8

Miscellaneous clustering methods

8.1 Introduction

The methods described in the preceding chapters form the major part of the body of work on cluster analysis. Nevertheless, there remain a substantial number of other methods that do not fall clearly into any of the previous categories, and in this chapter an attempt is made to describe a number of these techniques. While a comprehensive review is impossible simply because of the vastness of the literature involved, this situation is less daunting than it first appears, since some of the apparently specialized techniques are, in essence, very similar to standard clustering techniques. For example, some of the techniques developed in genetic research entail applying hierarchical clustering (see Chapter 4) but using a specialized distance measure such as Jukes–Cantor or the *optimal matching* coefficient (see Chapter 3). A method patented by the Hewlett Packard Development Company for text clustering uses a recursive hierarchical technique, but considering parts of speech in order (nouns, then verbs, then adjectives, for example) – see Kettenring (2009). Similarly, some of the newer pattern recognition techniques in imaging are closely related to traditional methods. The Kohonen self-organizing map, for example, discussed in Section 8.8.2 as an example of a neural network, is similar in principle to the *k*-means clustering technique described in Chapter 5.

The methods to be discussed in this chapter can be categorized as follows:

- Density search and mode analysis, where clusters are assumed to be concentrated in relatively dense patches in a metric space.
- Methods which allow overlapping clusters, including pyramids.

Cluster Analysis: 5th Edition Brian S. Everitt, Sabine Landau, Morven Leese and Daniel Stahl

- Direct clustering of data matrices, rather than proximity matrices, in order to cluster both variables and objects simultaneously.
- Constrained clustering, where the membership of clusters is determined partly by external information, for example spatial contiguity.
- Fuzzy methods, where objects are not assigned to a particular cluster but possess a membership function indicating the strength of membership to each cluster.
- Neural networks: pattern recognition algorithms that imitate the computational capabilities of large, highly connected networks such as the neurons in the human brain.

This categorization is mainly one of convenience, and some methods have characteristics of more than one of these categories. For example, direct clustering of data matrices may involve reordering rows and columns, so that sets of contiguous rows and columns form clusters of objects and variables, respectively. Having rearranged the matrix in this way it is clearly easy to obtain overlapping clusters. Additionally, overlapping or fuzzy clustering might be viewed as resulting from a relaxation of the usual constraints that some numerical measure of cluster membership, such as the conditional probability that an observation belongs to a particular cluster (see Chapter 6), should sum to 1 over clusters, or should take only the values 0 or 1 for, say, 'in cluster' and 'not in cluster', in a non-overlapping cluster solution.

8.2 Density search clustering techniques

If individuals are depicted as points in a metric space, a natural concept of clustering (see Chapter 1) suggests that there should be parts of the space in which the points are very dense, separated by parts of low density. Several methods of cluster analysis have been developed which search for regions of high density in the data, each such region being taken to signify a different group. The mixture approach described in Chapter 6 might be seen as a formal way of using this concept.

A number of these techniques have their origins in single linkage clustering (see Chapter 4), but attempt to overcome chaining, one of the main problems with this method. One such attempt is the *taxmap* method of Carmichael *et al.* (1968) and Carmichael and Sneath (1969). The clusters are formed initially in a way similar to single linkage, but criteria are used to prevent the addition of objects that are much further away from the last object admitted, such as the drop in average similarity on adding the candidate object. Objects that are rejected in this way initiate new clusters. Two other methods that rely on seeking dense regions will now be described in more detail.

8.2.1 Mode analysis

Mode analysis (Wishart, 1969) is a derivative of single linkage clustering which searches for natural subgroupings of the data by seeking disjoint density surfaces

in the sample distribution. The search is made by considering a sphere of some radius, R, surrounding each point, and counting the number of points falling in the sphere. Individuals are then labelled as *dense* or *non-dense* depending on whether their spheres contain more or fewer points than the value of the *linkage* parameter, K, which is preset at a value dependent on the number of individuals in the data set. (Some possible values of K for various values of n are suggested in Wishart, 1987.)

The parameter R is gradually increased and so more individuals become 'dense'. Four courses of action are possible with the introduction of each new dense point:

- The new point is separated from all other dense points by a distance that exceeds R. When this happens the point initiates a new cluster nucleus and the number of clusters is increased by one.
- The new point is within distance R of one or more dense points which belong to only one cluster nucleus. In this case, the new point is added to the existing cluster.
- The new point is within distance R of dense points belonging to two more clusters. If this happens, the clusters concerned are combined.
- At each 'introduction' cycle, the smallest distance, D, between dense points belonging to different clusters is found, and compared with a threshold value calculated from the average of the $2K$ smallest distance coefficients for each individual. If D is less than this threshold value, then the two clusters are combined. Sometimes only one cluster is produced (indicating a lack of cluster structure in the data), but usually the analysis reaches a point at which a maximum number of clusters is isolated. It is usually this solution which is taken as the most significant.

A difficulty with mode analysis is its failure to identify both large and small clusters simultaneously. A small radius R may distinguish two large, disjoint modes without finding a third smaller (but distinct) mode, because each of its individuals fails to qualify as a dense neighbourhood. Alternatively, if a larger R is specified, the small cluster might be found, but the two large clusters could possibly be merged. Such potential difficulties led Wishart (1973) to suggest an improved mode-seeking cluster method, in which the spherical neighbourhoods of two growing clusters may intersect at some large value of R, to the extent that they would have been fused in the original version of mode analysis; now the fusion level is merely noted, and the clusters are not united.

As an illustration of the use of mode analysis, it will be applied to the distance matrix for 11 forms of the bee *Hoplitis producta*, based on 23 variables (Michener, 1970). The proximity matrix and results are shown in Tables 8.1 and 8.2.

8.2.2 Nearest-neighbour clustering procedures

Wong (1982) and Wong and Lane (1983) describe a hierarchical clustering method which is similar in some respects to the method of mode analysis discussed in Section 8.2.1. The method is designed to detect what Hartigan (1975) defines as

Table 8.1 Matrix of distance coefficients (based on standardized data) for the forms of the bee *Hoplitis producta*[a].

	1	2	3	4	5	6	7	8	9	10	11
1	0										
2	0.940	0									
3	1.229	0.791	0								
4	1.266	0.847	0.303	0							
5	1.507	1.331	1.070	1.026	0						
6	1.609	1.306	0.778	0.573	1.175	0					
7	1.450	1.266	1.475	1.506	1.829	1.876	0				
8	1.239	1.286	1.510	1.540	1.908	1.832	1.665	0			
9	1.493	1.160	0.848	0.792	0.965	0.978	1.847	1.761	0		
10	1.494	1.396	1.497	1.528	1.724	1.687	1.954	1.733	1.721	0	
11	1.348	1.238	1.352	1.385	1.724	1.559	1.844	1.608	1.596	0.645	0

[a]The forms of *Hoplitis* are: 1, *H. gracilis;* 2, *H. subgracilis;* 3, *H. interior;* 4, *H. bernardina;* 5, *H. panamintana;* 6, *H. producta;* 7, *H. colei;* 8, *H. elongata;* 9, *H. uvularis;* 10, *H. grinelli;* 11, *H. septentrionalis.*
Source: Michener (1970).

high-density clusters, these being *maximal connected sets* of the form

$$\{x|f(x) \geq f^*\}, \tag{8.1}$$

where f is the population density of the observations, and f^* is some threshold value. Wong and Lane (1983) estimate the density at a point x by $f_n(x)$ given by

$$f_n(x) = k/[nV_k(x)], \tag{8.2}$$

where $V_k(x)$ is the volume of the smallest sphere centred at x containing k sample observations. A distance matrix arises from these density estimates according to the following two definitions:

Table 8.2 Results of applying mode analysis to the bee distance data in Table 8.1.

Stage	
1	Observation 3 initiates new cluster centre
2	Observation 5 joins observation 3
3	Observation 4 joins [3, 5]
4	Observation 6 joins [3, 5, 4]
5	Observation 7 joins [3, 5, 4, 6]
6	Observation 10 initiates new cluster centre
7	Observation 9 joins observation 10
8	Observation 2 joins [3, 5, 4, 6, 7]
9	Observation 8 joins [3, 5, 4, 6, 7, 2]
10	Observation 1 joins [3, 5, 4, 6, 7, 2, 8]
11	Observation 11 joins [10, 9]
Final results:	
Cluster 1: 3, 5, 4, 6, 7, 2, 8, 1	
Cluster 2: 10, 9, 11	

Definition 1: Two observations x_i and x_j are said to be neighbours if

$$d^*(x_i, x_j) \leq d_k(x_i) \text{ or } d_k(x_j), \tag{8.3}$$

where d^* is the Euclidean metric and $d_k(x_i)$ is the kth nearest-neighbour distance to point x_j.

Definition 2: The distance $d(x_i, x_j)$ between the observations x_i and x_j is

$$d(x_i, x_j) = \frac{1}{2}\left[\frac{1}{f_n(x_i)} + \frac{1}{f_n(x_j)}\right] \tag{8.4}$$

$$= \begin{cases} \dfrac{n}{2k}\left[V_k(x_i) + V_k(x_j)\right] & \text{if } x_i \text{ and } x_j \text{ are neighbours} \\ \infty & \text{otherwise.} \end{cases} \tag{8.5}$$

The single linkage clustering algorithm is then applied to this distance matrix to obtain the dendrogram of sample high-density clusters. The value of k controls the amount by which the data are 'smoothed' to give the density estimate on which the clustering procedure is based. There appears to be no unique recommendation concerning the choice of k, although Wong and Schaack (1982) derive empirical evidence for a rule of thumb of the form $k = 2\log_2 n$. Since the hierarchical clusterings obtained for different values of k can be very different, Wong and Lane (1983) suggest that several values around this value should be tried. To illustrate the operation of their proposed method, Wong and Lane (1983) apply it to the data shown in Figure 8.1. The dendrogram giving the hierarchical clustering obtained by the kth nearest-neighbour method with $k = 5$ appears in Figure 8.2. Two disjoint modal regions, corresponding to the crescentic clusters in Figure 8.1, can be identified in the dendrogram. Ling (1972) suggests another nearest-neighbour

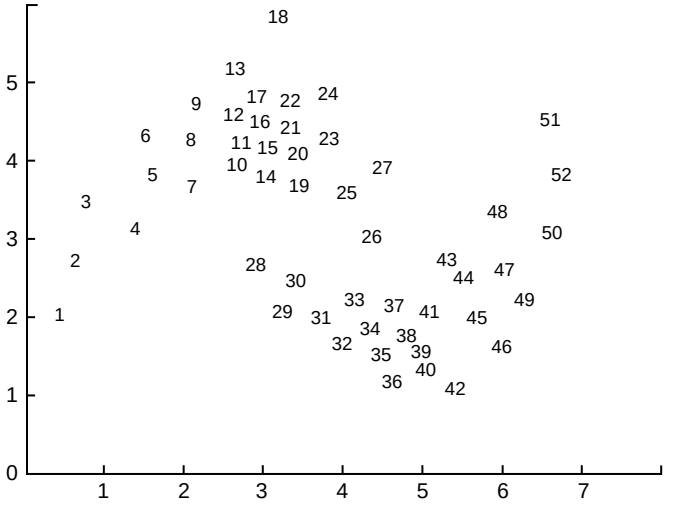

Figure 8.1 *Bivariate data containing crescentic clusters.* (Source: Wong and Lane, 1983.)

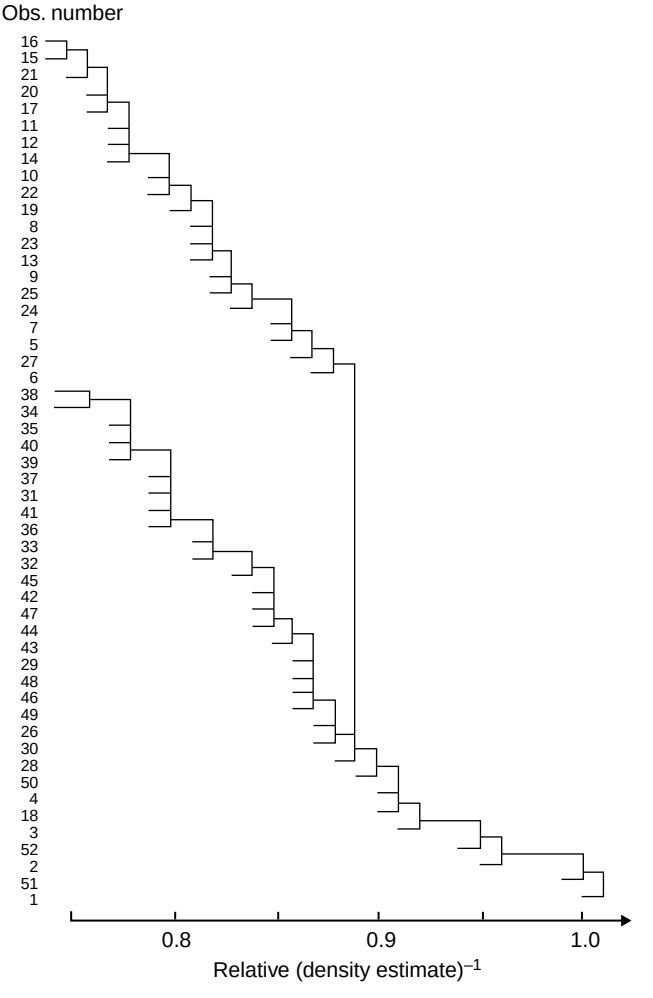

Figure 8.2 *Dendrogram obtained by applying Wong and Lane clustering to the data shown in Figure 8.1. (*Source: Wong and Lane, 1983.)

type clustering procedure, and Wong and Schaack (1982) propose a procedure for indicating the number of clusters when using the method outlined above. Other density-search-type clustering methods are described in Gitman and Levine (1970), Cattell and Coulter (1966) and Katz and Rohlf (1973).

