When their number is even, we define the median as the highest of the two middle values.

(a) Show that the dissimilarity between merging clusters is monotone.

(b) Show that the dissimilarity between clusters is unambiguous and statistically consistent in the sense of Section 5.2.

(c) Show that monotone transformations on the dissimilarities between objects do not change the clustering (and, in fact, the levels of the mergers are transformed in the same way).

10. Another alternative method is based on the dissimilarity

$$d(R, Q) = \max\left\{ \max_{i \in R}\left[\min_{j \in Q} d(i, j)\right], \max_{j \in Q}\left[\min_{i \in R} d(i, j)\right]\right\}$$

which is again uniquely defined and symmetric in R and Q.

(a) Show that this dissimilarity transforms properly under monotone transformations of the interobject dissimilarities $d(i, j)$.

(b) As opposed to the group average dissimilarity (5), now the dissimilarity of a cluster to itself is 0.

(c) Assuming that $d(i, j) = 0$ implies $i = j$, it even follows that

$$d(R, Q) = 0 \quad \text{implies} \quad R = Q$$

which is not true for any other method we considered, not even for single linkage or the centroid method.

CHAPTER 6

Divisive Analysis (Program DIANA)

1 SHORT DESCRIPTION OF THE METHOD

Divisive clustering techniques are hierarchical in nature. Their main difference with the agglomerative methods of Chapter 5 is that they proceed in the inverse order. At each step, a divisive method splits up a cluster into two smaller ones, until finally all clusters contain only a single element. This means that the hierarchy is again built in $n - 1$ steps when the data set contains n objects.

In the literature, divisive methods have been largely ignored. (In fact, when people talk about hierarchical clustering, they often mean agglomerative clustering.) Most books on cluster analysis pay little attention to divisive techniques, and many software packages do not include divisive algorithms at all. The main reason for this appears to be the computational aspect. In the first step of an agglomerative algorithm all possible fusions of two objects are considered, leading to

$$C_n^2 = \frac{n(n-1)}{2} \tag{1}$$

combinations. This number grows quadratically with n, so it does become large, but the computations are still feasible. A divisive algorithm based on the same principle would start by considering all divisions of the data set into two nonempty subsets, which amounts to

$$2^{n-1} - 1 \tag{2}$$

possibilities. The latter number grows exponentially fast and soon exceeds the current estimate of the number of atoms in the universe. Even for medium-sized data sets, such a complete enumeration approach is computationally prohibitive.

Nevertheless, it is possible to construct divisive methods that do not consider *all* divisions, most of which would be totally inappropriate anyway. In this chapter we will use an algorithm based on the proposal of Macnaughton-Smith et al. (1964). This algorithm can be applied to exactly the same types of data as in Chapters 2, 4, and 5. All data sets that can be clustered by means of the agglomerative nesting approach of the preceding chapter can also be analyzed in a divisive way. In most examples the computation time is merely increased by a factor of 2, so the algorithm is still faster than the nonhierarchical methods of Chapters 2 and 4.

To illustrate the divisive analysis algorithm, we cluster the same example that we used in Section 1 of Chapter 5 to explain the agglomerative approach. The data consist of a matrix of dissimilarities

	a	b	c	d	e
a	0.0	2.0	6.0	10.0	9.0
b	2.0	0.0	5.0	9.0	8.0
c	6.0	5.0	0.0	4.0	5.0
d	10.0	9.0	4.0	0.0	3.0
e	9.0	8.0	5.0	3.0	0.0

between the objects a, b, c, d, and e. Being divisive, the algorithm assumes that the objects initially form a single cluster $\{a, b, c, d, e\}$.

In the first step, the algorithm has to split up the data set into two clusters. This is not done by considering all possible divisions, but rather by means of a kind of iterative procedure. The mechanism somewhat resembles the way a political party might split up due to inner conflicts: First the most discontented member leaves the party and starts a new one, and then some others follow him until a kind of equilibrium is attained. So we first need to know which member disagrees most with the others. Translated back to the algorithm, we look for the object that is most dissimilar to all other objects. To make this precise, we have to define the dissimilarity between an object and a group of objects. As in Chapters 2, 3, and 5, we use the *average dissimilarity* for this purpose, so we look for the object for which the average dissimilarity to all other objects is largest. (When there are two such objects, we have to pick one at random.) In our example, we

obtain

Object	Average Dissimilarity to the Other Objects
a	$(2.0 + 6.0 + 10.0 + 9.0)/4 = 6.75$
b	$(2.0 + 5.0 + 9.0 + 8.0)/4 = 6.00$
c	$(6.0 + 5.0 + 4.0 + 5.0)/4 = 5.00$
d	$(10.0 + 9.0 + 4.0 + 3.0)/4 = 6.50$
e	$(9.0 + 8.0 + 5.0 + 3.0)/4 = 6.25$

so object a is chosen to initiate the so-called *splinter group*. At this stage we have the groups $\{a\}$ and $\{b, c, d, e\}$, but we don't stop here. For each object of the larger group we compute the average dissimilarity with the remaining objects, and compare it to the average dissimilarity with the objects of the splinter group:

Object	Average Dissimilarity to Remaining Objects	Average Dissimilarity to Objects of Splinter Group	Difference
b	$(5.0 + 9.0 + 8.0)/3 \approx 7.33$	2.00	5.33
c	$(5.0 + 4.0 + 5.0)/3 \approx 4.67$	6.00	−1.33
d	$(9.0 + 4.0 + 3.0)/3 \approx 5.33$	10.00	−4.67
e	$(8.0 + 5.0 + 3.0)/3 \approx 5.33$	9.00	−3.67

On the political scene, some party members are secretly asking themselves whether they disagree more (on average) with the people remaining in the old party than with the splinter group. The politician for whom this difference is largest will be the next to move. In our example the difference is largest for object b, which lies much further from the remaining objects than from the splinter group. Therefore object b changes sides, so the new splinter group is $\{a, b\}$ and the remaining group becomes $\{c, d, e\}$. When we repeat the computations we find

Object	Average Dissimilarity to Remaining Objects	Average Dissimilarity to Objects of Splinter Group	Difference
c	$(4.0 + 5.0)/2 = 4.50$	$(6.0 + 5.0)/2 = 5.50$	−1.00
d	$(4.0 + 3.0)/2 = 3.50$	$(10.0 + 9.0)/2 = 9.50$	−6.00
e	$(5.0 + 3.0)/2 = 4.00$	$(9.0 + 8.0)/2 = 8.50$	−4.50

At this stage, all the differences have become negative. Returning to our political analogy, the remaining party members have more quarrels with the splinter group than with each other. Therefore, no further moves are made. The process stops and we have completed the first divisive step, which splits the data into the clusters $\{a, b\}$ and $\{c, d, e\}$.

In the next step we divide the biggest cluster, that is, the cluster with the largest diameter. (As before, the diameter of a cluster is just the largest dissimilarity between two of its objects.) The diameter of $\{a, b\}$ is 2.0, and for $\{c, d, e\}$ we find 5.0. Therefore, the above procedure will be applied to the cluster $\{c, d, e\}$, with dissimilarity matrix

	c	d	e
c	0.0	4.0	5.0
d	4.0	0.0	3.0
e	5.0	3.0	0.0

To find the rebel to start the splinter group with, we compute

Object	Average Dissimilarity to the Other Objects
c	$(4.0 + 5.0)/2 = 4.50$
d	$(4.0 + 3.0)/2 = 3.50$
e	$(5.0 + 3.0)/2 = 4.00$

and obtain object c. Afterward, we find

Object	Average Dissimilarity to Remaining Objects	Average Dissimilarity to Objects of Splinter Group	Difference
d	3.0	4.00	−1.00
e	3.0	5.00	−2.00

and the process stops because all differences are negative. Therefore, our second step divides $\{c, d, e\}$ into $\{c\}$ and $\{d, e\}$, so we are left with the clusters $\{a, b\}$, $\{c\}$, and $\{d, e\}$. The cluster $\{c\}$ is called a singleton because it contains only one object.

