CHAPTER 4

The *K*-Means Algorithm

4.1 INTRODUCTION

The principal nutrients in meat, fish, and fowl are listed in Table 4.1. The foods are classified by food type and method of preparation. In the source book, *Yearbook of Agriculture* (1959), there are more details on the mode of preparation, as well as amounts of vitamins and basic amino acids, given for dairy products, meats, fowl, fish, vegetables, fruits, grains, oils, and sugars. There is a ready-made weighting scheme for the nutrients in the estimated daily dietary allowances: food energy (3200 cal), protein (70 g), calcium (0.8 g), and iron (10 mg). In Table 4.2, note that these foods deliver about 7% of the daily allowances in calories but about 25% of iron, so that the iron component is rather heavily weighted. An argument could be made that iron is less important than calories or protein and so should be given less weight or ignored entirely.

It is desired to partition the foods so that foods within clusters are close, in some sense, and foods in different clusters are distant. The discordance between the data and a given partition $P(M, K)$ of M objects into K clusters is measured by an error $e[P(M, K)]$. The very large number of possible partitions makes it impractical to search through all for the minimum of e. It is necessary instead to use the technique of local optimization. In this technique, a neighborhood of partitions is defined for each partition. Beginning with an initial partition, search through the set of partitions at each step, moving from a partition to that partition in its neighborhood for which $e[P(M, K)]$ is a minimum. (If the neighborhoods are very large, it is sometimes cheaper computationally to move to the first partition discovered in the neighborhood where $e[P(M, K)]$ is reduced from its present value.) A number of stopping rules are possible. For example, the search stops when $e[P(M, K)]$ is not reduced by movement to the neighborhood. The present partition is then locally optimal in that it is the best partition in its neighborhood.

As an example, consider partitions into three clusters of the beef foods BB, BR, BS, BC, and BH. A plausible neighborhood for a partition is the set of partitions obtained by transferring an object from one cluster to another. Thus, for the partition (BB BR) (BS) (BC BH), the neighborhood consists of the following ten partitions:

(BR) (BB BS) (BC BH),
(BR) (BS) (BB BC BH),
(BB) (BR BS) (BC BH),
(BB) (BS) (BR BC BH),
(BB BR BS) () (BC BH),
(BB BR) () (BS BC BH),
(BB BR BC) (BS) (BH),
(BB BR) (BS BC) (BH),
(BB BR BH) (BS) (BC),
(BB BR) (BS BH) (BC).

A typical search route beginning with (BB BR) (BS) (BC BH) might be

(BB BR) (BS) (BC BH)	with	$e = 8$
(BB BR) (BS BC) (BH)	with	$e = 6$
(BR) (BS BC) (BB BH)	with	$e = 5$
(BR BC) (BS) (BB BH)	with	$e = 4$.

No neighborhood of this partition reduces e, and the search stops.

4.2 K-MEANS ALGORITHM

Preliminaries. The Ith case of the Jth variable has value $A(I, J)$ $(1 \leq I \leq M, 1 \leq J \leq N)$. The variables are scaled so that euclidean distance between cases is appropriate. The partition $P(M, K)$ is composed of the clusters $1, 2, \ldots, K$. Each of the M cases lies in just one of the K clusters. The mean of the Jth variable over the cases in the Lth cluster is denoted by $B(L, J)$. The number of cases in L is $N(L)$. The distance between the Ith case and Lth cluster is

$$D(I, L) = (\sum \{1 \leq J \leq N\} [A(I, J) - B(L, J)]^2)^{1/2}.$$

The error of the partition is

$$e[P(M, K)] = \sum \{1 \leq I \leq M\} D[I, L(I)]^2,$$

where $L(I)$ is the cluster containing the Ith case. The general procedure is to search for a partition with small e by moving cases from one cluster to another. The search ends when no such movement reduces e.

STEP 1. Assume initial clusters $1, 2, \ldots, K$. Compute the cluster means $B(L, J)$ $(1 \leq L \leq K, 1 \leq J \leq N)$ and the initial error

$$e[P(M, K)] = \sum \{1 \leq I \leq M\} D[I, L(I)]^2,$$

where $D[I, L(I)]$ denotes the euclidean distance between I and the cluster mean of the cluster containing I.

STEP 2. For the first case, compute for every cluster L

$$\frac{N(L)D(1, L)^2}{N(L) + 1} - \frac{N[L(1)]D[1, L(1)]^2}{N[L(1)] - 1},$$

The increase in error in transferring the first case from cluster $L(1)$, to which it belongs at present, to cluster L. If the minimum of this quantity over all $L \neq L(1)$ is negative, transfer the first case from cluster $L(1)$ to this minimal L, adjust the cluster means of $L(1)$ and the minimal L, and add the increase in error (which is negative) to $e[P(M, K)]$.

STEP 3. Repeat Step 2 for the Ith case ($2 \leq I \leq M$).

STEP 4. If no movement of a case from one cluster to another occurs for any case, stop. Otherwise, return to Step 2.

Table 4.1 Nutrients in Meat, Fish, and Fowl

[*The Yearbook of Agriculture 1959* (The United States Department of Agriculture, Washington, D.C.) p. 244.] The quantity used is always 3 ounces.

		Food Energy (Calories)	Protein (Grams)	Fat (Grams)	Calcium (Milli Grams)	Iron (Milli Grams)
BB	Beef, braised	340	20	28	9	2.6
HR	Hamburger	245	21	17	9	2.7
BR	Beef, roast	420	15	39	7	2.0
BS	Beef, steak	375	19	32	9	2.6
BC	Beef, canned	180	22	10	17	3.7
CB	Chicken, broiled	115	20	3	8	1.4
CC	Chicken, canned	170	25	7	12	1.5
BH	Beef heart	160	26	5	14	5.9
LL	Lamb leg, roast	265	20	20	9	2.6
LS	Lamb shoulder, roast	300	18	25	9	2.3
HS	Smoked ham	340	20	28	9	2.5
PR	Pork roast	340	19	29	9	2.5
PS	Pork simmered	355	19	30	9	2.4
BT	Beef tongue	205	18	14	7	2.5
VC	Veal cutlet	185	23	9	9	2.7
FB	Bluefish, baked	135	22	4	25	.6
AR	Clams, raw	70	11	1	82	6.0
AC	Clams, canned	45	7	1	74	5.4
TC	Crabmeat, canned	90	14	2	38	.8
HF	Haddock, fried	135	16	5	15	.5
MB	Mackerel, broiled	200	19	13	5	1.0
MC	Mackerel, canned	155	16	9	157	1.8
PF	Perch, fried	195	16	11	14	1.3
SC	Salmon, canned	120	17	5	159	0.7
DC	Sardines, canned	180	22	9	367	2.5
UC	Tuna, canned	170	25	7	7	1.2
RC	Shrimp, canned	110	23	1	98	2.6

Table 4.2 Nutrients in Meat, Fish, and Fowl

As a percentage of recommended daily allowances.

	Food Energy	Protein	Fat (Grams)	Calcium	Iron
Beef, braised	11	29	28	1	26
Hamburger	8	30	17	1	27
Beef, roast	13	21	39	1	20
Beef, steak	12	27	32	1	26
Beef, canned	6	31	10	2	37
Chicken, broiled	4	29	3	1	14
Chicken, canned	5	36	7	2	15
Beef, heart	5	37	5	2	59
Lamb leg, roast	8	29	20	1	26
Lamb shoulder, roast	9	26	25	1	25
Ham, smoked	11	29	28	1	25
Pork roast	11	27	29	1	25
Pork simmered	11	27	30	1	25
Beef tongue	6	26	14	1	25
Veal cutlet	6	33	9	1	27
Bluefish, baked	4	31	4	3	6
Clams, raw	2	16	1	10	60
Clams, canned	1	10	1	9	54
Crabmeat, canned	3	20	2	5	8
Haddock, fried	4	23	5	2	5
Mackerel, broiled	6	27	13	1	10
Mackerel, canned	5	23	9	20	18
Perch, fried	6	23	11	2	13
Salmon, canned	4	24	5	20	7
Sardines, canned	6	31	9	46	25
Tuna, canned	5	36	7	1	12
Shrimp, canned	3	33	1	12	26

4.3 *K*-MEANS APPLIED TO FOOD NUTRIENT DATA

To keep the computations manageable, only the first eight foods will be considered for the nutrients' food energy, protein, and calcium as a percentage of recommended daily allowances. The eight foods will be partitioned in three clusters. The calculations appear in Table 4.3.

Table 4.3 Application of *K*-Means Algorithm to Food Data

	Energy	Protein	Calcium
BB	11	29	1
HR	8	30	1
BR	13	21	1
BS	12	27	1
BC	6	31	2
CB	4	29	1
CC	5	36	1
BH	5	37	2

INITIAL CLUSTER MEANS				e = 154.9
		Energy	Protein	Calcium
1.	HR, CB	8.5	25	1
2.	HR, BS	10	28.5	1
3.	BB, BC, CC, BH	6.75	33.25	1.5

FIRST CHANGE				e = 108.2
1.	HR, CB	8.5	25	1
2.	HR, BS, BB	10.33	28.67	1
3.	BC, CC, BH	5.33	34.67	1.67

SECOND CHANGE				e = 61.4
1.	BR	13	21	1
2.	HR, BS, BB	10.33	28.67	1
3.	BC, CC, BH, CB	5	33.25	1.5

STEP 1. A quick initial clustering, which often works well, is based on the case sums. Suppose these are denoted by SUM(I), having minimum value MIN and maximum value MAX. To obtain K initial clusters, set case I into the Jth cluster, where J is the integral part of K[SUM(I) − MIN]/(MAX − MIN) + 1. Here the case sums are 41, 39, 35, 40, 41, 34, 42, and 44. The corresponding clusters are 3, 2, 1, 2, 3, 1, 3, and 3. Thus the initial partition is (BR CB) (HR BS) (BB BC CC BH). The values of $B(L, J)$ ($1 \leq L \leq 3$, $1 \leq J \leq 3$) are next computed. For example, $B(1, 1)$, the mean of cases in the first cluster for the first variable, equals $(13 + 4)/2 = 8.5$. (See Table 4.3 for more.) The error for the initial partition is the sum of squared distances of cases from their cluster means,

$$\begin{aligned} e[P(8,3)] = {} & (11 - 6{\cdot}75)^2 + (29 - 33{\cdot}25)^2 + (1 - 1{\cdot}5)^2 + (8 - 10)^2 \\ & + (30 - 28{\cdot}5)^2 + (1 - 1)^2 + (13 - 8{\cdot}5)^2 + (21 - 25)^2 \\ & + (1 - 1)^2 + (12 - 10)^2 + (27 - 28{\cdot}5)^2 + (1 - 1)^2 \\ & + (6 - 6{\cdot}75)^2 + (31 - 33{\cdot}25)^2 + (2 - 1{\cdot}5)^2 + (8{\cdot}5 - 4)^2 \\ & + (29 - 25)^2 + (1 - 1)^2 + (5 - 6{\cdot}75)^2 + (36 - 33{\cdot}25)^2 \\ & + (1 - 1{\cdot}5)^2 + (5 - 6{\cdot}75)^2 + (37 - 33{\cdot}25)^2 + (2 - 1{\cdot}5)^2 \\ = {} & 154.9. \end{aligned}$$

The first three squares are the squared distance of BB from its cluster mean (6·75, 33·25, 1·5), and so on.

