i be the cluster and n_i is the number of objects in C_i. They are also known as linkage measures.

Minimum distance:

$$\text{dist}_{\min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} \{|p - p'|\} \tag{13.3}$$

Maximum distance:

$$\text{dist}_{\max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} \{|p - p'|\} \tag{13.4}$$

Mean distance:

$$\text{dist}_{\text{mean}}(C_i, C_j) = |m_i - m_j| \tag{13.5}$$

Average distance:

$$\text{dist}_{\text{avg}}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i, p' \in C_j} |p - p'| \tag{13.6}$$

When an algorithm uses the minimum distance, $d_{\min}(C_i, C_j)$, to measure the distance between clusters, it is called *nearest-neighbor clustering algorithm*. If the clustering process is terminated when the distance between the nearest clusters exceeds a user-defined threshold, it is called *single-linkage algorithm*. Agglomerative hierarchical clustering algorithm (with minimum distance measure) is called *minimum spanning tree algorithm* since spanning tree of a graph is a tree that connects all vertices and a minimal spanning tree is one with the least sum of edge weights.

An algorithm that uses the maximum distance, $d_{\max}(C_i, C_j)$, to measure the distance between clusters is called *farthest-neighbor clustering algorithm*. If clustering is terminated when the maximum distance exceeds a user-defined threshold, it is called *complete-linkage algorithm*.

The minimum and maximum measures tend to be sensitive to outliers or noisy data. The third method thus suggests to take the average distance to rule out outlier problems. Another advantage is that it can handle categoric data as well.

Algorithm: The agglomerative algorithm is carried out in three steps and the flowchart is shown in Fig. 13.4.

1. Convert object attributes to distance matrix.
2. Set each object as a cluster (thus, if we have N objects, we will have N clusters at the beginning).
3. Repeat until number of clusters is one.
   - Merge two closest clusters.
   - Update distance matrix.

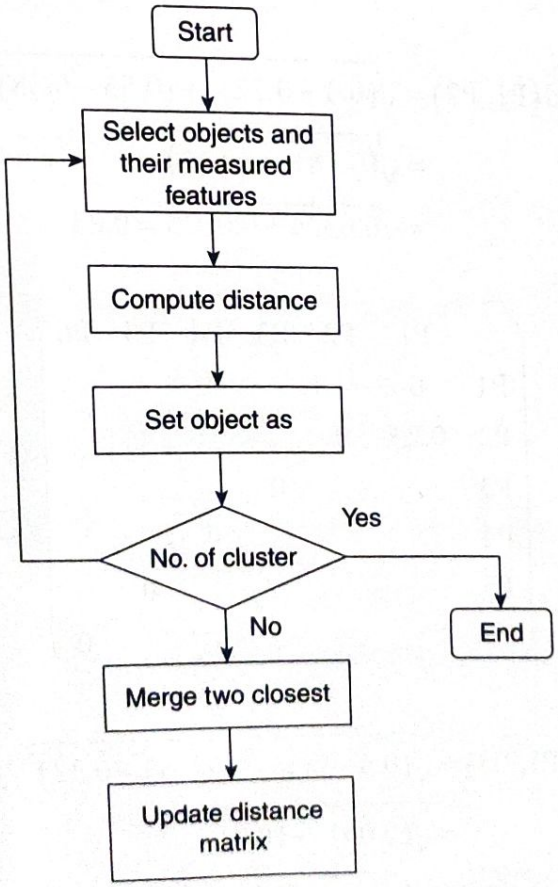

Figure 13.4 Flowchart of agglomerative algorithm.

13.4.1.2 Agglomerative Algorithm: Single Link

Single-nearest distance or single linkage is the agglomerative method that uses the distance between the closest members of the two clusters.

Solved Problem 13.5

Find the clusters using single link technique. Use Euclidean distance and draw the dendrogram.

Sample No.	*X*	*Y*
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.30

Solution:

To compute distance matrix:

$$d[(x,y)(a,b)] = \sqrt{(x-a)^2 + (y-b)^2}$$

Euclidean distance:

$$\begin{aligned} d(P1,P2) &= \sqrt{(0.4-0.22)^2 + (0.53-0.38)^2} \\ &= \sqrt{(0.18)^2 + (0.15)^2} \\ &= \sqrt{0.0324 + 0.0225} = 0.23 \end{aligned}$$