In the third step we must decide which of these clusters to split. Obviously, the singleton $\{c\}$ cannot be divided any further (also note that its diameter is 0). The cluster $\{a, b\}$ has diameter 2 and that of $\{d, e\}$ is 3.

Therefore, we have to divide the cluster $\{d, e\}$ with dissimilarity matrix

	d	e
d	0.0	3.0
e	3.0	0.0

To start the splinter group, we calculate very little:

Object	Average Dissimilarity to the Other Objects
d	3.00
e	3.00

Because the average dissimilarities are the same, we may choose either object to begin the splinter group with. Let us choose object d, so we obtain $\{d\}$ and $\{e\}$. It is clear that object e, being the last remnant, cannot join the splinter group. Therefore, step 3 divides $\{d, e\}$ into the singletons $\{d\}$ and $\{e\}$. (Note that the same result would be obtained if we had picked object e to start the splinter group with.) Step 3 leaves us with four clusters: $\{a, b\}$, $\{c\}$, $\{d\}$, and $\{e\}$.

In the fourth step we have to split up the cluster $\{a, b\}$, because all the others contain only a single object. As in the third step, $\{a, b\}$ is divided into the singletons $\{a\}$ and $\{b\}$. After this fourth step we only have singleton clusters $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, and $\{e\}$, so the algorithm stops.

The resulting hierarchy may be displayed as in Figure 1, in which the horizontal axis contains the step number. At step 0 there is still a single

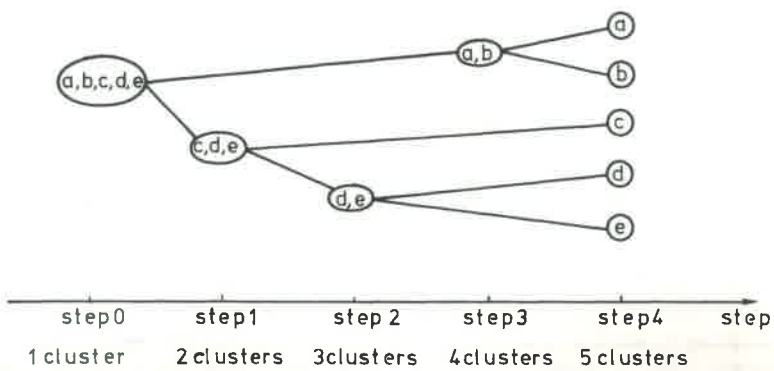

Figure 1 Graphical display of a divisive hierarchy.

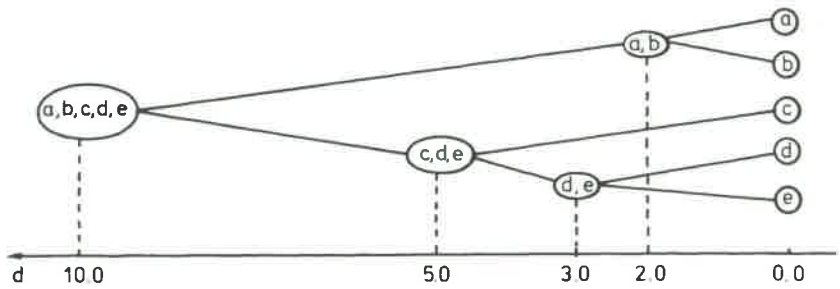

Figure 2 Diagram showing the same hierarchy as in Figure 1, but also displaying the level of each division.

cluster with all the objects, while at step 4 all objects are apart (in this example the objects retain their original ranking, but of course this need not be true in general).

It is possible to construct a more informative version of this diagram. Rather than just displaying the order in which the divisions take place, we could also indicate the diameter of the clusters involved, because we know that the bigger clusters are divided first. For this purpose, in Figure 2 each cluster is plotted versus its diameter. This diameter is a monotone function of the step number, because when a cluster has been split into two subclusters, neither can have a larger diameter than the original one. Also note that everything we have done only depends on the dissimilarity matrix, so the data need not consist of measurements. (When they do consist of measurements, we begin by computing dissimilarities, as will be illustrated in some examples.)

Instead of Figure 2, which was drawn by hand, it is possible to construct a display by means of an ordinary line printer. Figure 3 shows a banner (Rousseeuw, 1986) that contains the same information as Figure 2. At the extreme left (where the flagstaff can be imagined) all objects still stick together and the diameter is 10.0. The first split produces two clusters. The

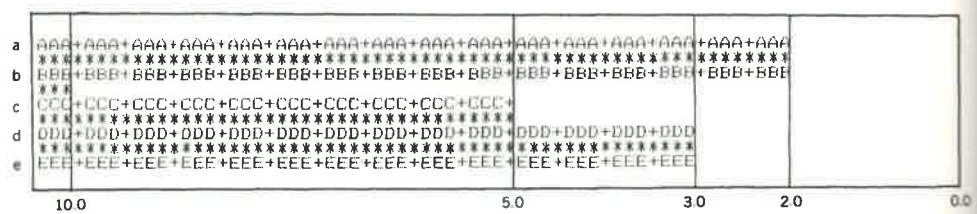

Figure 3 A divisive banner containing the same information as Figure 2.

first cluster is $\{a, b\}$, the objects of which are denoted by the labels *AAA* and *BBB*, and linked by a row of asterisks. The second cluster is $\{c, d, e\}$. Next, $\{c, d, e\}$ is split (at the level 5.0) into $\{c\}$ and $\{d, e\}$. Note that the label *CCC* is no longer continued after that, because $\{c\}$ is a singleton. The third step divides $\{d, e\}$ into $\{d\}$ and $\{e\}$ at the level 3.0 and the last step splits $\{a, b\}$ at the level 2.0.

Note that the representations in Figures 1, 2, and 3 are very similar to those of an agglomerative hierarchy, described in Section 1 of Chapter 5. The main difference is the direction of the algorithm, so the present displays look like mirror images of the previous ones. Also their interpretations are alike. For instance, one may again draw a vertical line through the banner to find the clustering corresponding to a certain level. However, the divisive algorithm is not the *exact* counterpart of the agglomerative one, so in general the resulting hierarchies do not coincide.

2 HOW TO USE THE PROGRAM DIANA

To apply this algorithm we use the program DIANA (from DIvisive ANAlysis), which was combined with the program AGNES described in Chapter 5. Both programs accept the same data sets. To run DIANA, we have to insert the floppy with the file TWINS.EXE in drive *A* and type

```
A:TWINS
```

which yields the screen

```
HIERARCHICAL CLUSTERING

DO YOU WANT AGGLOMERATIVE NESTING (AGNES)
OR DIVISIVE ANALYSIS (DIANA) ?
PLEASE ENTER YOUR CHOICE (A OR D) : D
```

By typing *D* we select the divisive algorithm. After that, the remaining input is exactly the same as for AGNES.

