FCM—the Fuzzy C-Means clustering-algorithm

Article *in* Computers & Geosciences · December 1984

DOI: 10.1016/0098-3004(84)90020-7

3 authors, including:

James C. Bezdek
University of Missouri
398 PUBLICATIONS **46,880** CITATIONS
SEE PROFILE

William Full
GXStat LLC
67 PUBLICATIONS **3,765** CITATIONS
SEE PROFILE

Computers & Geosciences Vol. 10, No. 2-3, pp. 191-203, 1984.
Printed in the U.S.A.

0098-3004/84 $3.00 + .00

FCM: THE FUZZY *c*-MEANS CLUSTERING ALGORITHM

JAMES C. BEZDEK
Mathematics Department, Utah State University, Logan, UT 84322, U.S.A.

ROBERT EHRLICH
Geology Department, University of South Carolina, Columbia, SC 29208, U.S.A.

WILLIAM FULL
Geology Department, Wichita State University, Wichita, KS 67208, U.S.A.

(*Received* 6 *May* 1982; *revised* 16 *May* 1983)

Abstract—This paper transmits a FORTRAN-IV coding of the fuzzy *c*-means (FCM) clustering program. The FCM program is applicable to a wide variety of geostatistical data analysis problems. This program generates fuzzy partitions and prototypes for any set of numerical data. These partitions are useful for corroborating known substructures or suggesting substructure in unexplored data. The clustering criterion used to aggregate subsets is a generalized least-squares objective function. Features of this program include a choice of three norms (Euclidean, Diagonal, or Mahalonobis), an adjustable weighting factor that essentially controls sensitivity to noise, acceptance of variable numbers of clusters, and outputs that include several measures of cluster validity.

Key Words: Cluster analysis, Cluster validity, Fuzzy clustering, Fuzzy QMODEL, Least-squared errors.

INTRODUCTION

In general, cluster analysis refers to a broad spectrum of methods which try to subdivide a data set X into c subsets (clusters) which are pairwise disjoint, all nonempty, and reproduce X. via union. The clusters then are termed a hard (i.e., nonfuzzy) c-partition of X. Many algorithms, each with its own mathematical clustering criterion for identifying "optimal" clusters, are discussed in the excellent monograph of Duda and Hart (1973). A significant fact about this type of algorithm is the defect in the underlying axiomatic model that each point in X is unequivocally grouped with other members of "its" cluster, and thus bears no apparent similarity to other members of X. One such manner to characterize an individual point's similarity to all the clusters was introduced in 1965 by Zadeh (1965). The key to Zadeh's idea is to represent the similarity a point shares with each cluster with a function (termed the membership function) whose values (called memberships) are between zero and one. Each sample will have a membership in every cluster, memberships close to unity signify a high degree of similarity between the sample and a cluster while memberships close to zero imply little similarity between the sample and that cluster. The history, philosophy, and derivation of such mathematical systems are documented in Bezdek (1981). The net effect of such a function for clustering is to produce fuzzy c-partitions of a given data set. A fuzzy c-partition of X is one which characterizes the membership of each sample point in all the clusters by a membership function which ranges between zero and one. Additionally, the sum of the memberships for each sample point must be unity.

Let $Y = \{y_1, y_2, \ldots, y_N\}$ be a sample of N observations in $\mathbf{R}^n$ (n-dimensional Euclidean space); y_k is the k-th feature vector; y_{kj} the j-th feature of y_k. If c is an integer, $2 \leqslant c < n$, a conventional (or "hard") c-partition of Y is a c-tuple $(Y_1, Y_2, \ldots, Y_c)$ of subsets of Y that satisfies three conditions:

$$Y_i \neq \phi' \quad 1 \leqslant i \leqslant c; \tag{1a}$$

$$Y_i \cap Y_j = \phi; \quad i \neq j \tag{1b}$$

$$\bigcup_{i=1}^{c} Y_i = Y \tag{1c}$$

In these equations, ϕ stands for the empty set, and $(\cap, \cup)$ are respectively, intersection, and union.

In the context discussed later, the sets $\{Y_i\}$ are termed "clusters in Y. Clusters analysis (or simply clustering) in Y refers to the identification of a distinguished c-partition $\{\hat{Y}_i\}$ of Y whose subsets contain points which have high intracluster resemblance; and, simultaneously, low intercluster similarity. The mathematical criterion of resemblance used to define an "optimal" c-partion is termed a cluster criterion. One hopes that the substructure of Y represented by $\{\hat{Y}_i\}$ suggests a useful division or relationship between the population variables of the real physical process from whence Y was drawn. One of the first questions one might ask is whether Y was drawn. One of the first questions one might ask is whether Y contains any clusters at all. In many

geological analyses, a value for c is known *a priori* on physical grounds. If c is unknown, then determination of an optimal c becomes an important issue. This question is sometimes termed the "cluster validity" problem. Our discussion, in addition to the clustering *a posteriori* measures of cluster validity (or "goodness of fit").

Algorithms for clustering and cluster validity have proliferated due to their promise for sorting out complex interactions between variables in high dimensional data. Excellent surveys of many popular methods for conventional clustering using deterministic and statistical clustering criteria are available; for example, consult the books by Duda and Hart (1973), Tou and Gonzalez (1974), or Hartigan (1975). The conventional methodologies discussed in these references include factor analytic techniques, which occupy an important place in the analysis of geoscientific data. The principal algorithms in this last category are embodied in the works of Klovan and Imbrie (1971), Klovan and Miesch (1976), and Miesch (1976a, 1976b). These algorithms for the factor analytical analysis of geoscientific data are known as the QMODEL algorithms (Miesch, 1976a).

In several recent studies, the inadequacy of the QMODEL algorithms for linear unmixing when confronted with certain geometrical configurations in grain shape data has been established numerically (Full, Ehrlich, and Klovan, 1981; Full, Ehrlich, and Bezdek, 1982; Bezdek, and others, 1982. The problem is caused by the presence of outliers. Aberrant points may be real outliers, noise, or simply due to measurement errors; however, peculiarities of this type can cause difficulties for QMODEL that cannot be resolved by standard approaches. The existence of this dilemma led the authors to consider fuzzy clustering methods as an adjunct procedure which might circumvent the problems caused by data of this type. Because fuzzy clustering is most readily understood in terms of the axioms underlying its rationale, we next give a brief description of the basic ideas involved in this model.

FUZZY CLUSTERING

The FCM algorithms are best described by recasting conditions (equation 1) in matrix-theoretic terms. Towards this end, let U be a real $c \times N$ matrix, $U = [u_{ik}]$. U is the matrix representation of the partition $\{Y_i\}$ in equation (1) in the situation

$$u_i(y_k) = u_{ik} = \begin{cases} 1; & y_k \in Y_i \\ 0; & \text{otherwise} \end{cases} \tag{2a}$$

$$\sum_{i=1}^{N} u_{ik} > 0 \quad \text{for all } i; \tag{2b}$$

$$\sum_{i=1}^{N} u_{ik} = 1 \quad \text{for all } k. \tag{2c}$$

In equation (2), u_i is a function; u_i: $Y \rightarrow \{0, 1\}$. In conventional models, u_i is the characteristic function of Y_i: in fact, u_i and Y_i determine one another, so there is no harm in labelling u_i the ith hard subset of the partition (it is unusual, of course, but is important in terms of understanding the term "fuzzy set"). Conditions of equations (1) and (2) are equivalent, so U is termed a hard c-partition of Y. Generalizing this idea, we refer to U as a fuzzy c-partition of Y when the elements of U are numbers in the unit interval [0, 1] that continue to satisfy both equations (2b) and (2c). The basis for this definition are c functions u_i: $Y \rightarrow [0, 1]$ whose values $u_i(y_k) \in [0, 1]$ are interpreted as the grades of membership of the y_ks in the "fuzzy subsets" u_i of Y. This notion is due to Zadeh (1965), who conceived the idea of the fuzzy set as a means for modelling physical systems that exhibit nonstatistical uncertainties. Detailed discussions for the rationale and philosophy of fuzzy sets are available in many recent papers and books (e.g., consult Bezdek (1981)).

