CHAPTER 1

Introduction

1 MOTIVATION

Cluster analysis is the art of finding groups in data. To see what is meant by this, let us look at Figure 1. It is a plot of eight objects, on which two variables were measured. For instance, the weight of an object might be displayed on the horizontal axis and its height on the vertical one. Because this example contains only two variables, we can investigate it by merely looking at the plot.

In this small data set there are clearly two distinct groups of objects, namely {TIN, TAL, KIM, ILA} and {LIE, JAC, PET, LEO}. Such groups are called *clusters*, and to discover them is the aim of cluster analysis. Basically, one wants to form groups in such a way that objects in the same group are similar to each other, whereas objects in different groups are as dissimilar as possible.

The classification of similar objects into groups is an important human activity. In everyday life, this is part of the learning process: A child learns to distinguish between cats and dogs, between tables and chairs, between men and women, by means of continuously improving subconscious classification schemes. (This explains why cluster analysis is often considered as a branch of pattern recognition and artificial intelligence.) Classification has always played an essential role in science. In the eighteenth century, Linnaeus and Sauvages provided extensive classifications of animals, plants, minerals, and diseases (for a recent survey, see Holman, 1985). In astronomy, Hertzsprung and Russell classified stars in various categories on the basis of two variables: their light intensity and their surface temperature. In the social sciences, one frequently classifies people with regard to their behavior and preferences. In marketing, it is often attempted to identify market segments, that is, groups of customers with similar needs. Many more examples could be given in geography (clustering of regions), medicine

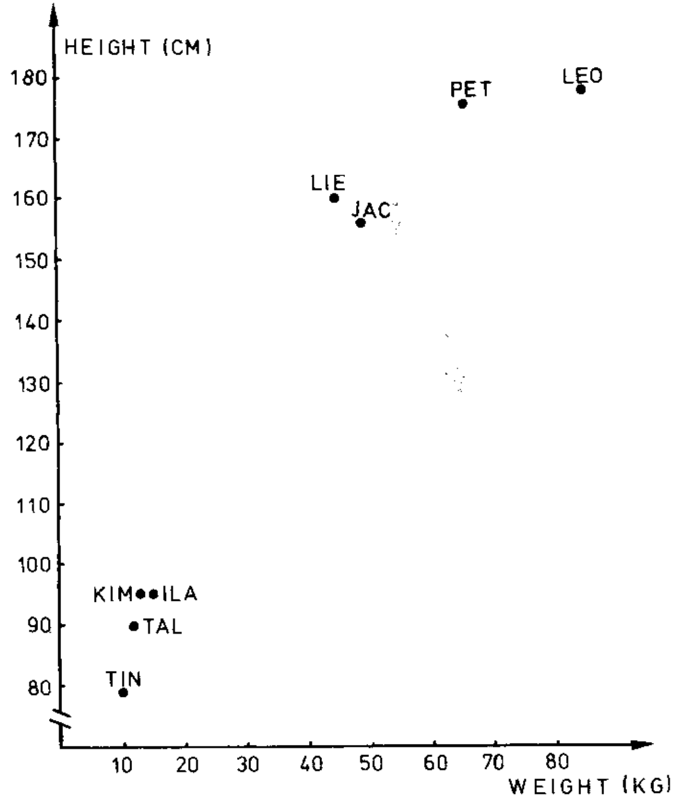

Figure 1 A plot of eight objects.

(incidence of specific types of cancer), chemistry (classification of compounds), history (grouping of archeological findings), and so on. Moreover, cluster analysis can be used not only to identify a structure already present in the data, but also to impose a structure on a more or less homogeneous data set that has to be split up in a "fair" way, for instance when dividing a country into telephone areas. Note that cluster analysis is quite different from discriminant analysis in that it actually establishes the groups, whereas discriminant analysis assigns objects to groups that were defined in advance.

In the past, clusterings were usually performed in a subjective way, by relying on the perception and judgment of the researcher. In the example of Figure 1, we used the human eye–brain system which is very well suited (through millenia of evolution) for classification in up to three dimensions.

However, the need to classify cases in more than three dimensions and the upcoming objectivity standards of modern science have given rise to so-called automatic classification procedures. Over the last 30 years, a wealth of algorithms and computer programs has been developed for cluster analysis. The reasons for this variety of methods are probably twofold. To begin with, automatic classification is a very young scientific discipline in vigorous development, as can be seen from the thousands of articles scattered over many periodicals (mostly journals of statistics, biology, psychometrics, computer science, and marketing). Nowadays, automatic classification is establishing itself as an independent scientific discipline, as witnessed by a full-fledged periodical (the *Journal of Classification*, first published in 1984) and the International Federation of Classification Societies (founded in 1985). The second main reason for the diversity of algorithms is that there exists no general definition of a cluster, and in fact there are several kinds of them: spherical clusters, drawn-out clusters, linear clusters, and so on. Moreover, different applications make use of different data types, such as continuous variables, discrete variables, similarities, and dissimilarities. Therefore, one needs different clustering methods in order to adapt to the kind of application and the type of clusters sought. Cluster analysis has become known under a variety of names, such as numerical taxonomy, automatic classification, botryology (Good, 1977), and typological analysis (Chandon and Pinson, 1981).

In this book, several algorithms are provided for transforming the data, for performing cluster analysis, and for displaying the results graphically. Section 2 of this introduction discusses the various types of data and what to do with them, and Section 3 gives a brief survey of the clustering methods contained in the book, with some guidelines as to which algorithm to choose. In particular the crucial distinction between partitioning and hierarchical methods is considered. In Section 4 a schematic overview is presented. In Section 5, it is explained how to use the program DAISY to transform your data.

2 TYPES OF DATA AND HOW TO HANDLE THEM

Our first objective is to study some types of data which typically occur and to investigate ways of processing the data to make them suitable for cluster analysis.

Suppose there are n objects to be clustered, which may be persons, flowers, words, countries, or whatever. Clustering algorithms typically operate on either of two input structures. The first represents the objects by means of p measurements or attributes, such as height, weight, sex, color, and so on. These measurements can be arranged in an n-by-p matrix, where the rows correspond to the objects and the columns to the attributes. In Tucker's (1964) terminology such an objects-by-variables matrix is said to be *two-mode*, since the row and column entities are different. The second structure is a collection of proximities that must be available for all pairs of objects. These proximities make up an n-by-n table, which is called a *one-mode* matrix because the row and column entities are the same set of objects. We shall consider two types of proximities, namely dissimilarities (which measure how far away two objects are from each other) and similarities (which measure how much they resemble each other). Let us now have a closer look at the types of data used in this book by considering them one by one.

2.1 Interval-Scaled Variables

In this situation the n objects are characterized by p *continuous* measurements. These values are positive or negative real numbers, such as height, weight, temperature, age, cost, . . . , which follow a linear scale. For instance, the time interval between 1905 and 1915 was equal in length to that between 1967 and 1977. Also, it takes the same amount of energy to heat an object of −16.4°C to −12.4°C as to bring it from 35.2°C to 39.2°C. In general it is required that intervals keep the same importance throughout the scale.

These measurements can be organized in an n-by-p matrix, where the rows correspond to the objects (or cases) and the columns correspond to the variables. When the fth measurement of the ith object is denoted by x_{if} (where $i = 1, \dots, n$ and $f = 1, \dots, p$) this matrix looks like

$$\begin{array}{cc} & p \text{ variables} \\ n \text{ objects} & \begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \vdots & & \vdots & & \vdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \vdots & & \vdots & & \vdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix} \end{array} \qquad (1)$$

For instance, consider the following real data set. For eight people, the weight (in kilograms) and the height (in centimeters) is recorded in Table 1. In this situation, $n = 8$ and $p = 2$. One could have recorded many more variables, like age and blood pressure, but as there are only two variables in this example it is easy to make a scatterplot, which corresponds to Figure 1 given earlier. Note that the units on the vertical axis are drawn to the same size as those on the horizontal axis, even though they represent different physical concepts. The plot contains two obvious clusters, which can in this case be interpreted easily: the one consists of small children and the other of adults.

Table 1 Weight and Height of Eight People, Expressed in Kilograms and Centimeters

Name	Weight (kg)	Height (cm)
Ilan	15	95
Jacqueline	49	156
Kim	13	95
Lieve	45	160
Leon	85	178
Peter	66	176
Talia	12	90
Tina	10	78

Note that other variables might have led to completely different clusterings. For instance, measuring the concentration of certain natural hormones might have yielded a clear-cut partition into three male and five female persons. By choosing still other variables, one might have found blood types, skin types, or many other classifications.

Let us now consider the effect of changing measurement units. If weight and height of the subjects had been expressed in pounds and inches, the results would have looked quite different. A pound equals 0.4536 kg and an inch is 2.54 cm. Therefore, Table 2 contains larger numbers in the column of weights and smaller numbers in the column of heights. Figure 2, although plotting essentially the same data as Figure 1, looks much flatter. In this figure, the relative importance of the variable "weight" is much larger than in Figure 1. (Note that Kim is closer to Ilan than to Talia in Figure 1, but that she is closer to Talia than to Ilan in Figure 2.) As a consequence, the two clusters are not as nicely separated as in Figure 1 because in this particular example the height of a person gives a better indication of adulthood than his or her weight. If height had been expressed in feet (1 ft = 30.48 cm), the plot would become flatter still and the variable "weight" would be rather dominant.

