1973

Unsupervised Iterative Clustering in Pattern Recognition

G. K. Kaveriappa

UNSUPERVISED ITERATIVE CLUSTERING
IN
PATTERN RECOGNITION

BY

G. K. KAVERIAPPA

A thesis submitted
in partial fulfillment of the requirements for the
degree of Master of Science, Major in
Electrical Engineering, South Dakota
State University

1973

UNSUPERVISED ITERATIVE CLUSTERING

IN

PATTERN RECOGNITION

This thesis is approved as a creditable and independent investigation by a candidate for the degree, Master of Science, and is acceptable as meeting the thesis requirements for this degree. Acceptance of this thesis does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

Thesis Adviser Date

Head, Electrical Engineering Dept. Date

ACKNOWLEDGEMENTS

The author expresses his appreciation to Dr. Gerald D. Nelson for his invaluable guidance during the course of this project. The author is also grateful to Mr. David V. Serreyn for his generous help and assistance in performing this research.

The author wishes to thank Mr. Victor I. Myers, Director of the Remote Sensing Institute and the Remote Sensing Institute staff for enabling the research to be performed. This work was partially supported by NASA Grant 42-003-007.

The author is grateful to Professor James K. (Tex) Lewis who provided the data to be analyzed and helped in evaluating the results of the research, to Miss Michelle Casper and Mrs. Ruth Ann Anderson for providing typing, and to Mr. Jack Smith for his photographic help.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

1.1 Remote Sensing

In the fast growing field of remote sensing — acquiring information through the use of cameras and related devices, such as radar and thermal infrared sensors, operated from aircraft and spacecraft — several useful developments are taking place. Several of these developments are centered around the remote sensing vehicle known as ERTS (the Earth Resources Technology Satellite), which is currently orbiting and photographing the earth at an altitude of approximately 570 miles.

Experience has shown that even an expert, well versed in remote sensing, usually cannot derive adequate information from merely a single "frame" of remote sensing imagery covering the area that he is interested in. Various types of remote sensing research, however, have recently demonstrated that the ease, accuracy, and completeness with which information can be derived is likely to be greatly improved through what might be termed the "multi" approach to remote sensing, in which several frames of imagery, all covering the same general geographic area, are variously enhanced and analyzed. This concept is applicable to several different aspects of remote sensing. Some of the "multi" approaches available require the obtaining of 1) multiband photographs, 2) multidate photographs, and 3) multistage photographs - obtaining progressively more detailed information for progressively smaller subsamples of areas being studied.

1.2 Analyzing the Data from Remote Sensing

The data obtained through remote sensing has to be analyzed and interpreted in order to give a useful meaning to the remote sensing problem. There are several ways of doing this, such as photo interpretation and pattern recognition. The last step in the remote sensing problem is to compare results of data analysis with the ground truth information for the same geographic area and to estimate the degree to which the analysis has been successful in deriving the correct information.

There are several pattern recognition techniques currently available by which a set of data can be analyzed and classified into groups or classes. These classification techniques can be broadly classified into two main groups, 1) supervised pattern recognition, in which the classifier needs to be trained by means of a training set of data under human supervision, 2) unsupervised pattern recognition, in which the classifier trains itself.

The typical multi-spectral classification experiment is conducted as follows: the data, collected from a single geographical region under favorable conditions by airborne or spaceborne sensors, are examined in their entirety by the investigator, who decides which areas are most representative of the geographical region under investigation as a whole. The data from these representative areas are assembled to form a training set, which is characterized by the ground truth information delineating the terrain types of interest. A statistical categorizer, or decision box, is constructed on the

basis of the statistical parameters extracted from the training set. The classification performance is then evaluated on another portion of the data (the test site) for which the location and extent of the different types of ground cover are also known to the investigator [1]. Fig. 1.1 is a block diagram representation of a remote sensing problem using pattern recognition techniques.

In unsupervised pattern recognition, the point of departure from standard statistical classification techniques lies in the application of an unsupervised learning approach, by means of mode seeking algorithms to circumvent the difficulties of collecting representative training sets [1].