STEP 2. For the first case, the distances to clusters are

$$D(1,1)^2 = (11 - 8{\cdot}5)^2 + (29 - 25)^2 + (1 - 1)^2 = 22.25,$$
$$D(1,2)^2 = 1.25,$$
$$D(1,3)^2 = 36.4.$$

The increase in error in transferring the first case to cluster 1 is $2 \times 22.5/3 - 4 \times 36.4/3$, and that to cluster 2 is $2 \times 1.25/3 - 4 \times 36.4/3$. The cluster that is best for the first case is thus the second cluster, and the error reduction is 47.7. The new value of $e[P(8,3)]$ is thus $154.9 - 47.7 = 108.2$.

It is necessary to update the means of clusters 2 and 3, since cluster 2 has gained the first case and cluster 3 has lost it. For example,

$$B(2,1) = (11 + 2 \times 10)/3 \quad = 10.33,$$
$$B(2,2) = (29 + 2 \times 28.5)/3 = 28.67,$$
$$B(2,3) = (1 + 2 \times 1)/3 \quad = 1.00.$$

STEP 3. Repeating Step 1 on all cases, for case 2, cluster 2 is far closer than any other, and case 2 remains in cluster 2, with no change taking place. Continuing, no change takes place until case 6, which moves to cluster 3. No further changes occur in this pass.

STEP 4. Since some changes occurred in the last pass, another pass is necessary through all cases. No changes occur on this pass and the algorithm stops with the final cluster (BR) (HR BS BB) (BC CC BH CB). These clusters are characterized by the variables as follows: The first cluster is high in energy and low in protein, the second is high in energy and protein, and the third is low in energy and high in protein Calcium hardly matters. The complete data set is partitioned in Table 4.4.

4.4 ANALYSIS OF VARIANCE

Some distributions which appear frequently in the analysis of variance are the following:

(i) the normal distribution $N(\mu, \sigma^2)$, which has mean μ and variance σ^2 and density $\exp[-\frac{1}{2}(x-\mu)^2\sigma^{-2}]/\sigma\sqrt{2\pi}$. The unit normal is $N(0,1)$, having mean 0 and variance 1;

(ii) the chi-square distribution χ^2_n, which is the distribution of a sum of squares of n independent unit normals; and

(iii) the F distribution $F_{m,n}$, which is the ratio of independent standardized chi squares, $(\chi^2_m/m)(\chi^2_n/n)^{-1}$.

Suppose $P(M, K)$ is a partition of M objects into K clusters, and let $e(M, K, J) = \sum\{1 \leq I \leq M\}\{A(I,J) - B[L(I),J]\}^2$, where the case I lies in cluster $L(I)$ and $B(L, J)$ is the mean of the Jth variable over cases in cluster L. If the clusters are selected without regard to the Jth variable and if $A(I,J) \sim N\{\mu[L(I)], \sigma^2\}$ independently for each I, $e(M,K,J) \sim \sigma^2\chi^2_{M-K}$. Furthermore, if the partition $P(M, K+1)$ is obtained

Table 4.4

Clusters and cluster means from K-means algorithm applied to food data on energy, protein, and calcium expressed as percentages of daily requirements. The two clusters obtained by splitting cluster 3 are denoted by 31 and 32.

PARTITION	CLUSTERS	ENERGY	PROTEIN	CALCIUM
1	1 : BB HR BR BS BC CB CC BH LL LL HS PR PS BT VC FB AR AC TC HF MB MC PF SC DS UC RC	6.5	27.1	5.5
2	11 : BB HR BR BS BC CB CC BH LL LL HS PR PS BT VC FB AR AC TC HF MB PF UC RC	6.7	27.3	2.6
	12 : MC SC DS	4.7	26.1	28.5
3	12			
	111 : BB HR BR BS BC CB CC BH LL LL HS PR PS BT VC FB HF MB PF UC RC	7.4	29.0	1.8
	112 : AR AC TC	2.1	15.2	8.1
4	112			
	2 : BB HR BR BS BC CB CC BH LL LL HS PR PS BT VC FB HF MB PF UC	7.5	28.8	1.3
	3 : MC SC RC	4.0	26.7	17.3
	4 : DS	5.6	31.4	45.9
5	112, 3, 4			
	21 : HR BC CB CC BH VC FB UC	5.6	32.4	1.5
	22 : BB BR BS LL LL HS PR PS BT HF MB PF	9.1	25.8	1.2
6	112, 21, 22, 4			
	31 : MC SC	4.3	23.6	19.8
	32 : RC	3.4	32.9	12.3
7	112, 21, 31, 32, 4			
	221 : BB BS LL LL HS PR PS BT HF MB PF	8.7	26.5	1.2
	222 : BR	13.1	21.4	.9
8	222, 31, 32, 4			
	5 : BC CC BH VC FB UC	5.2	34.0	1.7
	6 : AR AC	5.6	31.4	45.9
	7 : BB HR BS LL LL HS PR PS MB	9.6	27.8	1.1
	8 : CB BT TC HF PF	4.6	24.0	2.1
9	222, 31, 32, 4, 5, 6,			
	9 : BB HR BS LL LL HS PR PS	10.0	27.9	1.1
	10 : CB BT HF MB PF	5.3	25.4	1.2
	11 : TC	2.8	20.0	4.8

by splitting one of the clusters in $P(M, K)$, then the mean square ratio

$$\left(\frac{e(M, K, J)}{e(M, K+1, J)} - 1\right)(M - K - 1) \sim F_{1, M-K-1}.$$

The ratio is a measure of the reduction of within-cluster variance for the Jth variable between the partitions $P(M, K)$ and $P(M, K+1)$. The F distribution is not correct for evaluating K-means partitions because each variable influences the partition. The partition $P(M, K+1)$ is chosen to minimize

$$\sum \{1 \leq J \leq N\}\, e(M, K+1, J),$$

and this will tend to increase all the mean square ratios. Also, the partition $P(M, K+1)$ is not necessarily obtained by splitting one of the clusters in $P(M, K)$, so the mean

square ratio is conceivably negative. Nevertheless, as a crude rule of thumb, large values of the ratio (say, >10) justify increasing the number of clusters from K to $K+1$.

Suppose again that $P(M, K)$ is a given partition into K clusters, that $P(M, K+1)$ is obtained from it by splitting one of the clusters, and that

$$A(I, J) \sim N\{\mu[L(I), J], \sigma^2\}$$

independently over all I and J.

Then the overall mean square ratio

$$R = \left(\frac{e[P(M, K)]}{e[P(M, K+1)]} - 1\right)(M - K + 1) \approx F_{N,(M-K-1)N}.$$

Again this F distribution is not applicable in the K-means case, because the partition $P(M, K+1)$ is chosen to maximize the overall mean square ratio. Again, as a crude rule of thumb, overall mean square ratios greater than 10 justify increasing partition size.

Some ratio measures are given for the food data in Table 4.5. For the variables, notice that calcium is very much reduced at the second, fourth, and sixth partitions,

Table 4.5 Ratio Due to K-Partition

Decrease in the sum of squares from the $(K-1)$th to the Kth partition, divided by the mean sum of squares within the Kth partition.

MAXIMUM CLUSTER SIZE	PARTITION SIZE	OVERALL	ENERGY	PROTEIN	CALCIUM
24	2	23.6	1.0	0.1	64.1
21	3	13.0	9.6	25.8	4.1
20	4	18.4	2.4	0	200.8
12	5	16.1	15.4	20.3	0.3
12	6	6.7	0.1	6.6	35.0
11	7	3.2	4.8	3.0	0.1
9	8	12.5	18.4	10.4	11.6
8	9	6.5	11.9	3.5	31.9

while energy and protein are reduced for the third and fifth. The larger ratios for calcium follow from the large initial variance for calcium [the initial variances are 10 (energy), 37 (protein), and 95 (calcium)].

Plausible stopping points in the clustering are $K = 2$, $K = 5$, and $K = 8$, where the ratios are unusually large.

4.5 WEIGHTS

The weights will depend on considerations outside the data. If persons eating the food were known to have a diet abundant in calcium, then the calcium component would be down-weighted. If protein were scarce in other foods, then it would be given

more weight. It is clear that in the initial analysis calcium was much the most important variable in the first few partitions. This is partly due to the scaling and partly due to the good clustering qualities of calcium, which is extremely high in a few sea foods and low elsewhere.

