$$\sigma_i^{2(\text{new})} = \frac{1}{N_i}\left(\gamma_{i1}(x_1 - \mu_i^{(\text{new})})^2 + \cdots + \gamma_{iN}(x_1 - \mu_i^{(\text{new})})^2\right)$$

$$\pi_i^{(\text{new})} = \frac{N_i}{N}$$

Step 4. Evaluate the log-likelihood function given in Eq.(13.11) and check for convergence of either the parameters or the log-likelihood function. If the convergence criterion is not satisfied, return to Step 2.

13.8 Hierarchical clustering

Hierarchical clustering (also called hierarchical cluster analysis or HCA) is a method of cluster analysis which seeks to build a hierarchy of clusters (or groups) in a given dataset. The hierarchical clustering produces clusters in which the clusters at each level of the hierarchy are created by merging clusters at the next lower level. At the lowest level, each cluster contains a single observation. At the highest level there is only one cluster containing all of the data.

The decision regarding whether two clusters are to be merged or not is taken based on the *measure of dissimilarity* between the clusters. The distance between two clusters is usually taken as the measure of dissimilarity between the clusters.

In Section **??**, we shall see various methods for measuring the distance between two clusters.

13.8.1 Dendrograms

Hierarchical clustering can be represented by a rooted binary tree. The nodes of the trees represent groups or clusters. The root node represents the entire data set. The terminal nodes each represent one of the individual observations (singleton clusters). Each nonterminal node has two daughter nodes.

The distance between merged clusters is monotone increasing with the level of the merger. The height of each node above the level of the terminal nodes in the tree is proportional to the value of the distance between its two daughters (see Figure 13.9).

A *dendrogram* is a tree diagram used to illustrate the arrangement of the clusters produced by hierarchical clustering.

The dendrogram may be drawn with the root node at the top and the branches growing vertically downwards (see Figure 13.8(a)). It may also be drawn with the root node at the left and the branches growing horizontally rightwards (see Figure 13.8(b)). In some contexts, the opposite directions may also be more appropriate.

Dendrograms are commonly used in computational biology to illustrate the clustering of genes or samples.

Example

Figure 13.7 is a dendrogram of the dataset $\{a, b, c, d, e\}$. Note that the root node represents the entire dataset and the terminal nodes represent the individual observations. However, the dendrograms are presented in a simplified format in which only the terminal nodes (that is, the nodes representing the singleton clusters) are explicitly displayed. Figure 13.8 shows the simplified format of the dendrogram in Figure 13.7.

Figure 13.9 shows the distances of the clusters at the various levels. Note that the clusters are at 4 levels. The distance between the clusters $\{a\}$ and $\{b\}$ is 15, between $\{c\}$ and $\{d\}$ is 7.5, between $\{c, d\}$ and $\{e\}$ is 15 and between $\{a, b\}$ and $\{c, d, e\}$ is 25.

13.8.2 Methods for hierarchical clustering

There are two methods for the hierarchical clustering of a dataset. These are known as the *agglomerative method* (or the bottom-up method) and the *divisive method* (or, the top-down method).

