Finding Groups in Data: An Introduction to Cluster Analysis
Leonard Kaufman and Peter J. Rousseeuw

CHAPTER 4

Fuzzy Analysis (Program FANNY)

1 THE PURPOSE OF FUZZY CLUSTERING

Fuzzy clustering is a generalization of partitioning. In a partition, each object of the data set is assigned to one and only one cluster. Therefore, partitioning methods (such as those of Chapters 2 and 3) are sometimes said to produce a hard clustering, because they make a clear-cut decision for each object. On the other hand, a fuzzy clustering method allows for some ambiguity in the data, which often occurs.

Let us consider Figure 1, which contains 22 objects characterized by two interval-scaled variables. Our eye immediately sees three main clusters, as well as two intermediate points. Suppose that we run a partitioning method and ask for three clusters. In that case, the program would have to make a rather arbitrary choice whether to add object 6 to the cluster $\{1,2,3,4,5\}$ or to $\{7,8,9,10,11,12\}$, because this object lies at approximately the same distance from both. Also, it would be very difficult to decide where to put object 13, which lies between the three main groups.

A fuzzy clustering technique is much better equipped to describe such situations because each object is "spread out" over the various clusters. For instance, a fuzzy clustering method is able to say that object 1 belongs for the most part to the first cluster, whereas object 13 should be divided almost equally between all three clusters. This degree of belonging is quantified by means of *membership coefficients* that range from 0 to 1. When we apply a fuzzy clustering method to the data of Figure 1, we obtain a list of membership coefficients as in Table 1.

Table 1 contains 66 membership coefficients. Looking at the first row, we see that object 1 belongs for 87% to cluster 1, for 6% to cluster 2, and for 7% to cluster 3. Note that the sum of the membership coefficients in each row equals 1 (or 100%). Also objects 2, 3, 4, and 5 belong mainly to the first

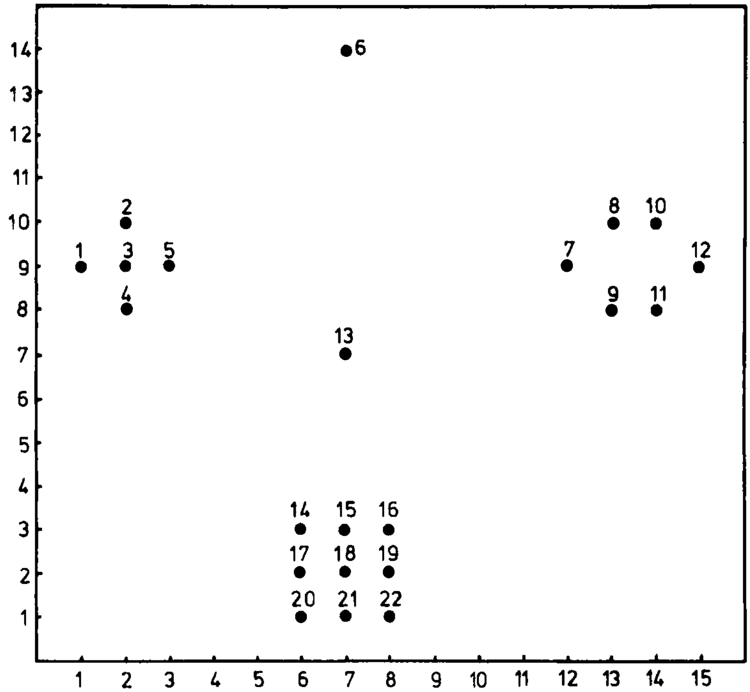

Figure 1 Data set with two intermediate objects.

cluster. Object 6 is an intermediate case, because it has substantial memberships (42% and 35%) in clusters 1 and 2, and a lower membership (23%) in cluster 3. This means that object 6 does not really belong to either cluster, but that it is still closer to clusters 1 and 2 than to cluster 3. One might say that object 6 forms a bridge between clusters 1 and 2.

The next objects (numbers 7 to 12) are quite strongly associated with cluster 2, whereas objects 14 to 22 have their largest memberships in cluster 3. Object 13 is hardest to classify, because it holds an intermediate position between clusters 1, 2, and 3. The fuzzy clustering does reflect this because the memberships of object 13 in the three clusters are nearly equal, showing no definite preference for any of them.

The main advantage of fuzzy clustering over hard clustering is that it yields much more detailed information on the structure of the data. On the other hand, this could also be considered a disadvantage, because the amount of output grows very fast with the number of objects and the number of clusters, so it may become too much to digest. Some other disadvantages are the absence of representative objects and the fact that

Table 1 A Fuzzy Clustering of the Data in Figure 1

	Membership		
Object	Cluster 1	Cluster 2	Cluster 3
1	0.87	0.06	0.07
2	0.88	0.05	0.07
3	0.93	0.03	0.04
4	0.86	0.06	0.08
5	0.87	0.06	0.07
6	0.42	0.35	0.23
7	0.08	0.82	0.10
8	0.06	0.87	0.07
9	0.06	0.86	0.08
10	0.06	0.87	0.07
11	0.06	0.86	0.08
12	0.07	0.84	0.09
13	0.36	0.27	0.37
14	0.12	0.08	0.80
15	0.08	0.07	0.85
16	0.10	0.10	0.80
17	0.08	0.06	0.86
18	0.04	0.04	0.92
19	0.07	0.07	0.86
20	0.10	0.08	0.82
21	0.07	0.06	0.87
22	0.09	0.09	0.82

fuzzy clustering algorithms are usually quite complicated and take considerable computation time. Nevertheless, we think that the fuzziness principle is very appealing because it allows a description of some of the uncertainties that often go with real data. Moreover, in Section 5.4 we will see that the list of memberships can itself be displayed graphically.

2 HOW TO USE THE PROGRAM FANNY

In order to perform fuzzy cluster analyses, we wrote the program FANNY which runs in an IBM-PC environment. The program handles data sets that either consist of interval-scaled measurements or of dissimilarities. As in the program PAM of Chapter 2, the actual algorithm only needs a collection of dissimilarities and does not depend on any measurements. When the data do consist of measurements, FANNY starts by computing the interobject distances and then uses these to construct the fuzzy clustering from.

2.1 Interactive Use and Input

The program FANNY accepts the same data sets as does PAM and it is operated in a similar way. To run FANNY, it suffices to type

A:FANNY

which yields the following dialogue:

```
FUZZY ANALYSIS

DO YOU WANT TO ENTER MEASUREMENTS ?
(PLEASE ANSWER M)
OR DO YOU PREFER TO GIVE DISSIMILARITIES ?
(THEN ANSWER D) : m

THE PRESENT VERSION OF THE PROGRAM CAN HANDLE UP TO 100
OBJECTS.
(IF MORE ARE TO BE CLUSTERED, THE ARRAYS IN THE PROGRAM
MUST BE ADAPTED)

HOW MANY OBJECTS ARE TO BE CLUSTERED ?
-------------------------------------
PLEASE GIVE A NUMBER BETWEEN 3 AND 100 : 22

CLUSTERINGS WILL BE CARRIED OUT IN K1 TO K2 CLUSTERS.
K1 SHOULD BE AT LEAST 2, AND K2 AT MOST 10.
  PLEASE ENTER K1 : 3
  PLEASE ENTER K2 : 3
```

FANNY yields fuzzy clusterings with k clusters, for all values of k between K1 and K2. Note that it is not allowed to ask for 1 cluster, which would be meaningless because all membership coefficients would equal 100%. Also, the maximal number of clusters has to be less than $n/2$, where n is the number of objects. There are numerical reasons for this restriction (having to do with the initialization of the algorithm) but it is also clear that large values of k would lead to a very extensive table of membership coefficients that would be hard to interpret.

The rest of the interactive input is identical to that of PAM:

```
THE PRESENT VERSION OF THE PROGRAM ALLOWS TO ENTER UP TO
80 VARIABLES, OF WHICH AT MOST 20 CAN BE USED IN THE ACTUAL
COMPUTATIONS.
(IF MORE ARE NEEDED, THE ARRAYS INSIDE THE PROGRAM MUST BE
ADAPTED)

WHAT IS THE TOTAL NUMBER OF VARIABLES IN YOUR DATA SET ?
--------------------------------------------------------
PLEASE GIVE A NUMBER BETWEEN 1 AND 80 : 2

HOW MANY VARIABLES DO YOU WANT TO USE IN THE ANALYSIS ?
--------------------------------------------------------
(AT MOST 2) : 2

VARIABLE TO BE USED        LABEL (AT MOST 10 CHARACTERS)
-----------------↓↓↓↓-----↓↓↓↓↓↓↓↓↓↓----------------
NUMBER :              1    x-coordina
NUMBER :              2    y-coordina

DO YOU WANT THE MEASUREMENTS TO BE STANDARDIZED ? (PLEASE
ANSWER YES)
OR NOT ? (THEN ANSWER NO) ..............................n

DO YOU WANT TO USE EUCLIDEAN DISTANCE ? (PLEASE ANSWER E)
OR DO YOU PREFER MANHATTAN DISTANCE ?
(THEN ANSWER M) ........................................e

PLEASE ENTER A TITLE FOR THE OUTPUT (AT MOST 60 CHARACTERS)
------------------------------------------------------------
Artificial data set with 22 points

DO YOU WANT LARGE OUTPUT ? (PLEASE ANSWER YES)
OR IS SMALL OUTPUT SUFFICIENT ? (THEN ANSWER NO)
(IN THE LATTER CASE NO DISSIMILARITIES ARE GIVEN) ..........n

DO YOU WANT GRAPHICAL OUTPUT (SILHOUETTES) ? PLEASE AN-
SWER YES OR NO .........................................y

DO YOU WANT TO ENTER LABELS OF OBJECTS ? PLEASE ANSWER YES
OR NO ..................................................n

DO YOU WANT TO READ THE DATA IN FREE FORMAT ?
---------------------------------------------
THIS MEANS THAT YOU ONLY HAVE TO INSERT BLANK(S) BETWEEN
NUMBERS.
```

```
(NOTE: WE ADVISE USERS WITHOUT KNOWLEDGE OF FORTRAN FOR-
MATS TO ANSWER YES.)
MAKE YOUR CHOICE (YES/NO) : y

PLEASE TYPE THE NAME OF THE FILE CONTAINING THE DATA (e.g.
A:EXAMPLE.DAT)
OR TYPE KEY IF YOU PREFER TO ENTER THE DATA BY KEYBOARD.
WHAT DO YOU CHOOSE ? a:22.dat

WHERE DO YOU WANT YOUR OUTPUT ?
-------------------------------
   TYPE CON IF YOU WANT IT ON THE SCREEN
OR TYPE PRN IF YOU WANT IT ON THE PRINTER
OR TYPE THE NAME OF A FILE (e.g. B:EXAMPLE.OUT)
(WARNING : IF THERE ALREADY EXISTS A FILE WITH THE SAME
NAME THEN THE OLD FILE WILL BE OVERWRITTEN.)
WHAT DO YOU CHOOSE ? ............................. a:22.fan

CAN MISSING DATA OCCUR IN THE MEASUREMENTS ?
PLEASE ANSWER YES OR NO : n
```

When there are missing measurements, these are treated in the same way as in PAM: The program tries to compute a complete dissimilarity matrix and then performs the actual clustering algorithm on the latter.