In some applications, changing the measurement units may even lead one to see a very different clustering structure. For example, the age (in years) and height (in centimeters) of four imaginary people are given in

Table 2 Weight and Height of the Same Eight People, But Now Expressed in Pounds and Inches

Name	Weight (lb)	Height (in.)
Ilan	33.1	37.4
Jacqueline	108.0	61.4
Kim	28.7	37.4
Lieve	99.2	63.0
Leon	187.4	70.0
Peter	145.5	69.3
Talia	26.5	35.4
Tina	22.0	30.7

Table 3 and plotted in Figure 3. It appears that $\{A, B\}$ and $\{C, D\}$ are two well-separated clusters. On the other hand, when height is expressed in feet one obtains Table 4 and Figure 4, where the obvious clusters are now $\{A, C\}$ and $\{B, D\}$. This partition is completely different from the first because each subject has received another companion. (Figure 4 would have been flattened even more if age had been measured in days.)

To avoid this dependence on the choice of measurement units, one has the option of standardizing the data. This converts the original measurements to unitless variables. First one calculates the mean value of variable

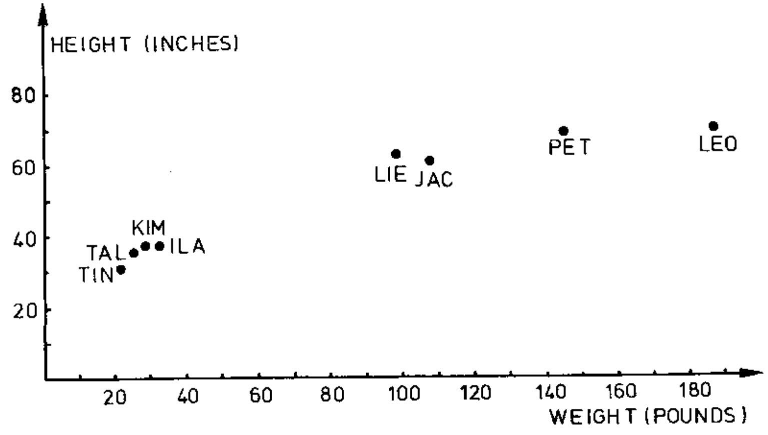

Figure 2 Plot corresponding to Table 2.

Table 3 Age (in years) and Height (in centimeters) of Four People

Person	Age (yr)	Height (cm)
A	35	190
B	40	190
C	35	160
D	40	160

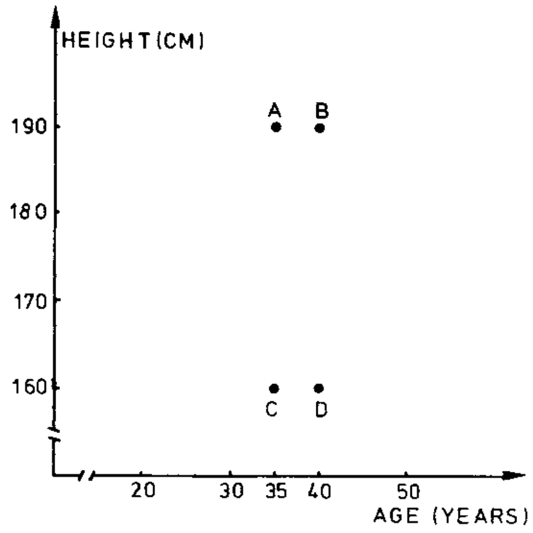

Figure 3 Plot of height (in centimeters) versus age for four people.

Table 4 Age (in years) and Height (in feet) of the Same Four People

Person	Age (yr)	Height (ft)
A	35	6.2
B	40	6.2
C	35	5.2
D	40	5.2

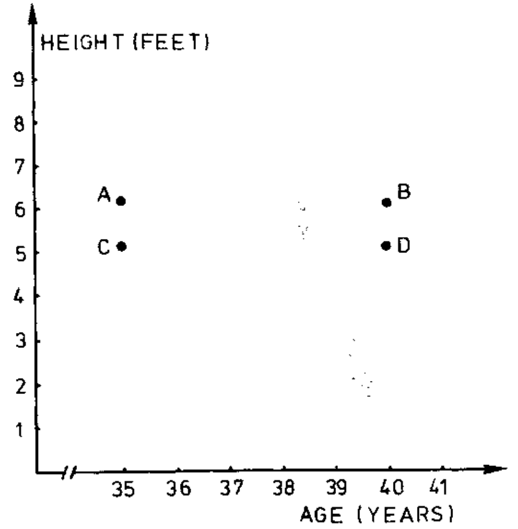

Figure 4 Plot of height (in feet) versus age for the same four people.

f, given by

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + \cdots + x_{nf}) \tag{2}$$

for each $f = 1, \ldots, p$. Then one computes a measure of the dispersion or "spread" of this fth variable. Traditionally, people use the standard deviation

$$\mathrm{std}_f = \sqrt{\frac{1}{n-1}\{(x_{1f} - m_f)^2 + (x_{2f} - m_f)^2 + \cdots + (x_{nf} - m_f)^2\}}$$

for this purpose. However, this measure is affected very much by the presence of outlying values. For instance, suppose that one of the x_{if} has been wrongly recorded, so that it is much too large. In this case std_f will be unduly inflated, because $x_{if} - m_f$ is squared. Hartigan (1975, p. 299) notes that one needs a dispersion measure that is not too sensitive to outliers. Therefore, from now on we will use the *mean absolute deviation*

$$s_f = \frac{1}{n}\{|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots + |x_{nf} - m_f|\} \tag{3}$$

where the contribution of each measurement x_{if} is proportional to the absolute value $|x_{if} - m_f|$. This measure is more robust (see, e.g., Hampel et al., 1986) in the sense that one outlying observation will not have such a large influence on s_f. (Note that there exist estimates that are much more robust, but we would like to avoid a digression into the field of robust statistics.)

Let us assume that s_f is nonzero (otherwise variable f is constant over all objects and must be removed). Then the standardized measurements are defined by

$$z_{if} = \frac{x_{if} - m_f}{s_f} \tag{4}$$

and sometimes called *z-scores*. They are unitless because both the numerator and the denominator are expressed in the same units. By construction, the z_{if} have mean value zero and their mean absolute deviation is equal to 1. When applying standardization, one forgets about the original data (1) and uses the new data matrix

$$\begin{array}{cc} & \text{variables} \\ \text{objects} & \begin{bmatrix} z_{11} & \cdots & z_{1f} & \cdots & z_{1p} \\ \vdots & & \vdots & & \vdots \\ z_{i1} & \cdots & z_{if} & \cdots & z_{ip} \\ \vdots & & \vdots & & \vdots \\ z_{n1} & \cdots & z_{nf} & \cdots & z_{np} \end{bmatrix} \end{array} \tag{5}$$

in all subsequent computations. The advantage of using s_f rather than std_f in the denominator of (4) is that s_f will not be blown up so much in the case of an outlying x_{if}, and hence the corresponding z_{if} will still "stick out" so the ith object can be recognized as an outlier by the clustering algorithm, which will typically put it in a separate cluster. [This motivation differs from that of Milligan and Cooper (1988), who recommended the use of a nonrobust denominator.]

The preceding description might convey the impression that standardization would be beneficial in all situations. However, it is merely an option that may or may not be useful in a given application. Sometimes the variables have an absolute meaning, and should not be standardized (for instance, it may happen that several variables are expressed in the same units, so they should not be divided by different s_f). Often standardization dampens a clustering structure by reducing the large effects because the variables with a big contribution are divided by a large s_f.

For example, let us standardize the data of Table 3. The mean age equals $m_1 = 37.5$ and the mean absolute deviation of the first variable works out to be $s_1 = \{2.5 + 2.5 + 2.5 + 2.5\}/4 = 2.5$. Therefore, standardization converts age 40 to $+1$ and age 35 to -1. Analogously, $m_2 = 175$ cm and

Table 5 Standardized Age and Height of the Same Four People

Person	Variable 1	Variable 2
A	−1.0	1.0
B	1.0	1.0
C	−1.0	−1.0
D	1.0	−1.0

$s_2 = \{15 + 15 + 15 + 15\}/4 = 15$ cm, so 190 cm is standardized to $+1$ and 160 cm to -1. The resulting data matrix, which is unitless, is given in Table 5. Note that the new averages are zero and that the mean deviations equal 1. Of course, standardizing Table 4 would yield exactly the same result, so Table 5 is the standardized version of both Tables 3 and 4. Even when the data are converted to very strange units (such as the proverbial fortnights and furlongs), standardization will always yield the same numbers. However, plotting the values of Table 5 in Figure 5 does not give a very exciting result. Figure 5 looks like an intermediate between Figures 3 and 4 and shows no clustering structure because the four points lie at the vertices of a square. One could say that there are four clusters, each consisting of a single point, or that there is only one big cluster containing four points.

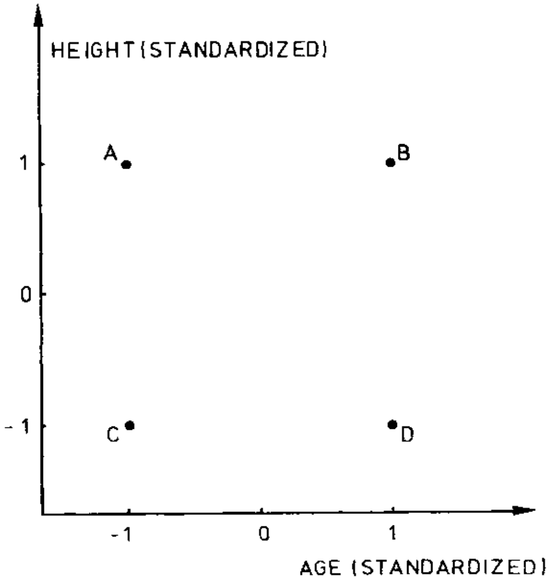

Figure 5 Standardized height versus standardized age.