For the present discussion, it suffices to note that hard partitions of Y are a special type of fuzzy ones, wherein each data point is grouped unequivocally with its intracluster neighbors. This requirement is a particularly harsh one for physical systems that contain mixtures, or hybrids, along with pure or antecedent strains. Outliers (noise or otherwise) generally fall into the category one should like to reserve for "unclassifiable" points. Most conventional models have no natural mechanism for absorbing the effects of undistinctive or aberrant data, this is a direct consequence of equation (1a). Accordingly, the fuzzy set, and, in turn, fuzzy partition, were introduced as a means for altering the basic axioms underlying clustering and classification models with the aim of accomodating this need. By this device, a point y_k may belong entirely to a single cluster, but in general, is able to enjoy partial membership in several fuzzy clusters (e.g., precisely the situation anticipated for hybrids). We denote the sets of all hard and fuzzy c-partitions of Y by:

$$M_c = \{U_{c\times N} | u_{ik} \in [0, 1]; \quad \text{equations (2b), (2c)}\}; \tag{3a}$$

$$M_{fc} = \{U_{c\times N} | u_{ik} \in [0, 1]; \quad \text{equations (2b), (2c)}]. \tag{3b}$$

Note that M_c is imbedded in M_{fc}. This means that fuzzy clustering algorithms can obtain hard c-partitions. On the other hand, hard clustering algorithms cannot determine fuzzy c-partitions of Y. In other words, the fuzzy imbedment enriches (not replaces!) the conventional partitioning model. Given that fuzzy c-partitions have at least intuitive appeal, how does one use the data to determine them? This is the next question we address.

Several clustering criteria have been proposed for identifying optimal fuzzy c-partitions in Y. Of these, the most popular and well studied method to date is

associated with the generalized least-squared errors functional

$$J_m(U, v) = \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^m \|y_k - v_i\|_A^2 \quad (4)$$

Equation (4) contains a number of variables: these are

$$Y = \{y_1, y_2, \ldots, y_N\} \subset \mathbf{R}^n = \text{the data}, \quad (5a)$$

$$c = \text{number of clusters in } Y; \quad 2 \leqslant c < n, \quad (5b)$$

$$m = \text{weighting exponent}; \quad 1 \leqslant m < \infty, \quad (5c)$$

$$U = \text{fuzzy } c\text{-partition of } Y; \quad U \in M_{fc} \quad (5d)$$

$$v = (v_1, v_2, \ldots, v_c) = \text{vectors of centers}, \quad (5e)$$

$$v_i = (v_{i1}, v_{i2}, \ldots, v_{in}) = \text{center of cluster } i, \quad (5f)$$

$$\| \ \|_A = \text{induced } A\text{-norm on } \mathbf{R}^n \quad (5g)$$

$$A = \text{positive-definite } (n \times n) \text{ weight matrix.} \quad (5h)$$

The squared distance between y_k and v_i shown in equation (4) is computed in the A-norm as

$$d_{ik}^2 = \|y_k - v_i\|_A^2 = (y_k - v_i)^T A (y_k - v_i). \quad (6)$$

The weight attached to each squared error is $(u_{ik})^m$, *the* mth power of y_ks membership in cluster i. The vectors $\{v_i\}$ in equation (5f) are viewed as "cluster centers" or centers of mass of the partitioning subsets. If $m = 1$, it can be shown that J_m minimizes only at hard U's $\in M_c$, and corresponding v_is are just the geometric centroids of the Y_is. With these observations, we can decompose J_m into its basic elements to see what property of the points $\{y_k\}$ it measures:

$$d_{ik}^2 = \text{squared } A\text{-distance from point } y_k \text{ to center of mass } v_i. \quad (7a)$$

$$(u_{ik})^m d_{ik}^2 = \text{squared } A\text{-error incurred by representing } y_k \text{ by } v_i \text{ weighted by (a power of) the membership of } y_k \text{ in cluster } i. \quad (7b)$$

$$\sum_{i=1}^{c} (u_{ik})^m d_{ik}^2 = \text{sum of squared } A\text{-errors due to } y_k\text{s partial replacement by all } c \text{ of the centers } \{v_i\}. \quad (7c)$$

$$\sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^m d_{ik}^2 = \text{overall weighted sum of generalized } A\text{-errors due to replacing } Y \text{ by } v. \quad (7d)$$

The role played by most of the variables exhibited in equation (5) is clear. Two of the parameters of J_m, however warrant further discussion, namely, m and A. Weighting exponent m controls the relative weights placed on each of the squared errors d_{ik}^2. As $m \to 1$ from earlier discussion partitions that minimize J_m become increasingly hard (and, as mentioned before, at $m = 1$, are necessarily hard). Conversely, each entry of optimal $\hat{U}$s for J_m approaches $(1/c)$ as $m \to \infty$. Consequently, increasing m tends to degrade (blur, defocus) membership towards the fuzziest state. Each choice for m defines, all other parameters being fixed, one FCM algorithm. No theoretical or computational evidence distinguishes an optimal m. The range of useful values seems to be [1, 30] or so. If a test set is available for the process under investigation, the best strategy for selecting m at present seems to be experimental. For most data, $1.5 \leqslant m \leqslant 3.0$ gives good results.

The other parameter of J_m that deserves special mention is weight matrix A. This matrix controls the shape that optimal clusters assume in $\mathbf{R}^n$. Because every norm on $\mathbf{R}^n$ is inner product induced via the formula

$$\langle x, y \rangle_A = x^T A y, \quad (8)$$

there are infinitely many A-norms available for use in equation (4). In practice, however, only a few of these norms enjoy widespread use. The FCM listing below allows a choice of three norms, each induced by a specific weight matrix. Let

$$c_y = \sum_{k=1}^{N} y_k / N; \quad (9a)$$

$$C_y = \sum_{k=1}^{N} (y_k - c_y)(y_k - c_y)^t, \quad (9b)$$

be the sample mean and sample covariance matrix of data set Y; and let $\{a_i\}$ denote the eigenvalues of C_y; let D_y be the diagonal matrix with diagonal elements $(d_y)_{ii} = a_i$; and finally, let I be the identity matrix. The norms of greatest interest for use with equation (4) correspond to

$$A = I \sim \text{-Euclidean Norm}, \quad (10a)$$

$$A = D_y^{-1} \sim \text{Diagonal Norm}, \quad (10b)$$

$$A = C_y^{-1} \sim \text{Mahalonobis Norm}. \quad (10c)$$

A detailed discussion of the geometric and statistical implications of these choices can be seen in Bezdek (1981). When $A = I$, J_m identifies hyperspherical clusters; for any other A, the clusters are essentially hyperellipsodial, with axes proportional to the eigenvalues of A. When the diagonal norm is used, each dimension is effectively scaled via the eigenvalues. The Euclidean norm is the only choice for which extensive experience with geological data is available.

