6

Cluster Analysis

6.1 Introduction

> An intelligent being cannot treat every object it sees as a unique entity unlike anything else in the universe. It has to put objects in categories so that it may apply its hard-won knowledge about similar objects encountered in the past to the object at hand (Pinker 1997).

One of the most basic abilities of living creatures involves the grouping of similar objects to produce a classification. The idea of sorting similar things into categories is clearly a primitive one because early humans, for example, must have been able to realise that many individual objects shared certain properties such as being edible, or poisonous, or ferocious, and so on. And classification in its widest sense is needed for the development of language, which consists of words that help us to recognise and discuss the different types of events, objects, and people we encounter. Each noun in a language, for example, is essentially a label used to describe a class of things that have striking features in common; thus animals are called cats, dogs, horses, etc., and each name collects individuals into groups. Naming and classifying are essentially synonymous.

As well as being a basic human conceptual activity, classification of the phenomena being studied is an important component of virtually all scientific research. In the behavioural sciences, for example, these "phenomena" may be individuals or societies, or even patterns of behaviour or perception. The investigator is usually interested in finding a classification in which the items of interest are sorted into a small number of *homogeneous groups* or *clusters*, the terms being synonymous. Most commonly the required classification is one in which the groups are *mutually exclusive* (an item belongs to a single group) rather than *overlapping* (items can be members of more than one group). At the very least, any derived classification scheme should provide a convenient method of organizing a large, complex set of multivariate data, with the class labels providing a parsimonious way of describing the patterns of similarities

B. Everitt and T. Hothorn, *An Introduction to Applied Multivariate Analysis with R*: Use R!,
DOI 10.1007/978-1-4419-9650-3_6, © Springer Science+Business Media, LLC 2011

and differences in the data. In market research, for example, it might be useful to group a large number of potential customers according to their needs in a particular product area. Advertising campaigns might then be tailored for the different types of consumers as represented by the different groups.

But often a classification may seek to serve a more fundamental purpose. In psychiatry, for example, the classification of psychiatric patients with different symptom profiles into clusters might help in the search for the causes of mental illnesses and perhaps even lead to improved therapeutic methods. And these twin aims of *prediction* (separating diseases that require different treatments) and *aetiology* (searching for the causes of disease) for classifications will be the same in other branches of medicine.

Clearly, a variety of classifications will always be possible for whatever is being classified. Human beings could, for example, be classified with respect to economic status into groups labelled *lower class*, *middle class*, and *upper class* or they might be classified by annual consumption of alcohol into *low*, *medium*, and *high*. Clearly, different classifications may not collect the same set of individuals into groups, but some classifications will be more useful than others, a point made clearly by the following extract from Needham (1965) in which he considers the classification of human beings into men and women:

> The usefulness of this classification does not begin and end with all that can, in one sense, be strictly inferred from it–namely a statement about sexual organs. It is a very useful classification because classifying a person as man or woman conveys a great deal more information, about probable relative size, strength, certain types of dexterity and so on. When we say that persons in class *man* are more suitable than persons in class *woman* for certain tasks and conversely, we are only incidentally making a remark about sex, our primary concern being with strength, endurance, etc. The point is that we have been able to use a classification of persons which conveys information on many properties. On the contrary a classification of persons into those with hairs on their forearms between 3/16 and $\frac{1}{4}$ inch long and those without, though it may serve some particular use, is certainly of no general use, for imputing membership in the former class to a person conveys information on this property alone. Put another way, there are no known properties which divide up a set of people in a similar way.

In a similar vein, a classification of books based on subject matter into classes such as dictionaries, novels, biographies, and so on is likely to be far more useful than one based on, say, the colour of the book's binding. Such examples illustrate that any classification of a set of multivariate data is likely to be judged on its usefulness.

6.2 Cluster analysis

Cluster analysis is a generic term for a wide range of numerical methods with the common goal of uncovering or discovering groups or clusters of observations that are homogeneous and separated from other groups. Clustering techniques essentially try to formalise what human observers do so well in two or three dimensions. Consider, for example, the scatterplot shown in Figure 6.1. The conclusion that there are three natural groups or clusters of dots is reached with no conscious effort or thought. Clusters are identified by the assessment of the relative distances between points, and in this example the relative homogeneity of each cluster and the degree of their separation makes the task very simple. The examination of scatterplots based either on the original data or perhaps on the first few principal component scores of the data is often a very helpful initial phase when intending to apply some form of cluster analysis to a set of multivariate data.

Cluster analysis techniques are described in detail in Gordon (1987, 1999) in and Everitt, Landau, Leese, and Stahl (2011). In this chapter, we give a relatively brief account of three types of clustering methods: *agglomerative hierarchical techniques*, *k-means clustering*, and *model-based clustering*.

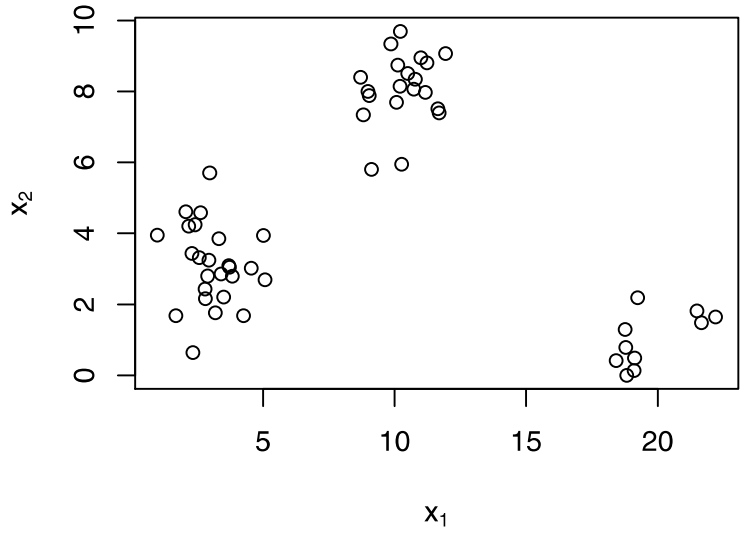

Fig. 6.1. Bivariate data showing the presence of three clusters.

6.3 Agglomerative hierarchical clustering

This class of clustering methods produces a *hierarchical classification* of data. In a hierarchical classification, the data are not partitioned into a particular number of classes or groups at a single step. Instead the classification consists of a series of partitions that may run from a single "cluster" containing all individuals to n clusters, each containing a single individual. Agglomerative hierarchical clustering techniques produce partitions by a series of successive fusions of the n individuals into groups. With such methods, fusions, once made, are irreversible, so that when an agglomerative algorithm has placed two individuals in the same group they cannot subsequently appear in different groups. Since all agglomerative hierarchical techniques ultimately reduce the data to a single cluster containing all the individuals, the investigator seeking the solution with the best-fitting number of clusters will need to decide which division to choose. The problem of deciding on the "correct" number of clusters will be taken up later.

An agglomerative hierarchical clustering procedure produces a series of partitions of the data, $P_n, P_{n-1}, \ldots, P_1$. The first, P_n, consists of n single-member clusters, and the last, P_1, consists of a single group containing all n individuals. The basic operation of all methods is similar:

(START) Clusters $C_1, C_2, \ldots, C_n$ each containing a single individual.
(1) Find the nearest pair of distinct clusters, say C_i and C_j, merge C_i and C_j, delete C_j, and decrease the number of clusters by one.
(2) If the number of clusters equals one, then stop; otherwise return to 1.

But before the process can begin, an inter-individual *distance matrix* or *similarity matrix* needs to be calculated. There are many ways to calculate distances or similarities between pairs of individuals, but here we only deal with a commonly used distance measure, Euclidean distance, which was defined in Chapter 1 but as a reminder is calculated as

$$d_{ij} = \sqrt{\sum_{k=1}^{q}(x_{ik} - x_{jk})^2},$$

where d_{ij} is the Euclidean distance between individual i with variable values $x_{i1}, x_{i2}, \ldots, x_{iq}$ and individual j with variable values $x_{j1}, x_{j2}, \ldots, x_{jq}$. (Details of other possible distance measures and similarity measures are given in Everitt et al. 2011). The Euclidean distances between each pair of individuals can be arranged in a matrix that is symmetric because $d_{ij} = d_{ji}$ and has zeros on the main diagonal. Such a matrix is the starting point of many clustering examples, although the calculation of Euclidean distances from the raw data may not be sensible when the variables are on very different scales. In such cases, the variables can be standardised in the usual way before calculating the distance matrix, although this can be unsatisfactory in some cases (see Everitt et al. 2011, for details).

Given an inter-individual distance matrix, the hierarchical clustering can begin, and at each stage in the process the methods fuse individuals or groups of individuals formed earlier that are closest (or most similar). So as groups are formed, the distance between an individual and a group containing several individuals and the distance between two groups of individuals will need to be calculated. How such distances are defined leads to a variety of different techniques. Two simple inter-group measures are

$$d_{AB} = \min_{\substack{i\in A\\ i\in B}}(d_{ij}),$$

$$d_{AB} = \max_{\substack{i\in A\\ i\in B}}(d_{ij}),$$

where d_{AB} is the distance between two clusters A and B, and d_{ij} is the distance between individuals i and j found from the initial inter-individual distance matrix.

The first inter-group distance measure above is the basis of *single linkage* clustering, the second that of *complete linkage* clustering. Both these techniques have the desirable property that they are invariant under monotone transformations of the original inter-individual distances; i.e., they only depend on the ranking on these distances, not their actual values.

A further possibility for measuring inter-cluster distance or dissimilarity is

$$d_{AB} = \frac{1}{n_A n_B}\sum_{i\in A}\sum_{i\in B} d_{ij},$$

where n_A and n_B are the numbers of individuals in clusters A and B. This measure is the basis of a commonly used procedure known as *group average* clustering. All three inter-group measures described above are illustrated in Figure 6.2.

Hierarchical classifications may be represented by a two-dimensional diagram known as a *dendrogram*, which illustrates the fusions made at each stage of the analysis. An example of such a diagram is given in Figure 6.3. The structure of Figure 6.3 resembles an *evolutionary tree*, a concept introduced by Darwin under the term "Tree of Life" in his book *On the Origin of Species by Natural Selection* in 1859, and it is in biological applications that hierarchical classifications are most relevant and most justified (although this type of clustering has also been used in many other areas).

As a first example of the application of the three clustering methods, single linkage, complete linkage, and group average, each will be applied to the chest, waist, and hip measurements of 20 individuals given in Chapter 1, Table 1.2. First Euclidean distances are calculated on the unstandardised measurements using the following R code:

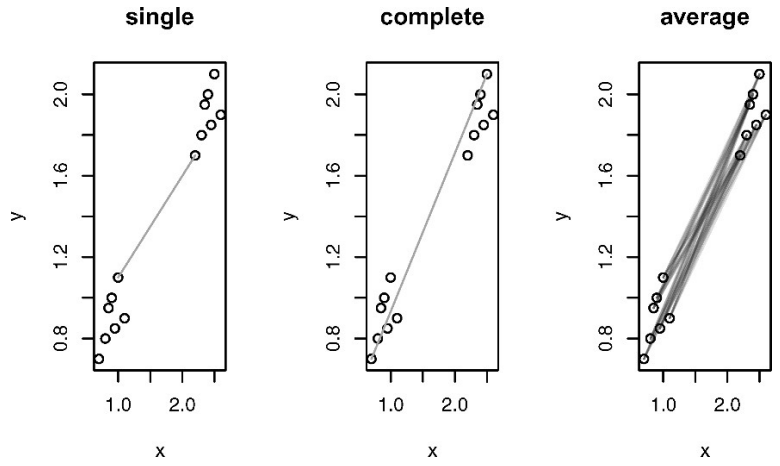

Fig. 6.2. Inter-cluster distance measures.

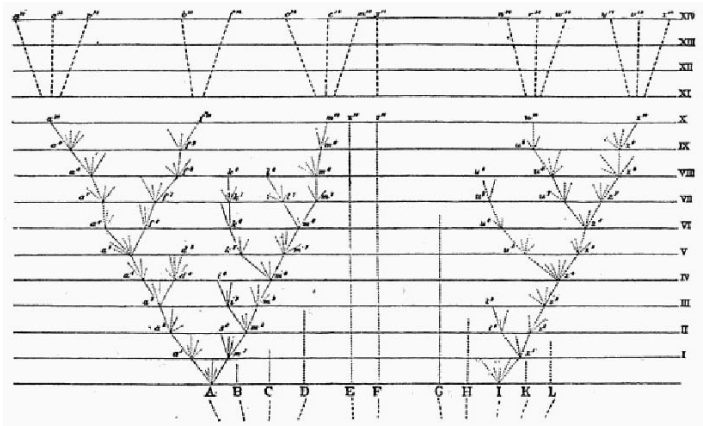

Fig. 6.3. Darwin's Tree of Life.

```
R> (dm <- dist(measure[, c("chest", "waist", "hips")]))
```

```
       1     2     3     4     5     6     7     8     9    10
2   6.16
3   5.66  2.45
4   7.87  2.45  4.69
5   4.24  5.10  3.16  7.48
6  11.00  6.08  5.74  7.14  7.68
7  12.04  5.92  7.00  5.00 10.05  5.10
8   8.94  3.74  4.00  3.74  7.07  5.74  4.12
9   7.81  3.61  2.24  5.39  4.58  3.74  5.83  3.61
10 10.10  4.47  4.69  5.10  7.35  2.24  3.32  3.74  3.00
11  7.00  8.31  6.40  9.85  5.74 11.05 12.08  8.06  7.48 10.25
12  7.35  7.07  5.48  8.25  6.00  9.95 10.25  6.16  6.40  8.83
13  7.81  8.54  7.28  9.43  7.55 12.08 11.92  7.81  8.49 10.82
14  8.31 11.18  9.64 12.45  8.66 14.70 15.30 11.18 11.05 13.75
15  7.48  6.16  4.90  7.07  6.16  9.22  9.00  4.90  5.74  7.87
16  7.07  6.00  4.24  7.35  5.10  8.54  9.11  5.10  5.00  7.48
17  7.81  7.68  6.71  8.31  7.55 11.40 10.77  6.71  7.87  9.95
18  6.71  6.08  4.58  7.28  5.39  9.27  9.49  5.39  5.66  8.06
19  9.17  5.10  4.47  5.48  7.07  6.71  5.74  2.00  4.12  5.10
20  7.68  9.43  7.68 10.82  7.00 12.41 13.19  9.11  8.83 11.53
      11    12    13    14    15    16    17    18    19
2
3
4
5
6
7
8
9
10
11
12  2.24
13  2.83  2.24
14  3.74  5.20  3.74
15  3.61  1.41  3.00  6.40
16  3.00  1.41  3.61  6.40  1.41
17  3.74  2.24  1.41  5.10  2.24  3.32
18  2.83  1.00  2.83  5.83  1.00  1.00  2.45
19  6.71  4.69  6.40  9.85  3.46  3.74  5.39  4.12
20  1.41  3.00  2.45  2.45  4.36  4.12  3.74  3.74  7.68
```

Application of each of the three clustering methods described earlier to the distance matrix and a plot of the corresponding dendrogram are achieved using the `hclust()` function:


```
R> plot(cs <- hclust(dm, method = "single"))
R> plot(cc <- hclust(dm, method = "complete"))
R> plot(ca <- hclust(dm, method = "average"))
```

The resulting plots (for single, complete, and average linkage) are given in the upper part of Figure 6.4.

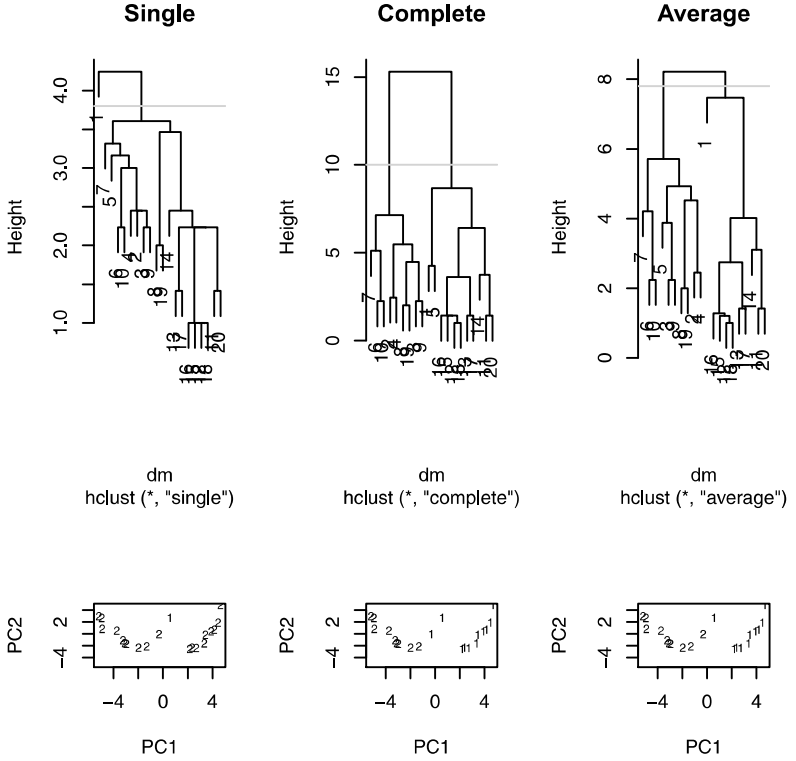

Fig. 6.4. Cluster solutions for measure data. The top row gives the cluster dendrograms along with the cutoff used to derive the classes presented (in the space of the first two principal components) in the bottom row.