From a philosophical point of view, standardization does not really solve the problem. Indeed, the choice of measurement units gives rise to relative *weights* of the variables. Expressing a variable in smaller units will lead to a larger range for that variable, which will then have a large effect on the resulting structure. On the other hand, by standardizing one attempts to give all variables an equal weight, in the hope of achieving objectivity. As such, it may be used by a practitioner who possesses no prior knowledge. However, it may well be that some variables are intrinsically more important than others in a particular application, and then the assignment of weights should be based on subject-matter knowledge (see, e.g., Abrahamowicz, 1985). On the other hand, there have been attempts to devise clustering techniques that are independent of the scale of the variables (Friedman and Rubin, 1967). The proposal of Hardy and Rasson (1982) is to search for a partition that minimizes the total volume of the convex hulls of the clusters. In principle such a method is invariant with respect to linear transformations of the data, but unfortunately no algorithm exists for its implementation (except for an approximation that is restricted to two dimensions). Therefore, the dilemma of standardization appears unavoidable at present and the programs described in this book leave the choice up to the user.

Of course, the data offered to the program may already be the result of some transformation: Often people find it useful to replace some variable by its inverse or its square, which may be more meaningful in that particular context. However, we shall assume from now on that such transformations have been performed prior to the cluster analysis.

The next step is to compute distances between the objects, in order to quantify their degree of dissimilarity. It is necessary to have a distance for each pair of objects i and j. The most popular choice is the *Euclidean distance*

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2} \qquad (6)$$

(When the data are being standardized, one has to replace all x by z in this expression.) Formula (6) corresponds to the true geometrical distance between the points with coordinates $(x_{i1}, \dots, x_{ip})$ and $(x_{j1}, \dots, x_{jp})$. To illustrate this, let us consider the special case with $p = 2$. Figure 6 shows two points with coordinates (x_{i1}, x_{i2}) and (x_{j1}, x_{j2}). It is clear that the actual distance between objects i and j is given by the length of the hypothenuse of the triangle, yielding expression (6) by virtue of Pythagoras' theorem. For this reason, Gower (1971b) calls (6) the *Pythagorean distance*.

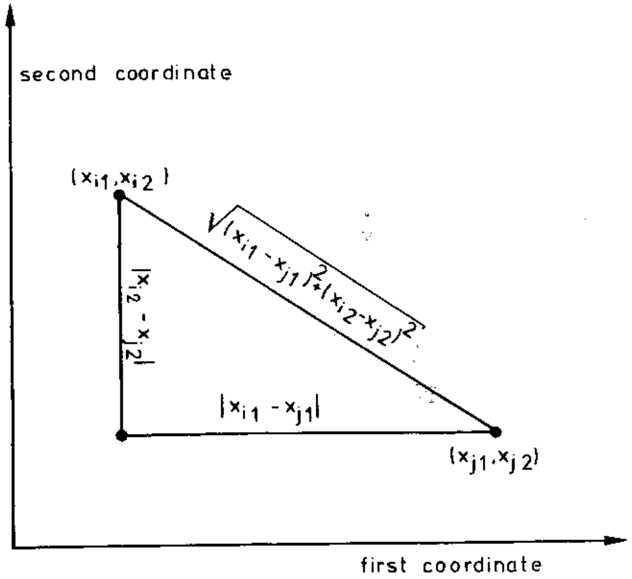

Figure 6 Illustration of the Euclidean distance formula.

Another well-known metric is the *city block* or *Manhattan distance*, defined by

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}| \qquad (7)$$

In Figure 6, this corresponds to the sum of the lengths of the other two sides of the triangle. The Manhattan distance was used in a cluster analysis context by Carmichael and Sneath (1969) and owes its peculiar name to the following reasoning. Suppose you live in a city where the streets are all north–south or east–west, and hence perpendicular to each other. Let Figure 6 be part of a street map of such a city, where the streets are portrayed as vertical and horizontal lines. Then the actual distance you would have to travel by car to get from location i to location j would total $|x_{i1} - x_{j1}| + |x_{i2} - x_{j2}|$, corresponding to (7). This would be the shortest length among all possible paths from i to j. Only a bird could fly straight from point i to point j, thereby covering the Euclidean distance between these points. The use of the Manhattan distance is advised in those situations where for example a difference of 1 in the first variable and of 3 in the second variable is the same as a difference of 2 in the first variable and of 2 in the second.

Both the Euclidean metric (6) and the Manhattan metric (7) satisfy the following mathematical requirements of a distance function:

(D1) $d(i, j) \geq 0$
(D2) $d(i, i) = 0$
(D3) $d(i, j) = d(j, i)$
(D4) $d(i, j) \leq d(i, h) + d(h, j)$

for all objects i, j, and h. Condition (D1) merely states that distances are nonnegative numbers and (D2) says that the distance of an object to itself is zero. Axiom (D3) is the symmetry of the distance function. The *triangle inequality* (D4) looks a little bit more complicated, but is necessary to allow a geometrical interpretation. It says essentially that going directly from i to j is shorter than making a detour over object h.

Note that $d(i, j) = 0$ does not necessarily imply that $i = j$, because it can very well happen that two different objects have the same measurements for the variables under study. However, the triangle inequality implies that i and j will then have the same distance to any other object h, because $d(i, h) \leq d(i, j) + d(j, h) = d(j, h)$ and at the same time $d(j, h) \leq d(j, i) + d(i, h) = d(i, h)$, which together imply that $d(i, h) = d(j, h)$.

A generalization of both the Euclidean and the Manhattan metric is the *Minkowski distance* given by

$$d(i, j) = \left(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \cdots + |x_{ip} - x_{jp}|^q\right)^{1/q}$$

where q is any real number larger than or equal to 1. This is also called the L_q metric, with the Euclidean ($q = 2$) and the Manhattan ($q = 1$) as special cases. Many other distance functions may be constructed (see, e.g., Bock, 1974, Section 3; Hartigan, 1975, Chapter 2; Romesburg, 1984, Chapter 8). The clustering programs accompanying this book provide a choice between Euclidean and Manhattan distances.

One sometimes computes weighted Euclidean distances like

$$d(i, j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \cdots + w_p(x_{ip} - x_{jp})^2} \qquad (8)$$

where each variable receives a weight according to its perceived importance. For instance, giving a variable weight 2 is the same thing as using it twice. However, applying such weighted distances on the raw data is equivalent to first choosing other measurement units, corresponding to rescaling the coordinates by the factors $\sqrt{w_1}, \ldots, \sqrt{w_p}$, and then computing ordinary distances. (Therefore, it was not necessary to provide weighted distances in our programs.) This leads us back to the discussion on standardization. The essential question remains: Which weights should be assigned to the variables? If one thinks the measurement units were not particularly well chosen and one wants to assign equal weights to the variables, it is preferable to standardize the data first and then to compute ordinary Euclidean distances (6). But if one wants to keep the data intact, because the measurement scales are believed to be meaningful, it is best not to standardize (and hence to use the weights inherent in the raw data). Furthermore, if one wants to *impose* certain weights on the variables, due to prior beliefs or background information, one can either change the measurement units or apply weighted distances like (8), which boils down to the same thing.

In all this, it should be noted that a variable not containing any relevant information (say, the telephone number of each person) is worse than useless, because it will make the clustering less apparent. The occurrence of several such "trash variables" will kill the whole clustering because they yield a lot of random terms in the distances, thereby hiding the useful information provided by the other variables. Therefore, such noninformative variables must be given a zero weight in the analysis, which amounts to deleting them. A recent discussion of variable selection can be found in Fowlkes et al. (1988). In general, the selection of "good" variables is a nontrivial task and may involve quite some trial and error (in addition to subject-matter knowledge and common sense). In this respect, cluster analysis may be considered an exploratory technique.

It often happens that not all measurements are actually available, so there are some "holes" in the data matrix (1). Such an absent measurement is called a *missing value* and it may have several causes. The value of the measurement may have been lost or it may not have been recorded at all by oversight or lack of time. Sometimes the information is simply not available, as in the example of the birthdate of a foundling, or the patient may not remember whether he or she ever had the measles, or it may be impossible to measure the desired quantity due to the malfunctioning of some instrument. In certain instances the question does not apply (such as the color of hair of a bald person) or there may be more than one possible answer (when two experimenters obtain very different results). Because of all this, missing values are often encountered.

How can we handle a data set with missing values? In the matrix (1) we indicate the absent measurements by means of some code (like the number 999.99, if it did not already occur), that can then be recognized by the program. If there exists an object in the data set for which all measurements are missing, there is really no information on this object so it has to be deleted. Analogously, a variable consisting exclusively of missing values has to be removed too.

If the data are standardized, the mean value m_f of the fth variable is calculated by making use of the present values only. The same goes for s_f, so in the denominator of (2) and (3) we must replace n by the number of nonmissing values for that variable. The z-scores z_{if} can then be computed as in (4), but of course only when the corresponding x_{if} is not missing itself.

In the computation of distances (based on either the x_{if} or the z_{if}) similar precautions must be taken. When calculating the distances $d(i, j)$ given by (6) or (7), only those variables are considered in the sum for which the measurements for both objects are present; subsequently the sum is multiplied by p and divided by the actual number of terms (in the case of Euclidean distances this is done before taking the square root). Obviously, such a procedure only makes sense when the variables are thought of as having the same weight (for instance, this can be done after standardization). When computing these distances, one might come across a pair of objects that do not have any common measured variables, so their distance cannot be computed by means of the abovementioned approach. Several remedies are possible: One could remove either object or one could fill in some average distance value based on the rest of the data. A totally different approach consists of replacing all missing x_{if} by the mean m_f of that variable; then all distances can be computed. Applying any of these methods, one finally possesses a "full" set of distances. From this point on, many clustering algorithms can be applied, even though the original data set was not complete.