Optimal fuzzy clusterings of Y are defined as pairs $(\hat{U}, \hat{v})$ that locally minimize J_m. The necessary conditions for $m = 1$ are well known (but hard to use, because M_c is discrete, but large). For $m > 1$, if $y_k \neq \hat{v}_j$ for all j and k, $(\hat{U}, \hat{v})$ may be locally optimal for J_m only if

$$\hat{v}_i = \sum_{k=1}^{N} (\hat{u}_{ik})^m y_k \Big/ \sum_{k=1}^{N} (\hat{u}_{ik})^m; \quad 1 \leqslant i \leqslant c; \quad (11a)$$

$$\hat{u}_{ik} = \left(\sum_{j=1}^{c} \left(\frac{\hat{d}_{ik}}{\hat{d}_{jk}} \right)^{2/(m-1)} \right)^{-1}; \quad 1 \leqslant k \leqslant N; \ 1 \leqslant i \leqslant c \quad (11b)$$

where $\hat{d}_{ik} = \|y_k - \hat{v}_i\|_A$. Conditions expressed in equations (11) are necessary, but not sufficient; they provide means for optimizing J_m via simple Picard iteration, by looping back and forth from equation (11a) to (11b) until the iterate sequence shows but small changes in successive entries of $\hat{U}$ or $\hat{v}$. We formalize the general procedure as follows:

Fuzzy c-Means (FCM) Algorithms

(A1) Fix c, m, A, $\|k\|_A$. Choose an initial matrix $U^{(o)} \in M_{fc}$. Then at step k, $k = 0, 1, \ldots, LMAX$.

(A) Compute means $\hat{v}^{(k)}$, $i = 1, 2, \ldots, c$ with equation (11)a.

(A3) Compute an updated membership matrix $\hat{U}^{(k+1)} = [\hat{u}_{ik}^{(k+1)}]$ with equation (11b).

(A4) Compare $\hat{U}^{k+1}$ to $\hat{U}^{(k)}$ in any convenient matrix norm. If $\|\hat{U}^{(k+1)} - \hat{U}^{(k)}\| < \in$, stop. Otherwise, set $\hat{U}^{(k)} = \hat{U}^{k+1}$ and return to (A2).

(A1)–(A4) is the basic algorithmic strategy for the FCM algorithms.

Individual control parameters, tie-breaking rules, and computing protocols are discussed in conjunction with the appended FORTRAN listing in Appendix 1.

Theoretical convergence of the sequence $\{\hat{U}^{(k)}, \hat{v}^{(k)}, k = 0, 1, \cdots\}$ generated by (A1)–(A4) has been studied (by Bezdek, 1981). Practically speaking, no difficulties have ever been encountered, and numerical convergence is usually achieved in 10–25 iterations. Whether local minima of J_m are good clusterings of Y is another matter, for it is easy to obtain data sets upon which J_m minimizes globally with visually unappealing substructure. To mitigate this difficulty, several types of cluster validity functionals are usually calculated on each $\hat{U}$ produced by FCM. Among the most popular are the partition coefficient and entropy of $\hat{U} \in M_{fc}$:

$$F_c(\hat{U}) = \sum_{k=1}^{N} \sum_{i=1}^{c} (\hat{u}_{ik})^2/N; \tag{12a}$$

$$H_c(\hat{U}) = -\sum_{k=1}^{N} \sum_{i=1}^{c} (\hat{u}_{ik} \log_a(\hat{u}_{ik}))/N. \tag{12b}$$

In equation (12b), logarithmic base $a \in (1, \infty)$. Properties of F_c and H_c utilized for validity checks are:

$$F_c = 1 \Leftrightarrow H_c = 0 \Leftrightarrow \hat{U} \in M_c \text{ is hard;} \tag{13a}$$

$$F_c = 1/c \Leftrightarrow H_c = \log_a(c) \Leftrightarrow \hat{U} = [1/c]; \tag{13b}$$

$$\frac{1}{c} \leqslant F_c \leqslant 1; \quad 0 \leqslant H_c \leqslant \log_a(c). \tag{13c}$$

Entropy H is a bit more sensitive than F to local changes in partition quality. The FCM program listed below calculates F, H, and $(1 - F)$, the latter quantity owing to the inequality $(1 - F) < H$ for $\hat{U} \not\in M_c$ (when $a = e = 2.71\cdots$).

Finally, we observe that generalizations of J_m which can accommodate a much wider variety of data shapes than FCM are now well known (see Bezdek (1981) for a detailed account). Nonetheless, the basic FCM algorithm remains one of the most useful general purpose fuzzy clustering routines, and is the one utilized in the FUZZY QMODEL algorithms discussed by Full, Ehrlich, and Bezdek (1982). Having given a brief account of the generalities, we now turn to computing protocols for the FCM listing accompanying this paper.

ALGORITHMIC PROTOCOLS

The listing of FCM appended below has some features not detailed in (A1)–(A4). Our description of the listing corresponds to the blocks as documented.

Input Variables. FCM arrays are listing documented. Symbolic dimensions are

NS = number of vectors in $Y = N$.

ND = number of features in $y_k = n$.

Present dimensions will accomodate up to $c = 20$ clusters, $N = 500$ data points, and $n = 20$ features. Input variables ICON specifies the weight matrix A as in equation (10):

ICON = $1 \Rightarrow A = I$

ICON = $2 \Rightarrow A = D_y^{-1}$.

ICON = $3 \Rightarrow A = C_y^{-1}$.

Other parameters read are:

QQ = Weighting exponent m: $1 <$ QQ.

KBEGIN = Initial number of clusters:

$2 \leqslant$ KBEGIN $\leqslant$ DCEASE.

KCEASE = Final number of clusters:

KCEASE $<$ NS.

At any step NCLUS $= C$ is the operating number of clusters. FCM iterates over NCLUS from KBEGIN to KCEASE, generating an optimal pair $(\hat{U}, \hat{v})_{\text{NCLUS}}$ for each number of clusters desired. Changes in m and A must be made between runs (although they could easily be made iterate parameters).

Control Parameters

EPS = Termination criterion $\in$ in (A4).

LMAX = Maximum number iterations at each c in (A1).

Current values of EPS and LMAX are 0.01 and 50. Lowering EPS almost always results in more iterations to termination.

Input Y

Compute Feature Means. Vector FM(ND) is the mean vector c_y of equation (9a).

Compute Scaling Matrix. Matrix CC(ND, ND) is matrix A of equation (10), depending upon the choice made for ICON. The inverse is constructed in the main to avoid dependence upon peripheral subs. Matrix CM = A^*A^{-1} calculated as a check on the computed inverse, but no residual is calculated; nor does the FCM routine contain a flag if CM is not "close" to I. The construction of weight matrices other than the three choices allowed depends on user definition.

Loop Control. NCLUS = c is the current number of clusters: QQ is the weighting exponent m.

Initial Guess. A pseudo-random initial guess for U_0 is generated in this block at each access.

Cluster Centers. Calculation of current centers V(NC, ND) via equation (11a).

Update Memberships. Calculations with equation (11b); W(NC, ND) is the updated membership matrix. The special situation $m = 1$ is not accounted for here. Many programs are available for this situation for example see Ball (1965). The authors will furnish a listing for hard c-means upon request. Note that this block does not have a transfer in situation $y_k = \hat{v}_i$ for some k and i. This eventuality to our knowledge, has never occurred in nearly 10 years of computing experience. If a check and assignment are desired, the method for assigning $\hat{u}_i$'s in any column k where such a singularity occurs is arbitrary, as long as constraints in equation (2) are satisfied. For example, one may, in this instance, place equal weights (that sum to one) on every row where $y_k = \hat{v}_i$, and zero weights otherwise. This will continue the algorithm, and roundoff error alone should carry the sequence away from such points.

