CLUSTER ANALYSIS IN PATTERN RECOGNITION

by

Paul William Warren

A thesis submitted for the degree of
Master of Philosophy in the Faculty of Science,
University of Southampton

Department of Electronics
University of Southampton

February 1975

ABSTRACT

FACULTY OF SCIENCE

ELECTRONICS

Master of Philosophy

CLUSTER ANALYSIS IN PATTERN RECOGNITION

by Paul William Warren

A comparison is made of most of the major types of clustering algorithm. Particular attention is paid to their applicability to the pattern recognition problem. Because of the size of the data set typically encountered in pattern recognition, the comparison includes a detailed study of the computational efficiency of the various techniques. A number of these techniques were used to try to cluster a data set composed of time domain descriptors describing an electroencephalographic waveform taken from a sleeping human. None of the techniques experimented with succeeded in revealing any significant cluster structure for this particular data set.

ACKNOWLEDGEMENTS

The author acknowledges the help given to him by his former colleagues in the Electronics Department at Southampton University. In particular the author is grateful to Dr D. W. Thomas, who read and commented on the manuscript; to Mr G. Smith, who is chiefly responsible for what little understanding the author has of the E.E.G.; and to Mr D. Hand, with whom the author has enjoyed many stimulating discussions on the theoretical nature of the problem. In addition, the author would like to record his appreciation for the staff of the Computer Advisory Service at Southampton University, without whose help the computational part of this work would have been impossible. Finally, thanks are due to his wife for typing the bulk of this thesis.

CONTENTS

GLOSSARY OF LESS COMMON MATHEMATICAL SYMBOLS AND NOTATION USED IN THIS THESIS

$E(x)$ — The expectation of the random variable x.

$E(\underline{x})$ — The expectation of the random vector $\underline{x}$.

$\ln(x)$ — The natural logarithm of x, i.e. $\log_e(x)$.

$O(N^k)$ — This is used to describe a function of N, f(N). '$f(N) = O(N^k)$' is a shorthand notation for '$\frac{f(N)}{N^k}$ tends to a constant value as N tends to infinity'.

W^T — The transpose of the matrix W.

tr W — The trace of the (square) matrix W. For an n x n matrix this is defined by:

$$\text{tr } W = \sum_{i=1}^{i=n} Wii$$

$|W|$ — The determinant of the matrix W.

$a \ll b$ — This denotes that the number a is very much smaller than the number b.

$A \subseteq B$ — This denotes that the set A is contained in the set B. It includes the case where the two sets are identical.

$\prod_{i=1}^{i=n} a_i$ — The product of all the terms $a_1, a_2, \ldots a_n$.

CHAPTER 1 INTRODUCTION

1.1 The Aim of Cluster Analysis

Cluster analysis studies the problem of how to divide a set of objects into a number of subsets such that all the members of one subset are similar and yet differ significantly from the members of the other subsets. Normally there is no a priori knowledge of the number of subsets. There are two basic reasons for wishing to cluster a set of objects. On the one hand one might wish to discern 'a true typology'. That is, one may wish to know whether the objects under study can be regarded as examples of a relatively small number of different kinds of objects. Alternatively, one may be concerned with data reduction. The set may contain too many objects to handle. If the set can be divided into a manageable number of subsets, one object can be taken from each subset. The result is a sample more representative of the original objects than a random sample would be.

Many different disciplines make use of cluster analysis. Consequently, the subject has been developed by workers in a variety of different fields. As a result, some techniques have been independently discovered several times over. The first development of the subject was in botany and zoology, where it is referred to as taxonomy. Here the aim is the finding of a true typology. The objective is to divide the animal and plant kingdom into genera, species, etc. The first problem here is how to measure the similarity (or dissimilarity) between objects. Consequently, much of the taxonomic literature is less about clustering techniques than about measures of dissimilarity (called dissimilarity coefficients). A good introduction to taxonomy is given by Sokal and Sneath (1963). More recently, the social sciences have made use of cluster analysis. Here again the objective is that of finding a true typology and the problem of measurement of dissimilarity is of great

importance. The problem is that the observations are largely qualitative and they must be made quantitative before the cluster analysis can proceed.

In pattern recognition the situation is rather different. The descriptors are typically quantitative in nature and consequently it is natural to regard the objects under study as points in a high-dimensional space. The dissimilarity between two objects can then be defined as the distance in the vector space between the points representing them. Normally the distance used is some special case of the Minkowski distance defined by

$$d_{ij} = \left(\sum_{k=1}^{q} |X_{ik} - X_{jk}|^r \right)^{\frac{1}{r}}$$

Here, d_{ij} is the distance in a q-dimensional space between the i'th and j'th points. X_{ik} represents the value of the k'th variable for the i'th point. When r=1, this is called the city block metric. When r=2, it is the Euclidean distance.

Furthermore, in pattern recognition the data sets frequently contain hundreds or even thousands of objects. Although this is sometimes the case in taxonomy, notably in microbiology, it is much less likely to be so. Consequently, in pattern recognition one needs to be much more concerned with the computational efficiency of an algorithm than in taxonomy.

There is a further important difference between the situation in pattern recognition and that in taxonomy. Consider the classification of animals. A typical descriptor might be the answer to the question, 'does the animal have a tail?'. Within any particular species (cluster) of animal the answer will be the same. In pattern recognition, because one normally deals with quantitative measurements, the same descriptor will frequently not possess exactly the same value for two co-classed

objects. Rather, the measurement will be characterised by a probability distribution. Thus, in taxonomy one has a deterministic situation whilst in pattern recognition one has a statistical situation.

In taxonomy, the aim is usually to construct a hierarchy. At the lowest level, say that of species, only one sample is needed from each object. The goal is to construct a dendrogram showing how the species are grouped to form genera and the genera are grouped to form families. This is illustrated for five hypothetical species (A, B, C, D, E) in Figure 1.1. A, B and C are members of one genus, D and E are members of another. All five species are members of the same family.

In pattern recognition, a typical problem is that of analysing a large number of noisy signals. One may believe that each signal can be expressed as 'pure signal plus noise', where the number of different 'pure signals' is quite few in number. Then cluster analysis would be used to attempt to determine the number and nature of these pure signals.

In view of what has been said it is hardly surprising that the subject has recently attracted the attention of statisticians. Rather, it is surprising that this did not happen earlier. This is probably partly because classical statistics depends upon making assumptions about the probability distribution of the population from which the data set comes. In cluster analysis this is often not possible. Perhaps it is also because the multi-dimensional aspect of the problem necessitates a considerable use of the digital computer, the full potential of which was probably apparent earlier to engineers than to statisticians.

1.2 Comparison with the Discrimination Problem in Pattern Recognition

It may be helpful to compare cluster analysis with the classical pattern recognition problem, supervised learning or 'learning with a teacher'. Here, one is given a set of objects which come from a known

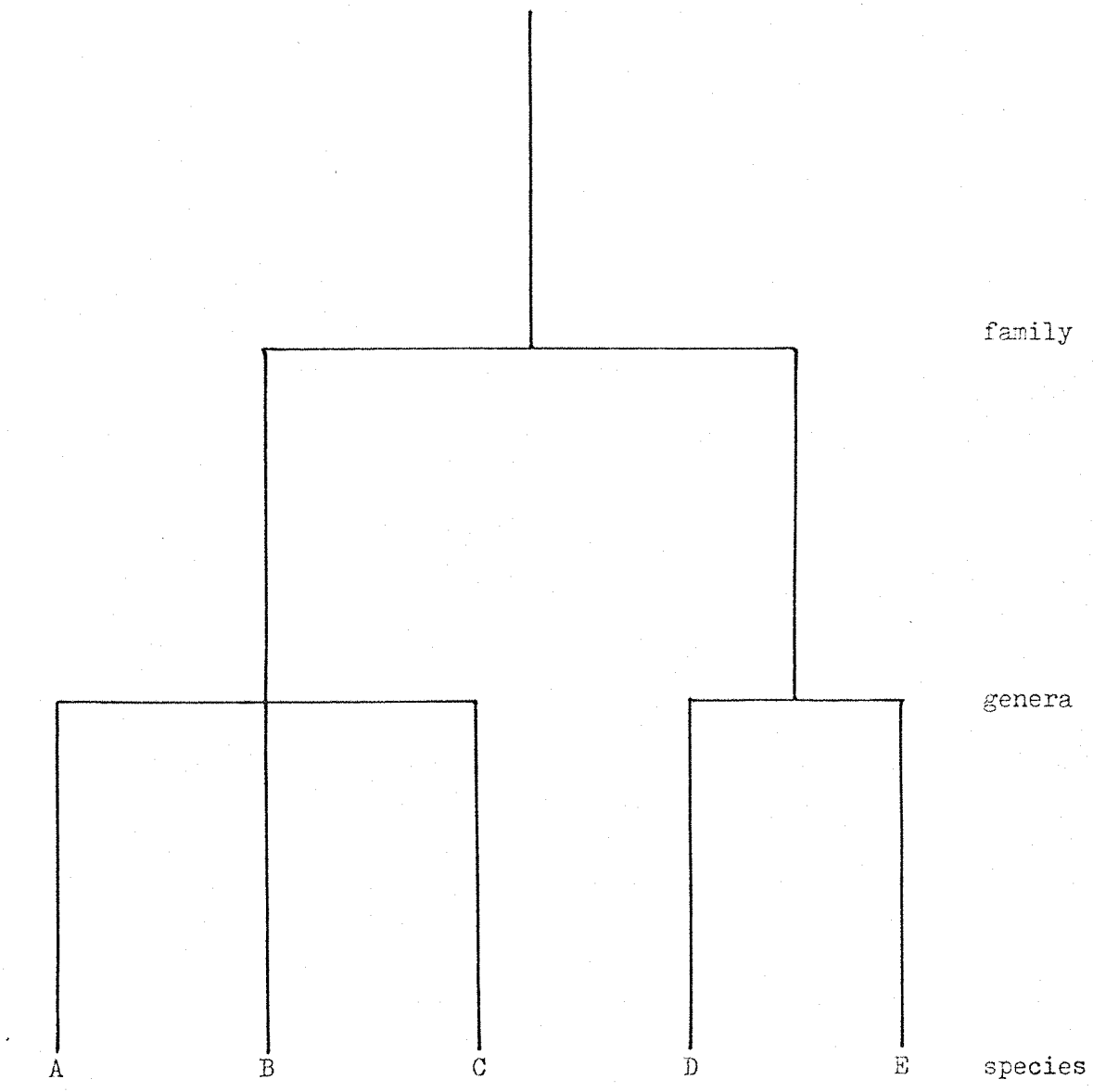


FIGURE 1.1

number of classes. In addition, the correct classification of each object is known. The problem is to determine the classification of a future (unclassified) point. This has been much studied, both from a statistical standpoint and from a 'geometrical' one. In statistical supervised learning, assumptions are made about the probability distributions of the various classes. In geometrical pattern recognition, no such assumptions are made. Instead error-correcting algorithms (such as the perceptron algorithm) are used. All that is required is that it is possible to separate the classes with a limited number of hyperplanes, hyperspheres, etc.

The term 'unsupervised learning' is also used in the pattern recognition literature. This can be a synonym for cluster analysis, but it seems to be more often restricted to the particular case where the number of clusters is known and some assumptions about the underlying probability distribution are made. For a review of both supervised and unsupervised learning see Nagy (1968).

1.3 Plan of the Thesis

For the purposes of this thesis the techniques of cluster analysis have been divided into five categories. Although this classification is in places rather arbitrary, it is believed that it does reflect real distinctions between the techniques. Each type of approach is discussed in one of Chapters 2 to 6.

Chapter 2 deals with linkage techniques. Here, the set of objects is divided into a partition on the basis of the dissimilarity matrix. This is the matrix whose i,j'th element is the dissimilarity coefficient between the i'th and j'th objects. Linkage techniques solely depend upon the dissimilarity matrix. Hence, once this matrix has been calculated the actual descriptor values can be discarded.

Chapter 3 deals with a number of techniques which have been termed 'optimization-partitioning' techniques (Everitt, 1974, Chapter 2). Here the object is to maximize (or minimize) some function determined by the data set and the partition under consideration.

Chapter 4 considers techniques whose origins stem from the concept of a probability density function.

Chapter 5 deals with techniques which attempt to map a multi-dimensional data set into two dimensions. The data set can then be displayed visually and use made of the human observer's ability to perform 'Gestalt' clustering.

Chapter 6 discusses a number of miscellaneous approaches. Some of these are 'one-off' techniques, bearing little relation to any others. Some of them represent general categories which the author feels are of little significance in pattern recognition.

In comparing the techniques a number of questions will be asked. Four in particular apply to all the techniques.

'Does the procedure produce a hierarchy of objects?'. As has been noted taxonomic clustering is usually of this form.

'Are the results invariant under non-singular linear transformations of the data space?'. Frequently, invariance will apply under orthogonal transformations (i.e. pure rotations), but rarely does it apply under general non-singular linear transformations (i.e. when the scale of each axis is altered). When the full invariance property does not apply it is usual to first normalize each dimension to zero mean and unit variance in order that no one variable unduly influences the analysis.

'Does the technique depend upon some a priori assumptions about the probability distribution of the population of which the objects represent a sample?'. Techniques which depend upon such assumptions are termed parametric. As will be seen later, totally misleading results may be

obtained when incorrect assumptions are made.

'Can the technique be applied to the large data sets found in pattern recognition?'. A data set may contain thousands of objects. As a result, the memory and time requirements of each algorithm have to be considered more carefully than is usual in, say, taxonomy.

Chapter 7 describes a typical pattern recognition data set which is of significance in the understanding of sleep.

Chapter 8 discusses the results of applying some cluster analysis techniques to the data set of Chapter 7.

Finally, Chapter 9 is concerned with the future of cluster analysis in pattern recognition. In particular, some questions are posed. It is hoped that the solutions to these will lead to further developments in the subject.

CHAPTER 2 LINKAGE TECHNIQUES

2.1 Introduction

All the techniques described in this chapter can be used to produce a hierarchical structure, or dendrogram. As already mentioned they all work from a similarity (or dissimilarity) matrix. In theory, one could first calculate this matrix and then discard the original variables. In practice, this is rarely done in pattern recognition. A dissimilarity matrix on N objects will require $\frac{1}{2}N(N-1)$ words of storage. The C.D.C. 7600 used in the work described in Chapter 8 has approximately 100K words of available core storage. Consequently if a clustering program stores the dissimilarity matrix it is limited to data sets of about 500 objects. Most machines are smaller than this. As a result, a dissimilarity coefficient is normally calculated when needed.

2.2 Nearest Neighbour or Single Link Cluster Analysis

This is probably the oldest clustering technique. In essence objects are co-classed whose dissimilarity coefficient is less than or equal to some threshold value (h). To illustrate this definition, consider one possible implementation. Start with one cluster, containing the first object. Then consider the dissimilarity coefficient between first and second objects. If it is less than or equal to h, put the second object also in the first cluster. Otherwise create a new cluster containing the second object. As each new object is introduced, consider all the objects with which its dissimilarity coefficient is less than or equal to h. If these objects occur in a number of different clusters, coalesce these clusters and put the new object in the resultant cluster. If the objects all occur in the same cluster, put the new object in this cluster. If no such objects occur, then create a new cluster containing

the new object.

A principal objection to this method is that it gives rise to a property called chaining. This means that the method tends to cluster together objects linked by chains of intermediates. This is illustrated in Figure 2.1. Here the presence of points A, B and C will cause single link cluster analysis to give a one cluster solution when otherwise it would suggest three clusters. Chaining may or may not be a defect in deterministic cluster analysis. In pattern recognition it certainly is so. For in the statistical situation, there is always the probability of 'noise points' occurring between clusters. Thus the results of the analysis can be critically dependent on the particular sample used. This is a serious disadvantage.

Another problem is the determination of what value of h to use. In taxonomy, this is generally overcome by producing a dendrogram structure. Such a structure was illustrated diagramatically on page 4. From a dendrogram, one can observe the cluster structure at all levels (i.e. for all values of h). Mathematically a dendrogram can be defined (Sibson, 1973) as a function c which maps from the range of values of h (i.e. 0 to ∞) into the set of all equivalence relations on the data set. Further, c must satisfy three conditions.

(1) $h \leq h'$ implies $c(h) \subseteq c(h')$

(2) When h is very large all objects become equivalent.

(3) $c(h+\delta)=c(h)$ for all small enough $\delta > 0$

If there are N objects in the data set the dendrogram can contain up to (N-1) different splitting levels.

The problem of how to construct such a dendrogram in the most efficient manner has been solved (Sibson, 1973). Sibson describes an algorithm capable of clustering 1000 objects on the Cambridge University

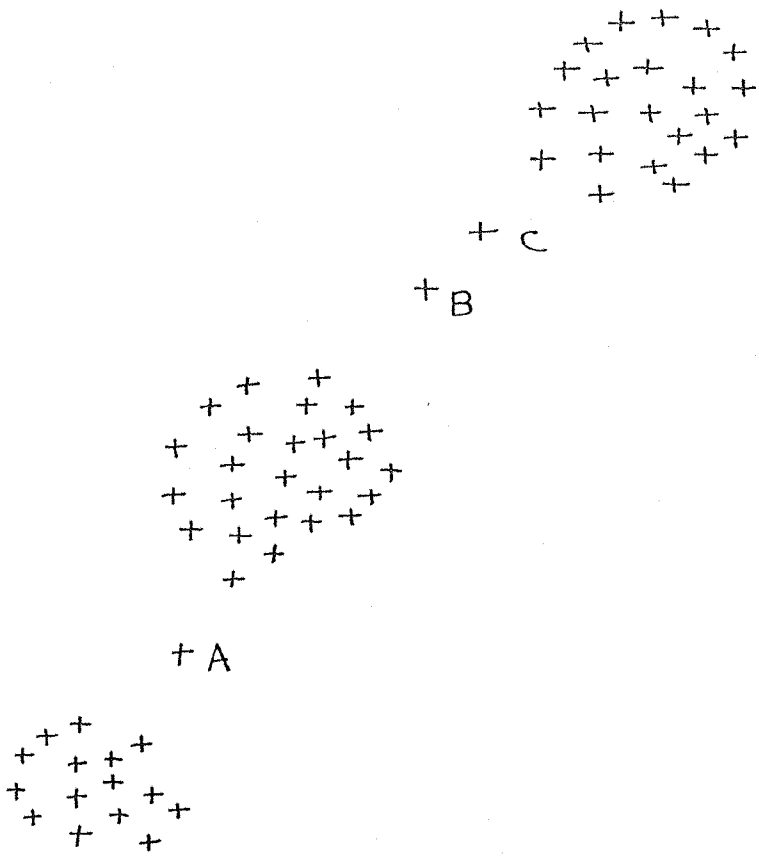


FIGURE 2.1

Computer Laboratory TITAN in 100 seconds, excluding the reading or generation of the dissimilarity coefficients. The time dependence of the algorithm is $O(N^2)$ and it requires storage of $O(N)$. The low storage is achieved by using the dissimilarity coefficients a part-row at a time. This avoids the $O(N^2)$ storage requirement necessary to store all the dissimilarity coefficients. Sibson suggests generating the part-rows of dissimilarity coefficients when necessary. Alternatively they could be stored on disc in the order in which they are required, i.e. 2-1; 3-1, 3-2; 4-1, 4-2, 4-3; 5-1 etc. However this can result in a program which spends almost all its time reading dissimilarity coefficient values.

It is still necessary to display the dendrogram and make some sense of it. For data sets of many hundreds, or even thousands of objects, this will not be an easy matter. Remembering that in pattern recognition one is not normally interested in obtaining a hierarchical structure, it can be seen that the dendrogram is not the ideal form for the results to take. This problem will be returned to later in the chapter.

2.3 Furthest Neighbour or Complete Linkage Cluster Analysis

This differs from the previous technique in that for an object to join a particular cluster its dissimilarity coefficient with each object in that cluster must be less than or equal to some value, h. This completely overcomes the chaining problem and produces compact clusters. While this eliminates the problem of spurious results due to noise points it has the disadvantage that an elongated cluster will appear as a number of clusters.

In practice the technique has the same disadvantages as single linkage cluster analysis. Either one has some criterion for choosing h, or one must output a dendrogram.

2.4 Clustering Using a Similarity Measure Based on Shared Near Neighbours

A technique has recently been suggested which makes use of a radically new definition of similarity coefficient (Jarvis and Patrick, 1973). The algorithm first establishes a nearest neighbour table. For each object a list is made of its k nearest neighbours, in order of proximity. Nearest is here defined with respect to some conventional dissimilarity measure (e.g. Euclidean distance). A number of possible definitions can then be used to calculate a new similarity matrix from these lists. The simplest approach is to make the i,j'th element of the matrix equal the number of members common to the nearest neighbour lists for the i'th and j'th objects. This similarity matrix can then be input to any linkage technique. In particular, Jarvis and Patrick use the single link approach with one modification. They suggest that an object X should join a cluster if there exists in that cluster an object Y such that two sets of conditions are satisfied. Firstly, the number of shared elements in the lists belonging to X and Y must be greater than or equal to some value, k_T. This is the single link criterion when applied to the similarity matrix defined above. Secondly, X must be in Y's list and vice-versa. As in single link analysis, if there are points Y and Y' in two distinct clusters but both satisfying the above conditions, then the clusters are coalesced and X joins the cluster so formed.

The philosophy underlying this algorithm can be understood by regarding the objects as points in hyperspace. Then two points close together should only be co-clustered if they come from the same continuous region of high point density. This will mean they share a large number of near neighbours. This is illustrated in Figure 2.2(a) and (b). In (a) there are two close points (X and Y) which share many near neighbours and should obviously be co-clustered. In (b) X and Y, although

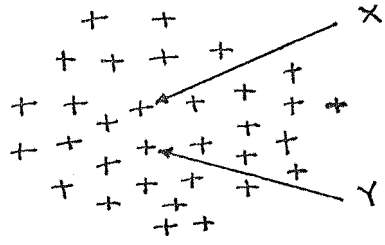


FIGURE 2.2(a)

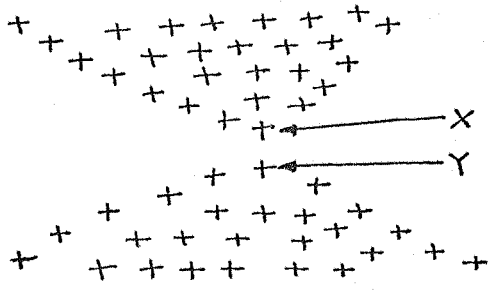


FIGURE 2.2(b)

quite close, come from different clusters (i.e. regions of high point density). These points do not possess so many near neighbours. Thus the algorithm is an attempt to overcome the defects of the chaining effect.

Not only can one vary k_T and repeat the analysis, one can also vary k. Having established the nearest neighbour lists for some value of k, k_0, one can use these lists for any value of k less than k_0 merely by forgetting about their final elements. The time to establish the lists will be proportional to $N^2(q+C(k))$, where N is the number of objects and q is the number of descriptors. C(k) is a relatively small factor to allow for testing for all k near neighbours for each point. The time taken to use the table for cluster analysis will be (neglecting zero and first-order terms in N) proportional to $N^2(k+1)^2$ at most. The actual time-dependence will be significantly dependent on the value of k_T.

The problem of the correct choice of parameter value is greater for this algorithm than for the two previously discussed. Now there are two parameters which may be varied, k and k_T. Some light may be thrown on this problem by considering the philosophy of the algorithm. As explained, the aim is to locate continuous regions of high point density. The point density can be regarded as an estimate of the probability density of the population from which the sample was drawn. Thus the real objective of the algorithm is to divide the space into continuous regions of high probability density. Now Loftsgaarden and Quesenberry (1965) have shown that the k'th nearest neighbour approach can be used to estimate the probability density function (p.d.f.) at a point. Let R be the distance from an arbitrary point $\underline{X}$ in the data space to the k'th nearest data point. Any choice of metric can be used. Then Loftsgaarden and Quesenberry have proved that one consistent estimate of the p.d.f. at $\underline{X}$ is

$$\frac{(k-1)}{NV}$$

Here V is the volume of a hypersphere centred on $\underline{X}$ and of radius R. Thus there is a relationship between this clustering technique and one form of p.d.f. estimation. The exact nature of this relationship is an open question posed by Jarvis and Patrick. Loftsgaarden and Quesenberry have considered the problem of what value of k to take. They suggest that about $\sqrt{N}$ 'appears to give good results'. It would seem reasonable that a value for k of the same order of magnitude should give good results in the clustering algorithm. In practice, the value of k may be limited by the available storage anyway.

Having decided on a suitable value (or values) for k it is still necessary to decide on values for k_T. As with ordinary single-linkage cluster analysis, one possibility is to produce a dendrogram, but again this is probably not suitable for large values of N.

2.5 Use of the Minimal Spanning Tree in Single Linkage Cluster Analysis

The problem of determining reasonable values for h in single linkage cluster analysis has recently been attacked by the use of concepts from graph theory. Before describing this approach it is necessary to define the concept of 'minimal spanning tree' (M.S.T.). Consider a set of N objects and a dissimilarity matrix on these objects. Then a spanning tree is a set of edges joining the objects such that all objects are connected but containing no closed paths. Consequently the spanning tree will contain (N-1) edges. To each edge a weight is assigned. For the purposes of cluster analysis this will be the dissimilarity coefficient between the objects defining the edge. The weight of a tree is then defined to be the sum of the weights of the edges in the tree. Then a minimal spanning tree will be a spanning tree whose weight is minimal amongst all spanning trees. It can be shown that clusters at any level h can be obtained from the M.S.T. by deleting all segments of weight

greater than h (Gower and Ross, 1969). Consequently having once produced an M.S.T. it is a relatively simple matter computationally to vary h and obtain a single linkage analysis that the observer regards as significant. In Chapter 8 an example is given where an M.S.T. was obtained from 2119 points in 8-space in approximately 60 seconds on a C.D.C. 7600 (using Euclidean distance as the measure of dissimilarity). To produce a single linkage cluster analysis at any given level from the M.S.T. took of the order of 10 seconds on an I.C.L. 1907, a machine which is at least 40 times slower than the C.D.C. 7600.