In any case, we now have a collection of distances (whether based on raw or standardized data that contained missing values or not) that we want to store in a systematic way. This can be achieved by arranging them in an n-by-n matrix. For example, when computing Euclidean distances between the objects of Table 1 we obtain

$$\begin{array}{c|cccccccc|}
 & \text{ILA} & \text{JAC} & \text{KIM} & \text{LIE} & \text{LEO} & \text{PET} & \text{TAL} & \text{TIN} \\
\text{ILA} & 0 & 69.8 & 2.0 & 71.6 & 108.6 & 95.7 & 5.8 & 17.7 \\
\text{JAC} & 69.8 & 0 & 70.8 & 5.7 & 42.2 & 26.3 & 75.7 & 87.2 \\
\text{KIM} & 2.0 & 70.8 & 0 & 72.5 & 109.9 & 96.8 & 5.1 & 17.3 \\
\text{LIE} & 71.6 & 5.7 & 72.5 & 0 & 43.9 & 26.4 & 77.4 & 89.2 \\
\text{LEO} & 108.6 & 42.2 & 109.9 & 43.9 & 0 & 19.1 & 114.3 & 125.0 \\
\text{PET} & 95.7 & 26.3 & 96.8 & 26.4 & 19.1 & 0 & 101.6 & 112.9 \\
\text{TAL} & 5.8 & 75.7 & 5.1 & 77.4 & 114.3 & 101.6 & 0 & 12.2 \\
\text{TIN} & 17.7 & 87.2 & 17.3 & 89.2 & 125.0 & 112.9 & 12.2 & 0
\end{array} \tag{9}$$

The distance between object JAC and object LEO can be located at the intersection of the fifth row and the second column, yielding 42.2. The same number can also be found at the intersection of the second row and the fifth column, because the distance between JAC and LEO is equal to the distance between LEO and JAC. This is the symmetry property (D3), which holds for any pair of objects [formula (6) gives the same result when i and j are interchanged]. Therefore, a distance matrix is always symmetric. Moreover, note that the entries on the main diagonal are always zero, because the distance of an object to itself has to be zero. (The same remarks apply to the Manhattan or any other distance.) Therefore, it would suffice to write down only the lower triangular half of the distance matrix, which looks like

$$\begin{array}{lrrrrrrr}
 & \text{ILA} & \text{JAC} & \text{KIM} & \text{LIE} & \text{LEO} & \text{PET} & \text{TAL} \\
\text{JAC} & 69.8 & & & & & & \\
\text{KIM} & 2.0 & 70.8 & & & & & \\
\text{LIE} & 71.6 & 5.7 & 72.5 & & & & \\
\text{LEO} & 108.6 & 42.2 & 109.9 & 43.9 & & & \\
\text{PET} & 95.7 & 26.3 & 96.8 & 26.4 & 19.1 & & \\
\text{TAL} & 5.8 & 75.7 & 5.1 & 77.4 & 114.3 & 101.6 & \\
\text{TIN} & 17.7 & 87.2 & 17.3 & 89.2 & 125.0 & 112.9 & 12.2
\end{array} \tag{10}$$

Note that in the latter form there are only seven rows and seven columns, because the upper row and the rightmost column of (9) were superfluous.

When the data are represented as in (9) or (10), the cluster structure we saw so easily in Figure 1 is rather hidden from visual inspection. Nevertheless, the clustering methods discussed in Chapters 2, 4, 5, and 6 only make use of this information, without having to return to the original data matrix.

2.2 Dissimilarities

This leads us to our second input data structure, namely an n-by-n matrix like (9), often presented as in (10). The entries of such a matrix may be Euclidean or Manhattan distances. However, there are many other possibilities, so we no longer speak of distances but of *dissimilarities* (or dissimilarity coefficients). Basically, dissimilarities are nonnegative numbers $d(i, j)$ that are small (close to zero) when i and j are "near" to each other and that become large when i and j are very different. We shall usually assume that dissimilarities are symmetric and that the dissimilarity of an object to itself is zero, but in general the triangle inequality does *not* hold. Indeed, it is often assumed that dissimilarities satisfy (D1), (D2), and (D3) (see, e.g., Bock, 1974, p. 25), although none of these properties is really essential and there are clustering methods that do not require any of them. But the main difference with distances is that (D4) can no longer be relied on.

Table 6 Subjective Dissimilarities between 11 Sciences

Astronomy	0.00										
Biology	7.86	0.00									
Chemistry	6.50	2.93	0.00								
Computer sci.	5.00	6.86	6.50	0.00							
Economics	8.00	8.14	8.21	4.79	0.00						
Geography	4.29	7.00	7.64	7.71	5.93	0.00					
History	8.07	8.14	8.71	8.57	5.86	3.86	0.00				
Mathematics	3.64	7.14	4.43	1.43	3.57	7.07	9.07	0.00			
Medicine	8.21	2.50	2.93	6.36	8.43	7.86	8.43	6.29	0.00		
Physics	2.71	5.21	4.57	4.21	8.36	7.29	8.64	2.21	5.07	0.00	
Psychology	9.36	5.57	7.29	7.21	6.86	8.29	7.64	8.71	3.79	8.64	0.00

Dissimilarities can be obtained in several ways. Often they can be computed from variables that are binary, nominal, ordinal, interval, or a combination of these (a description of such variables and possible formulas will be given later in this chapter). Also, dissimilarities can be simple subjective ratings of how much certain objects differ from each other, from the point of view of one or more observers. This kind of data is typical in the social sciences and in marketing.

Let us consider an example of this type. Fourteen postgraduate economics students (coming from different parts of the world) were asked to indicate the subjective dissimilarities between 11 scientific disciplines. All of them had to fill in a matrix like Table 6, where the dissimilarities had to be given as integer numbers on a scale from 0 (identical) to 10 (very different). The actual entries of Table 6 are the averages of the values given by the students. It appears that the smallest dissimilarity is perceived between mathematics and computer science, whereas the most remote fields were psychology and astronomy.

Another example of the construction of dissimilarities is to record how often consonants are misunderstood, because when two consonants are often confused (like "s" and "z") this indicates that their dissimilarity is small [see, e.g., Johnson's (1967) analysis of the Miller and Nicely (1955) data]. Such experiments lead to asymmetric matrices, because "z" may be more often inadvertently taken for "s" than vice versa. However, in such situations one can easily symmetrize the data [for instance by averaging $d(i, j)$ with $d(j, i)$ for each pair of consonants].

If one wants to perform a cluster analysis on a set of *variables* that have been observed in some population, there are other measures of dissimilarity. For instance, one can compute the (parametric) Pearson product-moment correlation

$$R(f, g) = \frac{\Sigma_{i=1}^{n}(x_{if} - m_f)(x_{ig} - m_g)}{\sqrt{\Sigma_{i=1}^{n}(x_{if} - m_f)^2}\sqrt{\Sigma_{i=1}^{n}(x_{ig} - m_g)^2}}$$

between the variables f and g, or alternatively the (nonparametric) Spearman correlation. Both coefficients lie between -1 and $+1$ and do not depend on the choice of measurement units. The main difference between them is that the Pearson coefficient looks for a linear relation between the variables f and g, whereas the Spearman coefficient searches for a monotone relation. Both coefficients are provided by most statistical packages, like SPSS, BMDP, or SAS, so they can simply be taken from their routine output. Correlation coefficients are useful for clustering purposes because they measure the extent to which two variables are related.

For instance, the Pearson correlation between the variables weight and height in Table 1 is 0.957. It is very high because there appears to be a positive relationship between these two variables: The larger somebody's weight, the larger his or her height is likely to be, as can be seen from the upward trend in Figure 1. Table 7 also lists some other variables measured on the same eight people, namely their month and year of birth. We see no apparent relation between month of birth and weight: There is no obvious reason why someone born in November (of any year) would be likely to be heavier than someone born in February. Indeed, the correlation between month and weight is approximately zero (the actual value in this example is -0.036). A third situation occurs when we correlate weight with the year of birth: The people with a large birth year will typically possess a smaller weight and vice versa. In such a situation the correlation coefficient becomes strongly negative (in this example it is -0.953, which is close to -1, because the relation is nearly linear). Continuing like this, we can fill up Table 8(a).

Table 7 Data on Eight People. Weight is Expressed in Kilograms and Height in Centimeters. Also the Month and Year of Birth are Provided

Name	Weight	Height	Month	Year
Ilan	15	95	1	82
Jacqueline	49	156	5	55
Kim	13	95	11	81
Lieve	45	160	7	56
Leon	85	178	6	48
Peter	66	176	6	56
Talia	12	90	12	83
Tina	10	78	1	84

Table 8 (a) Pearson Correlation Coefficients between the Four Variables in Table 7, (b) Corresponding Dissimilarities Obtained Through Formula (11), and (c) Dissimilarities Computed by Means of (12)

Quantity		Weight	Height	Month	Year
(a) Correlations	Weight	1.000			
	Height	0.957	1.000		
	Month	−0.036	0.021	1.000	
	Year	−0.953	−0.985	0.013	1.000
(b) Dissimilarities According to (11)	Weight	0.000			
	Height	0.021	0.000		
	Month	0.518	0.489	0.000	
	Year	0.977	0.992	0.493	0.000
(c) Dissimilarities According to (12)	Weight	0.000			
	Height	0.043	0.000		
	Month	0.964	0.979	0.000	
	Year	0.047	0.015	0.987	0.000

Correlation coefficients, whether parametric or nonparametric, can be converted to dissimilarities $d(f, g)$, for instance by setting

$$d(f, g) = (1 - R(f, g))/2 \tag{11}$$

With this formula, variables with a high positive correlation receive a dissimilarity coefficient close to zero, whereas variables with a strongly negative correlation will be considered very dissimilar. In other applications one might prefer to use

$$d(f, g) = 1 - |R(f, g)| \tag{12}$$

in which case also variables with a strongly negative correlation will be assigned a small dissimilarity. Lance and Williams (1979) compared these formulas by means of real data, and concluded that (11) was unequivocally the best, whereas (12) still did relatively well. (A third possibility, given by $d(f, g) = 1 - R(f, g)^2$, turned out to be uniformly unsatisfactory.) Table 8(b) contains the dissimilarities computed according to (11), in which case weight and year are perceived to be very different. On the other hand, the use of (12) yields Table 8(c) in which the variable year joins the cluster formed by weight and height.