Error Criteria and Cutoffs. The criterion used to terminate iteration at fixed NC is

$$\text{ERRMAX} = \max_{i,k}\{|\hat{u}_{ik}^{(k+1)} - \hat{u}_{ik}^{(k)}|\} < \text{EPS}. \quad (14)$$

Threshold EPS thus controls the accuracy of terminal output. An alternative method to terminate iteration would be to compare components of each $\hat{v}_i^{(k+1)}$ to $\hat{v}_i^{(k)}$. There may be differences in terminal pairs $(\hat{U}, \hat{v})$ obtained using a fixed EPS. Furthermore, there is a tradeoff in CPU time, equation (14) requires (cN) comparisons and $\max_{ij}\{|\hat{v}_{ij}^{(k+1)} - \hat{v}_{(ij)}^{(k)}|\}$ requires (cn) comparisons. Thus, if N is much larger than n, $(N \gg n)$, termination based on the quality of successive cluster centers computed via equation (11a) becomes more attractive. By the same token, this can reduce storage space (for updated centers instead of an updated membership matrix) significantly if $n \ll N$. If equation (14) is never satisfied, iteration at current NC will stop when k = LMAX: a convergence flag is issued, and NC advance to NC + 1. More than 25 iterations are rarely needed for EPS in the 0.001 range.

Cluster Validity Indicants. Values of J_m, F_c, H_c, and $1 - F_c$ are computed, and stored, respectively, in the vectors *VJM*, *F*, *H*, and *DIF*.

Output Block. For the current value of NCLUS, current terminal values of F_c, $1 - F_c$, H_c, J_m, $\{\hat{v}_i\}$, and $\hat{U}$ are printed.

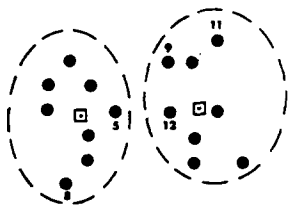
Output Summary. The final block of FCM outputs statistics for the entire run.

The listing provided is a very basic version of FCM: many embellishments are discussed in Bezdek (1981). As an aid for debugging a coded copy of the listing, we present a short example that furnishes a means for checking numerical outputs. This example highlights several of the important features of fuzzy z-partitions in general, and those generated by FCM in particular. Examples of the use of FCM in the context of geological data analysis are presented in Bezdek, and other (1982), and Full, Ehrlich, and Bezdek (1982).

Storage Requirements. The program listed in the appendix can handle 500 data samples with up to 50 variables. It will handle up to 20 clusters. The program, as written, used under 256 K of computer storage. If larger data sets are used, the program is clearly documented as to which parameters to change.

A NUMERICAL EXAMPLE

Figure 1 displays a set Y of 16 points in $\mathbf{R}^2$. This artificial data set was originally published in Sneath



No.	Coordinates y_{k1}	y_{k2}
1	0	4
2	0	3
3	1	5
4	2	4
5	3	3
6	2	2
7	2	1
8	1	0
9	5	5
10	6	5
11	7	6
12	5	3
13	7	3
14	6	2
15	6	1
16	8	1

⊡ = Terminal cluster centers from Table 2, col. 3

) = Terminal maximum membership "boundaries".

Fig. 1. An example: Artificial touching clusters.

and Sokal (1973) in connection with the illustration of a hard clustering algorithm called the unpaired group mean average (UPGMA) method. This data was subsequently studied in Bezdek (1974), where a comparison between the UPGMA and FCM methods was effected. The coordinates of $y_k \in Y$ are listed as columns two and three of the tabular display of Fig. 1. This is a good data set for our purposes because it is easily handled for validation, and further, has some of the geometric properties that necessitate the introduction of fuzzy models. Data of this type might be drawn from a mixture of two bivariate normal distributions. The region of overlap contains several points which might be considered "noise", *viz.* y_5 and y_{12}. Parameters for the outputs to be discussed were as follows:

Table	ICON = A	NCLUS = c	QQ = m	EPS = $\in$
1	1, 2, 3	2	2	0.01
2	2	2	1.25, 2.00	0.01
3	1	2–6	1.25–2.00	0.01

In other words, we illustrate in Tables 1 and 2, respectively, the effects of variation in the norm inducing matrix A, and weighting exponent on $(\hat{U}, \hat{v})$ with all other parameters being fixed; while Table 3 exhibits variations in F_c and H_c due to changes in m and c.

Initial guesses for U_0 were not chosen randomly here, so that users may validate their programs against these tables. Rather, the initial matrix used for all of the outputs discussed later had the following elements:

$$(U_0)_{ii} = \left(\frac{\alpha}{c} + B\right); \quad i = 1, 2, \ldots, c$$

$$(U_0)_{ij} = \left(\frac{\alpha}{c} + \beta\right); \quad j = c + 1, \ldots, n$$

$$(U_0)_{ij} = \left(\frac{\alpha}{c} + \beta\right); \quad \text{otherwise}$$

$$\alpha = 1 - (\sqrt{2}/2);$$

$$\beta = \sqrt{2}/2.$$

The starting value for F_c using this U_0 is always the midpoint of $[1/c, 1)$, the range of F_c, that is $f_c(U_0) = ((1/c) + 1)/2$. In real applications it is, of course, important to run FCM for several different U_0s, as the iteration method used, like all descent methods, is susceptible to local stagnations. If different U_0s result in different $(\hat{U}, \hat{v})$s, one thing is certain: further analysis should be made before one places much confidence in any algorithmically suggested substructure in Y.

Table 1 shows that maximizing F_c is equivalent to minimizing H_c but this behavior is not equivalent to minimizing J_m. Several examples of the general dilemma are documented in Bezdek (1981). Observe that all three partitions of Y are (qualitatively) more or less equivalent. Lower membership generally cor-

Table 1. Variation in $(\hat{U}, \hat{v})$ due to changes in Norm. There are only two clusters, hence $\hat{U}_{2k} = (1 - \hat{U}_{1k})$ as the sum of the $\hat{u}_{ik}$ equals one. Terminal membership U_{ik}

Data Point	ICON = 1 A = I	ICON = 2 A = D_y^{-1}	ICON = 3 A = C_y^{-1}
1	0.92	0.88	0.89
2	0.95	0.93	0.92
3	0.86	0.78	0.82
4	0.91	0.88	0.93
5	0.80	0.84	0.84
6	0.95	0.88	0.82
7	0.86	0.72	0.65
8	0.82	0.67	0.62
9	0.22	0.35	0.43
10	0.12	0.26	0.33
11	0.18	0.32	0.37
12	0.10	0.08	0.09
13	0.02	0.03	0.04
14	0.06	0.09	0.06
15	0.16	0.24	0.19
16	0.15	0.21	0.19
$\hat{v}_{11}$	6.18	5.99	5.96
$\hat{v}_{12}$	3.15	2.95	2.75
$\hat{v}_{21}$	1.44	1.67	1.73
$\hat{v}_{22}$	2.83	3.01	3.19
F_c	0.80	0.71	0.71
H_c	0.35	0.45	0.45
J_m	51.65	13.69	13.69
Iter.	6	6	12

Table 2. Variation in $(\hat{u}, \hat{v})$ due to changes in m (two cluster example). Terminal membership $\hat{u}_{ik}$: $\hat{u}_{2k} = (1 - \hat{u}_{1k})$

Data Point	QQ = m = 1.25	QQ = m = 2.00
1	1.00	0.92
2	1.00	0.95
3	1.00	0.86
4	1.00	0.91
5	1.00	0.80
6	1.00	0.95
7	1.00	0.86
8	1.00	0.82
9	0.00	0.22
10	0.00	0.12
11	0.00	0.18
12	0.00	0.10
13	0.00	0.02
14	0.00	0.06
15	0.00	0.16
16	0.00	0.15
$\hat{v}_{11}$	6.25	6.18
$\hat{v}_{12}$	3.25	3.15
$\hat{v}_{21}$	1.37	1.44
$\hat{v}_{22}$	2.75	2.83
F_c	1.00	0.80
H_c	0.00	0.35
J_m	60.35	51.65
Iter.	4	6

Table 3. Variation in F and H due to changes in m and c.