The algorithm used to construct the M.S.T. is due to Prim (1957). The advantage of this algorithm, as Gower and Ross point out, is that it only makes use of each dissimilarity coefficient once. Thus it is not necessary to store the dissimilarity matrix. Rather, each dissimilarity coefficient can be calculated when necessary. As a result, the storage requirements of the algorithm are only $O(N)$ and consequently it can be used on very large data sets. The time dependency of the algorithm is $O(N^2)$.

As previously stated, h can now be varied until a reasonable level of clustering is achieved. A more systematic alternative has recently been suggested (Zahn, 1971). Zahn suggests taking each edge of the M.S.T. in turn and determining the average weight of the neighbouring M.S.T. edges. If the edge under consideration has weight greater than this average edge weight multiplied by some factor, then it is termed inconsistent. The M.S.T. can then be 'disconnected' at all the inconsistent edges. This is not quite the same thing as true single linkage cluster analysis, since it takes into account local variations in average edge-weight. Thus of two identically weighted edges one may be inconsistent and the other not.

Lee (1974) has suggested using an approximate M.S.T. in cluster analysis. His technique is similar to a technique to be discussed in Chapter 5 for non-linear mapping. Lee constructs a 'sub-minimal spanning tree', i.e. a spanning tree which, although not actually an M.S.T., is (hopefully) quite close to being one. First, an M.S.T. (T*) is constructed for M objects taken from the N in the data set. The remaining (N-M) objects are joined to the tree by minimizing the sum of weights of edges between objects in T* and objects not in T*. The weights of edges between objects not in T* are ignored. The total number of dissimilarity coefficients required will be $\frac{1}{2}M(M-1)+M(N-M)$, rather than the $\frac{1}{2}N(N-1)$ needed to form the true M.S.T. Lee believes the final spanning tree will be sufficiently nearly minimal to obtain useful clustering results. Unfortunately difficulties arise if the members of T* are not representative of all the clusters. Lee believes this should be immediately discernible from the tree but in this author's opinion there is a danger that a valid cluster might be missed and regarded as merely a collection of outliers (i.e. isolated objects apparently not members of any significant cluster of objects). However, if care is taken to sample the N objects at random so as to obtain the initial M frame points, this technique may be valuable in the analysis of very large data sets.

2.6 Comments

All the techniques discussed in this chapter are non-parametric in nature. If the dissimilarity coefficient is taken to be Euclidean Distance the results will be invariant under orthogonal rotations of the data space. They will not be invariant under general non-singular linear transformations.

It is interesting to compare the Jarvis and Patrick algorithm with single linkage cluster analysis as performed by constructing an M.S.T.

Both procedures have similar objectives in that they attempt to define continuous regions of space of high point density without making assumptions about the shape of these regions. Both procedures have time requirements given by $O(N^2)$. Assuming that k is approximately $\sqrt{N}$, the storage requirements of the Jarvis and Patrick algorithm will be $O(N^{3/2})$. This compares unfavourably with the M.S.T. approach which has storage requirements $O(N)$. The M.S.T. allows far greater precision in the choice of a clustering level. As will be seen later, it is quite possible, by changing k_T by unity, to go from a situation where only one cluster is apparent to a situation where very many (e.g. hundreds) of small clusters are found. In between one has completely missed the level of clustering which would probably significant to the user of the technique.

CHAPTER 3 OPTIMIZATION-PARTITIONING TECHNIQUES

3.1 Introduction

The techniques discussed in this chapter attempt to find that partition of the data set which optimizes some criterion. In theory, given the number (g) of clusters required, one needs merely to consider all the possible partitions of the N data points and choose that one which optimizes the criterion. However, for any realistic value of N this will be computationally impossible. For example, for $g=2$ there will be $(2^{N-1}-1)$ partitions to be considered. Consequently an exhaustive search is impossible. Instead heuristics are used in an attempt to achieve a good value for the criterion.

All the criteria discussed in this chapter are functions of the scatter matrices defined below. Consequently they do not make use of the dissimilarity matrix but work directly with the data set.

Before discussing each criterion separately it is necessary to introduce some notation. Let the data be represented by a set of N q-dimensional column vectors $\underline{X}_1, \underline{X}_2, \ldots \underline{X}_N$. The object is to divide this data set into g groups $G_1, G_2, \ldots G_g$ with populations $N_1, N_2, \ldots N_g$. The N_i are not, of course, known a priori. The following 'scatter matrices' can then be defined (Fukunaga, 1970).

Total scatter:

$$S \triangleq \sum_{k=1}^{N} \underline{X}_k \underline{X}_k^T$$

Intragroup scatter:

$$W_j \triangleq \sum_{\underline{X}_k \in G_j} (\underline{X}_k - \underline{C}_j)(\underline{X}_k - \underline{C}_j)^T \qquad \text{where} \qquad \underline{C}_j = \frac{1}{N_j} \sum_{\underline{X}_k \in G_j} \underline{X}_k$$

Thus $\underline{C}_j$ is the centroid of the j'th group. W_j is related to the covariance matrix for the j'th group. ~~For a~~ consistent, unbiased estimate of this covariance matrix will be given by

$$\frac{W_j}{N_j-1}$$

Total intragroup scatter:

$$W \triangleq \sum_{j=1}^{g} W_j$$

Intergroup scatter:

$$B \triangleq \sum_{j=1}^{g} N_j \underline{C}_j \underline{C}_j^T$$

The superscript T denotes transposition. It can easily be shown that

$$T=W+B$$

3.2 The 'Error Sum of Squares Criterion'

The first criterion to be suggested was the 'error sum of squares'. The philosophy underlying its first use was one of data reduction (e.g. Thorndike, 1953). Imagine attempting to represent a set of N data points by g points ($g \ll N$) with minimum loss of information. One possible approach is to partition the data set into g groups and then replace each partition by its centroid (i.e. the $\underline{C}_j$). Then a suitable measure of the error will be

$$J_0 = \sum_{j=1}^{g} \sum_{\underline{X}_k \in G_j} d(\underline{X}_k, \underline{C}_j)$$

where $d(\underline{X}_k, \underline{C}_j)$ represents the distance between the points $\underline{X}_k$ and $\underline{C}_j$ in the hyperspace. Using squared Euclidean distance this becomes

$$J_0 = \sum_{j=1}^{g} \sum_{\underline{X}_k \in G_j} (\underline{X}_k - \underline{C}_j)^T (\underline{X}_k - \underline{C}_j)$$

$$= \sum_{i=1}^{g} W_{ii}$$

This is known as the trace of W ($\mathrm{tr}\ W$). Clearly, minimum loss of information will be achieved when J_0 is minimised.

In addition to its use in data reduction, the criterion may be applicable in the 'true typology' problem. If the clusters are spherical and approximately equal in size the use of this criterion will probably separate them. However, if these assumptions do not hold, the technique may give completely misleading results (see Wishart, 1969).

3.3 An Optimization Algorithm

Macqueen (1966) has suggested and analysed an algorithm for obtaining a suitable partition. Similar algorithms have appeared at several other places in the literature (e.g. Sebestyen, 1962, Chapter 4, section 5; Ball, 1965). The version of the algorithm presented here is as described in Fukunaga and Koontz (1970).

Firstly g initial group centres are chosen by some (possibly random) initialisation process. These may or may not correspond to actual data points. Each data point is considered as belonging to that cluster whose centre is nearest, in the Euclidean distance sense. When each data point has been allocated to a cluster the group centres are re-calculated as the centroids of the data points in each group. The algorithm is shown diagrammatically in Figure 3.1. The algorithm can be regarded as composed of two parts. In the first part the group membership is varied for each data point ($\underline{X}_k$) so as to minimize $(\underline{X}_k - \underline{C}_j)^T (\underline{X}_k - \underline{C}_j)$, which is the contribution of that point to J_0. In the second part the $\underline{C}_j$ are re-computed for

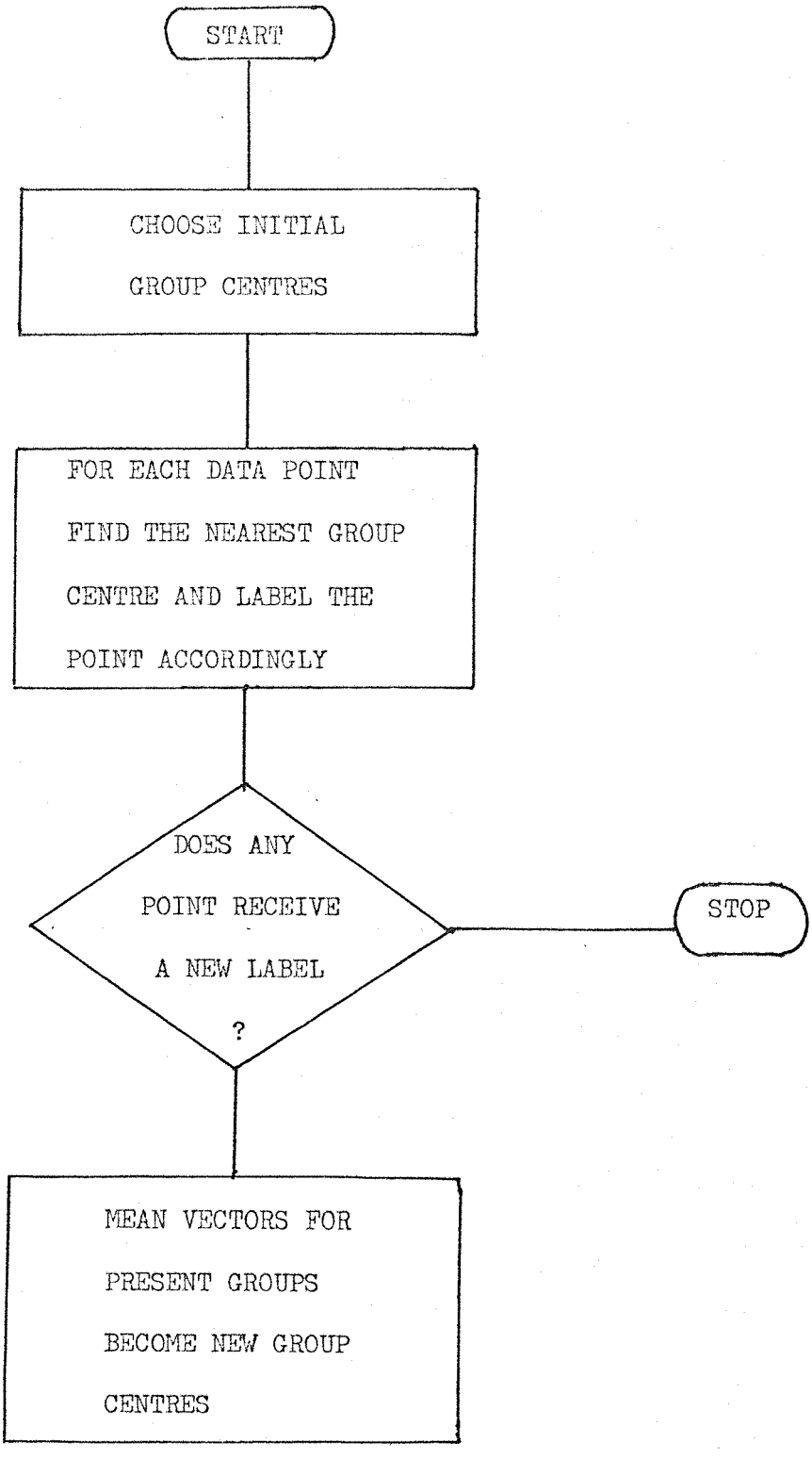

START
CHOOSE INITIAL GROUP CENTRES
FOR EACH DATA POINT FIND THE NEAREST GROUP CENTRE AND LABEL THE POINT ACCORDINGLY
DOES ANY POINT RECEIVE A NEW LABEL ?
STOP
MEAN VECTORS FOR PRESENT GROUPS BECOME NEW GROUP CENTRES

FIGURE 3.1

a fixed partition. As is well-known, the centroid of a set of data points is the point in the hyperspace about which there is a minimum scatter. Thus in each part of the algorithm J_0 will be minimized. As a result the error sum of squares will be reduced until eventually convergence is achieved. This will not necessarily represent the absolute minimum of J_0 but will probably be merely a local minimum.

For data reduction, as long as a low value of J_0 is achieved, it does not matter too much if that value is not close to the true minimum. This is not so when trying to find a true typology. Here convergence to a local minimum far from the true minimum is an unsatisfactory result. Consequently several authors have suggested elaborations of the algorithm (Macqueen 1966; Friedman and Rubin, 1967). Since the suggestions of Friedman and Rubin have been incorporated into a program (McRae, 1971) which has been used in the work described in Chapter 8, their approach is outlined here.

In addition to the iterative technique already mentioned, which they term 'reassignment passes', Friedman and Rubin also describe 'hill-climbing passes' and 'forcing passes'. In a hill-climbing pass every data point in turn is moved from its own group into each other group. If no move decreases J_0, the point is left where it is. Otherwise it is moved so as to achieve the maximum decrease in J_0. When each point has been considered, the program has performed one 'hill-climbing pass'. After several such passes, a stage is reached at which no move of a single point will further decrease J_0. A 'forcing pass' is then begun. Considering one group at a time, each data point in the group is placed into the outside group with the nearest centre of gravity. At each stage the point considered is that one nearest to an outside group. After the first re-assignment J_0 must increase, since otherwise the re-assignment

would have been achieved by the previous hill-climbing pass. Eventually, however, J_0 may decrease again. After processing all the objects of one group, the best partition yet found is restored, and the program passes on to the next group. When each group has been considered, one forcing pass has been completed. Forcing passes are repeated until they produce no improvement. At this stage the re-assignment pass already described is begun. This is also repeated until no further reduction in J_0 is achieved. The three stages are then repeated until convergence is reached. The final partition is then assumed to be the optimal one.

As a check, Friedman and Rubin suggest repeating the procedure with different initial partitions. The computation which leads to the lowest final value of J_0 is the one whose results are used.

It is difficult to analyse the time requirement of this algorithm theoretically because of its complexity. It would appear to be O(N). However, it also seems to be very dependent upon g and upon the structure of the data. In Chapter 8 some times will be quoted for the use of this program with a real data set. The storage requirements of the algorithm certainly are O(N).

3.4 The Invariant Criteria of Friedman and Rubin

The results, obtained by the use of J_0, are invariant only under orthogonal transformations, since J_0 is invariant only under these transformations. In the same paper in which they described the algorithm of section 3.3, Friedman and Rubin reported some experiments on the use of two criteria which are invariant under all non-singular linear transformations. Before describing these criteria it is helpful to prove the following theorem. The proof is taken from Anderson (1958), page 222.

Theorem

The eigenvalues of $W^{-1}B$ are invariant under all non-singular linear

transformations.

Proof

Consider a typical non-singular linear transformation, A. In the following, all transformed vectors and matrices are denoted by the superscript '. Then

$$\underline{X}'_k = A\,\underline{X}_k$$

Consequently

$$\underline{C}'_j \triangleq \frac{1}{N_j}\sum_{\underline{X}'_k \in G_j} \underline{X}'_k = \frac{1}{N_j}\sum_{\underline{X}_k \in G_j} A\,\underline{X}_k$$

$$= A\,\underline{C}_j$$

$$W'_j \triangleq \sum_{\underline{X}'_k \in G_j} (\underline{X}'_k - \underline{C}'_j)(\underline{X}'_k - \underline{C}'_j)^T$$

$$= \sum_{\underline{X}_k \in G_j} A(\underline{X}_k - \underline{C}_j)(\underline{X}_k - \underline{C}_j)^T A^T$$

$$= A\,W_j\,A^T$$

$$W' \triangleq \sum_{j=1}^{g} W'_j = A\Big(\sum_{j=1}^{g} W_j\Big)A^T$$

$$= A\,W\,A^T$$

$$\underline{B}' \triangleq \sum_{j=1}^{g} N_j \underline{C}'_j \underline{C}'^T_j$$

$$= \sum_{j=1}^{g} N_j\, A\,\underline{C}_j\,\underline{C}_j^T\,A^T$$

$$= A\,B\,A^T$$

Now the eigenvalues of $W^{-1}B$ are the solutions of the equation $|B - \lambda W| = 0$.

However

$$|B' - \lambda W'| = |ABA^T - \lambda AWA^T|$$

$$= |A(B - \lambda W)A^T|$$

$$= |A||B - \lambda W||A^T|$$

Thus the solutions of $|B - \lambda W| = 0$ are also the solutions of $|B' - \lambda W'| = 0$. By repeating the argument using the reverse transformation, A^{-1}, the converse is true. Thus the eigenvalues of $W^{-1}B$ are invariant under non-singular linear transformations.

The two invariant criteria used by Friedman and Rubin are defined as

$$J_1 = \frac{|S|}{|W|}$$

$$J_2 = tr(W^{-1}B)$$

The invariance of these two criteria can now most easily be demonstrated by showing that they are both functions of the eigenvalues of $W^{-1}B$, denoted by $\lambda_1, \lambda_2, \ldots \lambda_q$. For

$$J_1 = |W|^{-1}|S|$$

$$= |W^{-1}||S|$$

$$= |W^{-1}S|$$

$$= |W^{-1}(W+B)|$$

$$= |I+W^{-1}B|$$

$$= \prod_{i=1}^{q}(1 + \lambda_i)$$

$$J_2 = \mathrm{tr}(W^{-1} B)$$

$$= \mathrm{tr}(D^T D W^{-1} B)$$

$$= \mathrm{tr}(D W^{-1} B D^T)$$

$$= \prod_{i=1}^{q} \lambda_i$$

Here D is the (orthogonal) transformation which diagonalizes $W^{-1} B$.

Both criteria are to be maximised. However since S is fixed for any given data set maximizing $\frac{|S|}{|W|}$ is equivalent to minimizing $|W|$. For simplicity, the first of these two criteria will be referred to as the minimization of $|W|$.

No easily visualised significance can be attached to J_1. However, it has an importance in terms of statistical theory which will be explained in a later section. J_2 is more easily interpreted. Assume that all the clusters have the same covariance matrix (i.e. the W_j differ only by a multiplicative factor). Then a transformation which transforms W to the identity matrix will produce spherical clusters and will transform the intergroup scatter (B) to $W^{-1} B$. Thus maximizing $\mathrm{tr}(W^{-1} B)$ is equivalent to maximizing the scatter of the centres of a number of spherical clusters.

After a series of experiments on real data of biological significance, Friedman and Rubin were of the opinion that J_1 gave better results than J_2, in addition to being computationally easier.

In adapting the algorithm of section 3.3 to optimize J_1 and J_2, the definition of distance used is the Mahalanobis Euclidean distance. The Mahalanobis distance between two points, $\underline{X}_i$ and $\underline{X}_j$, is defined as

$$(\underline{X}_i - \underline{X}_j)^T W^{-1} (\underline{X}_i - \underline{X}_j)$$

This distance will be invariant under any non-singular linear transformation. It stands in the same relation to J_1 as the Euclidean distance does to J_0. That is to say, to minimize $|W|$ at each step during a 'reassignment pass', one assigns each point $\underline{X}_k$ to the cluster with centre $\underline{C}_j$ such that $(\underline{X}_k-\underline{C}_j)^T W^{-1} (\underline{X}_k-\underline{C}_j)$ is a minimum (Marriott, 1971). Clearly the use of J_1 and J_2 will necessitate considerably longer execution times than the use of J_0.

3.5 Further Invariant Criteria

Two further invariant criteria have been suggested in the literature. Fukunaga and Koontz (1970) first normalise the data space so as to transform the total scatter matrix to become the identity matrix.

$$S \longrightarrow A\ T\ A^T = I$$

As a consequence

$$\underline{C}_j \longrightarrow A\ \underline{C}_j \triangleq \underline{D}_j$$

$$W \longrightarrow A\ W\ A^T \triangleq P$$

$$B \longrightarrow A\ B\ A^T \triangleq Q$$

$$J_0 \longrightarrow \mathrm{tr}\ P \triangleq J_0'$$

The eigenvalues of $P^{-1}Q$ are of course $\lambda_1, \lambda_2, \ldots \lambda_q$, the eigenvalues of $W^{-1}B$. Let $\{\mu_i\}$ represent the eigenvalues of P. Then the eigenvalues of P^{-1} are μ_i^{-1}. The relationship $S = W+B$ transforms to $I = P+Q$. Hence

$$P^{-1} = I+P^{-1}Q$$

$$\Longrightarrow \frac{1}{\mu_i} = 1 + \lambda_i$$

$$\Longrightarrow \mu_i = \frac{1}{1+\lambda_i}$$

Consequently J_0' is also an invariant criterion, since

$$J_0' = \text{tr } P$$

$$= \sum_{i=1}^{q} \mu_i$$

$$= \sum_{i=1}^{q} \frac{1}{1+\lambda_i}$$

By analogy with J_0, J_0' is to be minimized. J_0' is computationally cheaper than J_1 and J_2 since after normalising the data space Euclidean distance is used, rather than the more complex Mahalanobis distance. Thus a considerable computational saving is achieved. When $g = 2$, Fukunaga and Koontz have shown that J_1, J_2 and J_0' are all equivalent criteria.

McRae (1971), in his program MICKA, makes use of (in addition to J_0', J_1 and J_2) a further invariant criterion defined by

$$J_3 = \text{largest eigenvalue of } W^{-1} B$$

This criterion is to be maximized. In the two-cluster case J_3 also becomes equivalent to the other invariant criteria. It is apparent from the experiments in Chapter 8 that the execution time for the optimization of this criterion is greater then for J_0 and J_1 and approximately the same as for J_2. To the author's knowledge, J_3 possesses no advantage over the other invariant criteria.

3.6 The Statistical Significance of the $|W|$ Criterion

Scott and Symons (1971) have shown that the criterion $|W|$ has a special significance when the data consists of a number of independent observations from a mixture of multivariate normal distributions with equal covariance matrices. In this case the maximum likelihood partition

of the data set into g subsets is that partition which minimizes $|W|$. This result seems to be dependent upon the assumption that there is negligible overlap between the distributions. This fact is not explicitly mentioned in the paper. Scott and Symons also point out that this result can be extended to the case of a mixture of normal distributions with unequal covariance matrices. Here the criterion to be minimized is $\sum_{j=1}^{g} |W_j|^{N_j}$. To the author's knowledge no-one has as yet attempted to use this criterion.

3.7 Determination of the Optimum Value for g

So far the problem of determining the optimum value for g has not been considered. One cannot merely optimize the criterion as g varies. To see this consider $J_0(\text{tr } W)$. In the extreme case when $g = N$, each cluster will contain one data point. Hence J_0 will be zero. No other partition will do this, unless some of the N points are identical. Consequently, the optimum partition would always be into single point clusters, which is certainly not the desired result.

Some authors, including Friedman and Rubin (1967), have suggested plotting the optimum value of the criterion against g. It is hoped that a sharp increase or decrease in the criterion will occur at the 'correct' value of g. This procedure has been shown to be unsatisfactory for J_0 (e.g. Thorndike, 1953). Friedman and Rubin report reasonable results by plotting $\log \max(|S| / |W|)$ against g. Marriott (1971) has shown that the optimum subdivision into g groups of a uniformly distributed population reduces $|W|$ by a factor g^2. This led him to suggest finding that value of g which minimizes $g^2 \min |W|$.

In the next chapter, another technique will be discussed which rests upon the assumption that the data consists of a number of independent observations from a mixture of multivariate normal distributions. This technique gives rise to significance tests for g.

CHAPTER 4 TECHNIQUES RELATED TO THE CONCEPT OF P.D.F.

4.1 Wishart's Mode Analysis

Wishart (1969) discusses the failure of an error sum of squares technique (see last chapter) when applied to a particular problem in astronomy. The data is two-dimensional and was first plotted by H. N. Russel in 1914. It shows temperature against luminosity for a large number of stars. According to Russel, and to most other observers, the data divides naturally into two elongated clusters, one considerably longer than the other. The members of the longer cluster have come to be known as 'dwarf stars'; the members of the other cluster are referred to as 'giants'. All astronomers seem agreed that this is the 'correct' classification for their purposes. However, when an error sum of squares technique is applied to this data set quite different results are obtained. The final classification into two groupings divides the dwarfs into two clusters and places most of the giants into one of these clusters. In view of what was said in the last chapter, it is not surprising that such unsatisfactory results should be obtained from an error sum of squares technique.

Wishart then goes on to give a review of some thirteen different cluster analysis techniques and points out that they all share the 'minimum variance' property. That is, they all attempt to minimize the within-group sum of squares. Wishart seems unaware of the work of Friedman and Rubin in extending the optimization-partitioning approach. Of course he was writing before the publications of Marriott and Scott and Symons. Consequently he concludes that this approach is unsatisfactory for many real problems.

He then discusses single-linkage cluster analysis and points out the

problem presented by noise points. This leads him to suggest a modification of single link analysis in which the clustering is performed only on those data points at which the estimated p.d.f. is above a certain level. The algorithm requires selecting a distance threshold r, and frequency (or density) threshold k. Any definition of distance could be used, but Wishart uses Euclidean distance. The number of points, k_i, within a distance r of each data point is then calculated. The points for which $k_i < k$ are regarded as 'noise points' and discarded. Single-linkage cluster analysis is then performed on the remaining points. Finally each noise point is allocated to the cluster containing its nearest dense point.