We now need to consider how we select specific partitions of the data (i.e., a solution with a particular number of groups) from these dendrograms. The answer is that we "cut" the dendrogram at some height and this will give a partition with a particular number of groups. How do we choose where to cut or, in other words, how do we decide on a particular number of groups that is, in some sense, optimal for the data? This is a more difficult question to answer.

One informal approach is to examine the sizes of the changes in height in the dendrogram and take a "large" change to indicate the appropriate number of clusters for the data. (More formal approaches are described in Everitt et al. 2011) Even using this informal approach on the dendrograms in Figure 6.4, it is not easy to decide where to "cut".

So instead, because we know that these data consist of measurements on ten men and ten women, we will look at the two-group solutions from each method that are obtained by cutting the dendrograms at suitable heights. We can display and compare the three solutions graphically by plotting the first two principal component scores of the data, labelling the points to identify the cluster solution of one of the methods by using the following code:

```
R> body_pc <- princomp(dm, cor = TRUE)
R> xlim <- range(body_pc$scores[,1])
R> plot(body_pc$scores[,1:2], type = "n",
+       xlim = xlim, ylim = xlim)
R> lab <- cutree(cs, h = 3.8)
R> text(body_pc$scores[,1:2], labels = lab, cex = 0.6)
```

The resulting plots are shown in the lower part of Figure 6.4. The plots of dendrograms and principal components scatterplots are combined into a single diagram using the `layout()` function (see the chapter demo for the complete R code). The plot associated with the single linkage solution immediately demonstrates one of the problems with using this method in practise, and that is a phenomenon known as *chaining*, which refers to the tendency to incorporate intermediate points between clusters into an existing cluster rather than initiating a new one. As a result, single linkage solutions often contain long "straggly" clusters that do not give a useful description of the data. The two-group solutions from complete linkage and average linkage, also shown in Figure 6.4, are similar and in essence place the men (observations 1 to 10) together in one cluster and the women (observations 11 to 20) in the other.

6.3.1 Clustering jet fighters

The data shown in Table 6.1 as originally given in Stanley and Miller (1979) and also in Hand et al. (1994) are the values of six variables for 22 US fighter aircraft. The variables are as follows:

`FFD`: first flight date, in months after January 1940;
`SPR`: specific power, proportional to power per unit weight;
`RGF`: flight range factor;
`PLF`: payload as a fraction of gross weight of aircraft;
`SLF`: sustained load factor;
`CAR`: a binary variable that takes the value 1 if the aircraft can land on a carrier and 0 otherwise.

Table 6.1: `jet` data. Jet fighters data.

FFD	SPR	RGF	PLF	SLF	CAR
82	1.468	3.30	0.166	0.10	no
89	1.605	3.64	0.154	0.10	no
101	2.168	4.87	0.177	2.90	yes
107	2.054	4.72	0.275	1.10	no
115	2.467	4.11	0.298	1.00	yes
122	1.294	3.75	0.150	0.90	no
127	2.183	3.97	0.000	2.40	yes
137	2.426	4.65	0.117	1.80	no
147	2.607	3.84	0.155	2.30	no
166	4.567	4.92	0.138	3.20	yes
174	4.588	3.82	0.249	3.50	no
175	3.618	4.32	0.143	2.80	no
177	5.855	4.53	0.172	2.50	yes
184	2.898	4.48	0.178	3.00	no
187	3.880	5.39	0.101	3.00	yes
189	0.455	4.99	0.008	2.64	no
194	8.088	4.50	0.251	2.70	yes
197	6.502	5.20	0.366	2.90	yes
201	6.081	5.65	0.106	2.90	yes
204	7.105	5.40	0.089	3.20	yes
255	8.548	4.20	0.222	2.90	no
328	6.321	6.45	0.187	2.00	yes

We shall apply complete linkage to the data but using only variables two to five. And given that the variables are on very different scales, we will standardise them to unit variance before clustering. The required R code for standardisation and clustering is as follows:

```
R> X <- scale(jet[, c("SPR", "RGF", "PLF", "SLF")],
+             center = FALSE, scale = TRUE)
R> dj <- dist(X)
R> plot(cc <- hclust(dj), main = "Jets clustering")
R> cc

Call:
hclust(d = dj)

Cluster method   : complete
Distance         : euclidean
Number of objects: 22
```

The resulting dendrogram in Figure 6.5 strongly suggests the presence of two groups of fighters. In Figure 6.6, the data are plotted in the space of the first two principal components of the correlation matrix of the relevant variables (SPR to SLF). And in Figure 6.6 the points are labelled by cluster number for the two-group solution and the colours used are the values of the CAR variable. The two-group solution largely corresponds to planes that can and cannot land on a carrier.

```
Call:
hclust(d = dj)

Cluster method   : complete
Distance         : euclidean
Number of objects: 22
```

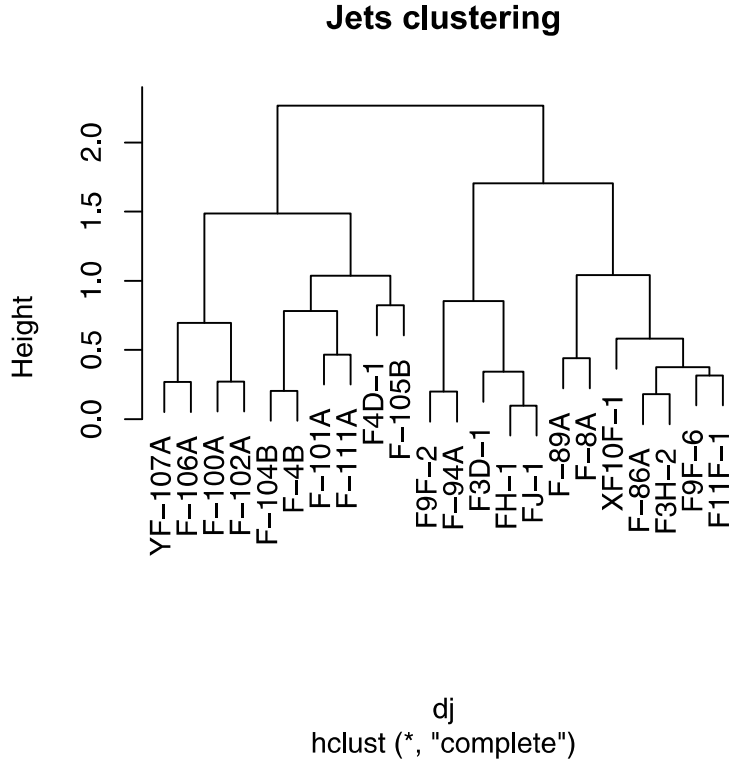

Fig. 6.5. Hierarchical clustering (complete linkage) of jet data.

Again, we cut the dendrogram in such a way that two clusters remain and plot the corresponding classes in the space of the first two principal components; see Figure 6.6.

```
R> pr <- prcomp(dj)$x[, 1:2]
R> plot(pr, pch = (1:2)[cutree(cc, k = 2)],
+       col = c("black", "darkgrey")[jet$CAR],
+       xlim = range(pr) * c(1, 1.5))
R> legend("topright", col = c("black", "black",
+                             "darkgrey", "darkgrey"),
+         legend = c("1 / no", "2 / no", "1 / yes", "2 / yes"),
+         pch = c(1:2, 1:2), title = "Cluster / CAR", bty = "n")
```

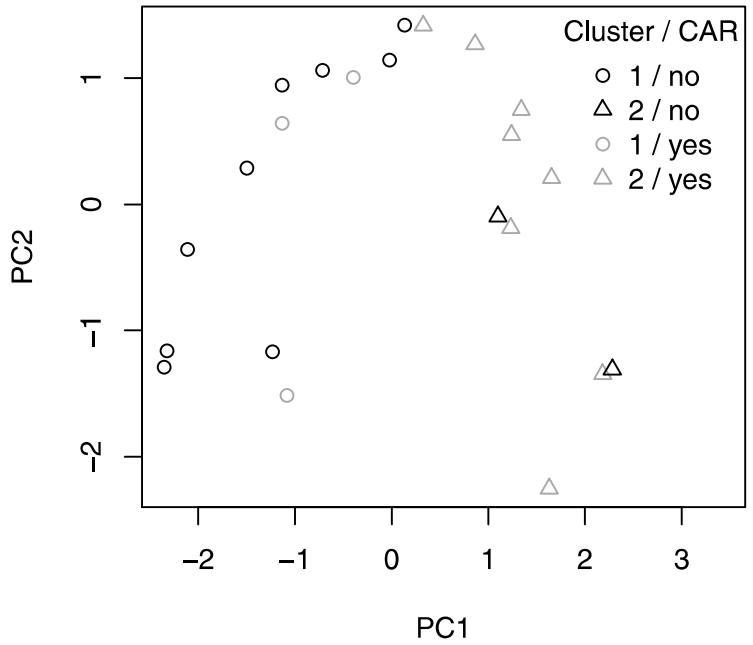

Fig. 6.6. Hierarchical clustering (complete linkage) of jet data plotted in PCA space.

6.4 K-means clustering

The k-means clustering technique seeks to partition the n individuals in a set of multivariate data into k groups or clusters, $(G_1, G_2, \ldots, G_k)$, where G_i denotes the set of n_i individuals in the ith group, and k is given (or a possible range is specified by the researcher–the problem of choosing the "true" value of k will be taken up later) by minimising some numerical criterion, low values of which are considered indicative of a "good" solution. The most commonly used implementation of k-means clustering is one that tries to find the partition of the n individuals into k groups that minimises the *within-group sum of squares* (WGSS) over all variables; explicitly, this criterion is

$$\text{WGSS} = \sum_{j=1}^{q} \sum_{l=1}^{k} \sum_{i \in G_l} (x_{ij} - \overline{x}_j^{(l)})^2,$$

where $\overline{x}_j^{(l)} = \frac{1}{n_i} \sum_{i \in G_l} x_{ij}$ is the mean of the individuals in group G_l on variable j.

The problem then appears relatively simple; namely, consider every possible partition of the n individuals into k groups, and select the one with the lowest within-group sum of squares. Unfortunately, the problem in practise is not so straightforward. The numbers involved are so vast that complete enumeration of *every* possible partition remains impossible even with the fastest computer. The scale of the problem immediately becomes clear by looking at the numbers in Table 6.2.

Table 6.2: Number of possible partitions depending on the sample size n and number of clusters k.

n	k	Number of possible partitions
15	3	2,375,101
20	4	45,232,115,901
25	8	690,223,721,118,368,580
100	5	10^{68}

The impracticability of examining every possible partition has led to the development of algorithms designed to search for the minimum values of the clustering criterion by rearranging existing partitions and keeping the new one only if it provides an improvement. Such algorithms do not, of course, guarantee finding the global minimum of the criterion. The essential steps in these algorithms are as follows:

1. Find some initial partition of the individuals into the required number of groups. (Such an initial partition could be provided by a solution from one of the hierarchical clustering techniques described in the previous section.)
2. Calculate the change in the clustering criterion produced by "moving" each individual from its own cluster to another cluster.
3. Make the change that leads to the greatest improvement in the value of the clustering criterion.
4. Repeat steps (2) and (3) until no move of an individual causes the clustering criterion to improve.

(For a more detailed account of the typical k-means algorithm see Steinley 2008)

The k-means approach to clustering using the minimisation of the within-group sum of squares over all the variables is widely used but suffers from the two problems of (1) not being scale-invariant (i.e., different solutions may result from clustering the raw data and the data standardised in some way) and (2) of imposing a "spherical" structure on the data; i.e., it will find clusters shaped like hyper-footballs even if the "true" clusters in the data are of some other shape (see Everitt et al. 2011, for some examples of the latter phenomenon). Nevertheless, the k-means method remains very popular. With k-means clustering, the investigator can choose to partition the data into a specified number of groups. In practice, solutions for a range of values for the number of groups are found and in some way the optimal or "true" number of groups for the data must be chosen. Several suggestions have been made as to how to answer the number of groups question, but none is completely satisfactory. The method we shall use in the forthcoming example is to plot the within-groups sum of squares associated with the k-means solution for each number of groups. As the number of groups increases the sum of squares will necessarily decrease, but an obvious "elbow" in the plot may be indicative of the most useful solution for the investigator to look at in detail. (Compare this with the scree plot described in Chapter 3.)

6.4.1 Clustering the states of the USA on the basis of their crime rate profiles

The *Statistical Abstract of the USA* (Anonymous 1988, Table 265) gives rates of different types of crime per 100,000 residents of the 50 states of the USA plus the District of Columbia for the year 1986. The data are given in Table 6.3.

Table 6.3: `crime` data. Crime data.

	`Murder`	`Rape`	`Robbery`	`Assault`	`Burglary`	`Theft`	`Vehicle`
ME	2.0	14.8	28	102	803	2347	164
NH	2.2	21.5	24	92	755	2208	228
VT	2.0	21.8	22	103	949	2697	181
MA	3.6	29.7	193	331	1071	2189	906

Table 6.3: `crime` data (continued).

	Murder	Rape	Robbery	Assault	Burglary	Theft	Vehicle
RI	3.5	21.4	119	192	1294	2568	705
CT	4.6	23.8	192	205	1198	2758	447
NY	10.7	30.5	514	431	1221	2924	637
NJ	5.2	33.2	269	265	1071	2822	776
PA	5.5	25.1	152	176	735	1654	354
OH	5.5	38.6	142	235	988	2574	376
IN	6.0	25.9	90	186	887	2333	328
IL	8.9	32.4	325	434	1180	2938	628
MI	11.3	67.4	301	424	1509	3378	800
WI	3.1	20.1	73	162	783	2802	254
MN	2.5	31.8	102	148	1004	2785	288
IA	1.8	12.5	42	179	956	2801	158
MO	9.2	29.2	170	370	1136	2500	439
ND	1.0	11.6	7	32	385	2049	120
SD	4.0	17.7	16	87	554	1939	99
NE	3.1	24.6	51	184	748	2677	168
KS	4.4	32.9	80	252	1188	3008	258
DE	4.9	56.9	124	241	1042	3090	272
MD	9.0	43.6	304	476	1296	2978	545
DC	31.0	52.4	754	668	1728	4131	975
VA	7.1	26.5	106	167	813	2522	219
WV	5.9	18.9	41	99	625	1358	169
NC	8.1	26.4	88	354	1225	2423	208
SC	8.6	41.3	99	525	1340	2846	277
GA	11.2	43.9	214	319	1453	2984	430
FL	11.7	52.7	367	605	2221	4373	598
KY	6.7	23.1	83	222	824	1740	193
TN	10.4	47.0	208	274	1325	2126	544
AL	10.1	28.4	112	408	1159	2304	267
MS	11.2	25.8	65	172	1076	1845	150
AR	8.1	28.9	80	278	1030	2305	195
LA	12.8	40.1	224	482	1461	3417	442
OK	8.1	36.4	107	285	1787	3142	649
TX	13.5	51.6	240	354	2049	3987	714
MT	2.9	17.3	20	118	783	3314	215
ID	3.2	20.0	21	178	1003	2800	181
WY	5.3	21.9	22	243	817	3078	169
CO	7.0	42.3	145	329	1792	4231	486
NM	11.5	46.9	130	538	1845	3712	343
AZ	9.3	43.0	169	437	1908	4337	419
UT	3.2	25.3	59	180	915	4074	223
NV	12.6	64.9	287	354	1604	3489	478

Table 6.3: `crime` data (continued).

	Murder	Rape	Robbery	Assault	Burglary	Theft	Vehicle
WA	5.0	53.4	135	244	1861	4267	315
OR	6.6	51.1	206	286	1967	4163	402
CA	11.3	44.9	343	521	1696	3384	762
AK	8.6	72.7	88	401	1162	3910	604
HI	4.8	31.0	106	103	1339	3759	328

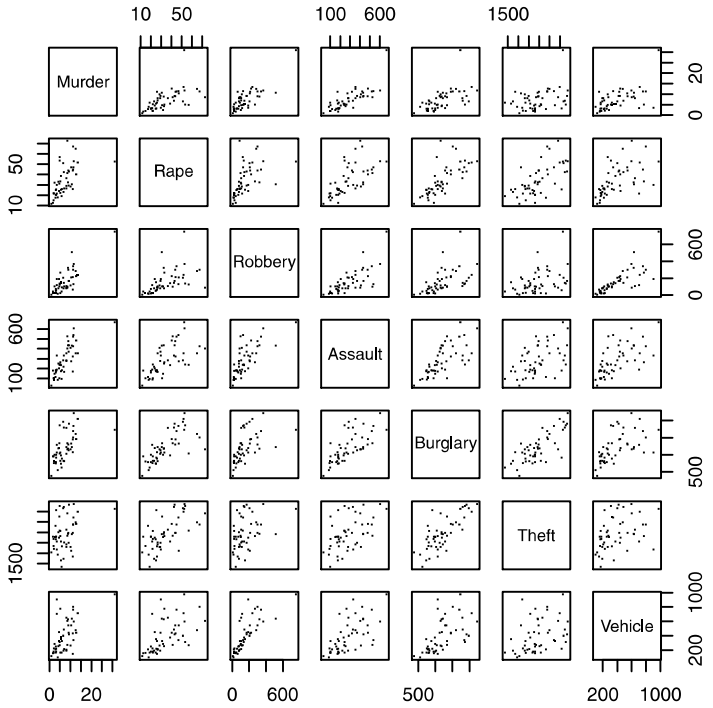

Fig. 6.7. Scatterplot matrix of crime data.

To begin, let's look at the scatterplot matrix of the data shown in Figure 6.7. The plot suggests that at least one of the cities is considerably different from the others in its murder rate at least. The city is easily identified using

```
R> subset(crime, Murder > 15)

   Murder Rape Robbery Assault Burglary Theft Vehicle
DC     31 52.4     754     668     1728  4131     975
```

i.e., the murder rate is very high in the District of Columbia. In order to check if the other crime rates are also higher in DC, we label the corresponding points in the scatterplot matrix in Figure 6.8. Clearly, DC is rather extreme in most crimes (the clear message is don't live in DC).