When we apply DIANA to the example treated in Section 1, we obtain the output shown in Figure 4. The first part is identical to the corresponding output of AGNES (Figure 6 of Chapter 5), but the cluster results and the banner are different. The program begins by telling us that in the first step the five objects are divided into two objects and three objects. This is useful because the first step of a divisive algorithm is very important. Indeed, the objects separated in the first step will stay apart throughout the

```
**********************
*                    *
*  DIVISIVE ANALYSIS *
*                    *
**********************

TITLE : Dissimilarities between five objects
DATA SPECIFICATIONS AND CHOSEN OPTIONS
--------------------------------------
 THERE ARE     5 OBJECTS
 LABELS OF OBJECTS ARE READ
 INPUT OF DISSIMILARITIES
 LARGE OUTPUT IS WANTED
 GRAPHICAL OUTPUT IS WANTED (BANNER)
 THE DISSIMILARITIES WILL BE READ IN FREE FORMAT

 YOUR DATA RESIDE ON FILE        : a:example.dat

DISSIMILARITY MATRIX
--------------------

AAA
BBB       2.00
CCC       6.00      5.00
DDD      10.00      9.00      4.00
EEE       9.00      8.00      5.00      3.00

CLUSTER RESULTS
---------------

AT THE FIRST STEP THE   5 OBJECTS ARE DIVIDED INTO
     2 OBJECTS AND    3 OBJECTS

THE FINAL ORDERING OF THE OBJECTS IS
       1              2              3              4              5

THE DIAMETERS OF THE CLUSTERS ARE
            2.000         10.000          5.000          3.000


************
*          *
*  BANNER  *
*          *
************

  1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 9 9 8 8 8 7 7 6 6 6 5 5 4 4 4 3 3 2 2 2 1 1 0 0 0
  0 6 2 8 4 0 6 2 8 4 0 6 2 8 4 0 6 2 8 4 0 6 2 8 4 0

AAA+AAA+AAA+AAA+AAA+AAA+AAA+AAA+AAA+AAA+AAA+AAA+AAA+AAA+AAA+AAA+AAA
**********************************************************************
BBB+BBB+BBB+BBB+BBB+BBB+BBB+BBB+BBB+BBB+BBB+BBB+BBB+BBB+BBB+BBB+BBB
***
CCC+CCC+CCC+CCC+CCC+CCC+CCC+CCC+CCC+CCC+CCC+
*********************************************
DDD+DDD+DDD+DDD+DDD+DDD+DDD+DDD+DDD+DDD+DDD+DDD+DDD+DDD+DDD+DDD+DDD
**********************************************************************
EEE+EEE+EEE+EEE+EEE+EEE+EEE+EEE+EEE+EEE+EEE+EEE+EEE+EEE+EEE

  1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 9 9 8 8 8 7 7 6 6 6 5 5 4 4 4 3 3 2 2 2 1 1 0 0 0
  0 6 2 8 4 0 6 2 8 4 0 6 2 8 4 0 6 2 8 4 0 6 2 8 4 0

THE ACTUAL DIAMETER OF THIS DATA SET IS              10.0000000000

THE DIVISIVE COEFFICIENT OF THIS DATA SET IS     .70

THE OUTPUT IS WRITTEN ON FILE : a:example.dia
```

Figure 4 DIANA output for the example of Section 1.

hierarchy, in which both clusters are divided further. We can easily identify the clusters found in the initial step because they consist of the first two and the last three objects in the "final ordering of the objects" list. The diameters of the clusters are listed below this ordering.

The final ordering and the diameters were already encountered in Figure 2. In fact, these two sequences of numbers together characterize the whole hierarchy. In this example, they are

1	2	3	4	5
2.0	10.0	5.0	3.0	

In the bottom row we first look up the largest diameter, 10.0, which stands

```
**********************
*                    *
*  DIVISIVE ANALYSIS *
*                    *
**********************

TITLE : Data set of Table 1 of Chapter 5
DATA SPECIFICATIONS AND CHOSEN OPTIONS
--------------------------------------
 THERE ARE     7 OBJECTS
 LABELS OF OBJECTS ARE NOT READ
 INPUT OF MEASUREMENTS
 LARGE OUTPUT IS WANTED
 GRAPHICAL OUTPUT IS WANTED (BANNER)

 THERE ARE     2 VARIABLES IN THE DATA SET
  AND     2 OF THEM WILL BE USED IN THE ANALYSIS
 THE LABELS OF THESE VARIABLES ARE :
        x
        y
 THE MEASUREMENTS WILL NOT BE STANDARDIZED
 EUCLIDEAN DISTANCE IS USED
 THERE ARE NO MISSING VALUES
 THE MEASUREMENTS WILL BE READ IN FREE FORMAT

 YOUR DATA RESIDE ON FILE        : a:seven.dat

DISSIMILARITY MATRIX
--------------------

001
002       4.03
003       4.24      1.12
004        .71      4.27      4.30
005       1.41      5.41      5.66      1.58
006       5.83      1.80      2.00      6.04      7.21
007       5.86      2.51      1.68      5.84      7.27      1.95

CLUSTER RESULTS
---------------

AT THE FIRST STEP THE   7 OBJECTS ARE DIVIDED INTO
     3 OBJECTS AND    4 OBJECTS

THE FINAL ORDERING OF THE OBJECTS IS
       1              4              5              2              3
       6              7

THE DIAMETERS OF THE CLUSTERS ARE
             .707          1.581          7.267          1.118          2.000
            2.512
```

Figure 5 DIANA output for the data in Table 1 of Chapter 5.

between two numbers of the upper row. This means that the whole data set will be split at the level 10.0, yielding a cluster with the objects 1 and 2 (standing to the left of 10.0) and a cluster with the objects 3, 4, and 5 (standing to the right of 10.0). The second largest diameter is 5.0, indicating that $\{3,4,5\}$ is to be split into $\{3\}$ and $\{4,5\}$ at that level. Continuing in this way, one can immediately reconstruct Figure 2.

The last part of the output is the banner that we already saw in Figure 3. It looks like the agglomerative banner in the AGNES output, but it floats in the opposite direction. Also the scales that surround it are plotted differently because they *decrease* from 1.00 to 0.00. Here, 0.00 indicates a zero diameter (corresponding to singletons) and 1.00 stands for the actual diameter of the data set (which is printed immediately below the banner). Again the overall width of the banner reflects the strength of the clustering: A very pronounced structure implies that the diameter of the entire data set is much larger than the diameters of the individual clusters, leading to a wide banner. As in the case of agglomerative nesting, we may summarize the blackness of the banner in a coefficient. For each object i we measure the length $l(i)$ of its line in the banner, with respect to the normalized scale. Therefore all $l(i)$ lie between 0 and 1, so the same holds for the *divisive*

```
            ************
            *          *
            *  BANNER  *
            *          *
            ************

  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  9  9  8  8  8  7  7  6  6  6  5  5  4  4  4  3  3  2  2  2  1  1  0  0  0
  0  6  2  8  4  0  6  2  8  4  0  6  2  8  4  0  6  2  8  4  0  6  2  8  4  0

001+001+001+001+001+001+001+001+001+001+001+001+001+001+001+001+001+00
*************************************************************************
004+004+004+004+004+004+004+004+004+004+004+004+004+004+004+004+004+00
*************************************************************************
005+005+005+005+005+005+005+005+005+005+005+005+005+005+005+0
***
002+002+002+002+002+002+002+002+002+002+002+002+002+002+002+002+002+00
*************************************************************************
003+003+003+003+003+003+003+003+003+003+003+003+003+003+003+003+003+00
*************************************************************************
006+006+006+006+006+006+006+006+006+006+006+006+006+006+006+0
*************************************************************
007+007+007+007+007+007+007+007+007+007+007+007+007+

  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  9  9  8  8  8  7  7  6  6  6  5  5  4  4  4  3  3  2  2  2  1  1  0  0  0
  0  6  2  8  4  0  6  2  8  4  0  6  2  8  4  0  6  2  8  4  0  6  2  8  4  0

THE ACTUAL DIAMETER OF THIS DATA SET IS              7.2672210000

THE DIVISIVE COEFFICIENT OF THIS DATA SET IS    .81

THE OUTPUT IS WRITTEN ON FILE : a:seven.dia
```

Figure 5 (*Continued*)

coefficient (DC) defined by

$$\mathrm{DC} = \frac{1}{n}\sum_{i=1}^{n} l(i) \tag{3}$$

The DC can be seen as the average width of the divisive banner, so it is the counterpart of the agglomerative coefficient (AC) defined in Section 2 of Chapter 5. Also the DC does not change when all the original dissimilarities are multiplied by a constant factor, nor does it depend on the ordering of the objects in the banner [because the $l(i)$ only depend on the actual hierarchy]. In Figure 4, the DC equals 0.70, which is quite good.