Another weighting scheme scales all variables to equal variance. As previously explained, this may be self-defeating in reducing the weight of variables that cluster well. Another weighting scheme repeats the partition a number of times, with one of the variables given weight 0 each time. The effect of the omitted variable on the clustering may be evaluated in this way. The 8-partitions corresponding to these weighting schemes are given in Table 4.6. Note there that calcium and protein are the best

Table 4.6 Effect of Changing Weights on 8-Partitions of Food Data by *K*-Means Algorithm

MEAN SQUARE ERROR WITHIN CLUSTERS

WEIGHTING	Energy$_2$ (Calories^2)	Protein (Grams^2)	Fat (Grams^2)	Calcium (Mgms^2)	Iron (Mgms^2)
% DAILY ALLOWANCE	4665	4.1	60	11	.99
EQUAL VARIANCE	924	3.0	10	1500	.42
OMITTING ENERGY	1392	3.8	17	1199	.45
OMITTING PROTEIN	1151	19.5	11	320	.20
OMITTING FAT	1632	3.6	20	382	.41
OMITTING CALCIUM	816	2.6	9	6787	.36
OMITTING IRON	791	4.1	9	167	1.25

clustering variables in that their omission from the sum of squares to be minimized vastly increases their mean square error within the clusters obtained. Iron is the worst in that omitting it does not much increase the iron mean square and does much reduce the other mean squares. Since iron is subjectively less important anyway, a good final scheme would be to weight so that all variables would have variance 1 except iron, which would have variance 0.

4.6 OTHER DISTANCES

The *K*-means algorithm searches for partitions $P(M, K)$ with a small error

$$e[P(M, K)] = \sum \{1 \leq I \leq M\}\, D^2[I, L(I)],$$

where D is the euclidean distance from I to the average object in $L(I)$, the cluster to which I belongs. The essential characteristics of the *K*-means method are the search method, changing partitions by moving objects from one cluster to another, and the measure of distance. Euclidean distance leads naturally to cluster means and an analysis of variance for each of the variables.

To consider more general measures of distance, denote the Ith case $\{A(I, J), 1 \leq J \leq N\}$ by $\mathbf{A}(I)$, and let $\mathbf{B}(L) = \{B(L, J), J = 1, \ldots, N\}$ denote a set of values corresponding to the Lth cluster.

A distance $F[\mathbf{A}(I), \mathbf{A}(K)]$ is defined between the Ith and Kth cases. The central case of the Lth cluster is $\mathbf{B}(L)$ minimizing $\sum \{I \in L\} F[\mathbf{A}(I), \mathbf{B}(L)]$.

The error of a particular partition is

$$e[P(M, K)] = \sum \{1 \leq I \leq M\} F\{\mathbf{A}(I), \mathbf{B}[L(I)]\}$$

where $L(I)$ is the cluster containing I.

Locally optimal clusters may be obtained by moving cases from one cluster to another, if this decreases e, and updating the central cases after each movement. For example, if the distance between two cases is the sum of the absolute deviations between the cases over variables, then the central case of a cluster is the median for each variable. The contribution of a cluster to the error is the sum of absolute deviations from the median, over all objects in the cluster and over all variables.

Approaching the error function from a statistical point of view, the cases $\{A(I, J), 1 \leq J \leq N\}$ will be denoted by $\mathbf{A}(I)$, and the parameters $\boldsymbol{\theta}(1), \boldsymbol{\theta}(2), \ldots, \boldsymbol{\theta}(K)$ will be associated with each of the clusters. The cases in cluster L are a random sample from a probability distribution determined by $\boldsymbol{\theta}(L)$; the probability of observing these cases is

$$\prod P[\mathbf{A}(I) \mid \boldsymbol{\theta}(L)],$$

where the product is over the cases in the Lth cluster. The probability of observing all cases is

$$\prod \{1 \leq I \leq M\} P\{\mathbf{A}(I) \mid \boldsymbol{\theta}[L(I)]\}.$$

The error function associated with the partition $P(M, K)$ is minus the log likelihood:

$$e = -\sum \{1 \leq I \leq M\} \log P\{\mathbf{A}(I) \mid \boldsymbol{\theta}[L(I)]\}.$$

To minimize this error function for a particular partition, the parameters $\boldsymbol{\theta}(L)$ are chosen by maximum likelihood. Searching over all possible partitions may now take place by using the K-means procedure. Choosing the values $\boldsymbol{\theta}(L)$ corresponds to selecting the cluster "centers" $B(L, J)$ in the K-means procedure. The probability distribution $P[\mathbf{A}(I) \mid \boldsymbol{\theta}(L)]$ specifies the joint distribution of all variables, given the cluster center $\boldsymbol{\theta}(L)$. If the variables are independent normal with equal variance and mean vector $\boldsymbol{\theta}(L)$, the K-means error function is minus the log likelihood as indicated above. Note that this requires that the variables have equal variance *within* clusters.

Independent Laplace distributions for each variable within clusters implies a distance function summing absolute deviations. Uniform distributions within clusters implies a distance between cases equal to the maximum deviations between the cases, over variables.

A further refinement is to use Bayes techniques, with some prior distribution of the parameters, $\boldsymbol{\theta}(1), \ldots, \boldsymbol{\theta}(K)$, which will be assumed to be independent and identical random variables. The random variables $\mathbf{A}(I)$ are marginally independent between clusters but dependent within, so that the probability of the observed cases is $\prod \{1 \leq L \leq K\} P(L)$, where $P(L)$ is the probability of the cases lying in cluster L. The value $-\log P(L)$ is a reasonable measure of cluster diameter, $e = \sum [-\log P(L)]$ is a measure of partition error, and the same search procedure as before is used to find good partitions.

4.7 THE SHAPE OF K-MEANS CLUSTERS

Some properties of the K-means algorithm, including the convexity of the clusters, are discussed in Fisher and Van Ness (1971). Consider a partition that is locally optimal in

$$e[P(M, K)] = \sum \{1 \leq I \leq M\} D^2[I, L(I)]$$

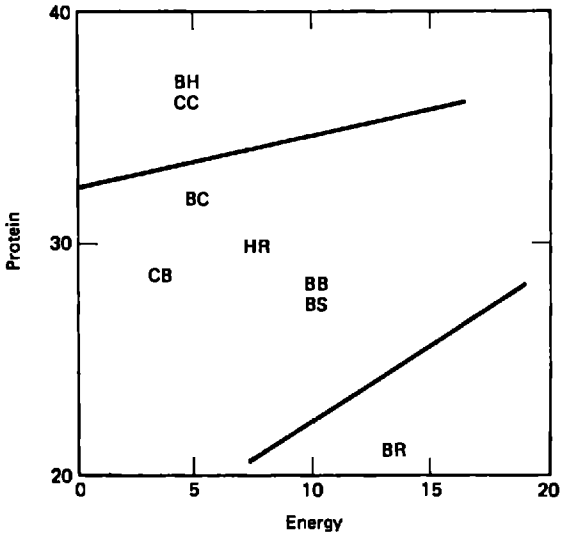

Figure 4.1 Clusters separated by hyperplanes based on food data in Table 4.3.

using the *K*-means search pattern. Here $L(I)$ is the cluster containing case I, and $D(I, L)$ is the distance between case I and the mean of cases in cluster L.

Let $L1$ and $L2$ be two different clusters. For each I in $L1$, $D(I, L1) < D(I, L2)$, for otherwise I would be removed from $L1$ during step 2 of the algorithm. Therefore, each case I in $L1$ satisfies

$$\sum \{1 \leq J \leq N\} [B(L1, J) - B(L2, J)]A(I, J) > c,$$

where

$$c = \sum \{1 \leq J \leq N\} \tfrac{1}{2}[B(L1, J)^2 - B(L2, J)^2],$$

and each case I in $L2$ satisfies

$$\sum \{1 \leq J \leq N\} [B(L1, J) - B(L2, J)]A(I, J) < c.$$

Geometrically, the cases in $L1$ and $L2$ are separated by a hyperplane normal to $\{B(L1, J) - B(L2, J)\}$ (see Figure 4.1). (This hyperplane is a linear discriminant function for separating cases into the clusters $L1$ and $L2$.)

Each cluster is convex, which means that a case lies in a cluster if and only if it is a weighted average of cases in the cluster. To show this, suppose that case II lies in cluster $L2$, but it is a weighted average of cases in cluster $L1$. For all cases in $L1$

$$\sum \{1 \leq J \leq N\} [B(L1, J) - B(L2, J)]A(I, J) > c,$$

and the reverse holds for cases in $L2$.

Since $A(II, J) = \sum W(I)A(I, J)$, where the summation is over cases in cluster $L1$ and $W(I) \geq 0$,

$$\sum \{1 \leq J \leq N\} [B(L1, J) - B(L2, J)]A(II, J) > c.$$

Therefore case II does not lie in $L2$. Thus each cluster is convex, as required.

The partition that is optimal over all partitions is also, of course, locally optimal in the neighborhood of the *K*-means search procedure. The globally optimal clusters are therefore also convex. In searching for a globally optimal partition, it is necessary to consider only convex clusters, and in certain special cases this constraint makes it feasible to insist on the global optimum.

In univariate problems, the convexity requirement means that clusters consist of all the cases lying in an interval. The clustering proceeds first by ordering all M cases and then by applying the Fisher algorithm (Chapter 6) to the ordered points. For example, the optimal 2-partition consists of clusters of the form $\{I \mid A(I, 1) < c\}$ and $\{I \mid A(I, 1) > c\}$, where there are only M choices of c. For the protein data in Table 4.3, the case values are BB = 29, HR = 30, BR = 21, BS = 27, BC = 31, CB = 29, CC = 36, and BH = 37. The clusters must be intervals in the sequence BR BS CB BB HR BC CC BH. The 2-partition is settled by trying the eight possible cuts giving (BR) (BS CB BB HR BC CC BH). The 3-partition is (BR) (BS CB BB HR BC) (CC BH), and so on.