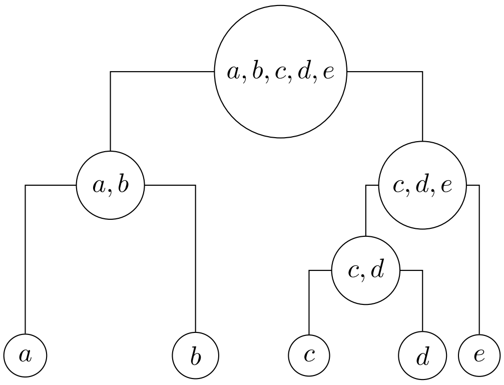


Figure 13.7: A dendrogram of the dataset $\{a, b, c, d, e\}$

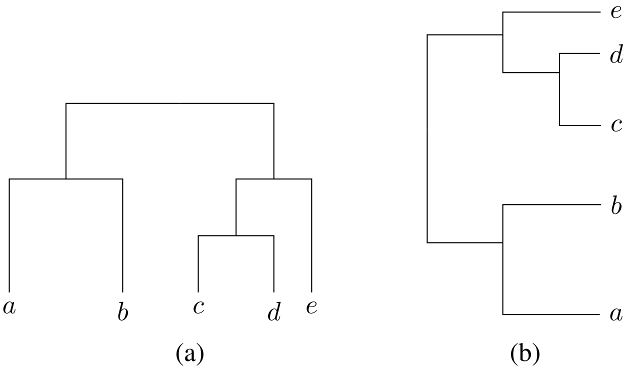


Figure 13.8: Different ways of drawing dendrogram

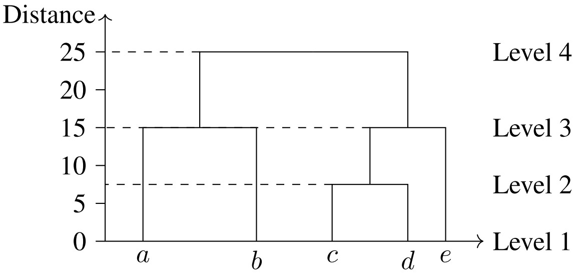
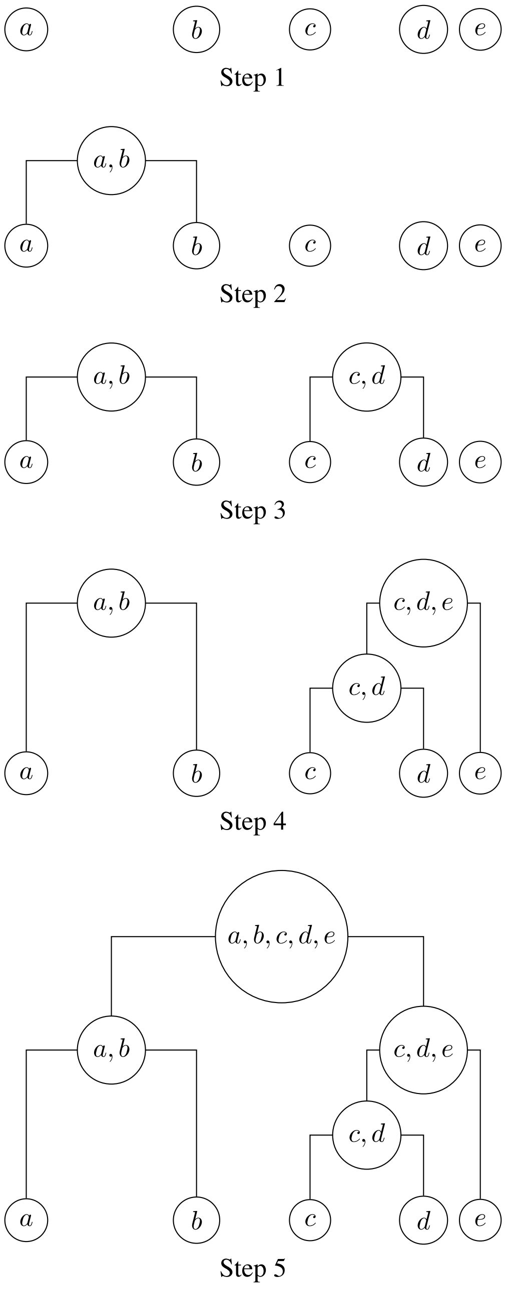


Figure 13.9: A dendrogram of the dataset $\{a, b, c, d, e\}$ showing the distances (heights) of the clusters at different levels

Agglomerative method

In the agglomerative we start at the bottom and at each level recursively merge a selected pair of clusters into a single cluster. This produces a grouping at the next higher level with one less cluster. If there are N observations in the dataset, there will be $N-1$ levels in the hierarchy. The pair chosen for merging consist of the two groups with the smallest "intergroup dissimilarity".

For example, the hierarchical clustering shown in Figure 13.7 can be constructed by the agglomerative method as shown in Figure 13.10. Each nonterminal node has two daughter nodes. The daughters represent the two groups that were merged to form the parent.



Step 1

Step 2

Step 3

Step 4

Step 5

Figure 13.10: Hierarchical clustering using agglomerative method

Divisive method

The divisive method starts at the top and at each level recursively split one of the existing clusters at that level into two new clusters. If there are N observations in the dataset, there the divisive method also will produce $N - 1$ levels in the hierarchy. The split is chosen to produce two new groups with the largest "between-group dissimilarity".

For example, the hierarchical clustering shown in Figure 13.7 can be constructed by the divisive method as shown in Figure 13.11. Each nonterminal node has two daughter nodes. The two daughters represent the two groups resulting from the split of the parent.

13.9 Measures of dissimilarity

In order to decide which clusters should be combined (for agglomerative), or where a cluster should be split (for divisive), a measure of dissimilarity between sets of observations is required. In most methods of hierarchical clustering, the dissimilarity between two groups of observations is measured by using an appropriate measure of distance between the groups of observations. The distance between two groups of observations is defined in terms of the distance between two observations. There are several ways in which the distance between two observations can be defined and also there are also several ways in which the distance between two groups of observations can be defined.

13.9.1 Measures of distance between data points

Numeric data

We assume that each observation or data point is a n-dimensional vector. Let $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{y} = (y_1, \ldots, y_n)$ be two observations. Then the following are the commonly used measures of distances in the hierarchical clustering of numeric data.