The distance matrix is:

$$\begin{pmatrix} & P1 & P2 & P3 & P4 & P5 & P6 \\ P1 & 0 & & & & & \\ P2 & 0.23 & 0 & & & & \\ P3 & & & 0 & & & \\ P4 & & & & 0 & & \\ P5 & & & & & 0 & \\ P6 & & & & & & 0 \end{pmatrix}$$

Similarly,

$$\begin{aligned} d(P1,P3) &= \sqrt{(0.4-0.35)^2 + (0.53-0.32)^2} \\ &= \sqrt{(0.05)^2 + (0.21)^2} \\ &= \sqrt{0.0025 + 0.0441} = 0.216 = 0.22 \end{aligned}$$

$$d(P1, P4) = \sqrt{(0.4-0.26)^2 + (0.53-0.19)^2}$$
$$= \sqrt{(0.14)^2 + (0.34)^2}$$
$$= \sqrt{0.0196 + 0.1156} = 0.3676 = 0.37$$

$$d(P1, P5) = \sqrt{(0.4-0.08)^2 + (0.53-0.41)^2}$$
$$= \sqrt{(0.32)^2 + (0.12)^2}$$
$$= \sqrt{0.1024 + 0.0144} = 0.3417 = 0.34$$

$$d(P1, P6) = \sqrt{(0.4-0.45)^2 + (0.53-0.30)^2}$$
$$= \sqrt{(0.05)^2 + (0.23)^2}$$
$$= \sqrt{0.0025 + 0.0529} = 0.2354 = 0.24$$

$$d(P2, P3) = \sqrt{(0.22-0.35)^2 + (0.38-0.32)^2}$$
$$= \sqrt{(-0.13)^2 + (0.06)^2}$$
$$= \sqrt{0.0169 + 0.0036} = 0.1432 = 0.14$$

$$d(P2, P4) = \sqrt{(0.22-0.26)^2 + (0.38-0.19)^2}$$
$$= \sqrt{(-0.04)^2 + (0.19)^2}$$
$$= \sqrt{0.0016 + 0.0361} = 0.1942 = 0.19$$

$$d(P2, P5) = \sqrt{(0.22-0.08)^2 + (0.38-0.41)^2}$$
$$= \sqrt{(0.14)^2 + (-0.03)^2}$$
$$= \sqrt{0.0196 + 0.0009} = 0.1432 = 0.14$$

$$d(P2, P6) = \sqrt{(0.22-0.45)^2 + (0.38-0.30)^2}$$
$$= \sqrt{(-0.23)^2 + (0.08)^2}$$
$$= \sqrt{0.0529 + 0.0064} = 0.2435 = 0.24$$

$$d(P3, P4) = \sqrt{(0.35-0.26)^2 + (0.32-0.19)^2}$$
$$= \sqrt{(0.03)^2 + (0.13)^2}$$
$$= \sqrt{0.0009 + 0.0169} = 0.1334 = 0.13$$

$$d(P3, P5) = \sqrt{(0.35-0.08)^2 + (0.32-0.41)^2}$$
$$= \sqrt{(0.27)^2 + (-0.09)^2}$$
$$= \sqrt{0.0729 + 0.0081} = 0.2846 = 0.28$$

$$d(P3, P6) = \sqrt{(0.35-0.45)^2 + (0.32-0.30)^2}$$
$$= \sqrt{(-0.1)^2 + (0.02)^2}$$
$$= \sqrt{0.01 + 0.0004} = 0.10198 = 0.10$$

$$d(P4, P5) = \sqrt{(0.26-0.08)^2 + (0.19-0.41)^2}$$
$$= \sqrt{(0.07)^2 + (-0.22)^2}$$
$$= \sqrt{0.0049 + 0.0484} = 0.2309 = 0.23$$

$$d(P4, P6) = \sqrt{(0.26-0.45)^2 + (0.19-0.30)^2}$$
$$= \sqrt{(-0.19)^2 + (-0.11)^2}$$
$$= \sqrt{0.0361 + 0.0121} = 0.2195 = 0.22$$

$$d(P5, P6) = \sqrt{(0.08-0.45)^2 + (0.41-0.30)^2}$$
$$= \sqrt{(-0.37)^2 + (0.11)^2}$$
$$= \sqrt{0.1369 + 0.0121} = 0.3860 = 0.39$$

The distance matrix is:

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.14	0			
P4	0.37	0.19	0.13	0		
P5	0.34	0.14	0.28	0.23	0	
P6	0.24	0.24	0.10	0.22	0.39	0

Merging the two closest members of the two clusters and finding the minimum element in distance matrix, we get