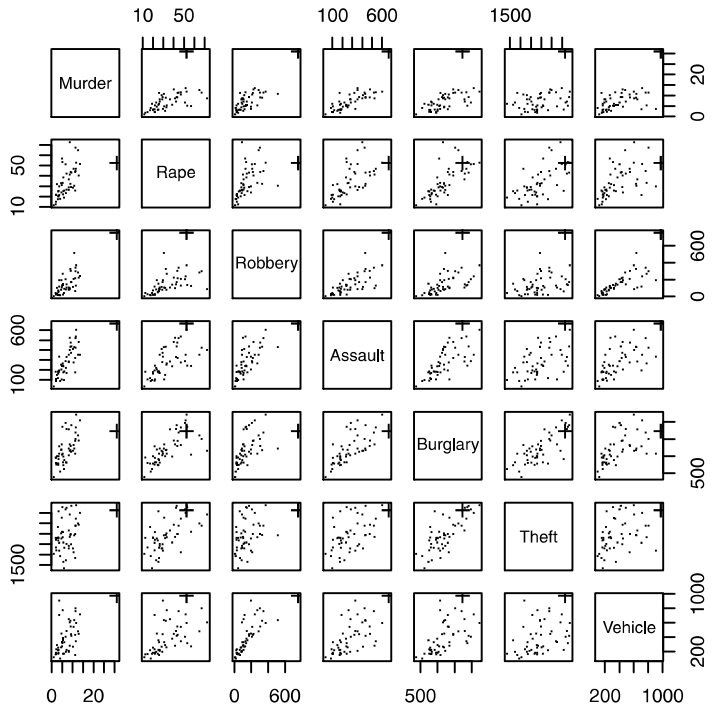

Fig. 6.8. Scatterplot matrix of crime data with DC observation labelled using a plus sign.

We will now apply k-means clustering to the crime rate data *after* removing the outlier, DC. If we first calculate the variances of the crime rates for the different types of crimes we find the following:

```
R> sapply(crime, var)
```



```
  Murder     Rape  Robbery  Assault Burglary    Theft  Vehicle
    23.2    212.3  18993.4  22004.3 177912.8 582812.8  50007.4
```

The variances are very different, and using k-means on the raw data would not be sensible; we must standardise the data in some way, and here we standardise each variable by its range. After such standardisation, the variances become

```
R> rge <- sapply(crime, function(x) diff(range(x)))
R> crime_s <- sweep(crime, 2, rge, FUN = "/")
R> sapply(crime_s, var)

  Murder     Rape  Robbery  Assault Burglary    Theft  Vehicle
 0.02578  0.05687  0.03404  0.05440  0.05278  0.06411  0.06517
```

The variances of the standardised data are very similar, and we can now progress with clustering the data. First we plot the within-groups sum of squares for one- to six-group solutions to see if we can get any indication of the number of groups. The plot is shown in Figure 6.9. The only "elbow" in the plot occurs for two groups, and so we will now look at the two-group solution. The group means for two groups are computed by

```
R> kmeans(crime_s, centers = 2)$centers * rge

  Murder  Rape Robbery Assault Burglary  Theft Vehicle
1  4.893 305.1   189.6  259.70     31.0  540.5   873.0
2 21.098 483.3  1031.4   19.26    638.9 2096.1   578.6
```

A plot of the two-group solution in the space of the first two principal components of the correlation matrix of the data is shown in Figure 6.10. The two groups are created essentially on the basis of the first principal component score, which is a weighted average of the crime rates. Perhaps all the cluster analysis is doing here is dividing into two parts a homogenous set of data. This is always a possibility, as is discussed in some detail in Everitt et al. (2011).

6.4.2 Clustering Romano-British pottery

The second application of k-means clustering will be to the data on Romano-British pottery given in Chapter 1. We begin by computing the Euclidean distance matrix for the standardised measurements of the 45 pots. The resulting 45×45 matrix can be inspected graphically by using an *image plot*, here obtained with the function `levelplot` available in the package **lattice** (Sarkar 2010, 2008). Such a plot associates each cell of the dissimilarity matrix with a colour or a grey value. We choose a very dark grey for cells with distance zero (i.e., the diagonal elements of the dissimilarity matrix) and pale values for cells with greater Euclidean distance. Figure 6.11 leads to the impression that there are at least three distinct groups with small inter-cluster differences (the dark rectangles), whereas much larger distances can be observed for all other cells.

```
R> n <- nrow(crime_s)
R> wss <- rep(0, 6)
R> wss[1] <- (n - 1) * sum(sapply(crime_s, var))
R> for (i in 2:6)
+      wss[i] <- sum(kmeans(crime_s,
+                           centers = i)$withinss)
R> plot(1:6, wss, type = "b", xlab = "Number of groups",
+       ylab = "Within groups sum of squares")
```

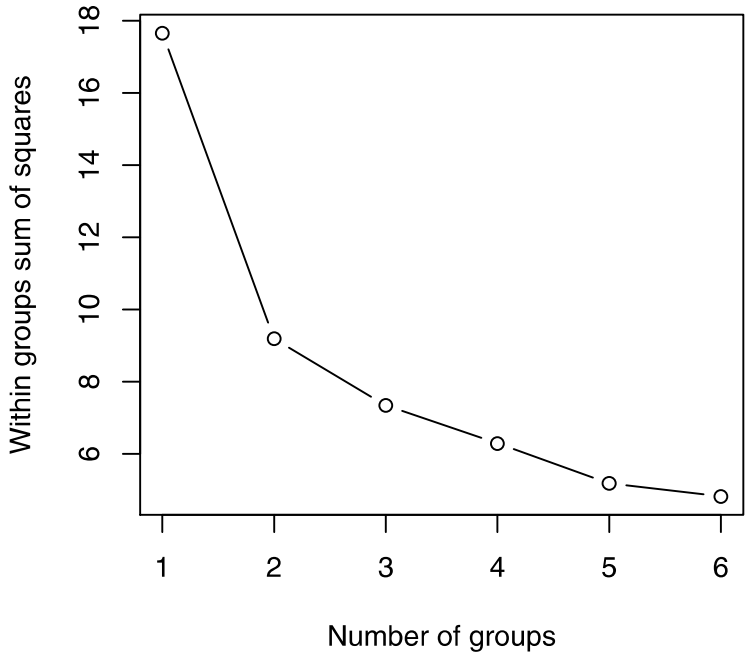

Fig. 6.9. Plot of within-groups sum of squares against number of clusters.

We plot the within-groups sum of squares for one to six group k-means solutions to see if we can get any indication of the number of groups (see Figure 6.12). Again, the plot leads to the relatively clear conclusion that the data contain three clusters.

Our interest is now in a comparison of the kiln sites at which the pottery was found.

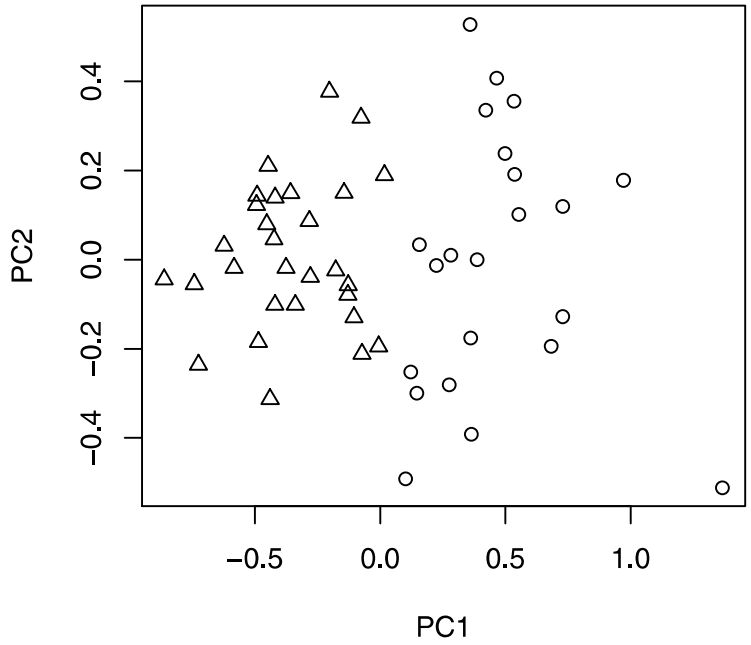

Fig. 6.10. Plot of k-means two-group solution for the standardised crime rate data.

```
R> set.seed(29)
R> pottery_cluster <- kmeans(pots, centers = 3)$cluster
R> xtabs(~ pottery_cluster + kiln, data = pottery)
```

```
               kiln
pottery_cluster  1  2  3  4  5
              1 21  0  0  0  0
              2  0 12  2  0  0
              3  0  0  0  5  5
```

The contingency table shows that cluster 1 contains all pots found at kiln site number one, cluster 2 contains all pots from kiln sites numbers two and three, and cluster three collects the ten pots from kiln sites four and five. In fact, the five kiln sites are from three different regions: region 1 contains just kiln one, region 2 contains kilns two and three, and region 3 contains kilns four

```
R> pottery_dist <- dist(pots <- scale(pottery[, colnames(pottery) != "kiln"],
+                                     center = FALSE))
R> library("lattice")
R> levelplot(as.matrix(pottery_dist), xlab = "Pot Number",
+            ylab = "Pot Number")
```

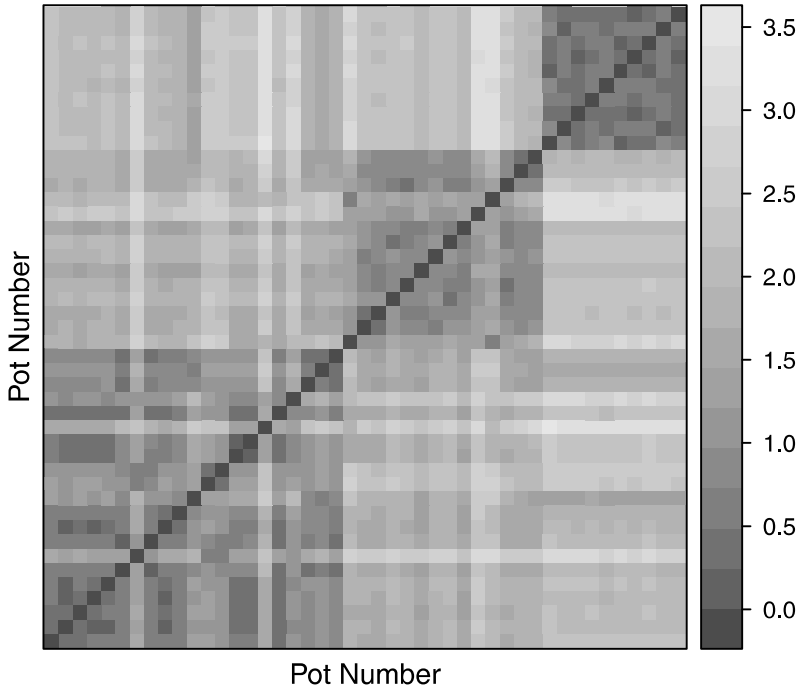

Fig. 6.11. Image plot of the dissimilarity matrix of the `pottery` data.

and five. So the clusters found actually correspond to pots from three different regions.

6.5 Model-based clustering

The agglomerative hierarchical and k-means clustering methods described in the previous two sections are based largely on heuristic but intuitively reasonable procedures. But they are not based on formal models for cluster structure in the data, making problems such as deciding between methods, estimating


```
R> n <- nrow(pots)
R> wss <- rep(0, 6)
R> wss[1] <- (n - 1) * sum(sapply(pots, var))
R> for (i in 2:6)
+      wss[i] <- sum(kmeans(pots,
+                           centers = i)$withinss)
R> plot(1:6, wss, type = "b", xlab = "Number of groups",
+       ylab = "Within groups sum of squares")
```

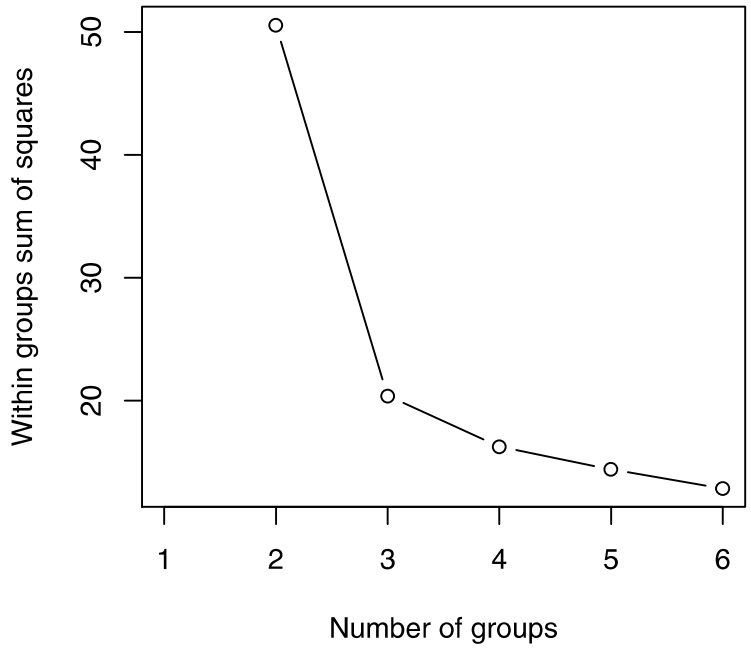

Fig. 6.12. Plot of within-groups sum of squares against number of clusters.

the number of clusters, etc, particularly difficult. And, of course, without a reasonable model, formal inference is precluded. In practise, these may not be insurmountable objections to the use of either the agglomerative methods or k-means clustering because cluster analysis is most often used as an "exploratory" tool for data analysis. But if an acceptable model for cluster structure could be found, then the cluster analysis based on the model might give more persuasive solutions (more persuasive to statisticians at least). In

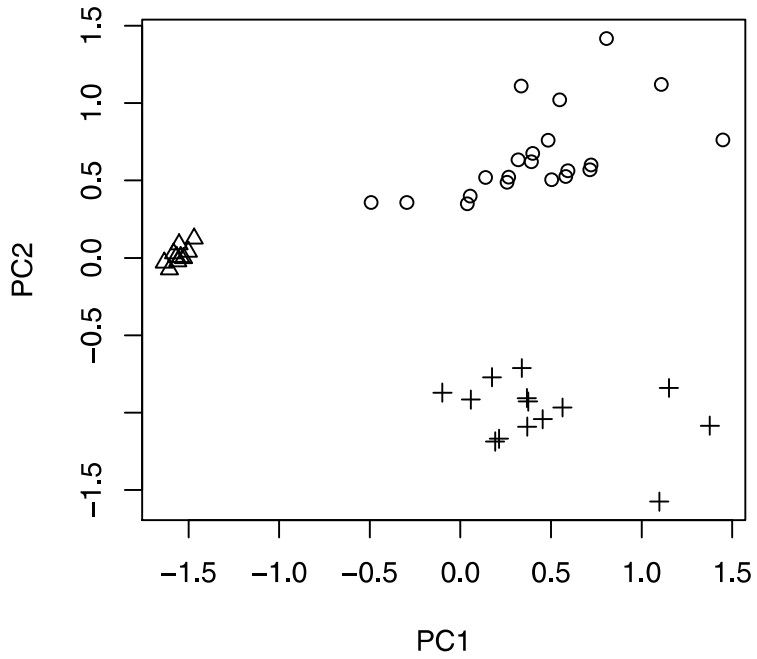

Fig. 6.13. Plot of the k-means three-group solution for the pottery data displayed in the space of the first two principal components of the correlation matrix of the data.

this section, we describe an approach to clustering that postulates a formal statistical model for the population from which the data are sampled, a model that assumes that this population consists of a number of subpopulations (the "clusters"), each having variables with a different multivariate probability density function, resulting in what is known as a *finite mixture density* for the population as a whole. By using finite mixture densities as models for cluster analysis, the clustering problem becomes that of estimating the parameters of the assumed mixture and then using the estimated parameters to calculate the posterior probabilities of cluster membership. And determining the number of clusters reduces to a model selection problem for which objective procedures exist.

Finite mixture densities often provide a sensible statistical *model* for the clustering process, and cluster analyses based on finite mixture models are also

known as *model-based* clustering methods; see Banfield and Raftery (1993). Finite mixture models have been increasingly used in recent years to cluster data in a variety of disciplines, including behavioural, medical, genetic, computer, environmental sciences, and robotics and engineering; see, for example, Everitt and Bullmore (1999), Bouguila and Amayri (2009), Branchaud, Cham, Nenadic, Andersen, and Burdick (2010), Dai, Erkkila, Yli-Harja, and Lahdesmaki (2009), Dunson (2009), Ganesalingam, Stahl, Wijesekera, Galtrey, Shaw, Leigh, and Al-Chalabi (2009), Marin, Mengersen, and Roberts (2005), Meghani, Lee, Hanlon, and Bruner (2009), Pledger and Phillpot (2008), and van Hattum and Hoijtink (2009).

Finite mixture modelling can be seen as a form of *latent variable analysis* (see, for example, Skrondal and Rabe-Hesketh 2004), with "subpopulation" being a latent categorical variable and the latent classes being described by the different components of the mixture density; consequently, cluster analysis based on such models is also often referred to as *latent class cluster analysis*.