8.3 Density-based spatial clustering of applications with noise

An approach that has had some success and has spawned a number of variants to cope with different scenarios is the DBSCAN – density-based spatial clustering of applications with noise (Sander *et al.*, 1998). The algorithm classifies objects as

clusters (dense regions) or noise (objects in low-density regions). Clusters are sets of at least M objects in a dense region with a radius R; M and R are defined by the user. Within the clusters, two types of object are defined: 'core' and 'non-core' objects. A core object has at least M objects contained within its neighbourhood with radius R, and forms an initial cluster. The neighbourhood (within a radius R) is examined and objects within the neighbourhood are assigned to the cluster. When the neighbourhoods of core objects overlap the clusters are merged. This algorithm defines some objects as 'border' objects; these are objects that are 'density reachable' from a core object. (Object p_1 is 'direct density reachable' from p_2 if it is within a radius R, and it is 'density reachable' from core object p_c if a chain of direct density reachable objects $p_1 \cdots p_n$ can be found). Border objects are not core (they do not have the requisite minimum number of points in their neighbourhood), but they are still sufficiently close (density reachable) to a core point to be included in its cluster. Objects that do not join any cluster at the end of the process are termed 'noise'. Euclidean distance is used as the similarity measure, but any suitable measure could be used in a similar way.

The basic algorithm of DBSCAN is easy to program and it copes with clusters of different shapes, although not so well with clusters of varying density. It is useful not only for clustering per se but where the detection of outliers is of interest. A disadvantage is that it may not work well for high-dimensional space (which is usually sparse). A variation of density searching which is more suitable in this situation is CLIQUE (Agrawal *et al*., 1998), where multivariate space is divided up into a grid of cells, which are classified according to density. Once the dimensions of high density are identified, clustering can proceed in subspaces of lower dimension.

Birant and Kut (2007) have adapted the DBSCAN algorithm to deal with spatially and temporally related data (ST-DBSCAN), and have illustrated it with an example from sea temperature and wave height in the Black Sea. Building-in constraints on clusters, for example based on geographical criteria, is described in Section 8.6. Here the spatial information is used as an additional clustering criterion which pre-specifies the acceptable radius for clusters, alongside the main clustering variables, which were sea characteristics such as temperature, wave height and surface height. Figure 8.3 shows the clusters found for sea surface

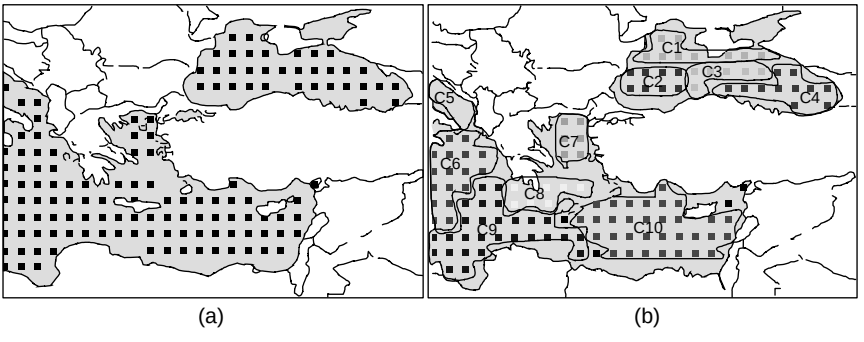

Figure 8.3 *Map of the Black Sea showing (a) the locations of 134 stations; (b) the results of ST-DBSCAN cluster analysis on sea surface height residual data (from Birant and Kut, 2007).*

height (which varies dues to currents, gravity and seawater temperature). The measurements were collected by satellite over five-day periods in years between 1992 and 2002 on a two-dimensional grid separated by one degree in latitude and longitude and stored in a 'data warehouse'.

8.4 Techniques which allow overlapping clusters

In the methods discussed so far, objects are assumed to belong to one particular cluster. However, in many applications it is plausible that objects should belong to more than one cluster simultaneously. An example might be market research where an individual person belongs to several consumer groups, or social network research where people belong to overlapping groups of friends or collaborators. In the latter type of application, it is often the individuals in the intersections of the clusters who are of most interest. An example of a type of study in which overlapping clusters would definitely be inappropriate would be archaeological provenancing, since an object can have only one specific origin. (It might well be that in such an application a single cluster could not be definitely assigned to each object, but this would suggest the use of fuzzy rather than overlapping clustering; see Section 8.7.)

Methods that directly reorder rows and columns of data matrices can be used to produce overlapping clusters, simply by extending the set of rows (columns) to cover a portion of adjoining clusters. These are described under the heading of direct data clustering (Section 8.5); here we discuss methods that have been more specifically designed to deal with overlap. In Section 8.4.1 two relatively early techniques, *clumping* and the B_k *technique*, are briefly discussed. In Section 8.4.2 a more recent and more widely used method, *additive clustering*, is described. Finally, a generalization of dendrograms that allow overlap, the *pyramid*, is discussed in Section 8.4.4.

8.4.1 Clumping and related techniques

Clumping techniques begin with the calculation of a similarity matrix, followed by the division of the data into two groups by a 'cohesion' function including a parameter controlling the degree of overlap. Needham (1967) considered a symmetric cohesion function $G_1(A)$ given by

$$G_1(A) = S_{AB}/S_{AA}S_{BB}. \tag{8.6}$$

Parker-Rhodes and Jackson (1969) suggest a modification, $G_2(A)$, given by

$$G_2(A) = \frac{S_{AB}}{S_{AA}}\left[\frac{n_A(n_A-1)}{S_{AA}} - \frac{S_{AA}}{bn_A(n_A-1)}\right], \tag{8.7}$$

where A and B refer to the two groups into which the data are divided, A being their putative clump. S_{AB} is the sum of the similarities between members of groups

A and B; that is

$$S_{AB} = \sum_{i \in A} \sum_{j \in B} s_{ij}, \tag{8.8}$$

where s_{ij} is an inter-individual similarity, n_A is the number of individuals in group A, and b is an arbitrary parameter which allows the investigator some control over the size of the clumps and the amount of overlap. Algorithms to minimize these functions proceed by successive reallocations of single individuals from an initial randomly chosen cluster centre (Jones and Jackson, 1967). By iterating from different starting points, many divisions into two groups may be found. In each case the members of the smaller group are noted and constitute a class to be set aside for further examination. The cohesion function $G_1(A)$ is designed to find good partitions of the set of individuals, whilst $G_2(A)$ allows the internal similarities of A and the separation of A from B to be adjusted relative to each other by the use of parameter b.

The B_k *technique* (Jardine and Sibson, 1968) is a hierarchical method in which individuals are represented by nodes on a graph, and pairs of nodes are connected which correspond to individuals having a similarity value above some specified threshold H. At each stage in the hierarchy, the set of *maximal complete subgraphs*, or *cliques*, is found. These are the largest sets of individuals for which all pairs of nodes are connected at some level of similarity. If all possible thresholds are considered, an unmanageable number of subgraphs are produced. Moon and Moser (1965) give formulae for the upper bounds to this number. For example, if the proximity matrix consisted of zeros and ones, the upper bound to the number of cliques would be 59 049 for $n = 30$. In the B_k technique, clusters are subsets of the cliques, and the number of clusters is restricted by choosing a value or range of values for k, such that a maximum of $k - 1$ objects belong to the overlap between clusters; any having more than $k - 1$ objects in common are amalgamated. An example is shown in Figure 8.4, using the distance matrix in Table 8.3.

Algorithms for applying this technique have been proposed by Jardine and Sibson (1968) and Rohlf (1975). Although the B_k method has been shown to have various favourable properties, such as stability and invariance under relabeling or monotonic transformation of the proximity matrix (Sibson, 1970), it has had little application other than in a study of acoustic confusion matrices (Morgan, 1973).

8.4.2 Additive clustering

The ADCLUS method (Shepard and Arabie, 1979) has found wider acceptance as a method for identifying overlapping clusters. In this approach, the similarity between two objects is a weighted sum of their common features. A model is fitted to an observed proximity matrix, **S**, such that the model proximity between any pair of objects is the sum of the weights (see below) of those clusters containing

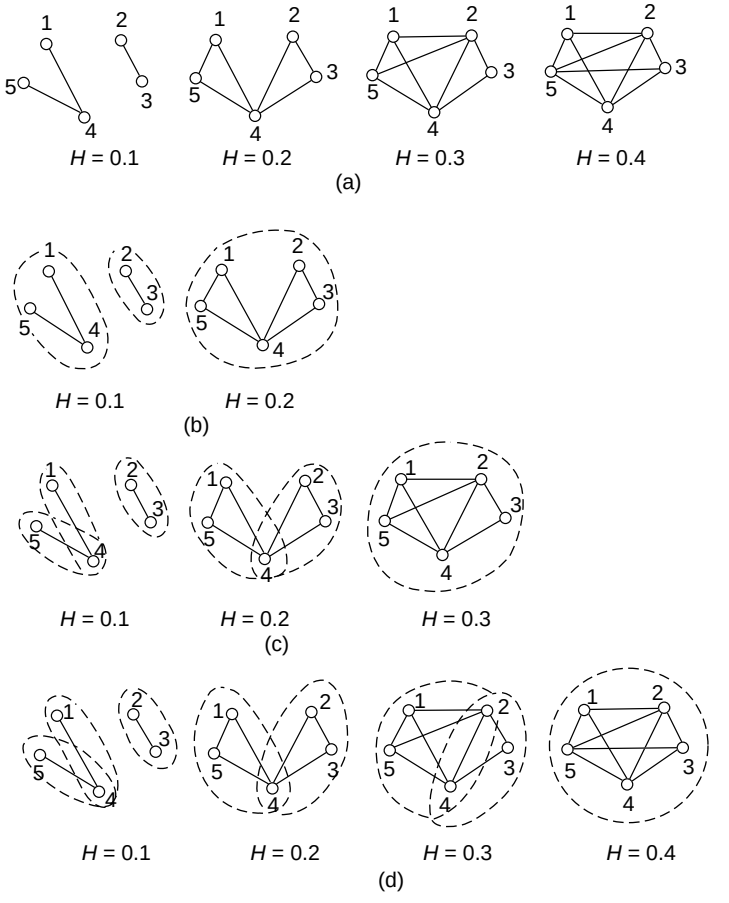

Figure 8.4 *Results of applying Jardine and Sibson's clustering method to the distance matrix in Table 8.3: (a) shows the maximal complete subgraphs for various thresholds H; (b)–(d) show any subsets found within these subgraphs that overlap by 0, 1 and 2 objects.*

Table 8.3 Hypothetical distance matrix to illustrate the B_k method.

Object	1	2	3	4
2	0.1			
3	0.4	0.1		
4	0.1	0.2	0.2	
5	0.2	0.3	0.4	0.1

that pair. The first step is to use the distinct entries s_{ij} in **S** as the possible thresholds for the maximal complete subgraphs; this defines all the potential clusters. These are then coded as columns in a binary $n \times m$ matrix, **P**, of cluster memberships, where n is the sample size and m is the number of potential clusters. The algorithm then fits a set of weights **W** to the clusters, and at the same time reduces the number of clusters. The model fitted is of the following form:

$$\hat{\mathbf{S}} = \mathbf{PWP}' + \mathrm{C}, \tag{8.9}$$

where **C** is a matrix with zeros in the diagonal and a constant in other entries, the constant being equal to the weight fitted to the complete sample; it is used to assess goodness of fit. A balance has to be struck between the goodness of fit and the number of clusters. Once the clusters (as defined by the cluster memberships **P** and the weights **W**) have been fitted to the data, it is hoped that each cluster can be labelled in terms of a discrete, latent common feature linking its constituent objects. The cluster weight indicates the proportion of the similarity between its objects that is attributable to this feature.

Even after fitting the model, an excessive number of cluster solutions may still be encountered, and a modified version in which the number of clusters is chosen in advance by the investigator has been developed, called MAPCLUS (Arabie and Carroll, 1980). An application of this is illustrated below. The technique has also been generalized (as 'INDCLUS') to a multi-observer situation, in which each observer has their own set of weights (Carroll and Arabie, 1983), and to two-mode data by De Sarbo (1982) and Both and Gaul (1986). Further developments have been described by Navarro and Griffiths (2008), who employ methods from nonparametric Bayesian statistics to obtain estimates of the number and importance of features in defining additive models.