The avoid the problem of having to choose two parameter values, Wishart suggested a second algorithm, called 'hierarchical mode analysis'. This algorithm requires only a density threshold, k. The distances from each point to its k'th nearest neighbour are computed and then ordered, with the smallest first. The points are considered in order of increasing k'th nearest neighbour distance. As each new point is introduced, a parameter PMIN is set to the value of that point's k'th nearest neighbour distance. The algorithm then tests to determine which of the following three possibilities holds.

(1) The new point does not lie within PMIN of another dense point (i.e. a point already considered), in which case it initialises a new cluster mode.

(2) The point lies within PMIN of dense points from one cluster only, in which case it joins that cluster.

(3) The point lies within PMIN of dense points from several clusters, in which case they are coalesced and the point joins the newly-formed cluster.

Finally, because the value of PMIN has been changed, the algorithm checks whether the nearest-neighbour distance of any two clusters is now

less than PMIN, in which case these clusters are coalesced.

As the algorithm proceeds the number of clusters will vary. Wishart suggests outputting information about the clusters immediately before any are coalesced. The maximum number of clusters can be taken as indicating the most significant level of clustering. Having found this level, each noise point can be attached to the cluster containing its nearest dense point.

The algorithm has an execution time $O(N^2)$ and storage requirements $O(N)$. Wishart claims that it performs well and is relatively insensitive to the choice of k. It is a pity, however, that he does not tell us how the algorithm performs on the data set that originally inspired it, i.e. the astronomical data of Russel.

Clearly the co-ordinates of the modes transform in the same way as the co-ordinates of the data points. Consequently any mode-seeking technique must be invariant under non-singular linear transformations. For a finite data set this is not absolutely true. To a certain extent any mode-seeking technique must be approximate and the accuracy of the result may depend on the scales chosen for each variable. However the results of the subsequent single-linkage cluster analysis are completely dependent on the particular definition of distance used. If Euclidean distance is used the results will only be invariant under orthogonal transformations. The same is true for the allocation of the noise points. As a consequence the resultant partition is invariant only under orthogonal transformations.

4.2 Gitman and Levine's Mode-Seeking Technique

Gitman and Levine (1970) describe a very similar mode-seeking technique. Their paper is presented in the language of fuzzy sets developed by Zadeh (1965). The approach is more mathematical than

Wishart's and they are able to show that the technique possesses some nice properties in the limit as the size of the data set tends to infinity. Unfortunately it is not clear to the author just how many points are necessary for any given data structure to obtain reasonable results.

Given a large enough data set, the technique will detect all the modes of point density. Clusters will then be formed around each mode. The algorithm derives the optimal partition in the sense of maximum separation as adopted by Zadeh. This means essentially that the cluster boundaries will lie in the valleys, i.e. regions of low point density.

The storage requirement is (20N + CN + S) words, where C is a constant whose size will depend upon the number of modes present in the data set, and S is the number of words required to store the data. Gitman and Levine comment that 'the amount of computing time is relatively small'. However they do not quote the exact relationship between execution time and N. Because of the complexity of the algorithm the author is unable to determine the nature of this relationship. Probably the execution time will also be very dependent upon the number of modes present.

Gitman and Levine also suggest two modifications to the slgorithm to accommodate very large data sets (e.g. greater than 30,000 samples). Neither of these 'short-cuts' had been tested when they wrote their paper.

Because it achieves maximal separation in the sense of Zadeh, the algorithm is (in the limiting case of an infinite data set) fully invariant under non-singular linear transformations.

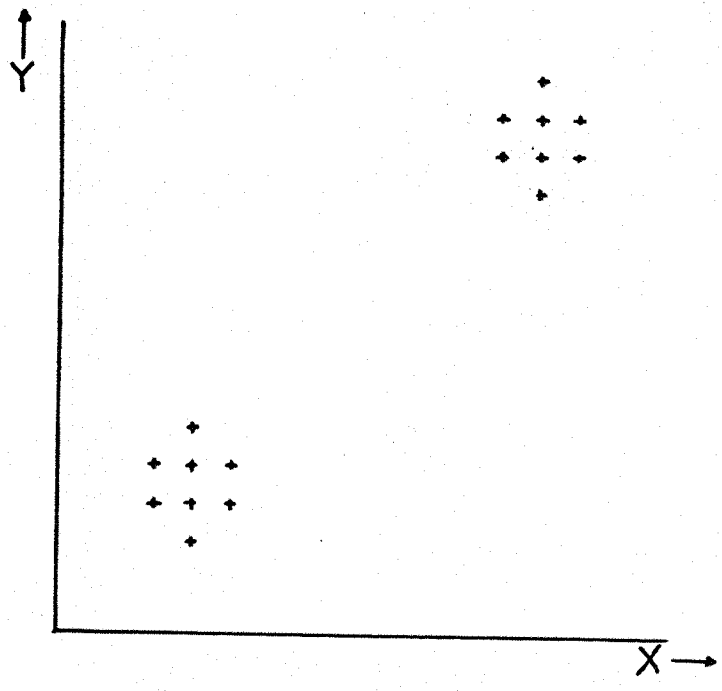
4.3 NSPACE

NSPACE is a mode-seeking technique proposed by Eigen, Fromm and Northouse (1974). For each dimension a histogram is constructed to

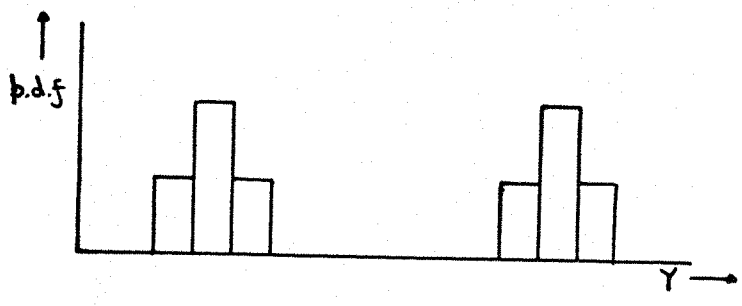
represent the marginal p.d.f. An algorithm is then used to find the modes of each histogram. The results of this algorithm are dependent upon two parameters, δ and Θ. δ is the number of intervals into which the range of each dimension is divided and Θ is a threshold parameter used to define a mode. Assume that the i'th dimension contains $\overline{Q}_i$ modes, situated at M_{i1}, M_{i2}, ... $M_{i\overline{Q}_i}$. If $\overline{Q}_j = 0$ the j'th dimension is ignored. The marginal p.d.f. in this dimension will be approximately uniform over the range considered. Consequently this dimension has no value in cluster analysis. In all, there will be $\prod_{i=1}^{q} \overline{Q}_i$ 'potential' modes. Each potential mode will have its co-ordinate in the i'th dimension equal to one of the M_{ik} $(k = 1, 2, ... \overline{Q}_i)$. In reality, there will probably be a smaller number of actual modes, as can be seen from the two-dimensional example in Figure 4.1. Here $\prod_{i=1}^{q} \overline{Q}_i$ equals 4, but in fact there are only two modes. Now each data point $\underline{X}$ $(=(X_1, X_2, ... X_q))$ is considered in turn and for each dimension the nearest mode (M_i') is found. Nearest here means 'so as to minimize $|X_i - M_{ij}|$'. The point is allocated to the cluster centred around the mode at $(M_1', M_2', ... M_q')$. By the end of the algorithm, some of the $\prod_{i=1}^{q} \overline{Q}_i$ potential modes will have points allocated to them, others may not. Thus the result is a partition of the N data points into g classes where $g \leq \min(\prod_{i=1}^{q} \overline{Q}_i, N)$.

Eigen et al. regard their technique as the first part of a 'global-local scheme'. That is, they regard its results as an approximation to the clustering structure which can improved on by more complex, and hence slower, techniques.

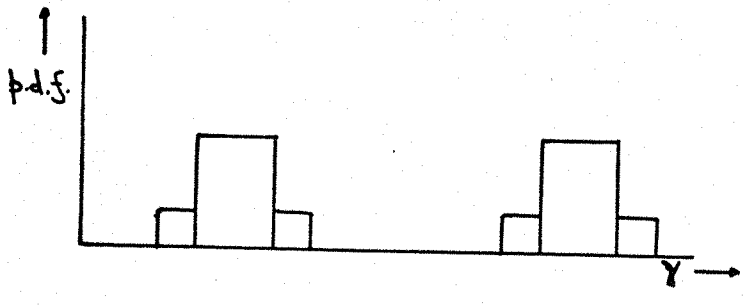
The paper describing NSPACE is made unnecessarily obscure by an excess of mathematical symbolism. The author is unclear about the rationale behind the details of the particular mode-seeking algorithm used. From the example given the modes appear to be positioned in the valleys, i.e. in the regions of low p.d.f.! There is in addition an



(a) A two-dimensional data set



(b) The marginal p.d.f. in the X dimension



(c) The marginal p.d.f. in the Y dimension.

FIGURE 4.1

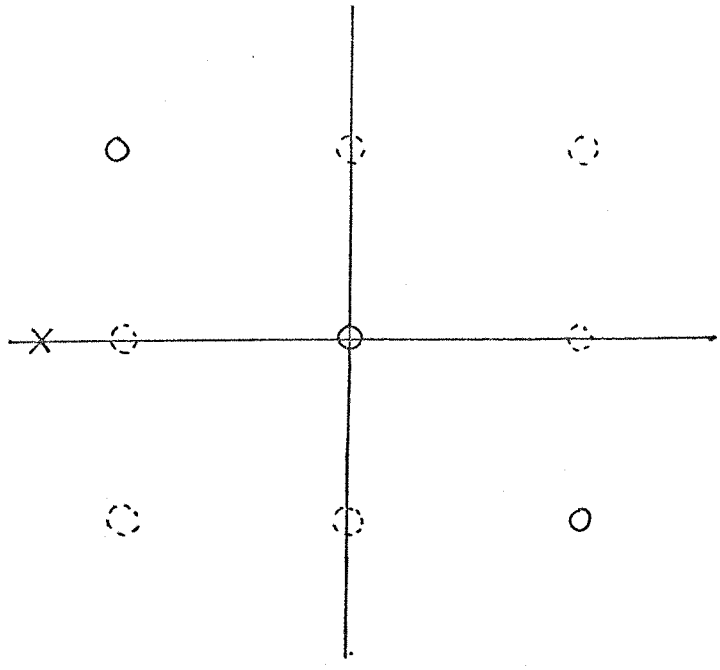
erroneous comment concerning the algorithm of Sebestyen and Edie, which will be described in section 4.5. Eigen et al. state that at certain stages in the Sebestyen and Edie procedure some of the cells may split into two. As will be seen later, this is not so.

NSPACE appears to be both fast and economical of storage. Both its time and storage requirements are O(N). However it is sensitive to the choice of control parameters (i.e. δ and Θ). Furthermore NSPACE is probably the least invariant of all the techniques discussed in this chapter. Even the number of potential modes may vary as the axes are rotated. An example of this is shown in Figure 4.2. Here there are three actual modes. In Figure 4.2(a) there are 9 potential modes, whilst in Figure 4.2(b) there are just 3, corresponding to the actual modes. It can also be seen that the point X will be allocated quite differently in both situations.

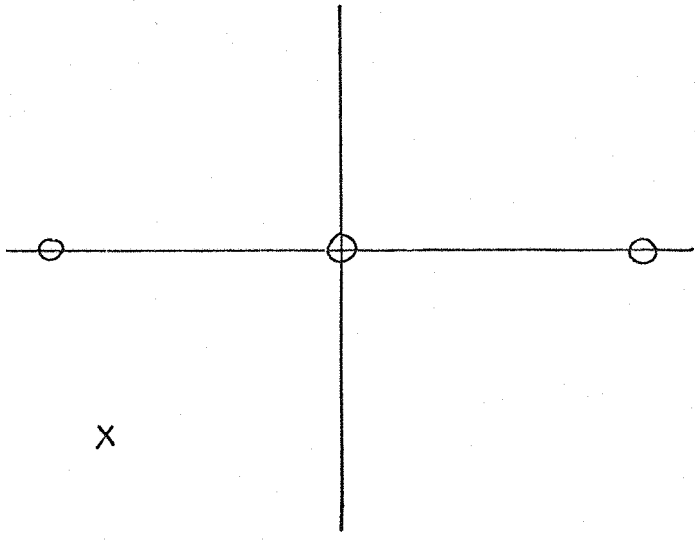
4.4 A Valley-Seeking Technique

All the schemes outlined so far in this chapter have been mode-seeking. However Koontz and Fukunaga (1972a) have suggested a technique which does the opposite. Their technique finds the valleys and partitions the data set so that the cluster boundaries lie in these valleys.

As in their other paper (Fukunaga and Koontz, 1970), discussed in the last chapter, they define the clustering problem as consisting of the definition of a clustering criterion and the construction of a clustering algorithm. This time they discuss a radically different criterion from those based on scatter matrices. Their initial criterion, J, is an attempt to measure the error involved in mapping from the set of data points to the set of clusters. J is of a very general form. However the bulk of the paper discusses the properties of one specific example of J, known as the 'fixed neighbourhood penalty rule' and denoted by J_{2R}. This



(a) The full circles represent the actual modes of point density for a data set. The dotted circles represent the other potential modes.



(b) The modes of the same data set as in (a) after the axes have been rotated through 45° in a clockwise direction. This time there are no other potential modes.

FIGURE 4.2

is defined as 'the total number of distinct pairs of vectors separated by a (Euclidean) distance less than R and assigned to different classes'. Thus in minimizing J_{2R} one ensures that there is little overlap between clusters. As Koontz and Fukunaga point out, any metric could be used to define distance. In their examples, they use Euclidean distance after first normalising to zero mean and unit covariance matrix. Unlike the more usual normalisation to zero mean and unit standard deviation, this normalisation involves rotation. This is equivalent to using the Mahalanobis distance defined on the whole data set.

The clustering algorithm is originally defined for the general criterion J. When applied to J_{2R} Koontz and Fukunaga restate it as a sequence of four steps.

Step 1: Choose an initial classification.

Step 2: For each vector, count how many vectors within a distance R are assigned to each class.

Step 3: Reclassify the vector to the class with the largest number of members within a distance R from it.

Step 4: If any vector is placed in a new class, repeat from Step 2. Otherwise, stop.

This differs from the 'hill-climbing pass' of Friedman and Rubin, mentioned in the last chapter, in that during Step 2 no account is taken of any reclassification already achieved in that particular pass through the data set. As a result, a boundary can only move by a distance R in any one pass.

Koontz and Fukunaga first use a verbal argument to convince their reader that what they have described is a valley-seeking technique.

'Consider vectors along the boundary separating class S_1 from class S_2 at the k'th iteration. Suppose there is a heavier concentration of vectors on the S_2 side of the boundary. Then vectors near the boundary

are re-classified into S_2. Hence the boundary moves into the region previously assigned to class S_1. Therefore, the boundary moves away from the higher concentrations and towards the valleys in the distribution.'

Once a valley of width greater than 2R has 'captured' a boundary it cannot escape, since a boundary moving towards a valley can overshoot by no more than R.

Koontz and Fukunaga then give a mathematical treatment to show that, in the limit as the data set becomes infinite, the only stationary boundaries are everywhere perpendicular to the gradient of the probability density function. Furthermore, the only stable boundaries (in the sense of tending to return to their original positions after a slight perturbation) correspond to 'valleys'. Unfortunately it is not clear how closely the algorithm will correspond to this behaviour for a finite number of data points.

Three problems remain. What value should R be given? How do we determine the optimum number of clusters? To what extent are the results dependent upon the initial partition?

The answers to the first two questions appear to be related. The paper reports a series of experiments in two dimensions on data consisting of 99 points from three normal clusters. Five initial classes were defined and R was varied from 0.1 to 7.0. When R was small no convergence occurred after twenty iterations, when the process was stopped. It was found that most of the data points fell in one class. For large R convergence occurred after a small number of iterations with nearly all the samples in a single class. However there appeared to be an intermediate range of values of R for which the 'correct' results were found. That is, there were three large clusters plus two empty, or nearly empty, clusters. The experiments were repeated with 198 and 300 samples, and similar results were found. Koontz and Fukunaga suggest that these very

satisfactory results are caused by the 'unwanted' boundaries moving downhill to the edge of the data set. Therefore it would appear that as long as there are enough initial classes the actual number does not matter. In fact, Koontz and Fukunaga comment that it may be wise to take more initial classes than are thought necessary in case some boundaries are lost by diverging to the edge of the data set.

The problem of how to define the original partition is not really treated. Presumably it was not found to be critical.

In the author's opinion these experiments exhibit two major defects. Firstly, the experiments are limited to two dimensions. Secondly, all three clusters have unit covariance matrices. The author wonders whether the algorithm could separate high-dimensional ellipsoidal clusters with differing covariance matrices?

Since several different values are taken for R, the most obvious implementation is to first calculate the dissimilarity matrix (using Euclidean distance) and store it for re-use. This produces a program with storage requirements $O(N^2)$. The time requirement for the calculation of the dissimilarity matrix will be $O(N^2)$, but for the bulk of the algorithm will be only $O(N)$. If there is not enough available storage space for the dissimilarity matrix the simplest alternative is to calculate each interpoint distance when needed. This reduces the storage requirements to $O(N)$ at the expense of increasing that part of the algorithm whose time requirement is $O(N^2)$.

However, there is a third possible implementation which is optimum irrespective of whether or not there is enough storage space for the dissimilarity matrix. First create for each data point a list containing all the points within a distance equal to the maximum value of R to be used. Markers can be inserted to partition the list and thereby indicate how much of the list is relevant for any particular choice of R. For

large wordlength machines it should be possible to achieve a further economy of storage by packing more than one element of each list into each word. The result will be a program faster than the previous two and with less storage requirements than the first. This is because the optimum value of R appears, from the experiments, to fall as N increases. Consequently the length of each list will probably not be as much as proportional to N, and hence the storage requirements will increase at a rate less than $O(N^2)$.

Valleys, like modes, are invariant under non-singular linear transformations. That is, each point along the line of a valley will transform in the same way as the data points. Consequently this technique, like Gitman and Levine's, is fully invariant under such transformations.

In a later paper, Koontz and Fukunaga (1972b) have extended their analysis to a more general form of criterion. The paper seems to suffer from most of the defects of the earlier one. In particular, it is not clear how many data points are necessary for reasonable results. Also the experiments they quote are once again limited to two-dimensional data composed of three normal distributions, each with unit covariance matrix. As before, the criterion depends upon a parameter which has to be varied to obtain good results. However the paper contains a heuristic argument which suggests a value for this parameter. This requires considerably less computer time than the trial-and-error method previously suggested.

4.5 The Algorithm of Sebestyen and Edie

Sebestyen and Edie (1966) describe an algorithm which they believe can be used to provide an economic representation of a multivariate p.d.f. Before explaining the relevance of this algorithm in cluster analysis, it will be necessary to describe it in some detail. The p.d.f. is represented as a mixture of multivariate normal distributions. Each

of these constituent distributions is constrained to have a diagonal covariance matrix. Thus each of the normal distributions can be thought of as being centred on an ellipsoidal cell with axes parallel to the axes of the data space.

Assume there are M normal distributions in a space of q dimensions. Let S_{mk} represent the k'th co-ordinate of the mean of the m'th distribution. Let σ_{mk} represent the standard deviation of the m'th distribution in the k'th dimension. Let X_k be the k'th co-ordinate of an arbitrary vector, $\underline{X}$. Finally c_m is a positive weighting factor such that

$$\sum_{m=1}^{M} c_m = 1$$

Then the probability density at $\underline{X}$ will be

$$\frac{1}{(2\pi)^{\frac{1}{2}q}} \sum_{m=1}^{M} \frac{c_m}{\prod_{k=1}^{q} \sigma_{mk}} \exp(-\tfrac{1}{2}Q_m(\underline{X})) \qquad (A)$$

where the quadratic form $Q_m(\underline{X})$ is given by

$$Q_m(\underline{X}) = \sum_{k=1}^{q} \left(\frac{X_k - S_{mk}}{\sigma_{mk}} \right)^2$$

The object of the algorithm is to establish the number and nature of these distributions. In what follows the S_{mk} and σ_{mk} no longer represent the true means and standard deviations but rather the values in use at any particular stage of the algorithm. Each distribution is regarded as being centred on a cell given by

$$Q_m(\underline{X}) \leqslant \tau^2$$

where τ is a parameter that must be defined before the beginning of the

algorithm. Another parameter, Θ , is used to define a 'guard-zone' around each cell thus

$$\tau^2 < Q_m(\underline{X}) \leq (\Theta\tau)^2$$

The algorithm begins by establishing a cell centred on the first data point. Initially each σ_{1k} is equated to a pre-determined constant $\sigma_k(0)$. As each new data point, $\underline{X}$, is presented the $Q_m(\underline{X})$ are calculated. The minimum of these quadratic forms is found. Assume this is $Q_{m_0}(\underline{X})$. Then there are three possibilities.

(1) $$Q_{m_0}(\underline{X}) \leq \tau^2$$

In this case, the point falls in the m_0'th cell. Estimates of the mean and standard deviation are kept for all the points which have fallen in each cell. Consequently, these estimates for the m_0'th cell must be modified to include the new point $\underline{X}$. The S_{m_0k} are equated to the estimates of the co-ordinates of the mean. Each σ_{m_0k} is equated to the maximum of $\sigma_k(0)$ and the estimate of the standard deviation in the k'th dimension.

(2) $$\tau^2 < Q_m(\underline{X}) \leq (\Theta\tau)^2$$

In this case, the point falls in the guard-zone. It is stored and re-considered at a later stage.

(3) $$(\Theta\tau)^2 < Q_{m_0}(\underline{X})$$

In this case, the point falls outside the guard-zone. A new cell is formed, centred on $\underline{X}$ and with its σ_{jk} equated to $\sigma_k(0)$.

Assume c_1 cells have been created after P_1 data points have been considered. Then when P_1 equals $c_1\omega$, where ω is another pre-determined parameter, the stored data points (i.e. the points that fell in guard zones) are allocated to the 'nearest' cell (in the sense of minimizing

$Q_m(\underline{X})$). This allocation of stored points will re-occur for the q'th time when

$$P_q = 2^{q-1} c_q \omega$$

Obviously, determining the values of the control parameters (i.e. τ, Θ, ω and $\sigma_k(0)$) is a major problem. The reason for having a minimum value for the σ_{mk} is to prevent a very large number of cells being established. Consequently, it is at least possible to guess an order of magnitude for the $\sigma_k(0)$. ω is not too much of a problem because the results of the algorithm are not critically dependent upon it. Sebestyen and Edie in fact suggest that a suitable value for ω is 4. However, slight changes in τ and Θ quite seriously affect the performance of the algorithm. Sebestyen and Edie suggest that τ should be approximately $1.4(q+2)^{\frac{1}{2}}$. Their only comment about Θ is that it should be greater than unity!

While Sebestyen and Edie were primarily concerned with representing multivariate p.d.f.'s, Mucciardi and Gose (1972) have considered the algorithm as a clustering algorithm. For, as Sebestyen and Edie pointed out, the cells will tend to stabilize around the modes of the distribution. Hopefully the algorithm will produce a few cells containing a large number of points, plus some additional nearly empty cells. The dense cells can be regarded as clusters, and each of the other cells can be coalesced with its nearest dense cell, using nearest in the same sense as before.

Mucciardi and Gose found Sebestyen and Edie's suggestions for evaluating τ unsatisfactory in high dimensions (i.e. greater than three). Instead they suggest that τ and Θ be chosen so that the initial cells (i.e. with σ_{jk} equal to $\sigma_k(0)$) contain, on average, three data points, whilst the guard-zones contain two. They also suggest a

second pass through the data set. This starts with the cells formed at the end of the first pass. In general the σ_{jk} are re-initialized to $\sigma_k(0)$. However if, for some j and k, σ_{jk} has remained at the value $\sigma_k(0)$ the initial value for that σ_{jk} is reduced to some fraction of $\sigma_k(0)$. The largest σ_{jk} for the cell in question is then re-initialized so as to maintain the same initial volume for the cell.

The result of all this is an algorithm which is undoubtedly fast. Assuming that the number of cells formed is dependent only upon the data structure (and not upon the number of points in the sample) the time requirement will be O(N). Whether the initialization techniques of Mucciardi and Gose work well over a wide range of data sets is still an open question.

The limitation to diagonal covariance matrices will clearly have severe consequences when dealing with ellipsoidal clusters whose axes are not parallel to the axes of the data space. The extension of this technique to employ cells with non-diagonal covariance matrices would increase the execution time for the algorithm by a factor of the order of $\frac{1}{2}(q+1)$. This is because so much of the execution time is spent in calculating the $Q_m(\underline{X})$. In Sebestyen and Edie's original algorithm each quadratic form is the sum of q terms. However in the modified algorithm each quadratic form is the sum of $\frac{1}{2}q(q+1)$ terms.

All the algorithms presented in the first four sections of this chapter are non-parametric. The algorithm in the next section is parametric. However this algorithm seems to occupy a half-way position. It will probably work best where each cluster comes from a multivariate normal population. However, because the algorithm is essentially a mode-seeking one, it will fail to distinguish two normal distributions when the means of the distributions are very close. A truly parametric

technique should continue distinguishing the two distributions until their means actually coincide. On the other hand, because the technique is a mode-seeking one, it will probably achieve reasonable results for ellipsoidal clusters possessing a distribution other than Gaussian.

Well-separated normal distributions remain well-separated and normal after non-singular linear transformations. Consequently, if these assumptions are valid the results of this technique (when adapted to include non-diagonal covariance matrices) should be invariant under such transformations. However, unlike all the other procedures discussed in this chapter, the results of this technique may depend upon the order of presentation of the data points. It is important that they should be randomly ordered. If this is not so, quite misleading results may occur.