Many other ad hoc dissimilarities between variables can be thought of. For example, in a psychological application (Lecompte et al., 1986) we once had to cluster nominal variables, some of which possessed two classes and some three. The resulting contingency tables of pairs of variables led to chi-squared statistics that could not be compared directly because they possessed different degrees of freedom. However, the computed significance level (also called P-value) of these statistics could be used to construct a dissimilarity measure. The stronger the relationship between two variables, the smaller their P-value becomes.

In many applications, the input data simply consist of a dissimilarity matrix, without any measurement values. Indeed, the dissimilarities may have been computed from attributes that were not published or even have been lost. It may also be that there never were any variables in the first place, because the dissimilarities were obtained in another way (from subjective assessments, confusion data, or whatever). For this reason it is useful to have clustering algorithms that can operate directly on a dissimilarity matrix, without having to resort to any measurements. This is the case for the programs PAM, FANNY, AGNES, and DIANA, which will be briefly introduced in Section 3.

2.3 Similarities

Instead of using a dissimilarity coefficient $d(i, j)$ to indicate how remote two objects i and j are, it is also possible to work with a *similarity coefficient* $s(i, j)$. The more objects i and j are alike (or close), the larger $s(i, j)$ becomes. Such a similarity $s(i, j)$ typically takes on values between 0 and 1, where 0 means that i and j are not similar at all and 1 reflects maximal similarity. Values in between 0 and 1 indicate various degrees of resemblance. Often it is assumed that the following conditions hold:

(S1) $0 \leq s(i, j) \leq 1$
(S2) $s(i, i) = 1$
(S3) $s(i, j) = s(j, i)$

for all objects i and j (see Bock, 1974). The numbers $s(i, j)$ can be arranged in an n-by-n matrix like (9) or (10), which is then called a *similarity matrix*. Both similarity and dissimilarity matrices are generally referred to as *proximity matrices*, or sometimes as *resemblance matrices*.

Similarities may arise in several ways. Like dissimilarities, they may be the results of subjective judgments. Also, there are formulas to compute similarities between objects characterized by attributes, even when these variables are of different types, as we shall see in Section 2.6 on mixed measurements.

In order to define similarities between *variables*, we can again resort to the Pearson or the Spearman correlation coefficient. However, neither correlation measure can be used directly as a similarity coefficient because they also take on negative values. Some transformation is in order to bring the coefficients into the zero-one range. There are essentially two ways to do this, depending on the meaning of the data and the purpose of the application. If variables with a strong negative correlation are considered to be very different because they are oriented in the opposite direction (like mileage and weight of a set of cars), then it is best to take something like

$$s(f, g) = (1 + R(f, g))/2 \tag{13}$$

which yields $s(f, g) = 0$ whenever $R(f, g) = -1$. On the other hand, there are situations in which variables with a strong negative correlation should be grouped, because they measure essentially the same thing. (For instance, this happens if one wants to reduce the number of variables in a regression data set by selecting one variable from each cluster.) In that case it is better to use a formula like

$$s(f, g) = |R(f, g)| \tag{14}$$

which yields $s(f, g) = 1$ when $R(f, g) = -1$.

It must be noted that people have sometimes used correlation coefficients for assessing similarity between *objects* by simply interchanging the roles of objects and variables in the expression of R. This does not make much sense because it involves such operations as averaging the measurements (in different units) of the same object. The use of the correlation coefficient between objects was criticized by Eades (1965), Fleiss and Zubin (1969), and others, on several grounds.

Suppose the data consist of a similarity matrix but one wants to apply a clustering algorithm designed for dissimilarities. Then it is necessary to transform the similarities into dissimilarities. The larger the similarity $s(i, j)$ between i and j, the smaller their dissimilarity $d(i, j)$ should be. Therefore, we need a decreasing transformation, such as

$$d(i, j) = 1 - s(i, j) \tag{15}$$

One could also take the square root of $1 - s(i, j)$, as advocated by Gower (1966) on the basis of a geometrical argument. This makes the differences between large similarities more important, but on the other hand makes it more difficult to obtain small dissimilarities. As a consequence, the resulting dissimilarity matrix might be rather homogeneous and less likely to yield clear-cut clusterings.

When applying (15) to correlation coefficients, expression (13) leads to formula (11), which means that negatively correlated variables are considered far apart. In the opposite case, (14) yields formula (12).

In order to be able to process similarities and correlation coefficients, the program DAISY executes (11), (12), and (15) as well as some other calculations. In Section 5 it will be explained how to use this program.

2.4 Binary Variables

Binary variables have only two possible outcomes (or states). For instance, when clustering people several binary variables may be used: male/female, smoker/nonsmoker, answered yes/no to a particular question, and so on. In the data matrix, such variables are often coded as zero or one. When variable f is binary, the objects i will have either $x_{if} = 0$ or $x_{if} = 1$. (It may be useful to allow a third code for missing values, e.g., to indicate that we do not know whether that particular person smokes or not.) Often 1 is taken to mean that a certain attribute is present (e.g., smoking), whereas 0 indicates its absence. Sometimes people treat binary variables just as if they were interval-scaled, that is, by applying the usual formulas for Euclidean or Manhattan distance. Although this may sometimes lead to decent results, it is good to know that there exist approaches designed specifically for binary data.

To begin with, there are special clustering algorithms for this situation, such as the *monothetic analysis* technique described in Chapter 7 and implemented in the program MONA. This algorithm operates directly on the binary data matrix, by dissecting the data according to a well-chosen variable. For instance, if the variable "smoking" were selected, the data would first be split into two clusters: the one consisting of smokers and the other of nonsmokers.

Another possibility is to compute a dissimilarity matrix (or a similarity matrix) from binary data and then simply to apply one of the clustering algorithms that operates on such a matrix (such as the methods described in Chapters 2, 4, 5, and 6). If all binary variables are thought of as having the same weight, one typically proceeds as follows. When computing a similarity $s(i, j)$ or a dissimilarity $d(i, j)$ between two objects i and j, one draws

a 2-by-2 contingency table (or association table) such as

$$\begin{array}{cc|c|c|c} & & \multicolumn{2}{c}{\text{object } j} & \\ & & 1 & 0 & \\ \hline \text{object } i & 1 & a & b & a+b \\ \hline & 0 & c & d & c+d \\ \hline & & a+c & b+d & p \end{array} \tag{16}$$

Here, a is the number of variables that equal 1 for both objects. Analogously, b is the number of variables f for which $x_{if} = 1$ and $x_{jf} = 0$, and so on. Obviously $a + b + c + d = p$, the total number of variables. (When missing values occur, one has to replace p by the number of variables that are available for both i and j. One could also compute weighted sums: If a variable is perceived to be very important, it may be given a higher weight than the other variables. In such a situation, p will be replaced by the sum of all the weights.) The values a, b, c, and d are then combined in a coefficient describing to what extent objects i and j agree with regard to the collection of binary variables.

Table 9 provides an example of binary data. For 8 people, a total of 10 binary variables were considered, such as male/female, blue eyes/brown eyes, round face/oval face, and so on. The attribute listed first is always the one coded as 1, for instance blue eyes = 1 and brown eyes = 0. When comparing Ilan with Talia, we make up a table like (16) which yields $a = 1$, $b = 3$, $c = 1$, and $d = 5$. Note that interchanging Ilan and Talia would permute b and c (while leaving a and d unchanged), so a good similarity or dissimilarity coefficient must treat b and c in the same way in order to satisfy (D3) or (S3).

At this point a crucial remark is in order. Following Gower (1971a, p. 858) and Bock (1974, Section 4) we can distinguish between *two* kinds of binary variables, depending on the particular application.