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.23	0				
P3	0.22	0.14	0			
P4	0.37	0.19	0.13	0		
P5	0.34	0.14	0.28	0.23	0	
P6	0.24	0.24	0.10	0.22	0.39	0

Here the minimum value is 0.10 and hence we combine P3 and P6. Now, form cluster of elements corresponding to minimum value and update distance matrix. To update the distance matrix

min ((P3, P6), P1) = min((P3, P1), (P6, P1)) = min(0.22, 0.24) = 0.22
min ((P3, P6), P2) = min((P3, P2), (P6, P2)) = min(0.14, 0.24) = 0.14
min ((P3, P6), P4) = min((P3, P4), (P6, P4)) = min(0.13, 0.22) = 0.13
min ((P3, P6), P5) = min((P3, P5), (P6, P5)) = min(0.28, 0.39) = 0.28

	P1	P2	P3, P6	P4	P5
P1	0				
P2	0.23	0			
P3, P6	0.22	0.14	0		
P4	0.37	0.19	0.13	0	
P5	0.34	0.14	0.28	0.23	0

Merging two closest members of the two clusters and finding the minimum element in distance matrix.

	P1	P2	P3, P6	P4	P5
P1	0				
P2	0.23	0			
P3, P6	0.22	0.14	0		
P4	0.37	0.19	0.13	0	
P5	0.34	0.14	0.28	0.23	0

Here the minimum value is 0.13 and hence we combine P3, P6 and P4. Now, form cluster of elements corresponding to minimum values and update distance matrix. To update the distance matrix

min (((P3, P6), P4), P1) = min(((P3, P6), P1), (P4, P1)) = min(0.22, 0.37) = 0.22
min (((P3, P6), P4), P2) = min(((P3, P6), P2), (P4, P2)) = min(0.14, 0.19) = 0.14
min (((P3, P6), P4), P5) = min(((P3, P6), P5), (P4, P5)) = min(0.28, 0.23) = 0.23

$$\begin{pmatrix} & P1 & P2 & P3, P6, P4 & P5 \\ P1 & 0 & & & \\ P2 & 0.23 & 0 & & \\ P3, P6, P4 & 0.22 & 0.14 & 0 & \\ P5 & 0.34 & 0.14 & 0.23 & 0 \end{pmatrix}$$

Merging two closest members of the two clusters and finding the minimum element in distance matrix.

$$\begin{pmatrix} & P1 & P2 & P3, P6, P4 & P5 \\ P1 & 0 & & & \\ P2 & 0.23 & 0 & & \\ P3, P6, P4 & 0.22 & 0.14 & 0 & \\ P5 & 0.34 & 0.14 & 0.23 & 0 \end{pmatrix}$$

Here the minimum value is 0.14 and hence we combine P2 and P5. Now, form cluster of elements corresponding to minimum values and update distance matrix. To update the distance matrix

min ((P2, P5), P1) = min((P2, P1), (P5, P1)) = min(0.23, 0.34) = 0.23
min ((P2, P5),(P3, P6, P4)) = min((P2, (P3, P6, P4)), (P5, (P3, P6, P4))) = min(0.14, 0.23) = 0.14

$$\begin{pmatrix} & P1 & P2, P5 & P3, P6, P4 \\ P1 & 0 & & \\ P2, P5 & 0.23 & 0 & \\ P3, P6, P4 & 0.22 & 0.14 & 0 \end{pmatrix}$$

Merging two closest members of the two clusters and finding the minimum element in distance matrix.

$$\begin{pmatrix} & P1 & P2, P5 & P3, P6, P4 \\ P1 & 0 & & \\ P2, P5 & 0.23 & 0 & \\ P3, P6, P4 & 0.22 & 0.14 & 0 \end{pmatrix}$$

Here the minimum value is 0.14 and hence we combine P2, P5 and P3, P6, P4. Now, form cluster of elements corresponding to minimum values and update distance matrix. To update the distance matrix

min ((P2, P5, P3, P6, P4), P1) = min((P2, P5), P1), ((P3, P6, P4), P1)) = min(0.23, 0.22) = 0.22

$$\begin{pmatrix} & P1 & P2, P5, P3, P6, P4 \\ P1 & 0 & \\ P2, P5, P3, P6, P4 & 0.22 & 0 \end{pmatrix}$$

The dendogram can now be drawn as shown in Fig. 13.5.

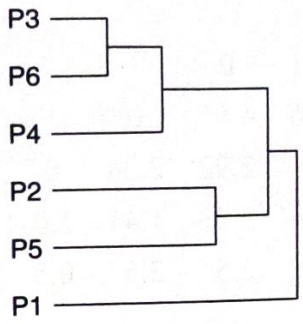

Figure 13.5 Dendogram of the cluster formed.