Name	Formula
Euclidean distance	$\lVert\vec{x} - \vec{y}\rVert_2 = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}$
Squared Euclidean distance	$\lVert\vec{x} - \vec{y}\rVert_2^2 = (x_1 - y_1)^2 + \cdots + (x_n - y_n)^2$
Manhattan distance	$\lVert\vec{x} - \vec{y}\rVert_1 = \lvert x_1 - y_1\rvert + \cdots + \lvert x_n - y_n\rvert$
Maximum distance	$\lVert\vec{x} - \vec{y}\rVert_\infty = \max\{\lvert x_1 - y_1\rvert, \ldots, \lvert x_n - y_n\rvert\}$

Non-numeric data

For text or other non-numeric data, metrics such as the Levenshtein distance are often used.

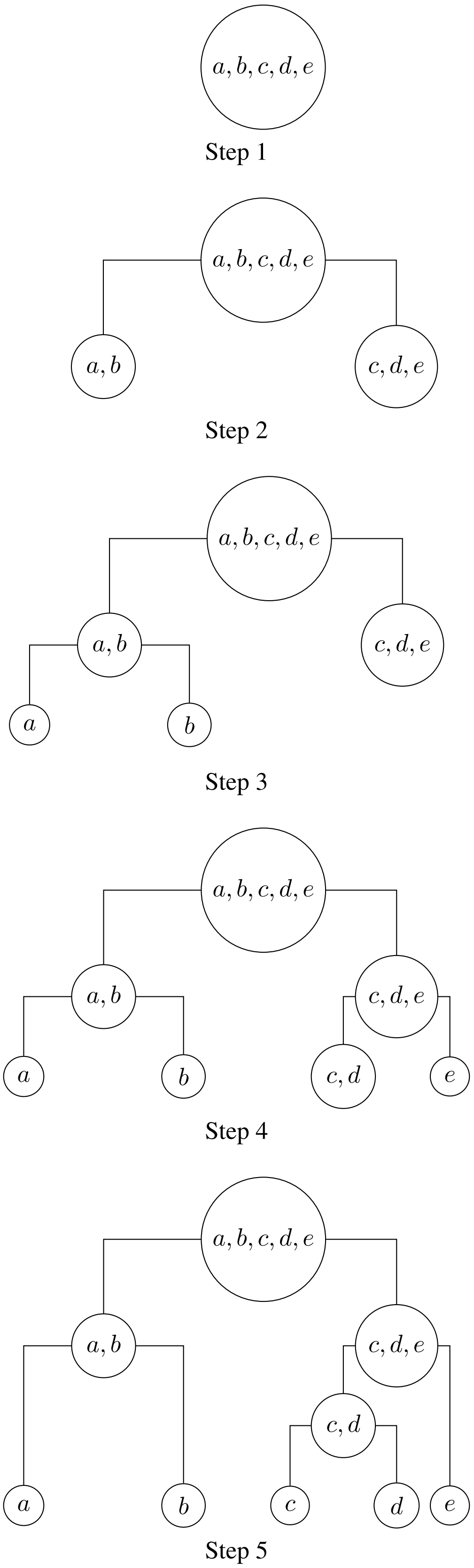
The *Levenshtein distance* is a measure of the "distance" between two words. The Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other.

For example, the Levenshtein distance between "kitten" and "sitting" is 3, since the following three edits change one into the other, and there is no way to do it with fewer than three edits:

kitten → sitten (substitution of "s" for "k")
sitten → sittin (substitution of "i" for "e")
sittin → sitting (insertion of'g" at the end)

13.9.2 Measures of distance between groups of data points

Let A and B be two groups of observations and let x and y be arbitrary data points in A and B respectively. Suppose we have chosen some formula, say Euclidean distance formula, to measure the distance between data points. Let $d(x, y)$ denote the distance between x and y. We denote by



Step 1

Step 2

Step 3

Step 4

Step 5

Figure 13.11: Hierarchical clustering using divisive method

$d(A, B)$ the distance between the groups A and B. The following are some of the different methods in which $d(A, B)$ is defined.

1. $d(A, B) = \max\{d(x, y) : x \in A, y \in B\}$.

   Agglomerative hierarchical clustering using this measure of dissimilarity is known as *complete-linkage clustering*. The method is also known as *farthest neighbour clustering*.

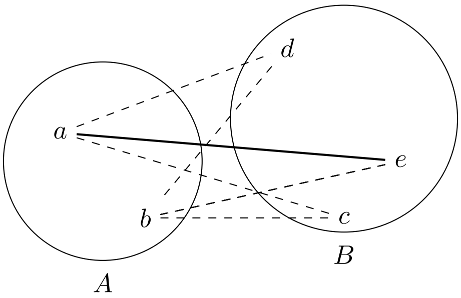


Figure 13.12: Length of the solid line "ae" is $\max\{d(x, y) : x \in A, y \in B\}$

2. $d(A, B) = \min\{d(x, y) : x \in A, y \in B\}$.