The binary variable "sex" possesses the possible states "male" and "female." Both are equally valuable and carry the same weight. There is no preference which outcome should be coded as 0 and which as 1. Such a variable is called *symmetric*. This is the first type of binary variable, which occurs very frequently. For symmetric variables, it is natural to work with *invariant* similarities, that is, the result must not change when some or all of the binary variables are coded differently. Therefore, a and d should play the same role. One looks for coefficients that only depend on the number of agreements $(a + d)$ and the number of disagreements $(b + c)$ between the objects i and j that are being compared. Table 10 gives the most common

Table 9 Binary Variables for Eight People

Person	Sex (Male = 1, Female = 0)	Married (Yes = 1, No = 0)	Fair Hair = 1, Dark Hair = 0	Blue Eyes = 1, Brown Eyes = 0	Wears Glasses (Yes = 1, No = 0)	Round Face = 1, Oval Face = 0	Pessimist = 1, Optimist = 0	Evening Type = 1, Morning Type = 0	Is an Only Child (Yes = 1, No = 0)	Left-Handed = 1, Right-Handed = 0
Ilan	1	0	1	1	0	0	1	0	0	0
Jacqueline	0	1	0	0	1	0	0	0	0	0
Kim	0	0	1	0	0	0	1	0	0	1
Lieve	0	1	0	0	0	0	0	1	1	0
Leon	1	1	0	0	1	1	0	1	1	0
Peter	1	1	0	0	1	0	1	1	0	0
Talia	0	0	0	1	0	1	0	0	0	0
Tina	0	0	0	1	0	1	0	0	0	0

Table 10 Some Invariant Coefficients for Binary Data

Name	$s(i, j)$	$d(i, j)$
Simple matching coefficient (Zubin, 1938; Dumas, 1955; Sokal and Michener, 1958; Sneath, 1962; Hill et al., 1965)	$\dfrac{a+d}{a+b+c+d}$	$\dfrac{b+c}{a+b+c+d}$
Rogers and Tanimoto (1960)	$\dfrac{a+d}{(a+d)+2(b+c)}$	$\dfrac{2(b+c)}{(a+d)+2(b+c)}$
Sokal and Sneath (1963) (Duran and Odell 1974)	$\dfrac{2(a+d)}{2(a+d)+(b+c)}$	$\dfrac{b+c}{2(a+d)+(b+c)}$

invariant similarities $s(i, j)$, together with the corresponding invariant dissimilarities $d(i, j) = 1 - s(i, j)$.

The most well known of these is the *simple matching* coefficient, which looks for the percentage of matches (i.e., agreements), or equivalently the percentage of mismatches (i.e., disagreements) between objects i and j. For the distance between Ilan and Talia it yields $d(i, j) = (3 + 1)/(1 + 3 + 1 + 5) = 0.4$. Sometimes it is also called *M-coefficient* or *affinity index*. If one treats binary variables as if they were interval-scaled and computes the Manhattan distance (without first standardizing the measurements), one obtains $d(i, j) = b + c$ which corresponds to the simple matching dissimilarity except for a constant factor p. In the same way, the Euclidean distance between objects i and j corresponds to the square root of the simple matching dissimilarity. Note that treating binary variables as if they were interval-scaled implies that they are assumed to be symmetric, because interchanging the codes 0 and 1 for some or all variables will still yield the same distance.

The other coefficients in Table 10 are less often used. In the Rogers and Tanimoto (1960) formulas, the disagreements $(b + c)$ carry twice the weight of the agreements $(a + d)$. On the other hand, Sokal and Sneath (1963) doubly weight the agreements. However, there is a simple monotone relation between all three coefficients, because the Rogers–Tanimoto dissimilarity can be written as a monotone function of the simple matching dissimilarity:

$$\frac{2(b+c)}{(a+d)+2(b+c)} = \frac{2}{1/((b+c)/(a+b+c+d))+1} \tag{17}$$

and the same holds for the dissimilarity coefficient proposed by Sokal and Sneath:

$$\frac{b+c}{2(a+d)+(b+c)} = \frac{1}{2/((b+c)/(a+b+c+d))-1} \tag{18}$$

Therefore, it often makes little difference which of these three coefficients is used (especially if one applies a clustering algorithm that only depends on the ranks of the dissimilarities, such as the single linkage method discussed later). In this book, we prefer to work with the matching coefficient because it is simple and intuitive. In Section 5 we shall explain how to compute it by means of the program DAISY.

The situation changes drastically if one works with *asymmetric* binary variables, for which the outcomes are not equally important. An example of such a variable is the presence or absence of a relatively rare attribute, such as bloodtype AB negative. While it can be said that two people with AB negative have something in common, it is not so clear if the same can be said of two people who do *not* have it. In medicine, one may want to study the incidence of diseases, the presence of which are indicated by 1 and their absence by 0. For a typical sample of people, the data matrix would contain many zeroes, and most of the counts of the contingency tables like (16) would be in d. Applying one of the invariant coefficients of Table 10 would lead to the conclusion that most people are very similar. Bock (1974, Section 4) gives an illuminating example concerning the color of flowers: The binary variable red = 1/not red = 0 is very asymmetric, as the statement "$x_{if} = 1$ and $x_{jf} = 1$" implies that flowers i and j have the same color, whereas "$x_{if} = 0$ and $x_{jf} = 0$" is much weaker and allows the flowers to have very different colors.

When working with asymmetric binary variables, we need other proximity coefficients. By convention, we shall always code the most important outcome (which is typically the rarest one) by 1, and the other by 0. Then the agreement of two 1s (called a *positive match* or a *1-1 match*) will be considered more significant than the agreement of two 0s (called a *negative match* or a *0-0 match*). Therefore, coefficients will be applied in which a, the number of positive matches, carries more weight than d, the number of negative matches. Such coefficients are no longer invariant and the most common of them, listed in Table 11, do not even count d at all.

The most famous noninvariant coefficient is due to Jaccard (1908) and looks like the simple matching coefficient except for leaving out d entirely. It has occasionally been called *S-coefficient*. The other formulas in Table 11 assign double weight to a or to $(b + c)$, and are monotonically related to the Jaccard coefficient in a manner analogous to (17) and (18). There are still other variants, some of which will be listed in Exercise 15. When dealing with asymmetric binary variables, we prefer to use the Jaccard coefficient, which has also been implemented in the program DAISY.

Table 11 Some Noninvariant Coefficients for Binary Data

Name	$s(i, j)$	$d(i, j)$
Jaccard coefficient (1908) (Sneath, 1957; Hill et al., 1965)	$\dfrac{a}{a+b+c}$	$\dfrac{b+c}{a+b+c}$
Dice (1945), Sorensen (1948)	$\dfrac{2a}{2a+b+c}$	$\dfrac{b+c}{2a+b+c}$
Sokal and Sneath (1963) (Duran and Odell, 1974)	$\dfrac{a}{a+2(b+c)}$	$\dfrac{2(b+c)}{a+2(b+c)}$

There have been some philosophical debates as to whether or not negative matches should be counted at all. From a mathematical point of view, the invariant coefficients are more elegant, whereas in some applications it may be more appropriate to use a formula of Table 11. In our opinion there can be no single best coefficient because one should make the distinction between symmetric and asymmetric variables. Symmetric binary variables possess two equally important states, so for them the simple matching coefficient appears to be a logical choice. On the other hand, asymmetric binary variables are mostly concerned with the *presence* of a relatively rare attribute (coded 1), the absence of which (coded 0) is uneventful. By abuse of the word binary, one might call them monary variables. In this situation 0-0 matches do not contribute much to the similarity between two individuals, so the Jaccard coefficient appears to give a reasonable description. Therefore, DAISY lets the user decide whether the binary variables are symmetric, in which case simple matching will be performed, or asymmetric, in which situation the Jaccard coefficient will be computed.

To illustrate why it is important to make this distinction, let us return to the example of Table 9. Based on their interpretation, these binary variables appear to be symmetric. When the simple matching coefficient is used, we find

$$d(\text{JAC}, \text{LIE}) = 0.3 \qquad d(\text{ILA}, \text{PET}) = 0.5$$

On the other hand, applying the Jaccard coefficient (which would be rather inappropriate in this context) would yield

$$d(\text{JAC}, \text{LIE}) = 0.750 \qquad d(\text{ILA}, \text{PET}) = 0.714$$

The main point is not that the actual values are different (which was to be expected), but that the results are not monotone: In the first situation we find that $d(\text{JAC}, \text{LIE}) < d(\text{ILA}, \text{PET})$, whereas in the second situation it turns out that $d(\text{JAC}, \text{LIE}) > d(\text{ILA}, \text{PET})$, which could lead to quite different clusterings. [Applying either of the remaining coefficients of Table 10 would still yield $d(\text{JAC}, \text{LIE}) < d(\text{ILA}, \text{PET})$ because they have a monotone relation with the simple matching coefficient, while the measures of Table 11 all yield $d(\text{JAC}, \text{LIE}) > d(\text{ILA}, \text{PET})$ because they depend in a monotone way on the Jaccard coefficient.]

When both symmetric and asymmetric binary variables occur in the same data set, one can apply the "mixed variables" approach described in Section 2.6.

2.5 Nominal, Ordinal, and Ratio Variables

Apart from binary and interval-scaled variables, there exist at least three other types of data, which are less commonly used. We shall briefly describe these scales with some discussion as to how to treat them.

a. Nominal Variables

In the previous section we studied binary variables, which can only take on two states, typically coded as 1 and 0. This generalizes naturally to the concept of a nominal variable, which may take on more than two states. For instance, in Table 9 we had the binary variable blue eyes/brown eyes, which was appropriate for that collection of people. However, in larger populations one will need at least four states: blue eyes/brown eyes/green eyes/grey eyes. In general, we denote the number of states by M and code the outcomes as $1, 2, \ldots, M$ in the data matrix (sometimes the codes $0, 1, \ldots, M-1$ are also used). For instance, we could choose 1 = blue eyes, 2 = brown eyes, 3 = green eyes, and 4 = grey eyes. Note that these states are not ordered in any way: It is not because grey eyes are given a higher code number than brown eyes that they would in some sense be better. The code numbers are only used to facilitate data handling, but one could just as well code the different outcomes by letters or other symbols. Some examples of nominal variables are the nationality of people (for which M may be very large) or their marital status (bachelor/married/divorced/widowed).

Sometimes nominal variables are converted to binary ones. By collapsing some states until only two remain, a binary variable results. For instance, one can group green eyes with brown eyes and grey with blue. However, this clearly amounts to a loss of information. Another strategy would be to recode the data to a larger number of (asymmetric) binary variables, for instance by creating a new binary variable for each of the M nominal states, and then to put it equal to 1 if the corresponding state occurs and to 0 otherwise. After that, one could resort to one of the dissimilarity coefficients of the previous subsection.