The mode seeking algorithm is used to divide the data into groups of sample points of similar spectral characteristics. The region on the ground corresponding to each cluster can then be examined to determine the correct label or category to be assigned with each cluster. Thus the classification is performed only once for each cluster rather than for each sample.

Once each sample in the data has been assigned to a cluster, it is necessary to determine which points should be inspected in order to label the cluster correctly. There are various procedures available to do this. One such simple procedure consists of the selection of an equal fraction of randomly located sample points from each cluster.

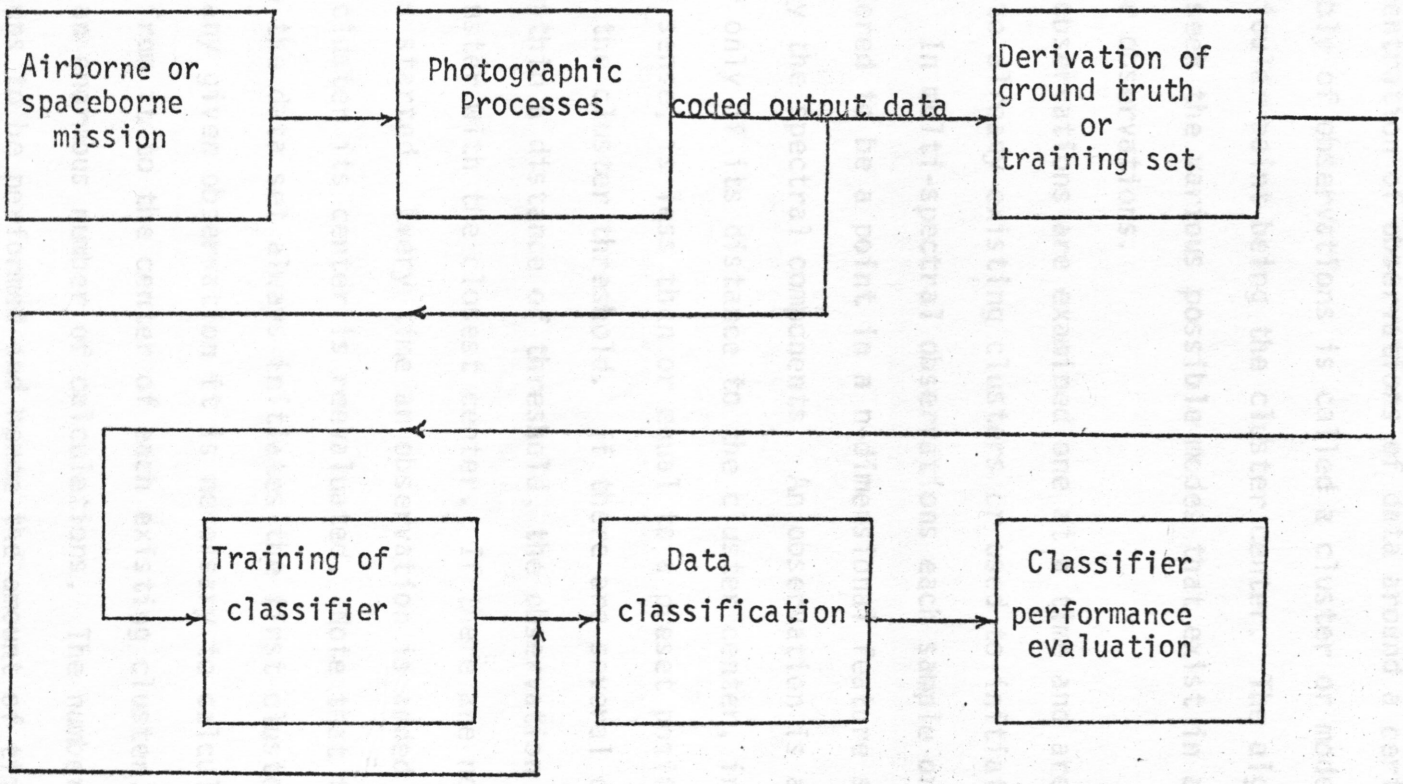


Fig. 1.1 - Block diagram representation of remote sensing problem using pattern recognition techniques.