8.4.3 Application of MAPCLUS to data on social relations in a monastery

The social structure of an American monastery during the mid 1960s, a period when there was considerable internal strife leading to the departure of many of the monks, has been studied by Sampson (1968). This data set has been analysed a number of times as an interesting example of data relating to a social group that was about to disintegrate. The social relations between 18 novice monks were assessed in 'sociograms', in which each monk rated (in order) his 4 colleagues who respectively stood out in terms of four positive qualities (like, esteem, influence and praise) and four negative qualities (antagonism, disesteem, negative influence, and blame). The novices had been separately classified, on the basis of Sampson's prose description and other methods of cluster analysis, as Young Turks (the newer arrivals), Loyal Opposition, and Outcasts, and also as Leaders or Followers.

The raw data are given by Fienberg *et al.* (1985), and a correlation matrix derived from them is given by Breiger *et al.* (1975). Both publications present alternative ways of analysing the data, which are shown in Table 8.4.

Table 8.5 shows the cluster solution provided by MAPCLUS, which accounts for 62% of the variance, with the weights of the clusters (the sums of the weights of the objects they contain) indicated. These weights are interpreted as the *psychological saliences* of the clusters. Figure 8.5 shows the overlapping clusters superimposed on a multidimensional scaling solution.

The authors observe that Outcasts 3, 17 and 18 were asked to leave the monastery, as was the leader of the Young Turks (2). All but one of the remaining Young Turks (12), the remaining Outcast (13) and only two of the Loyal Opposition (8 and 10) left voluntarily. These last two belong to the bipartisan clusters 7 and 8 that are the least heavily weighted and also have overlap between the two opposing factions. The sixth cluster links three Outcasts with two of the leading Young Turks and a Follower, indicating that the Outcasts' sympathies were with the Young Turks. On the basis of these and other observations, the authors conclude that the overlapping clustering was more realistic and informative in portraying the true situation than a partition-based model.

8.4.4 Pyramids

The pyramid, like the additive tree (see Figure 4.7), is another generalization of the dendrogram which can be used as a basis for clustering: a cut through the pyramid at any given height gives rise to a set of ordered, overlapping clusters. First developed by Diday (1986), the pyramid is less restrictive than the dendrogram, but more restrictive than general methods for overlapping clusters in that objects can belong to (at most) two clusters. It is constructed by means of an algorithm that is similar to those used in agglomerative hierarchical clustering (see Chapter 4), employing an aggregation index based on the dissimilarity matrix **D** to decide on which classes of objects to merge. However, in contrast to hierarchical clustering, it maintains certain order relationships that allow the borders of classes to retain objects in common. Figure 8.6 shows a typical pyramid, and illustrates how *classes* are linked to clusters by connected components of the graph, with objects at the borders of classes linking clusters.

Bertrand (1995) describes the mathematical properties of the pyramidal clustering model, showing that, if the entries in the corresponding cophenetic matrix (see Chapter 4) are the dissimilarities, they never decrease when moving away from the diagonal, so long as the order of the objects is *compatible* with the pyramid. This compatibility property is also known as *Robinsonian* after the developer of a method for seriating pottery assemblages in archaeology (Brainerd, 1951). The recognition of Robinsonian dissimilarities is discussed by Chepoi and Fichet (1997). Compatibility implies that

- partitions produced by cutting the pyramid at a given height are included within these orders;
- the pyramid could be represented with the objects in these orders along the bottom.

Table 8.4 Correlation matrix showing levels of effect, aggregated over four positive and four negative relations, among 18 monks.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1.0																	
0.23	1.0																
0.02	−0.07	1.0															
−0.33	−0.34	−0.06	1.0														
−0.29	−0.48	−0.10	0.15	1.0													
−0.08	−0.17	−0.23	0.41	−0.07	1.0												
0.12	0.24	−0.14	−0.42	−0.01	0.13	1.0											
−0.04	−0.28	−0.37	0.25	0.24	0.35	−0.09	1.0										
−0.19	−0.21	−0.15	0.44	0.26	0.15	−0.40	0.02	1.0									
−0.15	−0.34	−0.19	0.05	0.00	0.18	−0.02	0.21	0.00	1.0								
−0.35	−0.48	0.06	0.45	0.18	0.18	−0.17	−0.01	0.10	0.43	1.0							
0.13	0.19	−0.26	−0.25	−0.19	0.04	0.00	0.04	−0.17	−0.17	−0.25	1.0						
−0.06	−0.33	0.15	0.02	0.09	−0.23	−0.09	−0.05	0.04	0.00	0.04	−0.24	1.0					
0.10	0.31	−0.17	−0.17	−0.06	−0.13	−0.03	0.02	−0.04	−0.33	−0.39	0.19	−0.21	1.0				
0.26	0.38	−0.16	−0.41	−0.17	0.02	0.23	−0.12	−0.14	0.00	−0.33	0.17	−0.26	−0.01	1.0			
−0.12	0.31	−0.18	−0.24	−0.09	−0.28	−0.02	−0.16	−0.26	0.08	−0.18	0.17	0.10	−0.03	0.20	1.0		
0.11	−0.14	0.31	−0.43	−0.04	−0.24	0.12	−0.26	−0.17	−0.15	−0.22	0.05	0.19	0.11	−0.18	−0.10	1.0	
0.07	−0.15	0.25	−0.37	−0.05	−0.56	0.06	−0.27	−0.07	0.12	−0.09	−0.11	0.20	0.08	−0.06	−0.01	0.56	1.0

Source: Sampson (1968).

Table 8.5 Overlapping clusters of monks identified by additive clustering (based on correlation matrix in Table 8.4).

Cluster	Weight	Monks in the subset[a]
1	0.298	4 LOL, 6 LOL, 8 LOL, 5 LOF, 9 LOF, 10 LOF, 11 LOL
2	0.272	3 O, 17 O, 18 O, 13 LOF; O[b]
3	0.271	4 LOL, 11 LOL
4	0.261	4 LOL, 9 LOL
5	0.256	1 YTL, 2 YTL, 7 YTL, 12 YTL, 14 YTF, 15 YTF, 16 YTF
6	0.146	1 YTL, 2 YTL, 14 YTF, 3 O, 17 O, 18 O
7	0.134	5 LOF, 10 LOF, 13 LOF; O, 17 O, 18 O, 16 YTF
8	0.114	2 YTL, 12 YTL, 15 YTF, 6 LOL, 8 LOL

[a]LO, member of Loyal Opposition; YT, member of Young Turks; O, Outcast.
Suffix of L denotes Leader; F denotes Follower.
[b]Novice 13 was variously regarded as a Follower in the Loyal Opposition and as an Outcast.
With these eight clusters and an additive constant of 0.338, the variance accounted for was 62.4%.
Source:Arabie and Carroll (1989).

In Figure 8.6, compatible orders are $\{a, b, c, d, e\}$, as in the figure; $\{a, b, c, \boldsymbol{e}, \boldsymbol{d}\}$; $\{\boldsymbol{c}, \boldsymbol{b}, \boldsymbol{a}, d, e\}$; and $\{\boldsymbol{e}, \boldsymbol{d}, \boldsymbol{c}, \boldsymbol{b}, \boldsymbol{a}\}$ – bold indicating inversion of a cluster. The compatible orders are potentially important in subject matter interpretation (for example, they might indicate an archaeological or geological seriation). They also indicate clearly the objects responsible for overlaps, which themselves may be of particular interest.

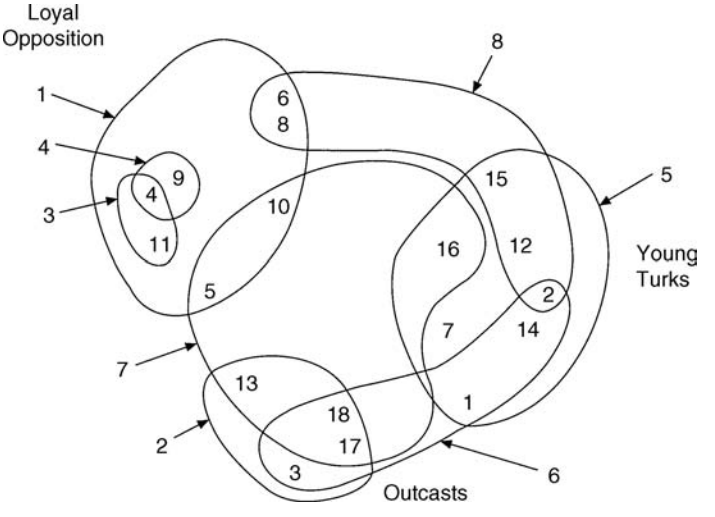

Figure 8.5 *Overlapping clusters of monks found by additive clustering, using* MAPCLUS, *superimposed on a multidimensional nonmetric scaling solution. Numbers with arrows indicate the rank of the clusters according to their weights; see also Table 8.5.* (Source: Arabie and Carroll, 1989.)

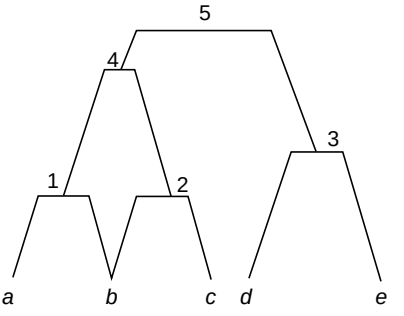

Figure 8.6 *Pyramidal representation with five classes. The heights of the nodes indicate the proximity of all the elements in a given class; b is an element which is on the border of {a, b} and {b, c} and forms an overlap; class 2 is a successor of class 4.* (Taken with permission of the publisher, Elsevier, from Aude et al., 1999.)

Before briefly outlining the algorithm for constructing a pyramid, two further relevant definitions are now given, beginning with the concepts of *order* (*before* and *after*) and *inclusion*.

Subset a is said to be *before* subset b if

$$(\min(a) < \min(b) \text{ and } \max(a) < \max(b)) \text{ or } a = b,$$

and b is said to be *included* in a if and only if

$$\min(a) < \min(b) \leq \max(b) < \max(a),$$

where for each class, min and max refer to the leftmost and rightmost singletons in the class, and $<$ ($>$) means 'to the left (right) of.

The *aggregation index* μ for a new class p, aggregated from classes a and b containing n_a and n_b objects respectively, with any other class q is

$$\mu(p, q) = [n_a \mu(a, q) + n_b \mu(b, q)/(n_a + n_b)]. \tag{8.10}$$

For singletons x and y,

$$\mu(x, y) = d(x, y). \tag{8.11}$$

The pyramid algorithm is initiated with an arbitrary order for the objects, and proceeds by aggregating classes p^* and q^* with the lowest aggregation index, subject to the following conditions:

- p^* is before q^*;
- either p^* or q^* is not included in a previously constructed class;
- if p^* and q^* belong to the same cluster they can be aggregated only if their intersection is not void;

- if p^* and q^* do not belong to the same cluster they can be aggregated only if p^* contains a border of $\min[C(p^*)]$; that is, if $\min[C(p^*)] = \min(p^*)$ or $\max[C(p^*)] = \max(p^*)$, and q^* contains a border of $C(q^*)$.

If, before aggregation, p^* and q^* do not belong to the same connected component, the objects in $C(q^*)$ are positioned before those of $C(p^*)$; this produces a compatible order. The process ends when all objects have been aggregated. Fitting pyramid structures with incomplete data is discussed by Gaul and Schader (1994).

8.4.5 Application of pyramid clustering to gene sequences of yeasts

Aude *et al.* (1999) give further details and a worked example of the algorithm outlined above, and a number of practical applications. One of these is concerned with 11 sequences from the yeast genome, which were analysed using average linkage (Figure 8.7(b)) and as a pyramid (Figure 8.7a)). The authors state that

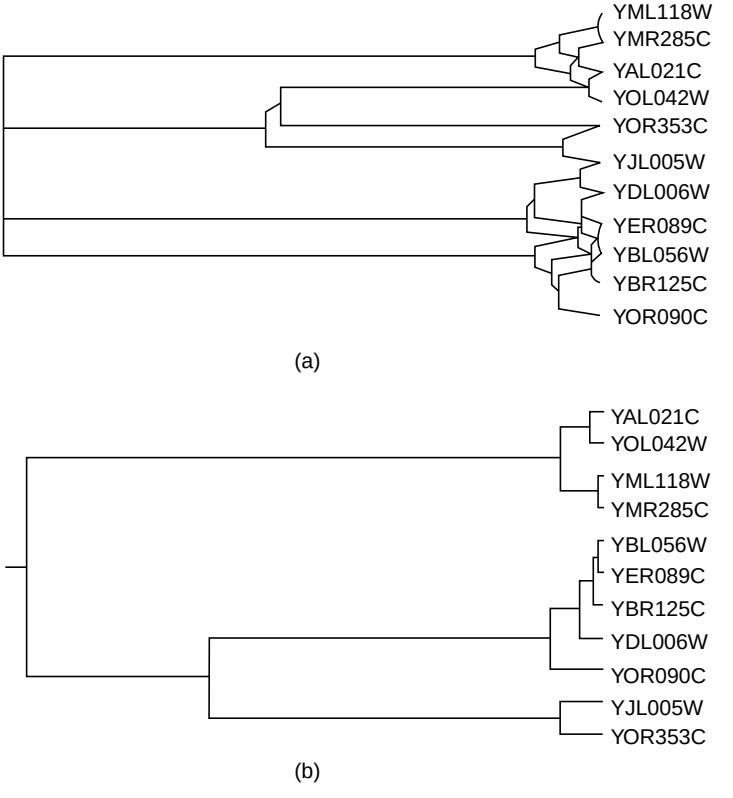

Figure 8.7 *(a) Pyramid and (b) average linkage representations of 11 sequences from the yeast genome.* (Taken with permission of the publisher, Elsevier, from Aude et al., 1999.)