At the end of the interactive session, a summary of the data specifications and options appears on the screen:

```
DATA SPECIFICATIONS AND CHOSEN OPTIONS
--------------------------------------
TITLE : Artificial data set with 22 points
  THERE ARE 22 OBJECTS
  LABELS OF OBJECTS ARE NOT READ
  INPUT OF MEASUREMENTS
  SMALL OUTPUT
  GRAPHICAL OUTPUT IS WANTED (SILHOUETTES)
  CLUSTERINGS ARE CARRIED OUT IN 3 TO 3 CLUSTERS

  THERE ARE 2 VARIABLES IN THE DATA SET,
    AND 2 OF THEM WILL BE USED IN THE ANALYSIS
  THE MEASUREMENTS WILL NOT BE STANDARDIZED
  EUCLIDEAN DISTANCE WILL BE USED
  THERE ARE NO MISSING VALUES
  THE MEASUREMENTS WILL BE READ IN FREE FORMAT
```

YOUR DATA RESIDE ON FILE : a:22.dat
YOUR OUTPUT WILL BE WRITTEN ON : a:22.fan

ARE ALL THESE SPECIFICATIONS OK ? YES OR NO : y

In the present run, the data are read from a file called 22.dat, which looks like

1	9
2	10
2	9
2	8
3	9
7	14
12	9
13	10
13	8
14	10
14	8
15	9
7	7
6	3
7	3
8	3
6	2
7	2
8	2
6	1
7	1
8	1

These are the data of Figure 1. As soon as the program run is completed, a message appears on the screen and the output can be found in the file 22.fan.

2.2 Output

The output file of the preceding example is displayed in Figure 2. The first part gives some general information on the type of data and the chosen options, exactly as in PAM. Then the results for $k = 3$ are given, under the heading NUMBER OF CLUSTERS 3. Note that FANNY does not use any representative objects. Instead, the algorithm attempts to minimize the objective function

$$\sum_{v=1}^{k} \frac{\sum_{i,j=1}^{n} u_{iv}^2 u_{jv}^2 d(i, j)}{2\sum_{j=1}^{n} u_{jv}^2}$$

where u_{iv} stands for the membership of object i in cluster v. At first sight this expression looks rather formidable, but we may note a few things. To begin with, it contains nothing but the dissimilarities $d(i, j)$ and the membership coefficients that we are trying to find. This explains why interval-scaled measurements are not required. Second, the sum in the numerator ranges over all *pairs* of objects $\{i, j\}$ (instead of making the sum of the distances of the objects to some cluster center, which does not exist here). Each pair $\{i, j\}$ is encountered twice because $\{j, i\}$ also occurs, which is why the sum is divided by 2. The outer sum is over all clusters v, so the objective function that we are trying to minimize is really a kind of total dispersion. More on this may be found in Sections 4 and 5, but for now it is enough to know that the algorithm is iterative and that it stops when the objective function converges. In Figure 2 we see that the algorithm needed five iteration steps and that the final value of the objective function is 16.0742.

Next, the actual memberships are printed. In this example there are three columns (because $k = 3$). Each object is identified by a three-character label (here the computer has generated the default labels 001, ..., 022, but the user may enter other labels when appropriate). The membership coefficients of this example were already listed in Table 1 and discussed in Section 1.

Some fuzzy clusterings are more fuzzy than others. When each object has equal memberships in all clusters (hence they are all $1/k$), we have complete fuzziness. On the other hand, when each object has a membership of 1 in some cluster (and hence a zero membership in all other clusters), the clustering is entirely hard (i.e., it is a partition). To measure how hard a fuzzy clustering is, we compute *Dunn's partition coefficient* (1976) given by

$$F_k = \sum_{i=1}^{n} \sum_{v=1}^{k} u_{iv}^2/n$$

For a completely fuzzy clustering (all $u_{iv} = 1/k$) this takes on its minimal value $1/k$, whereas a partition (all $u_{iv} = 0$ or 1) yields the maximal value $F_k = 1$. The *normalized version* given by

$$F_k' = \frac{F_k - (1/k)}{1 - (1/k)} = \frac{kF_k - 1}{k - 1}$$

```
                         *********************
                         *                   *
                         *  FUZZY  ANALYSIS  *
                         *                   *
                         *********************

TITLE  : Artificial data set with 22 points

DATA SPECIFICATIONS AND CHOSEN OPTIONS
--------------------------------------
 THERE ARE    22 OBJECTS
 LABELS OF OBJECTS ARE NOT READ
 INPUT OF MEASUREMENTS
 SMALL OUTPUT
 GRAPHICAL OUTPUT IS WANTED (SILHOUETTES)
 CLUSTERINGS ARE CARRIED OUT IN     3 TO    3 CLUSTERS

 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-coordina    (POSITION :   1)
         y-coordina    (POSITION :   2)
 THE MEASUREMENTS WILL NOT BE STANDARDIZED
 EUCLIDEAN DISTANCE WILL BE USED
 THERE ARE NO MISSING VALUES
 THE MEASUREMENTS WILL BE READ IN FREE FORMAT

 YOUR DATA RESIDE ON FILE        : a:22.dat


********************************
*                              *
*  NUMBER OF CLUSTERS      3   *
*                              *
********************************

ITERATION      OBJECTIVE FUNCTION

    1                16.8363
    2                16.0871
    3                16.0744
    4                16.0742
    5                16.0742

FUZZY CLUSTERING
****************

         1        2        3

001   .8677    .0564    .0759
002   .8785    .0551    .0664
003   .9362    .0274    .0364
004   .8606    .0562    .0832
005   .8741    .0549    .0709
006   .4205    .3545    .2250
007   .0849    .8188    .0963
008   .0618    .8718    .0664
009   .0629    .8564    .0807
010   .0596    .8745    .0659
011   .0606    .8614    .0781
012   .0734    .8386    .0880
013   .3553    .2713    .3734
014   .1156    .0853    .7992
015   .0787    .0689    .8524
016   .0972    .1017    .8012
017   .0794    .0617    .8589
018   .0424    .0380    .9196
019   .0687    .0714    .8599
020   .0982    .0796    .8222
021   .0696    .0636    .8668
022   .0873    .0902    .8226

PARTITION COEFFICIENT OF DUNN =     .71
ITS NORMALIZED VERSION        =     .57

CLOSEST HARD CLUSTERING
***********************

CLUSTER NUMBER     SIZE     OBJECTS

         1           6      001 002 003 004 005 006

         2           6      007 008 009 010 011 012

         3          10      013 014 015 016 017 018 019 020 021 022
```

Figure 2 FANNY output for the example of Figure 1.

```
CLUSTERING VECTOR
*****************

                 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 3 3
                 3 3

                                      ****************
                                      *              *
                                      *  SILHOUETTES *
                                      *              *
                                      ****************

                      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 1
                      . . . . . . . . . . . . . . . . . . . . . . . . . .
                      0 0 0 1 1 2 2 2 3 3 4 4 4 5 5 6 6 6 7 7 8 8 8 9 9 0
                      0 4 8 2 6 0 4 8 2 6 0 4 8 2 6 0 4 8 2 6 0 4 8 2 6 0
CLU  NEIG  S(I)     I ++++++++++++++++++++++++++++++++++++++++++++++++++++
                      +                                                  +
 1     3    .73    003+************************************              +
 1     3    .73    002+************************************              +
 1     3    .70    001+**********************************                +
 1     3    .68    005+**********************************                +
 1     3    .64    004+********************************                  +
 1     2    .14    006+*******                                           +
                      +                                                  +
 2     3    .83    010+*****************************************         +
 2     3    .82    008+****************************************          +
 2     3    .80    011+****************************************          +
 2     3    .80    012+****************************************          +
 2     3    .79    009+***************************************           +
 2     3    .75    007+*************************************             +
                      +                                                  +
 3     1    .82    018+*****************************************         +
 3     1    .80    021+***************************************           +
 3     2    .78    019+***************************************           +
 3     1    .78    015+***************************************           +
 3     1    .78    017+***************************************           +
 3     2    .76    022+**************************************            +
 3     1    .76    020+**************************************            +
 3     2    .74    016+*************************************             +
 3     1    .73    014+************************************              +
 3     1    .11    013+*****                                             +
                      +                                                  +
                      ++++++++++++++++++++++++++++++++++++++++++++++++++++

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

      CLUSTER     1 HAS AVERAGE SILHOUETTE WIDTH    .60
      CLUSTER     2 HAS AVERAGE SILHOUETTE WIDTH    .80
      CLUSTER     3 HAS AVERAGE SILHOUETTE WIDTH    .71

      FOR THE ENTIRE DATA SET, THE AVERAGE SILHOUETTE WIDTH IS    .70

The output is on file : a:22.fan
```

Figure 2 (*Continued*)

always varies from 0 to 1, whatever value of k was chosen. In Figure 2, we see that $F_k = 0.71$ and $F_k' = 0.57$, which lie somewhere between the extremes.

