works, architecture design and applica-
s, University of Athens, 1995.
ral network architectures for selecting
f Computer Mathematics, Vol. 68(1–2),

puting with neural nets," *IEEE ASSP*

Mathematics, McGraw-Hill, 1968.
n PCM," *IEEE Transactions on Infor-*

ssification and analysis of multivariate
on Mathematical Statistics and Proba-
669871, University of California Press,

Design, Prentice Hall, 1979.
-learning pattern classification using a
nition, Vol. 18(3/4), pp. 271–277, 1985.
an adaptive process of sample set con-
heory, Vol. 8(5), pp. S82–S91, 1962.
nomy, W.H. Freeman, San Francisco,

ss Horwood, 1980.
sequential clustering method," *Pattern*

CHAPTER 13

CLUSTERING ALGORITHMS II: HIERARCHICAL ALGORITHMS

13.1 INTRODUCTION

Hierarchical clustering algorithms are of a different philosophy from the algorithms described in the previous chapter. Specifically, instead of producing a single clustering they produce a hierarchy of clusterings. This kind of algorithms is usually met in the social sciences and biological taxonomy (e.g., [El-G 68, Prit 71, Shea 65, McQu 62]). In addition, they have been used in many other fields, including modern biology, medicine, and archaeology (e.g., [Stri 67, Bobe 93, Solo 71, Hods 71]). Also, applications of the hierarchical algorithms may be found in computer science and engineering (e.g., [Murt 95, Kank 96]).

Before we describe their basic idea, let us recall that

$$X = \{\boldsymbol{x}_i, \quad i = 1, \ldots, N\}$$

is a set of l-dimensional vectors that are to be clustered. Also, recall the definition of a clustering

$$\mathcal{R} = \{C_j, \quad j = 1, \ldots, m\}$$

where $C_j \subseteq X$, from Chapter 11.

A clustering $\mathcal{R}_1$ containing k clusters is said to be *nested* in the clustering $\mathcal{R}_2$, which contains r ($< k$) clusters, if *each* cluster in $\mathcal{R}_1$ is a subset of a set in $\mathcal{R}_2$ and at least one cluster of $\mathcal{R}_1$ is a proper subset of $\mathcal{R}_2$. In this case we write $\mathcal{R}_1 \sqsubset \mathcal{R}_2$. For example, the clustering $\mathcal{R}_1 = \{\{\boldsymbol{x}_1, \boldsymbol{x}_3\}, \{\boldsymbol{x}_4\}, \{\boldsymbol{x}_2, \boldsymbol{x}_5\}\}$ is nested in $\mathcal{R}_2 = \{\{\boldsymbol{x}_1, \boldsymbol{x}_3, \boldsymbol{x}_4\}, \{\boldsymbol{x}_2, \boldsymbol{x}_5\}\}$. On the other hand, $\mathcal{R}_1$ is nested neither in $\mathcal{R}_3 = \{\{\boldsymbol{x}_1, \boldsymbol{x}_4\}, \{\boldsymbol{x}_3\}, \{\boldsymbol{x}_2, \boldsymbol{x}_5\}\}$ nor in $\mathcal{R}_4 = \{\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_4\}, \{\boldsymbol{x}_3, \boldsymbol{x}_5\}\}$. It is clear that a clustering is not nested to itself.

Hierarchical clustering algorithms produce a *hierarchy of nested clusterings*. More specifically, these algorithms involve N steps, as many as the number of

data vectors. At each step t, a new clustering is obtained based on the clustering produced at the previous step $t-1$. There are two main categories of these algorithms, the *agglomerative* and the *divisive hierarchical algorithms*.

The initial clustering $\mathcal{R}_0$ for the agglomerative algorithms consists of N clusters each containing a single element of X. At the first step, the clustering $\mathcal{R}_1$ is produced. It contains $N-1$ sets, such that $\mathcal{R}_0 \sqsubset \mathcal{R}_1$. This procedure continues until the final clustering, $\mathcal{R}_{N-1}$, is obtained, which contains a single set, that is, the set of data, X. Notice that for the hierarchy of the resulting clusterings we have

$$\mathcal{R}_0 \sqsubset \mathcal{R}_1 \sqsubset \cdots \sqsubset \mathcal{R}_{N-1}$$

The divisive algorithms follow the inverse path. In this case, the initial clustering $\mathcal{R}_0$ consists of a single set, X. At the first step the clustering $\mathcal{R}_1$ is produced. It consists of two sets, such that $\mathcal{R}_1 \sqsubset \mathcal{R}_0$. This procedure continues until the final clustering $\mathcal{R}_{N-1}$ is obtained, which contains N sets, each consisting of a single element of X. In this case we have

$$\mathcal{R}_{N-1} \sqsubset \mathcal{R}_{N-2} \sqsubset \ldots, \sqsubset \mathcal{R}_0$$

The next section is devoted to the agglomerative algorithms. The divisive algorithms are briefly discussed in Section 13.4.