6.5.1 Finite mixture densities

Finite mixture densities are described in detail in Everitt and Hand (1981), Titterington, Smith, and Makov (1985), McLachlan and Basford (1988), McLachlan and Peel (2000), and Frühwirth-Schnatter (2006); they are a family of probability density functions of the form

$$f(\mathbf{x};\mathbf{p},\boldsymbol{\theta}) = \sum_{j=1}^{c} p_j g_j(\mathbf{x};\boldsymbol{\theta}_j), \tag{6.1}$$

where $\mathbf{x}$ is a p-dimensional random variable, $\mathbf{p}^\top = (p_1, p_2, \ldots, p_{c-1})$, and $\boldsymbol{\theta}^\top = (\boldsymbol{\theta}_1^\top, \boldsymbol{\theta}_2^\top, \ldots, \boldsymbol{\theta}_c^\top)$, with the p_j being known as mixing proportions and the g_j, $j = 1, \ldots, c$, being the component densities, with density g_j being parameterised by $\boldsymbol{\theta}_j$. The mixing proportions are non-negative and are such that $\sum_{j=1}^{c} p_j = 1$. The number of components forming the mixture (i.e., the postulated number of clusters) is c.

Finite mixtures provide suitable models for cluster analysis if we assume that each group of observations in a data set suspected to contain clusters comes from a population with a different probability distribution. The latter may belong to the same family but differ in the values they have for the parameters of the distribution; it is such an example that we consider in the next section, where the components of the mixture are multivariate normal with different mean vectors and possibly different covariance matrices.

Having estimated the parameters of the assumed mixture density, observations can be associated with particular clusters on the basis of the maximum value of the estimated posterior probability

$$\hat{\mathsf{P}}(\text{cluster } j|\mathbf{x}_i) = \frac{\hat{p}_j g_j(\mathbf{x}_i;\hat{\boldsymbol{\theta}}_j)}{f(\mathbf{x}_i;\hat{\mathbf{p}},\hat{\boldsymbol{\theta}})}, j = 1, \ldots, c. \tag{6.2}$$

6.5.2 Maximum likelihood estimation in a finite mixture density with multivariate normal components

Given a sample of observations $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, from the mixture density given in Equation (6.1) the log-likelihood function, l, is

$$l(\mathbf{p}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \ln f(\mathbf{x}_i; \mathbf{p}, \boldsymbol{\theta}). \tag{6.3}$$

Estimates of the parameters in the density would usually be obtained as a solution of the likelihood equations

$$\frac{\partial l(\boldsymbol{\varphi})}{\partial(\boldsymbol{\varphi})} = 0, \tag{6.4}$$

where $\boldsymbol{\varphi}^\top = (\mathbf{p}^\top, \boldsymbol{\theta}^\top)$. In the case of finite mixture densities, the likelihood function is too complicated to employ the usual methods for its maximisation; for example, an iterative Newton–Raphson method that approximates the gradient vector of the log-likelihood function $l(\boldsymbol{\varphi})$ by a linear Taylor series expansion (see Everitt (1984)).

Consequently, the required maximum likelihood estimates of the parameters in a finite mixture model have to be computed in some other way. In the case of a mixture in which the jth component density is multivariate normal with mean vector $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$, it can be shown (see Everitt and Hand 1981, for details) that the application of maximum likelihood results in the series of equations

$$\hat{p}_j = \frac{1}{n} \sum_{i=1}^{n} \hat{\mathsf{P}}(j|\mathbf{x}_i), \tag{6.5}$$

$$\hat{\boldsymbol{\mu}}_j = \frac{1}{n\hat{p}_j} \sum_{i=1}^{n} \mathbf{x}_i \hat{\mathsf{P}}(j|\mathbf{x}_i), \tag{6.6}$$

$$\hat{\boldsymbol{\Sigma}}_j = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)^\top \hat{\mathsf{P}}(j|\mathbf{x}_i), \tag{6.7}$$

where the $\hat{\mathsf{P}}(j|\mathbf{x}_i)$s are the estimated posterior probabilities given in equation (6.2).

Hasselblad (1966, 1969), Wolfe (1970), and Day (1969) all suggest an iterative scheme for solving the likelihood equations given above that involves finding initial estimates of the posterior probabilities given initial estimates of the parameters of the mixture and then evaluating the right-hand sides of Equations 6.5 to 6.7 to give revised values for the parameters. From these, new estimates of the posterior probabilities are derived, and the procedure is repeated until some suitable convergence criterion is satisfied. There are potential problems with this process unless the component covariance matrices

are constrained in some way; for example, it they are all assumed to be the same–again see Everitt and Hand (1981) for details.

This procedure is a particular example of the iterative *expectation maximisation* (EM) algorithm described by Dempster, Laird, and Rubin (1977) in the context of likelihood estimation for incomplete data problems. In estimating parameters in a mixture, it is the "labels" of the component density from which an observation arises that are missing. As an alternative to the EM algorithm, Bayesian estimation methods using the Gibbs sampler or other Monte Carlo Markov Chain (MCMC) methods are becoming increasingly popular–see Marin et al. (2005) and McLachlan and Peel (2000).

Fraley and Raftery (2002, 2007) developed a series of finite mixture density models with multivariate normal component densities in which they allow some, but not all, of the features of the covariance matrix (*orientation*, *size*, and *shape*–discussed later) to vary between clusters while constraining others to be the same. These new criteria arise from considering the reparameterisation of the covariance matrix $\boldsymbol{\Sigma}_j$ in terms of its eigenvalue description

$$\boldsymbol{\Sigma}_j = \mathbf{D}_j \boldsymbol{\Lambda}_j \mathbf{D}_j^\top, \tag{6.8}$$

where $\mathbf{D}_j$ is the matrix of eigenvectors and $\boldsymbol{\Lambda}_j$ is a diagonal matrix with the eigenvalues of $\boldsymbol{\Sigma}_j$ on the diagonal (this is simply the usual principal components transformation–see Chapter 3). The orientation of the principal components of $\boldsymbol{\Sigma}_j$ is determined by $\mathbf{D}_j$, whilst $\boldsymbol{\Lambda}_j$ specifies the size and shape of the density contours. Specifically, we can write $\boldsymbol{\Lambda}_j = \lambda_j \mathbf{A}_j$, where λ_j is the largest eigenvalue of $\boldsymbol{\Sigma}_j$ and $\mathbf{A}_j = \text{diag}(1, \alpha_2, \ldots, \alpha_p)$ contains the eigenvalue ratios after division by λ_j. Hence λ_j controls the size of the jth cluster and $\mathbf{A}_j$ its shape. (Note that the term "size" here refers to the volume occupied in space, not the number of objects in the cluster.) In two dimensions, the parameters would reflect, for each cluster, the correlation between the two variables, and the magnitudes of their standard deviations. More details are given in Banfield and Raftery (1993) and Celeux and Govaert (1995), but Table 6.4 gives a series of models corresponding to various constraints imposed on the covariance matrix. The models make up what Fraley and Raftery (2003, 2007) term the "MCLUST" family of mixture models. The mixture likelihood approach based on the EM algorithm for parameter estimation is implemented in the `Mclust()` function in the R package **mclust** and fits the models in the MCLUST family described in Table 6.4.

Model selection is a combination of choosing the appropriate clustering model for the population from which the n observations have been taken (i.e., are all clusters spherical, all elliptical, all different shapes or somewhere in between?) *and* the optimal number of clusters. A Bayesian approach is used (see Fraley and Raftery 2002), applying what is known as the *Bayesian Information Criterion* (BIC). The result is a cluster solution that "fits" the observed data as well as possible, and this can include a solution that has only one "cluster" implying that cluster analysis is not really a useful technique for the data.

Table 6.4: **mclust** family of mixture models. Model names describe model restrictions of volume λ_j, shape $\mathbf{A}_j$, and orientation $\mathbf{D}_j$, V= variable, parameter unconstrained, E= equal, parameter constrained, I = matrix constrained to identity matrix.

Abbreviation	Model
EII	spherical, equal volume
VII	spherical, unequal volume
EEI	diagonal, equal volume and shape
VEI	diagonal, varying volume, equal shape
EVI	diagonal, equal volume, varying shape
VVI	diagonal, varying volume and shape
EEE	ellipsoidal, equal volume, shape, and orientation
EEV	ellipsoidal, equal volume and equal shape
VEV	ellipsoidal, equal shape
VVV	ellipsoidal, varying volume, shape, and orientation

To illustrate the use of the finite mixture approach to cluster analysis, we will apply it to data that arise from a study of what gastroenterologists in Europe tell their cancer patients (Thomsen, Wulff, Martin, and Singer 1993). A questionnaire was sent to about 600 gastroenterologists in 27 European countries (the study took place before the recent changes in the political map of the continent) asking what they would tell a patient with newly diagnosed cancer of the colon, and his or her spouse, about the diagnosis. The respondent gastroenterologists were asked to read a brief case history and then to answer six questions with a yes/no answer. The questions were as follows:

Q1: Would you tell this patient that he/she has cancer, if he/she asks no questions?
Q2: Would you tell the wife/husband that the patient has cancer (In the patient's absence)?
Q3: Would you tell the patient that he or she has a cancer, if he or she directly asks you to disclose the diagnosis. (During surgery the surgeon notices several small metastases in the liver.)
Q4: Would you tell the patient about the metastases (supposing the patient asks to be told the results of the operation)?
Q5: Would you tell the patient that the condition is incurable?
Q6: Would you tell the wife or husband that the operation revealed metastases?

The data are shown in a graphical form in Figure 6.14 (we are aware that using finite mixture clustering on this type of data is open to criticism–it may even be a statistical sin–but we hope that even critics will agree it provides an interesting example).

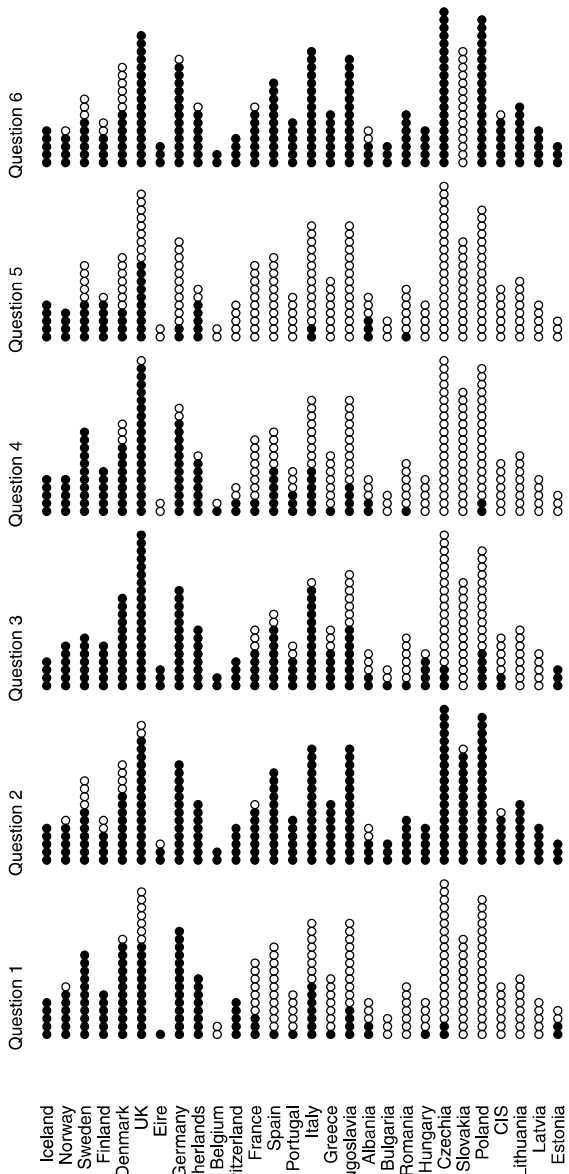

Fig. 6.14. Gastroenterologists questionnaire data. Dark circles indicate a 'yes', open circles a 'no'.

Applying the finite mixture approach to the proportions of 'yes' answers for each question for each country computed from these data using the R code utilizing functionality offered by package **mclust** (Fraley and Raftery 2010)

```
R> library("mclust")

by using mclust, invoked on its own or through another package,
you accept the license agreement in the mclust LICENSE file
and at http://www.stat.washington.edu/mclust/license.txt

R> (mc <- Mclust(thomsonprop))

 best model: ellipsoidal, equal shape with 3 components
```

where `thomsonprob` is the matrix of proportions of "yes" answers to questions Q1–Q6 in the different countries (i.e., the proportion of filled dots in Figure 6.14) available from the **MVA** add-on package. We can first examine the resulting plot of BIC values shown in Figure 6.15. In this diagram, the plot symbols and abbreviations refer to different model assumptions about the shapes of clusters as given in Table 6.4.

The BIC criterion selects model VEV (ellipsoidal, equal shape) and three clusters as the optimal solution. The three-cluster solution is illustrated graphically in Figure 6.16. The first cluster consists of countries in which the large majority of respondents gave "yes" answers to questions $1, 2, 3, 4$, and 6 and about half also gave a "yes" answer to question 5. This cluster includes all the Scandinavian countries the UK, Iceland, Germany, the Netherlands, and Switzerland. In the second cluster, the majority of respondents answer "no" to questions $1, 4$, and 5 and "yes" to questions $2, 3$ and 6; in these countries it appears that the clinicians do not mind giving bad news to the spouses of patients but not to the patients themselves unless they are directly asked by the patient about hispr her condition. This cluster contains Catholic countries such as Spain, Portugal, and Italy. In cluster three, the large majority of respondents answer "no" to questions $1, 3, 4$, and 5 and again a large majority answer "yes" to questions 2 and 6. In these countries, very few clinicians appear to be willing to give the patient bad news even if asked directly by the patient about his or her condition.

6.6 Displaying clustering solutions graphically

Plotting cluster solutions in the space of the first few principal components as illustrated earlier in this chapter is often a useful way to display clustering solutions, but other methods of displaying clustering solutions graphically are also available. Leisch (2010), for example, describes several graphical displays that can be used to visualise cluster analysis solutions. The basis of a number of these graphics is the shadow value, $s(\mathbf{x})$, of each multivariate observation, $\mathbf{x}$, defined as

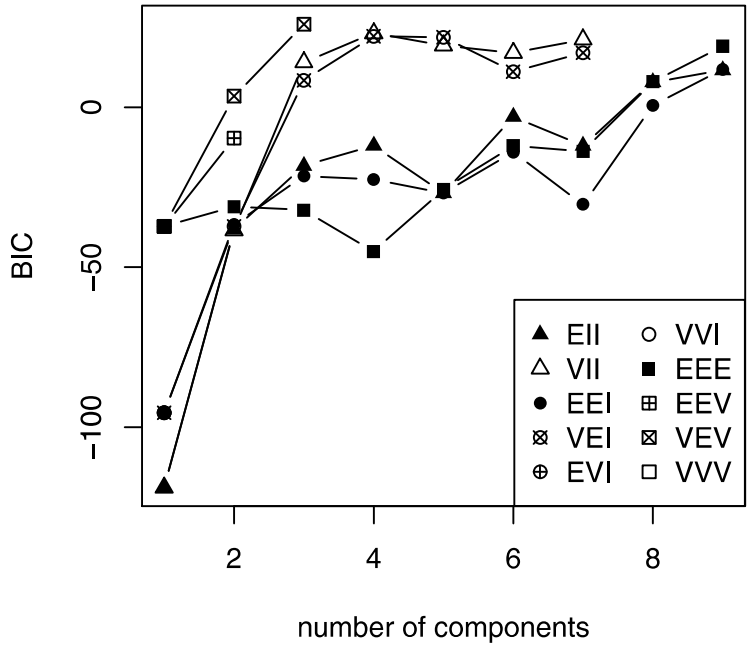

Fig. 6.15. BIC values for gastroenterologists questionnaire.

$$s(\mathbf{x}) = \frac{2d(\mathbf{x}, c(\mathbf{x}))}{d(\mathbf{x}, c(\mathbf{x})) + d(\mathbf{x}, \tilde{c}(\mathbf{x}))},$$

where $d(\mathbf{x}, c(\mathbf{x}))$ is the distance of the observation $\mathbf{x}$ from the centroid of its own cluster and $d(\mathbf{x}, \tilde{c}(\mathbf{x}))$ is the distance of $\mathbf{x}$ from the second closest cluster centroid. If $s(\mathbf{x})$ is close to zero, then the observation is close to its cluster centroid; if $s(\mathbf{x})$ is close to one, then the observation is almost equidistant from the two centroids (a similar approach is used in defining silhouette plots, see Chapter 5). The average shadow value of all observations where cluster i is closest and cluster j is second closest can be used as a simple measure of cluster similarity,

$$s_{ij} = \frac{1}{n_i} \sum_{\mathbf{x} \in A_{ij}} s(\mathbf{x}),$$

where n_i is the number of observations that are closest to the centroid of cluster i and A_{ij} is the set of observations for which the centroid of cluster i is

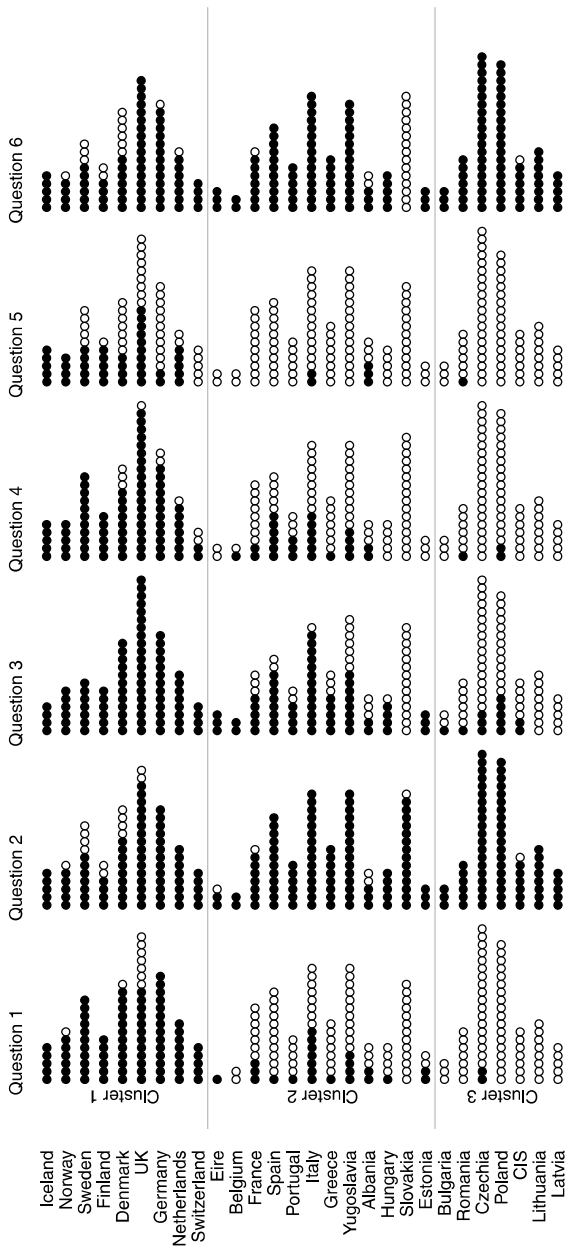

Fig. 6.16. Model-based clustering into three clusters for gastroenterologists questionnaire data. Dark circles indicate a 'yes', open circles a 'no'.

closest and the centroid of cluster j is second closest. The denominator of s_{ij} is taken to be n_i rather than n_{ij}, the number of observations in the set A_{ij}, to prevent inducing large cluster similarity when n_{ij} is small and the set of observations consists of poorly clustered points with large shadow values. For a cluster solution derived from bivariate data, a neighbourhood graph can be constructed using the scatterplot of the two variables, and where two cluster centroids are joined if there exists at least one observation for which these two are closest, and second closest with the thickness of the joining lines being made proportional to the average value of the corresponding s_{ij}. When there are more than two variables in the data set, the neighbourhood graph can be constructed on some suitable projection of the data into two dimensions; for example, the first two principal components of the data could be used. Such plots may help to establish which clusters are "real" and which are not, as we will try to illustrate with two examples.