Weighting Exponent (m)	Number of Clusters (c)	Partition Coefficient (F_c)	Lower Bound ($1-F_c$)	Normalized Entropy (H_c)
1.25	2	0.998	0.002	0.007
	3	0.983	0.017	0.037
	4	0.979	0.021	0.044
	5	0.996	0.004	0.013
1.50	2	0.955	0.045	0.103
	3	0.903	0.097	0.202
	4	0.901	0.099	0.201
	5	0.917	0.083	0.197
1.75	2	0.873	0.127	0.239
	3	0.791	0.209	0.404
	4	0.804	0.196	0.401
	5	0.776	0.224	0.468
2.00	2	0.794	0.206	0.352
	3	0.686	0.314	0.575
	4	0.700	0.300	0.600
	5	0.662	0.338	0.701

responds to points distant from the "core" (i.e. $\hat{v}_i$) of cluster i. Thus, point 8 is clearly signaled an outlier, for example in all three partitions. Notice, however, that the Mahalanobis norm emphasizes this much more heavily than, for example the Euclidean norm. This is because level sets in the former norm are elliptical, and in the latter circular. Thus, the variance of y_8 in the vertical direction weights its influence differently. In all situations, points near cluster centers in the A-norm have higher memberships. Note that $\hat{v}_1$ is more stable to changes in A than v_2: this indicates that points with a high affinity for membership in $\hat{u}_2$ have somewhat more variability than those seeking to associate with $\hat{u}_1$. Table 1 also demonstrates another general fact; the number of iterates needed using the Mahalonobis norm is usually higher than

the number required by other norms. See Bezdek (1981) for more discussion concerning characteristic cluster shapes associated with changes in A.

Table 2 illustrates the usual effect of increasing m lower m's yield harder partitions and higher ones, fuzzier memberships. For $m = 1.25$, $\hat{U}$ is hard (to 2 decimal places). Observe that F_c and H_c mirror this fact, but again, J_m does not, having a higher value at the lower m. Further observe that the cluster centers are rather stable to changes in m. This is not always the situation, and it is an unproven conjecture that the stability of the $\hat{v}_i$'s in the face of severe changes in m is in some sense an indication of cluster validity. Figure 1 exhibits $\hat{v}_1$ and $\hat{v}_2$ for $m = 2 = c$, $A = I$; their geometric positions are at least (visually) appealing.

Table 3 depicts the utility of F_c and H_c for the cluster validity question. For every m, F_c maximizes (and H_c minimizes) at $c = 2$. From this we can infer that the "hardest" substructure detectable in Y occurs are $c = 2$. These values do not, however, have any direct tie to Y. Being computed on algorithmic outputs based on Y rather than any concrete assumptions regarding the distribution of Y somewhat weakens the theoretical plausibility of using F_c and H_c for cluster validity. Nevertheless, they have been demonstrably reliable in many experimental studies, and are, at present, the most reliable indicants of validity for the FCM algorithms.

REFERENCES

Ball, G. 1965, Data analysis in the social sciences: what about the details?: Proc. FJCC, Spartan Books, Washington, D.C., p. 533–560.

Bezdek, J. C. 1974. Mathematical models for systematics and Taxonomy, *in*, Estabrook, G., ed., Proceedings of the 8th International Conference on Numerical Taxonomy: W. H. Freeman, San Francisco, p. 143–166.

Bezdek, J. C., 1980, A convergence theorem for the fuzzy *c*-means clustering algorithms: IEEE Trans. PAMI, PAMI-2(1), p. 1–8.

Bezdek, J. C., 1981, Pattern recognition with fuzzy objective function algorithms: Plenum, New York, 256. p.

Bezdek, J. C., Trivedi, M., Ehrlich, R., and Full, W., 1982, Fuzzy clustering: a new approach for geostatistical analysis: Int. Jour. Sys., Meas., and Decisions.

Duda, R., and Hart, P., 1973, Pattern classification and scene analysis: Wiley-Interscience, New York, 482. p.

Full, W., Ehrlich, R., and Bezdek, J., 1982, FUZZY QMODEL: A new approach for linear unmixing: Jour. Math. Geology.

Full, W., E., Erhlich, R., and Klovan, J. E., 1981, EXTENDED QMODEL—Objective definition of external end members in the analysis of mixtures: Jour. Math. Geology, v. 13, no. 4, p. 331–344.

Hartigan, J., 1975, Clustering algorithms: John Wiley and Sons, New York, 351 p.

Klovan, J., and Imbrie, J., 1971, An algorithm and FORTRAN-IV program for large scale Q-mode factor analysis and calculation of factor Scores: Jour. Math. Geology, v. 3, no. 1, p. 61–76.

Klovan, J. E., and Miesch, A., 1976, EXTENDED CABFAC and QMODEL Computer programs for Q-mode factor analysis of compositional data: Computers & Geosciences, v. 1, no. 3, p. 161–178.

Miesch, A. T., 1976a, Q-mode factor analysis of geochemical and petrologic data matrices with constant row-sums: U.S. Geol. Survey Prof. Paper 574–G, 47 p.

Miesch, A. T., 1976b, Interactive computer programs for petrologic modeling with extended Q-mode factor analysis: Computers & Geosciences, v. 2, no. 2, p. 439–492.

Sneath, P., and Sokal, R., 1973, Numerical taxonomy: Freeman, San Francisco, 573 p.

Tou, J., and Gonzalez, R., 1974, Pattern recognition principles: Addison-Wesley, Reading, Mass., 377 p.

Zadeh, L. A., 1965, Fuzzy sets: Inf. and Cont., v. 8, p. 338–353.