'the complex relationship between these sequences ... are correctly mirrored by the pyramidal classification'. In particular, YJL005W and YAL021C are multi-domain proteins, a fact that is not reflected in the hierarchical classification but is reflected in the pyramid. The authors point out that the orders are not necessarily unique and this might pose problems for interpretation (see Figure 8.5, where four orders were compatible). However, they also point out that in practice it is unusual to find many different compatible orders.

8.5 Simultaneous clustering of objects and variables

In many applications, especially in the social and health sciences, the interpretation of cluster membership (objects) and of cluster characteristics (variables) is equally important. Clustering both simultaneously is known as *biclustering*. Methods that operate on a two-mode data matrix without recourse to a proximity matrix are known as *two-way, two-mode* or *direct clustering* methods. Such methods potentially provide more information than the constituent separate analyses, since they allow the interpretation of the (possibly overlapping) clusters of both objects and variables simultaneously, and also the associative relations between them.

One class of techniques reorders the rows (objects) and columns (variables) of the data matrix, and is sometimes known as *two-way joining*. Early clustering methods of this type include the *bond energy* approach, proposed by McCormick *et al.* (1972) and also discussed by Arabie and Hubert (1990). Here the bond energy of two matrix elements is defined as their product, and rows and columns are permuted, sequentially placing rows (columns) together according to their contribution to the total bond energy. The form of the data has to be carefully considered for valid operation of these reordering methods, and it may be necessary to scale data from different variables so that a comparable response is induced on every object–variable combination. Hartigan (1975) discusses this issue in relation to a number of direct data clustering algorithms. An example of a data set which is naturally in the correct form would be a subject $\times$ stimuli matrix, in which each entry measures the subject's liking for each stimulus; this can be regarded as a rectangular proximity matrix.

Another approach to direct clustering is to fit a tree structure (either ultrametric or additive) to data, rather than to proximity matrices derived from the data. De Soete *et al.* (1984a) have proposed a least-squares procedure for fitting tree structures to data matrices. The algorithm fits a matrix to the observed data such that it obeys a generalization of the ultrametric inequality to two-mode data. Standard clustering methods can then be used to obtain a dendrogram of both objects and variables. Espejo and Gaul (1986) have developed a two-mode variant of average linkage clustering, and De Sarbo (1982) has adapted the ADCLUS model (see Section 8.4.2) for two-mode data.

A recent increase in biclustering methods has resulted from the field of bioinformatics, especially gene expression data, where it may be required to cluster

both genes and samples simultaneously. A review of two-mode clustering methods has been published by Van Mechelen *et al.* (2004), who give references to applications, and information on software from the field of bioinformatics. Prelić *et al.* (2006) have reviewed a number of biclustering methods for such data, and compared them on real and simulated data sets. They concluded that biclustering was generally preferable to conventional hierarchical methods. A number of the more recent algorithms have been included in an `R` package, `Biclust` (Kaiser and Leisch, 2008).

We now describe in more detail two methods for direct clustering of data matrices. The first is the *hierarchical classes* method, which is appropriate for binary data and is an example of a matrix reordering technique. The second is the *error variance* technique, which is less restrictive than the hierarchical classes method in that the data need not be binary, and is an example of a standard hierarchical method applied to a two-mode matrix. Data must be scored or normalized so that larger entries indicate stronger relationships between the corresponding row and column elements.

8.5.1 Hierarchical classes

The *hierarchical classes* method of De Boeck and Rosenberg (1988) is appropriate for binary attribute data. Two hierarchical class structures are defined, one for objects and one for variables, by reordering the matrix. The objects are first grouped into classes with identical attributes, and the classes are then ordered to reflect subset/superset relations. This is repeated for the attributes, in terms of the objects. The object classes are sets of objects having identical attributes, and attribute classes are similarly defined; classes with no objects (attributes) are termed 'undefined'. The hierarchy is defined on the basis of subset/superset relations and may overlap.

Table 8.6 shows a (hypothetical) simple data set concerned with an individual's perception of self and others (Rosenberg *et al.*, 1996). In this, a particular person describes eight 'targets' (objects) using eight traits (attributes). The corresponding hierarchical classes are shown in Figure 8.8, the two hierarchical structures being

Table 8.6 Hypothetical matrix of a person's perception of targets (objects) by traits (attributes).

	Successful	Articulate	Generous	Outgoing	Hardworking	Loving	Warm	Shy
Father	1	1	0	0	1	0	0	0
Boyfriend	0	0	1	0	1	1	1	0
Uncle	0	0	0	1	0	1	1	0
Brother	0	0	0	1	0	1	1	0
Mother	0	0	1	1	1	1	1	0
Me Now	0	0	1	1	1	1	1	0
Ideal Me	1	1	1	1	1	1	1	0
Casual Friend	0	0	0	0	0	0	0	0

Source: Rosenberg *et al.* (1996).

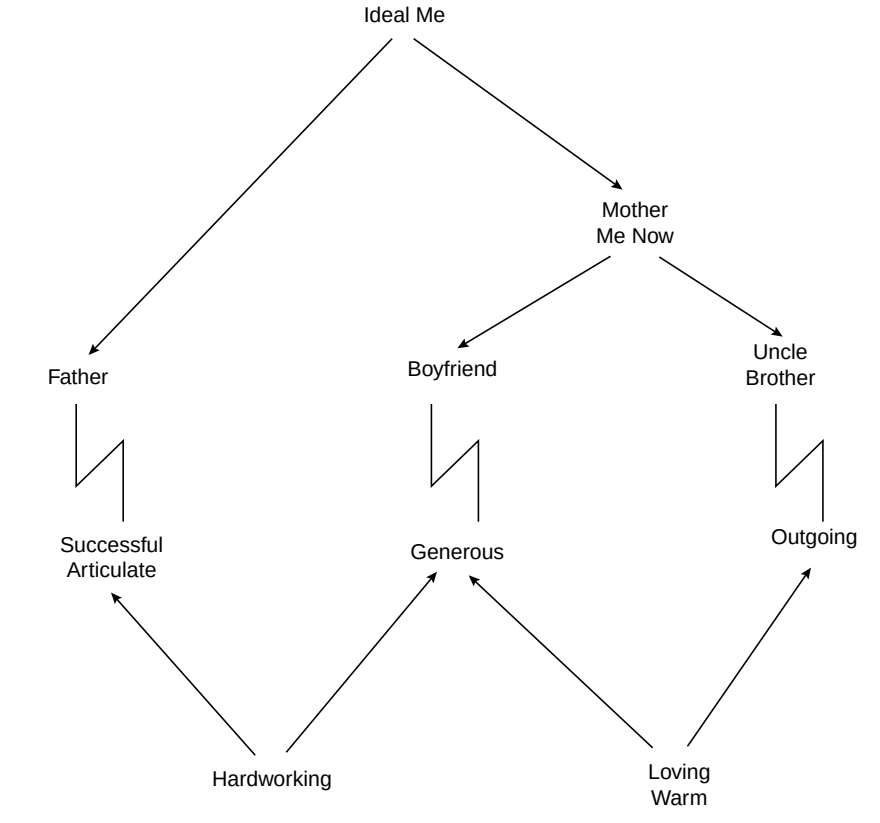

Figure 8.8 *Hierarchical classes representation of hypothetical data in Table 8.6.* (Source: Rosenberg et al., 1996.)

the upper and lower halves of the figure. The object *bundles* are the sets of objects associated with the same attributes (i.e. they are linked by a zigzag line), and similarly for the attribute bundles. All the objects in a bundle are associated with the same attribute bundle and vice versa. An example is the 'Ideal Me' and 'Father' object bundle, which is associated with the 'Hardworking', 'Successful' and 'Articulate' attribute bundle.

One hopes that, in fitting this model to real data, a small number of low-level clusters (bundles) will be found in which all members have exactly the same attributes. In practice, however, this ideal structure is usually impossible to achieve and there will be discrepancies between the model and the data.

The algorithm operates by fitting a structure to data such that the number of discrepancies between the structure obtained and the data is a minimum, using a Jaccard coefficient as the goodness-of-fit measure (see Chapter 3). The rank of

the model (the number of lowest-level clusters) is indicated by the number of zigzag lines in Figure 8.8. It has to be chosen as a trade-off with the goodness of fit, and typically the investigator chooses a rank where the goodness of fit changes little in comparison with the previous increases. In order to avoid local minima in the solution, an initial ordering of either rows or columns is found (e.g. through a conventional cluster analysis on the objects).

Leenen *et al*. (2008) have developed a new stochastic extension to the HICLAS model using Bayesian estimation. According to the authors, 'the benefit of the new extension is that the relation between the predicted values and the observed values is made explicit thanks to the inclusion of one or two error-probability parameters'. In addition to its other advantages over the original deterministic HICLAS algorithm, such as providing model checking and selection criteria, this model-based approach also potentially allows the fixing of any partial or full ordering which may be known *a priori.*

8.5.2 Application of hierarchical classes to psychiatric symptoms

In an application in psychiatry, Gara *et al.* (1998) described 1455 patients using primary care facilities at a community clinic in terms of 41 symptoms. These were grouped into eight body/organ systems (pseudoneurological (PN), gastrointestinal (GI), genitourinary (GU), musculoskeletal (MS), female reproductive urinary (FR), cardiorespiratory (CR), headache (H) and other pain and skin). A hierarchical classes model was fitted, giving a good overall fit to the data ($\kappa = 0.73$). A Jaccard coefficient for comparing clusters and symptoms was also calculated: for example, the coefficient for blurred vision compared to its associated cluster was 0.387. Most coefficients were greater than 0.7, but pseudoneurological and skin complaints (which were rare) did not fit well. Figure 8.9 shows the symptom clusters. The bottom line of the diagram consists of symptom clusters which can be identified with patient clusters (e.g. F has patients with exclusively cardiorespiratory symptoms). Clusters A–E are supersets describing patients with combinations of symptoms (e.g. patients in A have all symptom types). Note that the hierarchy applies to the symptoms (i.e. it is the attribute half of the model).

The authors discuss the relationship of this analysis to the results of a grade-of-membership analysis (a type of *fuzzy analysis:* see Section 8.7.1) on similar data (Swartz *et al*., 1986). They found a general convergence in the result, but preferred the hierarchical classes analysis as it was able to highlight the role of pseudoneurological symptoms, which always co-occur with all the other symptoms in the superset A, and were thus interpreted as evidence for somatization.

8.5.3 The error variance technique

A technique which combines features of both the direct clustering and tree fitting methods is the *error variance* approach of Eckes and Orlik (1993). The basic

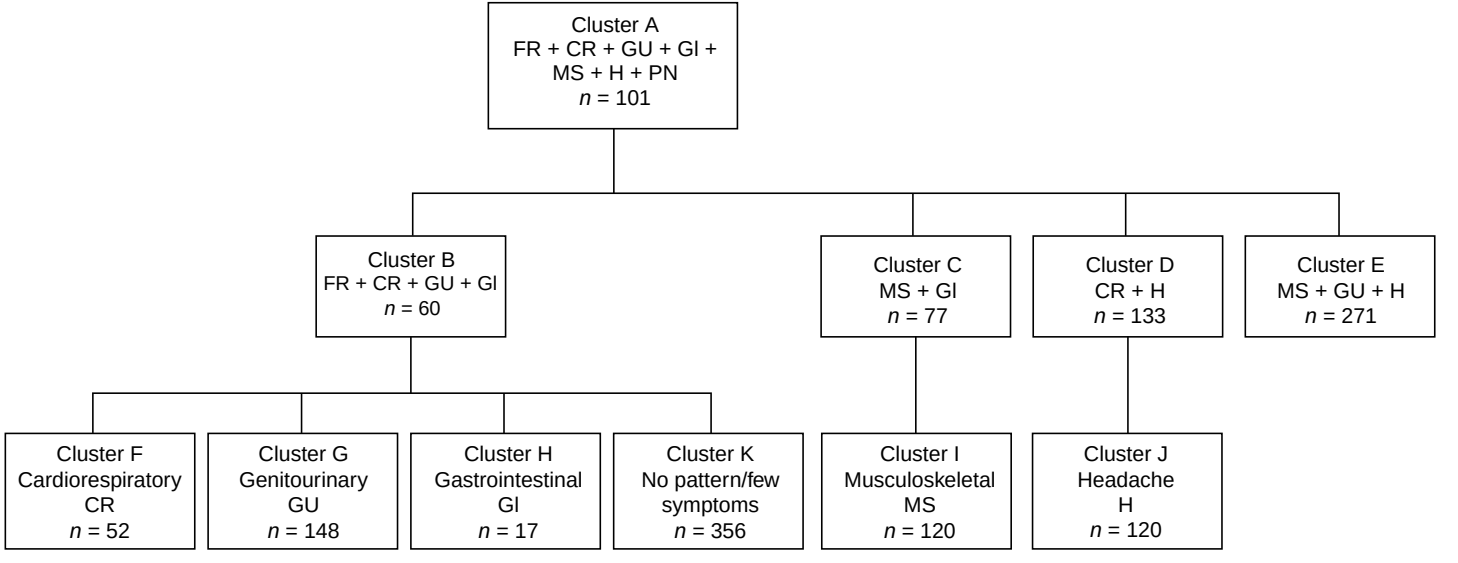

Figure 8.9 *Hierarchical classes analysis of 41 symptoms reported by 1455 patients using primary care facilities at a community clinic: pseudoneurological (PN), gastrointestinal (GI), genitourinary (GU), musculoskeletal (MS), female reproductive (FR), cardiorespiratory (CR), headache (H).* (Taken with permission of the publisher, Elsevier, from Gara et al., 1998.)

method is hierarchical and produces a dendrogram, although a modification allows overlapping clusters to be obtained. The method combines elements (either rows or columns or clusters obtained from them) in such a way as to minimize the increase in the internal heterogeneity of a two-mode cluster (the mean squared deviation or MSD), in a manner analogous to Ward's method for one-mode data (see Chapter 4).