We are also interested in the partition that is closest to our fuzzy clustering, especially when the output contains many membership coefficients. This closest hard clustering is obtained by assigning each object to the cluster in which it has the largest membership. In our example, objects $1, \ldots, 5$ are clearly to be assigned to cluster 1, objects $7, \ldots, 12$ to cluster 2, and objects $14, \ldots, 22$ to cluster 3. However, the intermediate objects also have to be assigned. Object 6 is put in cluster 1 because its membership in that cluster (0.42) is somewhat larger than that in the second cluster (0.35).

The assignment of object 13 is even more doubtful because its membership in the third cluster (0.37) is but slightly larger than its other memberships (0.36 and 0.27). We may conclude that the closest hard clustering does simplify the situation, but that for some objects quite a bit of information may be lost. (Another option, not implemented in FANNY, would be to delete the most fuzzy observations before constructing the hard clustering.)

In the output, the closest hard clustering is listed in the same way as the partitions obtained from PAM and CLARA. Each cluster is identified by its number, followed by its size and the labels of its objects. Also the clustering vector is printed, in which the different clusters are again encountered in ascending order, so the user may easily compare the results with the clusterings obtained from the other programs. (In fact, FANNY first numbers the hard clusters like this before printing the fuzzy clustering, where the clusters are ranked in the same way.)

The last part of the output is the silhouette plot of the hard clustering. In Section 2.2 of Chapter 2 we saw that the silhouettes can be computed for any partition, regardless of the clustering algorithm used (no representative objects are needed). The silhouette plot is constructed from the interobject dissimilarities only. In Figure 2 we see that the first cluster contains five objects with large $s(i)$ values, whereas for object 6 we find $s(i) = 0.14$, which is very low, indicating that this object is not "well-clustered". In the second column (NEIG) we see that the neighbor of object 6 is cluster 2, so object 6 is intermediate between its own cluster and cluster 2. The second cluster is very pronounced because all of its $s(i)$ values are quite large. In the third cluster we see that nine objects are well inside their cluster, whereas object 13 has a very low $s(i)$ of 0.11. [Note that object 13 is listed at the bottom because the $s(i)$ are ranked in decreasing order in each cluster.]

Below the plot some summary values are given. For each cluster the average silhouette width is listed [this is just the average of its $s(i)$ values]. They show that cluster 2 is more pronounced than clusters 1 and 3, both of which are stuck with an outlier. Finally, the average $s(i)$ of the entire data set is listed. The latter value corresponds to the blackness of the plot and assesses the overall quality of the partition. The present value (0.70) is quite high. It is also better than the average silhouette width for other values of k: If we run FANNY again for $k = 2$ we find 0.54 and for $k = 4, \ldots, 10$ we obtain still lower values. This leads us to assume that $k = 3$ is a fair choice for these data. The best average $s(i)$ is called the *silhouette coefficient* (SC). If we compare SC = 0.70 to Table 4 of Chapter 2, we may conclude that a reasonable clustering structure has been found.

If we run PAM on the same data for $k = 3$, we happen to find the same hard clustering and hence the same silhouette plot. Also for $k = 2$ they turn out to coincide, whereas for $k = 4$, 5, and 6, PAM and FANNY give different results (some better, some worse). It is often instructive to run both programs on the same data sets and to compare the outputs (preferably across different values of k).

3 EXAMPLES

Our first example is a collection of subjective dissimilarities between 12 countries, listed in Table 5 of Chapter 2. FANNY can be applied to these data in exactly the same way as PAM, except that it is no longer allowed to put $k = 1$. Figure 3 shows the output for $k = 3$, for which the algorithm needed 16 iteration steps. The fuzzy clustering does not immediately appeal to the intuition, partly because it still lists the countries in their original (alphabetical) order. None of the membership coefficients is very large, so there is a considerable amount of fuzziness, as is also reflected by the normalized partition coefficient which equals 0.11. The closest hard clustering is easier to interpret because the cluster {BEL, FRA, ISR, USA} contains the Western countries, {BRA, EGY, IND, ZAI} consists of the developing countries, and {CHI, CUB, USS, YUG} contains the Communist countries of this study. In each cluster, the countries are still listed alphabetically. Note that this hard clustering slightly differs from the partition found by PAM (Figure 8 of Chapter 2) in which EGY was added to the first cluster. Also the silhouettes are not quite the same, because in the FANNY output EGY now obtains $s(i) = -0.02$. Without EGY, the first cluster gets smaller within distances, and so USA, FRA, BEL, and ISR now obtain better $s(i)$ values than in the PAM output. The average silhouette width in the FANNY plot becomes 0.34, a little higher than the 0.33 value yielded by PAM. In either case, the structure is relatively weak.

In the silhouette plot, the countries are no longer listed in their original order but ranked according to their $s(i)$. This makes it easy to see that USA is the "best clustered" object in cluster 1, that ZAI ranks highest in cluster 2, and that CUB ranks highest in cluster 3. This is similar to the original fuzzy clustering, because USA had the highest membership (0.63) in cluster 1, ZAI had the highest membership (0.53) in cluster 2, and CUB had the highest membership (0.64) in cluster 3. In the silhouette plot we see that the "worst clustered" object is EGY, for which $s(i)$ is -0.02. Looking at the fuzzy clustering, we note that EGY was about evenly distributed over clusters 1 and 2, with memberships of 0.33 and 0.42.

Let us now try FANNY on the same extreme examples that we considered in Section 3 of Chapter 2. The first data set contains five objects coinciding with one geometrical point and three objects coinciding with

```
******************************
*                            *
*  NUMBER OF CLUSTERS     3  *
*                            *
******************************

ITERATION      OBJECTIVE FUNCTION

    1               10.7794
    2               10.0553
    3                9.9468
    4                9.9197
    5                9.9094
    6                9.9045
    7                9.9019
    8                9.9005
    9                9.8998
   10                9.8994
   11                9.8992
   12                9.8991
   13                9.8990
   14                9.8990
   15                9.8990
   16                9.8990

FUZZY CLUSTERING
****************

         1       2       3

BEL   .5984   .2400   .1616
BRA   .2602   .5293   .2104
CHI   .2239   .2823   .4939
CUB   .1582   .2058   .6360
EGY   .3318   .4172   .2509
FRA   .5997   .2313   .1690
IND   .2528   .4588   .2884
ISR   .4930   .2931   .2139
USA   .6305   .2243   .1452
USS   .1819   .2170   .6011
YUG   .2339   .2576   .5085
ZAI   .2514   .5345   .2141

PARTITION COEFFICIENT OF DUNN =    .41
ITS NORMALIZED VERSION        =    .11

CLOSEST HARD CLUSTERING
***********************

CLUSTER NUMBER    SIZE    OBJECTS

        1           4     BEL FRA ISR USA

        2           4     BRA EGY IND ZAI

        3           4     CHI CUB USS YUG

CLUSTERING VECTOR
*****************

            1  2  3  3  2  1  2  1  1  3  3  2

                                  ****************
                                  *              *
                                  *  SILHOUETTES *
                                  *              *
                                  ****************

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

 CLU  NEIG  S(I)     I   +++++++++++++++++++++++++++++++++++++++++++++++++++
                         +                                                 +
  1    2     .53     USA+**************************                        +
  1    2     .51     FRA+*************************                         +
  1    2     .50     BEL+*************************                         +
  1    2     .41     ISR+********************                              +
                         +                                                 +
  2    1     .23     ZAI+***********                                       +
  2    1     .20     BRA+**********                                        +
  2    3     .19     IND+*********                                         +
  2    1    -.02     EGY+                                                  +
                         +                                                 +
  3    2     .46     CUB+***********************                           +
  3    1     .44     USS+**********************                            +
  3    2     .36     CHI+******************                                +
  3    1     .31     YUG+***************                                   +
                         +                                                 +
                         +++++++++++++++++++++++++++++++++++++++++++++++++++

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

       CLUSTER     1 HAS AVERAGE SILHOUETTE WIDTH     .49
       CLUSTER     2 HAS AVERAGE SILHOUETTE WIDTH     .15
       CLUSTER     3 HAS AVERAGE SILHOUETTE WIDTH     .40

       FOR THE ENTIRE DATA SET, THE AVERAGE SILHOUETTE WIDTH IS    .34

The output is on file : a:country.fan
```

Figure 3 Output of FANNY for the 12 countries data, for $k = 3$.

another. Running FANNY with $k = 2$ yields Figure 4, in which only a single iteration step was necessary to reduce the objective function to 0. The fuzzy clustering is already quite hard, because the objects $1, \ldots, 5$ have a membership of approximately 1 in the first cluster and the objects 6, 7, and 8 have a membership of 1 in the second. Therefore Dunn's partition coefficient becomes 1.00, as well as its normalized version. The closest hard clustering is what it should be and the silhouette plot is the same as that of PAM. The average silhouette width attains its maximal value 1.00, so $k = 2$ is selected.

The second data set contains seven objects with zero dissimilarities to each other and one object which is far away from all of them. This kind of situation occurs when there is a far outlier, which makes all other dissimilarities look small by comparison. FANNY yields Figure 5, with again only a single iteration. The membership coefficients are already hard and they put only object 8 in the second cluster. Dunn's partition coefficient reaches 1.00, as does the normalized version. The clustering vector and silhouette plot correspond to those of PAM, so the overall average silhouette width is again $1 - \frac{1}{8} \approx 0.88$.