It is interesting to note that, given the validity of the assumptions of normality, the allocation of data points to clusters does not follow strict Bayesian decision theory. On the basis of decision theory, a point will be allocated to that cluster which makes the most contribution to the sum in (A). That is, the point is allocated to the m_0'th cluster where m_0 is that value of m which maximizes

$$\frac{c_m}{\prod_{k=1}^{q}\sigma_{mk}} \exp(-\tfrac{1}{2}Q_m(\underline{X})) \qquad (B)$$

c_m is the proportion of points in the m'th cluster. Assume that, of n points so far allocated, n_m have been allocated to the m'th cell. Then c_m can be estimated by

$$c_m = \frac{n_m}{n}$$

Substituting for c_m and taking natural logarithms (B) becomes

$$\ln n_m - \ln n - \sum_{k=1}^{q} \ln(\sigma_{mk}) - \tfrac{1}{2}Q_m(\underline{X})$$

Since n is independent of m, the requirement is to maximize

$$\ln n_m - \sum_{k=1}^{q} \ln(\sigma_{mk}) - \tfrac{1}{2}Q_m(\underline{X}) \qquad (C)$$

Clearly, this is not equivalent to minimizing $Q_m(\underline{X})$. The difference will become particularly evident when the cells differ either in size or in point density. The question naturally arises whether the adoption of the criterion given in (C) would give better or worse results than the original criterion. A criterion similar to (C) can, of course, also be determined for the case when non-diagonal covariance matrices are used.

4.6 Multivariate Mixture Analysis

This technique makes use of the maximum likelihood method, the properties of which were first deduced by Fisher (1922). The maximum likelihood technique is a way of estimating the parameters of a distribution given the general form of the distribution and a sample of independent observations from the population. The assumption made in this section is that the probability distribution, $f(\underline{X})$, of the data space is a mixture of g multivariate normal distributions.

Let there be N sample points, $\underline{X}_1$, $\underline{X}_2$, ... $\underline{X}_N$. Then the likelihood function is defined by

$$L = \prod_{j=1}^{N} f(\underline{X}_j)$$

The exact shape of $f(\underline{X})$ is defined by a number of parameters. These are the mean vectors and covariance matrices of the g normal distributions plus the proportion of each component distribution in the resultant mixture distribution. The maximum likelihood method regards these

parameters as variables and finds those values which maximize L.

Having estimated these parameters, one can regard each distribution as defining a cluster. The data set can then be partitioned amongst the g clusters by using Bayesian decision theory, as outlined in the last section.

Wolfe (1970) gives a brief review of some previous attempts to use the maximum likelihood approach for special cases of the above problem (e.g. when the distribution is univariate). He then goes on to develop the theory for the general case. Whilst it is easy to set up the maximum likelihood equations by differentiating ln L with respect to the various parameters, the solution of these equations is much more difficult. A large part of Wolfe's paper is concerned with the description of an iterative numerical technique to solve this problem.

Wolfe has written a program implementing his ideas. This program contains two options. NORMIX is the general case option whilst NORMAP is for the special case where each normal distribution is assumed to have equal covariance matrices. As Wolfe points out, 'NORMAP could be considered a continuous version of the discrete partitioning procedure of Friedman and Rubin. The two methods tend to coincide in the limiting case of widely separated types.' Similarly, NORMIX is a continuous version of the $\prod_{i=1}^{g} |W_i|^{N_i}$ criterion suggested by Scott and Symons.

In view of what has just been said, it is not surprising that NORMIX and NORMAP share a difficulty with the optimization-partitioning techniques. The results of the analysis are dependent upon the initial partition. Wolfe's iterative technique will even diverge for some initial partitions. In both cases, the solution to the difficulty appears to be to take a number of different initial partitions and compare the results. In the case of the optimization-partitioning techniques, the best overall result will be that which gives the optimum

value for the criterion. In the case of Wolfe's program, the best overall result will be that which gives the largest value for L. However, to have to repeat the analysis a number of times is clearly very expensive in computer time.

Wolfe's technique has one interesting property not really shared by any other technique described in this report. It allows a much more definite answer to the question, 'what is the best value for g?' Consider the two alternatives $g = r$ and $g = r'$. Let L_r and $L_{r'}$ be the maximum values of L for the two cases. Consider the quantity χ^2 defined by

$$\chi^2 = -2 \ln(L_r/L_{r'})$$

In the limit, as the number of data points tends to infinity, χ^2 will have a chi-squared distribution with degrees of freedom equal to the difference in the number of parameters to be estimated in the two cases (Wilks, 1962). This enables a simple test of the two hypotheses to be constructed. Such a test is known as a likelihood ratio test. As Scott and Symons point out, the necessary conditions are not fulfilled to enable this test to be used in the case of the optimization-partitioning criterion they discuss (this is because of the assumption of non-overlapping distributions).

Wolfe gives a number of examples of the use of NORMIX and NORMAP. In one example, three clusters were artificially generated consisting of 100, 75, and 50 points in two dimensions. The clusters each had multivariate normal distributions with different covariance matrices. The χ^2 likelihood ratio test applied to the results of NORMIX indicated the existence of more than three types in the data. However, it was found that the fourth cluster contained only seven points. The chi-squared approximation is, anyway, inaccurate for this sample size.

In addition, Wolfe quotes an example employing the Iris data published by Fisher (1936) and also used by Friedman and Rubin. The results obtained from NORMAP were found to be identical to those obtained by Friedman and Rubin with the $|W|$ criterion. Unfortunately, on the basis of the published results it is impoosible to make a comparison of the two procedures from the computational standpoint. Because of the complexity of NORMAP and NORMIX, the author is unable to deduce the relationship between execution time and number of data points. It is clear, however, that the storage requirement of the program is O(N).

A mixture of multivariate normal distributions remains a mixture of multivariate normal distributions after a non-singular linear transformation. Furthermore, if the covariance matrices are equal before the transformation they remain equal afterwards. Consequently the results of both NORMIX and NORMAP are invariant under such transformations. This of course also follows from their equivalence with the optimization-partitioning techniques of Scott and Symons.

4.7 A Simple Comparison

A simple one-dimensional example may help to illustrate the differences between the various techniques discussed in this chapter. Consider an equal mixture of two univariate normal distributions, each with variance σ^2. When the means coincide, the distribution is normal. As they separate the distribution does not become bimodal until they differ by 2σ (Marriott, 1971).

One would not expect any of the non-parametric mode or valley-seeking algorithms to separate the two distributions in the case when the means are separated by less than 2σ. Nor would one expect the Sebestyen and Edie algorithm to separate the two distributions, since it also is essentially a mode-seeking one. However NORMIX and NORMAP ought to be

successful in separating these distributions.

When the means are separated by rather more than 2σ and two well-defined modes exist, all the techniques ought to give a two-cluster solution. The way the data set is partitioned amongst the two clusters will depend on the particular technique used.

As the means are separated the Scott and Symons criteria should give a one-cluster solution until the degree of overlap becomes 'negligible'. What negligible means in this context is an open question. Presumably two distinct modes will have to exist and be quite well-separated.

CHAPTER 5 MAPPINGS

5.1 Introduction

The techniques discussed in this chapter differ from the others in this report in that they do not themselves output the cluster structure of a data set. These techniques map each data point in the original q-dimensional data space into a point in an r-dimensional space, where $r < q$. This reduction in dimensionality is sometimes termed feature extraction. There are two principal objectives. Firstly, by mapping from a space of high dimensionality to one of low dimensionality one achieves a data set which can be more economically manipulated. This was particularly the motivation behind the linear technique discussed in section 5.2. In addition, by mapping into a 1, 2, or 3-dimensional space one can display the data set visually. This was the main motivation behind the non-linear techniques discussed in sections 5.3 and 5.4. It is also the objective which is of interest in this chapter. For having so displayed the data it may be possible to cluster it visually. The human observer is able to achieve a global (or 'gestalt') clustering which is not sensitive to the presence of noise points. As a disadvantage, clusters which are distinct in the q-dimensional space may overlap in the r-dimensional space, thereby obscuring the data structure.

5.2 Principal Components Analysis

This is by far the oldest of the mapping techniques to be considered in this chapter. The procedures in general use today are due to Hotelling (1933) but the method was effectively suggested by Pearson (1901).

Let the N data points be represented by column vectors $\underline{X}_1, \underline{X}_2, \ldots \underline{X}_N$. Consider the expansion of each such vector

$$\underline{X}_i = \sum_{j=1}^{q} y_{ij}\underline{u}_j \qquad (A)$$

Here the $\underline{u}_j$ are a set of q orthonormal vectors. By ignoring all but r of the terms in the summation in (A) one obtains a reduction in dimensionality from q to r. The other (q-r) terms must be approximated by vectors independent of the choice of $\underline{X}_i$. This will give a vector $\underline{Y}_i$ which is an approximation to $\underline{X}_i$.

$$\underline{Y}_i = \sum_{j=1}^{r} y_{ij}\underline{u}_j + \sum_{j=r+1}^{q} b_j\underline{u}_j \qquad (B)$$

Here the b_j are constants. One measure of the error involved in the approximation (B) will be given by

$$K = \sum_{i=1}^{N} d(X_i, Y_i)$$

If d(,) represents the squared Euclidean distance, K is termed the 'mean-square error'. The objective of principal components analysis is to find an approximation of the form of (B) which minimizes this mean square error.

It can be shown (e.g. Fukunaga, 1972, Chapter 8) that such a mapping is defined in the following way. Let the $\underline{u}_j$ be the eigenvectors of the covariance matrix of the data set. Let the eigenvalues be arranged in order of magnitude, with the largest first. Assume that $\underline{u}_j$ is the eigenvector associated with the j'th eigenvalue. Finally define the b_j by

$$b_j = \underline{u}_j^T E(\underline{X})$$

Here the superscript T denotes transposition whilst the operator E denotes expectation.

By equating r to 2, one arrives at a set of two-dimensional vectors with co-ordinates (y_{i1}, y_{i2}) given by (B) which can be plotted and inspected visually. Frequently each of the two dimensiona is normalised

to unit variance before the data is plotted. Hopefully the two-dimensional plot will now allow the observer to cluster the data. Unfortunately there are several disadvantages. Firstly, the results are invariant only under orthogonal transformations. Clearly they will be completely altered by a change of scale. Secondly. two clusters may overlap completely in 2-space when they are well-separated in q-space. All that is necessary for this to happen is that the clusters overlap in the subspace spanned by $\underline{u}_1$ and $\underline{u}_2$. The point here is that the criterion which is being minimized (i.e. mean-square error) bears no natural relationship to the clustering problem. There is no real reason why it should give satisfactory results in clustering.

Computationally the algorithm is cheaper than the nonlinear techniques to be considered later. Both storage and time requirements are $O(N)$. As a result principal components analysis can be used on far larger data sets than the nonlinear techniques can handle. The procedure is iterative only to the extent that the algorithm for finding the eigenvectors is iterative. Furthermore it is not necessary to find all the eigenvectors but merely those associated with the two largest eigenvalues.

5.3 Sammon's Nonlinear Mapping

Sammon (1969) has attempted to overcome the deficiencies of principal components analysis by using a radically different criterion from the mean-square error. Sammon's algorithm attempts to maintain as unchanged as possible the relationship between each data point and those points close to it. This approach should be specially suitable for cluster analysis. He first defines a criterion which represents the extent to which the distances between each data point and its neighbours are altered by the mapping. The aim of his algorithm is to find a

mapping which minimizes this criterion.

Let d^*_{ij} and d_{ij} represent the distances between the i'th and j'th points in the q- and r-spaces respectively. Any definition of distance could be used but Sammon uses Euclidean distance. Then the error involved in the transformation $d^*_{ij} \longrightarrow d_{ij}$ can be represented by $(d^*_{ij} - d_{ij})^2$. This error must be summed over all possible combinations of i and j. However it is required to minimize this error chiefly for those points close together, if need be at the expense of those far apart. Consequently the i,j'th term in the summation is weighted by d^{*-1}_{ij}. This gives

$$\sum_{i<j} \frac{(d^*_{ij} - d_{ij})^2}{d^*_{ij}}$$

Finally a normalisation is performed by dividing the criterion by $\sum_{i<j} d^*_{ij}$. This renders the criterion dimensionless. It does not affect the algorithm but it does mean that when the final configuration has been achieved a number can be attached to it which measures the success of the mapping in preserving inter-neighbour dissimilarity. Thus the final criterion is given by

$$K = \frac{1}{\sum_{i<j} d^*_{ij}} \sum_{i<j} \frac{(d^*_{ij} - d_{ij})^2}{d^*_{ij}}$$

K can now be regarded as a function of the co-ordinates of each of the points in the r-space. Thus if there are N data points, K is a function of Nr variables. Sammon then uses an iterative steepest descent procedure to minimize K. The initial r-space configuration can be chosen at random. Alternatively Sammon suggests finding those r of the original axes along which there are the largest variances. The original configuration can then be taken as the projection of the data points in the sub-space spanned by these r dimensions. The final result of the

algorithm is a set of values for the Nr co-ordinates (and thus a configuration in r-space) which minimizes K.

Sammon's paper contains the results of a number of experiments, both on artificial and on real data. He shows that in some cases his algorithm will produce a mapping which allows 'correct' clustering when this is not possible from the results of a principal components analysis. The algorithm appears to be particularly useful when the data points are highly structured but in a very nonlinear way, e.g. when they lie along a helix in q-space. It is clear at once from the 2-space plot that there is a very definite structure. As with all dimensionality reducing mappings, there is no unique q-space structure corresponding to any given r-space structure. Furthermore, because the mapping cannot be represented in a simple mathematical form it does not seem to be possible to make many comments about the q-space structure.

Apart from the number of clusters present the only other definite piece of information one may obtain is the intrinsic dimensionality of the data. Assume that the original data points in q-space lie on (or in practice close to) a surface which is defined by a minimum of p parameters. Clearly p will be less than or equal to q. Then the data is said to possess an intrinsic dimensionality p. For example, in a three-dimensional problem all the data points may lie very close to a helix. Then although the dimensionality of the data space is 3, the intrinsic dimensionality of the data is 1. Each point on the helix can be defined by one parameter alone (e.g. the distance from one fixed point on it). Then if the mapping is performed from q-space into spaces of steadily increasing dimensionality (e.g. 1-space, 2-space, etc.) there should be a considerable reduction in the final value of K when the correct intrinsic dimensionality is chosen.

In practice the author suspects that data very rarely follows these highly non-linear but very well-defined forms. A typical pattern recognition clustering problem is the 'pure signal plus noise' situation outlined in Chapter 1. Here each pure signal would be represented by a point in hyperspace and the noise would cause the data points to be clustered around this point. Possibly if the statistics of the signal are continuously varying in a non-random way one might find the data points to be clustered around a line. However this would seem to be an exceptional situation.

One disadvantage of the algorithm is that, as with many iterative search techniques, one can never be sure that the solution obtained is not just a local minimum of K. One possibility is to repeat the analysis with a number of different starting configurations. In actual fact it may not matter that the overall minimum has not been found if the local minimum value for K is sufficiently small (say, $\leqslant 0.1$). As long as one has obtained a 'good' mapping the fact that it is not the best is irrelevant.

Another disadvantage is shared with principal components analysis. The results of the mapping will not be invariant under non-singular linear transformations. Because the algorithm depends upon the use of a distance measure the final configuration will depend upon the initial choice of scale. Consequently the clustering performed by the human observer will also be sensitive to the choice of scale.

Computationally, the algorithm will be more expensive than principal components analysis. The time requirement is $O(N^2)$. However it will also depend upon the data structure, that is how easily (in how few iterations) K can be minimized. Both the d^*_{ij} and the d_{ij} are each used more than once. Consequently to calculate them more or less when needed

will produce a very slow algorithm. In particular the d^*_{ij} are unchanged throughout the algorithm. As a result Sammon recommends storing both dissimilarity matrices. Consequently the storage requirement is $O(N^2)$ and this imposes a definite limit on the size of data set which can be handled. To overcome this problem Sammon suggests using a clustering algorithm to reduce the data set to a manageable size, say 250 vectors. For data reduction the choice of clustering algorithm is not critical. As was pointed out in Chapter 1 there is no unique solution to the data reduction problem but rather a multiplicity of acceptable solutions.

5.4 A Relaxation Method for Nonlinear Mapping

Chang and Lee (1973) make use of the same criterion as Sammon. The only difference is that they use squared Euclidean distance rather than ordinary Euclidean distance. However a relaxation method is then used to minimize K. Instead of modifying the whole r-space configuration in one step they take a pair of points at a time. A gradient method is then used to alter the co-ordinates of these two points to minimize K. A heuristic has to be introduced to ensure that if the points are close together they are affected more than if they are far apart. After all the pairs of points have been considered the algorithm has performed one iteration. Further iterations are repeated until convergence is reached.

Chang and Lee call this algorithm 1. As it stands it possesses much the same advantages and disadvantages as Sammon's algorithm. In algorithm 1 there is no need to store the interpoint distances in the r-space since they are essentially needed only once. However the interpoint distances in the q-space are used at every iteration and so by storing them the execution time of the algorithm is significantly reduced. Thus both the storage and time requirements are also $O(N^2)$.

Chang and Lee then suggest a modification to this algorithm,

algorithm 1*. This is known as the frame method and is similar to the frame method for forming a sub-minimal spanning tree suggested by Lee and discussed in Chapter 2. Firstly M points are chosen as frame points from the N original data points. Let the data points be represented by vectors $\underline{X}_1$, $\underline{X}_2$, ... $\underline{X}_N$. Assume that the first M of these are chosen as frame points. Then algorithm 1 is applied to these M points to produce M points in the r-space denoted by $\underline{Y}_1$, $\underline{Y}_2$, ... $\underline{Y}_M$. Algorithm 1* then attempts to find (N-M) points in the r-space (denoted by $\underline{Z}_1$, $\underline{Z}_2$, .. $\underline{Z}_{N-M}$) so that the distances between each frame point and each non-frame point are minimally distorted by the mapping. That is, each distance of the form $d(\underline{Y}_i,\underline{Z}_j)$ is computed in turn and compared with the original distance in the q-space, i.e. $d(\underline{X}_i,\underline{X}_{M+j})$. $\underline{Z}_j$ is then modified so as to minimize K. The extent of this modification is a monotonically decreasing function of $d(\underline{X}_i,\underline{X}_{M+j})$. Thus this stage of the algorithm is concerned with the relationships between the frame points and non-frame points. The relationships amongst the frame points has already been considered whilst the relationships amongst the non-frame points are ignored.

A complete pass through the M(N-M) distances of the form $d(\underline{Y}_i,\underline{Z}_j)$ is termed one iteration. The iterations are repeated until convergence occurs or a fixed number of iterations has been used up. Thus the first part of algorithm 1* requires the storage of $\frac{1}{2}M(M-1)$ interpoint distances whilst the second part requires the storage of M(N-M) distances. The storage used for the first set of distances could be used for the second set, since the intra-frame distances are not needed in the second stage of the algorithm. They are needed, however, if it is desired to evaluate K for the final configuration. Because of the reduced storage requirements of algorithm 1*, it can be used on much larger data sets than either algorithm 1 or Sammon's algorithm. Furthermore, since less interpoint distances need to be calculated, algorithm 1* will be faster

than algorithm 1 and Sammon's algorithm.

Apart from the computational differences algorithm 1 and algorithm 1* share the properties of Sammon's algorithm. Chang and Lee claim that for one particular example, algorithm 1* gives better results than Sammon's method. Whether this would tend to be so in general or whether it is a function of the particular data set is an open question. The comments made in Chapter 2 about the choice of frame points for Lee's sub-minimal spanning tree are also relevant here. A frame set chosen totally, or even largely, from one cluster could give quite misleading results.

CHAPTER 6 MISCELLANEOUS TECHNIQUES

6.1 ISODATA

A number of iterative techniques have been described in the literature which share the property of allowing the number of clusters to vary during the course of the algorithm. The most famous of these techniques has been developed by Ball and Hall (1967) and is called ISODATA. In the form given in the reference, the algorithm is applicable to binary-valued data. This is because Ball and Hall are social scientists and sociological data seems to be mainly of this form. However the modification to real-valued data is trivial and Ball and Hall claim they have developed a program for this case.

The algorithm can most easily be described as a number of steps.

(1) A 'typical' set of data points are chosen as initial 'cluster points'.

(2) Each data point is allocated to a group centred on the nearest cluster point. Euclidean distance is used here.

(3) For each group the centroid and 'within-group variability' are calculated. The centroids now become the new cluster points. The term 'within-group variability' is not defined in the paper but presumably it refers to something like the trace of the covariance matrix for each group. If any group's within-group variability exceeds a threshold θ_E the algorithm proceeds to step 4. Otherwise it stops and the results are output.

(4) Each group whose within-group variability is greater than θ_E is split into two. Firstly, the variable with greatest variance for this group is found. Two new cluster points are then formed identical in all but this variable to the centroid of the group being split. One of the cluster points takes the

value +1 in the maximum-variance dimension; the other takes the value -1. This is because the algorithm was designed for binary variables which can only take these two values. Clearly this step could easily be adapted for continuous variables.

(5) The data points are now re-allocated to their nearest cluster points and the centroids of each group are computed.

(6) The distances between each pair of centroids are calculated.

(7) All groups whose centroids are closer together than a threshold value Θ_C are combined. The algorithm then returns to step 2.

Clearly this algorithm is looking for compact spherical clusters. In this respect it resembles the error sum of squares techniques discussed in chapter 3. In fact, the algorithm is another elaboration of the basic algorithm shown in Figure 3.1. Consequently it suffers from the same disadvantages. The results will not be invariant under non-singular linear transformations and the algorithm will tend to split up elongated clusters. Ball and Hall describe it as a data-reducing algorithm and do not make any claims as to its value in finding a true typology. Like the basic algorithm of Figure 3.1 the storage and time requirements are O(N). One of the difficulties of this algorithm is the choice of Θ_E and Θ_C. It may be necessary to change one or both of these parameters and re-run the program until a suitable level of clustering has been found. Northouse and Fromm (1973) have suggested heuristics for computing reasonable values for these parameters but no really convincing proof of the general efficacy of these algorithms exist. In the opinion of the author ISODATA is likely to achieve results only a little superior to the results of the algorithm of Figure 3.1 at the cost of a much-increased execution time.

6.2 MAXIMINDIST

This technique, due to Batchelor and Wilkins (1969), was originally designed for initialising the compound classifier algorithm invented by Batchelor (1968). The compound classifier algorithm is an error-correcting procedure for learning with a teacher which uses hyperspheres as discriminant surfaces. Batchelor and Wilkins define the word cluster in a way which corresponds exactly to the usage of the word in complete linkage analysis (see Chapter 2).

The algorithm takes from the set of data points $\underline{X}_1$, $\underline{X}_2$, ... $\underline{X}_N$ a set of cluster points $\underline{B}_1$, $\underline{B}_2$, ... $\underline{B}_M$ where normally M is much less than N. Firstly $\underline{B}_1$ is set equal to $\underline{X}_1$. $\underline{B}_2$ is then equated to that data point furthest from $\underline{B}_1$. Euclidean distance is used. To find $\underline{B}_3$ the minimum of $d(\underline{X}_i,\underline{B}_1)$ and $d(\underline{X}_i,\underline{B}_2)$ is calculated for i=3 to N. The maximum of this set of (N-2) distances is then found. The data point which produces the maximum value becomes $\underline{B}_3$. Hence the name MAXIMINDIST. The algorithm proceeds in this fashion until M cluster points have been found. This is equivalent to performing a complete linkage analysis with a threshold set at a level which gives exactly M clusters. Any data point in an unrepresented cluster will always be further away from its nearest cluster point than any data point in a represented cluster. Thus if there are exactly M clusters (in the complete linkage sense) one cluster point must come from each one. Since in complete linkage analysis all intracluster distances are less than all intercluster distances it is now a trivial matter to construct the complete partition by assigning each data point to the cluster centred on its nearest cluster point. However MAXIMINDIST has the advantage that one does not need to know what threshold value to choose to give the desired value for M. Furthermore the algorithm may also give an indication of what is the most significant level of

clustering. That is to say, if there is a level at which all the intra-cluster distances are much smaller than the intercluster distances this will be apparent. When all the clusters at this level have been found the maximum of the minimum distances from each remaining data point to the cluster points will drop drastically.