Given a data matrix $\mathbf{X}$, the MSD for a cluster is the mean squared deviation of entries x_{tj} in the corresponding submatrix $\mathbf{X}_r$, from the maximum entry m in $\mathbf{X}$. The procedure involves merging one-mode clusters (either objects or variables) or two-mode clusters (containing both objects and variables), at each stage merging those with the smallest increase in MSD. In calculating this, only those cells representing new combinations are counted. This is illustrated in the example below, where, for example, in step 6, A1B2, A1B3, A2B2, A2B3, A3B1 and A4B1 are new combinations.

The MSD can be written as

$$MSD_r = s_r^2 + (\bar{x}_r - m)^2, \tag{8.12}$$

where s_r^2 is the variance of data in the rth cluster, and $\bar{x}_r$ is the mean. A measure of cluster cohesion is the *centroid effect ratio* (CER):

$$CER_r = \bar{x}_r^2/(s_r^2 + \bar{x}_r^2). \tag{8.13}$$

The CER reflects the contribution of the 'mean cluster effect size' to the 'total cluster effect size'. Clusters with a low CER have a low cohesion. If the CER is less than about 80%, the authors suggest excluding the cluster from further consideration.

The procedure is now illustrated on the data in Table 8.7. The steps in the procedure are as follows:

Step	CER	Two-mode cluster	Increase in MSD
1	1.00	{A2, B1}	$0 = (6.8 - 6.8)^2$
2	1.00	{A3, B3}	$0.49 = (6.1 - 6.8)^2$
3	1.00	{A4, B2}	$1.00 = (5.8 - 6.8)^2$
4	0.94	{A1, A2, B1}	$7.29 = (4.1 - 6.8)^2$
5	0.93	{A3, A4, B2, B3}	$11.60 = [(3.6 - 6.8)^2 + (3.2 - 6.8)^2]/2$
6	0.83	{A1, A2, A3, B1, B2, B3}	$19.63 = [(3 - 6.8)^2 + (3.1 - 6.8)^2 + (2.3 - 6.8)^2 + (2.4 - 6.8)^2 + (1.9 - 6.8)^2 + (1.7 - 6.8)^2]/6$

This example is strictly hierarchical, but overlapping clusters can be obtained by adding elements (rows or columns) to an existing cluster such that the increase in MSDs is below a threshold, and again a resulting CER of 80% would be a lower limit for including a new element.

Table 8.7 Hypothetical data to illustrate direct clustering of data matrices using the error variance method.

Object	Variable		
	B1	B2	B3
A1	4.1	1.9	3.1
A2	6.8	2.3	2.4
A3	1.7	3.6	6.1
A4	3.0	5.8	3.2

8.5.4 Application of the error variance technique to appropriateness of behaviour data

Eckes and Orlik (1993) give two examples of the application of this technique. One is concerned with brand-switching between soft drinks. The other examines the perceived appropriateness of various types of behaviour in different situations, rated by 52 people; the data had previously been analysed using two separate cluster analyses, one for variables and one for objects (Price and Bouffard, 1974), and are given in Table 8.8. Before clustering, each behaviour element was duplicated and multiplied by -1 to form a 15×30 matrix, so that inappropriateness as well as appropriateness of behaviours would be represented. The resulting, not altogether surprising, clusters of behaviours and situations are shown in Table 8.9. Elements have been added to the pre-existing clusters to form overlapping clusters, and the results show that 'job interview' allows for only one appropriate behaviour (talk), whereas 'own room' allows for any of the behaviours. The previous hierarchical analyses were not considered useful because they provided only separate classifications of the situations and behaviours, with no link between them.

8.6 Clustering with constraints

Clustering with constraints is necessary when the membership of clusters is to be restricted in some way, and often occurs when objects and clusters need to retain their spatial relationships. This situation is commonly encountered in geographical or image processing applications. Semi-supervised clustering methods should be mentioned as an important development which involves the use of constraints. To give just one example, in internet clustering of documents, users can feed back their response to the relevance of documents presented to them by a search engine (see Cohn, Caruana and McCallum, Chapter 2 *Semi-supervised clustering with user feedback* in Basu *et al*., 2009). A recent wide-ranging review by Basu *et al.* (2008) gives the theoretical background and many examples of constrained clustering.

While spatial constraints are usually two-dimensional (or occasionally three- or four-dimensional if time is included), one-dimensional constraints arise in

Table 8.8 Perceived appropriateness of various types of behaviour in different situations, rated by 52 people[a].

Situation	Run	Talk	Kiss	Write	Eat	Sleep	Mumble	Read	Fight	Belch	Argue	Jump	Cry	Laugh	Shout
Class	2.52	6.21	2.10	8.17	4.23	3.60	3.62	7.27	1.21	1.77	5.33	1.79	2.21	6.23	1.94
Date	5.00	8.56	8.73	3.62	7.79	3.77	3.12	2.88	3.58	2.23	4.50	4.42	3.04	8.00	3.79
Bus	1.44	8.08	4.27	4.87	5.48	7.04	5.17	7.17	1.52	2.15	4.17	3.12	3.08	7.10	3.00
Family dinner	2.56	8.52	4.92	2.58	8.44	2.29	2.54	3.96	1.67	2.50	3.25	2.29	3.21	7.13	1.96
Park	7.94	8.42	7.71	7.00	8.13	5.63	5.40	7.77	3.06	5.00	5.06	7.42	5.21	8.10	6.92
Church	1.38	3.29	2.38	2.85	1.38	1.77	3.52	3.58	0.62	1.42	1.92	1.71	3.13	2.60	1.33
Job interview	1.94	8.46	1.08	4.85	1.73	0.75	1.31	2.48	1.04	1.21	1.83	1.48	1.37	5.88	1.65
Sidewalk	5.58	8.19	4.75	3.39	4.83	1.46	4.96	4.81	1.46	2.81	4.08	3.54	3.71	7.40	4.88
Movies	2.46	4.98	6.21	2.73	7.48	4.08	4.13	1.73	1.37	2.58	1.71	2.31	7.15	7.94	2.42
Bar	1.96	8.25	5.17	5.38	7.67	2.90	6.21	4.71	1.90	5.04	4.31	3.75	3.44	8.23	4.13
Elevator	1.63	7.40	4.79	3.04	5.10	1.31	5.12	4.48	1.58	2.54	2.58	2.12	3.48	6.77	1.73
Restroom	2.83	7.25	2.81	3.46	2.35	2.83	5.04	4.75	1.77	5.12	3.48	3.65	4.79	5.90	3.52
Own room	6.15	8.58	8.52	8.29	7.94	8.85	7.67	8.58	4.23	6.81	7.52	6.73	8.00	8.17	6.44
Dorm lounge	4.40	7.88	6.54	7.73	7.19	6.08	5.50	8.56	2.40	4.00	4.88	4.58	3.88	7.75	3.60
Football game	4.12	8.08	5.08	4.56	8.04	2.98	5.23	3.69	2.04	3.85	4.98	7.12	4.31	7.90	7.94

[a]The higher the score, the more appropriate the behaviour in the situation.
Source:Price and Bouffard (1974).

Table 8.9 Clusters of behaviours and situations from data in Table 8.8. Elements have been added to the pre-existing clusters to form overlapping clusters.

Cluster	Original elements (non-overlapping)		Added elements (overlapping)	
	Behaviours	Situations	Behaviours	Situations
A	*Fight, run*	Church, class, sidewalk, elevator, restroom, bus		Job interview, bar, movies, family dinner
B	*Sleep, kiss, belch, mumble, cry, jump, shout, eat, argue, read*	Job interview	Talk, *fight, run*	Church
C	Laugh, eat, kiss	Bar, movies, dorm lounge, family dinner, park, football, date	Talk, *fight*	Own room, sidewalk, bus, elevator
D	Sleep, talk, read, write, cry, mumble, argue, belch, jump, shout, run	Own room	Kiss, *laugh, eat*	Park

Note: behaviours in italics are considered inappropriate in the respective situations.

stratigraphy, in areas such as archaeology and geology. Applications that require one-dimensional temporal constraints include the indexing and retrieval of multimedia documents. These may contain excerpts from audio or video tapes, for example scenes from films, annotated with text describing their features. Clustering these annotations should maintain their correct temporal relationship (see Yeung *et al*., 1996).

In some situations, constraints may not be spatially or temporally defined. In the globalization example in Section 4.5.3, cities were clustered according to measures of economic activity. If the purpose of the clustering had been, say, to locate a new company, the existence of good transport links might have been an appropriate constraint rather than geography. Constraints can also be applied, not to the clustering, but to the choice of cluster representative. Girgensohn and Boreczky (2000), for example, clustered videos of meetings in an unconstrained manner, but *keyframes* (frames considered to represent the clusters) were chosen to satisfy various constraints, for example to produce a fixed number of keyframes with an approximately even distribution in time.

Model-based methods can be adapted readily to the incorporation of constraints. Tree fitting methods, both ultrametric and additive, were briefly mentioned in Section 8.5, and the topologies of such trees (see Section 4.4.1) can be constrained as discussed by De Soete and Carroll (1996). These authors give an example of constraining an (additive) kinship tree so that lineal (e.g. father–son) and collateral (e.g. uncle–nephew) relationships are maintained in the appropriate paths. With spatial constraints, the actual distance between two objects can be incorporated into the metric computed from the clustering variables (see Jain and

Farrokhina, 1991, for example). One of the standard hierarchical or partitioning methods can then be employed, using this modified proximity matrix. A similar approach has been described for the DBSCAN algorithm. The difficulty with this approach is to weight the external contribution to the proximity with that defined on the basis of the clustering variables. A more commonly used method, especially for spatial constraints, involves the concept of contiguity, where a matrix is defined whose elements are 1 if two objects are contiguous and 0 otherwise. The contiguity matrix approach is now described in more detail.

8.6.1 Contiguity constraints

In contiguity-constrained clustering, standard distance measures and clustering methods can be employed, but the optimization procedure is constrained on the basis of the contiguity matrix. This is a binary matrix indicating which units are contiguous, most easily defined if the units to be clustered have a natural boundary in space such as an administrative department. Otherwise an unambiguous definition based on spatial relationships is used, such as a contiguity graph, for example the *Voronoi diagram.* In this, the plane in which the units to be clustered lie is subdivided into polygonal regions such that the points in each region are closer to the candidate points than to any other point. Points whose regions share a boundary are said to be contiguous. A mathematically equivalent representation of contiguity is *Delaunay triangulation,* in which contiguous points are joined by line segments.

Where there are very large numbers of candidate points, the spatial field in which the objects lie may be divided into units using a regular grid, for example pixels in image processing. In this situation point B is contiguous with point A if it is situated in one of the four or eight units which surround B. Figure 8.10 shows the Voronoi and grid definitions of contiguity.

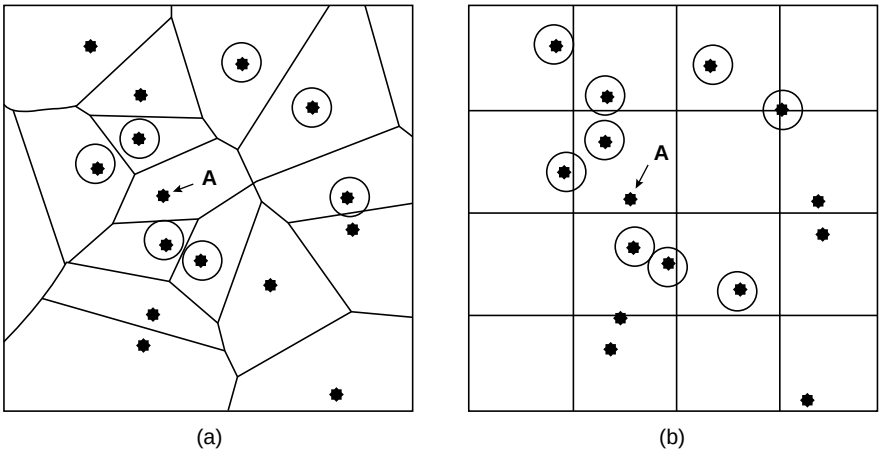

Figure 8.10 *Definitions of contiguity with object A, defined in terms of (a) the Voronoi diagram and (b) the eight surrounding cells in a grid. Circled points are contiguous with A.*

Contiguity graphs, and applications of constrained clustering making use of them, are described by Gordon (1999).