13.2 AGGLOMERATIVE ALGORITHMS

Let $g(C_i, C_j)$ be a function defined for all possible pairs of clusters of X. This function measures the proximity between C_i and C_j. Let t denote the current level of hierarchy. Then, the general agglomerative scheme may be stated as follows:

Generalized Agglomerative Scheme (GAS)

- 1. Initialization:

  —1.1 Choose $\mathcal{R}_0 = \{C_i = \{\boldsymbol{x}_i\},\ i = 1, \ldots, N\}$ as the initial clustering.

  —1.2 $t = 0$.

- 2. Repeat:

  —2.1. $t = t + 1$

  —2.2. Among all possible pairs of clusters (C_r, C_s) in $\mathcal{R}_{t-1}$ find the one, say (C_i, C_j), such that

$$g(C_i, C_j) = \begin{cases} \min_{r,s} g(C_r, C_s), & \text{if } g \text{ is a dissimilarity function} \\ \max_{r,s} g(C_r, C_s), & \text{if } g \text{ is a similarity function} \end{cases} \tag{13.1}$$

—2.3. Define $C_q = C_i \cup C_j$ and produce the new clustering $\Re_t = (\Re_{t-1} - \{C_i, C_j\}) \cup \{C_q\}$.

- Until all vectors lie in a single cluster.

It is clear that this scheme creates a hierarchy of N clusterings, so that each one is nested in all successive clusterings, that is, $\Re_t \sqsubset \Re_s$, for $t < s, s = 1, \ldots, N-1$. Alternatively, we can say that *if two vectors come together into a single cluster at level t of the hierarchy, they will remain in the same cluster for all subsequent clusterings*. This is another way of viewing the nesting property.

A disadvantage of the nesting property is that there is no way to recover from a "poor" clustering that may have occurred in an earlier level of the hierarchy (see [Gowe 67]).[1]

At each level t, there are $N - t$ clusters. Thus, in order to determine the pair of clusters that is going to be merged at the $t + 1$ level, $\binom{N-t}{2} \equiv \frac{(N-t)(N-t-1)}{2}$ pairs of clusters have to be considered. Thus, the total number of pairs that have to be examined throughout the whole clustering process is

$$\sum_{t=0}^{N-1} \binom{N-t}{2} = \sum_{k=1}^{N} \binom{k}{2} = \frac{(N-1)N(N+1)}{6}$$

that is, the total number of operations required by an agglomerative scheme is proportional to N^3. Moreover, the exact complexity of the algorithm depends on the definition of g.

13.2.1 Definition of Some Useful Quantities

There are two main categories of agglomerative algorithms. Algorithms of the first category are based on matrix theory concepts, while algorithms of the second one are based on graph theory concepts. Before we enter into their discussion, some definitions are required. The *pattern matrix* $D(X)$ is the $N \times l$ matrix, whose ith row is the (transposed) ith vector of X. The *similarity (dissimilarity) matrix*, $P(X)$, is an $N \times N$ matrix whose (i, j) element equals the similarity $s(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (dissimilarity $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$) between vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. It is also referred to as the *proximity matrix* to include both cases. P is a symmetric matrix.[2] Moreover, if P is a similarity matrix, its diagonal elements are equal to the maximum value of s. On the other hand, if P is a dissimilarity matrix, its diagonal elements are equal to the minimum value of d. Notice that for a single pattern matrix there

[1] Recently, a method that produces hierarchies which do not, necessarily, possess the nesting property has been proposed ([Frig 97]).

[2] Note, however, that in [Ozaw 83] a hierarchical clustering algorithm, called RANCOR, is discussed, which is based on asymmetric proximity matrices.

$\{\{x_1\},\{x_2\},\{x_3\},\{x_4\},\{x_5\}\}$

$\{\{x_1,x_2\},\{x_3\},\{x_4\},\{x_5\}\}$

$\{\{x_1,x_2\},\{x_3\},\{x_4,x_5\}\}$

$\{\{x_1,x_2\},\{x_3,x_4,x_5\}\}$

$\{\{x_1,x_2,x_3,x_4,x_5\}\}$

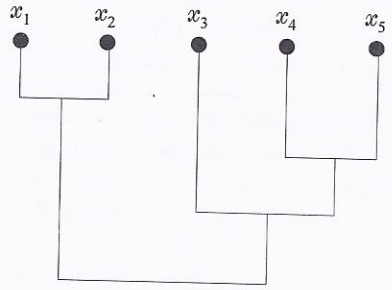


FIGURE 13.1: The clustering hierarchy for X of Example 13.1 and its corresponding dendrogram.

exists more than one proximity matrix depending on the choice of the proximity measure $\wp(\boldsymbol{x}_i, \boldsymbol{x}_j)$. However, fixing $\wp(\boldsymbol{x}_i, \boldsymbol{x}_j)$, one can easily observe that for a given pattern matrix there exists an associated single proximity matrix. On the other hand, a proximity matrix may correspond to more than one pattern matrices (see Problem 13.1).