Clearly MAXIMINDIST possesses all the defects of complete linkage analysis. The main objections are that it is invariant only under orthogonal transformations and that, since it is looking for compact spherical clusters, it will tend to break up any elongated clusters present. This makes it suitable for finding a true typology in only a limited number of cases. However it does seem very useful in data reduction. In the author's opinion it has one considerable advantage over the algorithm of Figure 3.1 and ISODATA, when a large number of points is to be represented by a smaller number. If these last two algorithms are initialised by selecting a random sample from the data set it may be possible to leave unrepresented a small number of data points well separated from the bulk of points. Consider the situation of Figure 6.1. Here the circles indicate the boundaries of clusters. Assume that the small cluster A contains only 10 points whilst the other two contain 100 each. Imagine that the objective is to pick out three points to represent the sample. Assume that three points are chosen at random to initialise the algorithm of Figure 3.1. Then if two of these initial points come from cluster B the algorithm could easily produce cluster points as shown by crosses in Figure 6.1(a). However MAXIMINDIST will pick one point from each of the three clusters, as shown in Figure 6.1(b). The cluster points in C and B will each represent 100 points whilst the cluster point in A will represent ten. It seems, then, that MAXIMINDIST will be specially useful as a data-reduction algorithm when the number of cluster points is very much less than the number of data

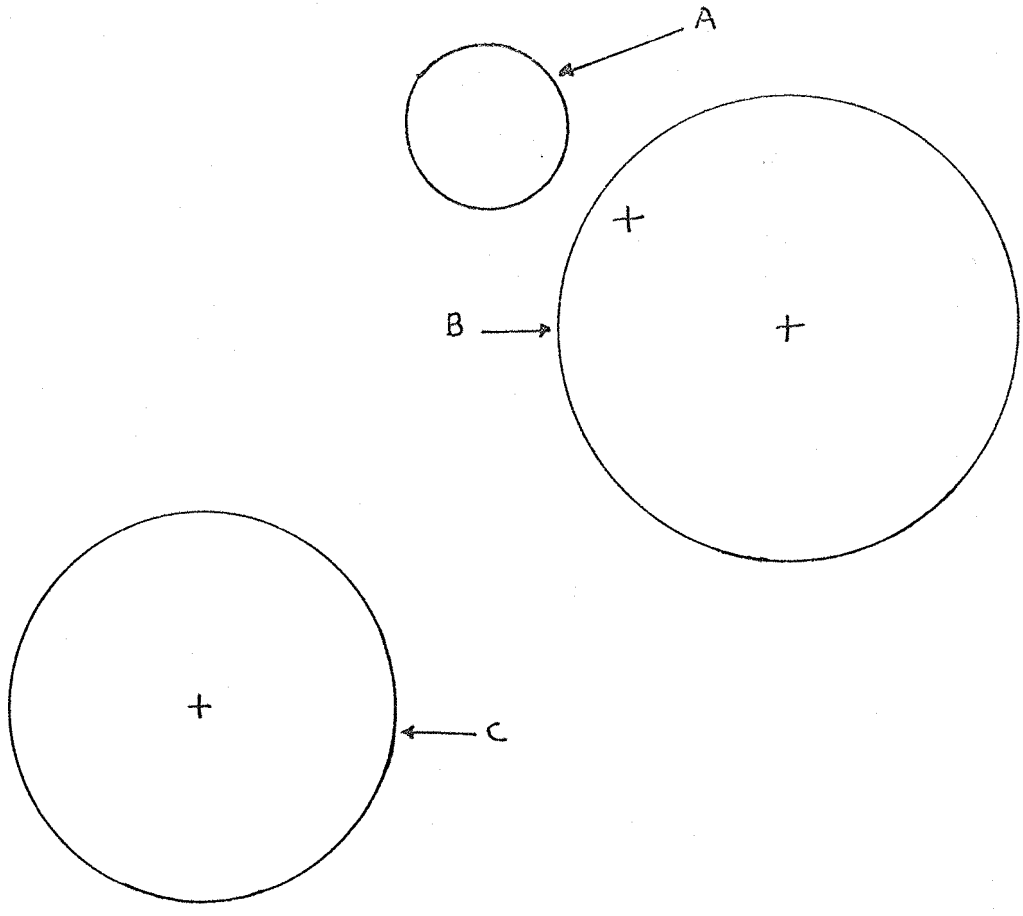


FIGURE 6.1(a)

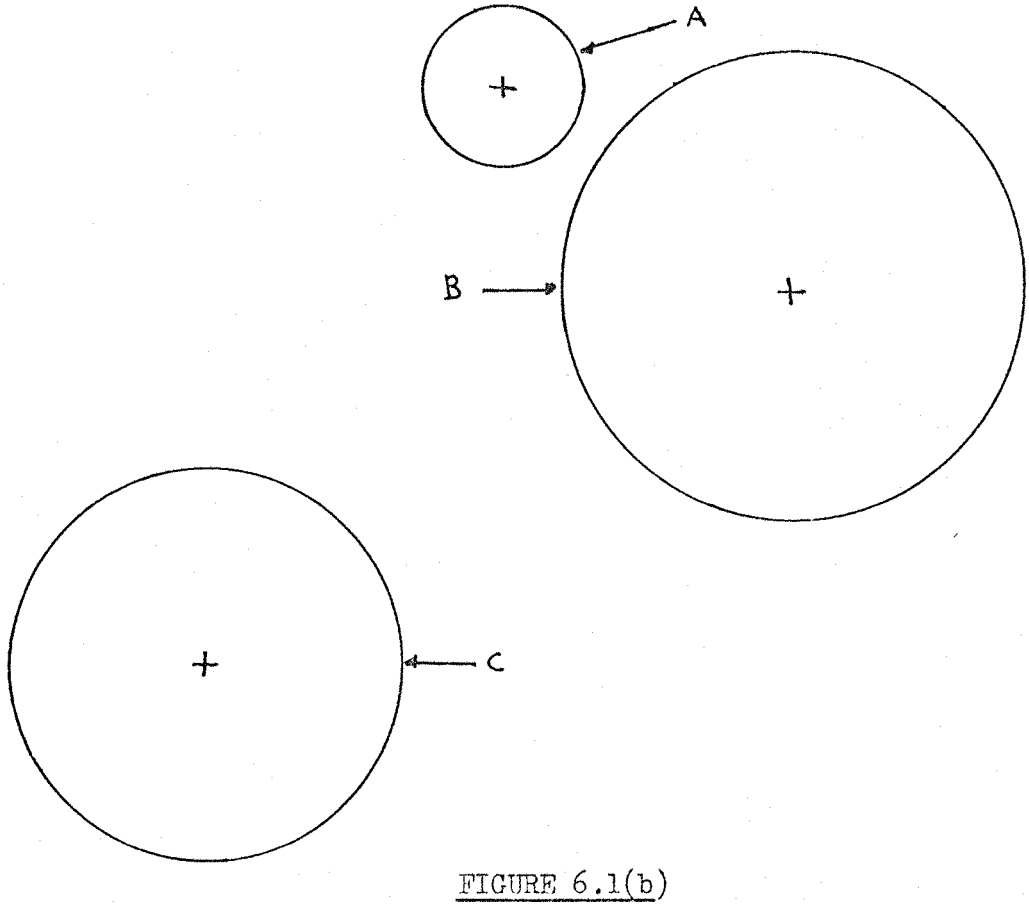


FIGURE 6.1(b)

points. MAXIMINDIST is thus a superior alternative to random sampling.

It is interesting to compare MAXIMINDIST and the usual complete linkage algorithm from the computational standpoint. Consider first the latter. As each data point is introduced its distance from the data points so far considered must be computed. Thus each interpoint distance is used once only. Consequently there is no need to store the dissimilarity matrix and the storage requirements will be O(N). Since for a high-dimensional example almost all the execution time of the program will be taken up in computing distances the execution time will be approximately proportional to $\frac{1}{2}N(N-1)$. MAXIMINDIST, on the other hand, continually re-uses the same interpoint distances. Consider $d(\underline{B}_1,\underline{X}_N)$. This will be needed (M-1) times. Consequently much time can be saved by storing the $d(\underline{B}_i,\underline{X}_j)$ as calculated. This involves the storage of (N-M-1)(M-2) distances. Consider searching for $\underline{B}_m$. One needs to know the distances between $\underline{B}_1$, $\underline{B}_2$, ... $\underline{B}_{m-1}$ and the (N-m+1) remaining data points, i.e. (N-m+1)(m-1) distances. The (m-1) distances from the cluster points to the data point that becomes $\underline{B}_m$ will not be used again. Hence one needs to store only (N-m)(m-1) distances. This is a monotonically increasing function of m for $m < \frac{1}{2}(N+1)$. Consequently it will be maximum for m=M, given $M < \frac{1}{2}(N+1)$. However any distances computed during the search for $\underline{B}_M$ will never be re-used since the algorithm does not search for $\underline{B}_{M+1}$. Consequently the maximum number of distances in storage at any time will be after $\underline{B}_{M-1}$ has been found, i.e. (N-M+1)(M-2) distances. The execution time will be approximately proportional to this number. Consequently MAXIMINDIST is the faster of the two algorithms. This becomes more emphasised when one remembers that the complete linkage analysis may have to be repeated several times to find the required level of clustering.

If the amount of mainframe memory available does not permit

MAXIMINDIST to store a large number of distances the argument becomes more complex. Assume that m cluster points have been found. Then the distances from each of these cluster points to each of the remaining (N-m) data points is needed to find B_{m+1}. This will be m(N-m) distances. Summing this for m = 1 to m = M-1 gives

$$\sum_{m=1}^{M-1} m(N-m) = \sum_{m=1}^{M-1} mN - \sum_{m=1}^{M-1} m^2$$

The first of the two terms on the right-hand side is simply an arithmetic series and equals $\frac{1}{2}M(M-1)N$. The second term can be evaluated by the formula

$$\sum_{i=1}^{k} i^2 = \frac{2k^3+3k^2+k}{6}$$

This gives

$$\sum_{m=1}^{M-1} m(N-m) = \frac{M(M-1)N}{2} - \frac{2M^3-3M^2+M}{6}$$

For $1 \ll M \ll N$ this can be approximated by $\frac{1}{2}M^2N$. The ordinary complete linkage algorithm computes $\frac{1}{2}N(N-1)$ distances. Assume that I different values of the threshold are necessary to find the desired clustering level. Then the total number of distances computed will be approximately $\frac{1}{2}IN^2$. Cimparing the execution time of the two algorithms gives the ratio $M^2 : IN$. Thus for $M < (IN)^{\frac{1}{2}}$ MAXIMINDIST is the algorithm to use. For $M > (IN)^{\frac{1}{2}}$ the usual complete linkage algorithm should be used. The factor I depends upon the way one attempts to find the 'correct' threshold. Presumably some goal-seeking heuristic could be used, and a reasonable value for I should be less than ten, say. In fact the advantage of this approach in data reduction will really arise only when $M \ll N$. For it is in this case that the problem illustrated in Figure 6.1 becomes

appreciable. Consequently MAXIMINDIST would normally be the algorithm to use.

6.3 Centroid Cluster Analysis and Median Cluster Analysis

Centroid cluster analysis was originally proposed by Sokal and Michener (1958). It is interesting because it is a technique which produces a dendrogram and yet it is not a linkage technique in the same way as those discussed in Chapter 2. Groups are represented by their centroids. The distance between two groups is calculated as the (Euclidean) distance between the centroids. At every stage in the analysis those two groups closest together are merged. The procedure starts with one data point in each group and proceeds until all the data points are in the same group. Thus the procedure differs from the techniques of Chapter 2 in that distances other than the original inter-point distances are used.

The technique is of course invariant only under orthogonal transformations. Furthermore the results have to be output as a dendrogram-like structure. Both the time and storage requirements will be $O(N^2)$. The storage requirements could be reduced to $O(N)$ only at the expense of continually re-calculating the same distances. However this would make the time requirement $O(N^3)$.

The centroid method also has the disadvantage that if two groups of very different size are combined the properties of the smaller group will virtually be swamped by those of the larger group. To overcome this deficiency, Gower (1967) has suggested another technique called median cluster analysis. This is identical to centroid cluster analysis except that when two groups are merged the new group is represented by a point mid-way between the points representing the two original groups. The name of the technique derives from the fact that if two points X and Y

are merged, and then a third point Z is joined to the resultant cluster, the point representing the whole group will lie along a median of the triangle defined by X, Y and Z.

Neither of these techniques are suited to finding a true typology. For data reduction they seem in no sense superior to the algorithm of Figure 3.1, which would be cheaper to implement.

6.4 'Dynamical' Clustering

At least two papers have appeared on the subject of what might be called 'dynamical cluster analysis' (Sneath, 1967; Watanabe and Harada, 1974). The data points are assumed to move around the data space in a manner similar to the motion of point masses in physical space. The algorithm simulates a force between the points similar to gravitational attraction. The points are supposed to collapse on each other. In doing so they coalesce to form clusters. Neither of the algorithms in the two papers quoted has (to the knowledge of the author) actually been implemented. Sneath's algorithm is particularly elaborate and he goes so far as to admit the difficulty being experienced in programming it. The author is suspicious whether these algorithms will give the expected behaviour. The analogy with dynamics used to defend them is in no sense an exact one. Consequently the predictions made on the basis of this analogy must be suspect. The real fault of the algorithm seems to be that they do not proceed from a rigorous definition of what the clustering algorithm should achieve but instead argue in terms of a vague and unconvincing analogy.

CHAPTER 7 SLEEP AND THE E.E.G.

7.1 Introduction

The purpose of this chapter is to give a little of the background behind the data set used in the next chapter to compare some clustering algorithms. In Chapter 1 it was pointed out that the typical pattern recognition data set possesses three chief characteristics. Firstly the variables are usually continuous; or rather the quantization level is very small compared with the range. Secondly the data sets are very large. Thousands of data points per sample is not uncommon. Thirdly there is an element of randomness in the data far exceeding that due to the error in measurement. The electroencephalographic (E.E.G.) data described in this chapter possesses all three of these characteristics. For this reason, and because of its availability, it was chosen as an example. However, throughout this and the following chapter it should be borne in mind that the primary objective here is not to achieve any new insight into the nature of the E.E.G. That is far beyond the scope of this thesis. The primary objective is to compare the behaviour of some clustering algorithms when applied to real data.

7.2 The Nature of the E.E.G.

For a detailed description of the nature and measurement of E.E.G. activity see Cooper, Osselton and Shaw (1969). Briefly, an E.E.G. can be defined as a recording from the scalp of the spontaneous electrical activity of the brain. This is measured as the electrical potential difference between two electrodes attached to two points on the scalp of some animal or human. Taking these measurements is itself a difficult technical problem which is described by Cooper et al. The measurements are of the order of tens of microvolts and most of the energy of the

signal is below 32 Hz. The most common way of recording E.E.G.'s is by use of a pen recorder. In this instrument a long strip of paper is moved past a pen which is being deflected in a direction perpendicular to the direction of motion of the paper and to an extent proportional to the magnitude of the E.E.G. The result is a trace of the E.E.G. waveform on the paper.

The experienced clinician is able to use these waveforms as a tool in the diagnosis of epilepsy and in the location of tumours. The E.E.G. waveform possesses a rhythmical nature. Consequently at any given time it can be characterised by its chief frequency components. It has been found convenient to classify the E.E.G. frequencies into the following ranges or bands.

Less than 4 Hz. (but not including any d.c. component):	delta
4 to less than 8 Hz.:	theta
8 to 13 Hz. inclusive:	alpha
Greater than 13 Hz.:	beta

Recently computers have come to be used in the analysis of E.E.G.'s. For this purpose the waveform is sampled and input to a computer by use of analogue-to-digital converters. Signal analysis techniques are then used to produce descriptors which characterize the data. This is more fully discussed in sections 7.4 and 7.5.

Some attempts have been made to explain the origin of the E.E.G. in terms of a model of neurone-functioning. For a brief discussion of this subject see Hjorth (1973). However the subject is not yet sufficiently developed to be able to influence the analysis of real E.E.G.'s.

7.3 The E.E.G. during Sleep

Most clinicians divide sleep into two categories, 'paradoxical' sleep and 'orthodox' sleep. The former is characterized by rapid eye

movements. For this reason it is also termed REM sleep and most dreaming probably occurs during this paradoxical sleep. Orthodox sleep is called non-REM sleep (NREM). It was originally thought that orthodox sleep was much 'deeper' than paradoxical sleep. However this is not so. As a proof of this it has been shown that the muscles of the larynx are actually more relaxed during paradoxical sleep than during orthodox sleep. Oswald (1966) gives an interesting layman's account of this and many other aspects of sleep research.

Dement and Kleitman (1957) have classified E.E.G. patterns found in orthodox sleep into the following four 'stages':

Stage 1. Low voltage signal with irregular frequency.

Stage 2. 13 to 15Hz. sleep 'spindles' and 'K-complexes' in a low voltage background. A spindle is defined by Cooper et al. as a 'sequence of sinusoidal-like waves lasting a second or two and of gradual onset and decay'. A K-complex is defined as a 'transient complex waveform consisting of slow waves sometimes associated with sharp components and often followed by a sequence of waves at about 14Hz.'

Stage 3. Sets of large delta waves appear frequently.

Stage 4. E.E.G. composed almost entirely of delta waves.

In addition to these four stages of orthodox sleep, a human subject spends some time in paradoxical sleep. During paradoxical sleep a low voltage irregular waveform appears, not unlike that found during drowsiness. This is referred to as stage REM and constitutes the fifth of the five stages of sleep.

7.4 Pattern Recognition and the Sleep E.E.G.

The classification of the E.E.G. of a sleeping person into stages has been found to be useful in sleep research. However the inspection of

an E.E.G. record takes a considerable amount of a skilled encephalographer's time. Consequently Viglione (1970) has attempted to use pattern recognition to automate this process. This is the supervised learning problem. One possesses a number of waveforms which have already been classified and one attempts to use them to design a machine (or computer program) for classifying 'unknown' waveforms.

Firstly each waveform is sampled and digitised and a frequency analysis is performed. For this purpose the waveform is divided into 16.4 second intervals. The waveform in each interval is Fourier transformed by use of a computer algorithm called the Fast Fourier Transform (Tukey and Cooley, 1965). The resultant spectrum is represented as 1024 frequency components covering the band from zero to 62.5 Hz. The quantity of information is then reduced by an averaging of approximately three adjacent values. This gives 312 frequency components covering the same range. Of these, the first 130 are used in the pattern recognition algorithm. This represents the range of interest (zero to 26 Hz.). Viglione then uses a technique called DAID (Discriminant Analysis-Iterative Design) to eliminate those descriptors which contribute little to classifying the waveform and to determine the discriminate boundaries in the resultant sub-space. The resultant classifier was then tested on waveforms some of which were not present in the training set. The results obtained seem quite promising.

The problem under consideration in this thesis is rather more radical than that discussed by Viglione. He was content to accept the already existing classification of the sleep E.E.G. and merely automate the discrimination process. The question here is whether cluster analysis can be used to provide a significant classification of the sleep E.E.G. with no prior knowledge of the classification used by clinicians.

Such a classification may vindicate the previously used classification. Alternatively it might be a radically different classification which would have to be judged on the basis of how it helps the clinician and neurophysiologist to formulate new hypothesis about the functioning of the brain.

7.5 Normalised Slope Descriptors

In order to avoid the cost of computing a Fourier transform, Hjorth (1970) has suggested a set of waveform descriptors called normalised slope descriptors. Once again the E.E.G. waveform is divided into intervals called epochs. However instead of Fourier transforming the waveform in each epoch, Hjorth defines his parameters in terms of the time domain. Hjorth's original paper describes three parameters: activity, mobility and complexity. However more recently he has re-named the third parameter 'form factor' and adopted the name complexity for a new parameter. All four parameters are present in the data set used in Chapter 8. The four parameters are:

Activity This is the mean power in the signal during any particular epoch. Assume the epoch under consideration runs from $t = 0$ to $t = T$. Let $f(t)$ represent the signal. Then the activity is defined as

$$\frac{1}{T}\int_0^T f^2(t)\,dt$$

Mobility Whilst the activity is a measure of the amplitude of the signal, the activity is a measure of its variability. The definition is

$$\sqrt{\frac{\int_0^T \left(\frac{df}{dt}\right)^2 dt}{\int_0^T f^2 \; dt}}$$

As a consequence of this definition mobility has units of frequency. Furthermore, if a signal is linearly amplified the mobility remains unchanged.

Form Factor This is the parameter which was originally referred to as complexity by Hjorth. It measures how rapidly the slope varies. It is defined as

$$\sqrt{\frac{\int_0^T \left(\frac{d^2f}{dt^2}\right)^2 dt \Big/ \int_0^T \left(\frac{df}{dt}\right)^2 dt}{\int_0^T \left(\frac{df}{dt}\right)^2 dt \Big/ \int_0^T f^2 dt}}$$

This is, in fact, the mobility of the first derivative of f divided by the mobility of f itself. It is dimensionless. Like the mobility it is not altered by linear amplification. In addition it is unchanged by a linear transformation of the time scale, i.e. a transformation of the form

$$t' = A\ t$$

where A is some constant. For a sine wave it takes its minimum value of unity. All other continuous real signals give larger values.

Complexity This parameter also measures the manner in which the slope varies. It is defined as

$$\sqrt{\frac{\int_0^T \left(\frac{d^2f}{dt^2}\right)^2 dt}{\int_0^T \left(\frac{df}{dt}\right)^2 dt} - \frac{\int_0^T \left(\frac{df}{dt}\right)^2 dt}{\int_0^T f^2 dt}}$$

Complexity has units of frequency. It is unchanged by linear amplification. Like the form factor it takes its minimum value, zero, for a

sine wave. After some algebra it becomes apparent that mobility, form factor, and complexity are related by

$$\text{complexity} = \text{mobility} \times \left((\text{form factor})^2 - 1 \right)^{\frac{1}{2}}$$

Clearly the three parameters mobility, form factor and complexity are not independent. Given any two of these the third can be calculated. Figure 7.1, adapted from Hjorth (1970), illustrates the significance of the four normalised slope descriptors.

7.6 The Relationship between the Normalised Slope Descriptors and the Power Spectrum

Although defined and computed in the time domain, the normalised slope descriptors have an interesting interpretation in terms of the power spectrum. Let $F(\omega)$ represent the Fourier transform of a function identical to $f(t)$ in the interval $[0,T]$ and zero elsewhere. I.e.

$$F(\omega) = \int_0^T f(t) e^{-j\omega t} \, dt \qquad \text{where } j = \sqrt{-1}$$

Then the energy spectrum will be

$$F(\omega) \, F^*(\omega)$$

The superscript * denotes complex conjugation. The power spectrum, represented by $S(\omega)$, can be obtained by dividing by the length of the interval, T. I.e.

$$S(\omega) = F(\omega) \, F^*(\omega)/T$$

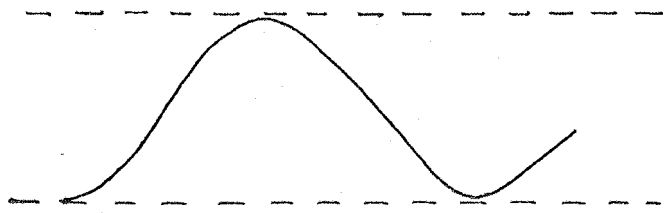
Furthermore the n'th moment of the power spectrum is defined by

$$m_n = \int_{-\infty}^{+\infty} \omega^n S(\omega) \, d\omega$$

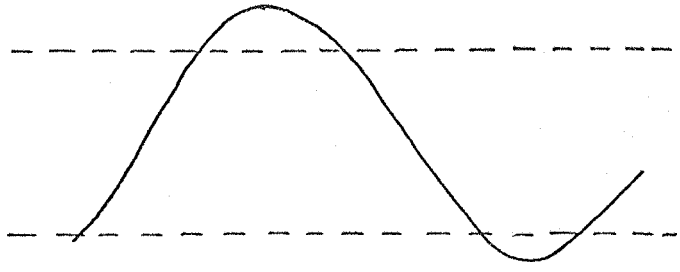
As a result the activity is equivalent to m_o. For

$$\begin{aligned} \text{activity} &= \frac{1}{T} \int_0^T f^2 \, dt \\ &= \frac{1}{T} \int_{-\infty}^{+\infty} F(\omega) \, F^*(\omega) \, d\omega \qquad \text{(Parseval's Theorem)} \\ &= \int_{-\infty}^{+\infty} S(\omega) \, d\omega \\ &= m_o \end{aligned}$$

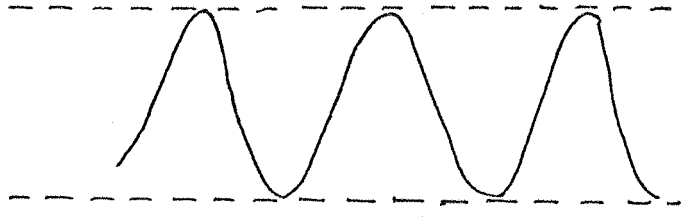
Now the Fourier transform of $\frac{df}{dt}$ is $\omega F(\omega)$. Consequently the power spectrum of the derivative of f is given by



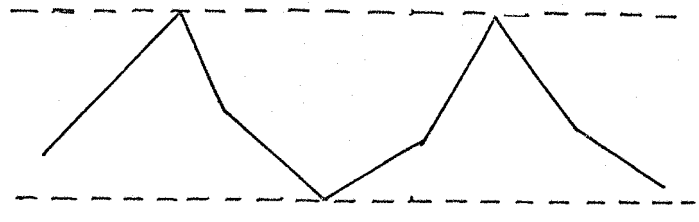
Arbitrary Reference



Increased Activity



Increased Mobility



Increased Form Factor and Complexity

FIGURE 7.1

$$\omega F(\omega).\omega F^{*}(\omega) = \omega^2 S(\omega)$$

Applying Parseval's Theorem gives

$$\frac{1}{T}\int_0^T \left(\frac{df}{dt}\right)^2 dt = \int_{-\infty}^{\infty} \omega^2 S(\omega)\, d\omega$$

$$= m_2$$

Consequently the mobility is equivalent to

$$\sqrt{\frac{\int_0^T \left(\frac{df}{dt}\right)^2 dt}{\int_0^T f^2\, dt}} = \sqrt{\frac{m_2}{m_0}}$$

The Fourier transform of $\frac{d^2f}{dt^2}$ is $\omega^2 F(\omega)$. Consequently the power spectrum of $\frac{d^2f}{dt^2}$ is $\omega^4 S(\omega)$. Applying Parseval's theorem again gives

$$\frac{1}{T}\int_0^T \left(\frac{d^2f}{dt^2}\right)^2 dt = \int_{-\infty}^{\infty} \omega^4 S(\omega)\, d$$

$$= m_4$$

Therefore the form factor is equivalent to

$$\sqrt{\frac{\int_0^T \left(\frac{d^2f}{dt^2}\right)^2 dt \Big/ \int_0^T \left(\frac{df}{dt}\right)^2 dt}{\int_0^T \left(\frac{df}{dt}\right)^2 dt \Big/ \int_0^T f^2\, dt}} = \sqrt{\frac{m_4/m_2}{m_2/m_0}}$$

Similarly the complexity is

$$\sqrt{\frac{\int_0^T \left(\frac{d^2f}{dt^2}\right)^2 dt}{\int_0^T \left(\frac{df}{dt}\right)^2 dt} - \frac{\int_0^T \left(\frac{df}{dt}\right)^2 dt}{\int_0^T f^2\, dt}} = \sqrt{\frac{m_4}{m_2} - \frac{m_2}{m_0}}$$

The Fourier transform of a real signal is symmetric about the origin, i.e.

$$F(\omega) = F(-\omega)$$

Consequently the power spectrum is also symmetric about the origin,

$$S(\omega) = S(-\omega)$$

As a result all the odd order moments of the power spectrum are zero. Thus knowing the activity, mobility, and form factor or complexity is equivalent to knowing the zeroth to fifth moments of the power spectrum.