APPENDIX

Listing of fuzzy C-means

```
FILE: KMEANS    FORTRAN  A 03/18/83 11:05  VM/SP CONVERSATIONAL MONITOR SYSTEM


C                                                                       00001000
C                                                                       00002000
C     THIS IS THE FCM (FUZZY C-MEANS) ROUTINE. THIS LISTING IS FOR A    00003000
C     IBM TYPE COMPUTER WITH A FORTRAN IV COMPILER. IT ADAPTS FOR ANY   00004000
C     FORTRAN COMPILER WITH MODIFICATIONS SET AT THE USER SITE.         00005000
C                                                                       00006000
C     REFERENCE: "PATTERN RECOGNITION WITH FUZZY OBJECTIVE FUNCTIONS,"  00007000
C                JAMES BEZDEK, PLENUM, NEW YORK, 1981.                  00008000
C                                                                       00009000
C                                                                       00010000
C     DESCRIPTION OF OPERATING VARIABLES:                               00011000
C     I. INPUT VARIABLES (FROM FILE 5)                                  00012000
C         CARD 1:                                                       00013000
C          TITLE(20).........80 CHARACTER HEADING                       00014000
C         CARD 2:                                                       00015000
C          FMT(20)...........FORTRAN FORMAT (CONTAINED IN PARENTHESIS)  00016000
C                            DESCRIBING THE INPUT FORMAT FOR THE RAW DATA00017000
C                            UP TO 80 CHARACTERS MAY BE USED            00018000
C         CARD 3:                                                       00019000
C          COL 1: ICON......DISTANCE MEASURE TO BE USED. IF:            00020000
C                            ICON=1 USE EUCLIDEAN NORM                  00021000
C                            ICON=2 USE DIAGONAL NORM                   00022000
C                            ICON=3 USE MAHALANOBIS NORM                00023000
C          COLS 2-7: QQ.....WEIGHTING EXPONENT FOR FCM                  00024000
C          COLS 8-9: ND.....NUMBER OF FEATURES PER INPUT VECTOR         00025000
```

```
C          COLS 10-11:KBEGIN.STARTING NUMBER OF CLUSTERS                  00026000
C          COLS 12-13:KCEASE.FINISHING NUMBER OF CLUSTERS (NOTE: KBEGIN   00027000
C                           MUST BE LESS THAN OR EQUAL TO KCEASE)         00028000
C         CARD 4 ON:                                                      00029000
C          Y(NS,ND).........FEATURE VECTORS, INPUT ROW-WISE               00030000
C     II. INTERNAL VARIABLES                                              00031000
C          NS...............NUMBER OF DATA VECTORS                       00032000
C          EPS..............MAXIMUM MEMBERRSHIP ERROR AT CONVERGENCE      00033000
C          NC...............CURRENT NUMBER OF CLUSTERS                    00034000
C          LMAX.............MAXIMUM NUMBER OF ITERATIONS WITHOUT          00035000
C                           CONVERGENCE                                   00036000
C          FM(ND)...........SAMPLE MEAN VECTOR                           00037000
C          FVAR(ND).........VECTOR OF MARGINAL VARIANCES                 00038000
C          CC(ND,ND)........SCALING MATRIX                               00039000
C          AA(ND,ND)........SAMPLE COVARIANCE MATRIX                     00040000
C          AI(ND,ND)........INVERSE OF SAMPLE COVARIANCE MATRIX          00041000
C          BB(ND)...........DUMMY HOLDING MATRIX                         00042000
C          CCC(ND)..........DUMMY HOLDING MATRIX                         00043000
C          ST(ND,ND)........DUMMY HOLDING MATRIX FOR AA                  00044000
C          CM(ND,ND)........CM=AA*(AA INVERSE)                           00045000
C          U(NC,NS).........MEMBERSHIP MATRIX                            00046000
C          W(NC,NS).........UPDATED MEMBERSHIP MATRIX                    00047000
C          V(NC,ND).........CLUSTER CENTERS                              00048000
C          ITT(NC)..........DUMMY HOLDING MATRIX                         00049000
C          H(NC)............ENTROPY MATRIX                               00050000
C          VJM(NC)..........PAYOFF MATRIX                                00051000
C          F(NC)............MATRIX OF PARTITION COEFFICIENTS              00052000
C          DIF(NC)..........MATRIX OF ENTROPY BOUNDS                     00053000
C                                                                         00054000
C                                                                         00055000

FILE: KMEANS   FORTRAN  A 03/18/83 11:05  VM/SP CONVERSATIONAL MONITOR SYSTEM

C*************************************************************************00056000
      DIMENSION FM(50),FVAR(50),F(20)                                     00057000
      DIMENSION BB(50),CCC(50),H(20),DIF(20),ITT(20)                      00058000
      DIMENSION Y(500,2),U(20,500),W(20,500)                              00059000
      DIMENSION AA(50,50),AI(50,50)                                       00060000
      DIMENSION CC(50,50),CM(50,50),ST(50,50)                             00061000
      DIMENSION V(20,50),VJM(20)                                          00062000
      DIMENSION FMT(20),TITLE(20)                                         00063000
      READ(5,1458) (TITLE(I),I=1,20)                                      00064000
1458  FORMAT(20A4)                                                        00065000
      READ(5,12321) (FMT(I),I=1,20)                                       00066000
12321 FORMAT(20A4)                                                        00067000
C-------------------------------------------------------------------------00068000
C     CONTROL PARAMETERS.                                                 00069000
C-------------------------------------------------------------------------00070000
      EPS=.01                                                             00071000
      NS=1                                                                00072000
      LMAX=50                                                             00073000
C-------------------------------------------------------------------------00074000
C     READ FEATURE VECTORS (Y(I,J)).                                      00075000
C-------------------------------------------------------------------------00076000
      READ(5,2021) ICON,QQ,ND,KBEGIN,KCEASE                               00077000
2021  FORMAT(I1,F6.3,3I2)                                                 00078000
      WRITE(6,410)                                                        00079000
410   FORMAT(///1H ,'*** *** BEGIN FUZZY C-MEANS OUTPUT *** ***')         00080000
      WRITE(6,1459) (TITLE(III),III=1,20)                                 00081000
1459  FORMAT(10X,20A4///)                                                 00082000
1     READ(5,399,END=3)(Y(NS,J),J=1,ND)                                   00083000
399   FORMAT (2F1.0)                                                      00084000
      WRITE(6,12738)(Y(NS,J),J=1,ND)                                      00085000
12738 FORMAT(2(10X,10(F7.2,1X)/))                                         00086000
      NS=NS+1                                                             00087000
      GO TO 1                                                             00088000
3     NS=NS-1                                                             00089000
      NDIM=ND                                                             00090000
      NSAMP=NS                                                            00091000
      WRITE(6,11111) NSAMP                                                00092000
11111 FORMAT(10X,'NUMBER OF SAMPLES = ',I5)                               00093000
      ANSAMP=NSAMP                                                        00094000
C-------------------------------------------------------------------------00095000
C     SCALED NORM REQUIRED IN STATEMENTS 31 AND 33.                       00096000
C     CALCULATION OF SCALING MATRIX FOLLOWS.                              00097000
C     FEATURE MEANS.                                                      00098000
C-------------------------------------------------------------------------00099000
```

```
      DO 350 I=1,NDIM                                                  00100000
      FM(I)=0.                                                         00101000
      DO 351 J=1,NSAMP                                                 00102000
351   FM(I)=FM(I)+Y(J,I)                                               00103000
350   FM(I)=FM(I)/ANSAMP                                               00104000
C-----------------------------------------------------------------------00105000
C     FEATURE VARIANCES.                                               00106000
C-----------------------------------------------------------------------00107000
      DO 352 I=1,NDIM                                                  00108000
      FVAR(I)=0.                                                       00109000
      DO 353 J=1,NSAMP                                                 00110000
```