By far the most common way of measuring the similarity or dissimilarity between some objects i and j that are characterized through nominal variables is to use the *simple matching* approach:

$$s(i, j) = \frac{u}{p} \quad \text{and} \quad d(i, j) = \frac{p-u}{p} \tag{19}$$

(Sokal and Michener, 1958). Here, u is the number of matches, that is, the number of variables for which objects i and j happen to be in the same state. As before, p is the total number of variables (or, in a situation with missing values, the number of variables that are available for both i and j). Therefore, simple matching has exactly the same meaning as in the preceding section. For instance, it is invariant with respect to different codings of the variables because this does not affect the number of matches.

Again it is possible to give a higher weight to u, the number of agreements, or to $p - u$, the number of disagreements. Such variants were considered by Rogers and Tanimoto (1960) and Sokal and Sneath (1963), corresponding to the formulas in Table 10. It must also be noted that different variables may have different values of M. Therefore, Hyvarinen (1962) assigns more weight to matches in variables with a large number of states. Lingoes (1967) extends this by counting, for all variables, the frequency with which each state actually occurs and by giving a higher weight to matches corresponding to rare states. (This is reminiscent of the treatment of asymmetric binary variables.) Some other variants can be found in Bock (1974, Section 5).

We personally prefer the simple matching approach (19) because of its intuitive appeal and widespread acceptance. Simple matching dissimilarities can be computed by means of the program DAISY. It is not necessary to know the number of states for each variable because the program will itself produce an inventory with the number of states and the number of missing values. Also, the codes entered may be arbitrary real numbers, so the variables do not have to be coded in a discrete way. The main purpose of DAISY is to deliver a dissimilarity matrix which can be used by some of the clustering algorithms described in the following chapters.

b. Ordinal Variables

A *discrete* ordinal variable looks like a nominal variable, only now the M states are ordered in a meaningful sequence. The codes $1, \ldots, M$ are no longer arbitrary. The distance between two states becomes larger when their codes are further apart, so the states coded 1 and M differ most from each other.

Ordinal variables are very useful for registering subjective assessments of qualities that cannot be measured objectively. For example, you may ask someone to convey his or her appreciation of some paintings in terms of the following categories: detest = 1/dislike = 2/indifferent = 3/like = 4/admire = 5. This person's taste will then be modelled as an ordinal variable with $M = 5$ states. Another possibility is to rank 20 paintings in increasing order of appreciation, yielding an ordinal variable with states $1, 2, \ldots, 20$ and $M = 20$. (Note that in the latter example each state will occur exactly once, whereas in the first it may happen that some states occur very often and others not at all.) One may also obtain ordinal variables from the discretization of interval-scaled quantities, by splitting up the continuous axis in a finite number of classes. Some examples are weight categories for boxers and tax brackets.

Sometimes there does exist an underlying interval-scaled variable, but it has not been measured. For instance, one can construct a ranking of the hardness of stones by making scratches on one stone with another, without necessarily being able to measure their hardness in absolute terms. Or one can organize a race without needing a stopwatch, by merely registering who came in first, second,..., and so on, as in the ancient olympics.

Continuous ordinal variables are very similar. They occur when the measurements are continuous, but one is not certain whether they are in anything like a linear scale, so the only trustworthy information is in the ordering of the observations. Indeed, if a scale is transformed by an exponential, a logarithmic or another nonlinear monotone transformation, it loses its interval property: A difference of 3.2 on one end of the new scale may be much more important than a difference of 3.2 on the other end. Therefore, one replaces the observations by their ranks $1, \ldots, M$ where M is the number of different values taken on by the continuous variable. (Of course, two equal measurements receive the same rank.) This is also very useful when the original data were roughly on an interval scale, but contained some gross errors. By switching to the ranks, such errors will have a much smaller influence on the result. It is as if we do have the running times of the race, but we are only interested in the ranking because we consider the exact time intervals irrelevant (imagine the last runner, seeing he is going to lose anyway and just walking the final part). Also, maybe we do not know whether the "right" variable should be the total running time or the average speed, which is on the inverse scale. In such situations, it is often useful to reduce the data to the essentials by converting them to ranks.

Whatever its origin, we are left with a variable with ordered states $1, 2, \ldots, M$. It would be a waste of information to treat it as if it were nominal, because the further two states are apart, the larger the resulting dissimilarity should become. Therefore, most authors advise treating the ranks as interval-scaled and applying the usual formulas for obtaining dissimilarities (like the Euclidean or Manhattan distance). As it may happen that the ordinal variables under study possess different values of M, it is useful to convert all variables to the 0-1 range in order to achieve equal weighting of the variables. This can be done by replacing the rank r_{if} of the ith object in the fth variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1} \tag{20}$$

where M_f is the highest rank for variable f. In this way, all z_{if} will lie between 0 and 1.

The program DAISY can be applied to a data set with ordinal variables, whether discrete or continuous. It will first convert each variable f to ranks $1, 2, \ldots, M_f$ in such a way that equal measurements lead to equal ranks and that each rank occurs at least once. Then it will replace all ranks by z_{if} as in (20). The final dissimilarity between objects i and j is then taken to be the Manhattan distance (7) divided by the number of variables that are nonmissing for both objects.

Note that when the *variables* are to be clustered one can compute a full set of nonparametric correlations between them (say, Spearman coefficients) by means of any standard statistical package and then apply DAISY to transform these into a dissimilarity matrix by means of (11) or (12).

c. *Ratio Scale Variables*

We have seen that interval-scaled variables are positive or negative numbers on some kind of linear scale, for instance, the interval between 41°C and 51°C is equally important as the interval between −28°C and −18°C. By contrast, ratio-scaled variables are always positive measurements, for which the distinction between 2 and 20 has the same meaning as the distinction between 20 and 200. Typical examples are the concentration of a chemical substance in a certain solvent or the radiation intensity of some radioactive isotope. Often such ratio-scaled quantities follow exponential laws in time. For instance, the total amount of microorganisms that evolve in a time t (in a closed system with abundant nourishment) is approximately given by

$$Ae^{Bt} \tag{21}$$

where A and B are positive constants. Formula (21) is usually referred to as exponential growth and has been a reasonable model for the world population over certain time periods. Similarly, the concentration of some alien substances in human blood or the radiation intensity of an isotope can be modelled by an exponential decay formula

$$Ae^{-Bt} \tag{22}$$

In both (21) and (22), equal time intervals will lead to equal ratios of the quantities described, for instance, each year the radioactivity will decrease by the same percentage when compared to the level of the previous year.

When clustering objects that are characterized by ratio scale variables, one has at least three options. The first is to simply treat them *as if they were on an interval scale*. This is often done by people who only distinguish between qualitative and quantitative variables, without considering the fine distinction between interval and ratio scales. In general, we would not recommend this because the scale might become distorted. A second approach, which is very common in chemistry, is to begin with a *logarithmic transformation* of the ratio scale variables x_{if}, at least when they are all nonzero, that is, one computes

$$y_{if} = \log(x_{if}) \qquad (23)$$

and treats these y_{if} as interval-scaled. This is quite a sensible procedure, especially in situations where (21) or (22) apply. A third approach is to treat the x_{if} as *continuous ordinal data* and switch to their ranks. This could also be done after the logarithmic transformation (23) and would then yield exactly the same result. The ranks are then treated as interval-scaled data, in the way already described. This third approach is especially suitable when there are doubts whether the original data are interval or ratio scaled, or in case of uncertainty about the quality of the measurements. By only making use of the ordinal information, the distinction between interval and ratio disappears. Dissimilarities between objects (according to all three approaches) can be computed by means of the program DAISY, the use of which will be explained in Section 5.

2.6 Mixed Variables

In this section we have seen six types of variables:

symmetric binary
asymmetric binary
nominal
ordinal
interval
ratio

and we have discussed methods of dealing with data sets of one of these types. However, in practical applications it can easily happen that several kinds of variables occur in the same data set. For example, we could combine the interval variables of Table 7 with the binary variables of Table 9 because they pertain to the same individuals. A larger example is Table 12, listing certain characteristics of garden flowers. In the first column it is indicated whether the plant winters, that is, whether it may be left in the garden when it freezes. The second column shows whether the flower may stand in the shadow; those for which this is not so should be planted where they are exposed to direct sunlight. These columns are symmetric binary variables, with equally important states. The third binary variable is coded 1 for tuberous plants and 0 for plants without tubers. This variable is asymmetric because two plants with tubers have at least something in common, whereas plants without tubers may grow in completely different ways. The next column describes the color of the flowers. This variable is nominal, with $m = 5$ states occurring in these data (white = 1, yellow = 2,