For 2-partitions with two variables, convexity requires that each partition be determined by a line. The first cluster lies on one side of the line and the second cluster lies on the other. The number of such partitions is $M(M-1)/2 + 1$. It is thus feasible for just two variables to find exactly optimal 2-partitions. The number of 2-partitions for N variables is $\sum \{0 \leq J \leq N\} \binom{M-1}{J}$, where $\binom{M-1}{J}$ is the number of ways of choosing J objects from $M-1$. For reasonably large N—say, $N > 4$—it quickly becomes impractical to obtain the optimal 2-partition by searching over hyperplanes.

To find the optimal 2-partition, consider a projection $V(I) = \sum \{1 \leq J \leq N\} E(J)A(I, J)$, where the coefficients $E(J)$ sum square to unity. Find the optimal 2-partition of the variable V and compute the mean square error between clusters—say, $B(V)$. The optimal 2-partition of all the data is the optimal 2-partition of V for that V maximizing $B(V)$. Searching over all coefficients $E(J)$ corresponds to searching over all hyperplanes. The error $B(V)$ is continuous but not differentiable everywhere as a function of the $E(J)$; it has many local maxima, and so setting derivatives equal to zero is useless. Fix $J = J1$ and consider coefficients $E(J1)$, $\{\alpha E(J), J \neq J1\}$, where $E(J1)$ and α vary but $E(J)$ for $J \neq J1$ are fixed. The variable V is determined by a single parameter, and the optimal $E(J1)$ may be found by using a similar procedure to the two-variable problem. In this way, each of the $E(J)$'s may be optimized one at a time. The final partition is not guaranteed globally optimal.

Now, asymptotic arguments will be used to speculate about cluster shape. For M infinite, with one variable, normally distributed, the clusters for 2, 3, 4, . . . , partitions have been computed by Cox (1957) and Dalenius (1951). For example, the 2-partition contains clusters $(-\infty, 0)$ $(0, \infty)$, each with 50% of the cases. The 3-partition contains clusters $(-\infty, -0.612)$ $(-0.612, 0.612)$ $(0.612, \infty)$, containing 27%, 46%, 27%, and so on (see Table 4.8). Note that the length of intervals increases towards the tails and the proportion of cases contained decreases. The cut points must be equidistant from the cluster means on either side, and the cluster means are determined by integration between the cutpoints. Beginning with an initial set of cutpoints, the cluster means are computed, then the cut points are taken anew as the averages of the neighboring cluster means, and so on until convergence. This is very similar to the *K*-means algorithm in concept and will not necessarily

Table 4.7 Relation between Weights and Mean Square Error within Clusters

For 27 observations from five-dimensional spherical normal. The weights used are 1, 2, 3, 4, 5. Table entries are weights multiplied by the mean square error within clusters.

	VARIABLE				
PARTITION SIZE	1	2	3	4	5
1	1.2	1.9	2.7	3.7	4.5
2	1.1	1.8	2.5	3.3	1.9
3	1.0	1.7	1.9	2.4	1.9
4	1.0	1.7	1.5	1.8	1.7
5	.9	1.6	1.3	1.3	1.5
10	.7	.8	1.1	1.0	.8
15	.5	.6	.5	.6	.6

converge to the optimum partition for any distribution (one with many modes, for example).

For N variables, M infinite, assume some joint distribution with continuous density on the variables such that each variable has finite variance. As $K \to \infty$, the mean square error within clusters approaches zero. If the joint density is positive everywhere, for each case there must be a cluster mean arbitrarily close to the case for K large enough. There will be some asymptotic N-dimensional distribution of cluster means

Table 4.8 Optimal Clusters of Normal Distribution

PARTITION SIZE		PROPORTION IN CLUSTERS					
2	0	.50	.50				
3	$\pm$ .612	.27	.46	.27			
4	0, $\pm$ .980	.16	.34	.34	.16		
5	$\pm$ 1.230, $\pm$ .395	.11	.24	.31	.24	.11	
6	0, $\pm$ 1.449, $\pm$ .660	.07	.18	.25	.25	.18	.07

that will depend on the original N-dimensional distribution of cases. In the neighborhood of a point, the density of cases is nearly constant. A certain large number of cluster means will be located in the neighborhood, and an approximately optimal location would occur if the density were exactly constant in the neighborhood. Therefore, there is no reason for the cluster shapes to be oriented in any one direction (the shapes will not be spheres, but polyhedra) and the within-cluster covariance matrix will be proportional to the identity matrix.

This is a heuristic argument. It has an important practical consequence. For large M and K, the within-cluster covariance matrix will be proportional to the identity matrix. Thus a hoped-for iterative weighting procedure may not work. It is desired to obtain weights from the within-cluster variances. The clusters must first be computed by using equal weights. The above argument shows, at least for large K, that the final within-cluster variances will be nearly equal and no change in weights will be suggested.

To test this empirically, 27 observations from a five-dimensional normal with mean zero and unit covariance matrix were clustered, using a distance function,

$$D(I, L) = \sum \{1 \leq J \leq 5\}\, W(J)[A(I, J) - A(L, J)]^2,$$

where the weights $W(J)$ take the values 1, 2, 3, 4, 5. In a later step, the inverses of the within-cluster variances might be used as weights. In Table 4.7, it will be seen that the inverses of the within-cluster variances approach the original weights as the number of clusters increases.

Thus in specifying the weights you are specifying the within-cluster variances. In specifying between-variable weights, you are specifying the within-cluster covariance matrix to be the inverse of the weight matrix. These consequences occur only for a large number of clusters, so that if the clustering is stopped soon enough there may be some value to iteration. For example, if there are very distinct clusters separated well by every variable, the partitioning might stop when these clusters are discovered and the weights might be based on these within-cluster variances. But, of course, if there are such distinct clusters, they will appear under any weighting scheme and careful choice of the weights is not important.

4.8 SIGNIFICANCE TESTS

Consider the division of M observations from a single variable into two clusters minimizing the within-cluster sum of squares. Since this is the maximum likelihood division, under the model that observations in the first cluster are $N(\mu_1, \sigma^2)$ and observations in the second cluster are $N(\mu_2, \sigma^2)$, it will be plausible to test $\mu_1 = \mu_2$ versus $\mu_1 \neq \mu_2$ on this normal model.

Let $L(I) = 1$ or 2, according as the observation $X(I)$ lies in the first or second cluster. Define

$$N(J) = \sum \{L(I) = J\}1 \qquad \text{(the number of observations in cluster } J\text{)},$$

$$Y(J) = \sum \{L(I) = J\} \frac{X(J)}{N(J)} \qquad \text{(the average in cluster } J\text{)},$$

$$\text{SSW} = \sum \{L(I) = J\}[X(I) - Y(J)]^2,$$

$$\text{SSB} = \frac{[Y(1) - Y(2)]^2}{1/N(1) + 1/N(2)}.$$

The likelihood ratio criterion is monotone in SSB/SSW, rejecting $\mu_1 = \mu_2$ if this quantity is large enough. The empirical distribution of this quantity for less than 50 observations is tabulated in Engelman and Hartigan (1969).

Suppose $\mu_1 = \mu_2 = 0$. It is sufficient to consider partitions where the first cluster consists of observations less than some split point c, and the second cluster consists of observations greater than c. Asymptotically, SSB and SSW vary negligibly over splits in the neighborhood of μ_1, so the split may be assumed to occur at $\mu_1 = 0$ and the second cluster of observations will be a sample from the half-normal. The half-normal density $f(x) = \exp(-\frac{1}{2}x^2)\sqrt{2/\pi}$, $(x > 0)$ has mean $\sqrt{2/\pi}$, variance $1 - 2/\pi$, third moment $\sqrt{2/\pi}(4/\pi - 1)$, and fourth moment $3 - 4/\pi - 12/\pi^2$. From this it

follows, by using standard asymptotic normal theory on the sums SSB and SSW,

$$\text{SSB} \approx N\left(\frac{2M}{\pi}, \frac{8(\pi-2)M}{\pi^2}\right),$$

$$\text{SSW} \approx N\left[M\left(1-\frac{2}{\pi}\right), 2M\left(1-\frac{8}{\pi^2}\right)\right],$$

and the covariance between them is $M(16/\pi^2 - 4/\pi)$. Note that SSB + SSW $\approx$ $N(M, 2M)$, which is correct because SSB + SSW is just the sum of squared deviations from the overall mean.

More generally, for an arbitrary symmetric parent distribution X, for which the optimal split point converges asymptotically to zero, define

$$\mu(1) = E|X|$$

and

$$\mu(I) = E[|X| - \mu(1)]^I.$$

Then

$$\text{SSB} \approx N[M\mu(1)^2, 4M\mu(1)^2\mu(2)]$$

and

$$\text{SSW} \approx N\{M\mu(2), [\mu(4) - \mu(2)^2]M\},$$

with covariance $2\mu(3)\mu(1)$. It follows that SSB/SSW is approximately normal with mean $\mu(1)^2/\mu(2)$ and variance

$$\left(\frac{4\mu(1)^2}{\mu(2)} + \frac{[\mu(4) - \mu(2)^2]\mu(1)^4}{\mu(2)^4} - \frac{4\mu(3)\mu(1)^3}{\mu(2)^3}\right)M^{-1}$$

(you should forgive the expression).

In the normal case,

$$\frac{\text{SSB}}{\text{SSW}} \approx N\left(\frac{2}{\pi-2}, \left(\frac{8}{\pi}\right)\left(1-\frac{3}{\pi}\right)\left(1-\frac{2}{\pi}\right)^{-4}M^{-1}\right)$$

or

$$\frac{\text{SSB}}{\text{SSW}} \approx N\left(1.75, \frac{6.58}{M}\right).$$

This asymptotic distribution is not applicable except for very large M because SSB/SSW is extremely skew. Empirical investigation shows that log (SSB/SSW) is nearly nonskew for small M (say, $M > 8$) and that the actual distribution is much closer to the asymptotic one,

$$\log\left(\frac{\text{SSB}}{\text{SSW}}\right) \approx N\left(0.561, \frac{2.145}{M}\right).$$

A comparison between SSB/SSW and log (SSB/SSW) for small sample sizes is given in Table 4.9 (see also Engelman and Hartigan, 1969). There remains a substantial bias in log (SSB/SSW) that is incorporated in the formula

$$\log\left(\frac{\text{SSB}}{\text{SSW}}\right) \approx N\left(0.561 + \frac{0.5}{M-1}, \frac{2.145}{M}\right).$$

Table 4.9 Empirical Distribution of SSB/SSW

Using random samples from a normal distribution, with 100 repetitions for sample size $n = 5$ and 10, with 200 repetitions for sample sizes $n = 20$ and 50.