FILE: KMEANS FORTRAN A 03/18/83 11:05 VM/SP CONVERSATIONAL MONITOR SYSTEM

```
353   FVAR(I)=FVAR(I)+((Y(J,I)-FM(I))**2)                              00111000
352   FVAR(I)=FVAR(I)/ANSAMP                                           00112000
      IF(ICON-1)380,380,382                                            00113000
380   DO 381 I=1,NDIM                                                  00114000
      DO 381 J=1,NDIM                                                  00115000
381   CC(I,J)=0.                                                       00116000
      DO 370 I=1,NDIM                                                  00117000
370   CC(I,I)=1.                                                       00118000
      GO TO 390                                                        00119000
382   IF(ICON-2)384,384,386                                            00120000
384   DO 385 I=1,NDIM                                                  00121000
      DO 385 J=1,NDIM                                                  00122000
385   CC(I,J)=0.                                                       00123000
      DO 371 I=1,NDIM                                                  00124000
371   CC(I,I)=1./FM(I)                                                 00125000
      GO TO 390                                                        00126000
386   DO 360 I=1,NDIM                                                  00127000
      DO 360 J=1,NDIM                                                  00128000
      AA(I,J)=0.                                                       00129000
      DO 361 K=1,NSAMP                                                 00130000
361   AA(I,J)=AA(I,J)+((Y(K,I)-FM(I))*(Y(K,J)-FM(J)))                  00131000
360   AA(I,J)=AA(I,J)/ANSAMP                                           00132000
      DO 550 I=1,NDIM                                                  00133000
      DO 550 J=1,NDIM                                                  00134000
550   ST(I,J)=AA(I,J)                                                  00135000
C-----------------------------------------------------------------------00136000
C     INVERSION OF COVARIANCE MATRIX AA TO AI                          00137000
C-----------------------------------------------------------------------00138000
      NN=NDIM-1                                                        00139000
      AA(1,1)=1./AA(1,1)                                               00140000
      DO 500 M=1,NN                                                    00141000
      K=M+1                                                            00142000
      DO 501 I=1,M                                                     00143000
      BB(I)=0.                                                         00144000
      DO 501 J=1,M                                                     00145000
501   BB(I)=BB(I)+AA(I,J)*AA(J,K)                                      00146000
      D=0.                                                             00147000
      DO 502 I=1,M                                                     00148000
502   D=D+AA(K,I)*BB(I)                                                00149000
      D=-D+AA(K,K)                                                     00150000
      AA(K,K)=1./D                                                     00151000
      DO 503 I=1,M                                                     00152000
503   AA(I,K)=-BB(I)*AA(K,K)                                           00153000
      DO 504 J=1,M                                                     00154000
      CCC(J)=0.                                                        00155000
      DO 504 I=1,M                                                     00156000
504   CCC(J)=CCC(J)+AA(K,I)*AA(I,J)                                    00157000
      DO 505 J=1,M                                                     00158000
505   AA(K,J)=-CCC(J)*AA(K,K)                                          00159000
      DO 500 I=1,M                                                     00160000
      DO 500 J=1,M                                                     00161000
500   AA(I,J)=AA(I,J)-BB(I)*AA(K,J)                                    00162000
      DO 520 I=1,NDIM                                                  00163000
      DO 520 J=1,NDIM                                                  00164000
520   AI(I,J)=AA(I,J)                                                  00165000
```

FILE: KMEANS FORTRAN A 03/18/83 11:05 VM/SP CONVERSATIONAL MONITOR SYSTEM

```
      DO 387 I=1,NDIM                                                  00166000
      DO 387 J=1,NDIM                                                  00167000
```

```
387   CC(I,J)=AI(I,J)                                                  00168000
C-----------------------------------------------------------------------00169000
C     CHECK INVERSE AA*AI=I                                            00170000
C-----------------------------------------------------------------------00171000
      DO 530 I=1,NDIM                                                  00172000
      DO 530 J=1,NDIM                                                  00173000
      CM(I,J)=0.                                                       00174000
      DO 530 K=1,NDIM                                                  00175000
530   CM(I,J)=CM(I,J)+ST(I,K)*AI(K,J)                                  00176000
      WRITE(6,531)                                                     00177000
531   FORMAT(' ',//,' CHECK MATRIX AI*AA=1, THE IDENTITY'//)           00178000
      DO 532 I=1,NDIM                                                  00179000
532   WRITE (6,533) (CM(I,J),J=1,NDIM)                                 00180000
533   FORMAT(10X,20F6.2)                                               00181000
390   WRITE(6,1460) (TITLE(III),III=1,20)                              00182000
1460  FORMAT('1',10X,20A4///)                                          00183000
      WRITE(6,420)                                                     00184000
420   FORMAT(' ',///,15X,'SCALING MATRIX CC',///)                      00185000
      DO 421 I=1,NDIM                                                  00186000
421   WRITE(6,422) (CC(I,J),J=1,NDIM)                                  00187000
422   FORMAT(5X,10(F10.1,1X)/5X,10(F10.1,1X)/)                         00188000
      WRITE(6,425)                                                     00189000
425   FORMAT(/////)                                                    00190000
C-----------------------------------------------------------------------00191000
C     QQ IS THE BASIC EXPONENT FOR FUZZY ISODATA.                      00192000
C-----------------------------------------------------------------------00193000
      PP=(1./(QQ-1.))                                                  00194000
      DO 55555 NCLUS=KBEGIN,KCEASE                                     00195000
      WRITE(6,1460) (TITLE(III),III=1,20)                              00196000
      WRITE(6,499) NCLUS,ICON,QQ                                       00197000
499   FORMAT(' ',' NUMBER OF CLUSTERS = ',I3,5X,' ICON = ',I3,5X,      00198000
     C'EXPONENT = ',F4.2,//)                                           00199000
      IT=1                                                             00200000
C-----------------------------------------------------------------------00201000
C     RANDOM INITIAL GUESS FOR U(I,J)                                  00202000
C     THE RANDOM GENERATOR SUBROUTINE RANDU FROM THE IBM SCIENTIFIC    00203000
C     SUBROUTINE PACKAGE (SSP) IS USED AND IS CALLED FROM AN EXTERNAL  00204000
C     LIBRARY. OTHER GENERATORS THAT PRODUCE VALUES ON THE INTERVAL    00205000
C     ZERO TO ONE CAN BE USED.                                         00206000
C-----------------------------------------------------------------------00207000
      RANDOM=.7731                                                     00208000
      IX=1                                                             00209000
      NCLUS1=NCLUS-1                                                   00210000
      DO 1100 K=1,NSAMP                                                00211000
      S=1.0                                                            00212000
      DO 1101 I=1,NCLUS1                                               00213000
C     CALL RANDU(IX,IY,RANDOM)                                         00214000
      RANDOM=RANDOM/2.                                                 00215000
      IX=IY                                                            00216000
      ANC=NCLUS-I                                                      00217000
      U(I,K)=S*(1.0-RANDOM**(1.0/ANC))                                 00218000
1101  S=S-U(I,K)                                                       00219000
1100  U(NCLUS,K)=S                                                     00220000
```