Once an appropriate contiguity matrix has been defined, standard partitioning or hierarchical methods can be applied with the appropriate modification to the algorithm so as to ensure that only contiguous points are clustered. Contiguity constraints can be applied, for example, in the package `Clustangraphics`. Maravalle *et al.* (1997) discuss computational issues in contiguity-constrained hierarchical clustering. The modified methods do not necessarily retain properties of their parent method, such as avoiding inversions (see Section 4.4.3). Ferligoj and Batagelj (1982) discuss constrained hierarchical methods and show that, of widely used methods, only constrained complete linkage does not give rise to inversions. More recent work by Murtagh (1995) also considered inversions, and showed that constrained single linkage also avoids them, so long as the *single* objects that are linked are contiguous (a technique known as *contiguity-constrained single link*). Two methods also discussed by Murtagh are the constrained centroid and minimum variance methods. These can produce inversions but have the advantage that they give rise to natural regional representatives (the exemplars or typical members).

One-dimensional constraints, where clustering is required to follow a given order, can be treated with special methods, as developed by Gordon (1973) for an application concerning palaeoecological samples from a vertical bore. Partitions of contiguous samples can be defined by (virtual) 'markers' placed between neighbouring samples; the number of possible placements and hence possible partitions is $(n-1)!/[(g-1)!(n-g)!]$, where g is the number of groups, and n the number of objects. Gordon (1980) suggests two approaches to finding the optimal partition. The first uses a divisive algorithm that begins by finding a single marker leading to minimum within-group sums of squares. Each of the two groups is then optimally divided, choosing the division that leads to maximum decrease in the sum of squares. The algorithm continues by successive division of existing groups.

The second procedure described by Gordon (1980) involves a dynamic programming algorithm. Let $s(i, j)$ denote the within-group sum of squares of objects i to j inclusive, and let $t(g, k)$ denote the total within-group sum of squares when objects 1 to k are optimally divided into g groups. We require $t(g, n)$ for $2 \leq g \leq n$, together with the corresponding markers. The solution is built up recursively, evaluating $\{t(g, k), k = g, g + 1, \ldots, n; g = 1, 2, \ldots, n\}$ by means of the following formulae:

$$\begin{aligned} t(1,k) &= s(1,k)(1 \leq k \leq n) \\ t(g,k) &= \min_{g-1 \leq i \leq k-1} [t(g-1,i) + s(i+1,k)](g \leq k \leq n; 2 \leq g \leq n). \end{aligned} \tag{8.14}$$

Equation (8.14) involves dividing the first k objects into two classes, the first class containing $g-1$ groups and the second class containing 1 group of objects. This is equivalent to placing the last marker at position i, and the algorithm finds the

optimal value of i. The complete set of $g-1$ markers is obtained using a *trace-back* procedure.

8.6.2 Application of contiguity-constrained clustering

An example of the use of a contiguity constraint in clustering is given in a study of the incidence of breast cancer in Argentina. Wojdyla *et al.* (1996) clustered administrative departments of Argentina to form regions, using a variant of the procedure described by Ferligoj and Batagelj (1982). Contiguous departments were clustered together according to their Euclidean distance, as computed from sociodemographic variables, with contiguity defined on the basis of shared administrative boundaries. Clustering was terminated, not according to a standard clustering criterion, but when regions had attained the minimum population regarded as sufficient to make valid statistical inferences regarding rates of breast cancer. Standardized mortality rates from breast cancer for each region are shown in Figure 8.11(a), and the regions found by clustering are shown in Figure 8.11(b). The authors concluded that, on the basis of the aggregated data, only two regions (Rosario and Córdoba, relatively underdeveloped regions) had significantly higher rates than the national rate. This is in contrast with the more irregular picture obtained before clustering.

8.7 Fuzzy clustering

In *fuzzy clustering*, objects are not assigned to a particular cluster: they possess a membership function indicating the *strength of membership* in all or some of the clusters. In most of the previous clustering techniques described in this text, 'strength of membership' has been either zero or one, with an object being either in or not in a cluster, except perhaps in the case of the mixture approach of Chapter 6, where it might be taken as the posterior probability of belonging to a cluster. In fuzzy clustering jargon, methods where strength of membership is zero or one are known as *crisp* methods.

Fuzzy clustering has two main advantages over crisp methods. Firstly, memberships can be combined with other information. In particular, in the special case where memberships are probabilities, results can be combined from different sources using Bayes' theorem. Secondly, the memberships for any given object indicate whether there is a 'second best' cluster that is almost as good as the 'best' cluster, a phenomenon which is often hidden when using other clustering techniques.

In a fuzzy cluster analysis, the number of subsets is assumed known, and the membership function of each object in each cluster is estimated using an iterative method, usually a standard optimization technique based on a heuristic objective function. In general, membership functions do not obey the rules of probability theory, although, once found, memberships can be scaled to lie between zero and one, and can then be interpreted as probabilities. (Mixtures methods, where

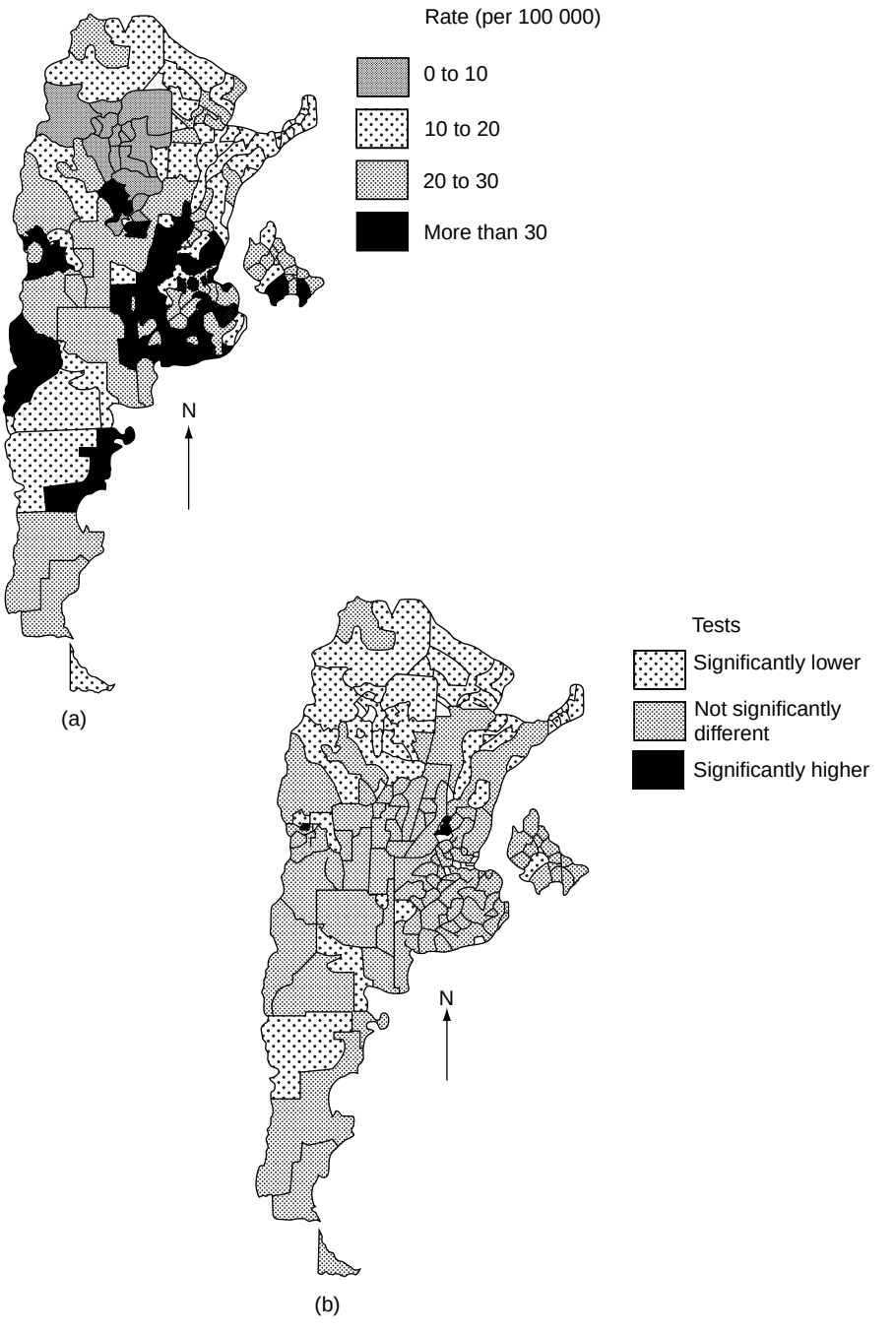

Figure 8.11 *Standardized mortality rates from breast cancer in the departments and regions of Argentina, (a) before and (b) after constrained clustering.* (Taken with permission of the publisher, John Wiley & Sons Ltd, from Wojdyla et al., 1996.)

the memberships *are* true probabilities, and where probability theory underlies the estimation method, have been discussed in Chapter 6.)

The concept of a membership function derives from *fuzzy logic*, an extension of Boolean logic in which the concepts of true and false are replaced by that of partial truth. Boolean logic can be represented by set theory, and in an analogous manner fuzzy logic is represented by fuzzy set theory. Such techniques were originally developed for the description of natural language (Zadeh, 1965). As an example of a fuzzy membership function, Figure 8.12 shows a possible function for the description of IQ.

The connection between fuzzy cluster analysis and fuzzy logic is usually only through the application of membership functions, and not the more comprehensive theory. However, an example in which the principles of fuzzy logic (as well as the membership functions) are used to derive a clustering algorithm is given by Zhang *et al.* (1998) in an application to a small data set concerned with monitoring mechanical equipment. Laviolette *et al.* (1995), along with a number of discussants in the same volume, compare fuzzy and probabilistic approaches in general, and among these contributions is a discussion of fuzzy cluster analysis (Rousseeuw, 1995).

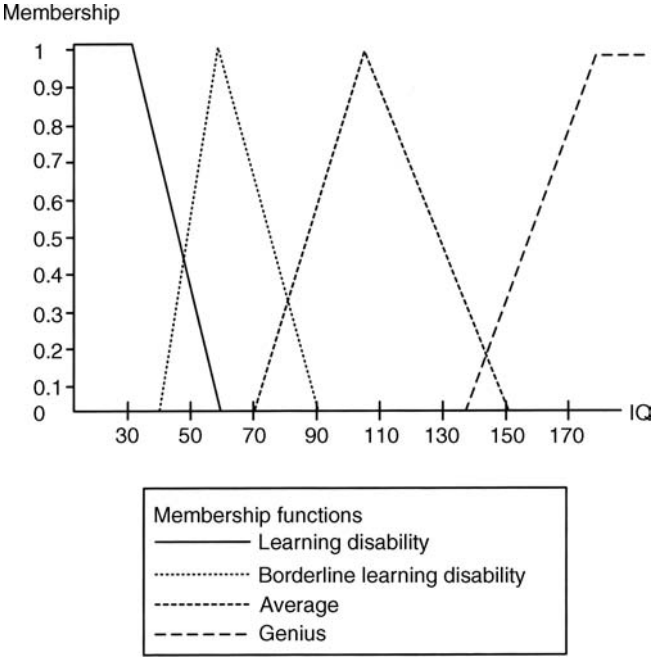

Figure 8.12 *A fuzzy membership function for the verbal description of intelligence quotient (IQ); an example is IQ 85, which has memberships of about 0.4 and 0.2 in the sets 'average' and 'borderline learning disability', respectively.*

The following subsection gives examples of three heuristic fuzzy methods: for binary data, for continuous data and for a proximity matrix.

8.7.1 Methods for fuzzy cluster analysis

Grade-of-membership (GOM) analysis (Woodbury and Manton, 1982; Manton *et al*., 2004) has been proposed for binary data, and has been used in health-related applications, for example psychiatry and genetics. Grade-of-membership analysis assigns to cases a 'grade of membership' in two or more latent classes, and is thus similar to latent class analysis (see Chapter 6). In GOM and latent class analysis, probabilities of cluster membership are provided and, in that sense, both methods are fuzzy. The difference is that in GOM analysis, the grades of membership are estimated as parameters as part of the clustering process, whereas in latent class analysis the probabilities of class membership are estimated once the latent class model has been estimated. Further details are given in Woodbury *et al.* (1994), and a recent application profiling doctors' practice styles in pain management is given by Maelzel *et al.* (2000).