SAMPLE SIZE		SSB/SSW		LOG (SSB/SSW)	
		MEAN	VARIANCE	MEAN	VARIANCE
5	OBSERVED	8.794	382.007	1.708	.582
	ASYMPTOTIC	1.752	1.316	.561	.429
10	OBSERVED	3.286	2.613	1.093	.180
	ASYMPTOTIC	1.752	.658	.561	.214
20	OBSERVED	2.316	.565	.792	.089
	ASYMPTOTIC	1.752	.329	.561	.107
50	OBSERVED	2.004	.207	.668	.052
	ASYMPTOTIC	1.752	.132	.561	.042

The quantity $\mu(1)^2/\mu(2)$ is a measure of the degree of bimodality of the distribution. It will be a maximum when the distribution (assumed symmetric) is concentrated at two points and a minimum (zero) for long-tailed distributions with infinite variance but finite first moment. Since SSB/SSW estimates this quantity, for symmetric distributions it might be better to estimate it directly by

$$\mu_1 = \sum \{1 \le I \le M\} \frac{|X(I) - \text{XBAR}|}{M}$$

$$\mu_2 = \sum \frac{[X(I) - \text{XBAR}]^2}{M} - \mu_1^2,$$

where $\text{XBAR} = \sum \{1 \le I \le M\} X(I)/M$. This method of estimation is faster than the splitting method, which requires ordering the M observations and then checking M possible splits. In a similar vein, a quick prior estimate of within-cluster variance for weighting purposes (the estimate works well if the two clusters are of approximately equal size) is

$$\sum \{1 \le I \le M\} \frac{[X(I) - \text{XBAR}]^2}{M} - \left(\sum \{1 \le I \le M\} \frac{|X(I) - \text{XBAR}|}{M}\right)^2.$$

In N dimensions, assume that the null distribution is multivariate normal and that the covariance matrix has eigenvalues $E(1) > E(2) > \cdots > E(N)$. The asymptotic argument reduces to the one-dimensional case by orthogonal transformation to the independent normal variables with variances $E(1), \ldots, E(N)$. The split will be essentially based on the first variable, and the remaining variables will contribute standard

chi-square-like terms to the sums of squares:

$$\text{SSB} \approx N\left(\frac{2ME(1)}{\pi}, 8(\pi-2)M\frac{E(1)^2}{\pi^2}\right),$$

$$E(\text{SSW}) = M\sum\{1 \le I \le N\}E(I) - 2M\frac{E(1)}{\pi},$$

$$\text{VAR}(\text{SSW}) = 2M\sum\{1 \le I \le N\}E(I)^2 - 16M\frac{E(1)^2}{\pi^2},$$

$$\text{COV}(\text{SSB}, \text{SSW}) = ME(1)^2\left(\frac{16}{\pi^2} - \frac{4}{\pi}\right).$$

From this, asymptotically,

$$E\left(\frac{\text{SSB}}{\text{SSW}}\right) = \frac{2E(1)}{\pi}\left(\sum\{1 \le I \le N\}E(I) - \frac{2E(1)}{\pi}\right)^{-1},$$

$$\text{VAR}\left(\frac{\text{SSB}}{\text{SSW}}\right) = \left[\left(\frac{8}{\pi^2}\right)E(1)^2\left(\sum\{1 \le I \le N\}E(I)^2 + (\pi-2)\left[\sum\{1 \le I \le N\}E(I)\right]^2\right) - \left(\frac{16}{\pi^2}\right)E(1)^3\sum\{1 \le I \le N\}E(I)\right]\left[M\left(\frac{E(\text{SSW})}{M}\right)^4\right]^{-1}.$$

For N large, with $E(1)$ making a relatively small contribution to $\sum\{1 \le I \le N\}E(I)$,

$$E\left(\frac{\text{SSB}}{\text{SSW}}\right) = \frac{2E(1)}{\pi}\left(\sum\{1 \le I \le N\}E(I)\right)^{-1}$$

and

$$\text{VAR}\left(\frac{\text{SSB}}{\text{SSW}}\right) = \left(\frac{8}{\pi^2}\right)E(1)^2\left[\sum\{1 \le I \le N\}E(I)\right]^{-2}\frac{\pi-2}{M}.$$

This reveals the obvious—that SSB/SSW has larger expectation if $E(1)$ is larger relative to the other eigenvalues.

In the important case where the null distribution is spherical normal (all eigenvalues equal), the asymptotic distribution of SSB and SSW is not joint normal and the complicated calculations will not be given here.

4.9 THINGS TO DO

4.9.1 Running K-Means

A good trial data set is expectations of life by country, age, and sex, in Table 4.10. Try running the K-means algorithm on these data. An initial decision is the question of rescaling the variables. The variances and covariances of the variables should be examined. In a problem like this, where the variables are on the same scale (here, years), no change should be made except for a compelling reason.

The number of clusters K should not be decided in advance, but the algorithm should be run with several different values of K. In this problem, try $K = 1, 2, \ldots, 6$. Analysis of variance, on each of the variables, for each clustering will help decide which number of clusters is best. It is also desirable to compute covariances within each cluster, the overall covariance matrix within clusters, and the overall covariance

Table 4.10 Expectations of Life by Country, Age, and Sex

Keyfitz, N., and Flieger, W. (1971). *Population*, Freeman.

COUNTRY (YEAR)	MALE				FEMALE			
AGE	0	25	50	75	0	25	50	75
1. ALGERIA 65	63	51	30	13	67	54	34	15
2. CAMEROON 64	34	29	13	5	38	32	17	6
3. MADAGASCAR 66	38	30	17	7	38	34	20	7
4. MAURITIUS 66	59	42	20	6	64	46	25	8
5. REUNION 63	56	38	18	7	62	46	25	10
6. SEYCHELLES 60	62	44	24	7	69	50	28	14
7. SOUTH AFRICA (COL) 61	50	39	20	7	55	43	23	8
8. SOUTH AFRICA (WH) 61	65	44	22	7	72	50	27	9
9. TUNISIA 60	56	46	24	11	63	54	33	19
10. CANADA 66	69	47	24	8	75	53	29	10
11. COSTA RICA 66	65	48	26	9	68	50	27	10
12. DOMINICAN REP. 66	64	50	28	11	66	51	29	11
13. EL SALVADOR 61	56	44	25	10	61	48	27	12
14. GREENLAND 60	60	44	22	6	65	45	25	9
15. GRENADA 61	61	45	22	8	65	49	27	10
16. GUATEMALA 64	49	40	22	9	51	41	23	8
17. HONDURAS 66	59	42	22	6	61	43	22	7
18. JAMAICA 63	63	44	23	8	67	48	26	9
19. MEXICO 66	59	44	24	8	63	46	25	8
20. NICARAGUA 65	65	48	28	14	68	51	29	13
21. PANAMA 66	65	48	26	9	67	49	27	10
22. TRINIDAD 62	64	43	21	7	68	47	25	9
23. TRINIDAD 67	64	43	21	6	68	47	24	8
24. UNITED STATES 66	67	45	23	8	74	51	28	10
25. UNITED STATES (NON-W) 66	61	40	21	10	67	46	25	11
26. UNITED STATES (W) 66	68	46	23	8	75	52	29	10
27. UNITED STATES 67	67	45	23	8	74	51	28	10
28. ARGENTINA 64	65	46	24	9	71	51	28	10
29. CHILE 67	59	43	23	10	66	49	27	12
30. COLUMBIA 65	58	44	24	9	62	47	25	10
31. ECUADOR 65	57	46	25	9	60	49	28	11

matrix between clusters. If the overall within covariance matrix differs significantly from a multiple of the unit matrix, transformation of the data to make it proportional to a unit matrix is suggested.

4.9.2 Varieties of the K-Means Algorithm

There are a number of versions of the K-means algorithm that need to be compared by sampling experiments or asymptotic analysis. The changeable components are (i) the starting clusters, (ii) the movement rule, and (iii) the updating rule. The criteria for evaluation are (i) the expected time of calculation and (ii) the expected difference between the local optimum and the global optimum. It is often required that an algorithm produce clusters that are independent of the input order of the cases; this requirement is not necessarily satisfied by the K-means algorithm but can always be met by some initial reordering of the cases. For example, reorder all cases by the first, second, . . . , Nth variables, in succession. (Note that this reordering will usually reduce the number of iterations in the algorithm.)

The following are some starting options:

(i) Choose the initial clusters at random. The algorithm is repeated several times from different random starting clusters with the hope that the spread of the local optima will give a hint about the likely value of the true global optimum. To justify this procedure, it is necessary to have a distribution theory, finite or asymptotic, connecting the local and global optima.

(ii) Choose a single variable, divide it into K intervals of equal length, and let each cluster consist of the cases in a single interval. The single variable might be the average of all variables or the weighted combination of variables that maximizes variance, the first row eigenvector.

(iii) Let the starting clusters for K be the final clusters for $K - 1$, with that case furthest from its cluster mean split off to form a new cluster.

The following are some movement options:

(i) Run through the cases in order, assigning each case according to the cluster mean it is closest to.

(ii) Find the case whose reassignment most decreases the within-cluster sum of squares and reassign it.

(iii) For each cluster, begin with zero cases and assign every case to the cluster, at each step finding the case whose assignment to the cluster most decreases (or least increases) the within-cluster sum of squares. Then take the cluster to consist of those cases at the step where the criterion is a minimum. This procedure makes it possible to move from a partition which is locally optimal under the movement of single cases.

The following are some updating options:

(i) Recompute the cluster means after no further reassignment of cases decreases the criterion.