Of course, DIANA can also deal with data consisting of interval-scaled measurements. To illustrate this, we analyze the data listed in Table 1 of Chapter 5. When running DIANA, we specify that there are seven objects and two variables, which we decide not to standardize. We also instruct the program to compute Euclidean distances. Figure 5 contains the resulting output, which again looks very similar to the corresponding AGNES output in Figure 7 of Chapter 5. In the first (and most important) step, the data are split up into the clusters $\{1,4,5\}$ and $\{2,3,6,7\}$, as could be anticipated from the scatterplot (Figure 1 of Chapter 5). The second largest value in the "diameters of the clusters" list is 2.512, so $\{2,3,6,7\}$ is divided into $\{2,3,6\}$ and $\{7\}$ in the second step (note that all these diameters are dissimilarities between two objects, so they can always be traced back in the dissimilarity matrix). Because the diameter of the entire data set (7.267) is much larger than those of the clusters, the banner becomes rather wide, yielding a DC of 0.81. The ordering of the objects in the banner corresponds to that in Figure 7 of Chapter 5. This is partly due to the particular implementation of AGNES and DIANA, because in all our programs we have strived to modify the original ordering as little as possible to enable the user to compare the results of different methods.

3 EXAMPLES

Let us look at some examples to gain a better insight into the divisive approach. Applying DIANA to the 12 countries data in Table 5 of Chapter 2 yields Figure 6. (The first part of the output is identical to that produced by AGNES, so only the second part is shown here.) The first step divides the 12 objects into the clusters {BEL, FRA, USA, ISR, BRA, ZAI, EGY, IND} and {CHI, CUB, USS, YUG}, which means that the Communist countries are split off first. In the second step, the remainder

```
CLUSTER RESULTS
---------------

AT THE FIRST STEP THE  12 OBJECTS ARE DIVIDED INTO
     8 OBJECTS AND    4 OBJECTS

THE FINAL ORDERING OF THE OBJECTS IS

      1              6              9              8              2
     12              5              7              3              4
     10             11

THE DIAMETERS OF THE CLUSTERS ARE

          2.170          2.500          3.920          6.420          3.000
          5.080          4.670          8.170          4.500          2.670
          3.750

                 ************
                 *          *
                 *  BANNER  *
                 *          *
                 ************

  1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 9 9 8 8 8 7 7 6 6 6 5 5 4 4 4 3 3 2 2 2 1 1 0 0 0
  0 6 2 8 4 0 6 2 8 4 0 6 2 8 4 0 6 2 8 4 0 6 2 8 4 0

BEL+BEL+BEL+BEL+BEL+BEL+BEL+BEL+BEL+BEL+BEL+BEL+BEL+BE
**************************************************
FRA+FRA+FRA+FRA+FRA+FRA+FRA+FRA+FRA+FRA+FRA+FRA+FRA+FR
**************************************************
USA+USA+USA+USA+USA+USA+USA+USA+USA+USA+USA+USA+USA+USA
**************************************
ISR+ISR+ISR+ISR+ISR+ISR+ISR+ISR+ISR+ISR+IS
******************
BRA+BRA+BRA+BRA+BRA+BRA+BRA+BRA+BRA+BRA+BRA+BR
**********************************************
ZAI+ZAI+ZAI+ZAI+ZAI+ZAI+ZAI+ZAI+ZAI+ZAI+ZAI+ZA
**********************************
EGY+EGY+EGY+EGY+EGY+EGY+EGY+EGY+EGY+EGY
******************************
IND+IND+IND+IND+IND+IND+IND+IND+IND
***
CHI+CHI+CHI+CHI+CHI+CHI+CHI+CHI+CHI+
************************************
CUB+CUB+CUB+CUB+CUB+CUB+CUB+CUB+CUB+CUB+CUB+CUB+C
**************************************************
USS+USS+USS+USS+USS+USS+USS+USS+USS+USS+USS+USS+U
**************************************************
YUG+YUG+YUG+YUG+YUG+YUG+YUG+YUG+YUG+YUG+YUG+YUG

  1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  0 9 9 8 8 8 7 7 6 6 6 5 5 4 4 4 3 3 2 2 2 1 1 0 0 0
  0 6 2 8 4 0 6 2 8 4 0 6 2 8 4 0 6 2 8 4 0 6 2 8 4 0

THE ACTUAL DIAMETER OF THE DATA SET IS                8.1700000000

THE DIVISIVE COEFFICIENT OF THIS DATA SET IS     .60

THE OUTPUT IS WRITTEN ON FILE : a:country.dia
```

Figure 6 Output of DIANA for the 12 countries data.

is divided into {BEL, FRA, USA, ISR}, the Western countries, and {BRA, ZAI, EGY, IND}, the developing ones. (This may be seen from the cluster results or from the banner.) As usual, the most important information is provided by the first few steps of the divisive algorithm.

This clustering of the 12 countries is very similar to the results we obtained in previous chapters by means of PAM, FANNY, and AGNES.