For continuous data, a weighted sum-of-squares criterion leading to *fuzzy k*-means (also known as fuzzy *c*-means), clustering has been described by Bezdek (1974). For a set of n objects and g clusters this is, for data vectors $\mathbf{x}_i$,

$$\sum_{t=1}^{g}\sum_{i=1}^{n} u_{it}^{v}\, d^2(\mathbf{x}_i, \mathbf{m}_t), \tag{8.15}$$

where $\mathbf{m}_t$ is the centre of cluster t, $u_{it} \geq 0$ for all $i = 1,\ldots, n$ and $\Sigma_{t=1}^{g} u_{it} = 1$. The memberships u_{it} are unknown; the $d(\mathbf{x}_i, \mathbf{m}_t)$ are Euclidean distances between the data point and the cluster centres; v is called the *fuzzifier* and affects the final membership distribution; typically it is 2 (setting $v = 1$ leads to the crisp solution). The cluster centres are weighted cluster means, given for the kth variable by

$$m_t = \sum_{i=1}^{n} u_{it}^{v}\mathbf{x}_i \Bigg/ \sum_{i=1}^{n} u_{it}^{v}. \tag{8.16}$$

The clustering algorithm computes the optimal memberships by minimizing (8.15); see, for example, Hathaway and Bezdek (1988) for details of algorithms. If the u_{it} are restricted to zero or one, the usual k-means method is obtained (see Chapter 5). Kettenring (2009) illustrates the use of fuzzy k-means in the rapid valuation of portfolio assets in a study of the use of cluster analysis in patents. Nasibov and Ulutagay (2010) compare fuzzy k-means with fuzzy neighbourhood DBSCAN, a fuzzy version of DBSCAN, described earlier in Section 8.3.

As in traditional agglomerative or optimization methods, various other choices of d can be made, such as city block or Mahalanobis distance. Kaufman and Rousseeuw (2005) describe a method (called 'FANNY') in which the

following objective function is minimized:

$$\sum_{t=1}^{k}\left\{\sum_{i,j=1}^{n} u_{it}^2 u_{jt}^2 d(\mathbf{x}_i,\mathbf{x}_j)\right\Big/ 2\sum_{j=1}^{n} u_{jt}^2\right\}, \tag{8.17}$$

where $d(\mathbf{x}_i,\mathbf{x}_j)$ are dissimilarities between objects. Note that in this case the means do not enter the objective function, and are not squared. For this method only a proximity matrix is required since there is no need to estimate central values for the clusters. Furthermore, the inclusion of $d(\mathbf{x}_i,\mathbf{x}_j)$ rather than $d(\mathbf{x}_i,\mathbf{x}_j)^2$ means that the method is relatively robust to nonspherical clusters.

8.7.2 The assessment of fuzzy clustering

The *silhouette plot* (Rousseeuw, 1987), an example of which is given in Figure 5.5, is useful in connection with partitioning methods in general, but particularly so in the context of fuzzy clustering. It reflects the strength of a classification to the nearest crisp cluster, compared to the next best cluster. The width of each bar is the 'silhouette value', which is one if the object is well classified, zero if it is intermediate between the best and second best, and negative if it is nearest to the second-best cluster (see also Section 5.5).

Dunn's partition coefficient (Dunn, 1974) is a criterion for assessing the strength of membership specifically designed for fuzzy methods. When normalized to lie in the range [0,1], it has the form

$$\left(k\sum_{i=1}^{n}\sum_{t=1}^{k}\frac{u_{it}^2}{n}\right)\Big/(k-1), \tag{8.18}$$

and is equal to 1 for completely distinct clustering.

Dunn's coefficient and silhouette plots give information to allow a number of clusters to be chosen so that a balance can be struck in the degree of fuzziness in different clusters. However, like other internal methods of cluster validation, they do not provide a definitive guide to the number of clusters, and this is a subject of continuing research (Pal and Bezdek, 1995). It is usually advisable to use a low-dimensional plot such as principal components analysis in addition to the silhouette plots in order to assess the degree of fuzziness present in the data.

8.7.3 Application of fuzzy cluster analysis to Roman glass composition

The chemical composition of Roman glass found in Norway has been studied by Christie *et al.* (1979) with a view to identifying its origins. A subset of the data was reanalysed by Baxter (1994) in order to illustrate the results obtained by four standard methods of hierarchical cluster analysis. Baxter concluded that there were probably two main clusters, possibly with one outlier. This was also concluded by Christie *et al.* and is suggested by the principal components plot in Figure 8.13,

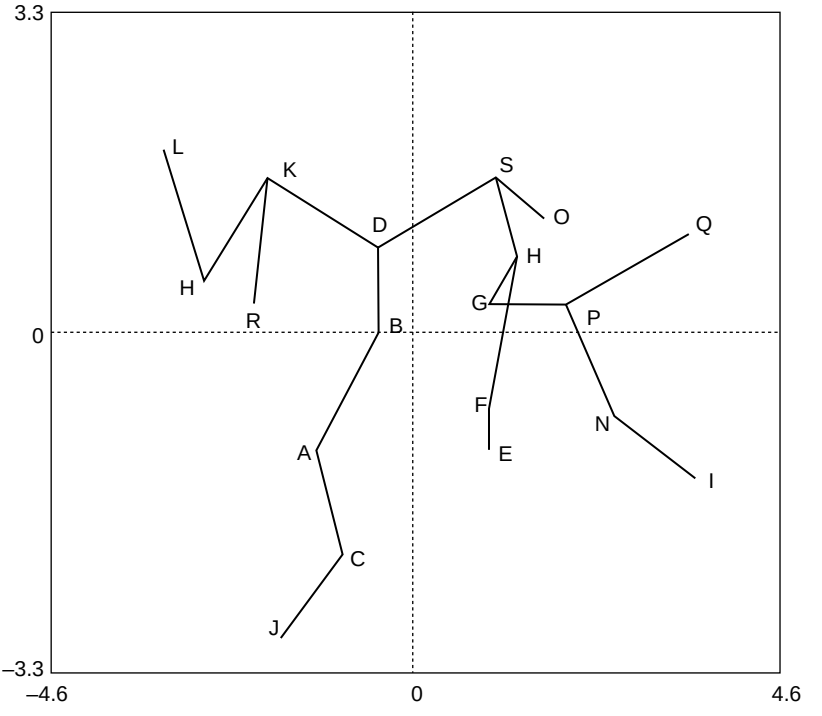

Figure 8.13 *Principal components plot of Roman glass composition (standardized to unit standard deviation), showing the minimum spanning tree; note the intermediate position of D, between two main clusters (see also Table 8.10).* (Source: Baxter, 1994.)

where there seem to be two 'clouds' of points, although the two main clusters are not very distinct. However, one point (object D) is intermediate between the two, and links the clusters in the minimum spanning tree.

The data, which are the percentage compositions of eight oxides in 19 specimens, are shown in Table 8.10, along with the average linkage clusters suggested by Baxter, the nearest crisp clusters in the six- and two-cluster case, and the fuzzy memberships for the two-cluster solution. The method used was that proposed by Kaufman and Rousseeuw (see Section 8.7.1). The two-cluster solution is shown in the silhouette plot (Figure 8.14).

The normalized Dunn coefficient is 0.06 for the two-cluster solution, which is close to zero, indicating a very high fuzziness. The silhouette plot shows that cluster 2 is less fuzzy (average width 0.41) than cluster 1 (average width 0.20) and that object D has a negative width indicating a bad classification; this is due to its intermediate position between the two clusters. If a six-cluster solution is chosen, the average width drops, and while cluster 3 is more compact, clusters 1 and 2 are more dispersed. Two singleton clusters J and L now become apparent and the Dunn coefficient is higher (0.18). The nearest crisp solutions to the fuzzy two-, three- and

Table 8.10 Data and fuzzy cluster analysis of Roman glass composition. Data are percentages of eight oxides.

Specimen	Ti	Al	Fe	Mn	Mg	Ca	Na	K	Fuzzy 6-cluster	Average linkage 2-cluster	Fuzzy 2-cluster	Memberships for fuzzy 2-cluster solution (%)	
A	0.10	2.0	0.8	1.5	1.18	6.3	18.0	0.58	1	1	1	58	42
B	0.10	2.0	0.5	1.4	1.16	6.4	18.4	0.43	2	1	1	56	44
C	0.10	2.0	1.0	1.2	0.77	7.0	19.0	0.61	2	1	1	55	45
D	0.20	2.0	0.7	1.2	0.90	6.1	19.3	0.36	2	2	1	56	44
E	0.09	1.8	0.95	1.0	0.70	6.2	16.2	0.45	3	2	2	42	58
F	0.09	1.8	1.1	0.9	0.68	6.0	16.1	0.44	3	2	2	43	57
G	0.08	1.7	0.6	1.4	0.71	6.35	17.6	0.37	4	2	2	34	66
H	0.08	1.7	0.6	1.3	0.70	6.2	17.2	0.32	4	2	2	32	68
I	0.05	1.5	0.2	0.02	0.53	6.2	18.9	0.45	2	2	2	43	57
J	0.30	1.8	1.0	1.4	1.01	8.8	18.1	0.53	5	0[a]	1	54	46
K	0.30	2.2	1.0	1.9	1.06	6.2	18.6	0.34	1	1	1	64	35
L	0.35	2.8	1.2	2.0	0.96	5.9	18.5	0.37	6	1	1	62	38
M	0.30	2.5	1.0	2.0	0.96	6.7	18.5	0.41	1	1	1	65	35
N	0.07	1.5	0.45	0.95	0.58	6.85	17.5	0.35	4	2	2	35	65
O	0.07	1.5	0.45	1.0	0.78	6.25	19.4	0.27	2	2	2	43	57
P	0.08	1.6	0.5	1.1	0.65	6.2	17.5	0.37	4	2	2	29	71
Q	0.06	1.3	0.3	0.85	0.50	5.9	16.8	0.29	4	2	2	37	63
R	0.35	2.2	1.0	1.5	1.20	6.5	18.0	0.40	1	1	1	65	35
S	0.07	2.0	0.4	1.2	0.80	6.0	18.0	0.30	2	2	2	40	60

[a]Interpreted as an outlier in the average linkage solution.
Source: Baxter (1994); original compositional data from Jackson (1994).

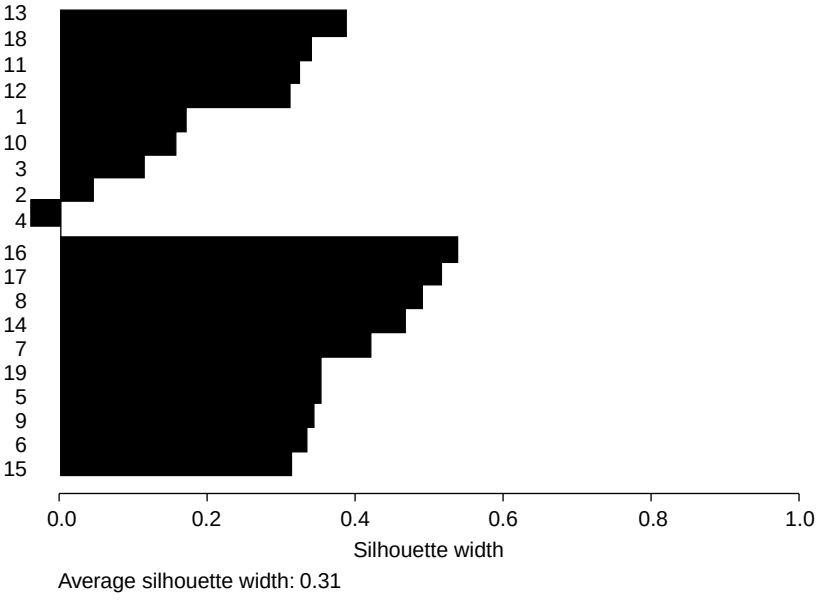

Figure 8.14 *Silhouette plot of Roman glass clusters, two-cluster solution (see also Figure 8.13 and Table 8.10). The width of the bars indicates the strength of clustering for each object. Negative bars indicate unsatisfactory classification, for example object D (no. 4)*

four-cluster solution are very similar to the average linkage solution (see Chapter 4 for a description of the latter technique).

8.8 Clustering and artificial neural networks

Neural networks have received a great deal of attention over the last few years. They have been used for a wide variety of applications, where conventional statistical methods such as regression and discriminant function analysis might normally be employed. But just what are neural networks? Essentially they are computing algorithms that attempt to imitate the computational capabilities of large, highly connected networks of relatively simple elements such as the neurons in the human brain. The interest in such techniques arises from the desire to mimic some of the desirable pattern-recognition type tasks for which the human brain is so well adapted. In the beginning, neural network models were intended as realistic models of neural activity in the human or animal brain, and this is still true in some areas of psychology and biology. But general interest now centres on the computational potential of neural network algorithms or computers without regard for their realism. Cheng and Titterington (1994) and Ripley (1994) provide surveys describing the relevance of neural networks in statistics. Ripley (1996) discusses

neural networks using a style and language familiar to statisticians, and Dewdney (1997, Chapter 5) gives a readable, if somewhat sceptical, general introduction.

8.8.1 Components of a neural network

The three essential features of a neural network are the basic computing elements usually referred to as *neurons*, the network architecture describing the connections between computing units, and the training algorithm used to find values of the network parameters for performing a particular task. A very simple neural network is the *single-unit perceptron* or McCulloch–Pitts neuron (McCulloch and Pitts, 1943). This is illustrated in Figure 8.15.

From a set of 'inputs' (predictors) $x_1, x_2, \ldots, x_p$ and weights $w_1, w_2, \ldots, w_p$, the neuron provides an 'output' (response) y given by

$$y = \text{sign}\left(w_0 + \sum_{i=1}^{p} w_i x_i\right), \tag{8.19}$$

where sign$(\cdot)$ equals 1 if its argument is positive and -1 otherwise. In this simple *binary thresholding* model, the neuron 'fires' or does not fire, depending on whether or not the summation is positive. Networks of artificial neurons such as these are constructed by forming interconnected banks of such elements with xs feeding into every such node, such as the circle in Figure 8.15, and the outputs of these nodes feeding into similar nodes at another level and so on. Such an arrangement is known as a *layered feed-forward neural network* or, equivalently, a *multilayer perceptron.*

Layers in between input and output are called *hidden layers* since they are not observable directly; Figure 8.16 depicts such a neural network with three inputs, a single layer of four hidden units and two outputs.