(ii) Recompute the cluster means after each reassignment.

4.9.3 Bounds on the Global Optimum

It would be good to have empirical or analytic results connecting the local optima and the global optimum. How far is the local optimum likely to be from the global optimum? How different are the two partitions?

For some data configurations, the local optimum is more likely global than for others. Some bad things happen. Let $\{A(I,J), 1 \leq I \leq M, 1 \leq J \leq N\}$ be divisible into K clusters such that the euclidean distance between any pair of cases inside the same clusters is less than ρ and the euclidean distance between any pair of cases in different clusters is greater than ρ. Then this partition is a local optimum but not necessarily global.

Yet, if the clusters are widely separated, it should be possible to prove that there is only one local optimum. Let there be K clusters and fix the distances inside each of the clusters, but let the distances between cluster means all approach infinity. Then eventually there is a single local optimum.

The interesting problem is to make the relation precise between the within-cluster distances and the between-cluster distances, so that there is a unique local optimum. For example, suppose there are K clusters, $M(I)$ points in the Ith cluster, $D(I)$ is the maximum distance within the Ith cluster, and $E(I, J)$ is the minimum distance between the Ith and Jth clusters. For what values of $M(I)$, $D(I)$, and $E(I, J)$ is there a unique local optimum at this partition?

Some asymptotic results suggest that for large sample sizes there will be only a few local optima, differing only a little from the global optimum. The assumption required is that the points are drawn from some parent distribution which itself has a unique local optimum. If these results are expected, it means that a crude algorithm arriving quickly at some local optimum will be most efficient. It would be useful to check the asymptotics in small samples by empirical sampling.

4.9.4 Other Criteria

The points in cluster J may come from a population with parameters, possibly vector valued, $\theta(J)$, φ. The log likelihood of the whole data set is then

$$-\sum \{1 \leq I \leq M\} F\{\mathbf{A}(I), \theta[L(I)], \varphi\},$$

where $\mathbf{A}(I)$ denotes the Ith case, $L(I)$ denotes the cluster to which it belongs, and φ denotes a general parameter applying across clusters. A generalized K-means algorithm is obtained by first assigning $\mathbf{A}(I)$ to minimize the criteria and then changing the parameter values $\theta[L(I)]$ and φ to minimize the criteria.

A first generalization is to allow an arbitrary within-cluster covariance matrix. The algorithm will first assign each case according to its distances from cluster means relative to the covariance matrix. It will then recompute the covariance matrix according to the redefined clusters. Both steps increase the log likelihood. The final clusters are invariant under arbitrary linear transformations of the variables, provided the initial clusters are invariant.

4.9.5* Asympototics

Consider the simplest case of division of real observations into two clusters. (The following results generalize to arbitrary numbers of dimensions and clusters and to more general optimization criteria.) The cluster means are θ and φ, and a typical point is x. Define the 2-vector

$$W(x, \theta, \varphi) = \begin{pmatrix} \theta - x \\ 0 \end{pmatrix}, \qquad \text{if} \quad |\theta - x| \leq |\varphi - x|$$

$$= \begin{pmatrix} 0 \\ \varphi - x \end{pmatrix}, \qquad \text{if} \quad |\varphi - x| < |\theta - x|.$$

For data $X(1), \ldots, X(M)$, the criterion

$$\sum \{X(I) \in C(1)\}[X(I) - \theta]^2 + \sum \{X(I) \in C(2)\}\,[X(I) - \varphi]^2$$

has a local minimum if no reallocation of an $X(I)$ between clusters $C(1)$ and $C(2)$ reduces it and if no change of θ or φ reduces it. The criterion has a local minimum if and only if $\sum \{1 \leq I \leq M\}\, W[X(I), \theta, \varphi] = 0$.

Asymptotic distributions for θ and φ follow from asymptotic distributions for $\sum W$, which for each fixed θ, φ is a sum of identically distributed independent random variables. Let $X(I)$ be sampled from a population with three finite moments. Let

$$E(W) = \begin{bmatrix} \displaystyle\int_{X \leq 1/2(\theta+\varphi)} (\theta - X)\, dP \\ \displaystyle\int_{X > 1/2(\theta+\varphi)} (\varphi - X)\, dp \end{bmatrix} \qquad \text{for} \qquad \theta < \varphi,$$

$$V(W) = \begin{bmatrix} \displaystyle\int_{X \leq 1/2(\theta+\varphi)} (\theta - X)^2\, dP & 0 \\ 0 & \displaystyle\int_{X > 1/2(\theta+\varphi)} (\varphi - X)^2 dP \end{bmatrix} - E(W)E(W').$$

Suppose $E(W) = 0$ for a unique θ, φ, $\theta < \varphi$—say, θ_0, φ_0. Suppose the population has a density $f > 0$ at $x_0 = \frac{1}{2}(\theta_0 + \varphi_0)$, and that $V(W)$ is evaluated at θ_0, ϕ_0.

Let $\hat{\theta}_n$ and $\hat{\varphi}_n$ denote solutions to the equation $\sum W[X(I), \theta, \varphi] = 0$. Asymptotically,

$$\sqrt{nE}[W(X, \hat{\theta}_n, \hat{\varphi}_n)] \approx N[0, V(W)].$$

This means

$$\sqrt{n}\begin{bmatrix} \dfrac{\partial EW}{\partial \theta_0} & \dfrac{\partial EW}{\partial \varphi_0} \end{bmatrix}\begin{bmatrix} \hat{\theta}_n - \theta_0 \\ \hat{\varphi}_n - \varphi_0 \end{bmatrix} \approx N[0, V(W)]$$

and

$$\sqrt{n}\begin{bmatrix} \hat{\theta}_n - \theta_0 \\ \hat{\varphi}_n - \varphi_0 \end{bmatrix} \approx N[0, U],$$

where

$$U = \boldsymbol{\Sigma}^{-1} V \boldsymbol{\Sigma}^{-1},$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} P(X \leq x_0) + \delta & \delta \\ \delta & P(X > x_0) + \delta \end{bmatrix},$$

and

$$\delta = \tfrac{1}{4}(\theta_0 - \varphi_0) f(x_0).$$

It turns out that different locally optimal solutions $\hat{\theta}_n$ and $\hat{\varphi}_n$ differ from one another by terms of $O(n^{-1})$.

For symmetric parent distributions,

$$\sqrt{n}\,(\hat{\varphi}_n - \hat{\theta}_n) \approx N(-\sqrt{n}2\theta_0, V),$$

$$\theta_0 = 2\int_{X<0} X\, dP,$$

and

$$V = 8\int_{X<0} (X - \theta_0)^2\, dP.$$

For a unit normal variable

$$\sqrt{n}(\hat{\varphi}_n - \hat{\theta}_n) \approx N[2\sqrt{2n/\pi}, 4(1 - 2/\pi)].$$

In general, the cluster centers are asymptotically normal with the covariance matrix computed in a similar way to the above. Each cluster center is the average of a number of observations lying closest to it. Its covariance matrix is just the covariance matrix of a mean with a δ term added due to the boundary of the region varying.

Since usually the quantities $\partial EW/\partial\theta$ and $V(W)$ are not known, they must be estimated from the data—$V(W)$ from the observed quantities $W[X(I), \theta, \varphi]$ at $\theta = \hat{\theta}_n$ and $\varphi = \hat{\varphi}_n$, and the derivatives from

$$\sum \{W[X(I), \theta, \varphi] - W[X(I), \hat{\theta}_n, \hat{\varphi}_n]\}$$

for θ, φ near $\hat{\theta}_n$, $\hat{\varphi}_n$.

It would be useful to check the asymptotics by empirical sampling, at least for the normal distribution above. It would be useful to check that the different locally optimal solutions vary by $O(n^{-1})$. It would be useful to check the formulas of Section 4.8 empirically.

4.9.6 Subsampling

To avoid thought and asymptotic formulas, distributions for the cluster means that agree with the asymptotic ones may be obtained empirically as follows. A subset of cases is formed by randomly including each case with probability 0.5. The algorithm produces cluster means. A new subset is formed, and the algorithm produces a new set of cluster means. Repeating this procedure a few times, a sample of cluster means is obtained that agrees asymptotically with a sample from the posterior distribution of true cluster means.

4.9.7 Symmetric Paradox

In the univariate case, find a set of values, symmetric about zero, for which an optimal division into two clusters does not occur at zero.

4.9.8 Large Data Sets

For large numbers of cases (say, $M = 5000$) it is wasteful to do many runs on all cases. No matter what the eventual analysis, there will usually be no great loss in reducing the data to 100 cases using the leader algorithm. Each data point will be within a threshold distance d of one of these 100 cases. Suppose that the leading case is replaced by the average case in each cluster. A K-means algorithm is run on these 100 cases, with each case weighted by the number of original cases in the corresponding cluster. This produces a partition of the average cases with a weighted within-cluster sum of squares w. Show that the within-cluster sum of squares of the corresponding partition of the original cases lies between w and $w + Md^2$.

REFERENCES

COX, D. R. (1957). "Note on grouping." *J. Am. Stat. Assoc.* **52**, 543–547. It is desired to classify an individual into one of K groups using an observation X. Define a grouping function $\xi(X)$ that takes at most K different values. Measure the loss associated with the grouping by $E[X - \xi(X)]^2$. Then $\xi_i P[\xi(X) = \xi_i] = \int_{\xi(X)=\xi_i} X\,dP$. This equation

determines the value of $\xi(X)$ in any group, but it does not specify the groups. The groups are specified, without proof, for $K = 1, 2, \ldots, 6$ from a normal distribution.

DALENIUS, TORE (1951). "The problem of optimum stratification." *Skandinavisk Aktuarietidskrift* **34**, 133–148. In dividing a continuous population into K clusters to minimize within-cluster variance, the cut point between neighboring clusters is the average of the means in the clusters. An iterative method is proposed for attaining this condition from an initial trial division.