Example 13.1. Let $X = \{\boldsymbol{x}_i,\ i = 1, \ldots, 5\}$, with $\boldsymbol{x}_1 = [1, 1]^T$, $\boldsymbol{x}_2 = [2, 1]^T$, $\boldsymbol{x}_3 = [5, 4]^T$, $\boldsymbol{x}_4 = [6, 5]^T$, and $\boldsymbol{x}_5 = [6.5, 6]^T$. The pattern matrix of X is

$$D(X) = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 5 & 4 \\ 6 & 5 \\ 6.5 & 6 \end{bmatrix}$$

and its corresponding dissimilarity matrix, when the Euclidean distance is in use, is

$$P(X) = \begin{bmatrix} 0 & 1 & 5 & 6.4 & 7.4 \\ 1 & 0 & 4.2 & 5.7 & 6.7 \\ 5 & 4.2 & 0 & 1.4 & 2.5 \\ 6.4 & 5.7 & 1.4 & 0 & 1.1 \\ 7.4 & 6.7 & 2.5 & 1.1 & 0 \end{bmatrix}$$

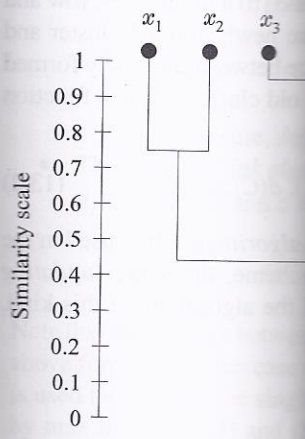
When the Tanimoto measure is used, the similarity matrix of X becomes

$$P'(X) = \begin{bmatrix} 1 & 0.75 & 0.26 & 0.21 & 0.18 \\ 0.75 & 1 & 0.44 & 0.35 & 0.20 \\ 0.26 & 0.44 & 1 & 0.96 & 0.90 \\ 0.21 & 0.35 & 0.96 & 1 & 0.98 \\ 0.18 & 0.20 & 0.90 & 0.98 & 1 \end{bmatrix}$$

Note that in $P(X)$ all diagon
diagonal elements are equal t

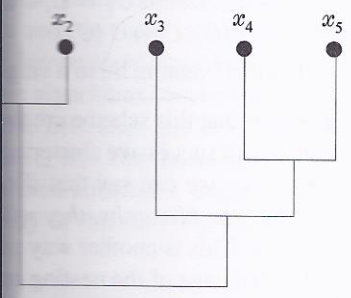
A *threshold dendrogram*
representing the sequence
To clarify this idea, let us
Let us define $g(C_i, C_j)$ as
may easily see that, in this
generalized agglomerative
vectors is used, is the one s
new cluster. At the second s
the third step $\boldsymbol{x}_3$ joins the cl
$\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ and $\{\boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5\}$ ar
Figure 13.1 shows the corre
to a level of the dendrogram
clustering.

A *proximity dendrogram*
proximity where two cluste
(similarity) measure is in u
(similarity) dendrogram. T
forced formation of clusters
clustering that best fits the



(a)

FIGURE 13.2: (a) The p
from Example 13.1. (b) Th
$P(X)$ from Example 13.1.



r X of Example 13.1 and its corre-

ding on the choice of the proximity
), one can easily observe that for a
ed single proximity matrix. On the
nd to more than one pattern matrices

ith $x_1 = [1, 1]^T$, $x_2 = [2, 1]^T$, $x_3 =$
ern matrix of X is

$$\begin{bmatrix} 1 \\ 1 \\ 4 \\ 5 \\ 6 \end{bmatrix}$$

e Euclidean distance is in use, is

$$\begin{bmatrix} & 6.4 & 7.4 \\ 2 & 5.7 & 6.7 \\ & 1.4 & 2.5 \\ 4 & 0 & 1.1 \\ 5 & 1.1 & 0 \end{bmatrix}$$

rity matrix of X becomes

$$\begin{bmatrix} 5 & 0.21 & 0.18 \\ 4 & 0.35 & 0.20 \\ & 0.96 & 0.90 \\ 6 & 1 & 0.98 \\ 0 & 0.98 & 1 \end{bmatrix}$$

Note that in $P(X)$ all diagonal elements are 0, since $d_2(x, x) = 0$, while in $P'(X)$ all diagonal elements are equal to 1, since $s_T(x, x) = 1$.