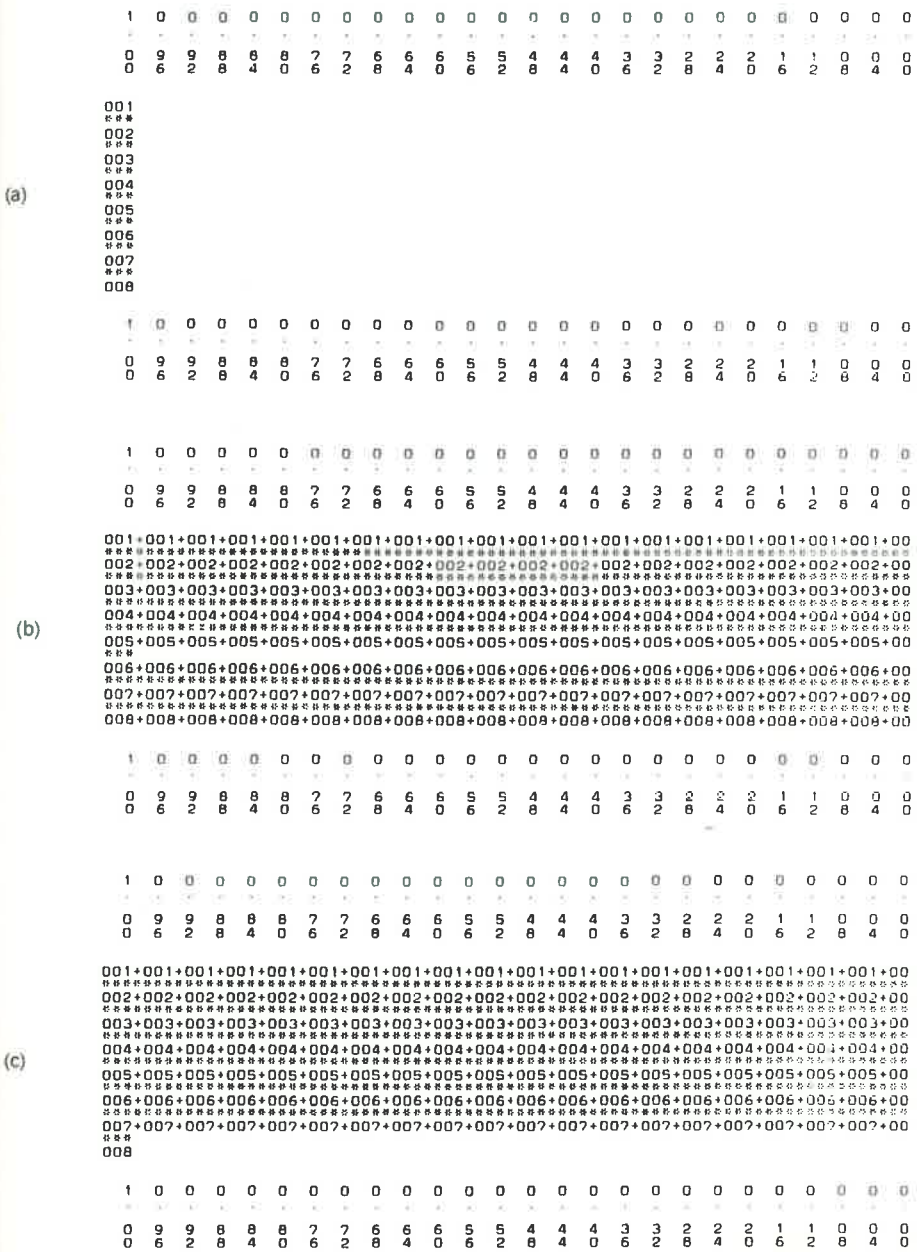

Figure 7 Divisive banners of some extreme examples, with (a) no clustering structure, (b) two very tight clusters, and (c) one distant outlier.

Table 1 Logarithmic Surface Temperature (x) and Logarithmic Light Intensity (y) of 47 Stars

i	x_i	y_i
1	4.37	5.23
2	4.56	5.74
3	4.26	4.93
4	4.56	5.74
5	4.30	5.19
6	4.46	5.46
7	3.84	4.65
8	4.57	5.27
9	4.26	5.57
10	4.37	5.12
11	3.49	5.73
12	4.43	5.45
13	4.48	5.42
14	4.01	4.05
15	4.29	4.26
16	4.42	4.58
17	4.23	3.94
18	4.42	4.18
19	4.23	4.18
20	3.49	5.89
21	4.29	4.38
22	4.29	4.22
23	4.42	4.42
24	4.49	4.85
25	4.38	5.02
26	4.42	4.66
27	4.29	4.66
28	4.38	4.90
29	4.22	4.39
30	3.48	6.05
31	4.38	4.42
32	4.56	5.10
33	4.45	5.22
34	3.49	6.29
35	4.23	4.34
36	4.62	5.62
37	4.53	5.10
38	4.45	5.22
39	4.53	5.18
40	4.43	5.57
41	4.38	4.62
42	4.45	5.06

Table 1 (*Continued*)

i	x_i	y_i
43	4.50	5.34
44	4.45	5.34
45	4.55	5.54
46	4.45	4.98
47	4.42	4.50

Source: Rousseeuw and Leroy (1987).

The main features are found in all four analyses. The DC = 0.60, so it is somewhat larger than the AC, as is often the case.

To illustrate the interpretation of the divisive banner and the DC, we turn to the same extreme examples that were considered in Section 3 of the preceding chapter. Figure 7a is the banner of the situation where all dissimilarities between objects equal the same positive constant. It clearly tells us that the data have no clustering structure, hence the DC is 0. Figure 7b corresponds to two very tight and well-separated clusters, yielding the widest possible banner with DC = 1.00. When the data contain an extreme outlier, we obtain the banner in Figure 7c, with DC = $1 - 1/n \approx 0.88$. All

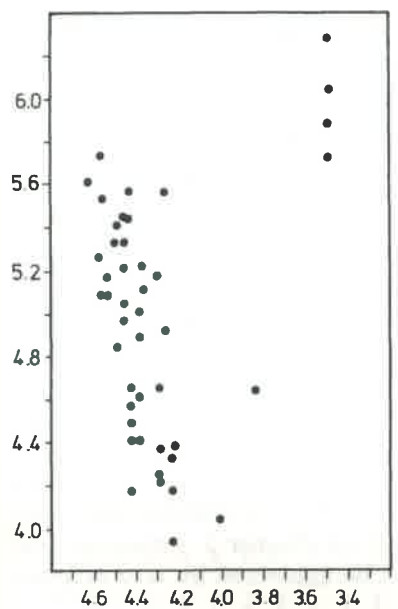

Figure 8 Scatterplot of the data in Table 1.

three banners are mirror images of those produced by AGNES, so the results of both hierarchical methods are consistent with each other.

The data in Table 1 are about the star group denoted by CYG OB1, which consists of 47 stars in the direction of Cygnus. This set of real data was given to us by C. Doom (personal communication). The variable x stands for the logarithm of the star's surface temperature (as measured from its spectrum) and y is the logarithm of its light intensity. In astronomy it is common practice to draw the so-called Hertzsprung–Russell diagram (Figure 8), which is simply a scatterplot of y versus x (but note that x is plotted from right to left).

In Figure 8, our eyes see two rather distinct clusters. Most of the points lie close to a steep band, whereas four points sit in the upper right corner of the plot. These regions of the Hertzsprung–Russell diagram are well known: The 43 stars belong to the so-called main sequence, whereas the four remaining ones (numbers 11, 20, 30, and 34) are giant stars. Running DIANA yields the banner of Figure 9, in which the first step does distinguish the main sequence from the giants. In subsequent steps the algorithm then splits up both clusters until all points are isolated. Also when the data are standardized, DIANA finds the "right" clusters in the first step. By means of AGNES we obtain a similar (though reversed) picture. Note that this example is not easy, because the first cluster is much larger and contains many more objects than the second, from which it is not clearly separated. For this reason, many algorithms have difficulties with these data. For instance, PAM yields two other clusters when $k = 2$, but it does isolate the giants for $k = 3$. One may also apply a totally different approach: Because the largest cluster (the main sequence) is nearly linear, it may be identified by means of a robust regression technique, as done by Rousseeuw and Leroy (1987, p. 28) who apply the least median of squares method of Rousseeuw (1984).

Our last example is the data set of Ruspini (Table 6 of Chapter 2) that was already analyzed in Chapters 2 and 4. In the scatterplot (Figure 12 of Chapter 2) four natural groups were indicated, namely $A = \{1, \ldots, 20\}$, $B = \{21, \ldots, 43\}$, $C = \{44, \ldots, 60\}$, and $D = \{61, \ldots, 75\}$. When we apply DIANA we obtain the banner of Figure 10. In the first step, the data are divided into $A \cup D$ on the one hand and $B \cup C$ on the other. In step 2 the cluster $A \cup D$ is split into A and D, and in step 3 also $B \cup C$ is divided into B and C. In step 4 the trio $\{46, 47, 48\}$ is taken out of C, whereas step 5 bisects A (but note that steps 4 and 5 occur at a much lower level, so it would seem reasonable to stop after step 3). This whole process corresponds exactly to the partitions obtained by PAM for 2, 3, 4, 5, and 6 clusters, the silhouettes of which are shown in Figure 13 of Chapter 2. Apparently the natural structure is sufficiently clear-cut to be uncovered by very different clustering algorithms.