ENGELMAN, L., and HARTIGAN, J. A. (1969). "Percentage points of a test for clusters." *J. Am. Stat. Assoc.* **64**, 1647–1648. This paper gives empirical distributions of SSB/SSW for sample sizes 3–10, 15, 20, 25, 50. From examination of the empirical distributions, it is suggested that

$$\log\left(1 + \frac{\text{SSB}}{\text{SSW}}\right) \approx N\left[-\log\left(1 - \frac{2}{\pi}\right) + \frac{2.4}{M-2}, \frac{1}{M-2}\right].$$

This formula agrees approximately with the asymptotic theory of Section 4.8.

FISHER, L., and VAN NESS, J. W. (1971). "Admissible clustering procedures." *Biometrika* **58**, 91–104. The authors compare a number of "joining" algorithms and the K-means algorithm, which they call "hill climbing least squares." Their technique is to test whether or not the procedures produce clusters satisfying certain admissibility conditions for every possible set of data.

The K-means algorithm shows up rather badly failing every test but one—the final clusters are convex. That is, if a case is a weighted average of cases in a cluster, the case also lies in the cluster. The tests it fails include the following:

(i) *Perfect data.* If there exists a clustering such that all distances within clusters are less than all distances between clusters, K means might not discover this clustering.

(ii) *Duplication of cases.* If a case is repeated exactly, the final clustering might change.

They also note that it is practicable for very large numbers of observations, when a joining algorithm requires too much storage and computer time.

MACQUEEN, J. (1967). "Some methods for classification and analysis of multivariate observations." *5th Berkeley Symposium on Mathematics, Statistics, and Probability*. Vol. 1, pp. 281–298. The K-means procedure starts with K groups, each of which contains a single random point. A sequence of points is sampled from some distribution, and each point is added in turn to the group whose mean it is closest to. After a point is added, the group mean is adjusted.

Suppose the population has density $p(z)$ and that the cluster means, after n points are sampled, are $x_1, \ldots, x_K$.

Set

$$W(x_1, \ldots, x_K) = \sum_{i=1}^{K} \int_{S_i} |z - x_i|^2 \, p(z)\, dz,$$

where

$$S_i = \left\{ z \,\middle|\, |z - x_i| = \min_j |z - x_j| \right\}.$$

Then the principal theorem is that $W(x_1, \ldots, x_k)$ converges to $W(u_1, \ldots, u_k)$, where $u_i = (\int_{S_i} z\, p(z)\, dz)/\int_{S_i} p(z)\, dz$. In words, the population variance within the sample clusters converges to the population variance within a locally optimal clustering of the population.

SEBESTYEN, GEORGE S. (1962). *Decision Making Processes in Pattern Recognition*, Macmillan, New York. "Pattern detection is the process of learning the characterization of a class of inputs by detecting the common pattern of attributes of inputs of the same class. Pattern recognition is a process of decision making in which a new input is recognized as a member of a given class by a comparison of its attributes with the already known pattern of common attributes of members of that class." A principal part of the book is concerned with the second problem, discrimination between known classes. A *K*-means-type of algorithm is considered on p. 47, although it is described only in general terms. A number of inputs are introduced in sequence. Each input is assigned to one of a number of classes, according to its distance to the mean input of each class. The input is left unassigned if it is not close enough to any mean input. The mean inputs are updated after each assignment.

PROGRAMS

There are a series of routines that construct partitions of various sizes and print summary statistics about the clusters obtained.

BUILD	calls programs constructing optimal partition of given size and increases partition size by splitting one of the clusters.
KMEANS	assigns each case optimally.
SINGLE	computes summary statistics.
OUTPUT	prints information about clusters.

The following programs perform the basic operations of *K* means.