Table 12 Characteristics of Some Garden Flowers

Garden Flower	Winters (Yes = 1, No = 0)	Shadow (Yes = 1, No = 0)	Tubers (Yes = 1, No = 0)	Color of Flowers (White = 1, Yellow = 2, Pink = 3, Red = 4, Blue = 5)	Soil (Dry = 1, Normal = 2, Humid = 3)	Preference (Low = 1, High = 18)	Height (cm)	Planting distance (cm)
1. Begonia (*Bertinii boliviensis*)	0	1	1	4	3	15	25	15
2. Broom (*Cytisus praecox*)	1	0	0	2	1	3	150	50
3. Camellia (Japonica)	0	1	0	3	3	1	150	50
4. Dahlia (Tartini)	0	0	1	4	2	16	125	50
5. Forget-me-not (*Myosotis sylvatica*)	0	1	0	5	2	2	20	15
6. Fuchsia (Marinka)	0	1	0	4	3	12	50	40
7. Geranium (Rubin)	0	0	0	4	3	13	40	20
8. Gladiolus (Flowersong)	0	0	1	2	2	7	100	15
9. Heather (*Erica carnea*)	1	1	0	3	1	4	25	15
10. Hydrangea (Hortensis)	1	1	0	5	2	14	100	60
11. Iris (Versicolor)	1	1	1	5	3	8	45	10
12. Lily (*Lilium regale*)	1	1	1	1	2	9	90	25
13. Lily-of-the-valley (Convallaria)	1	1	0	1	2	6	20	10
14. Peony (*Paeonia lactiflora*)	1	1	1	4	2	11	80	30
15. Pink Carnation (Dianthus)	1	0	0	3	2	10	40	20
16. Red Rose (*Rosa rugosa*)	1	0	0	4	2	18	200	60
17. Scotch Rose (*Rosa pimpinella*)	1	0	0	2	2	17	150	60
18. Tulip (*Tulipa sylvestris*)	0	0	1	2	1	5	25	10

pink = 3, red = 4, and blue = 5). The fifth column says whether the plant thrives best in dry (1), normal (2), or humid (3) soil. This is an ordinal variable, the states being ranked according to increasing moisture. The sixth column is someone's preference ranking, going from 1 to 18. The code 18 next to the red rose indicates that this flower is best liked, whereas the code 1 is assigned to the plant least liked. This ordinal variable possesses many states, but each state occurs only once. The last columns list the height of the plants and the distances that should be left between them, both expressed in centimeters. Therefore, this data set contains only two interval-scaled variables out of a total of eight attributes.

Data with mixed variables can be treated in several ways. To begin with, it is possible not to mix these types at all but to perform a separate cluster analysis for each kind of variable. When the conclusions of these analyses more or less agree, all is well. However, when different results are obtained, it may be difficult to reconcile them.

Therefore, it is more practical to process the data together and then to perform a single cluster analysis. For instance, one can treat all variables as if they were interval-scaled. This is quite appropriate for symmetric binary variables, for the ranks originating from ordinal variables, and for the logarithms of ratio variables. However, for nominal variables with more than two states this does not make much sense because some codes may be further apart than others without reflecting an intrinsic "remoteness" of the corresponding states. Also, asymmetric binary variables would be treated symmetrically.

The opposite approach is to reduce everything to binary variables. How to do this for nominal variables was already discussed. It is also easy to obtain a binary variable from interval-scaled measurements y_{if} by cutting the measurement axis in two, that is, by applying a rule like

$$\begin{aligned} &\text{if } y_{if} < a_f, \text{ then put } x_{if} = 0 \\ &\text{if } y_{if} \geq a_f, \text{ then put } x_{if} = 1 \end{aligned}$$

where the threshold a_f may be chosen by means of subject-matter information or simply by selecting a value in the center of the data. (It may even be that the y_{if} form two clear clusters in one dimension, in which case a_f may be chosen between them.) The same rule can be applied to ordinal and ratio variables. However, by converting the whole data set to binary attributes one may lose quite a bit of information, which is often considered a disadvantage.

In our opinion, the most convenient approach is to combine the different variables into a single proximity matrix, as was proposed by Ducker et al. (1965), Rubin (1967), and Colless (1967). The definition of Gower (1971a) takes care of interval, nominal, and binary data. We shall describe a slight generalization of this method, also covering ordinal and ratio variables. Actually, Gower's original definition was a similarity coefficient between 0 and 1, but we shall transform it to a dissimilarity by means of (15). Conversely, one can always return to similarities by computing $s(i, j) = 1 - d(i, j)$ at the end.

Suppose the data set contains p variables of mixed nature. Then the dissimilarity $d(i, j)$ between objects i and j is defined as

$$d(i, j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}} \tag{24}$$

The indicator $\delta_{ij}^{(f)}$ is put equal to 1 when both measurements x_{if} and x_{jf} for the fth variable are nonmissing, and it is put equal to 0 otherwise. Moreover, $\delta_{ij}^{(f)}$ is also put equal to 0 when variable f is an asymmetric binary attribute and objects i and j constitute a 0-0 match. Expression (24) cannot be computed when all $\delta_{ij}^{(f)}$ are zero, in which case $d(i, j)$ must be assigned a conventional value or object i or j must be removed.

The number $d_{ij}^{(f)}$ is the contribution of the fth variable to the dissimilarity between i and j. We may assume that both x_{if} and x_{jf} are nonmissing: otherwise $d_{ij}^{(f)}$ does not have to be computed. If variable f is either *binary* or *nominal*, then $d_{ij}^{(f)}$ is defined as

$$\begin{aligned} d_{ij}^{(f)} &= 1 \quad \text{if } x_{if} \neq x_{jf} \\ &= 0 \quad \text{if } x_{if} = x_{jf} \end{aligned} \tag{25}$$

If all variables are nominal, expression (24) becomes the number of matches over the total number of available pairs, so it coincides with the simple matching coefficient (19). The same holds for symmetric binary variables, for which the simple matching coefficient of Table 10 is recovered. When the data consist of asymmetric binary variables, we obtain the Jaccard coefficient of Table 11 because the 0-0 matches are not counted (because their $\delta_{ij}^{(f)}$ equals zero).

If variable f is *interval-scaled*, then $d_{ij}^{(f)}$ is given by

$$d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{R_f} \tag{26}$$

where R_f is the range of variable f, defined as

$$R_f = \max_h x_{hf} - \min_h x_{hf} \tag{27}$$

where h runs over all nonmissing objects for variable f. Therefore, (26) is always a number between 0 and 1. *Ordinal* variables are first replaced by their ranks, after which (26) is applied. *Ratio* variables may be treated as interval variables: They can be converted to ranks or a logarithmic transformation may be carried out. In either case, (26) is applied to the result.

When all variables are interval-scaled, Gower's formula (24) becomes the Manhattan distance, assuming that the variables were first divided by their range [note that this standardization is quite different from (4)]. When all variables are ordinal, (24) yields the same result as the method described in Section 2.5. The same is true for ratio variables.

We conclude that the combined method (24) generalizes the dissimilarities of the homogeneous data discussed earlier. The computations can be performed with the program DAISY, as described in Section 5. For instance, it easily deals with Table 12. The same program is used for data with variables of a single type and for processing similarities and correlation coefficients. In all cases, it delivers a dissimilarity matrix that can be used to run four of the clustering programs of this book. Figure 15 of Section 4 contains a survey of the function of DAISY, among other information.

Note that we followed Gower in restricting $d_{ij}^{(f)}$ to the 0-1 range, so each variable yields a contribution between 0 and 1 to the average dissimilarity (24). [As a consequence, the resulting dissimilarity $d(i, j)$ also lies between 0 and 1, and can be turned back into Gower's original formula by computing $s(i, j) = 1 - d(i, j)$.] Why restrict ourselves to this range? Suppose we were to allow contributions with very different ranges, as done by Romesburg (1984) in his combined resemblance matrix approach. Then some interesting anomalies are possible. Take an example with a few asymmetric binary variables and many interval variables, the latter yielding contributions $d_{ij}^{(f)}$ around 3 or 4. Consider an asymmetric binary variable with $x_{if} = 0$ and $x_{jf} = 1$, which yields a contribution $d_{ij}^{(f)}$ of 1. Now change x_{jf} to 0, so we obtain a 0-0 match and the term $d_{ij}^{(f)}$ vanishes both in the numerator and the denominator. This yields a *larger* dissimilarity $d(i, j)$ than before. This effect is, of course, opposite to what was expected. It appears necessary to have equal ranges if one wants to be able to delete certain contributions, or else the effects of such deletions may be unwarranted.

As a final remark, it must be noted that it is even possible to cluster objects that are characterized by a combination of measurements and proximities. For instance, suppose we have a similarity matrix, a dissimilarity matrix, and a mixed collection of attributes, all pertaining to the same n objects. Then DAISY can convert the similarity matrix into a dissimilarity matrix as in (15), and compute another dissimilarity matrix from the attributes according to (24). The three resulting dissimilarity matrices can then be combined into a single one by means of

$$d(i, j) = \frac{w_1 d_1(i, j) + w_2 d_2(i, j) + w_3 d_3(i, j)}{w_1 + w_2 + w_3}$$

where w_1, w_2, and w_3 are some positive weights that may be chosen in a subjective way.

This section was devoted to clustering *objects* that are characterized by attributes of mixed types. In some situations, however, one might want to cluster the *variables* themselves. This was discussed by Lance and Williams (1979), who compute dissimilarities between variables of mixed types.

3 WHICH CLUSTERING ALGORITHM TO CHOOSE

Let us give an overview of the clustering methods implemented in this book, together with their most important characteristics and some hints toward typical applications. The choice of a clustering algorithm depends both on the type of data available and on the particular purpose. Sometimes several algorithms are applicable, and a priori arguments may not suffice to narrow down the choice to a single method. In such a situation it is probably a good idea to run more than one program and to carefully analyze and compare the resulting classifications, making use of their graphical displays. The interpretation of these results must then be based on insight into the meaning of the original data, together with some experience with the algorithms used. It is permissible to try several algorithms on the same data, because cluster analysis is mostly used as a descriptive or exploratory tool, in contrast with statistical tests which are carried out for inferential or confirmatory purposes. That is, we do not wish to prove (or disprove) a preconceived hypothesis; we just want to see what the data are trying to tell us.

Of course there exist very many clustering algorithms in the literature, and it would be infeasible to try to review all of them. Bock (1974)