   Agglomerative hierarchical clustering using this measure of dissimilarity is known as *single-linkage clustering*. The method is also known as *nearest neighbour clustering*.

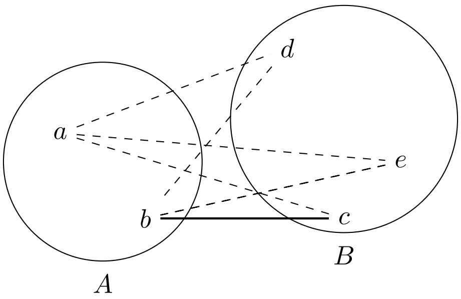


Figure 13.13: Length of the solid line "bc" is $\min\{d(x, y) : x \in A, y \in B\}$

3. $d(A, B) = \dfrac{1}{|A|\,|B|} \sum_{x \in A, y \in B} d(x, y)$ where $|A|$, $|B|$ are respectively the number of elements in A and B.

   Agglomerative hierarchical clustering using this measure of dissimilarity is known as *mean or average linkage clustering*. It is also known as UPGMA (Unweighted Pair Group Method with Arithmetic Mean).

13.10 Algorithm for agglomerative hierarchical clustering

Given a set of N items to be clustered and an $N \times N$ distance matrix, required to construct a hierarchical clustering of the data using the agglomerative method.

Step 1. Start by assigning each item to its own cluster, so that we have N clusters, each containing just one item. Let the distances between the clusters equal the distances between the items they contain.

Step 2. Find the closest pair of clusters and merge them into a single cluster, so that now we have one less cluster.

Step 3. Compute distances between the new cluster and each of the old clusters.

Step 4. Repeat Steps 2 and 3 until all items are clustered into a single cluster of size N.

13.10.1 Example

Problem 1

Given the dataset $\{a, b, c, d, e\}$ and the following distance matrix, construct a dendrogram by complete-linkage hierarchical clustering using the agglomerative method.

	a	b	c	d	e
a	0	9	3	6	11
b	9	0	7	5	10
c	3	7	0	9	2
d	6	5	9	0	8
e	11	10	2	8	0

Table 13.4: Example for distance matrix

Solution

The complete-linkage clustering uses the "maximum formula", that is, the following formula to compute the distance between two clusters A and B:

$$d(A, B) = \max\{d(x, y) : x \in A, y \in B\}$$

1. Dataset : $\{a, b, c, d, e\}$.

   Initial clustering (singleton sets) C_1: $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$.

2. The following table gives the distances between the various clusters in C_1:

	$\{a\}$	$\{b\}$	$\{c\}$	$\{d\}$	$\{e\}$
$\{a\}$	0	9	3	6	11
$\{b\}$	9	0	7	5	10
$\{c\}$	3	7	0	9	**2**
$\{d\}$	6	5	9	0	8
$\{e\}$	11	10	**2**	8	0

   In the above table, the minimum distance is the distance between the clusters $\{c\}$ and $\{e\}$. Also

$$d(\{c\}, \{e\}) = 2.$$

   We merge $\{c\}$ and $\{e\}$ to form the cluster $\{c, e\}$.

   The new set of clusters C_2: $\{a\}$, $\{b\}$, $\{d\}$, $\{c, e\}$.

3. Let us compute the distance of $\{c, e\}$ from other clusters.

   $d(\{c, e\}, \{a\}) = \max\{d(c, a), d(e, a)\} = \max\{3, 11\} = 11.$

   $d(\{c, e\}, \{b\}) = \max\{d(c, b), d(e, b)\} = \max\{7, 10\} = 10.$

   $d(\{c, e\}, \{d\}) = \max\{d(c, d), d(e, d)\} = \max\{9, 8\} = 9.$

   The following table gives the distances between the various clusters in C_2.

	$\{a\}$	$\{b\}$	$\{d\}$	$\{c,e\}$
$\{a\}$	0	9	6	11
$\{b\}$	9	0	**5**	10
$\{d\}$	6	**5**	0	9
$\{c,e\}$	11	10	9	0

In the above table, the minimum distance is the distance between the clusters $\{b\}$ and $\{d\}$. Also

$$d(\{b\},\{d\}) = 5.$$

We merge $\{b\}$ and $\{d\}$ to form the cluster $\{b,d\}$.

The new set of clusters C_3: $\{a\}$, $\{b,d\}$, $\{c,e\}$.

4. Let us compute the distance of $\{b,d\}$ from other clusters.

   $d(\{b,d\},\{a\}) = \max\{d(b,a), d(d,a)\} = \max\{9,6\} = 9.$

   $d(\{b,d\},\{c,e\}) = \max\{d(b,c), d(b,e), d(d,c), d(d,e)\} = \max\{7,10,9,8\} = 10.$

   The following table gives the distances between the various clusters in C_3.