The first example uses some two-dimensional data generated to contain three clusters. The neighbourhood graph for the k-means five-cluster solution from the application of k-means clustering is shown in Figure 6.17. The thicker lines joining the centroids of clusters 1 and 5 and clusters 2 and 4 strongly suggest that both pairs of clusters overlap to a considerable extent and are probably each divisions of a single cluster.

For the second example we return to the pottery data previously met in the chapter. From the k-means analysis, it is clear that these data contain three clusters; Figure 6.18 shows the neighbourhood plot for the k-means three-cluster solution in the space of the first two principal components of the data. The three clusters are clearly visible in this plot.

A further graphic for displaying clustering solutions is known as a stripes plot. This graphic is a simple but often effective way of visualising the distance of each point from its closest and second closest cluster centroids. For each cluster, $k = 1, \ldots, K$, a stripes plot has a rectangular area that is vertically divided into K smaller rectangles, with each smaller rectangle, i, containing information about distances of the observations in cluster i from the centroid of that cluster along with the corresponding information about observations that have cluster i as their second closest cluster. The explanation of how the plot is constructed becomes more transparent if we look at an actual example. Figure 6.19 shows a stripes plot produced with that package **flexclust** (Leisch and Dimitriadou 2019) for a five-cluster solution on a set of data generated to contain five relatively distinct clusters. Looking first at the rectangle for cluster one, we see that observations in clusters two and three have the cluster one centroid as their second closest. These observations form the two other stripes within the rectangle. Observations in cluster three are further away from cluster one, but a number of observations in cluster three have a distance to the centroid of cluster one similar to those observations that belong to cluster one. Overall though, the stripes plot in Figure 6.19 suggests that the five-cluster solution matches quite well the actual structure in the data. The situation is quite different in Figure 6.20, where the stripes plot

```
R> library("flexclust")
R> library("mvtnorm")
R> set.seed(290875)
R> x <- rbind(rmvnorm(n = 20, mean = c(0, 0),
+                     sigma = diag(2)),
+             rmvnorm(n = 20, mean = c(3, 3),
+                     sigma = 0.5 * diag(2)),
+             rmvnorm(n = 20, mean = c(7, 6),
+                     sigma = 0.5 * (diag(2) + 0.25)))
R> k <- cclust(x, k = 5, save.data = TRUE)
R> plot(k, hull = FALSE, col = rep("black", 5), xlab = "x", ylab = "y")
```

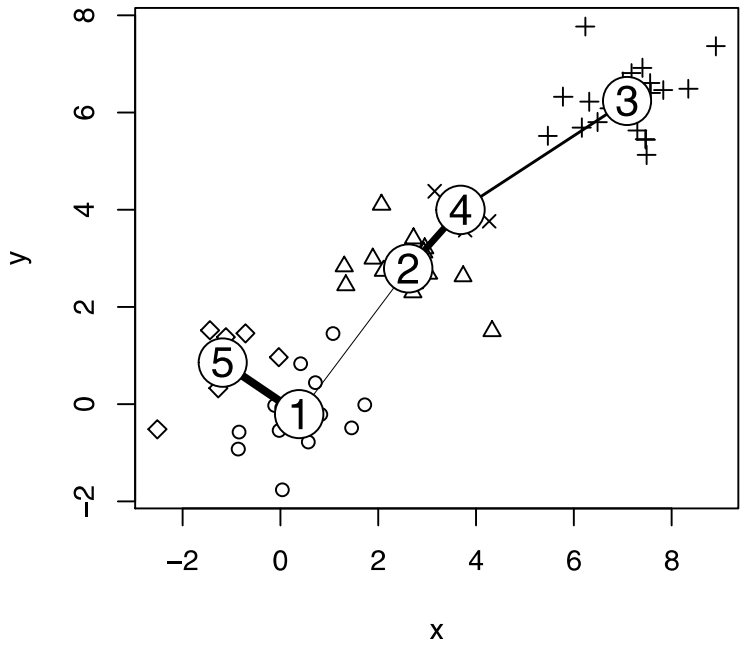

Fig. 6.17. Neighbourhood plot of k-means five-cluster solution for bivariate data containing three clusters.


```
R> k <- cclust(pots, k = 3, save.data = TRUE)
R> plot(k, project = prcomp(pots), hull = FALSE, col = rep("black", 3),
+       xlab = "PC1", ylab = "PC2")
```

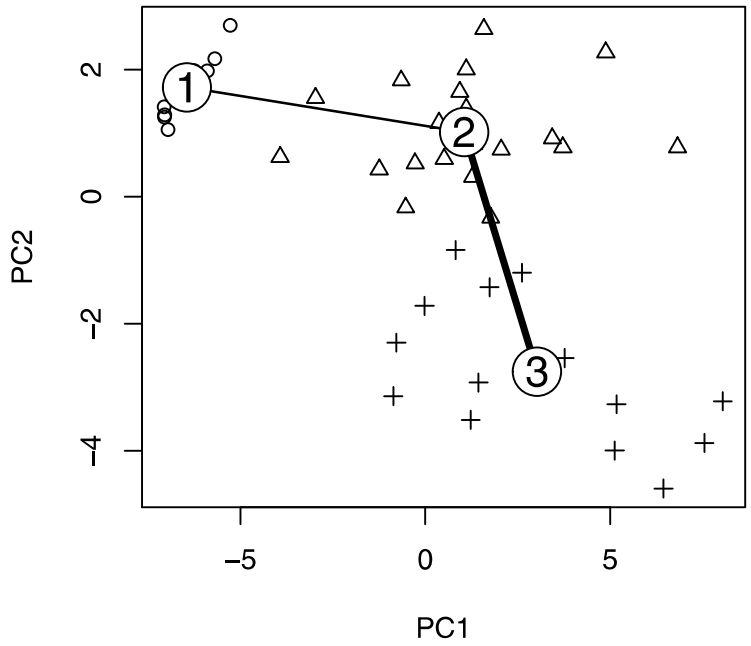

Fig. 6.18. Neighbourhood plot of k-means three-cluster solution for pottery data.

for the k-means five-group solution suggests that the clusters in this solution are not well separated, implying perhaps that the five-group solution is not appropriate for the data in this case. Lastly, the stripes plot for the k-means three-group solution on the pottery data is shown in Figure 6.21. The graphic confirms the three-group structure of the data.

All the information in a stripes plot is also available from a neighbourhood plot, but the former is dimension independent and may work well even for high-dimensional data where projections to two dimensions lose a lot of information about the structure in the data. Neither neighbourhood graphs nor stripes plots are infallible, but both offer some help in the often difficult task of evaluating and validating the solutions from a cluster analysis of a set of data.

```
R> set.seed(912345654)
R> x <- rbind(matrix(rnorm(100, sd = 0.5), ncol= 2 ),
+             matrix(rnorm(100, mean =4, sd = 0.5), ncol = 2),
+             matrix(rnorm(100, mean =7, sd = 0.5), ncol = 2),
+             matrix(rnorm(100, mean =-1.0, sd = 0.7), ncol = 2),
+             matrix(rnorm(100, mean =-4.0, sd = 1.0), ncol = 2))
R> c5 <- cclust(x, 5, save.data = TRUE)
R> stripes(c5, type = "second", col = 1)
```

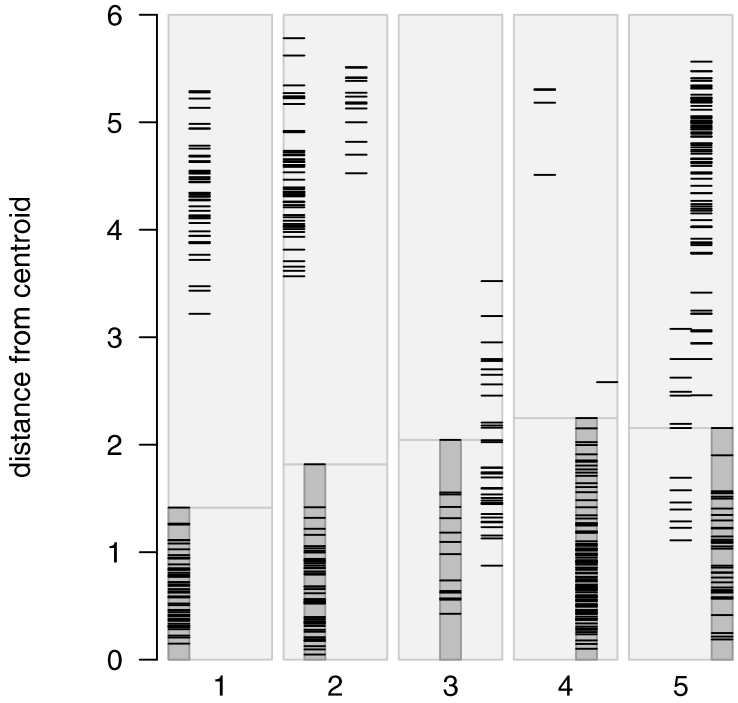

Fig. 6.19. Stripes plot of k-means solution for artificial data.

6.7 Summary

Cluster analysis techniques are used to search for clusters or groups in a priori unclassified multivariate data. Although clustering techniques are potentially very useful for the exploration of multivariate data, they require care in their application if misleading solutions are to be avoided. Many methods of cluster analysis have been developed, and most studies have shown that no one method is best for all types of data. But the more statistical techniques covered


```
R> set.seed(912345654)
R> x <- rbind(matrix(rnorm(100, sd = 2.5), ncol = 2),
+             matrix(rnorm(100, mean = 3, sd = 0.5), ncol = 2),
+             matrix(rnorm(100, mean = 5, sd = 0.5), ncol = 2),
+             matrix(rnorm(100, mean = -1.0, sd = 1.5), ncol = 2),
+             matrix(rnorm(100, mean = -4.0, sd = 2.0), ncol = 2))
R> c5 <- cclust(x, 5, save.data = TRUE)
R> stripes(c5, type = "second", col = 1)
```

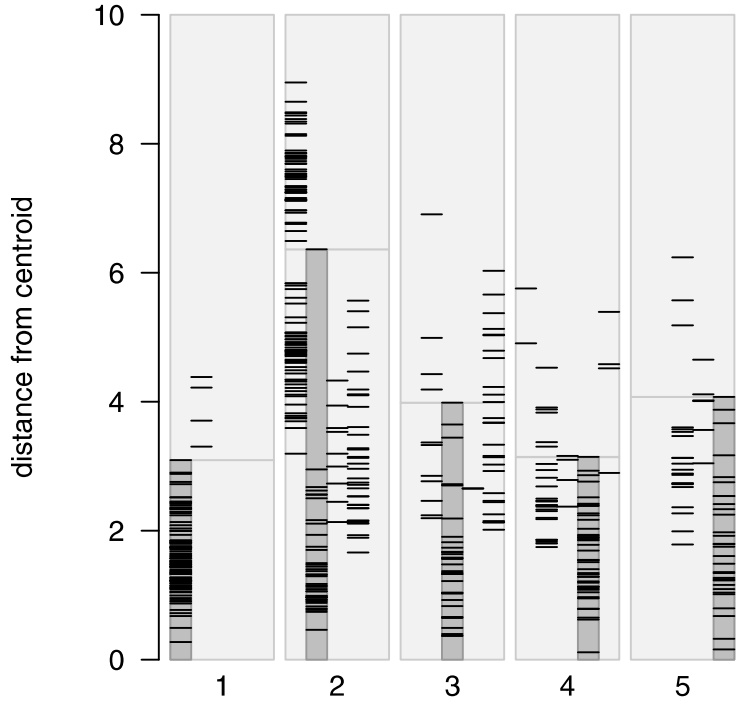

Fig. 6.20. Stripes plot of k-means solution for artificial data.

briefly in Section 6.5 and in more detail in Everitt et al. (2011) have definite *statistical* advantages because the clustering is based on sensible models for the data. Cluster analysis is a large area and has been covered only briefly in this chapter. The many problems that need to be considered when using clustering in practice have barely been touched upon. For a detailed discussion of these problems, again see Everitt et al. (2011).

```
R> set.seed(15)
R> c5 <- cclust(pots, k = 3, save.data = TRUE)
R> stripes(c5, type = "second", col = "black")
```

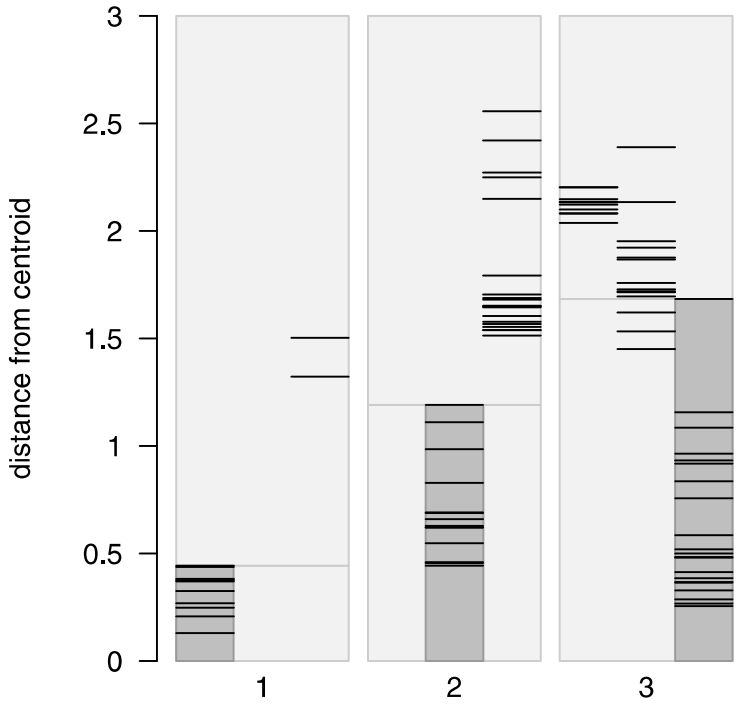

Fig. 6.21. Stripes plot of three-group k-means solution for pottery data.

Finally, we should mention in passing a technique known as projection pursuit. In essence, and like principal components analysis, projection pursuit seeks a low-dimensional projection of a multivariate data set but one that may be more likely to be successful in uncovering any cluster (or more exotic) structure in the data than principal component plots using the first few principal component scores. The technique is described in detail in Jones and Sibson (1987) and more recently in Cook and Swayne (2007).

6.8 Exercises

Ex. 6.1 Apply k-means to the crime rate data after standardising each variable by its standard deviation. Compare the results with those given in the text found by standardising by a variable's range.

Ex. 6.2 Calculate the first five principal components scores for the Romano-British pottery data, and then construct the scatterplot matrix of the scores, displaying the contours of the estimated bivariate density for each panel of the plot and a boxplot of each score in the appropriate place on the diagonal. Label the points in the scatterplot matrix with their kiln numbers.

Ex. 6.3 Return to the air pollution data given in Chapter 1 and use finite mixtures to cluster the data on the basis of the six climate and ecology variables (i.e., excluding the sulphur dioxide concentration). Investigate how sulphur dioxide concentration varies in the clusters you find both graphically and by formal significance testing.

7

Confirmatory Factor Analysis and Structural Equation Models

7.1 Introduction

An exploratory factor analysis as described in Chapter 5 is used in the early investigation of a set of multivariate data to determine whether the factor analysis model is useful in providing a parsimonious way of describing and accounting for the relationships between the observed variables. The analysis will determine which observed variables are most highly correlated with the common factors and how many common factors are needed to give an adequate description of the data. In an exploratory factor analysis, no constraints are placed on which manifest variables load on which factors. In this chapter, we will consider *confirmatory factor analysis models* in which *particular* manifest variables are allowed to relate to *particular* factors whilst other manifest variables are constrained to have zero loadings on some of the factors. A confirmatory factor analysis model may arise from theoretical considerations or be based on the results of an exploratory factor analysis where the investigator might wish to postulate a specific model for a new set of similar data, one in which the loadings of some variables on some factors are fixed at zero because they were "small" in the exploratory analysis and perhaps to allow some pairs of factors but not others to be correlated. It is important to emphasise that whilst it is perfectly appropriate to arrive at a factor model to submit to a confirmatory analysis from an exploratory factor analysis, the model *must* be tested on a fresh set of data. Models must not be generated and tested on the same data.