In the third data set all dissimilarities between objects are equal, so there is no clustering structure at all. FANNY gives an adequate description of this by putting all memberships equal to $\frac{1}{2}$ (up to numerical accuracy). This is exactly the situation in which Dunn's partition coefficient takes on its minimal value $1/k = \frac{1}{2}$, so its normalized version becomes 0. The output in Figure 6 also contains the closest hard clustering, which is rather arbitrary because no object shows a definite preference for either cluster. Although this hard clustering differs from the one found by PAM, the silhouette plot is again empty because all $s(i)$ remain 0. Therefore, the average silhouette width becomes 0, telling the user that no real clusters have been found.

In one of the first papers on fuzzy clustering, Ruspini (1970) used the data listed in Table 6 of Chapter 2. Their scatterplot (Figure 12 of Chapter 2) clearly contains four groups of points when viewed with the human eye. Group A equals $\{1, \ldots, 20\}$, whereas $B = \{21, \ldots, 43\}$, $C = \{44, \ldots, 60\}$, and $D = \{61, \ldots, 75\}$. If we apply FANNY with $k = 2$, we obtain the clusters $A \cup D$ and $B \cup C$ that were also found by PAM, so the average silhouette width $\bar{s}(k)$ again equals 0.58. Figure 7 shows the silhouette plots produced by FANNY for $k = 2, \ldots, 6$. For $k = 3$ we find A, $B \cup C$, and D, whereas PAM yielded $A \cup D$, B, and C. For $k = 4$ FANNY yields the expected clustering into A, B, C, and D. The corresponding average silhouette width $\bar{s}(k) = 0.74$ is the highest across all values of k, so $k = 4$ is a reasonable choice. Also for $k = 5$ and $k = 6$ FANNY yields hard clusterings that differ from those of PAM: For $k = 5$ the cluster B is split in two, whereas for $k = 6$ the cluster A is bisected. Although FANNY and

```
DISSIMILARITY MATRIX
--------------------

001
002      .00
003      .00      .00
004      .00      .00      .00
005      .00      .00      .00      .00
006     3.60     3.60     3.60     3.60     3.60
007     3.60     3.60     3.60     3.60     3.60      .00
008     3.60     3.60     3.60     3.60     3.60      .00      .00

********************************
*                              *
*  NUMBER OF CLUSTERS     2    *
*                              *
********************************

ITERATION     OBJECTIVE FUNCTION

    1                  .0000

FUZZY CLUSTERING
****************

        1       2

001  .9998   .0002
002  .9999   .0001
003  .9999   .0001
004  .9999   .0001
005  .9999   .0001
006  .0000  1.0000
007  .0000  1.0000
008  .0000  1.0000

PARTITION COEFFICIENT OF DUNN =  1.00
ITS NORMALIZED VERSION        =  1.00

CLOSEST HARD CLUSTERING
***********************

CLUSTER NUMBER    SIZE    OBJECTS

         1          5     001 002 003 004 005

         2          3     006 007 008

CLUSTERING VECTOR
*****************

            1  1  1  1  1  2  2  2

                                              ****************
                                              *              *
                                              *  SILHOUETTES *
                                              *              *
                                              ****************

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

 CLU   NEIG   S(I)    I +++++++++++++++++++++++++++++++++++++++++++++++++++
                          +                                                 +
  1      2    1.00    001+**************************************************+
  1      2    1.00    002+**************************************************+
  1      2    1.00    003+**************************************************+
  1      2    1.00    004+**************************************************+
  1      2    1.00    005+**************************************************+
                          +                                                 +
  2      1    1.00    006+**************************************************+
  2      1    1.00    007+**************************************************+
  2      1    1.00    008+**************************************************+
                          +                                                 +
                          +++++++++++++++++++++++++++++++++++++++++++++++++++

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

      CLUSTER    1 HAS AVERAGE SILHOUETTE WIDTH  1.00
      CLUSTER    2 HAS AVERAGE SILHOUETTE WIDTH  1.00

      FOR THE ENTIRE DATA SET, THE AVERAGE SILHOUETTE WIDTH IS  1.00
```

Figure 4 FANNY output for a data set with two tight clusters.

```
DISSIMILARITY MATRIX
--------------------

001
002      .00
003      .00     .00
004      .00     .00     .00
005      .00     .00     .00     .00
006      .00     .00     .00     .00     .00
007      .00     .00     .00     .00     .00     .00
008     3.60    3.60    3.60    3.60    3.60    3.60    3.60

*******************************
*                             *
*  NUMBER OF CLUSTERS     2   *
*                             *
*******************************

ITERATION     OBJECTIVE FUNCTION

    1                .0000

FUZZY CLUSTERING
****************

        1       2

001   .9998   .0002
002   .9999   .0001
003   .9999   .0001
004   .9999   .0001
005   .9999   .0001
006  1.0000   .0000
007  1.0000   .0000
008   .0000  1.0000

PARTITION COEFFICIENT OF DUNN =  1.00
ITS NORMALIZED VERSION        =  1.00

CLOSEST HARD CLUSTERING
***********************

CLUSTER NUMBER    SIZE    OBJECTS

      1             7     001 002 003 004 005 006 007

      2             1     008

CLUSTERING VECTOR
*****************

            1  1  1  1  1  1  1  2

                                  ****************
                                  *              *
                                  *  SILHOUETTES *
                                  *              *
                                  ****************

                         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 1
                         . . . . . . . . . . . . . . . . . . . . . . . . . .
                         0 0 0 1 1 2 2 2 3 3 4 4 4 5 5 6 6 6 7 7 8 8 8 9 9 0
                         0 4 8 2 6 0 4 8 2 6 0 4 8 2 6 0 4 8 2 6 0 4 8 2 6 0
CLU  NEIG  S(I)     I  +++++++++++++++++++++++++++++++++++++++++++++++++++
                        +                                                 +
 1    2   1.00     001+***************************************************+
 1    2   1.00     002+***************************************************+
 1    2   1.00     003+***************************************************+
 1    2   1.00     004+***************************************************+
 1    2   1.00     005+***************************************************+
 1    2   1.00     006+***************************************************+
 1    2   1.00     007+***************************************************+
                        +                                                 +
 2    1    .00     008+                                                 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 1
                         . . . . . . . . . . . . . . . . . . . . . . . . . .
                         0 0 0 1 1 2 2 2 3 3 4 4 4 5 5 6 6 6 7 7 8 8 8 9 9 0
                         0 4 8 2 6 0 4 8 2 6 0 4 8 2 6 0 4 8 2 6 0 4 8 2 6 0

      CLUSTER    1 HAS AVERAGE SILHOUETTE WIDTH  1.00
      CLUSTER    2 HAS AVERAGE SILHOUETTE WIDTH   .00
      FOR THE ENTIRE DATA SET, THE AVERAGE SILHOUETTE WIDTH IS    .88
```

Figure 5 FANNY output for a data set with an extreme outlier.

```
DISSIMILARITY MATRIX
--------------------

001
002      3.60
003      3.60      3.60
004      3.60      3.60      3.60
005      3.60      3.60      3.60      3.60
006      3.60      3.60      3.60      3.60      3.60
007      3.60      3.60      3.60      3.60      3.60      3.60
008      3.60      3.60      3.60      3.60      3.60      3.60      3.60

********************************
*                              *
*  NUMBER OF CLUSTERS      2   *
*                              *
********************************

ITERATION      OBJECTIVE FUNCTION

    1                6.5880
    2                6.3721
    3                6.3211
    4                6.3064
    5                6.3019
    6                6.3006
    7                6.3002
    8                6.3001
    9                6.3000
   10                6.3000
   11                6.3000

FUZZY CLUSTERING
****************

         1       2

001  .5001   .4999
002  .5000   .5000
003  .4997   .5003
004  .4992   .5008
005  .4999   .5001
006  .5000   .5000
007  .5002   .4998
008  .5006   .4994

PARTITION COEFFICIENT OF DUNN =    .50
ITS NORMALIZED VERSION        =    .00

CLOSEST HARD CLUSTERING
***********************

CLUSTER NUMBER     SIZE     OBJECTS

         1           4      001  006  007  008

         2           4      002  003  004  005

CLUSTERING VECTOR
*****************

            1  2  2  2  2  1  1  1

                                   *****************
                                   *               *
                                   *  SILHOUETTES  *
                                   *               *
                                   *****************

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

 CLU  NEIG  S(I)       I +++++++++++++++++++++++++++++++++++++++++++++++++++
                         +                                                 +
  1    2    .00      001+                                                  +
  1    2    .00      006+                                                  +
  1    2    .00      007+                                                  +
  1    2    .00      008+                                                  +
                         +                                                 +
  2    1    .00      002+                                                  +
  2    1    .00      003+                                                  +
  2    1    .00      004+                                                  +
  2    1    .00      005+                                                  +
                         +                                                 +
                         +++++++++++++++++++++++++++++++++++++++++++++++++++
                         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 1
                         . . . . . . . . . . . . . . . . . . . . . . . . . .
                         0 0 0 1 1 2 2 2 3 3 4 4 4 5 5 6 6 6 7 7 8 8 8 9 9 0
                         0 4 8 2 6 0 4 8 2 6 0 4 8 2 6 0 4 8 2 6 0 4 8 2 6 0

        CLUSTER     1 HAS AVERAGE SILHOUETTE WIDTH     .00
        CLUSTER     2 HAS AVERAGE SILHOUETTE WIDTH     .00

        FOR THE ENTIRE DATA SET, THE AVERAGE SILHOUETTE WIDTH IS     .00
```

Figure 6 FANNY output for a data set without clusters.