1.3 Mode Seeking Algorithm

Clustering algorithm is used to sort the observations into relatively homogeneous groups, called the clusters. Whenever there is a concentration of observations of data around a certain point, the assembly of observations is called a cluster or mode, with that particular point being the cluster center. The algorithm tries to seek the various possible modes that exist in a given set of data or observations.

The observations are examined one at a time and are either attached to already existing clusters or used to initiate new clusters. In multi-spectral observations each sample or observation is considered to be a point in a n-dimensional feature space defined by the spectral components. An observation is assigned to a cluster only if its distance to the cluster center, in the euclidean sense, is less than or equal to a preset positive constant, the cluster threshold. If there are several cluster centers within a distance of threshold, the observation is assigned to the cluster with the closest center. If there are none, a new cluster is started. Every time an observation is added to an existing cluster its center is reevaluated. Note that the first sample in the data set always initiates the first cluster.

For any given observation it is necessary to calculate the distance from it to the center of each existing cluster. This involves an enormous number of calculations. The number of calculations to be performed and hence the amount of time required can be considerably reduced by techniques such as "sequential

search" and "strip formation" [1].

To evaluate the performance of the clustering process, the classification of each sample can be compared with the ground truth for that sample. It is possible to prepare a table showing which types of ground cover are most likely to be mistaken for one another by this procedure.

1.4 Project Objectives

The goal of this project was to study in detail the mode seeking algorithm and its application in pattern recognition. The algorithm was first applied to a set of computer generated data and based on that certain observations were made. These are discussed in Chapter 3. The mode seeking algorithm was then modified slightly. It was then applied to two practical problems; 1) a three class problem in a four feature space (photographic transmission in four spectral bands), and 2) a three class problem in a single feature space. Results of these example problems are given in Chapter 4. In Chapter 5, a possible application of mode seeking algorithm in training the K-class classifier is discussed.

CHAPTER 2. LITERATURE REVIEW

The basic assumption in cluster analysis is that the data has local concentrations, or clusters. Thus, if the unclassified patterns are represented in the measurement space, then all data is clustered and cluster analysis will attempt to sort these patterns into classes. In general, the cluster analysis problem is to discover the cluster and classify data into classes corresponding to the natural clusters. For classification to be useful, "variation" within a class must be relatively unimportant in comparison with "variation" among classes. This principle underlines much of cluster analysis.

According to Ball [2] there are relatively few different approaches to such a problem. One such approach exploits the idea that variation within a cluster is small compared with the variation between clusters. An attempt is thus made to find those sets $[S_i]_{i=1}^{K}$ which make the sum of the variances conditioned on each set S_i a minimum. If such a minimum is found, the collection $[S_i]_{i=1}^{K}$ is called a minimum variance partition of order K. Fukanaga and Koontz [3] have analyzed the properties of several procedures which optimize such criteria. There are, however, certain theoretical difficulties with all proposed algorithms, such as in determining the proper value for K.

There are a number of clustering techniques currently available. Widely used in numerical taxonomy are agglomerative and

divisive hierarchical clustering schemes. The agglomerative procedures generally link together the most similar patterns. Then the similarities between the groups of linked patterns and the remaining patterns are computed using the minimum, maximum or mean similarity between the two groups. The procedure continues in this manner linking together the most similar patterns or groups.

The divisive procedures begin with all the patterns in the same group and split the group into two most dissimilar groups. Splitting can be done by examining all possible partitions of two parts for each group and selecting the partition of that group which reduces the within group variance the most. Lance and Williams [4] suggest successively splitting the groups by thresholding that variable in a way which is expected to reduce the variance the greatest for the split group. Mattson and Dammann [5] suggest successively splitting each group by thresholding the dominant eigenvector of the covariance matrix for that group. Wirth et al. [6] suggests thresholding the association or similarity matrix and defining the components of the resulting graph as the clusters.

Nonhierarchical schemes have also been used in clustering a fixed small set of patterns. Most popular among these have been those iterative schemes beginning with an arbitrary set of exhaustive and mutually exclusive clusters and successively improving the set of clusters by transferring patterns from one cluster to another until no further improvement is available [7].