RELOC	moves cluster center to cluster means.
ASSIGN	reassigns each case to closest cluster center.

```
      SUBROUTINE BUILD(A,M,N,K,SUM,XMISS,NCLUS,DCLUS,X,ITER)
C.....................................................................20 MAY 1973
C     BUILDS K CLUSTERS BY K-MEANS METHOD.  A CLUSTER IS ADDED AT EACH STEP, THE
C     WORST OBJECT FROM THE PREVIOUS STEP.
C     FOR VERY LARGE MATRICES, THIS PROGRAM MUST BE MODIFIED TO AVOID STORING
C     THE COMPLETE DATA MATRIX A IN CORE.   THERE WILL BE K PASSES THROUGH THE
C     DATA MATRIX FOR K CLUSTERS.  THE DIMENSION STATEMENT MUST BE MODIFIED, AND
C     STATEMENT NUMBERED 12.
C.... A = M BY N BORDERED ARRAY
C.... M = NUMBER OF ROWS
C.... N = NUMBER OF COLUMNS
C.... K = NUMBER OF CLUSTERS
C.... XMISS = MISSING VALUE
C.... NCLUS = M BY 1 ARRAY SPECIFYING A CLUSTER NUMBER FOR EACH CASE
C.... DCLUS = 1 BY M ARRAY SPECIFYING DISTANCE OF EACH ROW TO CLOSEST CLUSTER.
C.... X = N BY 1 SCRATCH VECTOR
C.... ITER = NUMBER OF ITERATIONS AT EACH PARTITION SIZE
C.....................................................................................
      DIMENSION SUM(8,N,K),A(M,N),X(N),NCLUS(M),DCLUS(M)
      DO 20 I=1,8
      DO 20 J=2,N
      DO 20 KK=1,K
   20 SUM(I,J,KK)=0
      KL=K-1
      DO 10 KK=1,KL
      DO 14 NC=1,ITER
      DMAX=0.
      ERR=0.
      DO 13 KKK=1,KK
      DO 13 J=2,N
      IF(NC.EQ.1.OR.SUM(1,J,KKK).NE.SUM(3,J,KKK)) ERR=1.
   13 CONTINUE
      IF(ERR.EQ.0) GO TO 15
      DO 16 KKK=1,KK
      DO 16 J=2,N
      SUM(8,J,KKK)=SUM(2,J,KKK)
      IF(NC.EQ.1) SUM(8,J,KKK)=1.
      SUM(2,J,KKK)=0
   16 SUM(1,J,KKK)=SUM(3,J,KKK)
      DO 11 I=2,M
      DO 12 J=2,N
   12 X(J)=A(I,J)
      NCLUS(I)=NC
      CALL KMEANS(N,KK,SUM,X,NCLUS(I),DCLUS(I),XMISS)
   11 CONTINUE
   14 CONTINUE
   15 CONTINUE
      CALL OUTPUT(M,N,KK,SUM,A,NCLUS,DCLUS)
C.... CREATE A NEW CLUSTER BY SPLITTING VARIBLE WITH LARGE WITHIN VARIANCE
      SM=0
      DO 30 J=2,N
      DO 30 KKK=1,KK
      IF(SUM(4,J,KKK).LT.SM) GO TO 30
      SM=SUM(4,J,KKK)
      JM=J
      KM=KKK
   30 CONTINUE
      KN=KK+1
      DO 31 JJ=2,N
      SUM(2,JJ,KN)=0
      SUM(2,JJ,KM)=0
      SUM(3,JJ,KM)=0
   31 SUM(3,JJ,KN)=0
      DO 32 I=2,M
      IF(NCLUS(I).NE.KM) GO TO 32
      DO 33 JJ=2,N
      IF(A(I,JJ).EQ.XMISS) GO TO 33
      IF(A(I,JJ).LT.SUM(1,JJ,KM)) GO TO 34
      SUM(2,JJ,KN)=SUM(2,JJ,KN)+1
      SUM(3,JJ,KN)=SUM(3,JJ,KN)+A(I,JJ)
      GO TO 33
   34 SUM(2,JJ,KM)=SUM(2,JJ,KM)+1
      SUM(3,JJ,KM)=SUM(3,JJ,KM)+A(I,JJ)
   33 CONTINUE
   32 CONTINUE
      DO 35 JJ=2,N
      IF(SUM(2,JJ,KN).NE.0) SUM(3,JJ,KN)=SUM(3,JJ,KN)/SUM(2,JJ,KN)
      IF(SUM(2,JJ,KM).NE.0) SUM(3,JJ,KM)=SUM(3,JJ,KM)/SUM(2,JJ,KM)
   35 CONTINUE
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE KMEANS(N,K,SUM,X,JMIN,DMIN,XMISS)
C.....20 MAY 1973
C     ASSIGNS THE VECTOR X TO THAT CLUSTER WHOSE CLUSTER CENTRE IT IS CLOSEST TO
C     UPDATES FOR THIS CLUSTER, VARIOUS SUMMARY STATISTICS SUCH AS MEAN,SD,MIN,
C     MAX,SSQ.   NOTE THAT CLUSTER CENTERS ARE NOT CHANGED BY THE ADDITION OF X.
C.... N = LENGTH OF VECTOR X
C.... K = TOTAL NUMBER OF CLUSTERS
C.... SUM = 7 BY N BY K ARRAY, CHANGED DURING SUBROUTINE
C       SUM(1,J,I) = VALUE OF JTH VARIABLE AT CLUSTER CENTER
C       SUM(2,J,I) = NUMBER OF NON MISSING OBSERVATIONS,JTH VARIABLE, ITH CLUSTE
C       SUM(3,J,I) = AVERAGE, JTH VARIABLE, ITH CLUSTER
C       SUM(4,J,I) = STANDARD DEVIATION
C       SUM(5,J,I) = MINIMUM
C       SUM(6,J,I)= MAXIMUM
C       SUM(7,J,I) = SUM OF SQUARED DEVIATIONS FROM CLUSTER MEAN
C.... X = N BY 1 VECTOR TO BE ALLOCATED AMONG THE K CLUSTERS
C.... JMIN = NUMBER OF CLUSTER WHOSE CENTRE X IS CLOSEST TO.
C.... DMIN = EUCLIDEAN DISTANCE BETWEEN X AND CENTER OF JMIN CLUSTER
C.... XMISS = MISSING VALUE
C.....
      DIMENSION SUM(8,N,K),X(N)
      JMIN=1
      DMIN=10.**20
      DO 20 J=1,K
      XP=10.**(-10)
      DD=0
      DO 21 I=2,N
      IF (X(I).EQ.XMISS) GO TO 21
      DD=DD+(X(I)-SUM(1,I,J))**2
      XP=XP+1.
   21 CONTINUE
      DD=(DD/XP)**0.5
      IF(DD.GT.DMIN) GO TO 20
      DMIN=DD
      JMIN=J
   20 CONTINUE
      XM=N
      DO 31 I=2,N
      IF(X(I).EQ.XMISS) GO TO 31
   30 CALL SINGLE(X(I),SUM(2,I,JMIN),SUM(3,I,JMIN),SUM(4,I,JMIN)
     1,SUM(5,I,JMIN),SUM(6,I,JMIN),SUM(7,I,JMIN))
   31 CONTINUE
      RETURN
      END




      SUBROUTINE SINGLE(X,COUNT,AVE,SD,XMIN,XMAX,SSQ)
C.....20 MAY 1973
C     INCORPORATES NEW VALUE X INTO SUMMARY STATISTICS
C     THE MEANING OF EACH VARIABLE IS GIVEN IN KMEANS.
C.....
      IF(COUNT.NE.0.)GO TO 10
      AVE=0
      SD=0
      XMIN=10.**20
      XMAX=-10.**20
      SSQ=0
   10 COUNT=COUNT+1.
      AVE=AVE+(X-AVE)/COUNT
      IF(COUNT.NE.1) SSQ=SSQ+COUNT*(X-AVE)**2/(COUNT-1.)
      SD=(SSQ/COUNT)**0.5
      IF(XMIN.GT.X) XMIN=X
      IF(XMAX.LT.X) XMAX=X
      RETURN
      END
```

```
      SUBROUTINE OUTPUT(M,N,KK,SUM,A,NCLUS,DCLUS)
C.....20 MAY 1973
C     OUTPUT ROUTINE FOR KMEANS ALGORITHM
C     PUTS OUT SUMMARY STATISTICS FOR EACH VARIABLE FOR EACH CLUSTER
C     ALSO PUTS OUT OVERALL ANALYSIS OF VARIANCE FOR EACH VARIABLE
C.... A = M BY N BORDERED DATA MATRIX
C.... M = NUMBER OF CASES
C.... N = NUMBER OF VARIABLES
C.... KK = NUMBER OF CLUSTERS
C.... SUM = 7 BY N BY KK MATRIX OF SUMMARY STATISTICS(SEE KMEANS ROUTINE)
C.... NCLUS = M BY 1 ARRAY IDENTIFYING CLUSTER TO WHICH EACH ROW BELONGS
C.... DCLUS = EUCLIDEAN DISTANCE OF EACH ROW TO CLOSEST CLUSTER
C.....
      DIMENSION SUM(8,N,KK),NCLUS(M),DCLUS(M),A(M,N)
      DIMENSION AA(10),DD(10)
      DIMENSION R(50)
      DATA NPAGE/0/
      DATA LC/0/
C.... MEAN SQUARE CALCULATION OVER ALL CLUSTERS
      NPAGE=NPAGE+1
      WRITE(6,7) NPAGE
    7 FORMAT(1H1,110X,I5)
      WRITE(6,9) KK
    9 FORMAT(' OVERALL MEAN SQUARE CALCULATIONS, FOR EACH VARIABLE, ',
     1' WITH',I5,'  CLUSTERS')
      ASSW=0
      DO 40 J=2,N
      SD=0.
      SC=0.
      SSB=0.
      SSW=0.
      DO 41 K=1,KK
      SD=SD+SUM(3,J,K)*SUM(2,J,K)
      SSB=SSB+SUM(3,J,K)**2*SUM(2,J,K)
      SSW=SSW+SUM(7,J,K)
   41 SC=SC+SUM(2,J,K)
      DFB=KK-1
      DFW=SC-DFB-1.
      TH=10.**(-10)
      IF(SC.EQ.0) SC=TH
      IF(DFW.EQ.0) DFW=TH
      IF(DFB.EQ.0) DFB=TH
      ASSW=ASSW+SSW
      SSB=SSB-SD**2/SC
      SSB=SSB/DFB
      SSW=SSW/DFW
      IF(SSW.EQ.0) SSW=TH
      RATIO=0
      IF(LC.NE.0)RATIO=(R(J)/SSW-1)*(1+DFW)+1
      R(J)=SSW
      WRITE(6,8)A(1,J),SSW,DFW,SSB,DFB,RATIO
    8 FORMAT(' VARIABLE',A8,F20.6,'(WITHIN MEAN SQ.)',F4.0,'(WITHIN DF)'
     1,F20.6,'(BETWEEN MSQ)',F4.0,'(BETWEEN DF)',F6.1,'(FRATIO)')
   40 CONTINUE
      WRITE(6,10) ASSW
   10 FORMAT(' OVERALL WITHIN SUM OF SQUARES',F20.6)
      LC=LC+1
      DO 20 K=1,KK
      WRITE(6,11)
   11 FORMAT(1X,131(1H-))
      WRITE(6,1) K,KK
    1 FORMAT(I5,'   TH CLUSTER OF',I5)
      WRITE(6,2)(I,I=1,10)
    2 FORMAT('0CLUSTER MEMBERS WITH THEIR DISTANCES TO THE CLUSTER CENTRE'/(13X,
     1E'/13X,10I11)
      L=0
      DO 21 I=2,M
      IF(NCLUS(I).NE.K) GO TO 22
      L=L+1
      AA(L)=A(I,1)
      DD(L)=DCLUS(I)
   22 IF (L.LT.10.AND.I.LT.M) GO TO 21
      IF(L.EQ.0) GO TO 21
      WRITE(6,3)(AA(LL),LL=1,L)
    3 FORMAT(15X,10(7X,A4))
      WRITE(6,12)(DD(LL),LL=1,L)
   12 FORMAT(15X,10F11.4)
      L=0
```

```
   21 CONTINUE
      WRITE(6,4)
    4 FORMAT('0SUMMARY STATISTICS FOR THE CLUSTER')
      WRITE(6,5)
    5 FORMAT(' LABEL',5X,'CENTRE',8X,
     I           'COUNT',12X,'AVE',13X,'SD',11X,'XMIN',11X,'XMAX',12X,'SSQ''
     1SQ')
      DO 30 J=2,N
   30 WRITE(6,6)A(1,J),(SUM(I,J,K),I=1,7)
    6 FORMAT(1X,A4,7F15.6)
   20 CONTINUE
   81 CONTINUE
      RETURN
      END



      SUBROUTINE RELOC(M,N,K,A,X,NC)
C....................................................................20 MAY 1973
C.... RELOCATES EACH CLUSTER CENTRE TO BE A CLUSTER MEAN
C.... M = NUMBER OF ROWS
C.... N = NUMBER OF COLUMNS
C.... K = NUMBER OF CLUSTERS
C.... A = M BY N BORDERED ARRAY
C.... X = N BY K BORDERED ARRAY OF CLUSTER CENTRES
C.... NC = M BY 1 ARRAY ASSIGNING EACH ROW TO A CLUSTER
C.....................................................................
      DIMENSION A(M,N),X(N,K),NC(M)
      DIMENSION CC(10)
      DATA CC/4HCLUS,2HC1,2HC2,2HC3,2HC4,2HC5,2HC6,2HC7,2HC8,2HC9/
      XM=99999.
C.... COMPUTE MEANS
      DO 10 L=2,K
      DO 10 J=2,N
      X(J,L)=0
      IF(NC(I).NE.L) GO TO 20
      IF(A(I,J).EQ.XM) GO TO 20
      P=0
      DO 20 I=2,M
      P=P+1
      X(J,L)=X(J,L)+A(I,J)
   20 CONTINUE
      IF(P.NE.0) X(J,L)=X(J,L)/P
      IF(P.EQ.0) X(J,L)=XM
   10 CONTINUE
C.... LABEL CLUSTER MEANS
      DO 40 J=2,N
   40 X(J,1)=A(1,J)
      DO 50 L=1,K
      IF(L.GT.10) RETURN
   50 X(1,L)=CC(L)
      RETURN
      END
```

```
      SUBROUTINE ASSIGN(M,N,K,A,X,NC)
C.....................................................................20 MAY 1973
C.... ASSIGNS EACH ROW OF BORDERED ARRAY TO CLOSEST OF CLUSTER CENTRES X
C.... M = NUMBER OF ROWS
C.... N = NUMBER OF COLUMNS
C.... K = NUMBER OF CLUSTERS
C.... A = M BY N BORDERED ARRAY
C.... X = BORDERED ARRAY OF CLUSTER CENTRES
C.... X = N BY K BORDERED ARRAY OF CLUSTER CENTRES
C.... NC = M BY 1 ARRAY ASSIGNING EACH ROW TO A CLUSTER
C          NC(1) = ROW FURTHEST FROM ITS CLUSTER CENTRE
C.....................................................................
      DIMENSION A(M,N),X(N,K),NC(M)
C.... INITIALISE
      O=0
      NC(1)=0
      XM=99999.
      DO 10 I=2,M
      DC=10.**10
      DO 20 J=2,K
      OJ=0
      OP=0
C.... FIND DISTANCE TO CLUSTER CENTRE
      DO 30 L=2,N
      IF(X(L,J).EQ.XM.OR.A(I,L).EQ.XM) GO TO 20
      OJ=OJ+(X(L,J)-A(I,L))**2
   30 CONTINUE
      IF(OP.GT.O) OJ=(OJ/OP)**0.5
C.... FIND CLOSEST CLUSTER CENTRE
      WRITE(6,1) I,J,NC(I),OP,DC,OJ
    1 FORMAT(3I5,3F20.6)
      IF(OJ.GT.DC) GO TO 20
      DC=OJ
      NC(I)=J
   20 CONTINUE
C.... FIND ROW FURTHEST FROM ITS CENTRE
      IF(DC.LT.O) GO TO 10
      O=DC
      NC(1)=I
   10 CONTINUE
      RETURN
      END
```