13.4.1.3 Agglomerative Algorithm: Complete Link

Complete farthest distance or complete linkage is the agglomerative method that uses the distance between the members that are farthest apart.

Solved Problem 13.6

For the given set of points, identify clusters using complete link agglomerative clustering.

Solution:

To compute distance matrix:

$$d\left[(x,y)(a,b)\right]=\sqrt{(x-a)^2+(y-b)^2}$$

The Euclidean distance is:

$$d(P1,P2)=\sqrt{(1.0-1.5)^2+(1.0-1.5)^2}$$
$$=\sqrt{0.25+0.25}=\sqrt{0.5}=0.71$$

The distance matrix is:

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.71	0				
P3	5.66	4.95	0			
P4	3.6	2.92	2.24	0		
P5	4.24	3.53	1.41	1.0	0	
P6	3.20	2.5	2.5	0.5	1.12	0

Merging two closest members of the two clusters and finding the minimum element in distance matrix and forming the clusters, we get

$$\begin{pmatrix} & P1 & P2 & P3 & P4 & P5 & P6 \\ P1 & 0 & & & & & \\ P2 & 0.71 & 0 & & & & \\ P3 & 5.66 & 4.95 & 0 & & & \\ P4 & 3.6 & 2.92 & 2.24 & 0 & & \\ P5 & 4.24 & 3.53 & 1.41 & 1.0 & 0 & \\ P6 & 3.20 & 2.5 & 2.5 & 0.5 & 1.12 & 0 \end{pmatrix}$$

Here the minimum value is 0.5 and hence we combine P4 and P6. To update the distance matrix
max (d(P4, P6), P1)) = max(d(P4, P1), d(P6, P1)) = max(3.6, 3.2) = 3.6

$$\begin{pmatrix} & P1 & P2 & P3 & P4, P6 & P5 \\ P1 & 0 & & & & \\ P2 & 0.71 & 0 & & & \\ P3 & 5.66 & 4.95 & 0 & & \\ P4, P6 & 3.6 & 2.92 & 2.5 & 0 & \\ P5 & 4.24 & 3.53 & 1.41 & 1.12 & 0 \end{pmatrix}$$

Merging two closest by finding the minimum element in distance matrix and forming the clusters, we get

$$\begin{pmatrix} & P1, P2 & P3 & P4, P6 & P5 \\ P1, P2 & 0 & & & \\ P3 & 5.66 & 0 & & \\ P4, P6 & 3.6 & 2.5 & 0 & \\ P5 & 4.24 & 1.41 & 1.12 & 0 \end{pmatrix}$$

Merging two closest by finding the minimum element in distance matrix and forming the clusters, we get

$$\begin{pmatrix} & P1, P2 & P3 & P4, P6, P5 \\ P1, P2 & 0 & & \\ P3 & 5.66 & 0 & \\ P4, P6, P5 & 3.6 & 2.5 & 0 \end{pmatrix}$$

Merging two closest by finding the minimum element in distance matrix and forming the clusters, we get

$$\begin{pmatrix} & P1, P2 & P4, P6, P5, P3 \\ P1, P2 & 0 & \\ P4, P6, P5, P3 & 5.66 & 0 \end{pmatrix}$$

The final cluster formed can now be drawn as shown in Fig. 13.6.

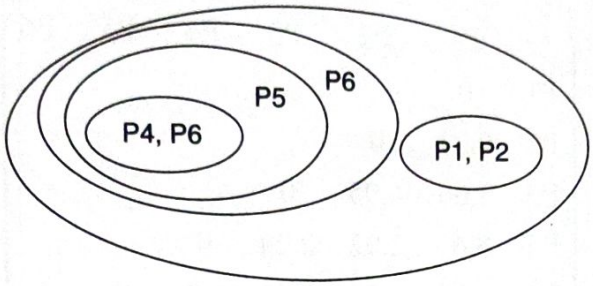

Figure 13.6 Cluster formed after merging all data points.

13.4.1.4 Agglomerative Algorithm: Average Link

Average–average distance or average linkage is the method that involves looking at the distances between all pairs and averages all of these distances. This is also called *Unweighted Pair Group Mean Averaging*.

Solved Problem 13.7

For the given set of points, identify clusters using average link agglomerative clustering.