It is now a relatively simple matter to show that the form factor and complexity take their minimum values for the pure sine wave. Firstly,

$$(\text{mobility})^2 = \frac{\int_{-\infty}^{\infty} \omega^2 S(\omega)\, d\omega}{\int_{-\infty}^{\infty} S(\omega)\, d\omega}$$

$$= \int_{-\infty}^{\infty} \omega^2 \frac{S(\omega)}{\int_{-\infty}^{\infty} S(\omega')\, d\omega'}\, d\omega$$

(ω' is merely a dummy variable here)

$$= \int_{-\infty}^{\infty} \omega^2 \, p(\omega)\, d\omega$$

Here $p(\omega)$ is a density function whose integral is unity. Since $p(\omega)$ is symmetric about the origin the mean of the power spectrum is zero. Consequently the above expression is the variance of the power spectrum and the mobility is the standard deviation of the power spectrum, σ_1.

Now

$$\sqrt{\frac{\int_0^T \left(\frac{d^2 f}{dt^2}\right)^2 dt}{\int_0^T \left(\frac{df}{dt}\right)^2 dt}}$$

can be regarded as the mobility of $\frac{df}{dt}$, and hence as the standard

deviation of the power spectrum of $\frac{df}{dt}$, σ_2. Consequently

$$\text{form factor} = \frac{\sigma_2}{\sigma_1}$$

$$\text{complexity} = (\sigma_2^2 - \sigma_1^2)^{\frac{1}{2}}$$

When a signal is differentiated with respect to time, the proportion of high frequency components is increased. Consequently the standard deviation of the power spectrum of $\frac{df}{dt}$ is greater than the standard deviation of the power spectrum of f, except when only one frequency is present in the spectrum. It is for this reason that the form factor and complexity take their minimum values for a pure sine wave.

7.7 Details of the Data Set Used

The data set used in the next chapter is eight dimensional. The first four descriptors are the activity, mobility, form factor and complexity for the E.E.G. of a sleeping human. The waveform measured is the potential difference between an electrode attached to the vertex of the head and an electrode attached to a point on the mid-line at the back of the head. The second four descriptors are the normalised slope descriptors for a waveform representing the potential difference between a point to the left of the left eye and a point to the right of the right eye. This waveform contains some E.E.G. signal. However mostly it is determined by the movement of the eyeballs. There is a potential difference of about 100 mV between the aqueous and vitreous humours of the eyeball. A movement of the eyeballs causes a change in potential field that will affect electrodes in their vicinity. The use of these four parameters is an attempt to include the same information as is available to a clinician when discriminating between paradoxical and orthodox sleep.

In all there are 2119 vectors. Since the epoch length is ten seconds this represents about six hours of sleep. The measuring instrumentation has upper and lower half-power points at about 17.5 Hz. and 2 Hz. respectively. Prior to any other computation being performed, the data set was normalised to zero mean and unit variance in each dimension.

CHAPTER 8 AN EXPERIMENTAL COMPARISON

8.1 Introduction

This chapter describes an experimental comparison of some of the techniques discussed previously. Once again it must be stressed that the primary objective of the work was to gain insight into the nature of the clustering techniques, rather than to reach any new conclusions about the sleep E.E.G. The clustering techniques are being compared as techniques for finding a true typology. From what has been said in the past chapters it will be clear that the choice of an algorithm for data reduction is not very critical. There are a number of algorithms that will do the job quite satisfactorily. The determination of a true typology is a much harder problem. However the most important aspect of this comparison is probably the information about execution times that it has produced. The theoretical arguments of the preceding chapters often gave the form of the relationship between execution time and number of data points, but these arguments were never sufficiently detailed to indicate the actual magnitude of execution time.

Most of the computation involved in this work was performed on the C.D.C. 7600 at the University of London Computing Centre (U.L.C.C.). Jobs were input to this machine via a telephone link from Southampton. The central processor for this machine has a 27.5 nanosecond clock period. The wordlength of the machine is 60 bits. The mainframe memory contains approximately 280 K words. However only about 124 K words of mainframe storage are available to an individual program, the rest being taken up by the operating system, etc. In addition, a small amount of computation has been performed on the I.C.L. 1907 at the University of Southampton Computing Centre. This is a 24 bit wordlength machine. Depending upon the structure of the program, it is between 40 and 80 times slower than the C.D.C. 7600.

8.2 Single Link Cluster Analysis Using the Minimal Spanning Tree

The M.S.T. of the 2119 vectors in the data set was computed on the C.D.C. 7600. The squared Euclidean distance between each pair of data points was used as the dissimilarity coefficient. A program to compute the M.S.T. is available at the U.L.C.C. It was modified slightly by the author to accommodate the large E.E.G. data set. The program is a FORTRAN version of the Algorithm of Ross (1969), which uses the method of Prim (1957). Approximately 66 seconds were taken to compute the M.S.T. and output a description of it on to punched cards.

The M.S.T. was used to perform a single link cluster analysis for a number of values of the threshold (h). Because of the expense of sending the information describing the M.S.T. along the telephone link, the I.C.L. 1907 was used for this part of the computation. Table 8.1 shows the number of clusters obtained, plus the number of points in the largest four clusters, for each value of h. It can be seen that for large values of h, one large cluster was obtained plus a number of smaller clusters. This situation continued down to h equal to 0.13. For h equal to 0.12, however, there were two moderately large clusters. Since the larger of these two contained only about 15% of the data points, most of the points fell in the small clusters. For h equal to 0.11 there were three moderately large clusters, but most of the points were outside these three. The time taken on the I.C.L. 1907 to perform the analyses for all the values of h shown from 0.9 to 0.2 was approximately 90 seconds. The time taken to perform the analyses for the remaining values of h was about 65 seconds.

The failure of the data to fall neatly into a small number of large clusters seems to suggest a one-cluster situation. However it may be due to the presence of a small number of noise points between clusters.

8.3 The Algorithm of Jarvis and Patrick

The algorithm of Jarvis and Patrick was programmed, by the author, in

h	Number of Clusters	Number of points in Four Largest Clusters			
0.9	96	1986	10	9	5
0.8	109	1978	10	8	5
0.7	134	1871	90	4	3
0.6	158	1852	88	7	3
0.5	188	1829	83	7	3
0.4	240	1774	71	7	6
0.3	346	1661	37	17	6
0.2	595	1321	13	10	10
0.19	636	1267	13	10	10
0.18	678	1225	11	10	10
0.17	732	1161	9	7	7
0.16	781	1095	11	9	7
0.15	835	1036	11	11	8
0.14	907	926	19	16	11
0.13	993	753	35	15	4
0.12	1067	327	281	51	32
0.11	1172	177	148	112	32

TABLE 8.1

FORTRAN. The program consists of three subroutines. The first and last are relatively trivial. The first subroutine takes each data point and constructs a list of the k nearest neighbours, in order of proximity. The final subroutine merely counts the number of clusters, counts the number of points in each cluster, and then outputs this information. The second subroutine actually performs the cluster analysis.

Because the second subroutine is more complex than the other two, a flowchart is given for it in Figure 8.1. Although not showing the detailed steps of the algorithm, the flowchart illustrates the major factors which help to improve the computational efficiency of the subroutine. The first feature to note is that it is necessary to compare each data point only with those points contained in its nearest neighbour list. Secondly, one must be careful not to compare two points twice. For this reason the I'th point is compared only with those whose index is greater than I. Thirdly, there is nothing to be gained in comparing two points if they have already been co-clustered by having been both successfully compared with a third point. A further economy can be achieved if, for a given value of k, the subroutine is repeated for a number of values of k_T. Assume that each value of k_T is greater than the previous value. Then if two points have not been co-clustered for the previous value of k_T, they will not be co-clustered for the current value. Consequently there is no use in comparing them. The I'th element of the array LABEL used in the flowchart contains the minimum of the indices referencing all the points which have so far been co-clustered with the I'th point. Consequently, when the I'th and J'th points are successfully compared, all those elements of LABEL containing the maximum of LABEL (I) and LABEL (J) are changed so as to contain the minimum of these two quantities. Thus the array LABEL keeps a check on which points have so far been co-clustered.

The program was used with four different values of k. For each value of k several values of k_T were used. The results are shown in Tables 8.2

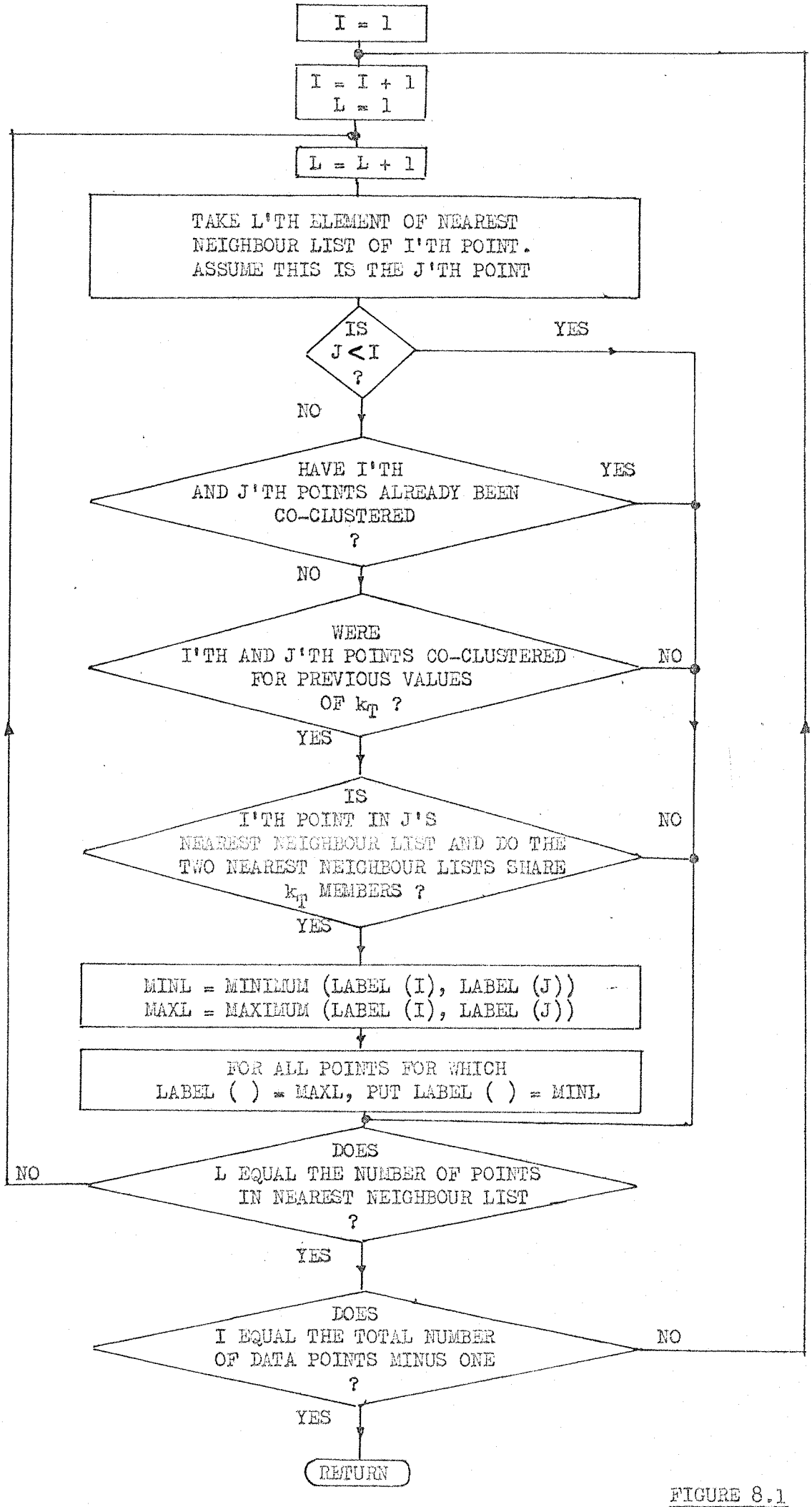

I = 1
I = I + 1
L = 1
L = L + 1
TAKE L'TH ELEMENT OF NEAREST NEIGHBOUR LIST OF I'TH POINT. ASSUME THIS IS THE J'TH POINT
IS J < I ?
YES
NO
HAVE I'TH AND J'TH POINTS ALREADY BEEN CO-CLUSTERED ?
YES
NO
WERE I'TH AND J'TH POINTS CO-CLUSTERED FOR PREVIOUS VALUES OF k_T ?
NO
YES
IS I'TH POINT IN J'S NEAREST NEIGHBOUR LIST AND DO THE TWO NEAREST NEIGHBOUR LISTS SHARE k_T MEMBERS ?
NO
YES
MINL = MINIMUM (LABEL (I), LABEL (J))
MAXL = MAXIMUM (LABEL (I), LABEL (J))
FOR ALL POINTS FOR WHICH LABEL ( ) = MAXL, PUT LABEL ( ) = MINL
DOES L EQUAL THE NUMBER OF POINTS IN NEAREST NEIGHBOUR LIST ?
NO
YES
DOES I EQUAL THE TOTAL NUMBER OF DATA POINTS MINUS ONE ?
NO
YES
RETURN

FIGURE 8.1

and 8.3. For each value of k and k_T the number of clusters is given plus the number of points in the four largest clusters. As can be seen the results are very similar to those for single link analysis described in the last section. For small values of k_T there is one large cluster plus a number of very small clusters. Then when k_T reaches a certain size, a large number of relatively small clusters appear. As with single link analysis the results seem to suggest that the data forms one cluster. The time taken to analyse the date for k = 20, k_T = 0 to 13 was 130 seconds on the C.D.C. 7600. Thus the amount of computation time required is approximately twice that needed to compute the M.S.T.

8.4 MICKA

MICKA is the name given by McRae (1971) to a FORTRAN program he has written which attempts to optimize one of four criteria. These criteria are (in the notation of Chapter 3) :

(1) tr W (to be minimized)

(2) $|W|$ (to be minimized)

(3) Largest eigenvalue of $W^{-1}B$ (to be maximized)

(4) tr $(W^{-1}B)$ (to be maximized)

The optimization algorithm is rather like that suggested by Friedman and Rubin (1967) and described in Chapter 3 of this thesis. The program provides a choice of three distance measures. These are :

(1) Squared Euclidean distance

(2) Weighted Euclidean distance. Let X_i and Y_i be the i'th co-ordinates in a q-dimensional space of the vectors $\underline{X}$ and $\underline{Y}$. Let σ_i be an estimate of the standard deviation in the i'th dimension. Then the weighted Euclidean distance between $\underline{X}$ and $\underline{Y}$ is given by :

$$d(\underline{X},\underline{Y}) = \sum_{i=1}^{q} \frac{(X_i - Y_i)^2}{\sigma_i^2}$$

(3) Mahalanobis distance.

k	k_T	Number of Clusters	Number of Points in Four Largest Clusters			
5	0	111	1814	91	22	15
5	1	188	1499	81	41	27
5	2	794	22	16	16	16
10	0	31	2085	2	2	2
10	1	32	2084	2	2	2
10	2	36	2079	2	2	2
10	3	55	1915	106	26	12
10	4	122	1822	105	18	11
10	5	452	96	95	83	79
15	0	15	2105	1	1	1
15	1	15	2105	1	1	1
15	2	15	2105	1	1	1
15	3	16	2104	1	1	1
15	4	18	2102	1	1	1
15	5	24	2094	2	2	1
15	6	38	2068	11	2	2
15	7	102	1863	93	26	17
15	8	278	1111	282	56	45
15	9	642	88	80	77	51

TABLE 8.2

k	k_T	Number of Clusters	Number of Points in Four Largest Clusters			
20	0	9	2111	1	1	1
20	1	9	2111	1	1	1
20	2	9	2111	1	1	1
20	3	9	2110	1	1	1
20	4	10	2110	1	1	1
20	5	11	2109	1	1	1
20	6	12	2108	1	1	1
20	7	13	2106	2	1	1
20	8	18	2101	2	1	1
20	9	36	1970	110	2	2
20	10	79	1913	110	7	5
20	11	185	1962	109	33	21
20	12	406	191	177	161	130
20	13	743	79	63	63	56

TABLE 8.3

McRae's program is available on the C.D.C. 7600 at the U.L.C.C. As with the M.S.T. program it was necessary for the author to modify it slightly to cope with the large E.E.G. data set.

8.4.1 Tr W

The program was run using tr W as the criterion. Squared Euclidean distance was used. As explained in Chapter 3 Euclidean distance (or squared Euclidean distance) is the best distance measure to use when attempting to minimize this criterion. About 1400 seconds were needed on the C.D.C. 7600 to try to find the optimum partition for g (the number of clusters) equal to 2 to 14. Figure 8.2 shows the minimum value of the criterion for each value of g. The value for $g = 1$ is simply the trace of the total scatter matrix (i.e. tr S). As can be seen the graph is alarmingly smooth and there is no discontinuity to suggest a definite number of clusters. This is presumably because the data does not fall into compact spherical clusters.

8.4.2 |W|

The program was next run using |W| as the criterion. Mahalanobis distance was used for the reason explained in section 3.4. This time about 1600 seconds were needed to try to find the optimum partitions for $g = 2$ to $g = 13$. Figure 8.3 shows minimum |W| for the various values of g. Figure 8.4 shows $\log_{10}$ (|S|/min |W|), as suggested by Friedman and Rubin. As can be seen neither graph has any discontinuity. Figure 8.5 shows $g^2 \min|W|$. It will be recalled that Marriott suggested looking for the minimum value of $g^2 \min|W|$ to find the correct value for g (see section 3.7). However, as can be seen, $g^2 \min|W|$ is almost a monotonically decreasing function of g and certainly seems to be tending to zero as g increases. The rather high values of $g^2 \min|W|$ for $g = 2$ and $g = 12$ probably occur because the true optimum partition has not been found.