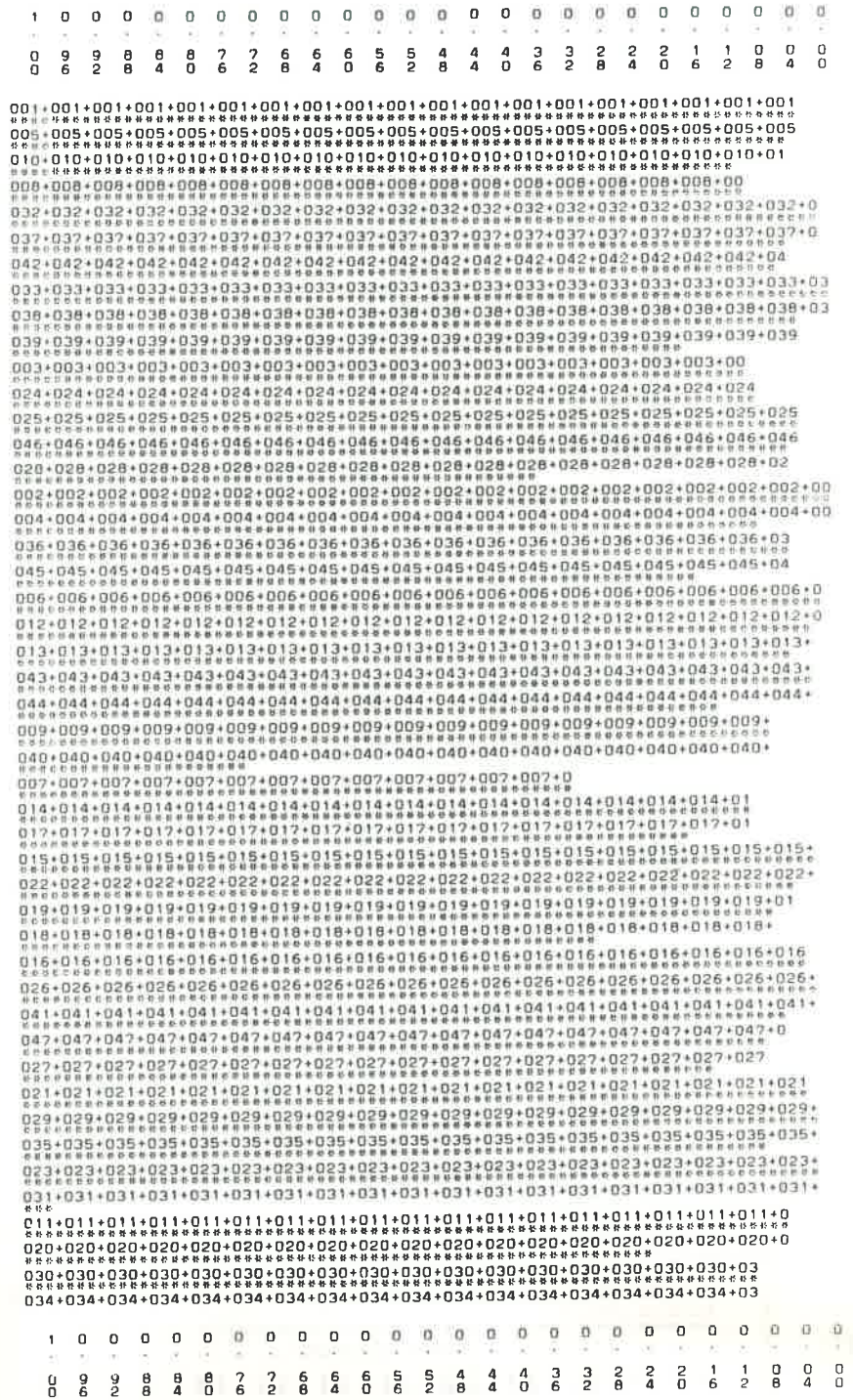

Figure 9 Divisive banner of the data in Table 1.

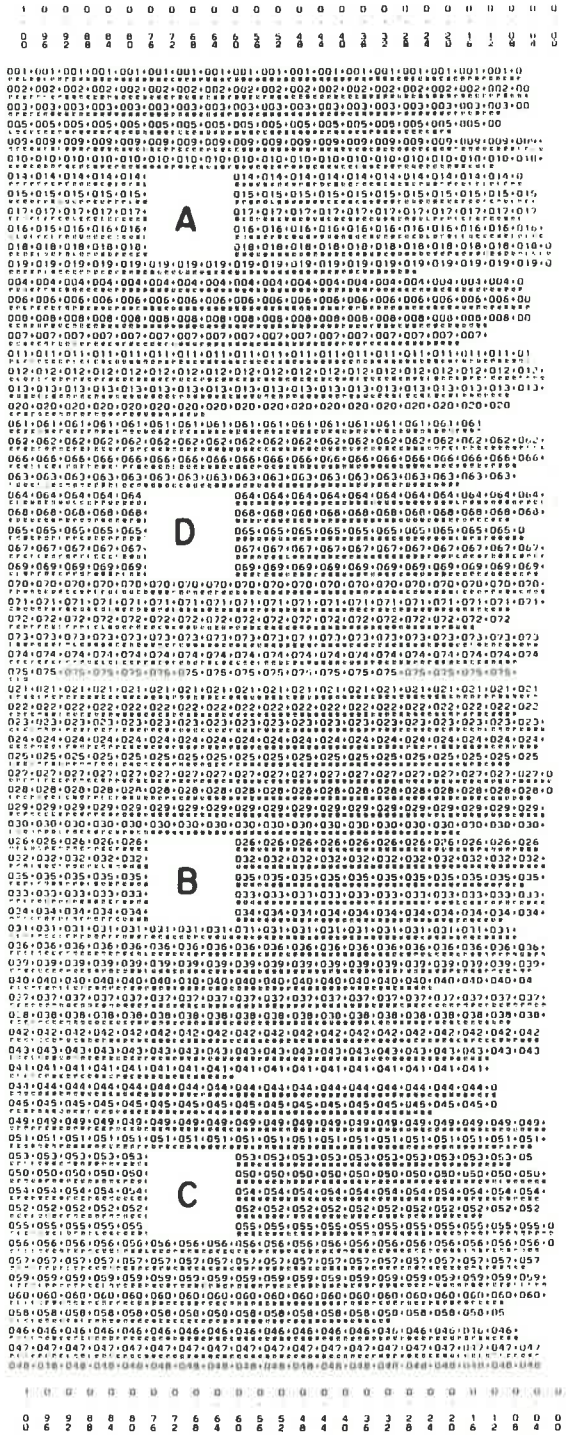

Figure 10 Divisive banner of the Ruspini data.

*4 MORE ON THE ALGORITHM AND THE PROGRAM

4.1 Description of the Algorithm

A divisive analysis proceeds by a series of successive splits. At step 0 (before starting the algorithm) all objects are together in a single cluster. At each step a cluster is divided, until at step $n-1$ all objects are apart (forming n clusters, each with a single object).

Each step divides a cluster, let us call it R, into two clusters A and B. This is done according to the iterative method of Macnaughton-Smith et al. (1964) that was illustrated in Section 1. Initially, A equals R and B is empty. In a first stage, we have to move one object from A to B. (Naturally, it is assumed here that A contains more than one object, or there is nothing to split.) For each object i of A, we compute the average dissimilarity to all other objects of A:

$$d(i, A \setminus \{i\}) = \frac{1}{|A|-1} \sum_{\substack{j \in A \\ j \neq i}} d(i, j) \tag{4}$$

The object i' for which (4) attains its maximal value will be moved, so we put

$$\begin{aligned} A_{\text{new}} &= A_{\text{old}} \setminus \{i'\} \\ B_{\text{new}} &= B_{\text{old}} \cup \{i'\} \end{aligned} \tag{5}$$

In the next stages, we look for other points to move from A to B. As long as A still contains more than one object, we compute

$$d(i, A \setminus \{i\}) - d(i, B) = \frac{1}{|A|-1} \sum_{\substack{j \in A \\ j \neq i}} d(i, j) - \frac{1}{|B|} \sum_{h \in B} d(i, h) \tag{6}$$

for each object i of A and we consider the object i'' that maximizes this quantity. When the maximal value of (6) is strictly positive, we move i'' from A to B as in (5) and then look in the new A for another object that might be moved. On the other hand, when the maximal value of (6) is negative or 0 we stop the process and the division of R into A and B is completed.