Confirmatory factor analysis models are a subset of a more general approach to modelling latent variables known as *structural equation modelling* or *covariance structure modelling*. Such models allow both response *and* explanatory latent variables linked by a series of linear equations. Although more complex than confirmatory factor analysis models, the aim of structural equation models is essentially the same, namely to explain the correlations or covariances of the observed variables in terms of the relationships of these variables to the assumed underlying latent variables and the relationships postulated between the latent variables themselves. Structural equation models represent the convergence of relatively independent research traditions in psychiatry, psychology, econometrics, and biometrics. The idea of latent variables in psychometrics arises from Spearman's early work on general intelligence. The concept of simultaneous directional influences of some variables on others has been part of economics for several decades, and the resulting simultaneous equation models have been used extensively by economists but essentially only with observed variables. Path analysis was introduced by Wright (1934) in a biometrics context as a method for studying the direct and indirect effects of variables. The quintessential feature of path analysis is a diagram showing how a set of explanatory variables influence a dependent variable under consideration. How the paths are drawn determines whether the explanatory variables are correlated causes, mediated causes, or independent causes. Some examples of path diagrams appear later in the chapter. (For more details of path analysis, see Schumaker and Lomax 1996).

Later, path analysis was taken up by sociologists such as Blalock (1961), Blalock (1963) and then by Duncan (1969), who demonstrated the value of combining path-analytic representation with simultaneous equation models. And, finally, in the 1970s, several workers most prominent of whom were Jöreskog (1973), Bentler (1980), and Browne (1974), combined all these various approaches into a general method that could in principle deal with extremely complex models in a routine manner.

B. Everitt and T. Hothorn, *An Introduction to Applied Multivariate Analysis with R*: Use R!,
DOI 10.1007/978-1-4419-9650-3_7, © Springer Science+Business Media, LLC 2011

7.2 Estimation, identification, and assessing fit for confirmatory factor and structural equation models

7.2.1 Estimation

Structural equation models will contain a number of parameters that need to be estimated from the covariance or correlation matrix of the manifest variables. Estimation involves finding values for the model parameters that minimise a *discrepancy function* indicating the magnitude of the differences between the elements of $\mathbf{S}$, the observed covariance matrix of the manifest variables and those of $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, the covariance matrix implied by the fitted model (i.e., a matrix the elements of which are functions of the parameters of the model), contained in the vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_t)^\top$.

There are a number of possibilities for discrepancy functions; for example, the ordinary least squares discrepancy function, FLS, is

$$\text{FLS}(\mathbf{S}, \boldsymbol{\Sigma}(\boldsymbol{\theta})) = \sum_{i<j} \sum_{j} (s_{ij} - \sigma_{ij}(\boldsymbol{\theta}))^2,$$

where s_{ij} and $\sigma_{ij}(\boldsymbol{\theta})$ are the elements of $\mathbf{S}$ and $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. But this criterion has several problems that make it unsuitable for estimation; for example, it is not

independent of the scale of the manifest variables, and so different estimates of the model parameters would be produced using the sample covariance matrix and the sample correlation matrix. Other problems with the least squares criterion are detailed in Everitt (1984).

The most commonly used method of estimating the parameters in confirmatory factor and structural equation models is maximum likelihood under the assumption that the observed data have a multivariate normal distribution. It is easy to show that maximising the likelihood is now equivalent to minimising the discrepancy function, FML, given by

$$\text{FML}(\mathbf{S}, \boldsymbol{\Sigma}(\boldsymbol{\theta})) = \log|\boldsymbol{\Sigma}(\boldsymbol{\theta})| - \log|\mathbf{S}| + \text{tr}(\mathbf{S}\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}) - q$$

(cf. maximum likelihood factor analysis in Chapter 5). We see that by varying the parameters $\theta_1, \ldots, \theta_t$ so that $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ becomes more like $\mathbf{S}$, FML becomes smaller. Iterative numerical algorithms are needed to minimise the function FML with respect to the parameters, but for details see Everitt (1984) and Everitt and Dunn (2001).

7.2.2 Identification

Consider the following simple example of a model in which there are three manifest variables, x, x', and y, and two latent variables, u and v, with the relationships between the manifest and latent variables being

$$\begin{aligned} x &= u + \delta, \\ y &= v + \epsilon, \\ x' &= u + \delta'. \end{aligned}$$

If we assume that δ, δ', and ϵ have expected values of zero, that δ and δ' are uncorrelated with each other and with u, and that ϵ is uncorrelated with v, then the covariance matrix of the three manifest variables may be expressed in terms of parameters representing the variances and covariances of the residuals and the latent variables as

$$\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \begin{pmatrix} \theta_1 + \theta_2 & & \\ \theta_3 & \theta_4 + \theta_5 & \\ \theta_3 & \theta_4 & \theta_4 + \theta_6 \end{pmatrix},$$

where $\boldsymbol{\theta}^\top = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ and $\theta_1 = \mathsf{Var}(v)$, $\theta_2 = \mathsf{Var}(\epsilon)$, $\theta_3 = \mathsf{Cov}(v, u)$, $\theta_4 = \mathsf{Var}(u)$, $\theta_5 = \mathsf{Var}(\delta)$, and $\theta_6 = \mathsf{Var}(\delta')$. It is immediately apparent that estimation of the parameters in this model poses a problem. The two parameters θ_1 and θ_2 are not uniquely determined because one can be, for example, increased by some amount and the other decreased by the same amount without altering the covariance matrix predicted by the model. In other words, in this example, different sets of parameter values (i.e., different $\boldsymbol{\theta}$s) will lead to the same predicted covariance matrix, $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. The model is said to be *unidentifiable*. Formally, a model is identified if and only if $\boldsymbol{\Sigma}(\boldsymbol{\theta}_1) = \boldsymbol{\Sigma}(\boldsymbol{\theta}_2)$ implies

that $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$. In Chapter 5, it was pointed out that the parameters in the exploratory factor analysis model are not identifiable unless some constraints are introduced because different sets of factor loadings can give rise to the same predicted covariance matrix. In confirmatory factor analysis models and more general covariance structure models, identifiability depends on the choice of model and on the specification of fixed, constrained (for example, two parameters constrained to equal one another), and free parameters. If a parameter is not identified, it is not possible to find a consistent estimate of it. Establishing model identification in confirmatory factor analysis models (and in structural equation models) can be difficult because there are no simple, practicable, and universally applicable rules for evaluating whether a model is identified, although there *is* a simple necessary but not sufficient condition for identification, namely that the number of free parameters in a model, t, be less than $q(q+1)/2$. For a more detailed discussion of the identifiability problem, see Bollen and Long (1993).

7.2.3 Assessing the fit of a model

Once a model has been pronounced identified and its parameters estimated, the next step becomes that of assessing how well the model-predicted covariance matrix fits the covariance matrix of the manifest variables. A global measure of fit of a model is provided by the likelihood ratio statistic given by $X^2 = (N-1)\text{FML}_{\text{min}}$, where N is the sample size and FML_{min} is the minimised value of the maximum likelihood discrepancy function given in Subsection 7.2.1. If the sample size is sufficiently large, the X^2 statistic provides a test that the population covariance matrix of the manifest variables is equal to the covariance implied by the fitted model against the alternative hypothesis that the population matrix is unconstrained. Under the equality hypothesis, X^2 has a chi-squared distribution with degrees of freedom ν given by $\frac{1}{2}q(q+1) - t$, where t is the number of free parameters in the model.

The likelihood ratio statistic is often the only measure of fit quoted for a fitted model, but on its own it has limited practical use because in large samples even relatively trivial departures from the equality null hypothesis will lead to its rejection. Consequently, in large samples most models may be rejected as statistically untenable. A more satisfactory way to use the test is for a comparison of a series of nested models where a large difference in the statistic for two models compared with the difference in the degrees of freedom of the models indicates that the additional parameters in one of the models provide a genuine improvement in fit.

Further problems with the likelihood ratio statistic arise when the observations come from a population where the manifest variables have a non-normal distribution. Browne (1982) demonstrates that in the case of a distribution with substantial kurtosis, the chi-squared distribution may be a poor approximation for the null distribution of X^2. Browne suggests that before using the test it is advisable to assess the degree of kurtosis of the data by using

Mardia's coefficient of multivariate kurtosis (see Mardia et al. 1979). Browne's suggestion appears to be little used in practise.

Perhaps the best way to assess the fit of a model is to use the X^2 statistic alongside one or more of the following procedures:

- Visual inspection of the residual covariances (i.e., the differences between the covariances of the manifest variables and those predicted by the fitted model). These residuals should be small when compared with the values of the observed covariances or correlations.
- Examination of the standard errors of the parameters and the correlations between these estimates. If the correlations are large, it may indicate that the model being fitted is almost unidentified.
- Estimated parameter values outside their possible range; i.e., negative variances or absolute values of correlations greater than unity are often an indication that the fitted model is fundamentally wrong for the data.

In addition, a number of fit indices have been suggested that can sometimes be useful. For example, the *goodness-of-fit index* (GFI) is based on the ratio of the sum of squared distances between the matrices observed and those reproduced by the model covariance, thus allowing for scale.

The GFI measures the amount of variance and covariance in $\mathbf{S}$ that is accounted for by the covariance matrix predicted by the putative model, namely $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, which for simplicity we shall write as $\boldsymbol{\Sigma}$. For maximum likelihood estimation, the GFI is given explicitly by

$$\text{GFI} = 1 - \frac{\text{tr}\left(\mathbf{S}\hat{\Sigma}^{-1} - \mathbf{I}\right)\left(\mathbf{S}\hat{\boldsymbol{\Sigma}}^{-1} - \mathbf{I}\right)}{\text{tr}\left(\mathbf{S}\hat{\boldsymbol{\Sigma}}^{-1}\mathbf{S}\hat{\boldsymbol{\Sigma}}^{-1}\right)}.$$

The GFI can take values between zero (no fit) and one (perfect fit); in practise, only values above about 0.9 or even 0.95 suggest an acceptable level of fit.

The *adjusted goodness of fit index* (AGFI) adjusts the GFI index for the degrees of freedom of a model relative to the number of variables. The AGFI is calculated as follow;

$$\text{AGFI} = 1 - (k/\text{df})(1 - \text{GFI}),$$

where k is the number of unique values in $\mathbf{S}$ and df is the number of degrees of freedom in the model (discussed later). The GFI and AGFI can be used to compare the fit of two different models with the same data or compare the fit of models with different data, for example male and female data sets.

A further fit index is the *root-mean-square residual* (RMSR), which is the square root of the mean squared differences between the elements in $\mathbf{S}$ and $\hat{\boldsymbol{\Sigma}}$. It can be used to compare the fit of two different models with the same data. A value of RMSR < 0.05 is generally considered to indicate a reasonable fit.

A variety of other fit indices have been proposed, including the *Tucker-Lewis index* and the *normed fit index*; for details, see Bollen and Long (1993).

7.3 Confirmatory factor analysis models

In a confirmatory factor model the loadings for some observed variables on some of the postulated common factors will be set a priori to zero. Additionally, some correlations between factors might also be fixed at zero. Such a model is fitted to a set of data by estimating its *free* parameters; i.e., those not fixed at zero by the investigator. Estimation is usually by maximum likelihood using the FML discrepancy function.

We will now illustrate the application of confirmatory factor analysis with two examples.

7.3.1 Ability and aspiration

Calsyn and Kenny (1977) recorded the values of the following six variables for 556 white eighth-grade students:

`SCA`: self-concept of ability;
`PPE`: perceived parental evaluation;
`PTE`: perceived teacher evaluation;
`PFE`: perceived friend's evaluation;
`EA`: educational aspiration;
`CP`: college plans.

Calsyn and Kenny (1977) postulated that two underlying latent variables, *ability* and *aspiration*, generated the relationships between the observed variables. The first four of the manifest variables were assumed to be indicators of ability and the last two indicators of aspiration; the latent variables, ability and aspiration, are assumed to be correlated. The regression-like equations that specify the postulated model are

$$\begin{aligned}
\text{SCA} &= \lambda_1 f_1 + 0 f_2 + u_1,\\
\text{PPE} &= \lambda_2 f_1 + 0 f_2 + u_2,\\
\text{PTE} &= \lambda_3 f_1 + 0 f_2 + u_3,\\
\text{PFE} &= \lambda_4 f_1 + 0 f_2 + u_4,\\
\text{AE} &= 0 f_1 + \lambda_5 f_2 + u_5,\\
\text{CP} &= 0 f_1 + \lambda_6 f_2 + u_6,
\end{aligned}$$

where f_1 represents the ability latent variable and f_2 represents the aspiration latent variable. Note that, unlike in exploratory factor analysis, a number of factor loadings are fixed at zero and play no part in the estimation process. The model has a total of 13 parameters to estimate, six factor loadings (λ_1 to λ_6), six specific variances (ψ_1 to ψ_6), and one correlation between ability and aspiration (ρ). (To be consistent with the nomenclature used in Subsection 7.2.1, all parameters should be suffixed thetas; this could, however, become confusing, so we have changed the nomenclature and use lambdas, etc., in a manner similar to how they are used in Chapter 5.) The observed

correlation matrix given in Figure 7.1 has six variances and 15 correlations, a total of 21 terms. Consequently, the postulated model has $21 - 13 = 8$ degrees of freedom. The figure depicts each correlation by an ellipse whose shape tends towards a line with slope 1 for correlations near 1, to a circle for correlations near zero, and to a line with negative slope -1 for negative correlations near -1. In addition, 100 times the correlation coefficient is printed inside the ellipse and colour-coding indicates strong negative (dark) to strong positive (light) correlations.

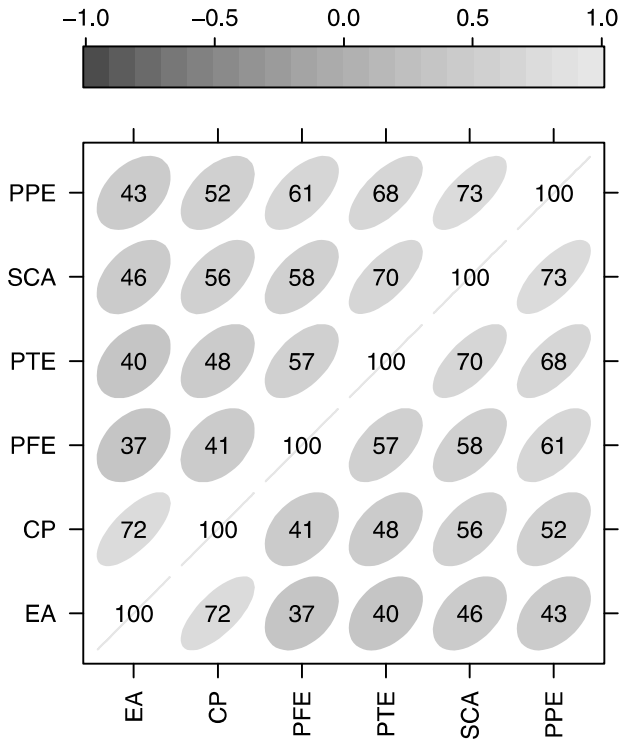

Fig. 7.1. Correlation matrix of ability and aspiration data; values given are correlation coefficients $\times 100$.

The R code, contained in the package **sem** (Fox, Kramer, and Friendly 2010), for fitting the model is

```
R> ability_model <- specify.model(file = "ability_model.txt")
R> ability_sem <- sem(ability_model, ability, 556)
```


Here, the model is specified in a text file (called `ability_model.txt` in our case) with the following content:

```
Ability     -> SCA, lambda1, NA
Ability     -> PPE, lambda2, NA
Ability     -> PTE, lambda3, NA
Ability     -> PFE, lambda4, NA
Aspiration  -> EA, lambda5, NA
Aspiration  -> CP, lambda6, NA
Ability    <-> Aspiration, rho, NA
SCA        <-> SCA, theta1, NA
PPE        <-> PPE, theta2, NA
PTE        <-> PTE, theta3, NA
PFE        <-> PFE, theta4, NA
EA         <-> EA, theta5, NA
CP         <-> CP, theta6, NA
Ability    <-> Ability, NA, 1
Aspiration <-> Aspiration, NA, 1
```

The model is specified via arrows in the so-called reticular action model (RAM) notation. The text consists of three columns. The first one corresponds to an arrow specification where single-headed or directional arrows correspond to regression coefficients and double-headed or bidirectional arrows correspond to variance parameters. The second column denotes parameter names, and the third one assigns values to fixed parameters. Further details are available from the corresponding pages of the manual for the **sem** package.