A *threshold dendrogram*, or simply a *dendrogram*, is an effective means of representing the sequence of clusterings produced by an agglomerative algorithm. To clarify this idea, let us consider again the data set given in Example 13.1. Let us define $g(C_i, C_j)$ as $g(C_i, C_j) = d_{min}^{ss}(C_i, C_j)$ (see Section 11.2). One may easily see that, in this case, the clustering sequence for X produced by the generalized agglomerative scheme, when the Euclidean distance between two vectors is used, is the one shown in Figure 13.1. At the first step x_1 and x_2 form a new cluster. At the second step x_4 and x_5 stick together, forming a single cluster. At the third step x_3 joins the cluster $\{x_4, x_5\}$ and, finally, at the fourth step the clusters $\{x_1, x_2\}$ and $\{x_3, x_4, x_5\}$ are merged into a single set, X. The right-hand side of Figure 13.1 shows the corresponding dendrogram. Each step of GAS corresponds to a level of the dendrogram. *Cutting the dedrogram at a specific level results in a clustering.*

A *proximity dendrogram* is a dendrogram that takes into account the level of proximity where two clusters are merged *for the first time*. When a dissimilarity (similarity) measure is in use, the proximity dendrogram is called a *dissimilarity (similarity) dendrogram*. This tool may be used as an indicator of the natural or forced formation of clusters at any level. That is, it may provide a clue about the clustering that best fits the data, as will be explained in Section 13.5. Figure 13.2

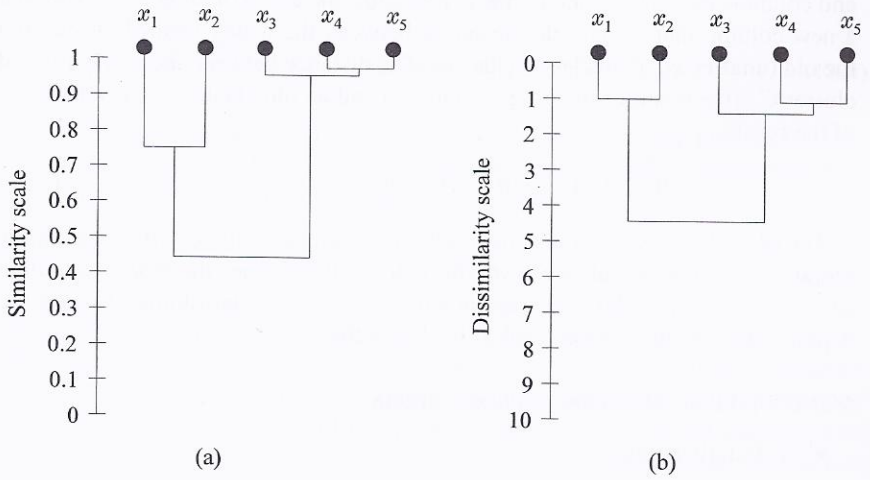


FIGURE 13.2: (a) The proximity (similarity) dendrogram for X using $P'(X)$ from Example 13.1. (b) The proximity (dissimilarity) dendrogram for X using $P(X)$ from Example 13.1.

shows the similarity and dissimilarity dendrograms for X of Example 13.1 when $P'(X)$ and $P(X)$ are in use, respectively.

Before we proceed to a more detailed discussion of the hierarchical algorithms, an important note is in order. As explained earlier, this kind of algorithm determines a whole hierarchy of clusterings, rather than a single clustering. The determination of the whole dendrogram may be very useful in some applications, such as biological taxonomy (e.g., see [Prit 71]). However, in other applications we are interested only in the specific clustering that best fits the data. If one is willing to use hierarchical algorithms for applications of the latter type, he or she has to decide which clustering of the produced hierarchy is most suitable for the data. Equivalently, one must determine the appropriate level to cut the dendrogram that corresponds to the resulting hierarchy. Similar comments also hold for the divisive algorithms to be discussed later. Methods for determining the cutting level are discussed in the last section of the chapter.

In the sequel, unless otherwise stated, we consider only dissimilarity matrices. Similar arguments hold for similarity matrices.

13.2.2 Agglomerative Algorithms Based on Matrix Theory

These algorithms may be viewed as special cases of GAS. The input in these schemes is the $N \times N$ dissimilarity matrix, $P_0 = P(X)$, derived from X. At each level, t, when two clusters are merged into one, the size of the dissimilarity matrix P_t becomes $(N-t) \times (N-t)$. P_t follows from P_{t-1} by (a) deleting the two rows and columns that correspond to the merged clusters and (b) adding a new row and a new column that contain the distances between the newly formed cluster and the old (unaffected at this level) clusters. The distance between the newly formed cluster C_q (the result of merging C_i and C_j) and an old cluster, C_s, is a function of the form

$$d(C_q, C_s) = f(d(C_i, C_s), d(C_j, C_s), d(C_i, C_j)) \tag{13.2}$$

The procedure justifies the name *matrix updating algorithms*, often used in the literature. In the sequel, we give an algorithmic scheme, the *matrix updating algorithmic scheme (MUAS)*, that includes most of the algorithms of this kind. Again, t denotes the current level of the hierarchy.

Matrix Updating Algorithmic Scheme (MUAS).

- 1. Initialization:

  —1.1. $\Re_0 = \{\{\boldsymbol{x}_i\},\ i = 1, \ldots, N\}$.

  —1.2. $P_0 = P(X)$.