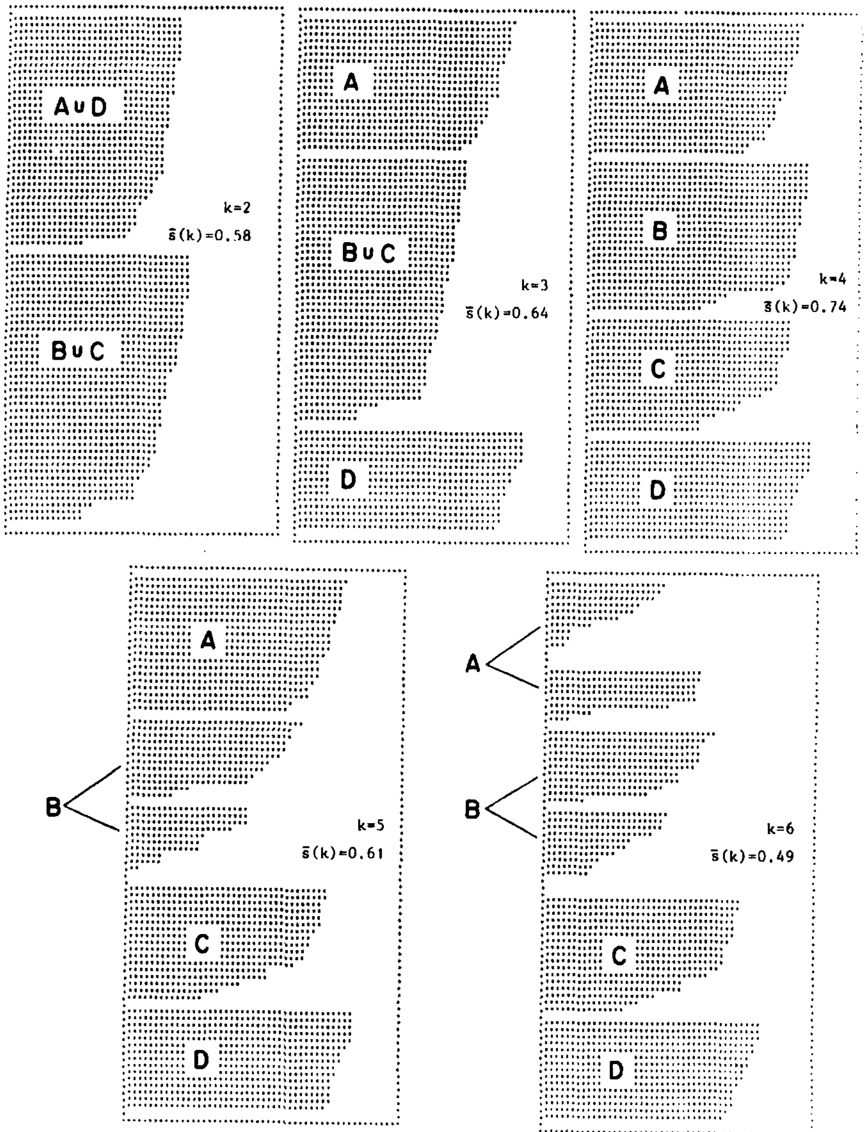

Figure 7 Silhouette plots obtained by running FANNY on the Ruspini data, for k ranging from 2 to 6.

PAM operate quite differently, they both select the natural clustering with $k = 4$ on the basis of the $\bar{s}(k)$.

*4 MORE ON THE ALGORITHM AND THE PROGRAM

4.1 Description of the Algorithm

The fuzzy clustering technique used in this program aims at the minimization of the objective function

$$C = \sum_{v=1}^{k} \frac{\sum_{i,j=1}^{n} u_{iv}^2 u_{jv}^2 d(i,j)}{2\sum_{j=1}^{n} u_{jv}^2} \tag{1}$$

in which the $d(i, j)$ represent the given distances (or dissimilarities) between objects i and j, whereas u_{iv} is the unknown membership of object i to cluster v. Note that each term appears two times in the multiple sum. The factor 2 in the denominator compensates for this duplicity, while assuring the coherence with other algorithms (see Section 5.1). The membership functions are subject to the constraints

$$u_{iv} \geq 0 \quad \text{for } i = 1, \ldots, n;\ v = 1, \ldots, k \tag{2}$$

$$\sum_{v} u_{iv} = 1 \quad \text{for } i = 1, \ldots, n \tag{3}$$

expressing that memberships cannot be negative and that each object has a constant total membership, distributed over the different clusters; by convention this total membership is normalized to 1.

The remainder of this section is a technical description of the numerical algorithm used to minimize (1). Readers wishing to skip this material may continue directly with Section 5.

A characterization of the local optima of (1) can be found from the Lagrange equation

$$L = \sum_{v} \frac{\sum_{ij} u_{iv}^2 u_{jv}^2 d(i,j)}{2\sum_{j} u_{jv}^2} - \sum_{j} \gamma_j \left(\sum_{v} u_{jv} - 1 \right) - \sum_{j} \sum_{v} \psi_{jv} u_{jv} \tag{4}$$

in which the γ_j and the ψ_{jv} are known as Lagrange multipliers. Its derivatives with respect to the membership variables are

$$\frac{\partial L}{\partial u_{iv}} = \frac{2u_{iv} \sum_j u_{jv}^2 d(i,j)}{\sum_j u_{jv}^2} - \frac{u_{iv} \sum_h \sum_j u_{jv}^2 u_{hv}^2 d(h,j)}{\left(\sum_j u_{jv}^2\right)^2} - \gamma_i - \psi_{iv} \tag{5}$$

Taking into account the objective function and the constraints (2) and (3) of the original minimization problem, we can write down the corresponding Kuhn and Tucker conditions:

$$\psi_{iv} \geq 0 \tag{6}$$

$$\frac{\partial L}{\partial u_{iv}} = 0 \tag{7}$$

$$u_{iv}\psi_{iv} = 0 \tag{8}$$

The relations (7) and (5) can be combined into

$$a_{iv}u_{iv} - \gamma_i - \psi_{iv} = 0 \tag{9}$$

in which

$$a_{iv} = \frac{2\Sigma_j u_{jv}^2 d(i, j)}{\Sigma_j u_{jv}^2} - \frac{\Sigma_h \Sigma_j u_{hv}^2 u_{jv}^2 d(h, j)}{\left(\Sigma_j u_{jv}^2\right)^2} \tag{10}$$

The function a_{iv} can be either positive, zero, or negative. Considering first the nonzero cases, one can divide all terms of Eq. (9) by a_{iv} and upon summation for all values of v, taking into account Eq. (3), one finds

$$\gamma_i = \frac{1 - \Sigma_v(\psi_{iv}/a_{iv})}{\Sigma_v(1/a_{iv})}$$

Replacing this term in (9), one gets

$$u_{iv} = \frac{1/a_{iv}}{\Sigma_w(1/a_{iw})} + \frac{\psi_{iv}}{a_{iv}} - \frac{\Sigma_w(\psi_{iw}/a_{iw})}{a_{iv}\Sigma_w(1/a_{iw})} \tag{11}$$

Considering the conditions (6), the relations (11) can take one of two forms for each object i:

1. $\psi_{iv} = 0$ for $v = 1, \ldots, k$ so that

$$u_{iv} = \frac{1/a_{iv}}{\Sigma_w(1/a_{iw})} \tag{12}$$

Taking into account constraint (2), this set of solutions is only possible for each object i if

$$\frac{1/a_{iv}}{\Sigma_w(1/a_{iw})} \geq 0 \quad \text{for } v = 1, \ldots, k \tag{13}$$

If this condition is not fulfilled, the preceding solution is not valid and we have to consider the alternative form:

2. $\psi_{iv} > 0$ for at least some v. According to (8) and (11) this solution implies that

$$u_{iv} = 0 = \frac{1/a_{iv}}{\Sigma_w(1/a_{iw})} + \frac{\psi_{iv}}{a_{iv}} - \frac{\Sigma_w(\psi_{iw}/a_{iw})}{a_{iv}\Sigma_w(1/a_{iw})} \tag{14}$$

for at least some v. Because Eq. (3) must also be satisfied, it is clear that this solution is not valid for all v of one object i. Hence let us define the partition:

$$\begin{aligned} V- &= \{v;\ u_{iv} = 0\} \\ V+ &= \{v;\ u_{iv} > 0 \Rightarrow \psi_{iv} = 0\} \neq \varnothing \end{aligned} \tag{15}$$

If one can exclude the case where $1/a_{iv} = 0$, solution (14) implies

$$\psi_{iv} = \frac{\Sigma_w(\psi_{iw}/a_{iw})}{\Sigma_w(1/a_{iw})} - \frac{1}{\Sigma_w(1/a_{iw})} \quad \text{for } v \in V- \tag{16}$$

In Eq. (16) the right-hand term is independent of v; hence all ψ_{iv} are identical for all $v \in V-$ (but may differ for different i). Solution (16) can hence be written:

$$\psi_{iv} = -\frac{1}{\Sigma_{w \in V+}(1/a_{iw})} \quad \text{for } v \in V- \tag{17}$$

As for $v \in V+$, (11) becomes

$$u_{iv} = \frac{1/a_{iv}}{\Sigma_w(1/a_{iw})} - \frac{\Sigma_w(\psi_{iw}/a_{iw})}{a_{iv}\Sigma_w(1/a_{iw})} \tag{18}$$

Introducing (17) in (18) provides, after some transformation,

$$u_{iv} = \frac{1/a_{iv}}{\Sigma_{w \in V+}(1/a_{iw})} \tag{19}$$

which is analogous to (12).

Equations (15), (17), and (19) are, according to Kuhn and Tucker, necessary conditions for a local minimum, but in general they are not sufficient conditions. Actually it can be seen that these equations also describe local maxima and saddle points.

In fact, the local minima will be reached for

$$u_{iv} = 0 \quad \text{for } v \in V- \tag{20}$$

and

$$u_{iv} = \frac{1/a_{iv}}{\sum_{w \in V+}(1/a_{iw})} \quad \text{for } v \in V+ \tag{21}$$

where

$$V- = \left\{ v;\ \frac{1/a_{iv}}{\sum_w (1/a_{iw})} \leq 0 \right\} \tag{22}$$

and

$$V+ = \left\{ v;\ \frac{1/a_{iv}}{\sum_w (1/a_{iw})} > 0 \right\} \tag{23}$$

As for the degenerate case where $1/a_{iv} = 0$, it can be seen from (10) that it corresponds to $\sum_i u_{iv}^2 = 0$, meaning that some cluster v has no membership at all: Again it can be shown that this solution is not a minimum.