	A	*B*
P1	1	1
P2	1.5	1.5
P3	5	5
P4	3	4
P5	4	4
P6	3	3.5

Solution:

The distance matrix is:

$$\begin{pmatrix} & P1 & P2 & P3 & P4 & P5 & P6 \\ P1 & 0 & & & & & \\ P2 & 0.71 & 0 & & & & \\ P3 & 5.66 & 4.95 & 0 & & & \\ P4 & 3.6 & 2.92 & 2.24 & 0 & & \\ P5 & 4.24 & 3.53 & 1.41 & 1.0 & 0 & \\ P6 & 3.20 & 2.5 & 2.5 & 0.5 & 1.12 & 0 \end{pmatrix}$$

Merging two closest members of the two clusters and finding the minimum element in distance matrix, we get

	P1	P2	P3	P4	P5	P6
P1	0					
P2	0.71	0				
P3	5.66	4.95	0			
P4	3.6	2.92	2.24	0		
P5	4.24	3.53	1.41	1.0	0	
P6	3.20	2.5	2.5	0.5	1.12	0

Here the minimum value is 0.5 and hence we combine P4 and P6. To update the distance matrix average (d(P4, P6), P1)) = average(d(P4, P1), d(P6, P1)) = average(3.6, 3.2) = 3.4

	P1	P2	P3	P4,P6	P5
P1	0				
P2	0.71	0			
P3	5.66	4.95	0		
P4,P6	3.2	2.71	2.37	0	
P5	4.24	3.53	1.41	1.06	0

Merging two closest by finding the minimum element in distance matrix and forming the clusters:

	P1,P2	P3	P4,P6	P5
P1,P2	0			
P3	5.31	0		
P4,P6	2.96	2.5	0	
P5	3.89	1.41	1.12	0

Merging two closest by finding the minimum element in distance matrix and forming the clusters:

	P1,P2	P3	P4,P6,P5
P1,P2	0		
P3	5.66	0	
P4,P6,P5	3.43	1.96	0

Merging two closest by finding the minimum element in distance matrix and forming the clusters:

	P1,P2	P4,P6,P5,P3
P1,P2	0	
P4,P6,P5,P3	4.55	0

The final cluster formed can now be drawn as shown in Fig. 13.7.

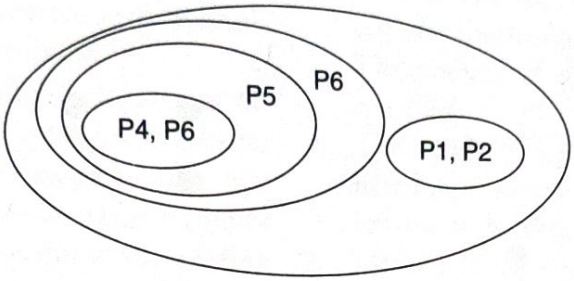

Figure 13.7 The final cluster formed merging all data points.

Case Study

There are diverse applications using clustering. As discussed in previous sections, the concept of clustering is one wherein the output is not known. In other words, the training dataset has only input data samples. So based on some metrics of grouping or clustering, similar data items are grouped together in one cluster. Let us now see some of the major areas wherein this concept is extensively used.

Case Study 1: Grouping of Similar Companies Based on Wikipedia Articles

The *k*-means clustering algorithm is used to segment S&P (500 index) listed companies based on the text of Wikipedia articles about each one. Initially, the data is taken and preprocessed. This data is nothing but articles from Wikipedia about the companies. In the preprocessing phase, the Wikipedia formatting is removed, all texts are converted to lowercase, and non-alphanumeric characters are removed.

Around 500 companies were taken as input data. From this input data, feature hashing module tokenizes the text string and transforms the data into a series of numbers based on the hash value of each token. No lingusitic analysis is performed in this step. Internally, the feature hashing module creates a dictionary of n-grams. After the dictionary has been built, the feature hashing module converts the dictionary terms into hash values. It then computes whether a feature was used in each case. For each row of text data, the module outputs a set of columns – one column for each hashed feature. Here, the dimentionality was reduced using PCA. So based on the highest variance, the first one or two columns of the transformed matrix is selected and clusters were built using the transformed dataset.

Case Study 2: Telecom Cluster Analysis

We know that there are a number of telecom companies attract customers with a variety of packages. In the telecom sector, you have to realize that not every customer has similar needs and you need to strategize accordingly to attract all of them. Based on customer segmentation, the company as well as customers can have a the win–win scenario. Taking a sample of customers and based on their international and national call durations, clustering techniques are used to classify clusters into two main categories.