	$\{a\}$	$\{b,d\}$	$\{c,e\}$
$\{a\}$	0	**9**	11
$\{b,d\}$	**9**	0	10
$\{c,e\}$	11	10	0

   In the above table, the minimum distance is the distance between the clusters $\{a\}$ and $\{b,d\}$. Also

$$d(\{a\},\{b,d\}) = 9.$$

   We merge $\{a\}$ and $\{b,d\}$ to form the cluster $\{a,b,d\}$.

   The new set of clusters C_4: $\{a,b,d\}$, $\{c,e\}$

5. Only two clusters are left. We merge them form a single cluster containing all data points. We have

$$\begin{aligned} d(\{a,b,d\},\{c,e\}) &= \max\{d(a,c), d(a,e), d(b,c), d(b,e), d(d,c), d(d,e)\} \\ &= \max\{3,11,7,10,9,8\} \\ &= 11 \end{aligned}$$

6. Figure 13.14 shows the dendrogram of the hierarchical clustering.

Problem 2

Given the dataset $\{a,b,c,d,e\}$ and the distance matrix given in Table 13.4, construct a dendrogram by single-linkage hierarchical clustering using the agglomerative method.

Solution

The complete-linkage clustering uses the "maximum formula", that is, the following formula to compute the distance between two clusters A and B:

$$d(A,B) = \min\{d(x,y) : x \in A, y \in B\}$$

1. Dataset : $\{a,b,c,d,e\}$.

   Initial clustering (singleton sets) C_1: $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\{e\}$.

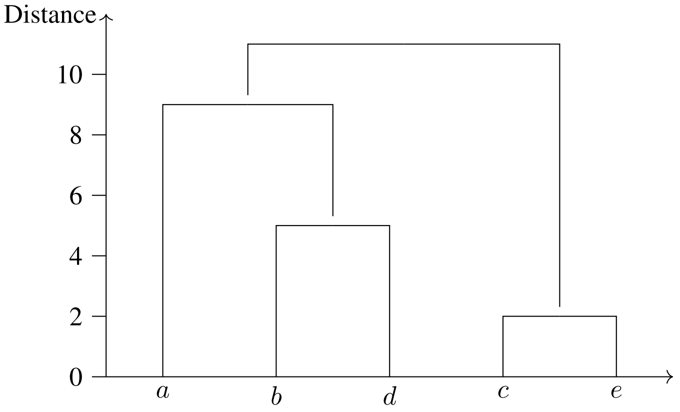


Figure 13.14: Dendrogram for the data given in Table 13.4 (complete linkage clustering)

2. The following table gives the distances between the various clusters in C_1:

	$\{a\}$	$\{b\}$	$\{c\}$	$\{d\}$	$\{e\}$
$\{a\}$	0	9	3	6	11
$\{b\}$	9	0	7	5	10
$\{c\}$	3	7	0	9	**2**
$\{d\}$	6	5	9	0	8
$\{e\}$	11	10	**2**	8	0

In the above table, the minimum distance is the distance between the clusters $\{c\}$ and $\{e\}$. Also

$$d(\{c\},\{e\}) = 2.$$

We merge $\{c\}$ and $\{e\}$ to form the cluster $\{c, e\}$.

The new set of clusters C_2: $\{a\}$, $\{b\}$, $\{d\}$, $\{c, e\}$.

3. Let us compute the distance of $\{c, e\}$ from other clusters.

$d(\{c, e\}, \{a\}) = \min\{d(c, a), d(e, a)\} = \max\{3, 11\} = 3.$

$d(\{c, e\}, \{b\}) = \min\{d(c, b), d(e, b)\} = \max\{7, 10\} = 7.$

$d(\{c, e\}, \{d\}) = \min\{d(c, d), d(e, d)\} = \max\{9, 8\} = 8.$

The following table gives the distances between the various clusters in C_2.

	$\{a\}$	$\{b\}$	$\{d\}$	$\{c, e\}$
$\{a\}$	0	9	6	**3**
$\{b\}$	9	0	5	7
$\{d\}$	6	5	0	8
$\{c, e\}$	**3**	7	8	0

In the above table, the minimum distance is the distance between the clusters $\{a\}$ and $\{c, e\}$. Also

$$d(\{a\},\{c, e\}) = 3.$$

We merge $\{a\}$ and $\{c, e\}$ to form the cluster $\{a, c, e\}$.

The new set of clusters C_3: $\{a, c, e\}$, $\{b\}$, $\{d\}$.

4. Let us compute the distance of $\{a, c, e\}$ from other clusters.

   $d(\{a, c, e\}, \{b\}) = \min\{d(a, b), d(c, b), d(e, b)\} = \{9, 7, 10\} = 7$

   $d(\{a, c, e\}, \{d\}) = \min\{d(a, d), d(c, d), d(e, d)\} = \{6, 9, 8\} = 6$

   The following table gives the distances between the various clusters in C_3.