The fact that (13) is not always satisfied implies that a_{iv} can be either positive or negative. Therefore one must also consider the possibility that a_{iv} be zero, which invalidates expression (11). However, this particular case can be solved by regarding it as the limiting solution for a_{iv} being any positive or negative small value, and it can be seen that both limits render $u_{iv} = 1$.

The system of Eqs. (20) and (21) does not provide a straightforward analytical solution to the problem of minimizing the objective function (1), because the right-hand term of (21) still contains all the unknowns. However, it does provide a way to find the solutions iteratively. Indeed, having some initial values for all u_{iv}, one can compute all a_{iv} according to (10) and calculate new u_{iv} from (20) and (21), and so on. Hence the iterative algorithm can take the following form:

A1 Initialize the membership functions as

$$^0u_{iv} \quad \text{for all } i = 1, \ldots, n \quad \text{and all } v = 1, \ldots, k$$

taking into account constraints (2) and (3). Calculate the objective function ${}^{0}C$ by (1). (Note that the left superscripts stand for the number of the iteration step.)

A2 Compute for each $i = 1, \ldots, n$ the following quantities:
A2.1 Compute for each $v = 1, \ldots, k$

$$
{}^{m}a_{iv} = \frac{2\left(\sum_{j=1}^{i-1} {}^{m+1}u_{jv}^{2}\, d(i, j) + \sum_{j=i}^{n} {}^{m}u_{jv}^{2}\, d(i, j)\right)}{\sum_{j=1}^{i-1} {}^{m+1}u_{jv}^{2} + \sum_{j=i}^{n} {}^{m}u_{jv}^{2}}
$$

$$
- \frac{\begin{array}{c}\sum_{j=1}^{i-1}\sum_{h=1}^{i-1} {}^{m+1}u_{jv}^{2}\, {}^{m+1}u_{hv}^{2}\, d(i, j) + \sum_{j=1}^{i-1}\sum_{h=i}^{n} {}^{m+1}u_{jv}^{2}\, {}^{m}u_{hv}^{2}\, d(i, j) \\ + \sum_{j=i}^{n}\sum_{h=1}^{i-1} {}^{m}u_{jv}^{2}\, {}^{m+1}u_{hv}^{2}\, d(i, j) + \sum_{j=i}^{n}\sum_{h=i}^{n} {}^{m}u_{jv}^{2}\, {}^{m}u_{hv}^{2}\, d(i, j)\end{array}}{\sum_{j=1}^{i-1} {}^{m+1}u_{jv}^{2} + \sum_{j=i}^{n} {}^{m}u_{jv}^{2}}
$$

A2.2 Compute for each $v = 1, \ldots, k$:

$$
A_{v} = \frac{1/{}^{m}a_{iv}}{\sum_{w}\left(1/{}^{m}a_{iw}\right)}
$$

A2.2.1 if $A_{v} \leq 0 \Rightarrow V- = V- \cup \{v\}$
A2.2.2 if $A_{v} > 0 \Rightarrow V+ = V+ \cup \{v\}$
A2.3 Put for all $v \in V-$

$$
{}^{m+1}u_{iv} = 0
$$

A2.4 Compute for all $v \in V+$

$$
{}^{m+1}u_{iv} = \frac{1/{}^{m}a_{iv}}{\sum_{w \in V+}\left(1/{}^{m}a_{iw}\right)}
$$

A2.5 Put $V- = V+ = \varnothing$ and restart from A2.1 with the next i.
A3 Calculate the new objective function value ${}^{m+1}C$ by (1). If $({}^{m}C/{}^{m+1}C - 1) < \varepsilon$, then go to A2; otherwise stop.

The rather cumbersome calculations of A2.1 can be simplified by keeping track of intermediate results and by limiting the computations to the updating of the partial sums at each step.

The preceding algorithm has always shown good convergence performance. However, if ever some convergence problems might arise, it is still possible to improve on the iteration steps by applying some adequate converging method, such as the steepest descent method, the Newton–Raphson method, or a similar technique.

Any fuzzy solution found by the preceding algorithm satisfies the general constraints (2) and (3). The corresponding constraints for a hard solution, obtained by replacing the u_{iv} by w_{iv}, would require (2) and (3) to be changed into

$$w_{iv} = 0 \text{ or } 1 \quad \text{for } i = 1, \ldots, n \quad \text{and } v = 1, \ldots, k$$

$$\sum_{v} w_{iv} = 1 \quad \text{for } i = 1, \ldots, n$$

In a fuzzy clustering the membership coefficients of each object can all be strictly positive, provided their sum over all clusters is 1. On the contrary, in a hard cluster solution each object must have one and only one nonzero membership coefficient, which necessarily takes the value 1.

The hard clustering solutions are hence limiting cases of fuzzy ones. How far off a fuzzy solution is from a hard clustering can be evaluated by Dunn's partition coefficient (1976), which is defined as the sum of squares of all the membership coefficients, divided by the number of objects, i.e.,

$$F_k(U) = \sum_{i=1}^{n} \sum_{v=1}^{k} u_{iv}^2/n \tag{24}$$

in which U is the matrix of all memberships:

$$U = \begin{array}{c} \\ 1 \\ \vdots \\ n \end{array} \begin{array}{c} 1 \quad \cdots \quad k \\ \left(\begin{array}{ccc} & & \\ & u_{iv} & \\ & & \end{array} \right) \end{array}$$

It can be seen that for a partition (u_{iv} restricted to 0s and 1s) $F_k(U)$ gets the maximum value of 1, whereas it takes on the minimum value of $1/k$ when all $u_{iv} = 1/k$. This coefficient can thus be normalized to vary from 1 (hard clusters) to 0 (entirely fuzzy), independently of the number of clusters, by the following transformation:

$$F_k'(U) = \frac{F_k(U) - (1/k)}{1 - (1/k)} = \frac{kF_k(U) - 1}{k - 1} \tag{25}$$

This normalized coefficient has sometimes been called "nonfuzziness index" (Roubens, 1982).

It is often interesting to translate a fuzzy clustering allocation to a hard one. By definition, the *closest hard clustering corresponding to a fuzzy one* is given by putting $w_{iq} = 1$ for the cluster q with largest u_{iv}. In case of ties an (arbitrary) choice is made.

This transformation can be used to compare fuzzy solutions to hard ones and for the evaluation of the fuzziness of solutions (see Section 5.3). It should however be appreciated that the fact that no optimal fuzzy cluster remains empty does not preclude the preceding fuzzy-to-hard transformation to produce less than k hard clusters. This occurrence has actually been observed in some special cases.

4.2 Structure of the Program

The structure of the program FANNY is very similar to that of the former programs; the main difference is the actual clustering subroutine, which in the present case is called FUZZY. This subroutine constructs an initial allocation of the objects to the k clusters and it performs the iterations until convergence is reached. It calculates and prints Dunn's partition coefficient (24) as well as its normalized version (25).

The numerical output subroutine is CADDY, which gives both the fuzzy clustering output and the closest hard clustering. It first renumbers the clusters as in the other programs.

The optional graphical output is made by the subroutine FYGUR, which yields the silhouettes (see Chapter 2) that are based on the closest hard clustering.

All these subroutines are called from the main program, which is otherwise identical to that of the other programs except for a slightly different way of storing the dissimilarities in a one-dimensional array.

Table 2 lists some computation times on an IBM/XT with an 8087 accelerator. The same computer and the same data were used as in Table 7 of Chapter 2. It seems that the extra information provided by FANNY (as

Table 2 Computation Times (in minutes) on an IBM / XT with 8087 Accelerator, for Some Randomly Generated Data

Objects	Variables	Clusters	Time
20	2	5	1.35
40	2	5	6.88
60	2	5	11.95
80	2	5	34.57
100	2	5	27.30

compared to the hard clustering yielded by PAM) is obtained at the expense of a substantial increase of computation time. Note that the computation time for 100 objects is less than that for 80 objects, because FANNY needed only 14 iteration steps in the experiment with 100 objects, whereas 29 iterations were necessary to cluster the smaller data set with 80 objects. In general, the computation time of FANNY is hard to predict because the number of iteration steps depends on the actual data.

*5 RELATED METHODS AND REFERENCES

5.1 Fuzzy k-Means and the MND2 Method

One of the first fuzzy clustering techniques was the well known *fuzzy k-means* method proposed by Dunn (1974) and Bezdek (1974). This method is based on the minimization of the objective function

$$\sum_i \sum_v u_{iv}^2 \|x_i - m_v\|^2 = \sum_i \sum_v u_{iv}^2 \sum_{f=1}^{p} (x_{if} - m_{vf})^2 \tag{26}$$

in which m_v is the center of the cluster v, calculated for each variable f by

$$m_{vf} = \frac{\sum_i u_{iv}^2 x_{if}}{\sum_i u_{iv}^2} \tag{27}$$

and the norm $\|x_i - m_v\|$ is the Euclidean distance between an object x_i and the center of cluster v. Part of the popularity of this approach certainly rests on the fact that it generalizes the classical k-means approach of hard clustering (described in Section 5.3 of Chapter 2).

Implicit in the fuzzy k-means approach is the assumption that the different objects are given by means of coordinates in a p-dimensional space. This is a restrictive condition in comparison with FANNY, for which no such representation of the objects is needed, because only the distances or dissimilarities between objects are required.