At each step of the divisive algorithm we also have to decide which cluster to split. For this purpose we compute the diameter

$$\operatorname{diam}(Q) = \max_{\substack{j \in Q \\ h \in Q}} d(j, h) \tag{7}$$

for each cluster Q that is available after the previous step, and choose the cluster for which (7) is largest. This value of (7) is also used as the level for representing the division of Q in the banner. We can do this because (7) is monotone:

$$A \subset R \quad \text{implies} \quad \text{diam}(A) \leq \text{diam}(R) \tag{8}$$

from which it follows that the levels of the successive steps form a nonincreasing sequence.

4.2 Structure of the Program

A list of subroutines and functions of the program TWINS was already given in Section 4.2 of Chapter 5. When we ask the program to perform a divisive analysis, the subroutines AVERL and BANAG (corresponding to agglomerative nesting) are no longer used, but instead SPLYT, SUPCL, and BANDY are activated.

The subroutine SPLYT divides a cluster by means of (4) to (6), except when the cluster has only two objects, because it can then be split without these computations. At each step the array NER, which contains the ordering of the objects, is updated in such a way as to disturb the original ranking as little as possible. (This implies that the splinter group B may either be listed as the first or as the second subcluster, depending on the situation.) The diameter (7) of a cluster is computed in subroutine SUPCL and afterward stored in the array BAN, which is updated accordingly. When the algorithm is finished, the numbers in NER ("final ordering of the objects") and BAN ("diameters of the clusters") are printed. Finally, the subroutine BANDY combines the arrays BAN and NER to draw the banner of the divisive hierarchy.

Table 2 lists some computation times for various values of n. In order to compare the speed of DIANA with that of AGNES, we used the same data

Table 2 Computation Times of DIANA (in minutes on an IBM-XT with 8087 coprocessor) for Increasing Numbers of Objects

Objects	Variables	Time
20	2	0.20
40	2	0.60
60	2	1.50
80	2	3.45
100	2	6.65

as in Table 2 of Chapter 5. It turns out that DIANA consumes about twice as much time as AGNES, which is perfectly feasible. In fact, DIANA is still faster than the nonhierarchical algorithms of Chapters 2 and 4, because a single run of DIANA yields partitions for all $k = 1, 2, \ldots, n$.

*5 RELATED METHODS AND REFERENCES

5.1 Variants of the Selected Method

In their very concise article, Macnaughton-Smith et al. (1964) describe how a cluster may be split in two by means of (4) to (6). Instead of (4), they also consider other measures of dissimilarity between clusters, particularly for binary data. They argue that divisive methods are safer than agglomerative ones, because "wrong" decisions in the early stages of an agglomerative analysis cannot be corrected later on, and one is mostly interested in the large clusters. Also, they note that it is infeasible to consider all possible divisions, of which there are $2^{n-1} - 1$. One could, of course, restrict attention to *monothetic* methods, that make each split according to a single variable, whereas *polythetic* methods use all variables simultaneously. (Our own point of view is to give preference to methods that only depend on interobject distances or dissimilarities, such as the algorithms of Chapters 2, 4, 5, and 6. Such methods are necessarily polythetic.) The method of Macnaughton-Smith et al. combines the three advantages of being divisive rather than agglomerative, polythetic rather than monothetic, and computationally manageable even for large n. To demonstrate the latter property, Rousseeuw (1983b) implemented this technique on an Apple II microcomputer in a way to minimize the use of memory. The implementation in DIANA takes somewhat more memory but less computation time.

Lance and Williams (1966a) investigated the computational complexity of several hierarchical methods. They found that the number of operations needed for the first step of the Macnaughton-Smith technique varies with the evenness of the division it produces. When the data set is divided into two clusters with n' and $n - n'$ objects, the number of operations was $(n' + 1)(2n - 1) - n$. In the worst case, when $n' = n/2$, this is of the order of n^2. In the best case, when $n' = 1$, it reduces to $3n - 2$. Therefore, the actual computation time depends on the course of the analysis, as in many iterative methods (such as our programs PAM and FANNY). But at any rate, the worst-case computation time of the Macnaughton-Smith technique remains feasible.

In order to reduce the amount of computation, Macnaughton-Smith et al. (1964) also considered the following variant of their procedure.

Provided that $d(i, A \setminus \{i\}) - d(i, B) > 0$, they chose the object i minimizing $d(i, B)$ instead of maximizing (6). It seems that this version might lead to a good internal cohesion in the splinter group B, but maybe not in the remaining cluster A.

Another variant that we might think of would be to start as in (4) and (5), and at each following stage move the object i that maximizes the group average dissimilarity

$$d(A \setminus \{i\}, B \cup \{i\}) \tag{9}$$

A possible stopping rule would be to halt the process as soon as $d(A_{\text{new}}, B_{\text{new}})$ no longer increases. Note that this procedure is internally consistent, because (4) is a special case of (9). Unfortunately, it takes somewhat more computation time than the Macnaughton-Smith method.

Chandon and Pinson (1981, p. 122) go even further when they propose to divide the data into any clusters A and B maximizing $d(A, B)$. This is a complete enumeration approach in which all $2^{n-1} - 1$ possible divisions must be considered, making the method extremely time-consuming [although much more elegant from a theoretical point of view, because it is dual to the group average method described in Chapter 5, in which $d(A, B)$ is minimized at each agglomerative step].

It appears that divisive clustering is only computationally feasible when performed in some iterative way. However, one aspect of the Macnaughton-Smith technique one might object to is its asymmetric treatment of the two clusters A and B, one of which arises as a splinter group whereas the other may be a rather loose collection of remaining objects. Therefore, one may prefer to grow the clusters from *two* kernel objects, instead of the single object i determined in (4). One proposal (Hubert, 1973, p. 51) would be to start with the objects i and j with the largest interobject dissimilarity $d(i, j)$. Afterward one could assign the remaining objects to i or j, whichever is nearer, or apply an iterative procedure in which the objects are assigned one at a time, taking into account their average dissimilarities to the clusters being formed. The main problem with this method appears to be its lack of robustness, as both i and j may be outliers.

A more robust approach would be to begin with two objects i and j that are centrally located in the clusters they determine. For instance, one might select i and j so as to maximize $d(A, B)$ where A contains all objects that are nearer to i than to j and B contains all objects that are nearer to j than to i. Note that this is similar to the k-medoid partition (for $k = 2$) described in Chapter 2. Indeed, one could also carry out a 2-medoid

clustering at each step and obtain another divisive hierarchy. However, these methods would also require more computation time.

Until now, we have only discussed *how* to split up a cluster. In the original method of Macnaughton-Smith et al. (1964) this was sufficient because at each step they split all available clusters. However, in our concept of a divisive hierarchy we want to split one cluster at a time, so we also need to know *which* cluster to split next. Leaving aside trivial solutions (such as making an arbitrary choice or splitting the cluster with the largest number of objects) this begs for the definition of a *level*. For instance, one may use a dispersion measure of the cluster to be split, or a kind of average dissimilarity between the two subclusters. Such a level must be meaningful (in relation to the interobject dissimilarities) and *monotone*, which means that neither subcluster may have a larger level than the original one. For some of the divisive methods sketched above the choice of a level is quite natural, but for the Macnaughton-Smith technique it is not so simple. We have tried out many definitions, but most of them were not monotone in all data sets (sometimes monotonicity could be proved for the splinter group but not for the remaining subcluster). This left us with the diameter, which is always monotone by (8) but not very satisfactory otherwise because of its sensitivity to outliers and its lack of consistency when n goes to infinity.