FILE: KMEANS FORTRAN A 03/18/83 11:05 VM/SP CONVERSATIONAL MONITOR SYSTEM

```
C-----------------------------------------------------------------------00221000
C     CALCULATION OF CLUSTER CENTERS V(I).                             00222000
C-----------------------------------------------------------------------00223000
7000  DO 20 I=1,NCLUS                                                  00224000
      DO 20 J=1,NDIM                                                   00225000
      V(I,J)=0.                                                        00226000
      D=0.                                                             00227000
      DO 21 L=1,NSAMP                                                  00228000
      V(I,J)=V(I,J)+((U(I,L)**QQ)*Y(L,J))                              00229000
21    D=D+(U(I,L)**QQ)                                                 00230000
20    V(I,J)=V(I,J)/D                                                  00231000
C-----------------------------------------------------------------------00232000
C     UPDATE MEMBERSHIP FUNCTIONS.                                     00233000
C-----------------------------------------------------------------------00234000
6111  DO 38 I=1,NCLUS                                                  00235000
      DO 38 J=1,NSAMP                                                  00236000
      W(I,J)=0.                                                        00237000
      A=0.                                                             00238000
      DO 31 L=1,NDIM                                                   00239000
      DO 31 M=1,NDIM                                                   00240000
```

```
31    A=A+((Y(J,L)-V(I,L))*CC(L,M)*(Y(J,M)-V(I,M)))                    00241000
      A=1./(A**PP)                                                     00242000
      SUM=0.                                                           00243000
      DO 32 N=1,NCLUS                                                  00244000
      C=0.                                                             00245000
      DO 33 L=1,NDIM                                                   00246000
      DO 33 M=1,NDIM                                                   00247000
33    C=C+((Y(J,L)-V(N,L))*CC(L,M)*(Y(J,M)-V(N,M)))                    00248000
      C=1./(C**PP)                                                     00249000
32    SUM=SUM+C                                                        00250000
      W(I,J)=A/SUM                                                     00251000
38    CONTINUE                                                         00252000
C-----------------------------------------------------------------------00253000
C     ERROR CRITERIA AND CUTOFFS.                                      00254000
C-----------------------------------------------------------------------00255000
9000  ERRMAX=0.                                                        00256000
      DO 40 I=1,NCLUS                                                  00257000
      DO 40 J=1,NSAMP                                                  00258000
      ERR=ABS(U(I,J)-W(I,J))                                           00259000
      IF(ERR.GT.ERRMAX) ERRMAX=ERR                                     00260000
40    CONTINUE                                                         00261000
      WRITE(6,400) IT,ERRMAX,NCLUS                                     00262000
400   FORMAT(1H ,'ITERATION = ',I4,5X,'MAXIMUM ERROR = ',F10.4,        00263000
     110X,'NUMBER OF CLUSTERS = ',I4)                                  00264000
      DO 42 I=1,NCLUS                                                  00265000
      DO 42 J=1,NSAMP                                                  00266000
42    U(I,J)=W(I,J)                                                    00267000
      IF(ERRMAX.LE.EPS) GO TO 6000                                     00268000
43    IT=IT+1                                                          00269000
      IF(IT-LMAX) 7000,7000,6000                                       00270000
C-----------------------------------------------------------------------00271000
C     CALCULATION OF CLUSTER VALIDITY STATISTICS F, H, 1-E             00272000
C-----------------------------------------------------------------------00273000
6000  ITT(NCLUS)=IT                                                    00274000
      F(NCLUS)=0.0                                                     00275000
```

FILE: KMEANS FORTRAN A 03/18/83 11:05 VM/SP CONVERSATIONAL MONITOR SYSTEM

```
      H(NCLUS)=0.0                                                     00276000
      DO 100 I=1,NCLUS                                                 00277000
      DO 100 K=1,NSAMP                                                 00278000
      AU=U(I,K)                                                        00279000
      F(NCLUS)=F(NCLUS)+AU**2/ANSAMP                                   00280000
      IF (AU) 100,100,101                                              00281000
101   H(NCLUS)=H(NCLUS)-AU*ALOG(AU)/ANSAMP                             00282000
100   CONTINUE                                                         00283000
      DIF(NCLUS)=1.0-F(NCLUS)                                          00284000
C-----------------------------------------------------------------------00285000
C     CALCULATION OF OBJECTIVE FUNCTION                                00286000
C-----------------------------------------------------------------------00287000
      A=0.                                                             00288000
      DO 80 I=1,NCLUS                                                  00289000
      DO 80 J=1,NSAMP                                                  00290000
      DIST=0.                                                          00291000
      DO 81 L=1,NDIM                                                   00292000
      DO 81 M=1,NDIM                                                   00293000
81    DIST=DIST+((Y(J,L)-V(I,L))*CC(L,M)*(Y(J,M)-V(I,M)))              00294000
      A=A+((U(I,J)**QQ)*DIST)                                          00295000
80    VJM(NCLUS)=A                                                     00296000
C-----------------------------------------------------------------------00297000
C     OUTPUT BLOCK FOR CURRENT NCLUS                                   00298000
C-----------------------------------------------------------------------00299000
      WRITE(6,401)                                                     00300000
401   FORMAT(' '///' FSTOP',7X,'1-FSTOP',5X,'ENTROPY',5X,'PAYOFF',5X,/) 00301000
      WRITE(6,699) F(NCLUS),DIF(NCLUS),H(NCLUS),VJM(NCLUS)             00302000
699   FORMAT(1H ,2(F6.3,4X),4X,F6.3,5X,E8.3)                           00303000
      WRITE(6,59)                                                      00304000
59    FORMAT(1X,100('-')//)                                            00305000
      WRITE(6,402)                                                     00306000
402   FORMAT(///,15X,'CLUSTER CENTERS V(I,J)',///)                     00307000
      DO 415 I=1,NCLUS                                                 00308000
415   WRITE(6,404) (I,J,V(I,J),J=1,NDIM)                               00309000
404   FORMAT(' I=',I3,3X,'J=',I3,3X,'V(I,J)=',F8.4)                    00310000
405   FORMAT(1H ,7(F6.4,3X))                                           00311000
      WRITE(6,59)                                                      00312000
      WRITE(6,406)                                                     00313000
```

```
406   FORMAT(1H ,///,25X,'MEMBERSHIP FUNCTIONS',///)                    00314000
      DO 407 J=1,NSAMP                                                  00315000
407   WRITE(6,408) J,(U(I,J),I=1,NCLUS)                                 00316000
408   FORMAT(1H ,'J=',I3,5X,8(F6.4,3X))                                 00317000
54444 CONTINUE                                                          00318000
55555 CONTINUE                                                          00319000
C-----------------------------------------------------------------------00320000
C     OUTPUT SUMMARY FOR ALL VALUES OF C                                00321000
C-----------------------------------------------------------------------00322000
      WRITE(6,450)                                                      00323000
450   FORMAT('1',25X,'RUN SUMMARY')                                     00324000
      WRITE(6,460) NSAMP                                                00325000
460   FORMAT(' '///' NUMBER OF SUBJECTS N = ',I4)                       00326000
      WRITE(6,461) NDIM                                                 00327000
461   FORMAT(1HO,'NUMBER OF FEATURES NDIM = ',I4)                       00328000
      WRITE(6,462) EPS                                                  00329000
462   FORMAT(1HO,'MEMBERSHIP DEFECT BOUND EPS = ',F6.4)                 00330000

FILE: KMEANS   FORTRAN  A 03/18/82 11:05  VM/SP CONVERSATIONAL MONITOR SYSTEM

      WRITE(6,464) ICON                                                 00331000
464   FORMAT(1HO,'NORM THIS RUN ICON = ',I1)                            00332000
      WRITE(6,465) QQ                                                   00333000
465   FORMAT(1HO,'WEIGHTING EXPONENT M = ',F4.2)                        00334000
      IF(IT.LE.49) GO TO 476                                            00335000
      WRITE(6,70107)                                                    00336000
70107 FORMAT(' ','CONVERGENCE FLAG: UNABLE TO ACHIEVE SATISFACTORY CLUST00337000
     1ERS AFTER 50 ITERATIONS.')                                        00338000
476   WRITE(6,466)                                                      00339000
466   FORMAT(' '//' NO. OF CLUSTERS',3X,'PART. COEFF.',5X,              00340000
     C'LOWER BOUND',5X,'ENTROPY',5X,'NUMBER OF ITERATIONS')             00341000
      WRITE(6,467)                                                      00342000
467   FORMAT(1HO,6X,'C',17X,'F',15X,'1-F',12X,'H',10X,'IT')             00343000
      DO 468 J=KBEGIN,KCEASE                                            00344000
468   WRITE(6,469) J,F(J),DIF(J),H(J),ITT(J)                            00345000
469   FORMAT(1H ,6X,I2,14X,F6.3,11X,F6.3,7X,F6.3,8X,I4)                 00346000
55556 CONTINUE                                                          00347000
616   WRITE(6,411)                                                      00348000
411   FORMAT(////1H ,'*** *** NORMAL END OF JOB *** ***')               00349000
      STOP                                                              00350000
      END                                                               00351000
```