	$\{a, c, e\}$	$\{b\}$	$\{d\}$
$\{a, c, e\}$	0	7	6
$\{b\}$	7	0	**5**
$\{d\}$	6	**5**	0

   In the above table, the minimum distance is between $\{b\}$ and $\{d\}$. Also

   $$d(\{b\}, \{d\}) = 5.$$

   We merge $\{b\}$ and $\{d\}$ to form the cluster $\{b, d\}$.

   The new set of clusters C_4: $\{a, c, e\}$, $\{b, d\}$

5. Only two clusters are left. We merge them form a single cluster containing all data points. We have

   $$\begin{aligned} d(\{a, c, e\}, \{b, d\}) &= \min\{d(a, b), d(a, d), d(c, b), d(c, d), d(e, b), d(e, d)\} \\ &= \min\{9, 6, 7, 9, 10, 8\} \\ &= 6 \end{aligned}$$

6. Figure 13.15 shows the dendrogram of the hierarchical clustering.

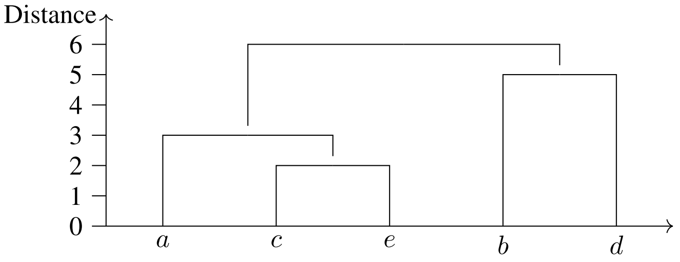


Figure 13.15: Dendrogram for the data given in Table 13.4 (single linkage clustering)

13.11 Algorithm for divisive hierarchical clustering

Divisive clustering algorithms begin with the entire data set as a single cluster, and recursively divide one of the existing clusters into two daughter clusters at each iteration in a top-down fashion. To apply this procedure, we need a separate algorithm to divide a given dataset into two clusters.

- The divisive algorithm may be implemented by using the k-means algorithm with $k = 2$ to perform the splits at each iteration. However, it would not necessarily produce a splitting sequence that possesses the monotonicity property required for dendrogram representation.

13.11.1 DIANA (DIvisive ANAlysis)

DIANA is a divisive hierarchical clustering technique. Here is an outline of the algorithm.

Step 1. Suppose that cluster C_l is going to be split into clusters C_i and C_j.

Step 2. Let $C_i = C_l$ and $C_j = \emptyset$.

Step 3. For each object $x \in C_i$:

(a) For the first iteration, compute the average distance of x to all other objects.

(b) For the remaining iterations, compute

$$D_x = \text{average}\,\{d(x,y) : y \in C_i\} - \text{average}\{d(x,y) : y \in C_j\}.$$

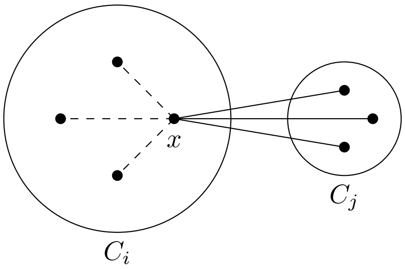


Figure 13.16: D_x= (average of dashed lines) − (average of solid lines)

Step 4. (a) For the first iteration, move the object with the maximum average distance to C_j.

(b) For the remaining iterations, find an object x in C_i for which D_x is the largest. If $D_x > 0$ then move x to C_j.

Step 5. Repeat Steps 3(b) and 4(b) until all differences D_x are negative. Then C_l is split into C_i and C_j.

Step 6. Select the smaller cluster with the largest diameter. (The diameter of a cluster is the largest dissimilarity between any two of its objects.) Then divide this cluster, following Steps 1-5.

Step 7. Repeat Step 6 until all clusters contain only a single object.

13.11.2 Example

Problem

Given the dataset $\{a,b,c,d,e\}$ and the distance matrix in Table 13.4, construct a dendrogram by the divisive analysis algorithm.

Solution

1. We have, initially
$$C_l = \{a,b,c,d,e\}$$

2. We write
$$C_i = C_l, \quad C_j = \emptyset.$$

3. Division into clusters

(a) Initial iteration

Let us calculate the average dissimilarities of the objects in C_i with the other objects in C_i.

Average dissimilarity of a

$$= \frac{1}{4}(d(a,b) + d(a,c) + d(a,e)) = \frac{1}{4}(9+3+6+11) = 7.25$$

Similarly we have :

Average dissimilarity of $b = 7.75$

Average dissimilarity of $c = 5.25$

Average dissimilarity of $d = 7.00$

Average dissimilarity of $e = 7.75$

The highest average distance is 7.75 and there are two corresponding objects. We choose one of them, b, arbitrarily. We move b to C_j.