The program DIANA uses diameters for displaying the divisive hierarchy (as we did in Figures 2 and 3), which is better than just plotting the number of the divisive step (as in Figure 1) because it may happen that two clusters are split at the same level, in which case their step number would be arbitrary. Moreover, the step number always ranges from 1 to $n - 1$, so it does not reflect the particular clustering structure of the data. When a continuous level is plotted, the user may notice when a data set has been split into two clusters with much smaller levels than the original one (as in Figure 7b), leading to a wide banner and thus a large DC, whereas a banner merely displaying the step number would not have revealed the strength of the clustering structure.

The graphical displays that were considered in Section 5.3 of Chapter 5, such as those of Ward and Johnson as well as icicle plots and dendrograms, may also be used to describe a divisive hierarchy. It suffices to draw these displays in the inverse direction. Note that the DC does not depend on the type of display or the ranking of the objects in it.

5.2 Other Divisive Techniques

The methods reviewed in Section 5.1 only needed a matrix of distances or dissimilarities between the objects. Some other divisive methods are more

restrictive, in that they need a matrix of objects by variables. In this subsection we discuss some polythetic divisive methods, whereas monothetic divisive methods will be treated in Chapter 7.

The most well-known polythetic divisive method is that of Edwards and Cavalli-Sforza (1965). It assumes that the n objects are characterized by p interval-scaled variables. It is basically a least squares (L_2) method, just like minimum variance partitioning (Section 5.3 of Chapter 2) and Ward's agglomerative method (Section 5.2 of Chapter 5). Everything is based on the error sum of squares

$$\mathrm{ESS}(R) = \sum_{i \in R} \| x_i - \bar{x}(R) \|^2 \tag{10}$$

where $\bar{x}(R)$ is the centroid of the cluster R, with coordinates

$$\bar{x}_f(R) = \frac{1}{|R|} \sum_{i \in R} x_{if} \quad \text{for } f = 1, \ldots, p$$

Edwards and Cavalli-Sforza divide the cluster R into two subclusters A and B in such a way that the objective function

$$\mathrm{ESS}(A) + \mathrm{ESS}(B) \tag{11}$$

is minimized. In Section 5.2 of Chapter 5 [between (23) and (24), in connection with Ward's method] we proved that

$$\mathrm{ESS}(R) = \mathrm{ESS}(A) + \mathrm{ESS}(B) + \frac{|A||B|}{|R|} \| \bar{x}(A) - \bar{x}(B) \|^2$$

This means that the total sum of squares $\mathrm{ESS}(R)$ can be written as the *within* sum of squares $\mathrm{ESS}(A) + \mathrm{ESS}(B)$ plus the *between* sum of squares

$$\frac{|A||B|}{|R|} \| \bar{x}(A) - \bar{x}(B) \|^2 \tag{12}$$

Therefore, minimizing (11) is equivalent to maximizing (12), and the latter objective function is somewhat easier to compute. Unfortunately, the method of Edwards and Cavalli-Sforza still imposes a computational burden because *all* possible divisions of R into some clusters A and B must be tried out. A simplified sequential algorithm was announced, but not described. Such an approximate algorithm was, however, proposed by Gower (1967, p. 632). Gower advised against the exhaustive method of Edwards and Cavalli-Sforza because it is computationally impracticable and because the centroid is not always a desirable measure of the location of a cluster. When comparing this method with agglomerative nesting and a monothetic divisive method, he recommended the agglomerative technique. Also Lance and Williams (1966a) dismissed the method of Edwards and Cavalli-Sforza because of the large number of dichotomous choices that would have to be examined. Steinhausen and Langer (1977, p. 100) decided not to pursue such divisive methods, because they consume more computation time than agglomerative methods and at the same time yield less satisfactory results than partitioning techniques. Roux (1985, pp. 86–87) independently rediscovered the Edwards and Cavalli-Sforza criterion (12) and proposed another heuristic algorithm yielding an approximate solution.

Edwards and Cavalli-Sforza split all available clusters at each step. MacQueen (1967) proposed a variant of this method in which at each step only the cluster with the largest error sum of squares (10) is divided, so that there are again $n - 1$ steps. This algorithm may be considered as a divisive counterpart to Ward's agglomerative method. Instead of making the splits by total enumeration, it is also possible to apply the k-means algorithm (Section 5.3 of Chapter 2) with $k = 2$ to find approximate solutions for some or all steps.

The reasons why we did not select any of these variance minimization techniques are similar to our objections to k-means clustering and Ward's method. First of all, we prefer clustering techniques that only need a collection of dissimilarities because measurements are not always available, and second, methods based on sums of squares are generally nonrobust to the presence of outliers.

EXERCISES AND PROBLEMS

1. Draw a diagram as in Figure 2 for the divisive clustering of the countries in Section 3.

2. Show that for a three object cluster, the final splinter group consists of the object for which the average dissimilarity to the other two objects is maximal.

3. Run DIANA again on the 47 stars in Table 1, but select the option to standardize the data first.

4. Apply DIANA to the nine diseases in Figure 12 of Chapter 5, and compare the results with those of AGNES.

5. Apply DIANA to cluster the 11 sciences in Table 6 of Chapter 1. Compare the results with those obtained by running AGNES.

6. Run DIANA to cluster the objects of the artificial data set of Figure 1 in Chapter 4. (The actual data are listed in Section 2.1 of that chapter.) Do not standardize the measurements, and use Euclidean distance. What does DIANA do with the intermediate objects 6 and 13?

7. As suggested in Section 5.1, splitting a cluster into two subclusters could be done with the program PAM. (Note that for $k = 2$ the algorithm in PAM is exact, so this approach should lead to a tight clustering.) Apply this method (repeatedly!) to the dissimilarity matrix in Section 1 and compare the results with those of DIANA.

8. Often agglomerative and divisive algorithms yield similar clustering results. To show that this is not always the case, consider the following bivariate data set with 18 points:

$$(0,0)\quad (0,2)\quad (0,4)\quad (2,0)\quad (2,2)\quad (2,4)\quad (4,0)\quad (4,2)\quad (4,4)$$
$$(5,5)\quad (5,7)\quad (5,9)\quad (7,5)\quad (7,7)\quad (7,9)\quad (9,5)\quad (9,7)\quad (9,9)$$

   (a) Make a scatterplot of this data set.
   (b) Use both the agglomerative and divisive options of the program TWINS to cluster the data.
   (c) Indicate on the plot the clusterings into two clusters, obtained with both methods.

9. **(a)** Apply DIANA to the bacteria data in Exercise 1 of Chapter 2. Does the clustering into two clusters correspond to the result of PAM?
   (b) When a data set consists of two L^*-clusters (in the sense of Section 2.2 of Chapter 2) show that DIANA always recovers these groups. (Note that AGNES also finds the same groups by Exercise 6 of Chapter 5.)

10. Show that the number of possible partitions of n objects into two nonempty subsets is $2^{n-1} - 1$ (either directly, or by means of binomial coefficients, or as a special case of Stirling's result for a general number of subsets).

11. At each step of the algorithm, a cluster R is split into two subclusters A and B. The level of this division is presently defined as the diameter of R, which is monotone in the sense that the diameters of A and B are smaller than that of R. One could think of several alternative definitions of a level, to be used with the same splitting method. Show that the following variants, although intuitively appealing, are not always monotone:

   (a) $\underset{\substack{i,j\in R\\ i\neq j}}{\text{average}}\, d(i,j)$

   (b) $\max_{i\in R} \underset{\substack{j\in R\\ j\neq i}}{\text{average}}\, d(i,j)$

   (c) $\underset{\substack{i\in A\\ j\in B}}{\text{average}}\, d(i,j)$