  —1.3. $t = 0$

rams for X of Example 13.1 when

sion of the hierarchical algorithms,
r, this kind of algorithm determines
single clustering. The determina-
eful in some applications, such as
vever, in other applications we are
est fits the data. If one is willing
of the latter type, he or she has to
rchy is most suitable for the data.
e level to cut the dendrogram that
comments also hold for the divi-
for determining the cutting level

nsider only dissimilarity matrices.

latrix Theory

ases of GAS. The input in these
$= P(X)$, derived from X. At each
the size of the dissimilarity matrix
P_{t-1} by (a) deleting the two rows
ters and (b) adding a new row and
en the newly formed cluster and
istance between the newly formed
d an old cluster, C_s, is a function

$$\ldots, C_s), d(C_i, C_j)) \qquad (13.2)$$

ting algorithms, often used in the
nic scheme, the *matrix updating*
st of the algorithms of this kind.
by.

- 2. Repeat:
  - —2.1 $t = t + 1$
  - —2.2 Find C_i, C_j such that $d(C_i, C_j) = \min_{r,s=1,\ldots,N,\ r\neq s} d(C_r, C_s)$.
  - —2.3 Merge C_i, C_j into a single cluster C_q and form $\Re_t = (\Re_{t-1} - \{C_i, C_j\}) \cup \{C_q\}$.
  - —2.4 Define the proximity matrix P_t from P_{t-1} as explained in the text.
- Until $\Re_{N-1}$ clustering is formed, that is, all vectors lie in the same cluster.

Notice that this scheme is in the spirit of the GAS. In [Lanc 67] it is pointed out that a number of distance functions comply with the following update equation:

$$d(C_q, C_s) = a_i d(C_i, C_s) + a_j d(C_j, C_s) + b d(C_i, C_j) + c|d(C_i, C_s) - d(C_j, C_s)| \qquad (13.3)$$

Different values of a_i, a_j, b, and c correspond to different choices of the dissimilarity measure $d(C_i, C_j)$. Equation (13.3) is also a recursive definition of a distance between two clusters, initialized from the distance between the initial point clusters. Another formula, not involving the last term and allowing a_i, a_j, and b to be functions of C_i, C_j, and C_s, is discussed in [Bobe 93]. In the sequel we present algorithms stemming from MUAS and following from Eq. (13.3) for different values of the parameters a_i, a_j, b, c.

The simpler algorithms included in this scheme are:

- The *single link algorithm*. This is obtained from Eq. (13.3) if we set $a_i = 1/2$, $a_j = 1/2$, $b = 0$, $c = -1/2$. In this case,

$$d(C_q, C_s) = \min\{d(C_i, C_s), d(C_j, C_s)\} \qquad (13.4)$$

  The d_{min}^{ss} measure, defined in Section 11.2, falls under this umbrella.
- The *complete link algorithm*. This follows from Eq. (13.3) if we set $a_i = \frac{1}{2}$, $a_j = \frac{1}{2}$, $b = 0$ and $c = \frac{1}{2}$. Then we may write[3]

$$d(C_q, C_s) = \max\{d(C_i, C_s), d(C_j, C_s)\}. \qquad (13.5)$$

Note that the distance between the merged clusters C_i and C_j does not enter into the above formulae. In the case where a similarity, instead of a dissimilarity, measure is used then (a) for the single link algorithm the operator min should be replaced by max in Eq. (13.4) and (b) for the complete link algorithm the operator max should be replaced by the operator min in Eq. (13.5). To gain a further insight into the behaviour of the above algorithms, let us consider the following example.

[3] Equations (13.4) and (13.5) suggest that merging clusters is a min/max problem for the complete link and a min/min problem for the single link.