We now have

$$C_i = \{a,c,d,e\}, \quad C_j = \varnothing \cup \{b\} = \{b\}.$$

(b) Remaining iterations

(i) 2-nd iteration.

$$\begin{aligned}
D_a &= \frac{1}{3}(d(a,c) + d(a,d) + d(a,e)) - \frac{1}{1}(d(a,b)) = \frac{20}{3} - 9 = -2.33\\
D_c &= \frac{1}{3}(d(c,a) + d(c,d) + d(c,e)) - \frac{1}{1}(d(c,b)) = \frac{14}{3} - 7 = -2.33\\
D_d &= \frac{1}{3}(d(d,a) + d(d,c) + d(d,e)) - \frac{1}{1}(d(c,b)) = \frac{23}{3} - 7 = 0.67\\
D_e &= \frac{1}{3}(d(e,a) + d(e,c) + d(e,d)) - \frac{1}{1}(d(e,b)) = \frac{21}{3} - 7 = 0
\end{aligned}$$

D_d is the largest and $D_d > 0$. So we move, d to C_j.

We now have

$$C_i = \{a,c,e\}, \quad C_j = \{b\} \cup \{d\} = \{b,d\}.$$

(ii) 3-rd iteration

$$\begin{aligned}
D_a &= \frac{1}{2}(d(a,c) + d(a,e)) - \frac{1}{2}(d(a,b) + d(a,d)) = \frac{14}{2} - \frac{15}{2} = -0.5\\
D_c &= \frac{1}{2}(d(c,a) + d(c,e)) - \frac{1}{2}(d(c,b) + d(c,d)) = \frac{5}{2} - \frac{16}{2} = -13.5\\
D_e &= \frac{1}{2}(d(e,a) + d(e,c)) - \frac{1}{2}(d(e,b) + d(e,d)) = \frac{13}{2} - \frac{18}{2} = -2.5
\end{aligned}$$

All are negative. So we stop and form the clusters C_i and C_j.

4. To divide, C_i and C_j, we compute their diameters.

$$\begin{aligned}
\text{diameter}(C_i) &= \max\{d(a,c), d(a,e), d(c,e)\}\\
&= \max\{3, 11, 2\}\\
&= 11\\
\text{diameter}(C_j) &= \max\{d(b,d)\}\\
&= 5
\end{aligned}$$

The cluster with the largest diameter is C_i. So we now split C_i.

We repeat the process by taking $C_l = \{a,c,e\}$. The remaining computations are left as an exercise to the reader.

13.12 Density-based clustering

In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set. Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points. The most popular density based clustering method is DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

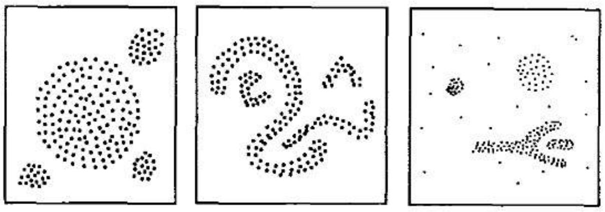


Figure 13.17: Clusters of points and noise points not belonging to any of those clusters

13.12.1 Density

We introduce some terminology and notations.

- Let ϵ (epsilon) be some constant distance. Let p be an arbitrary data point. The *ϵ-neighbourhood of p* is the set
$$N_\epsilon(p) = \{q : d(p,q) < \epsilon\}$$
- We choose some number m_0 to define points of "high density": We say that a point p is point of *high density* if $N_\epsilon(p)$ contains at least m_0 points.
- We define a point p as a *core point* if $N_\epsilon(p)$ has more than m_0 points.
- We define a point p as a *border point* if $N_\epsilon(p)$ has fewer than m_0 points, but is in the ϵ-neighbourhood of a core point.
- A point which is neither a core point nor a border point is called a *noise point*.

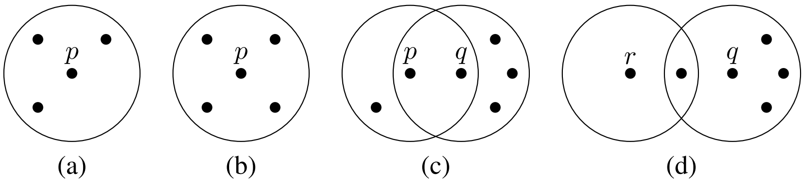


Figure 13.18: With $m_0 = 4$: (a) p a point of high density (b) p a core point (c) p a border point (d) r a noise point

- An object q is *directly density-reachable* from object p if p is a core object and q is in $N_\epsilon(p)$.
- An object q is *indirectly density-reachable* from an object p if there is a finite set of objects $p_1, \ldots, p_r$ such that p_1 is directly density-reachable form p, p_2 is directly density reachable from p_1, etc., q is directly density-reachable form p_r.