The results from fitting the ability and aspiration model to the observed correlations are available via

```
R> summary(ability_sem)

 Model Chisquare =  9.2557   Df =  8 Pr(>Chisq) = 0.32118
 Chisquare (null model) =  1832.0   Df =  15
 Goodness-of-fit index =  0.99443
 Adjusted goodness-of-fit index =  0.98537
 RMSEA index =  0.016817   90% CI: (NA, 0.05432)
 Bentler-Bonnett NFI =  0.99495
 Tucker-Lewis NNFI =  0.9987
 Bentler CFI =  0.9993
 SRMR =  0.012011
 BIC =  -41.310

 Normalized Residuals
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.4410 -0.1870  0.0000 -0.0131  0.2110  0.5330

 Parameter Estimates
```

```
        Estimate Std Error z value Pr(>|z|)
lambda1 0.86320  0.035182  24.5355 0.0000000
lambda2 0.84932  0.035489  23.9323 0.0000000
lambda3 0.80509  0.036409  22.1123 0.0000000
lambda4 0.69527  0.038678  17.9757 0.0000000
lambda5 0.77508  0.040365  19.2020 0.0000000
lambda6 0.92893  0.039417  23.5665 0.0000000
rho     0.66637  0.030955  21.5273 0.0000000
theta1  0.25488  0.023502  10.8450 0.0000000
theta2  0.27865  0.024263  11.4847 0.0000000
theta3  0.35184  0.026916  13.0715 0.0000000
theta4  0.51660  0.034820  14.8365 0.0000000
theta5  0.39924  0.038214  10.4475 0.0000000
theta6  0.13709  0.043530   3.1493 0.0016366

lambda1 SCA <--- Ability
lambda2 PPE <--- Ability
lambda3 PTE <--- Ability
lambda4 PFE <--- Ability
lambda5 EA <--- Aspiration
lambda6 CP <--- Aspiration
rho     Aspiration <--> Ability
theta1  SCA <--> SCA
theta2  PPE <--> PPE
theta3  PTE <--> PTE
theta4  PFE <--> PFE
theta5  EA <--> EA
theta6  CP <--> CP

 Iterations =  28
```

(Note that the two latent variables have their variances fixed at one, although it is the fixing that is important, not the value at which they are fixed; these variances cannot be free parameters to be estimated.) The z values test whether parameters are significantly different from zero. All have very small associated p-values. Of particular note amongst the parameter estimates is the correlation between "true" ability and "true" aspiration; this is known as a *disattenuated correlation* and is uncontaminated by measurement error in the observed indicators of the two latent variables. In this case the estimate is 0.666, with a standard error of 0.031. An approximate 95% confidence interval for the disattenuated correlation is $[0.606, 0.727]$.

A *path diagram* (see Everitt and Dunn 2001) for the correlated, two-factor model is shown in Figure 7.2. Note that the R function `path.diagram()` "only" produces a textual representation of the graph, here in file `ability_sem.dot`.

The graphviz graph visualisation software (Gansner and North 2000) needs to be installed in order to compile the corresponding PDF file.

```
R> path.diagram(ability_sem)
```

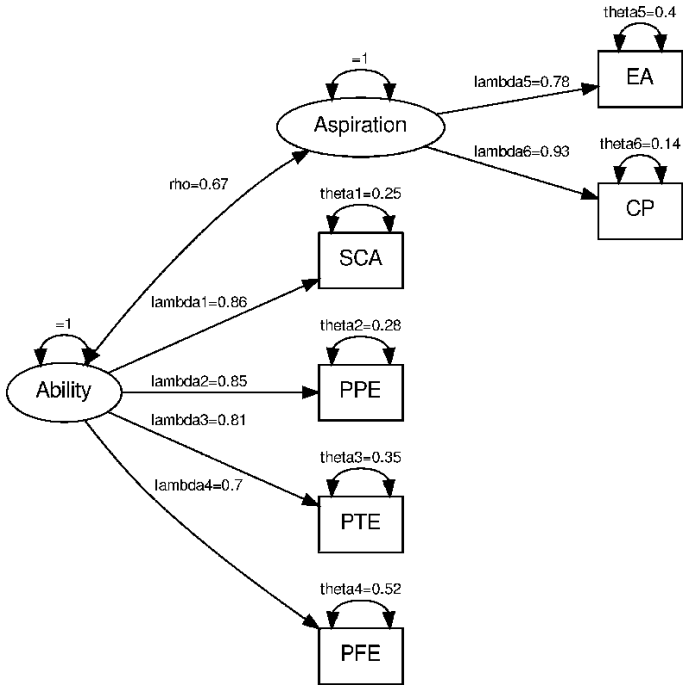

Fig. 7.2. Ability path diagram.

The fit of the model can be partially judged using the chi-square statistic described in Subsection 7.2.3, which in this case takes the value 9.256 with eight degrees of freedom and an associated p-value of 0.321, suggesting that the postulated model fits the data very well. (The chi-square test for the null model is simply a test that the population covariance matrix of the observed variables is diagonal; i.e., that the observed variables are independent. In most cases, this null model will be rejected; if it is not, the model-fitting exercise is a waste of time.) The various fit indices also indicate that the

model is a good fit for the data. Also helpful in assessing the fit of the model are the summary statistics for the *normed residuals*, which are essentially the differences between corresponding elements of $\mathbf{S}$ and $\hat{\boldsymbol{\Sigma}}(\boldsymbol{\theta})$ but scaled so that they are unaffected by the differences in the variances of the observed variables. The normed residuals, r_{ij}^*, are defined as

$$r_{ij}^* = \frac{s_{ij} - \hat{\sigma}_{ij}}{(\hat{\sigma}_{ij}\hat{\sigma_j}^2 + \hat{\sigma}_{ij}^2)/n)^{1/2}}.$$

Generally the absolute values of the normed residuals should all be less than 2 to claim that the current model fits the data well. In the ability and aspiration example, this seems to be the case. (Note that with confirmatory factor models the standard errors of parameters become of importance because they allow the investigator to assess whether parameters might be dropped from the model to find a more parsimonious model that still provides an adequate fit to the data. In exploratory factor analysis, standard errors of factor loadings can be calculated, but they are hardly ever used; instead an informal interpretation of factors is made.)

7.3.2 A confirmatory factor analysis model for drug use

For our second example of fitting a confirmatory factor analysis model, we return to the drug use among students data introduced in Chapter 5. In the original investigation of these data reported by Huba et al. (1981), a confirmatory factor analysis model was postulated, the model arising from consideration of previously reported research on student drug use. The model involved the following three latent variables:

f_1: Alcohol use, with non-zero loadings on beer, wine, spirits, and cigarette use.

f_2: Cannabis use, with non-zero loadings on marijuana, hashish, cigarette, and wine use. The cigarette variable is assumed to load on both the first and second latent variables because it sometimes occurs with both alcohol and marijuana use and at other times does not. The non-zero loading on wine was allowed because of reports that wine is frequently used with marijuana and that consequently some of the use of wine might be an indicator of tendencies toward cannabis use.

f_3: Hard drug use, with non-zero loadings on amphetamines, tranquillizers, hallucinogenics, hashish, cocaine, heroin, drug store medication, inhalants, and spirits. The use of each of these substances was considered to suggest a strong commitment to the notion of psychoactive drug use.

Each pair of latent variables is assumed to be correlated so that these correlations are allowed to be free parameters that need to be estimated. The variance of each latent variance must, however, be fixed–they are not free parameters that can be estimated–and here as usual we will specify that each of these variances takes the value one. So the proposed model can be specified by the following series of equations:

$$\begin{aligned}
\text{cigarettes} &= \lambda_1 f_1 + \lambda_2 f_2 + 0 f_3 + u_1,\\
\text{beer} &= \lambda_3 f_1 + 0 f_2 + 0 f_3 + u_2,\\
\text{wine} &= \lambda_4 f_1 + \lambda_5 f_2 + 0 f_3 + u_3,\\
\text{spirits} &= \lambda_6 f_1 + 0 f_2 + \lambda_7 f_3 + u_4,\\
\text{cocaine} &= 0 f_1 + 0 f_2 + \lambda_8 f_3 + u_5,\\
\text{tranquillizers} &= 0 f_1 + 0 f_2 + \lambda_9 f_3 + u_6,\\
\text{drug store medication} &= 0 f_1 + 0 f_2 + \lambda_{10} f_3 + u_7,\\
\text{heroin} &= 0 f_1 + 0 f_2 + \lambda_{11} f_3 + u_8,\\
\text{marijuana} &= 0 f_1 + \lambda_{12} f_2 + 0 f_3 + u_9,\\
\text{inhalants} &= 0 f_1 + 0 f_2 + \lambda_{15} f_3 + u_{11},\\
\text{hallucinogenics} &= 0 f_1 + 0 f_2 + \lambda_{16} f_3 + u_{12},\\
\text{amphetamines} &= 0 f_1 + 0 f_2 + \lambda_{17} f_3 + u_{13}.
\end{aligned}$$

The proposed model also allows for non-zero correlations between each pair of latent variables and so has a total of 33 parameters to estimate–17 loadings (λ_1 to λ_{17}), 13 specific variances (ψ_1 to ψ_{13}), and three correlations between latent variables (ρ_1 to ρ_3). Consequently, the model has $91 - 33 = 58$ degrees of freedom. We first abbreviate the names of the variables via

```
R> rownames(druguse) <- colnames(druguse) <- c("Cigs",
+      "Beer", "Wine", "Liqr", "Cocn", "Tran", "Drug",
+      "Hern", "Marj", "Hash", "Inhl", "Hall", "Amph")
```

To fit the model, we can use the R code

```
R> druguse_model <- specify.model(file = "druguse_model.txt")
R> druguse_sem <- sem(druguse_model, druguse, 1634)
```

where the model (stored in the text file `druguse_model.txt`) reads

```
Alcohol   -> Cigs, lambda1, NA
Alcohol   -> Beer, lambda3, NA
Alcohol   -> Wine, lambda4, NA
Alcohol   -> Liqr, lambda6, NA
Cannabis  -> Cigs, lambda2, NA
Cannabis  -> Wine, lambda5, NA
Cannabis  -> Marj, lambda12, NA
Cannabis  -> Hash, lambda13, NA
Hard      -> Liqr, lambda7, NA
Hard      -> Cocn, lambda8, NA
Hard      -> Tran, lambda9, NA
Hard      -> Drug, lambda10, NA
Hard      -> Hern, lambda11, NA
```

```
Hard      -> Hash, lambda14, NA
Hard      -> Inhl, lambda15, NA
Hard      -> Hall, lambda16, NA
Hard      -> Amph, lambda17, NA
Cigs     <-> Cigs, theta1, NA
Beer     <-> Beer, theta2, NA
Wine     <-> Wine, theta3, NA
Liqr     <-> Liqr, theta4, NA
Cocn     <-> Cocn, theta5, NA
Tran     <-> Tran, theta6, NA
Drug     <-> Drug, theta7, NA
Hern     <-> Hern, theta8, NA
Marj     <-> Marj, theta9, NA
Hash     <-> Hash, theta10, NA
Inhl     <-> Inhl, theta11, NA
Hall     <-> Hall, theta12, NA
Amph     <-> Amph, theta13, NA
Alcohol  <-> Alcohol, NA, 1
Cannabis <-> Cannabis, NA, 1
Hard     <-> Hard, NA, 1
Alcohol  <-> Cannabis, rho1, NA
Alcohol  <-> Hard, rho2, NA
Cannabis <-> Hard, rho3, NA
```

The results of fitting the proposed model are

```
R> summary(druguse_sem)

 Model Chisquare =  324.09   Df =  58 Pr(>Chisq) = 0
 Chisquare (null model) =  6613.7   Df =  78
 Goodness-of-fit index =  0.9703
 Adjusted goodness-of-fit index =  0.9534
 RMSEA index =  0.053004   90% CI: (0.047455, 0.058705)
 Bentler-Bonnett NFI =  0.951
 Tucker-Lewis NNFI =  0.94525
 Bentler CFI =  0.95929
 SRMR =  0.039013
 BIC =  -105.04

 Normalized Residuals
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-3.0500 -0.8800  0.0000 -0.0217  0.9990  4.5800

 Parameter Estimates
         Estimate Std Error z value Pr(>|z|)
lambda1   0.35758 0.034332  10.4153 0.0000e+00
```



```
lambda3   0.79159 0.022684  34.8962 0.0000e+00
lambda4   0.87588 0.037963  23.0716 0.0000e+00
lambda6   0.72176 0.023575  30.6150 0.0000e+00
lambda2   0.33203 0.034661   9.5793 0.0000e+00
lambda5  -0.15202 0.037155  -4.0914 4.2871e-05
lambda12  0.91237 0.030833  29.5907 0.0000e+00
lambda13  0.39549 0.030061  13.1559 0.0000e+00
lambda7   0.12347 0.022878   5.3971 6.7710e-08
lambda8   0.46467 0.025954  17.9038 0.0000e+00
lambda9   0.67554 0.024001  28.1468 0.0000e+00
lambda10  0.35842 0.026488  13.5312 0.0000e+00
lambda11  0.47591 0.025813  18.4367 0.0000e+00
lambda14  0.38199 0.029533  12.9343 0.0000e+00
lambda15  0.54297 0.025262  21.4940 0.0000e+00
lambda16  0.61825 0.024566  25.1667 0.0000e+00
lambda17  0.76336 0.023224  32.8695 0.0000e+00
theta1    0.61155 0.023495  26.0284 0.0000e+00
theta2    0.37338 0.020160  18.5210 0.0000e+00
theta3    0.37834 0.023706  15.9597 0.0000e+00
theta4    0.40799 0.019119  21.3398 0.0000e+00
theta5    0.78408 0.029381  26.6863 0.0000e+00
theta6    0.54364 0.023469  23.1644 0.0000e+00
theta7    0.87154 0.031572  27.6051 0.0000e+00
theta8    0.77351 0.029066  26.6126 0.0000e+00
theta9    0.16758 0.044839   3.7374 1.8592e-04
theta10   0.54692 0.022352  24.4691 0.0000e+00
theta11   0.70518 0.027316  25.8159 0.0000e+00
theta12   0.61777 0.025158  24.5551 0.0000e+00
theta13   0.41729 0.021422  19.4797 0.0000e+00
rho1      0.63317 0.028006  22.6079 0.0000e+00
rho2      0.31320 0.029574  10.5905 0.0000e+00
rho3      0.49893 0.027212  18.3349 0.0000e+00

lambda1  Cigs <--- Alcohol
lambda3  Beer <--- Alcohol
lambda4  Wine <--- Alcohol
lambda6  Liqr <--- Alcohol
lambda2  Cigs <--- Cannabis
lambda5  Wine <--- Cannabis
lambda12 Marj <--- Cannabis
lambda13 Hash <--- Cannabis
lambda7  Liqr <--- Hard
lambda8  Cocn <--- Hard
lambda9  Tran <--- Hard
lambda10 Drug <--- Hard
```

```
lambda11 Hern <--- Hard
lambda14 Hash <--- Hard
lambda15 Inhl <--- Hard
lambda16 Hall <--- Hard
lambda17 Amph <--- Hard
theta1   Cigs <--> Cigs
theta2   Beer <--> Beer
theta3   Wine <--> Wine
theta4   Liqr <--> Liqr
theta5   Cocn <--> Cocn
theta6   Tran <--> Tran
theta7   Drug <--> Drug
theta8   Hern <--> Hern
theta9   Marj <--> Marj
theta10  Hash <--> Hash
theta11  Inhl <--> Inhl
theta12  Hall <--> Hall
theta13  Amph <--> Amph
rho1     Cannabis <--> Alcohol
rho2     Hard <--> Alcohol
rho3     Hard <--> Cannabis

 Iterations =  31
```

Here the chi-square test for goodness of fit takes the value 324.092, which with 58 degrees of freedom has an associated p-value that is very small; the model does not appear to fit very well. But before we finally decide that the fitted model is unsuitable for the data, we should perhaps investigate its fit in other ways. Here we will look at the differences of the elements of the observed covariance matrix and the covariance matrix of the fitted model. We can find these differences using the following R code:

```
R> round(druguse_sem$S - druguse_sem$C, 3)
```

```
       Cigs   Beer   Wine   Liqr   Cocn   Tran   Drug   Hern
Cigs  0.000 -0.002  0.010 -0.009 -0.015  0.015 -0.009 -0.050
Beer -0.002  0.000  0.002  0.002 -0.047 -0.021  0.014 -0.055
Wine  0.010  0.002  0.000 -0.004 -0.039  0.005  0.039 -0.028
Liqr -0.009  0.002 -0.004  0.000 -0.047  0.022 -0.003 -0.069
Cocn -0.015 -0.047 -0.039 -0.047  0.000  0.035  0.042  0.100
Tran  0.015 -0.021  0.005  0.022  0.035  0.000 -0.021  0.034
Drug -0.009  0.014  0.039 -0.003  0.042 -0.021  0.000  0.030
Hern -0.050 -0.055 -0.028 -0.069  0.100  0.034  0.030  0.000
Marj  0.003 -0.012 -0.002  0.009 -0.026  0.007 -0.013 -0.063
Hash -0.023  0.025  0.005  0.029  0.034 -0.014 -0.045 -0.057
Inhl  0.094  0.068  0.075  0.065  0.020 -0.044  0.115  0.030
```



```
Hall -0.071 -0.065 -0.049 -0.077 -0.008 -0.051  0.010  0.026
Amph  0.033  0.010  0.032  0.026 -0.077  0.029 -0.042 -0.049
       Marj   Hash   Inhl   Hall   Amph
Cigs  0.003 -0.023  0.094 -0.071  0.033
Beer -0.012  0.025  0.068 -0.065  0.010
Wine -0.002  0.005  0.075 -0.049  0.032
Liqr  0.009  0.029  0.065 -0.077  0.026
Cocn -0.026  0.034  0.020 -0.008 -0.077
Tran  0.007 -0.014 -0.044 -0.051  0.029
Drug -0.013 -0.045  0.115  0.010 -0.042
Hern -0.063 -0.057  0.030  0.026 -0.049
Marj  0.000 -0.001  0.054 -0.077  0.047
Hash -0.001  0.000 -0.013  0.010  0.025
Inhl  0.054 -0.013  0.000  0.004 -0.022
Hall -0.077  0.010  0.004  0.000  0.039
Amph  0.047  0.025 -0.022  0.039  0.000
```

Some of these "raw" residuals look quite large in terms of a correlational scale; for example, that corresponding to drug store medication and inhalants. And the summary statistics for the normalised residuals show that the largest is far greater than the acceptable value of 2 and the smallest is rather less than the acceptable value of -2. Perhaps the overall message for the goodness-of-fit measures is that the fitted model does not provide an entirely adequate fit for the relationships between the observed variables. Readers are referred to the original paper by Huba et al. (1981) for details of how the model was changed to try to achieve a better fit.