Note that, when the data do consist of measurements, it is possible to make a direct comparison between the two methods. In order to do so it is sufficient to replace m_v in (26) by its value from (27). Hence

$$\sum_i \sum_v u_{iv}^2 \sum_f (x_{if} - m_{vf})^2$$

$$= \sum_i \sum_v u_{iv}^2 \sum_f \left(x_{if} - \frac{\Sigma_j u_{jv}^2 x_{jf}}{\Sigma_j u_{jv}^2} \right)^2$$

$$= \sum_v \sum_f \left[\sum_i u_{iv}^2 x_{if}^2 - 2 \frac{\Sigma_i u_{iv}^2 x_{if} \Sigma_j u_{jv}^2 x_{jf}}{\Sigma_j u_{jv}^2} + \frac{(\Sigma_j u_{jv}^2 x_{jf})^2}{\Sigma_j u_{jv}^2} \right]$$

$$= \sum_v \sum_f \left[\sum_j u_{jv}^2 x_{jf}^2 - \frac{(\Sigma_j u_{jv}^2 x_{jf})^2}{\Sigma_j u_{jv}^2} \right] = \sum_v \sum_f \frac{\Sigma_j u_{jv}^2 \Sigma_i u_{iv}^2 x_{if}^2 - \Sigma_i \Sigma_j u_{iv}^2 u_{jv}^2 x_{if} x_{jf}}{\Sigma_j u_{jv}^2}$$

$$= \sum_v \sum_f \frac{\Sigma_i \Sigma_j u_{iv}^2 u_{jv}^2 (x_{if} - x_{jf})^2}{2\Sigma_j u_{jv}^2} = \sum_v \frac{\Sigma_i \Sigma_j u_{iv}^2 u_{jv}^2 \|x_i - x_j\|^2}{2\Sigma_j u_{jv}^2} \tag{28}$$

The last expression is exactly the objective function of FANNY, apart from the fact that the distances are squared, which is a variant of the current L_1 form of our program. This demonstrates the possible equivalence between FANNY and fuzzy k-means, for objects described by a set of measurements. (It would suffice to enter a dissimilarity matrix consisting of *squared* Euclidean distances into FANNY, without changing the program.)

The equivalence of the two techniques not only concerns their objective functions: The algorithms can be so constructed that each iteration produces the same result with both programs. The only difference is that within each iteration cycle FANNY performs a loop over all pairs of objects, whereas fuzzy k-means loops for each object over the measurement variables. Because the number of objects is usually larger than twice the number of variables, the FANNY algorithm will be somewhat slower than fuzzy k-means, but with the computation speed of modern computers this should only be a minor drawback.

Another program that shows some resemblance to FANNY is the *MND2* algorithm of Roubens (1978), with the objective function

$$\sum_v \sum_{ij} u_{iv}^2 u_{jv}^2 \, d(i, j) \tag{29}$$

Although (29) looks similar to the FANNY objective (1), it is their difference that is most important and that explains why FANNY was preferred to MND2. The importance of this difference, the denominator $\Sigma_j u_{jv}^2$, can best be explained by reference to a hard clustering: In that case, with all u_{jv} either 0 or 1, this denominator is equal to the number of objects in each cluster. Hence FANNY minimizes a fuzzy equivalent of a sum of error functions for each cluster, whereas MND2 does the same with a sum of products of the number of objects times the error functions. It can be shown that the MND2 minimization tends to bias the result toward clusters of more equal number of objects and error functions than does FANNY. This systematic bias is the main reason for preferring FANNY to MND2.

5.2 Why Did We Choose FANNY?

As seen above, the exponent of the distance term in the objective function can be 1 or 2. Putting the distance exponent equal to 2 corresponds to fuzzy k-means, so the latter is an L_2 method. (Already the classical k-means approach was of the least squares type because its objective was to minimize an error sum of squares.) Putting the exponent of the distance equal to 1 yields FANNY, which is therefore a fuzzy L_1 method. It was shown by Trauwaert (1987) that our fuzzy L_1 method has some definite advantages over fuzzy L_2: a lower sensitivity to outliers or otherwise erroneous data and a better recognition of nonspherical clusters.

Not only the exponents of the distances can vary, but also those of the membership functions u_{iv} and u_{jv}. This possibility was already implemented by Bezdek (1974) in fuzzy k-means and by Libert and Roubens (1982) in MNDR. The same potential exists of course for FANNY; attention must, however, be paid to the fact that the described algorithm needs membership exponents strictly larger than 1 in order to converge properly. This is also the case for the other two algorithms. Recent convergence results for fuzzy k-means with a general membership exponent are given by Hathaway and Bezdek (1988).

Changing the exponent of the memberships in FANNY has some influence on the allocation of the objects in the clustering, although it is not easy to describe this effect. What is certain is that decreasing the exponent will yield higher values of the largest membership functions, i.e., the clusters will appear less fuzzy. However, because the aim of fuzzy clustering is to use the particular features of fuzziness, one should not go too far in that direction. Moreover, exponent values near 1 cause the algorithm to converge more slowly. Therefore, exponents equal to 2 seem to be a reasonable choice, as is confirmed by actual clustering analyses.

5.3 Measuring the Amount of Fuzziness

It has been seen in Section 4.1 how a fuzzy clustering solution can be transformed into a hard one, allowing the results to be compared to partitions obtained by hard clustering methods, such as the technique described in Chapter 2.

However, by this transformation, a lot of information contained in the membership functions gets lost. This fuzzy information could be of value in all sorts of comparisons between fuzzy solutions, such as:

Comparison between different fuzzy algorithms.

Comparison between solutions for various numbers of clusters.

Comparisons between:
- membership function exponents
- distance function exponents
- varying positions of one or a few objects
- varying choices of distances

and so on.

Although the effect of some of these variations could show up in the hard clusterings, this representation is not as sensitive as the fuzzy one. In the latter case, with membership values for each object and each cluster, the problem is rather to summarize the situation. For this purpose, different coefficients have been proposed by various authors. One of these coefficients was already defined in Section 4.1. However, considering the $k \times n$ elements of matrix U, the reduction to only one value $[F_k(U)]$ is probably too drastic a condensation to be able to express the most important features.

Taking into account a second function of U, defined as

$$D_k(U) = \sum_i \sum_v (w_{iv} - u_{iv})^2/n, \tag{30}$$

one obtains a wider basis for the evaluation of a fuzzy clustering. In this definition, W is the closest hard representation of the fuzzy U, as derived in Section 4.1. Hence $D_k(U)$ represents the average squared error of a fuzzy clustering with respect to the closest hard clustering. It can be shown to vary between 0 (hard clustering) and $1 - (1/k)$ (completely fuzzy).

Normalizing this function (in the same way as Dunn's partition coefficient), we obtain

$$D_k'(U) = D_k(U)/(1 - (1/k)) = k D_k(U)/(k - 1) \tag{31}$$

It is possible to represent each fuzzy clustering as a point in the (D_k', F_k') system shown in Figure 8 (Trauwaert, 1988). These points have to stay inside a certain region. The points on the lower boundary correspond to a

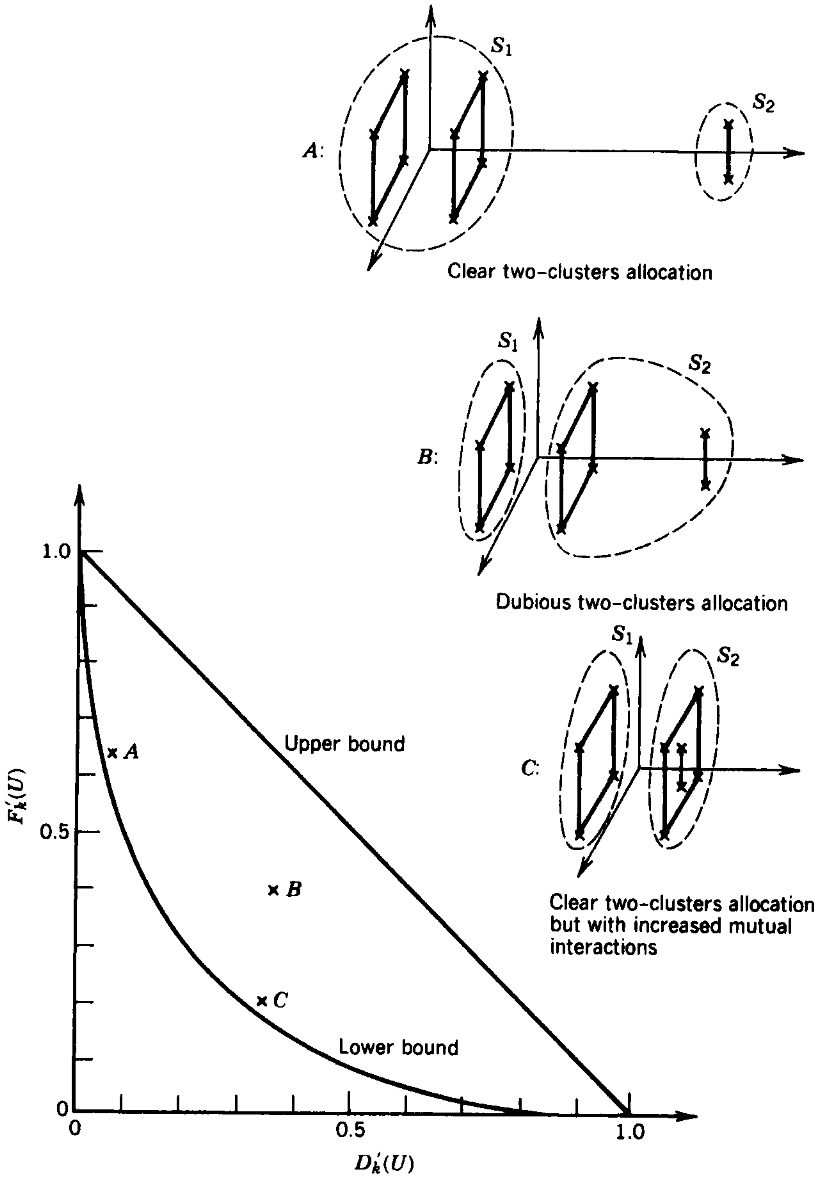

Figure 8 Representation of some fuzzy clusterings (with $n = 10$ and $k = 2$) in a partition coefficient diagram.

membership matrix of the form

$$U = \begin{array}{c} 1 \\ \vdots \\ n \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & \cdots & k-1 & k \end{array} \\ \begin{pmatrix} \dfrac{\beta}{k-1} & \cdots & \dfrac{\beta}{k-1} & 1-\beta \end{pmatrix} \end{array} \quad \text{with } 0 \le \beta \le (k-1)/k \tag{32}$$

where each object is primarily attributed to one of the clusters, with a membership $(1 - \beta)$, and for all other clusters it has a lower membership of $\beta/(k-1)$. Points on the upper boundary correspond to a membership matrix of the form

$$U = \begin{array}{c} 1 \\ \vdots \\ n \end{array} \begin{array}{c} \begin{array}{ccc} 1 & \cdots\cdots\cdots & k \end{array} \\ \begin{pmatrix} \overbrace{\dfrac{1}{v}\ \dfrac{1}{v}\ \cdots\ \dfrac{1}{v}}^{v} & 0 & \cdots & 0 \end{pmatrix} \end{array} \tag{33}$$

indicating that each object is uniformly distributed over some clusters and not at all present in the others.