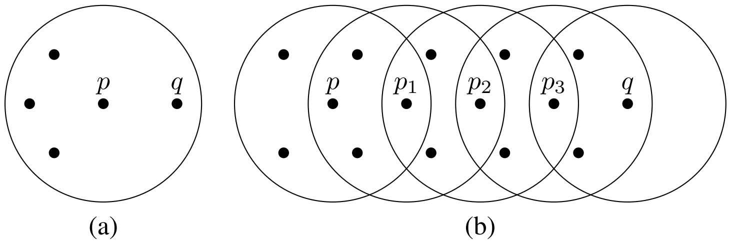


Figure 13.19: With m_0 = 4: (a) q is directly density-reachable from p (b) q is indirectly density-reachable from p

13.12.2 DBSCAN algorithm

Let $X = \{x_1, x_2, \ldots, x_n\}$ be the set of data points. DBSCAN requires two parameters: ϵ (eps) and the minimum number of points required to form a cluster (m_0).

Step 1. Start with an arbitrary starting point p that has not been visited.

Step 2. Extract the ϵ-neighborhood $N_\epsilon(p)$ of p.

Step 3. If the number of points in $N_\epsilon(p)$ is not greater than m_0 then the point p is labeled as noise (later this point can become the part of the cluster).

Step 4. If the number of points in $N_\epsilon(p)$ is greater than m_0 then the point p is a core point and is marked as visited. Select a new *cluster-id* and mark all objects in $N_\epsilon(p)$ with this cluster-id.

Step 5. If a point is found to be a part of the cluster then its ϵ-neighborhood is also the part of the cluster and the above procedure from step 2 is repeated for all ϵ-neighborhood points. This is repeated until all points in the cluster are determined.

Step 6. A new unvisited point is retrieved and processed, leading to the discovery of a further cluster or noise.

Step 7. This process continues until all points are marked as visited.

13.13 Sample questions

(a) Short answer questions

1. What is clustering?
2. Is clustering supervised learning? Why?
3. Explain some applications of the k-means algorithm.
4. Explain how clustering technique is used in image segmentation problem.
5. Explain how clustering technique used in data compression.
6. What is meant by the mixture of two normal distributions?
7. Explain hierarchical clustering.
8. What is a dendrogram? Give an example.
9. Is hierarchical clustering unsupervised learning? Why?
10. Describe the two methods for hierarchical clustering.

11. In a clustering problem, what does the measure of dissimilarity measure? Give some examples of measures of dissimilarity.

12. Explain the different types of linkages in clustering.

13. In the context of density-based clustering, define high density point, core point, border point and noise point.

14. What is agglomerative hierarchical clustering?

(b) Long answer questions

1. Apply k-means algorithm for given data with k = 3. Use $C_1(2)$, $C_2(16)$ and $C_3(38)$ as initial centers. Data:

$$2, 4, 6, 3, 31, 12, 15, 16, 38, 35, 14, 21, 3, 25, 30$$

2. Explain K-means algorithm and group the points (1, 0, 1), (1, 1, 0), (0, 0, 1) and (1, 1, 1) using K-means algorithm.

3. Applying the k-means algorithm, find two clusters in the following data.

x	185	170	168	179	182	188	180	180	183	180	180	177
y	72	56	60	68	72	77	71	70	84	88	67	76

4. Use k-means algorithm to find 2 clusters in the following data:

No.	1	2	3	4	5	6	7
x_1	1.0	1.5	3.0	5.0	3.5	4.5	3.5
x_2	1.0	2.0	4.0	7.0	5.0	5.0	4.5

5. Give a general outline of the expectation-maximization algorithm.

6. Describe EM algorithm for Gaussian mixtures.

7. Describe an algorithm for agglomerative hierarchical clustering.

8. Given the following distance matrix, construct the dendrogram using agglomerative clustering with single linkage, complete linkage and average linkage.

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0

9. Describe an algorithm for divisive hierarchical clustering.

10. For the data in Question 8, construct a dendrogram using DIANA algorithm.

11. Describe the DBSCAN algorithm for clustering.

Bibliography

[1] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[2] Ethem Alpaydin, *Introduction to Machine Learning*, The MIT Press, Cambridge, Massachusetts, 2004.

[3] Margaret H. Dunham, *Data Mining: Introductory and Advanced Topics*, Pearson, 2006.

[4] Mitchell T., *Machine Learning*, McGraw Hill.

[5] Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, *Machine Learning : An Artificial Intelligence Approach*, Tioga Publishing Company.

[6] Michael J. Kearns and Umesh V. Vazirani, *An Introduction to Computational Learning Theory*, The MIT Press, Cambridge, Massachusetts, 1994.

[7] D. H. Wolpert, W. G. Macready (1997), “No Free Lunch Theorems for Optimization”, IEEE Transactions on Evolutionary Computation 1, 67.

Index