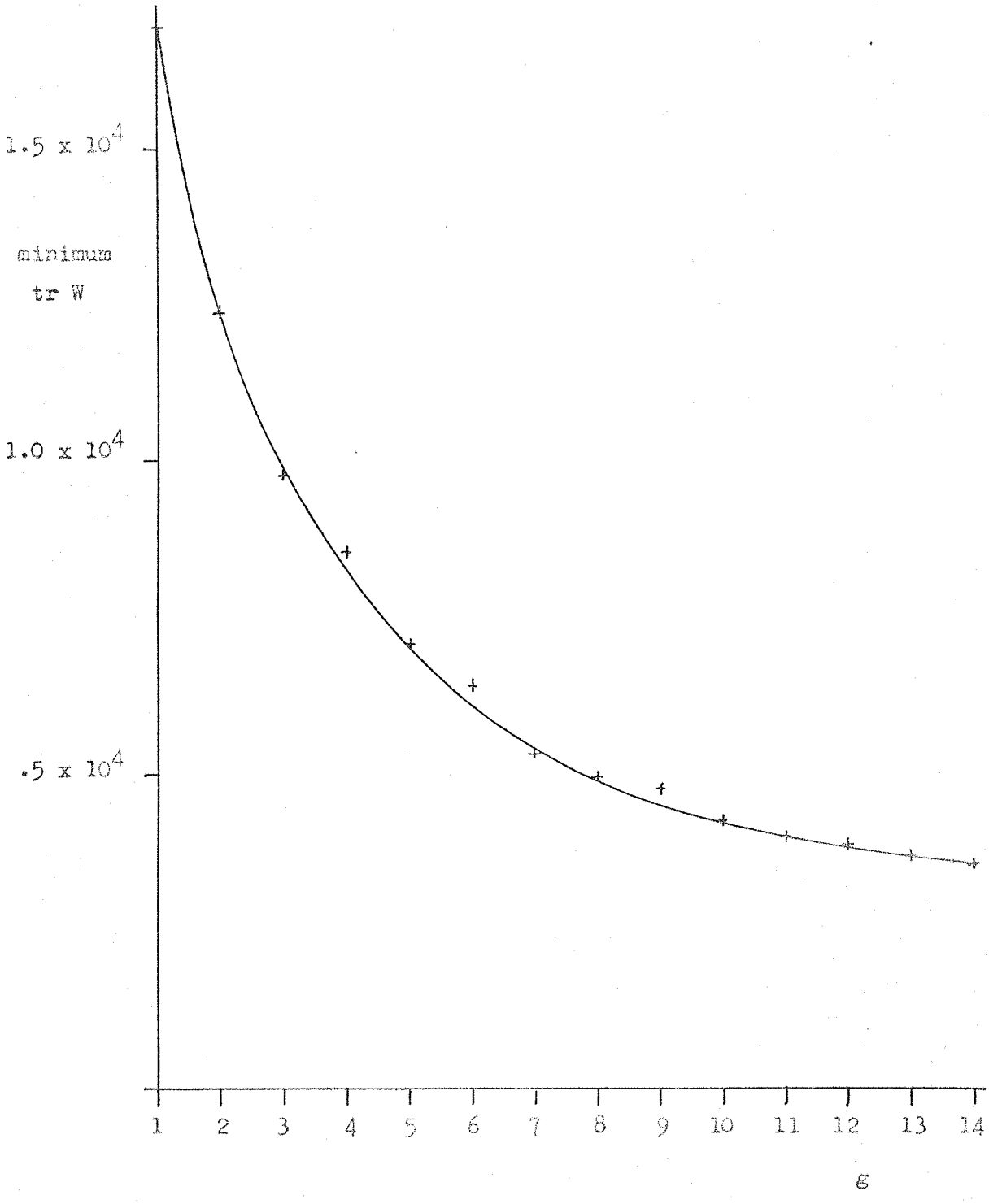


FIGURE 8.2

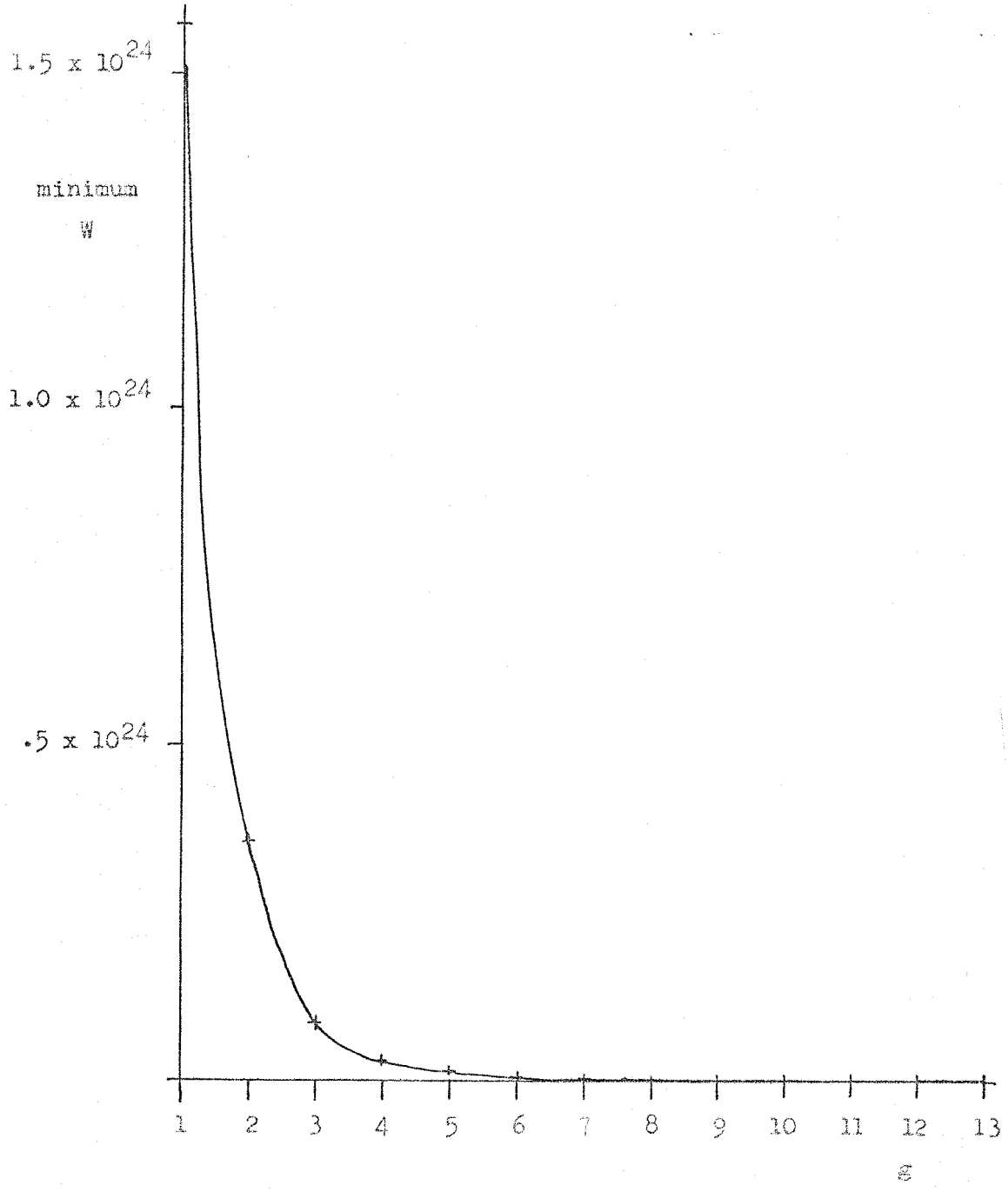


FIGURE 8.3

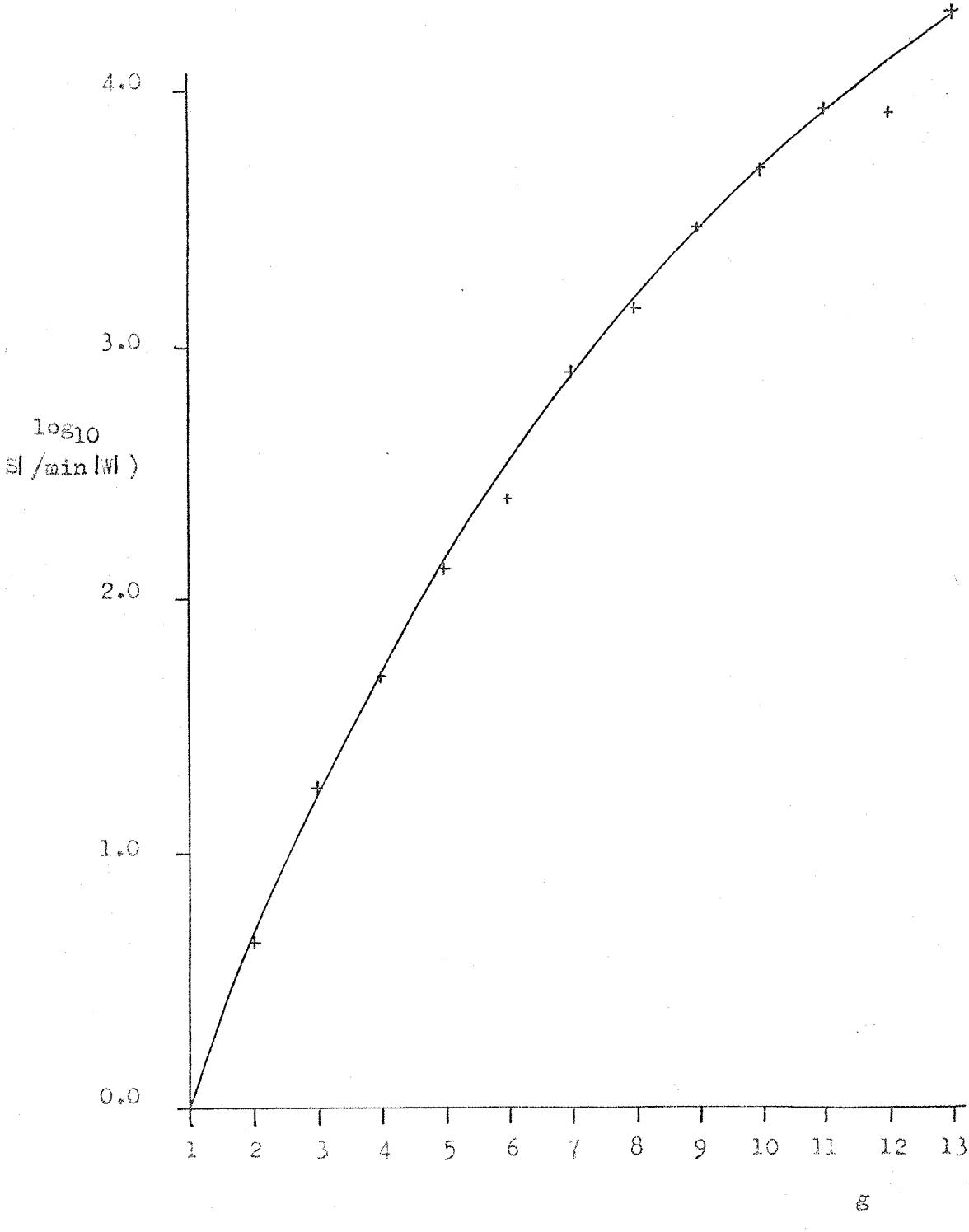


FIGURE 8.4

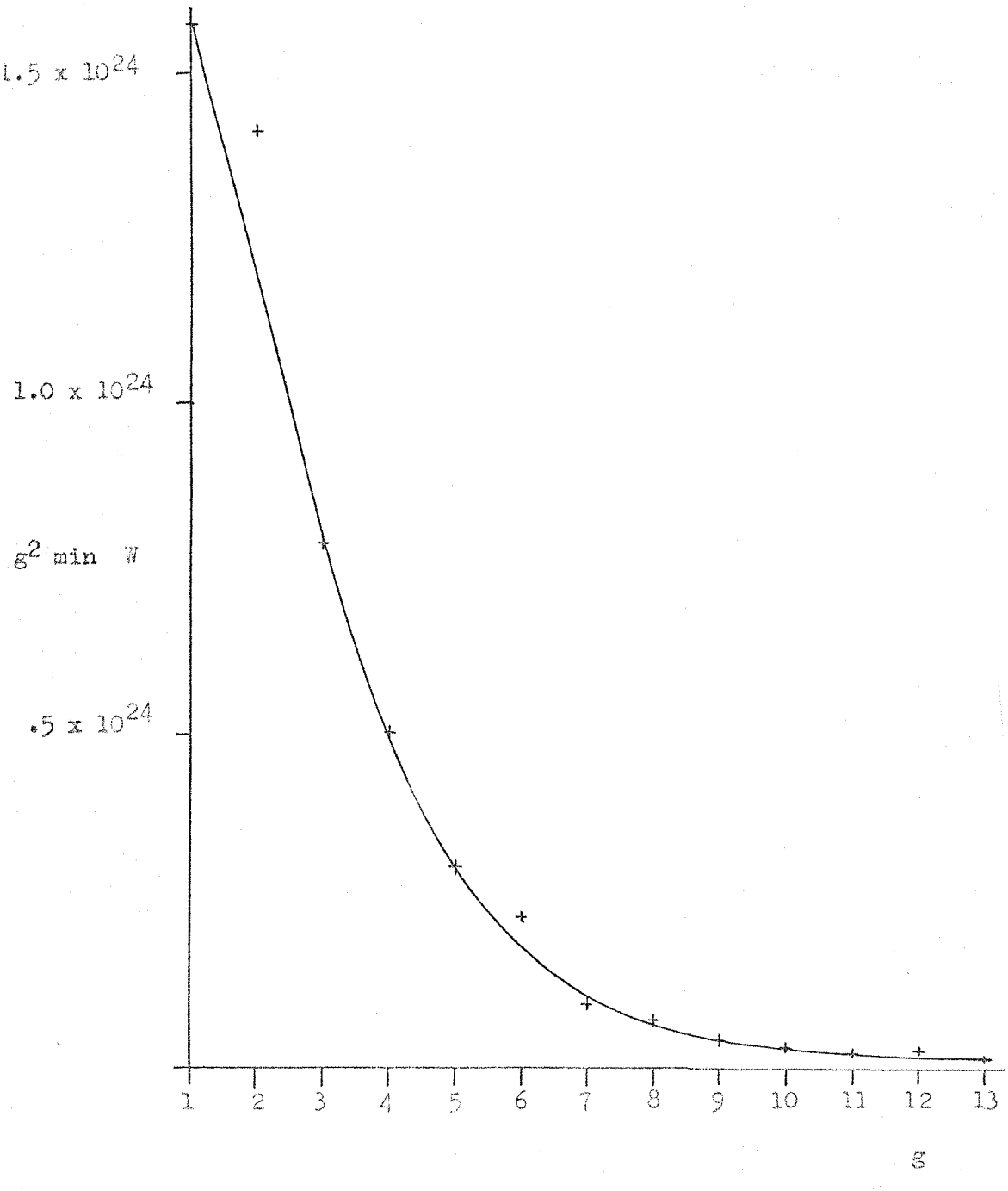


FIGURE 8.5

This is also reflected in the rather low value of $\log_{10}(|S|/\min|W|)$ for $g = 12$. Once again no obvious clustering has been revealed.

8.4.3 Largest eigenvalue of $W^{-1}B$

Mahalanobis distance was used when attempting to maximize this criterion. To the author's knowledge this distance measure does not possess the same optimal quality when used with this criterion as it does when used to minimize W. However when $g = 2$ the two criteria become identical, as Fukunaga and Koontz (1970) have shown. Consequently Mahalanobis distance is certainly the right distance measure to use in that one case. MICKA took about 20 minutes on the C.D.C. 7600 to try to find the optimum partition for $g = 2$ to $g = 5$. To find the optimum partition for 6 clusters took of the order of 10 minutes. Similar times were taken for $g = 7$ and $g = 8$. The maximum value of the criterion achieved is shown in Figure 8.6 for each value of g. The value for $g = 1$ is equated to zero. Although the matrix B is not really defined for this value of g, if it is taken to be the zero matrix this preserves the equality $T = W + B$, since $T = W$ for $g = 1$. The 'eigenvalues' of the zero matrix can be taken to be zero, since when this matrix is pre-multiplied into any column vector the result is the zero vector. Once again the curve is remarkably smooth and gives no evidence of a definite clustering.

8.4.4 $\mathrm{Tr}(W^{-1}B)$

Mahalanobis distance was used when attempting to maximize this criterion. As with the criterion of 8.4.3 this distance measure is, to the author's knowledge, optimal only when $g = 2$. The results for $g = 2$ to $g = 11$ are shown in Figure 8.7. For reasons similar to those given in section 8.4.3, the criterion was equated to zero for $g = 1$. The program took 860 seconds to try to find the optimum value of the criterion for $g = 2$ to $g = 6$. For each of the remaining values of g approximately 10 minutes

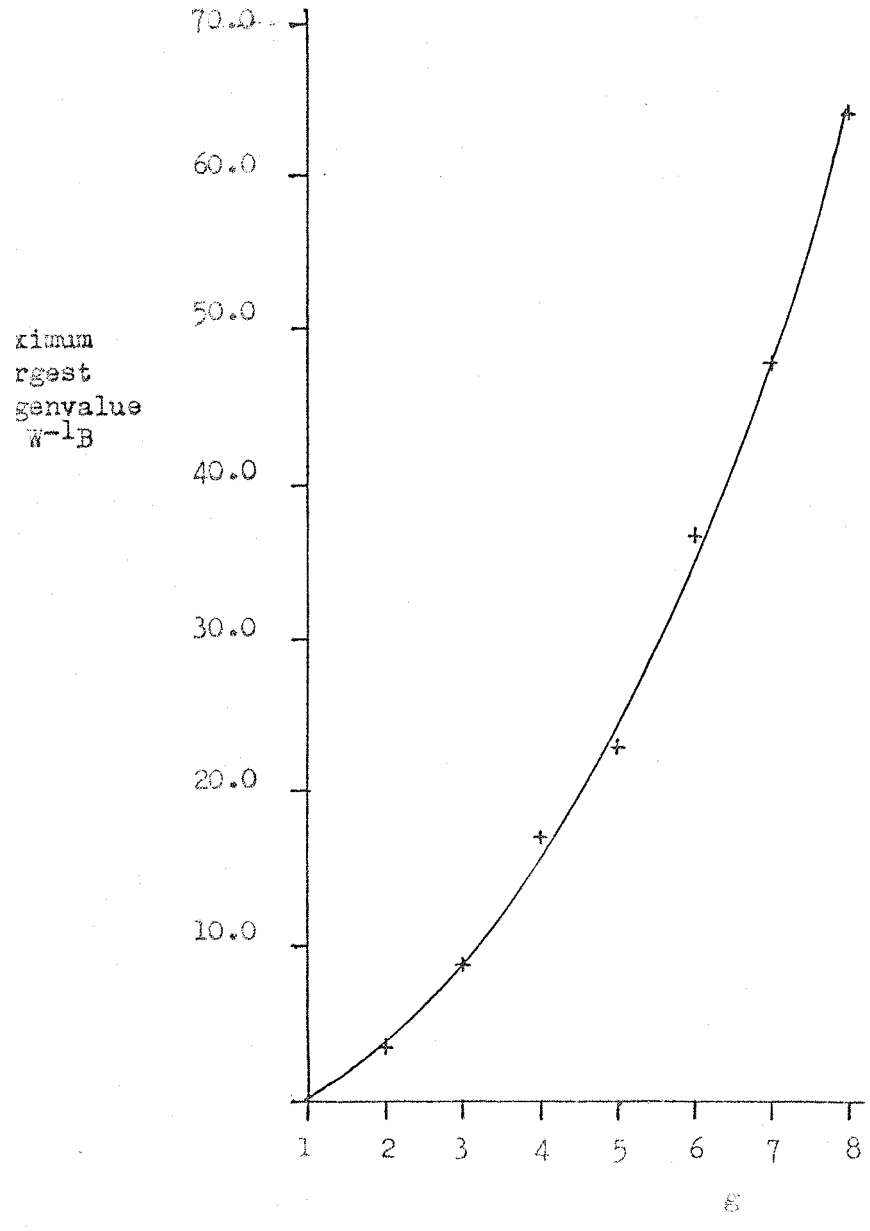

70.0
60.0
50.0
40.0
30.0
20.0
10.0
ximum
rgest
genvalue
$W^{-1}B$
1
2
3
4
5
6
7
8
s

FIGURE 8.6

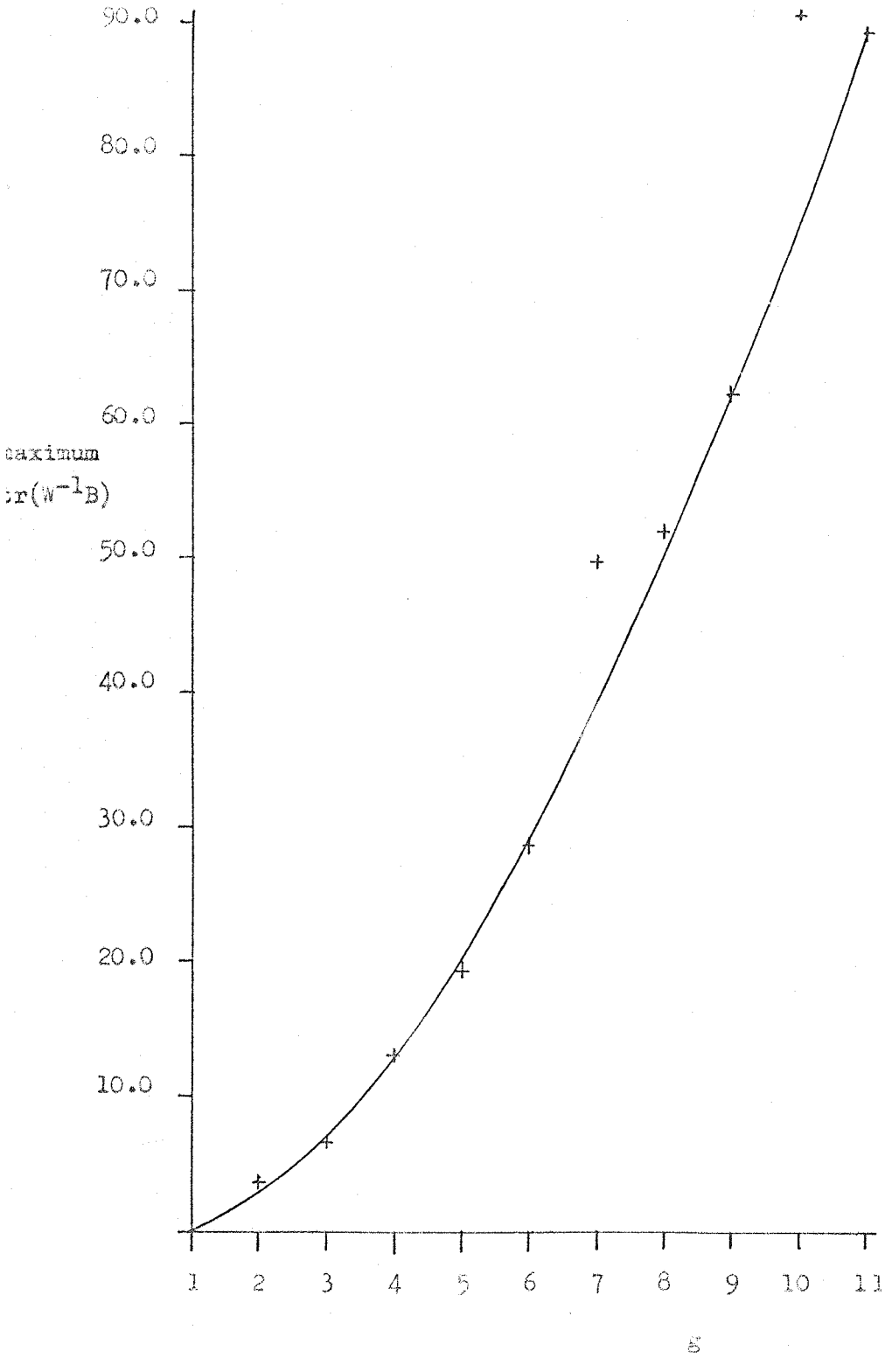

90.0
80.0
70.0
60.0
aximum
:r(W^-1B)
50.0
40.0
30.0
20.0
10.0
1
2
3
4
5
6
7
8
9
10
11

FIGURE 8.7

were needed.

It is difficult to know how to interpret these results. The curve as drawn in Figure 8.7 suggests a 7- cluster or 10- cluster solution. However it may be that the criterion values obtained for g = 8, 9 and 11 are not the true maximum values. It could be that, if the true values were known, all the points would lie on a smooth curve. This illustrates a difficulty which is always present in interpreting this kind of graph. How can one strictly define the shape of graph which would indicate a definite cluster structure? In addition the author is reluctant to place any faith in a cluster structure which is revealed by one technique alone and which is not apparent in any other set of results. Investigation of the 7 and 10 - cluster structure also revealed that, as the waveform varied with time, the majority of periods for which the waveform remained in one cluster were of length only one epoch, i.e. 10 seconds. Clearly to validate the hypothesis that the waveform changes its nature as frequently as every ten seconds it would be necessary to estimate the descriptors over a period much less than ten seconds. It might, indeed, be interesting to re-compute a set of descriptors on the basis of a much smaller epoch, and then analyze this new data set with the clustering techniques. However, visual inspection implies that the waveform changes much more slowly than every ten seconds. Since the information present on visually inspecting the waveform is much greater than that in the Hjorth parameters, it would be difficult to sustain an interpretation of the sleep E.E.G. so radically different from clinicians' interpretation of the waveform.

8.5 FUZZY

The program FUZZY is also available on the C.D.C. 7600 at the U.L.C.C. Once again minor modifications were necessary to accommodate the very large data set. As explained in section 4.2 the program seeks to place the

cluster boundaries along the valleys of the p.d.f. In order to do this the p.d.f. at each data point is estimated. This is done by counting the number of data points within a hypersphere of radius $T^{\frac{1}{2}}$ centred on each data point. Consequently before the program can be used it is necessary for the user to decide on a suitable value for T. T must be sufficiently large that most of the hyperspheres contain enough data points to confidently estimate the p.d.f. at the centre of the hypersphere. On the other hand, if T is too large, the p.d.f. will vary considerably over the volume of each hypersphere. If either of these two conditions occur the results of the cluster analysis will be invalid. Between these extremes there may be a range of values for T which give the same cluster structure. This structure is then assumed to be the actual structure.

The results of using this program with the E.E.G. data set are shown in Table 8.4. For $T^{\frac{1}{2}} = 0.1$ only 10 of the hyperspheres contained data points other than the central data point. Even in each of these 10 hyperspheres there was only one other data point. Clearly this value of T is too small. For $T^{\frac{1}{2}} = 0.33015$ the situation was very different. There were 5 hyperspheres containing 11 data points each. Many other hyperspheres contained more than one data point. The program partitioned the data set into very many clusters. Similar results were obtained for the other values of T shown. Leaving aside the first result which, as explained, is not significant, there appears to be no simple cluster structure. From the large number of clusters it was not possible to pick out even a small number of very large clusters. A surprising thing about the results is that, as T increases, the number of clusters does not increase monotonically to a maximum and then decrease monotonically, but actually oscillates. Even if the results from one particular choice of T could be regarded as more significant than the other results, it would be difficult to believe that the physical process under investigation

$T^{\frac{1}{2}}$	Number of Clusters	Maximum number of points in a hypersphere of radius $T^{\frac{1}{2}}$ and centred on a data point.
0.1	4	2
0.33015	54	11
1.0	47	220
2.0	41	901
3.0	60	1626
4.0	57	1913

TABLE 8.4

(i.e. the generation of the E.E.G.) can usefully be divided into so many categories.

The execution time for this program varies with T. For $T^{\frac{1}{2}} = 0.1$ it was approximately 400 seconds. For the other values of T it was rather less than 200 seconds.

8.6 Nonlinear Mappings

Both Sammon's program and the program of Chang and Lee are available on the C.D.C. 7600 at U.L.C.C. Neither of these programs could easily be modified to accommodate the very large E.E.G. data set. Consequently the data was pre-clustered to reduce it to 250 points in 8- space. This is the data -reduction situation. MAXIMINDIST was used, followed by two 'reassignment' passes as defined in section 3.3. The whole process took approximately 1000 seconds on the C.D.C. 7600. In retrospect this seems a very wasteful computation. As explained in section 6.2, when the sample being taken is as large as 250 points out of 2119, the reassignment passes themselves will almost certainly suffice to produce a representative sample.

8.6.1 Sammon's Program *

The 250 8- dimensional points produced by the pre-clustering phase described above were mapped into 250 2- dimensional points by Sammon's program. 99 iterations were used and the program took approximately 170 seconds on the C.D.C. 7600. In fact the program had practically converged after 52 iterations. For after the 52nd iteration the mapping error was 0.031 and it remained at this value after all subsequent iterations. The resultant 2- space configuration is shown in Figure 8.8. At first the author felt that this plot might suggest a 3- cluster structure, as shown by the dotted lines. However further investigation revealed that the

---

* This program was developed at the Rome Air Development Center, Grifiss AFB, Rome, New York.

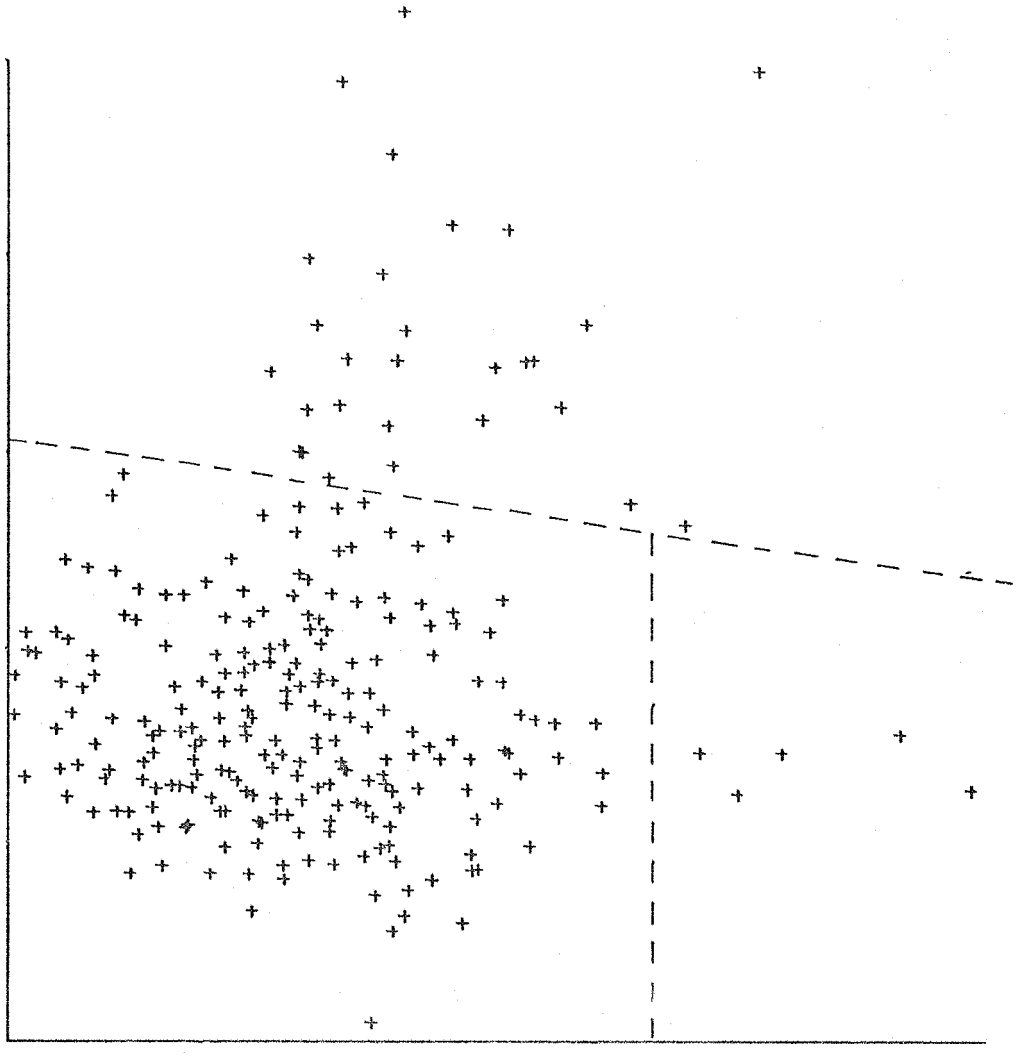


FIGURE 8.8

5 points in the smallest 'cluster' represented only 6 of the original data points. The 28 points in the intermediate cluster represented only 30 of the original data points. Thus the great majority of the original points are represented by the main cluster, whilst the other two clusters are too small to be regarded as anything but sets of outliers. This illustrates the danger in using a data - reduction technique in which each point in the reduced data set does not represent the same number of points in the original data set.