In terms of the quality of the clustering, a point at or near the upper boundary indicates that, apart from the well-defined clusters, some objects are bridges or outliers to a number of clusters. On the other hand, a point at or near the lower boundary means that all objects show a definite preference for one cluster, with however some constant membership to all other clusters. It is as if some background noise is added to an otherwise fairly well-defined clustering. This background noise could be due to the algorithm itself.

Moreover, a point near the upper left edge of the diagram corresponds to a fairly hard clustering, whereas on the lower right edge the clusters are completely fuzzy.

From this two-dimensional representation one gets a much better image of a fuzzy clustering than from $F_k'(U)$ alone. It appears clearly from this diagram that the higher values of $F_k'(U)$ do not necessarily indicate a better classification.

A variant of the nonfuzziness index (25) was provided by Libert and Roubens (1982). It is defined as

$$L_k(U) = \Big(\sum_i u_{iq}/n + \min_i u_{iq}\Big)/2$$

with normalized version

$$L'_k(U) = \frac{L_k(U) - (1/k)}{1 - (1/k)} = \frac{kL_k(U) - 1}{k - 1}$$

in which q is the cluster for which u_{iv} is maximal.

5.4 A Graphical Display of Fuzzy Memberships

The primary output of FANNY is a list of membership coefficients, which may contain many numbers and therefore becomes difficult to visualize. To summarize this mass of information, Rousseeuw et al. (1989) proposed computing the principal components of the membership coefficients. One merely applies a standard principal components program (as available in SPSS, BMDP, SAS, etc.) to the memberships, in the same way that it is usually applied to measurements. The number of nondegenerate principal components is the number of fuzzy clusters minus 1 (because the sum of memberships is a constant, for each object). For instance, applying FANNY with $k = 3$ to the 12 countries data yielded the fuzzy memberships in Figure 3. Because there are three fuzzy clusters, we obtain two principal components, the scores of which are plotted in Figure 9. The vertical

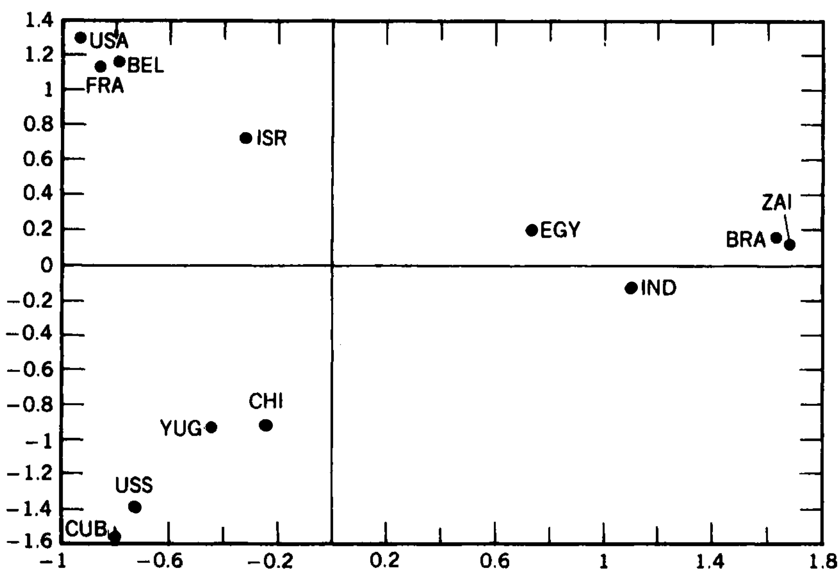

Figure 9 Principal components of memberships of 12 countries in three fuzzy clusters.

component can be interpreted as the countries' political orientation, whereas the horizontal component seems to correspond to the degree of industrialization. In this plot we clearly see the three clusters (consisting of Capitalist, developing, and Communist countries) as well as the fact that Egypt holds an intermediate position.

Because this method is based on the memberships only, the original data need not consist of measurements (indeed, the 12 countries example was based on dissimilarities). The idea to compute the principal components of the memberships did not appear to be in the literature yet.

Let us now look at some special cases. When there are only two fuzzy clusters, only one nondegenerate component will be left. It is then sufficient to plot the membership u_{i1} of each object in the first cluster, because its membership in the second cluster can then be read from right to left as $u_{i2} = 1 - u_{i1}$.

When there are three fuzzy clusters, each object has memberships (u_{i1}, u_{i2}, u_{i3}). The possible combinations fill an equilateral triangle in three-dimensional space. Principal components recover this triangle (as in Figure 9), but one can also plot the memberships directly by means of barycentric coordinates (also called trilinear coordinates). For instance, one could plot

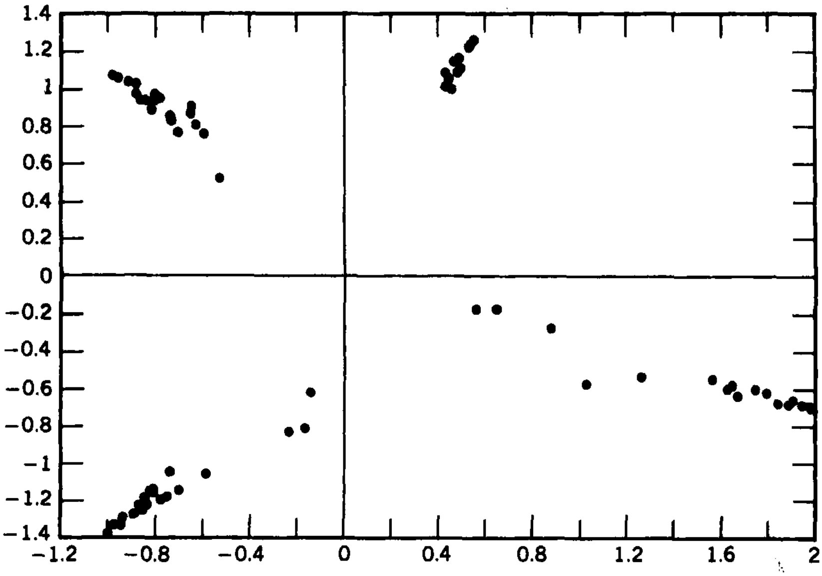

Figure 10 First two principal components of memberships of the Ruspini data in four fuzzy clusters.

$u_{i3}\sqrt{3}/2$ versus $u_{i2} + u_{i3}/2$ for each object, thereby using an equilateral triangle with vertices $(0,0)$, $(1,0)$, and $(1/2, \sqrt{3}/2)$.

When there are more than three fuzzy clusters, there will be more than two principal components. We then follow the customary practice of displaying the two components with largest eigenvalues, thereby "explaining" the largest portion of the variability. An example is given in Figure 10, displaying the two main components of the memberships of the Ruspini data in four fuzzy clusters (as given by FANNY). Sometimes one will want to make a three-dimensional plot or perhaps draw several plots in which each component is plotted versus every other component.

The plot can also be refined by adding "ideal" objects, corresponding to the clusters themselves. Indeed, each cluster can be represented by an object with membership 1 to that cluster and zero membership to all others. By transforming these "membership coordinates" in the same way as the actual objects, the plot will be enriched by as many additional points as there are clusters. In this way the final plot contains both objects and clusters (in the same way that correspondence analysis yields plots containing both objects and variables).

EXERCISES AND PROBLEMS

1. Carefully examine the membership coefficients shown in Table 1, in particular the differences between memberships of objects belonging to the same cluster. Explain these differences by the positions of the objects in Figure 1.

2. Run the program FANNY on the data set of Figure 1. Use the same options as in Section 2.1 but with the number of clusters varying between 2 and 5. Then select a value of k with the silhouettes.

3. Apply FANNY to the sciences data in Table 6 of Chapter 2. Check that Dunn's normalized partition coefficient and the silhouettes yield different choices of k.

4. Apply FANNY to the data in Table 1 of Chapter 2, for $k = 2$. Which object is most fuzzy?

5. Apply FANNY with $k = 2$ to the dissimilarity matrix in Figure 12 of Chapter 5, concerning nine diseases. What do you think of the cluster-

ing quality of these data based on (a) the memberships, (b) Dunn's normalized partition coefficient, and (c) the average silhouette width?

6. A possible nonfuzziness index is the average difference between the largest and the second largest membership for each object. (Observe that this index can take all values between 0 and 1.) Calculate this index for the clusterings in Exercises 4 and 5.

7. Consider the bacteria data in Exercise 1 of Chapter 2. Cluster these data into 2 and 3 clusters using the program FANNY and compare the results with those obtained from the program PAM.

8. Show that the exact minimum of the objective function (1) is a decreasing function of the number of clusters.

9. Compute the fuzzy k-means clustering (with $k = 3$) of the data of Figure 1 by entering the matrix of *squared* Euclidean distances as input to FANNY. Compare the result with Table 1.

10. Consider the dissimilarities between 12 countries listed in Table 5 of Chapter 2. Calculate the coefficients $D_k(U)$ and $D_k'(U)$ for the FANNY clustering with $k = 3$ and draw a plot in which this fuzzy clustering is represented by the point (D_k', F_k').

11. Compute both principal components of the memberships in Table 1 (this can be done by means of any standard software package). Compare the resulting bivariate plot with the original data in Figure 1.