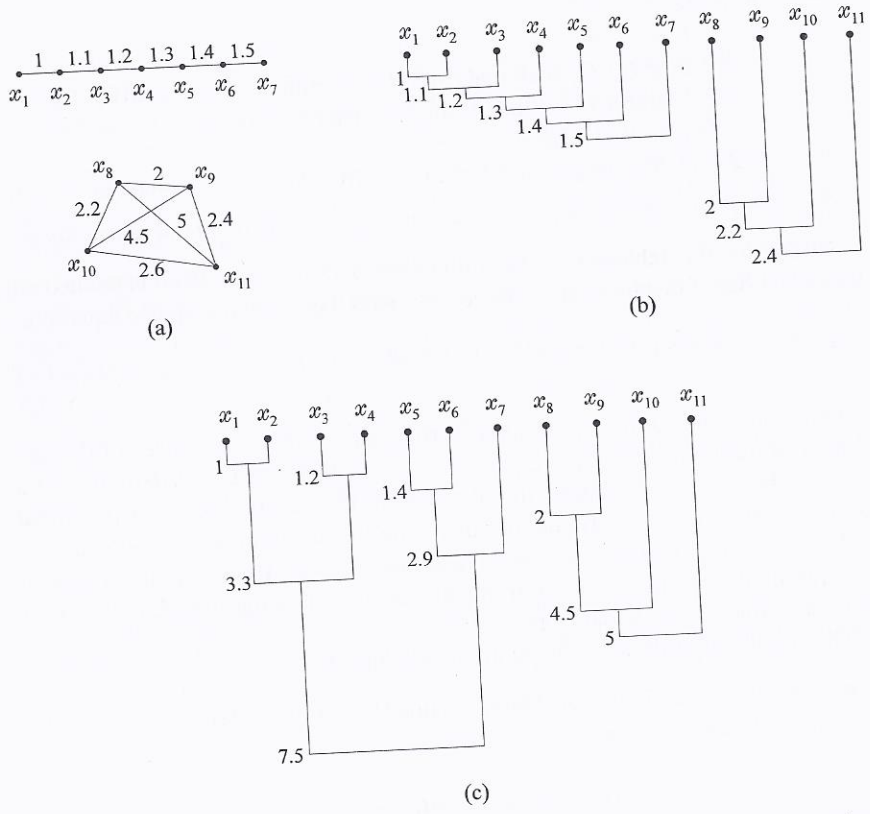


FIGURE 13.3: (a) The data set X. (b) The dissimilarity dendrogram produced by the single link algorithm. (c) The dissimilarity dendrogram produced by the complete link algorithm (the level of the final clustering is not shown).

Example 13.2. Consider the data set shown in Figure 13.3a. The first seven points form an elongated cluster while the remaining four form a rather compact cluster. The numbers on top of the edges connecting the points correspond to the respective (Euclidean) distances between vectors. These distances are also taken to measure the distance between two initial point clusters. Distances that are not shown are assumed to have very large values. Figure 13.3b shows the dendrogram produced by the application of the single link algorithm to this data set. As one can easily observe, the algorithm first recovers the elongated cluster, and the second cluster is recovered at a higher dissimilarity level.

Figure 13.3c shows the dendrogram produced by the complete link algorithm. It is easily noticed that this algorithm proceeds by recovering compact clusters.

Remark

- The preceding algorithms are the two extremes of the family described by Eq. (13.3). Indeed, the clusters produced by the single link algorithm are formed at low dissimilarities in the dissimilarity dendrogram. On the other hand, the clusters produced by the complete link algorithm are formed at high dissimilarities in the dissimilarity dendrogram. This happens because in the single link (complete link) algorithm the minimum (maximum) of the distances $d(C_i, C_s)$ and $d(C_j, C_s)$ is used as the distance between $d(C_q, C_s)$. This implies that the single link algorithm has a tendency to favour elongated clusters. This characteristic is also known as *chaining effect*. On the other hand, the complete link algorithm proceeds by recovering small compact clusters, and it should be preferred if there is evidence that compact clusters underlie X.

The rest of the algorithms, to be discussed next, are compromises between these two extremes.[4]

- The *weighted pair group method average (WPGMA)* algorithm is obtained from Eq. (13.3) if we set $a_i = a_j = \frac{1}{2}$, $b = 0$, and $c = 0$, that is,

$$d(C_q, C_s) = \frac{1}{2}(d(C_i, C_s) + d(C_j, C_s)) \tag{13.6}$$

  Thus, in this case the distance between the newly formed cluster C_q and an old one C_s is defined as the average of distances between C_i and C_s and C_j and C_s.
- The *unweighted pair group method average (UPGMA)* algorithm is defined if we choose $a_i = \frac{n_i}{n_i+n_j}$, $a_j = \frac{n_j}{n_i+n_j}$, $b = 0$, $c = 0$, where n_i and n_j are the cardinalities of C_i and C_j, respectively. In this case the distance between C_q and C_s is defined as

$$d(C_q, C_s) = \frac{n_i}{n_i + n_j} d(C_i, C_s) + \frac{n_j}{n_i + n_j} d(C_j, C_s) \tag{13.7}$$

- The *unweighted pair group method centroid (UPGMC)* algorithm results on setting $a_i = \frac{n_i}{n_i+n_j}$, $a_j = \frac{n_j}{n_i+n_j}$, $b = -\frac{n_i n_j}{(n_i+n_j)^2}$, c = 0, that is,

$$d_{qs} = \frac{n_i}{n_i + n_j} d_{is} + \frac{n_j}{n_i + n_j} d_{js} - \frac{n_i n_j}{(n_i + n_j)^2} d_{ij} \tag{13.8}$$

[4]The terminology used here follows that given in [Jain 88].

This algorithm has an interesting interpretation. Let the representatives of the clusters be chosen as the respective means (centroids), that is,

$$\boldsymbol{m}_q = \frac{1}{n_q} \sum_{\boldsymbol{x} \in C_q} \boldsymbol{x} \tag{13.9}$$

and the dissimilarity to be the squared Euclidean distance between cluster representatives. Then it turns out that this recursive definition of d_{qs} is nothing but the square Euclidean distance between the respective representatives (see Problem 13.2), that is,

$$d_{qs} = ||\boldsymbol{m}_q - \boldsymbol{m}_s||^2 \tag{13.10}$$

- The *weighted pair group method centroid (WPGMC)* algorithm is obtained if we choose $a_i = a_j = \frac{1}{2}$, $b = -\frac{1}{4}$, and $c = 0$. That is,

$$d_{qs} = \frac{1}{2} d_{is} + \frac{1}{2} d_{js} - \frac{1}{4} d_{ij} \tag{13.11}$$