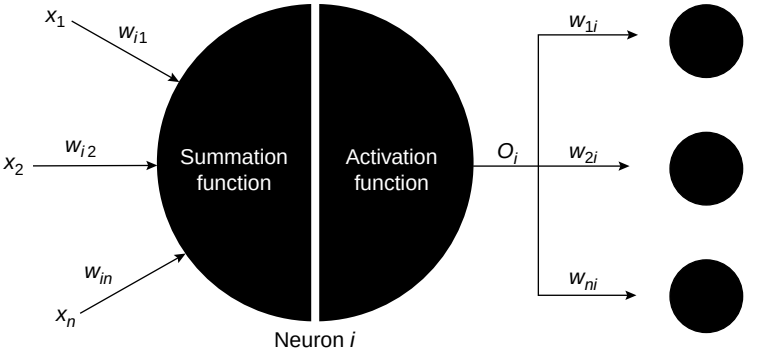

Figure 8.15 *Artificial neuron: weighted inputs are summed, processed by an activation function and output to the next layer of neurons. There is a layer of input nodes, a middle layer of hidden nodes and a layer of output nodes.* (Reprinted by permission of Sage Publications Ltd, from Garson, Neural Networks: An Introductory Guide for Social Scientists, 1998, Sage, London.)

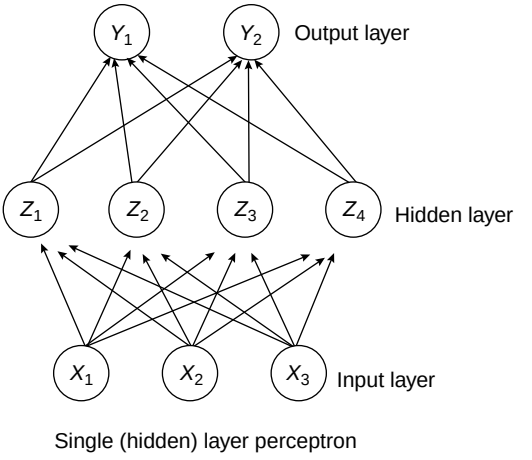

Figure 8.16 *A network diagram representing a single layer neural network with three inputs (predictors), four hidden units and two output (responses).* (Source: Hastie, 1998.)

In more traditional terms the model specified in Figure 8.16 can be written as follows:

$$z_j = \sigma(\alpha_{j0} + \boldsymbol{\alpha}'_j \mathbf{x}), j = 1, \ldots, 4, \tag{8.20}$$

$$\hat{y}_k = f_k(\beta_{k0} + \boldsymbol{\beta}'_k \mathbf{z}), k = 1, 2, \tag{8.21}$$

where $\mathbf{x}' = (x_1, x_2, x_3)$, $\mathbf{z}' = (x_1, x_2, x_3, x_4)$, $\boldsymbol{\alpha}'_j = (\alpha_{j1}, \alpha_{j2}, \alpha_{j3})$ and $\boldsymbol{\beta}'_k = (\beta_{k1}, \beta_{k2}, \beta_{k3}, \beta_{k4})$. The other terms in 8.20 and 8.21 are as follows:

- σ is known as the activation function and is used to allow a possible nonlinearity at the hidden layer; in the simple example above it was the *sign* function, but commonly it is taken to be the *sigmoid* function $\sigma(z) = 1/(1 + \mathrm{e}^{-z})$.
- The parameters α_{ji} and β_{ki} are the weights mentioned previously and define linear combinations of the input vector $\mathbf{x}$ and the hidden output vector $\mathbf{z}$, respectively.
- The intercept terms α_{j0} and β_{k0} are known here as *biases*.
- The function f_k is included to permit a final transformation of the output, for example the inverse logit function when responses should lie in [0,1].

The weights to be used in a neural network model are estimated from the training set data by least squares; for example, for the network described above, by minimizing

$$R(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_k (y_k - \hat{y}_k)^2, \tag{8.22}$$

a criterion that is nonlinear in the parameters. It is not always easy to minimize R, since it may have local minima and typically neural networks are overparameterized, often with more parameters than observations. (In the neural network literature this iterative estimation stage is often described as 'training' the network.)

8.8.2 The Kohonen self-organizing map

Most applications of neural networks have been in an assignment context, where the groups are known *a priori* (see Chapter 1). Details of such applications are given in Ripley (1996). But one well-known neural network method, the self-organizing map (SOM) due to Kohonen (1982), is an example of unsupervised learning because the assignment classes for the output vectors are not known *a priori* (see Kohonen, 1997, for further details and examples). Consequently, the network classifies the observations according to internally generated allocation rules; its performance can therefore be compared with more conventional approaches to cluster analysis.

The Kohonen model is illustrated in Figure 8.17. The network contains two layers:

- an input layer consisting of p-dimensional observations $\mathbf{x}$;
- an output layer (represented by a grid) consisting of k nodes for the k clusters, each of which is associated with a p-dimensional weight $\mathbf{w}$.

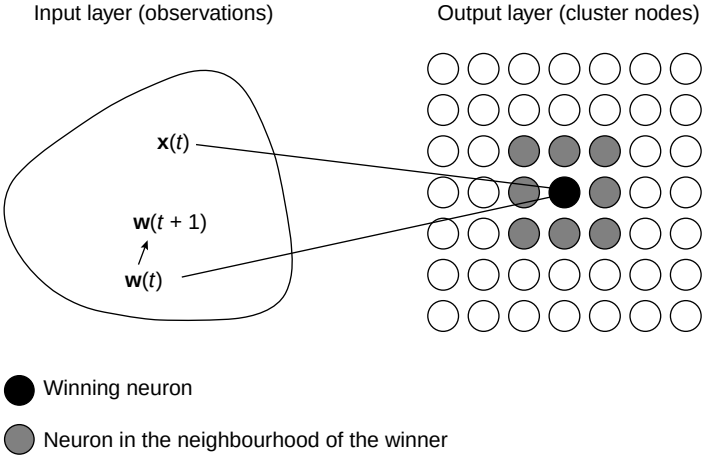

Figure 8.17 *A Kohonen self-organizing map, showing an observation* $\boldsymbol{x}$ *and its closest (winning) neuron at stage* t *in the iterative process. The weights* $\boldsymbol{w}$ *associated with winning neurons (and to a lesser extent the weights of those in their neighbourhoods) are moved towards the observation at stage* t + 1 (adapted from the webpage of the SOMLib Digital Library Project, Vienna University of Technology.)

Classification (clustering) occurs where an input vector is assigned to an output node. Operationally, each output node has a p-dimensional vector of synaptic weights $\mathbf{w}$. The output node is initially assigned a random weight; as the network learns, the input cluster points are provisionally assigned to clusters and the weights are modified. The iterative process eventually stabilizes with the weights corresponding to cluster centres in such a way that clusters that are similar to one another are situated close together on the map (the result being somewhat analogous to multidimensional scaling: see Chapter 2).

The SOM method thus makes the surface of the neurons recreate (i.e. change associated weight values) in accordance with the outside world as represented by the input vectors. In more mathematical terms the process can be described as follows.

- Consider p-dimensional weight vectors associated with neurons, each of the values of which is initially random and in the interval (0, 1).
- A p-dimensional observation, also scaled to be in (0, 1), is presented with the values of this weight vector.
- The Euclidean distance (or some other preferred distance measure) is calculated between the observation and the vector associated with each neuron.
- The neuron with the smallest distance (the 'winner') is then updated, as are a small neighbourhood of neurons around the 'winner'. The winner's weight vector $\mathbf{w}_{\text{old}}$ is brought closer to the input patterns $\mathbf{x}$ as follows:

$$\mathbf{w}_{\text{new}} = \mathbf{w}_{\text{old}} + \alpha(\mathbf{x} - \mathbf{w}_{\text{old}}). \tag{8.23}$$

  The value of α is a small fraction, which decreases as learning takes place, as does the size of the neighbourhood. The excited neurons in the neighbourhood of the 'winner' are updated in an identical manner but with a smaller α.
- As the network 'learns', the weights are modified and the input observations are provisionally assigned to clusters.

It is clear from this description that the SOM procedure is in many respects similar to a standard iterative partitioning method such as k-means, as described in Chapter 5. One difference is that a number of parameters need to be initialized in the Kohonen algorithm at the start. Advantages of the SOM are that it produces a low-dimensional plot as a visual representation of the clustering, and that it can handle very large data sets.

A comparison of the neural network approach to clustering with a number of more conventional methods is reported in Waller *et al.* (1998). These authors applied their chosen methods to 2580 data sets with known cluster structure. Overall, the performance of the Kohonen network was similar to, or better than, the performance of the other methods. Further simplification can be achieved by making use of the topological relationships in the output layer, using methods of constrained clustering (Murtagh, 1995). Ambroise and Govaert (1996) have adopted a probabilistic approach, making use of the neighbourhood interaction

function as in the SOM, but in the context of the EM algorithm (see Chapter 6), which they term a 'Kohonen type EM'. An application involving a very large text database is given by Kohonen *et al.* (2000). See also Janasik *et al.* (2009) for a general description of SOM applied to qualitative data. Interesting astronomical and meteorological applications of neural networks used for cluster analysis are given in Murtagh and Hernández-Pajares (1995). A bibliography of earlier SOM papers is given by Kaski *et al.* (1998) and recent developments are described by Principe and Miikkulainen (2009).

8.8.3 Application of neural nets to brainstorming sessions

In research aimed at categorizing World Wide Web pages based on concepts (rather than keyword searches or hypertext browsing as used, for instance, by Lycos and Yahoo), Chen *et al.* (1996) developed a multilayered self-organizing map for 10 000 internet pages concerned with entertainment (the 'ET-Map'). They then assessed its success with a manually catalogued system. A smaller application was concerned with electronic brainstorming sessions, and this is reported here. In this application, 20 group participants were asked to comment on the most important information technology problems with respect to collaborative systems, using electronic keyboards in parallel. The group generated 201 comments in 30 minutes; the single two-dimensional representation of the layer SOM produced from these is shown in Figure 8.18.

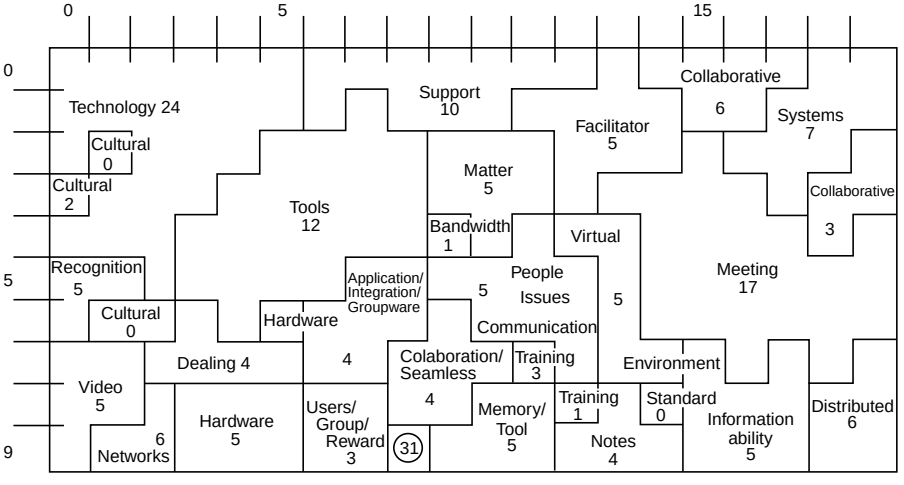

Figure 8.18 *A Kohonen self-organizing map of comments made by participants during an electronic brainstorming session on the future of collaborative systems. Clusters are represented by distinct areas on the map and are characterized by phrases such as 'Tools'.* (Figure from 'Internet categorization and search: A self-organizing approach', in *Journal of Visual Communication and Image Representation,* Volume 7, pp. 88–102, copyright © 1996 by Academic Press, reproduced by permission of the publisher.)

A small-scale experiment compared the performance of a human facilitator (who manually compiled topic lists) with the self-organizing map approach by asking a further eight individuals to revise the lists to provide a 'gold standard'. This showed that the facilitator had better *precision* than the SOM (i.e. a high proportion of the terms identified were gold-standard terms) at 81% compared to 55%. The *recall* (i.e. returning a high proportion of gold-standard terms) was similar at 89% compared to 81%. However, the time to produce the lists was very much in favour of the SOM: 4 minutes compared to 45 minutes. The authors concluded that the SOM output was an efficient algorithm for producing a 'straw man' list that could be used as a basis for improvement.

8.9 Summary

Standard clustering methods have been developed in many directions to encompass realistic situations, such as those involving constraints and overlapping clusters. Many of these developments allow for more comprehensive interpretation of clusters in terms of both objects and variables, or incorporate features such as fuzzy memberships. Application fields such as multimedia documentation, genetics and image analysis, combined with increasing computing power, have prompted some of these developments.