8.6.2 The Program of Chang and Lee

This program also used the 250 8- dimensional points produced by the data reduction program. The program was run with the first 100 points in the frame, and the result is shown in Figure 8.9. The program was also run with all 250 points in the frame. The resultant plot is shown in Figure 8.10. The first time approximately 28 seconds were needed for 50 iterations and the final mapping error was 0.046. The second time the program took approximately 75 seconds for 50 iterations and the final mapping error was 0.036. Thus, as might be expected, a more accurate mapping was obtained on the second occasion.

The results of these mappings are very similar to those obtained by Sammon's algorithm. Once again there appear to be 3 clusters present. However further investigation revealed that in both mappings the majority of points in the two smaller clusters represented only one of the original data points each. Consequently these points can only be regarded as outliers.

8.7 Conclusions

None of the techniques used here gives any conclusive evidence that the data can be divided into a sufficiently small number of clusters to be of value in understanding the E.E.G. However, as explained in Chapter 7,

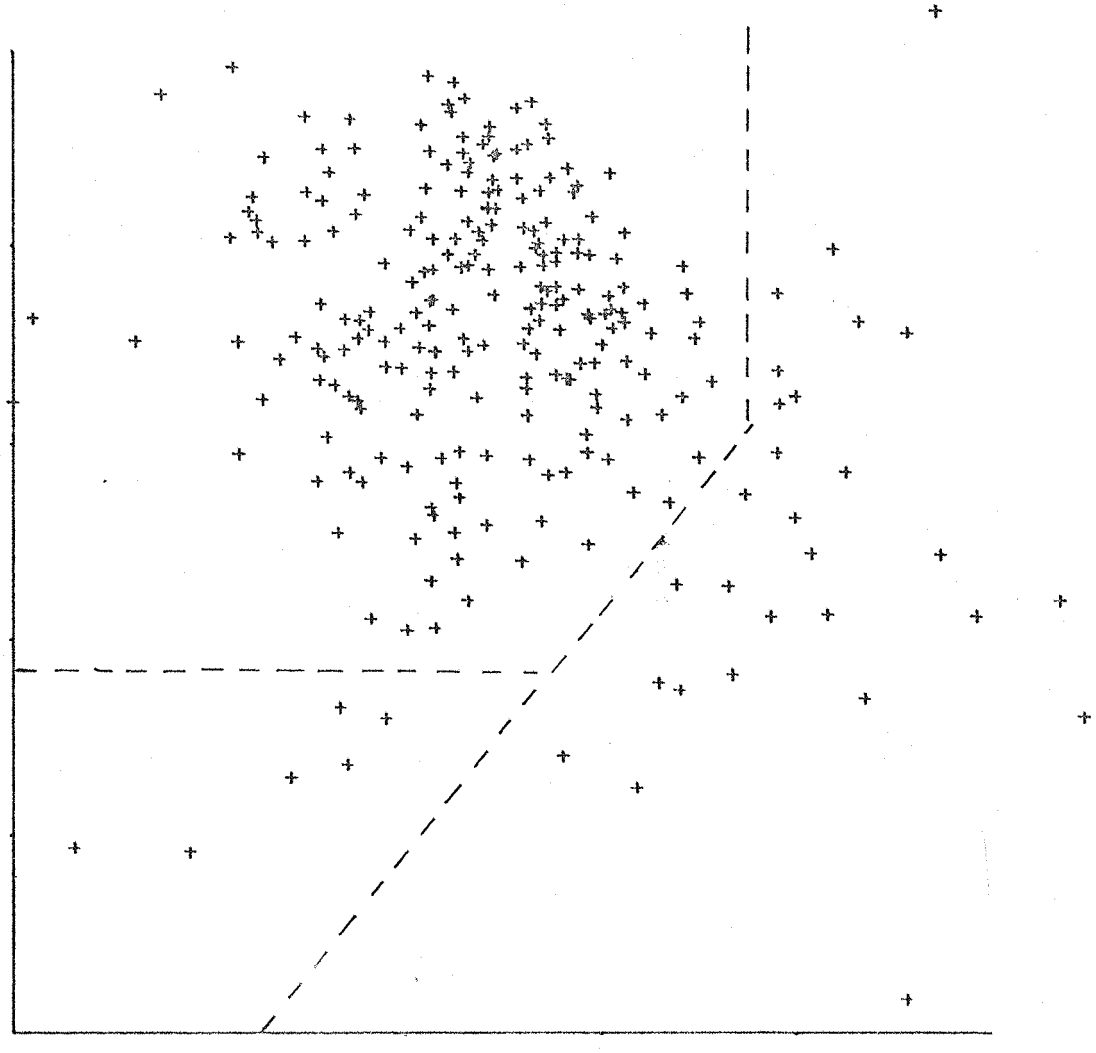


FIGURE 8.9

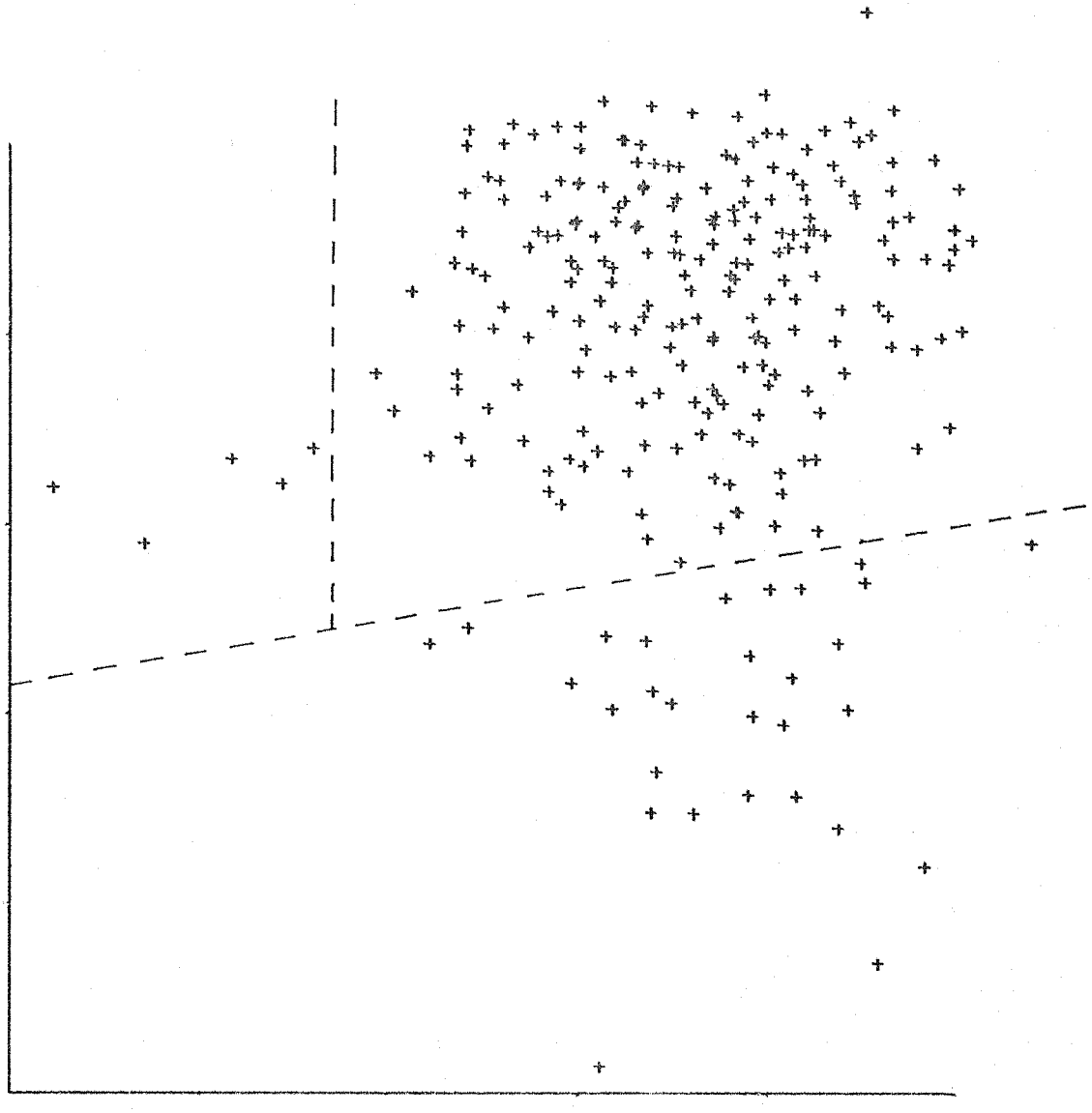


FIGURE 8.10

clinicians feel able, with the help of information about eye movement, to divide the sleep E.E.G. into five stages. Figure 8.11 shows how the period of sleep (of approximately 6 hours) represented by the data set, can be divided between the various stages. But the implication of this chapter is that the data occupies a 'continuous' region of the data space. There are essentially three ways of resolving this conflict. Firstly, the lack of apparent cluster structure may be due to the inadequacies of the clustering algorithms. However, despite the obvious failings of many of them, it seems difficult to believe that they can all be so bad as to miss any real cluster structure. Secondly, the boundaries between stages which exist intuitively in the mind of the clinician may have no real significance. It could be that the divisions between the various stages of NREM sleep are entirely arbitrary. However the clear distinction between the presence and absence of rapid eye movements ought to at least lead to a 2- cluster structure. Thirdly, and (to the author's mind) most likely, the Hjorth parameters may not contain enough of the relevant information. It would be interesting to repeat the analyses with a more comprehensive set of descriptors, such as those used by Viglione (see section 7.4).

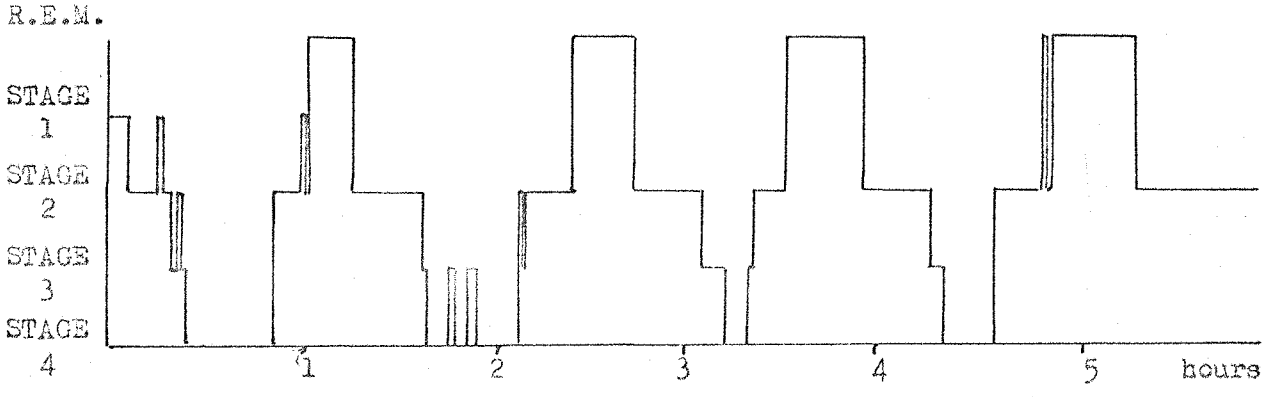


FIGURE 8.11

CHAPTER 9 CONCLUSIONS

9.1 Parametric Cluster Analysis

Three of the algorithms discussed in this thesis are truly parametric. NORMAP assumes that the p.d.f. of the population from which the data set is a sample is the sum of a number of normal density functions with equal covariance matrices. NORMIX makes similar assumptions but removes the restriction that the covariance matrices need be equal. Finally, MICKA, when used to minimize $|W|$, is specially suited to the case of several normal distributions with equal covariance matrices.

Because NORMAP and MICKA have the same objective it would be interesting to compare them. There are three principal questions to be answered. Firstly, how often do the programs give the correct results when used with data of known structure? Secondly, how do they compare from the standpoint of computational efficiency? Thirdly, how sensitive are their results to slight deviations from the assumptions of normality and equal covariance matrices? Unfortunately the author has not had time to investigate NORMAP. Wolfe himself admits that the program sometimes diverges and that initialisation is a problem. Consequently there may be room for improvement of the numerical technique used to solve the likelihood equations. It may be that NORMAP is more susceptible to a bad initialisation than is MICKA. One important problem is the determination of the optimum value for the number of clusters (g). More work could be done to see how effective is the $g^2\min|W|$ criterion of Marriott. However, it would appear that whatever the other advantages of MICKA, NORMAP has the advantage here since it leads to a definite statistical test to compare two hypotheses about the value of g. This degree of mathematical exactitude is a very rare thing in cluster analysis! It may be that when there is no possibility of making a reasonable guess at the initial parameter

values, MICKA is the program to use to generate approximate values for these parameters. NORMAP could then be initialised with these values and used to produce more accurate values and to give a more definite idea of the best value for g.

It would also be interesting to modify MICKA so as to use it to minimise the $\prod_{j=1}^{g} |W_j|^{N_j}$ criterion suggested by Scott and Symons. The modified program could then be compared with NORMIX in the same way as the original program, when used to minimize $|W|$, can be compared with NORMAP.

Another interesting question is whether the Sebestyen and Edie algorithm can be used successfully to separate Gaussian clusters. For reasons explained in Chapter 4 the author hesitates to call this a parametric algorithm. However it does seem most suited to dealing with the case in which the clusters have normal distributions. If it does work satisfactorily it will certainly be much faster than the other three algorithms discussed in this section.

In addition it might be useful to consider how these techniques could be altered to deal with density functions other than the Gaussian function. There are two reasons for the frequent assumption of normality in pattern recognition. Firstly, the Central Limit Theorem frequently permits this assumption. Secondly, it tends to simplify the mathematics! However, applications may arise in which other density functions are relevant. In the same paper in which NORMAP and NORMIX are described, Wolfe briefly refers to the 'Latent Class' model. This is applicable to binary - valued data. The assumption here is that the probability of a data vector occupying a given point in a q - dimensional data space can be represented as the sum of a number of functions of the form:

$$\alpha_s(\underline{x}, \underline{\mu}_s) = \prod_{i=1}^{q} \mu_{si}^{x_i} (1-\mu_{si})^{(1-x_i)} \qquad s = 1, 2, \ldots g.$$

Here the X_i are the descriptor values, which may be 0 or 1, whilst μ_{si} is the probability, for the s'th function of this form, that $X_i = 1$.

9.2 Nonparametric Cluster Analysis

The great bulk of clustering techniques do not make any definite assumptions about the statistical structure of the population from which the data is drawn. In addition, until quite recently, all such nonparametric techniques had only very vaguely defined objectives. Nonparametric clustering algorithms were designed intuitively and defined operationally, i.e. in terms of how they actually worked rather than what they were attempting to achieve. Such techniques are frequently satisfactory for data - reduction. But when the problem is to find a true typology, if one cannot make assumptions like those discussed in the last section, it is not at all apparent what criterion a 'good' partition should satisfy. Furthermore, because the algorithms are defined operationally, it is frequently not easy to predict how they would behave when used to analyse a sample from a known population. Because one cannot do this it is difficult to comprehend the significance of any results that the algorithms give when used on a sample from an unknown population. One can attempt to understand an algorithm's behaviour experimentally by testing it with artificial data of known structure. But, in the author's view, this leaves the user lacking confidence in the technique. There are always nagging questions. One can test the algorithm for only a few situations. What will happen for some other case? And how are the results dependent upon sample size? The only way to be confident of the results of cluster analysis is by understanding exactly what an algorithm will achieve in any given situation. To be sure of this one must either analyse fully an operationally defined algorithm or use an algorithm which is designed with a well-defined objective in mind. Sometimes the simpler, intuitive techniques may be useful in a preliminary investigation. Sometimes the

data may be so well clustered that almost any technique will illustrate this clustering. But to be fully confident of the results of an algorithm in all situations its properties must be fully understood.

To the knowledge of the author the only attempt made to define an objective in nonparametric cluster analysis has been made by Koontz and Fukunaga (1972 a and b). Their algorithm (the fixed neighbourhood penalty rule) partitions the clusters along the valleys of the probability density. It would be interesting to compare the fixed neighbourhood penalty rule with FUZZY. Although the objective of FUZZY is not clearly stated in Gitman and Levine's paper it appears to be the same as that of the fixed neighbourhood penalty rule. However FUZZY is quite a time-consuming procedure, as has been seen in the last Chapter. In addition it is not at all obvious which value of the control parameter (T) gives the most significant results. It is to be hoped that the fixed neighbourhood penalty rule will produce a faster algorithm and one with more definite results. There may, however, be simpler ways of finding the valleys of probability density.

In addition it might be possible to define different objectives for a nonparametric clustering algorithm. An attempt could then be made to design algorithms to achieve these objectives. Such algorithms ought to give results invariant under any change of scale. For what confidence can one have in a partition which is known to be dependent upon an arbitrary choice of scale? Furthermore, as the sample size increases, the algorithm ought to achieve the desired objective with a greater probability. This is a property analogous to consistency in the estimation of statistical parameters. To achieve the same kind of confidence in cluster analysis as one possesses in more conventional statistical techniques it seems essential to use algorithms satisfying these rigorous criteria.

REFERENCES

ANDERSON, T. (1958) : An introduction to multivariate statistical analysis. Wiley : New York.

BALL, G.H. (1965) : Data analysis in the social sciences: What about the details? Proceedings of the Fall Joint Computer Conferences, Stanford, pp. 533-559. New York : Macmillan.

BALL, G.H. and HALL, D.J. (1967) : A clustering technique for summarizing multivariate data. Behavioral Science, Vol. 12, pp. 153-155.

BATCHELOR, B.G. (1968) : Learning machines for pattern recognition. Ph.D. Thesis, University of Southampton.

BATCHELOR, B.G. and WILKINS, B.R. (1969) : Method for location of clusters of patterns to initialize a learning machine. Electronics Letters, Vol. 5, 1969, pp. 481-483.

CHANG, C.L. and LEE, R.C.T. (1973) : A heuristic relaxation method for nonlinear mapping in cluster analysis. I.E.E.E. Trans. on Systems, Man, and Cybernetics, March 1973, pp. 197-200.

COOPER, R., OSSELTON, J.W., and SHAW, J.C. (1969) : E.E.G. Technology. Butterworth : London.

DEMENT, W.C. and KLEITMAN, N. (1957) : Cyclic variations in E.E.G. during sleep and their relation to eye movements, body motility and dreaming. Electro-encephalography and Clinical Neurophysiology, 9, pp. 673-710.

EIGEN, D.J., FROMM, F.R., and NORTHOUSE, R.A. (1974) : Cluster analysis based on dimensional information with applications to feature selection and classification. I.E.E.E. Trans. on Systems, Man, and Cybernetics, Vol. SMC - 4, No. 3, May 1974.

EVERITT, B. (1974) : Cluster Analysis. Heinemann : London.

FISHER, R.A. (1922) : On the mathematical foundations of theoretical statistics. Phil. Trans. Roy. Soc. London. Series A, Vol. 222, pp. 309-368.

FISHER, R.A. (1936) : Multiple measurements in taxonomic problems.

FRIEDMAN, H.P. and RUBIN, J. (1967) : On some invariant criteria for grouping data. Amer. Stat. Assoc. J., Vol. 62, pp. 1159-1178, December 1967.

FUKUNAGA, K. and KOONTZ, W.L.G. (1970) : A criterion and an algorithm for grouping data. I.E.E.E. Trans. on Computers, Vol. C-19, No. 10, October 1970.

FUKUNAGA, K. (1972) : Introduction to statistical pattern recognition. Academic Press : New York.

GITMAN, I. and LEVINE, M.D. (1970) : An algorithm for detecting unimodal fuzzy sets and its application as a clustering technique. I.E.E.E. Trans. on Computers, Vol. C-19, pp. 583-593.

GOWER, J.C. (1967) : A comparison of some methods of cluster analysis. Biometrics, 23, pp. 623-628.

GOWER, J.C. and ROSS, G.J.S. (1969) : Minimum spanning trees and single linkage cluster analysis. Applied Statistics, Vol. 18, pp. 54-64.

HJORTH, B. (1970) : E.E.G. analysis based on time domain properties. Electroencephalography and Clinical Neurophysiology, Vol. 29, pp. 306-310.

HJORTH, B. (1973) : The physical significance of time domain descriptors in E.E.G. analysis. Electroencephalography and Clinical Neurophysiology, Vol. 34, pp. 321-325.

HOTELLING, H. (1933) : Analysis of a complex of statistical variables into principal components. J. Educ. Psychol., Vol. 24, pp. 417-441 and pp. 498-520.

JARVIS, R.A. and PATRICK, E.A. (1973) : Clustering using a similarity measure based on shared near neighbours. I.E.E.E. Trans. on Computers, Vol. C-22, No. 11, November 1973, pp. 1025-1034.

KOONTZ, W.L.G. and FUKUNAGA, K. (1972a) : A nonparametric valley-seeking technique for cluster analysis. I.E.E.E. Trans. on Computers, Vol. C-21, No.2, February 1972, pp. 171-178.

KOONTZ, W.L.G. and FUKUNAGA, K. (1972b) : Asymptotic analysis of a nonparametric clustering technique. I.E.E.E. Trans. on Computers, Vol. C-21, No. 9, September 1972, pp. 967-974.

LEE, R.C.T. (1974) : A sub-minimal spanning tree approach for large data clustering. Proceedings of the Second International Joint Conference on Pattern Recognition, Copenhagen, August 1974.

LOFTSGAARDEN, D.O. and QUESENBERRY, C.P.A. (1965) : A nonparametric estimate of a multivariate density function. Annals of Mathematical Statistics, Vol. 36, pp. 1049-1051.

MACQUEEN, J. (1966) : Some methods for classification and analysis of multivariate observations. Proc. 5th. Berkeley Symp. on Probability and Statistics, pp. 281-297.

MARRIOTT, F.H.C. (1971) : Practical problems in a method of cluster analysis. Biometrics, 27, pp. 501-514.

McRAE, D.J. (1971) : MICKA, a FORTRAN IV iterative k-means cluster analysis program. Behavioral Science, 16, pp. 423-424.

MUCCIARDI, A.N. and GOSE, E.E. (1972) : An automatic clustering algorithm and its properties in high-dimensional spaces. I.E.E.E. Trans. on Systems, Man, and Cybernetics, Vol. SMC - 2, No. 2, April 1972, pp. 247-254.

NAGY, G. (1968) : State of the art in pattern recognition. Proceedings of the I.E.E.E., Vol. 56, No. 5, May 1968, pp. 836-862.

NORTHOUSE, R.A. and FROMM, F.R. (1973) : Some results of non-parametric clustering on large data problems. Proceedings of International Joint Conference on Pattern Recognition.

OSWALD, I. (1966) : Sleep. Penguin Books.

PEARSON, K. (1901) : On lines and planes of closest fit to systems of points in space. Phil. Mag., 6th series, pp.559.

PRIM, R.C. (1957) : Shortest connection matrix network and some generalizations. Bell System Tech. J., 36, pp. 1389-1401.

ROSS, G.J.S. (1969) : Minimum spanning tree, Algorithm AS 13. Applied Statistics, Vol. 18, p. 103.

SAMMON, J.W. (1969) : A nonlinear mapping for data structure analysis. I.E.E.E. Trans. on Computers, Vol. C-18, No. 5, May 1969, pp.401-409.

SCOTT, A.J. and SYMONS, M.J. (1971) : Clustering methods based on likelihood ratio criteria. Biometrics, 27, pp. 387-398.

SEBESTYEN, G.S. (1962) : Decision making processes in pattern recognition. New York : Macmillan.

SEBESTYEN, G. and EDIE, J. (1966) : An algorithm for nonparametric pattern recognition. I.E.E.E. Trans. on Computers, EC - 15, pp. 908-915.

SIBSON, R. (1973) : SLINK : An optimally efficient algorithm for the single-link cluster method. Computer J., Vol. 16, No. 1, pp. 30-34.

SNEATH, P.H.A. (1966) : A method for curve seeking from scattered points. Computer J., Vol. 8, pp. 383-391.

SOKAL, R.R. and MICHENER, C.D. (1958) : A statistical method for evaluating systematic relationships. Univ. Kansas Sci. Bull., 38, pp. 1409-1438.

SOKAL, R.R. and SNEATH, P.H.A. (1963) : Principles of numerical taxonomy. Freeman : San Francisco and London.

THORNDIKE, R.L. (1953) : Who belongs in the family? Psychometrika, Vol. 18, No. 4, December 1953, pp. 267-276.

TUKEY, J.W. and COOLEY, J.W. (1965) : An algorithm for the machine calculation of complex Fourier series. Mathematics of computation, April 1965, Vol. 19, No. 90, pp. 297-301.

VIGLIONE, S.S. (1970) : Applications of pattern recognition technology. In 'Adaptive, learning and pattern recognition systems : theory and applications', ed. J.M. Mendel and K.S. Fu. Academic Press : New York and London.

WATANABE, S. and HARADA, E. (1974) : A dynamical model of clustering. Proceedings of the Second International Joint Conference on Pattern

Recognition, Copenhagen, August 1974.

WILKS, S.S. (1962) : Mathematical Statistics. Wiley : New York and London.

WISHART, D. (1969) : Mode analysis: a generalization of nearest neighbour which reduces chaining effects. In 'Numerical Taxonomy', ed. A.J. Cole, pp. 282-308. New York : Academic Press.

WOLFE, J.H. (1970) : Pattern clustering by multivariate mixture analysis. Multiv. Behav. Res., 5, pp. 329-350.

ZADEH, L.A. (1965) : Fuzzy sets. Information and control, 8, pp. 338-353.

ZAHN, C.T. (1971) : Graph-theoretical methods for detecting and describing Gestalt clusters. I.E.E.E. Trans. on Computers, Vol. C-20, No. 1, Jan. 1971, pp. 68-86.