  Note that Eq. (13.11) results from (13.8) if the merging clusters have the same number of vectors. Of course, this is not true in general, and the algorithm basically computes the distance between weighted versions of the respective centroids. A notable feature of the WPGMC algorithm is that $d_{qs} \leq \min(d_{is}, d_{js})$ (Problem 13.3).
- The *Ward or minimum variance algorithm.* Here, the distance between two clusters C_i and C_j, d'_{ij}, is defined as a weighted version of the squared Euclidean distance of their mean vectors, that is,

$$d'_{ij} = \frac{n_i n_j}{n_i + n_j} d_{ij} \tag{13.12}$$

  where $d_{ij} = ||\boldsymbol{m}_i - \boldsymbol{m}_j||^2$. Thus, in step 2.2 of MUAS we seek the pair of clusters C_i, C_j so that the quantity d'_{ij} is minimum. Furthermore, it can be shown (Problem 13.4) that this distance belongs to the family of Eq. (13.3) and we can write

$$d'_{qs} = \frac{n_i + n_s}{n_i + n_j + n_s} d'_{is} + \frac{n_j + n_s}{n_i + n_j + n_s} d'_{js} - \frac{n_s}{n_i + n_j + n_s} d'_{ij} \tag{13.13}$$

The preceding distance can also be viewed from a different perspective. Let us define

$$e_r^2 = \sum_{\boldsymbol{x} \in C_r} ||\boldsymbol{x} - \boldsymbol{m}_r||^2$$

as the variance of the rth cluster around its mean and

$$E_t = \sum_{r=1}^{N-t} e_r^2 \tag{13.14}$$

as the total variance of the clusters at the tth level (where $N - t$ clusters are present). We will now show that *Ward's algorithm forms* $\Re_{t+1}$ *by merging the two clusters that lead to the smallest possible increase of the total variance.* Suppose that clusters C_i and C_j are chosen to be merged into one, say C_q. Let E_{t+1}^{ij} be the total variance after the clusters C_i and C_j are merged in C_q at the $t+1$ level. Then, since all other clusters remain unaffected, the difference $\Delta E_{t+1}^{ij} = E_{t+1}^{ij} - E_t$ is equal to

$$\Delta E_{t+1}^{ij} = e_q^2 - e_i^2 - e_j^2 \tag{13.15}$$

Taking into account that

$$\sum_{\boldsymbol{x} \in C_r} ||\boldsymbol{x} - \boldsymbol{m}_r||^2 = \sum_{\boldsymbol{x} \in C_r} ||\boldsymbol{x}||^2 - n_r ||\boldsymbol{m}_r||^2 \tag{13.16}$$

Eq. (13.15) is written as

$$\Delta E_{t+1}^{ij} = n_i ||\boldsymbol{m}_i||^2 + n_j ||\boldsymbol{m}_j||^2 - n_q ||\boldsymbol{m}_q||^2 \tag{13.17}$$

Using the fact that

$$n_i \boldsymbol{m}_i + n_j \boldsymbol{m}_j = n_q \boldsymbol{m}_q \tag{13.18}$$

Eq. (13.17) becomes

$$\Delta E_{t+1}^{ij} = \frac{n_i n_j}{n_i + n_j} ||\boldsymbol{m}_i - \boldsymbol{m}_j||^2 = d_{ij}' \tag{13.19}$$

which is the distance minimized by Ward's algorithm. This justifies the name minimum variance.

Example 13.3. Consider the following dissimilarity matrix:

$$P_0 = \begin{bmatrix} 0 & 1 & 2 & 26 & 37 \\ 1 & 0 & 3 & 25 & 36 \\ 2 & 3 & 0 & 16 & 25 \\ 26 & 25 & 16 & 0 & 1.5 \\ 37 & 36 & 25 & 1.5 & 0 \end{bmatrix}$$

where the corresponding squared Euclidean distance is adopted. As one can easily observe, the first three vectors, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, and $\boldsymbol{x}_3$, are very close to each other and far away from the others. Likewise, $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$ lie very close to each other and far away from the first

three vectors. For this problem all seven algorithms discussed before result in the same dendrogram. The only difference is that each clustering is formed at a different dissimilarity level.