7.4 Structural equation models

Confirmatory factor analysis models are relatively simple examples of a more general framework for modelling latent variables and are known as either structural equation models or covariance structure models. In such models, observed variables are again assumed to be indicators of underlying latent variables, but now regression equations linking the latent variables are incorporated. Such models are fitted as described in Subsection 7.2.1. We shall illustrate these more complex models by way of a single example.

7.4.1 Stability of alienation

To illustrate the fitting of a structural equation model, we shall look at a study reported by Wheaton, Muthen, Alwin, and Summers (1977) concerned with the stability over time of attitudes such as alienation and the relationship of such attitudes to background variables such as education and occupation. For this purpose, data on attitude scales were collected from 932 people in two rural regions in Illinois at three time points, 1966, 1967, and 1971. Here we

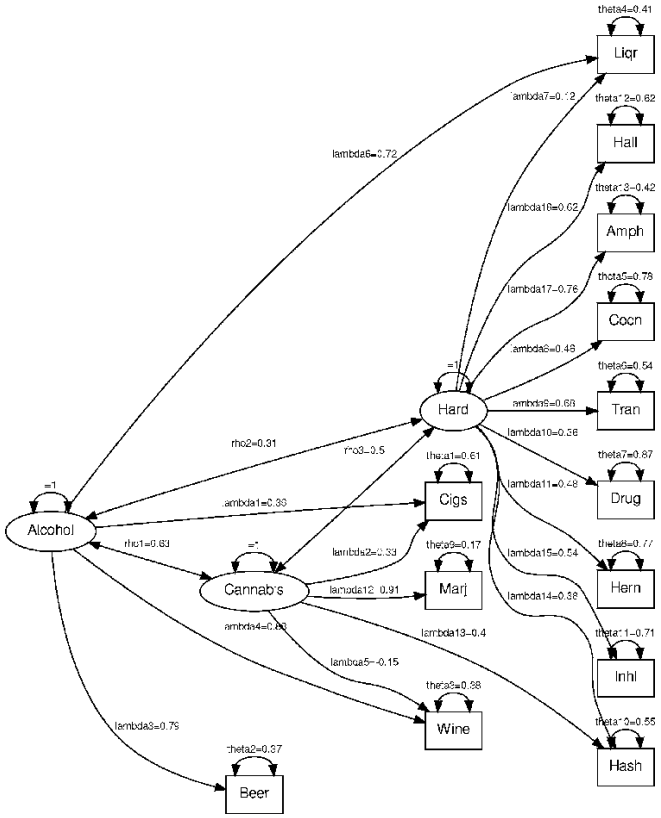

Fig. 7.3. Drug use path diagram.

shall only consider the data from 1967 and 1971. Scores on the anomia scale and powerlessness scale were taken to be indicators of the assumed latent variable, *alienation*. A respondent's years of schooling (education) and Duncan's socioeconomic index were assumed to be indicators of a respondent's *socioeconomic status*. The correlation matrix for the six observed variables is shown in Figure 7.4. The path diagram for the model to be fitted is shown in Figure 7.5. The latent variable socioeconomic status is considered to affect alienation at both time points, and alienation in 1967 also affects alienation in 1971.

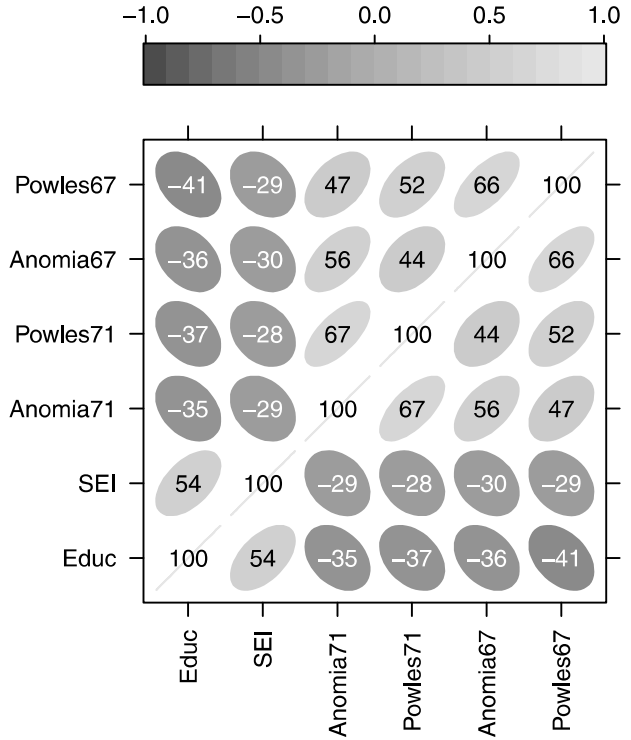

Fig. 7.4. Correlation matrix of alienation data; values given are correlation coefficients $\times 100$.

The scale of the three latent variables, SES, alienation 67, and alienation 71, are arbitrary and have to be fixed in some way to make the model identifiable. Here each is set to the scale of one of its indicator variables by fixing the corresponding regression coefficient to one. Consequently, the equations defining the model to be fitted are as follows:

$$\begin{aligned}
\text{Education} &= \text{SES} + u_1,\\
\text{SEI} &= \lambda_1\text{SES} + u_2,\\
\text{Anomia67} &= \text{Alienation67} + u_3,\\
\text{Powerlessness67} &= \lambda_2\text{Alienation67} + u_4,
\end{aligned}$$

$$\begin{aligned}
\text{Anomia71} &= \text{Alienation71} + u_5,\\
\text{Powerlessness71} &= \lambda_3\text{Alienation71} + u_6,\\
\text{Alienation67} &= \beta_1\text{SES} + u_7,\\
\text{Alienation71} &= \beta_2\text{SES} + \beta_3\text{Alienation67} + u_8.
\end{aligned}$$

In addition to the six regression coefficients in these equations, the model also has to estimate the variances of the eight error terms, $u_1, \ldots, u_8$, and the variance of the error term for the latent variables, SES. The necessary R code for fitting the model is

```
R> alienation_model <- specify.model(
+      file = "alienation_model.txt")
R> alienation_sem <- sem(alienation_model, alienation, 932)
```

where the model reads

```
SES            -> Educ, NA, 1
SES            -> SEI, lambda1, NA
Alienation67   -> Anomia67, NA, 1
Alienation67   -> Powles67, lambda2, NA
Alienation71   -> Anomia71, NA, 1
Alienation71   -> Powles71, lambda3, NA
SES            -> Alienation67, beta1, NA
SES            -> Alienation71, beta2, NA
Alienation67   -> Alienation71, beta3, NA
Educ          <-> Educ, theta1, NA
SEI           <-> SEI, theta2, NA
SES           <-> SES, delta0, NA
Anomia67      <-> Anomia67, theta3, NA
Powles67      <-> Powles67, theta4, NA
Anomia71      <-> Anomia71, theta5, NA
Powles71      <-> Powles71, theta6, NA
Alienation67 <-> Alienation67, delta1, NA
Alienation71 <-> Alienation71, delta2, NA
```

The parameter estimates are

```
R> summary(alienation_sem)

 Model Chisquare =  71.532   Df =  6 Pr(>Chisq) = 1.9829e-13
 Chisquare (null model) =  2131.5   Df =  15
 Goodness-of-fit index =  0.97514
 Adjusted goodness-of-fit index =  0.913
 RMSEA index =  0.10831   90% CI: (0.086636, 0.13150)
 Bentler-Bonnett NFI =  0.96644
 Tucker-Lewis NNFI =  0.9226
 Bentler CFI =  0.96904
```


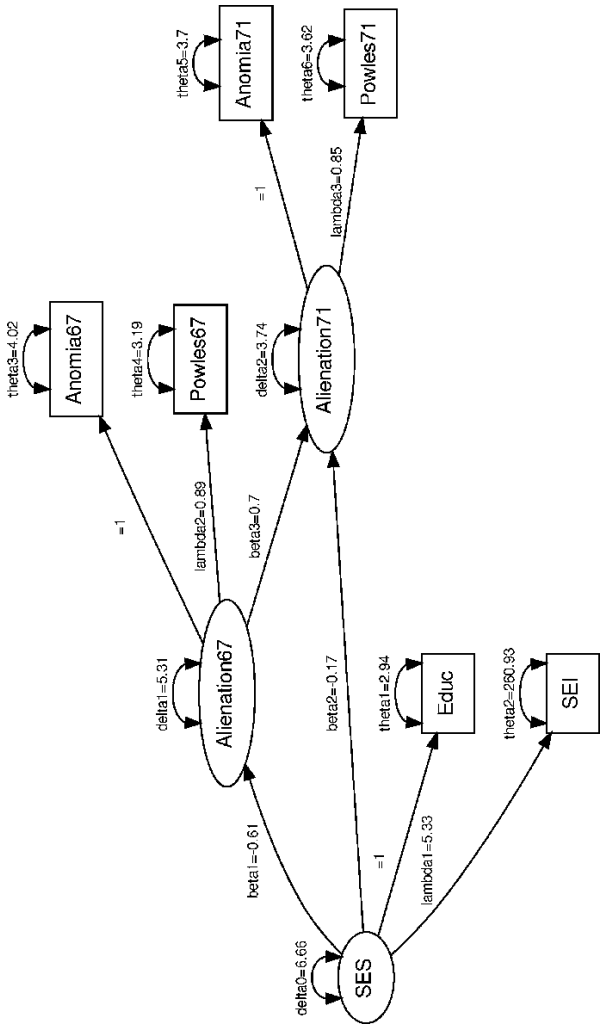

Fig. 7.5. Alienation path diagram.

```
 SRMR =  0.021256
 BIC =  30.508

 Normalized Residuals
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.2600 -0.2090 -0.0001 -0.0151  0.2440  1.3300

 Parameter Estimates
        Estimate  Std Error z value  Pr(>|z|)
lambda1   5.33054  0.430948  12.3693 0.0000e+00
lambda2   0.88883  0.043229  20.5609 0.0000e+00
lambda3   0.84892  0.041567  20.4229 0.0000e+00
beta1    -0.61361  0.056262 -10.9063 0.0000e+00
beta2    -0.17447  0.054221  -3.2178 1.2920e-03
beta3     0.70463  0.053387  13.1984 0.0000e+00
theta1    2.93614  0.500979   5.8608 4.6064e-09
theta2  260.93220 18.275902  14.2774 0.0000e+00
delta0    6.66392  0.641907  10.3814 0.0000e+00
theta3    4.02305  0.359231  11.1990 0.0000e+00
theta4    3.18933  0.284033  11.2288 0.0000e+00
theta5    3.70315  0.391837   9.4508 0.0000e+00
theta6    3.62334  0.304359  11.9048 0.0000e+00
delta1    5.30685  0.484186  10.9603 0.0000e+00
delta2    3.73998  0.388683   9.6222 0.0000e+00

lambda1 SEI <--- SES
lambda2 Powles67 <--- Alienation67
lambda3 Powles71 <--- Alienation71
beta1   Alienation67 <--- SES
beta2   Alienation71 <--- SES
beta3   Alienation71 <--- Alienation67
theta1  Educ <--> Educ
theta2  SEI <--> SEI
delta0  SES <--> SES
theta3  Anomia67 <--> Anomia67
theta4  Powles67 <--> Powles67
theta5  Anomia71 <--> Anomia71
theta6  Powles71 <--> Powles71
delta1  Alienation67 <--> Alienation67
delta2  Alienation71 <--> Alienation71

 Iterations =  84
```

The value of the chi-square fit statistic is 71.532, which with 6 degrees of freedom suggests that the model does not fit well. Jöreskog and Sörbom (1981) suggest that the model can be improved by allowing the measurement errors for anomia in 1967 and in 1971 to be correlated. Fitting such a model in R requires the addition of the following line to the code above:


```
Anomia67 <-> Anomia71, psi, NA
```

The chi-square fit statistic is now 6.359 with 5 degrees of freedom. Clearly the introduction of correlated measurement errors for the two measurements of anomia has greatly improved the fit of the model. However, Bentler (1982), in a discussion of this example, suggests that the importance of the structure remaining to be explained after fitting the original model is in practical terms very small, and Browne (1982) criticises the tendency to allow error terms to become correlated simply to obtain an improvement in fit unless there are sound theoretical reasons why particular error terms should be related.

7.5 Summary

The possibility of making causal inferences about latent variables is one that has great appeal, particularly for social and behavioural scientists, simply because the concepts in which they are most interested are rarely measurable directly. And because such models can nowadays be relatively easily fitted, researchers can routinely investigate quite complex models. But perhaps a caveat issued more than 20 years ago still has some relevance–the following is from Cliff (1983):

> Correlational data are still correlational and no computer program can take account of variables that are not in the analysis. Causal relations can only be established through patient, painstaking attention to all the relevant variables, and should involve active manipulation as a final confirmation.

The maximum likelihood estimation approach described in this chapter is based on the assumption of multivariate normality for the data. When a multivariate normality assumption is clearly untenable, for example with categorical variables, applying the maximum likelihood methods can lead to biased results, although when there are five or more response categories (and the distribution of the data is normal) the problems from disregarding the categorical nature of variables are likely to be minimised (see Marcoulides 2005). Muthen (1984) describes a very general approach for structural equation modelling that can be used when the data consist of a mixture of continuous, ordinal, and dichotomous variables.

Sample size issues for structural equation modelling have been considered by MacCullum, Browne, and Sugawara (1996), and Muthen and Muthen (2002) illustrate how to undertake a Monte Carlo study to help decide on sample size and determine power.

7.6 Exercises

Ex. 7.1 The matrix below shows the correlations between ratings on nine statements about pain made by 123 people suffering from extreme pain. Each statement was scored on a scale from 1 to 6, ranging from agreement to disagreement. The nine pain statements were as follows:

1. Whether or not I am in pain in the future depends on the skills of the doctors.
2. Whenever I am in pain, it is usually because of something I have done or not done.
3. Whether or not I am in pain depends on what the doctors do for me.
4. I cannot get any help for my pain unless I go to seek medical advice.
5. When I am in pain, I know that it is because I have not been taking proper exercise or eating the right food.
6. People's pain results from their own carelessness.
7. I am directly responsible for my pain,
8. relief from pain is chiefly controlled by the doctors.
9. People who are never in pain are just plain lucky.

$$\begin{pmatrix}
1.00 & & & & & & & & \\
-0.04 & 1.00 & & & & & & & \\
0.61 & -0.07 & 1.00 & & & & & & \\
0.45 & -0.12 & 0.59 & 1.00 & & & & & \\
0.03 & 0.49 & 0.03 & -0.08 & 1.00 & & & & \\
-0.29 & 0.43 & -0.13 & -0.21 & 0.47 & 1.00 & & & \\
-0.30 & 0.30 & -0.24 & -0.19 & 0.41 & 0.63 & 1.00 & & \\
0.45 & -0.31 & 0.59 & 0.63 & -0.14 & -0.13 & -0.26 & 1.00 & \\
0.30 & -0.17 & 0.32 & 0.37 & -0.24 & -0.15 & -0.29 & 0.40 & 1.00
\end{pmatrix}.$$

Fit a correlated two-factor model in which questions $1, 3, 4$, and 8 are assumed to be indicators of the latent variable *Doctor's Responsibility* and questions $2, 5, 6$, and 7 are assumed to be indicators of the latent variable *Patient's Responsibility*. Find a 95% confidence interval for the correlation between the two latent variables.

Ex. 7.2 For the stability of alienation example, fit the model in which the measurement errors for anomia in 1967 and anomia in 1971 are allowed to be correlated.

Ex. 7.3 Meyer and Bendig (1961) administered the five Thurstone Primary Mental Ability tests, verbal meaning (V), space (S), reasoning (R), numerical (N), and word fluency (W), to 49 boys and 61 girls in grade 8 and again three and a half years later in grade 11. The observed correlation matrix is shown below. Fit a single-factor model to the correlations that allows the factor at time one to be correlated with the factor at time two.

$$\begin{pmatrix}
 & \text{V} & \text{S} & \text{R} & \text{N} & \text{W} & \text{V} & \text{S} & \text{R} & \text{N} & \text{W} \\
\text{V} & 1.00 & & & & & & & & & \\
\text{S} & 0.37 & 1.00 & & & & & & & & \\
\text{R} & 0.42 & 0.33 & 1.00 & & & & & & & \\
\text{N} & 0.53 & 0.14 & 0.38 & 1.00 & & & & & & \\
\text{W} & 0.38 & 0.10 & 0.20 & 0.24 & 1.00 & & & & & \\
\text{V} & 0.81 & 0.34 & 0.49 & 0.58 & 0.32 & 1.00 & & & & \\
\text{S} & 0.35 & 0.65 & 0.20 & -0.04 & 0.11 & 0.34 & 1.00 & & & \\
\text{R} & 0.42 & 0.32 & 0.75 & 0.46 & 0.26 & 0.46 & 0.18 & 1.00 & & \\
\text{N} & 0.40 & 0.14 & 0.39 & 0.73 & 0.19 & 0.55 & 0.06 & 0.54 & 1.00 & \\
\text{W} & 0.24 & 0.15 & 0.17 & 0.15 & 0.43 & 0.24 & 0.15 & 0.20 & 0.16 & 1.00
\end{pmatrix}$$