Let us first consider the single link algorithm. Since P_0 is symmetric, we consider only the upper diagonal elements. The smallest of these elements equals 1 and occurs at position (1, 2) of P_0. Thus, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ come into the same cluster and $\mathfrak{R}_1 = \{\{\boldsymbol{x}_1, \boldsymbol{x}_2\}, \{\boldsymbol{x}_3\}, \{\boldsymbol{x}_4\}, \{\boldsymbol{x}_5\}\}$ is produced. In the sequel, the dissimilarities among the newly formed cluster and the remaining ones have to be computed. This can be achieved via Eq. (13.4). The resulting proximity matrix, P_1, is

$$P_1 = \begin{bmatrix} 0 & 2 & 25 & 36 \\ 2 & 0 & 16 & 25 \\ 25 & 16 & 0 & 1.5 \\ 36 & 25 & 1.5 & 0 \end{bmatrix}$$

Its first row and column correspond to the cluster $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$. The smallest of the upper diagonal elements of P_1 equals 1.5. This means that at the next stage, the clusters $\{\boldsymbol{x}_4\}$ and $\{\boldsymbol{x}_5\}$ will get together into a single cluster, producing $\mathfrak{R}_2 = \{\{\boldsymbol{x}_1, \boldsymbol{x}_2\}, \{\boldsymbol{x}_3\}, \{\boldsymbol{x}_4, \boldsymbol{x}_5\}\}$. Employing Eq. (13.4), we obtain

$$P_2 = \begin{bmatrix} 0 & 2 & 25 \\ 2 & 0 & 16 \\ 25 & 16 & 0 \end{bmatrix}$$

where the first row (column) corresponds to $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$, and the second and third rows (columns) correspond to $\{\boldsymbol{x}_3\}$ and $\{\boldsymbol{x}_4, \boldsymbol{x}_5\}$, respectively. Proceeding as before, at the next stage $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ and $\{\boldsymbol{x}_3\}$ will get together in a single cluster and $\mathfrak{R}_3 = \{\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, \{\boldsymbol{x}_4, \boldsymbol{x}_5\}\}$ is produced. The new proximity matrix, P_3, becomes

$$P_3 = \begin{bmatrix} 0 & 16 \\ 16 & 0 \end{bmatrix},$$

where the first and the second row (column) correspond to $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$ and $\{\boldsymbol{x}_4, \boldsymbol{x}_5\}$ clusters, respectively. Finally, $\mathfrak{R}_4 = \{\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5\}\}$ will be formed at dissimilarity level equal to 16.

Working in a similar fashion, we can apply the remaining six algorithms to P_0. Note that in the case of Ward's algorithm, the initial dissimilarity matrix should be $\frac{1}{2}P_0$, due to the definition in Eq. (13.12). However, care must be taken when we apply WPGMA, WPGMC, and Ward's method. In these cases, when a merging takes place the parameters a_i, a_j, b, and c must be properly adjusted. The proximity levels at which each clustering is formed for each algorithm are shown in Table 13.1.

It is worth noting that the considered task is a nice problem with two well-defined compact clusters lying away from each other. The preceding example demonstrates that in such "easy" cases all algorithms work satisfactorily (as happens with most of the clustering algorithms proposed in the literature). The particular characteristics of each algorithm are revealed when more demanding situations are faced. Thus, in Example 13.2, we saw the

Table 13.1: The results obtained with the seven algorithms discussed when they are applied to the proximity matrix of Example 13.3

	SL	CL	WPGMA	UPGMA	WPGMC	UPGMC	Ward's Algorithm
$\mathfrak{R}_0$	0	0	0	0	0	0	0
$\mathfrak{R}_1$	1	1	1	1	1	1	0.5
$\mathfrak{R}_2$	1.5	1.5	1.5	1.5	1.5	1.5	0.75
$\mathfrak{R}_3$	2	3	2.5	2.5	2.25	2.25	1.5
$\mathfrak{R}_4$	16	37	25.75	27.5	24.69	26.46	29.74

different behaviors of the single link and complete link algorithms. Characteristics of other algorithms, such as the WPGMC and the UPGMC, are discussed next.

13.2.3 Monotonicity and Crossover

Let us consider the following dissimilarity matrix:

$$P = \begin{bmatrix} 0 & 1.8 & 2.4 & 2.3 \\ 1.8 & 0 & 2.5 & 2.7 \\ 2.4 & 2.5 & 0 & 1.2 \\ 2.3 & 2.7 & 1.2 & 0 \end{bmatrix}$$

Application of the single and complete link algorithms to P gives rise to the dissimilarity dendrograms depicted in Figure 13.4a and 13.4b, respectively. Application of the UPGMC and WPGMC algorithms to P results in the same dissimilarity dendrogram, which is shown in Figure 13.4c. In this dendrogram something interesting occurs. The cluster $\{x_1, x_2, x_3, x_4\}$ is formed at a lower dissimilarity level than cluster $\{x_1, x_2\}$. This phenomenon is called *crossover*. More specifically, crossover occurs when a cluster is formed at a lower dissimilarity level than any of its components. The opposite of the crossover is *monotonicity*. Satisfaction of the latter condition implies that each cluster is formed at a higher dissimilarity level than any one of its components. More formally, the monotonicity condition may be stated as follows:

"If clusters C_i and C_j are selected to be merged in cluster C_q, at the tth level of the hierarchy, then the following condition must hold:

$$d(C_q, C_k) \geq d(C_i, C_j)$$

for all C_k, $k \neq i, j, q$."

Monotonicity is a property that is exclusively related to the clustering algorithm and not to the (initial) proximity matrix. Recall Eq. (13.3) defined in terms of the parameters a_i, a_j, b, and c.