Another type of nonhierarchical scheme [8] starts out by thresholding the association or similarity matrix and defining as "core clusters" the maximal complete subgraphs (cliques) of the resulting graph. Then the smaller core clusters are merged into the large core cluster and largely overlapping core clusters are also merged.

When the number of patterns is not a small and fixed set, all of the preceding methods seem unfeasible, since they each involve too many calculations. In the case of a large number of patterns another approach has been developed. Here, the clustering problem is conceived of in a different way. It is assumed that the patterns are generated by a number of different "sources" according to some unique source probability distribution. The probability distribution for the collection of patterns is then a mixture of the probability distributions of the sources. The clustering problem is then concerned with decomposing the mixture by identifying from the mixture distribution the individual probability distributions of the sources and then constructing a minimum risk Baye's decision rule to assign any pattern to the most probable source. The decision rule, of course, determines a partition and the cells of the partition are the clusters. Work on the identifiability of mixture distributions has been done by Teicher [9-11], Yakowitz [12,13], Yakowitz and Spragins [14] and Stanat [15]. Application and development of this technique under the name "learning without a teacher" or "unsupervised learning" has been done mainly by Fralick [16], Spragins [17], Patrick and Hancock [18], and Patrick [19].

Gitman and Levine [20] have discussed an algorithm which partitions a given sample from a multimodal fuzzy set into unimodal fuzzy sets. Assuming certain assumptions are satisfied, the algorithm derives an optimal partition in the sense of maximum separation. They have suggested a method that can handle data containing more than 30,000 points in a many dimensional space. The method is known as threshold filtering in which a small threshold T_1 is employed for filtering purposes while a large value T_2 is used to evaluate the final grade of membership. The first point, say, X_1, is introduced. Then all other points are introduced sequentially and the distance from X_1 to every point is measured. If the distance between the points X_1 and X_i is less than or equal to T_1, then the grade of membership of X_1 is increased by 1, the corresponding point X_i is assigned finally into the group to which X_1 will later be assigned. Thus, X_i is not considered further in the application of the algorithm. On the other hand, if the distance between the points X_1 and X_i is greater than T_1, then X_i will again be introduced until every point has been assigned. When this process of filtering is terminated, there remains a smaller set of points $X_1, X_2 \ldots, X_n$ with the temporary grades of membership $n_1, n_2, \ldots, n_n$; where the summation $(n_1 + n_2 + \ldots + n_n)$ is equal to the total number of points in the original data set.

Application of certain linear and nonlinear transformations on measurement space brings together like events by clustering them most [21]. Sebestyen has also introduced certain criteria for

developing optimum transformations that not only cluster events of the same class but also separate those that belong to different classes. He has also developed a simple and rapidly applicable decision rule, known as the adaptive sample set construction. This involves the updating of the decision rule, incorporating the effect of a newly introduced sample of the class into the decision making process.

Samples or patterns have local concentrations or clusters and to aid in the design of classifiers, Blasgen [22] has introduced algorithms which estimate the location of these concentrations. More precisely the algorithms estimate modes of essentially unknown probability distributions, given only a sequence of samples from the distribution. The number of modes should be equal to the number of "natural" classes, or clusters, since bi-modal cluster is a contradiction in terms. Thus by estimating both the number and location of the modes, one discovers the number and location of the clusters.

Another way of looking at the problem of clustering multivariate observations is the replacement of a set of vectors with a set of labels and representative vectors [23]. Data points within a cluster are highly similar [24] so that interpretation of the cluster code can be meaningfully made on the basis of knowing what sort of data point is typical (representative vector) of those in the cluster. Jones and Jackson [25] have suggested an iterative technique where clusters are found one at a time. An initial

pattern is picked to be the first pattern in the cluster. Patterns are successively transferred into and out of the cluster in a way which increases the within cluster similarities and decreases the in-cluster to out-cluster similarities. Each pattern is put into the cluster for which the squared distance between it and the cluster mean is the least [26]. Then the new cluster means are computed and the whole procedure repeated.

Nagy and Tolaba [1] have suggested a method of selecting sample regions for assigning identities to the spectral signatures on the basis of statistically determined similarities rather than on a prior consideration.

The clustering algorithm that is used in this work is on the lines of work done by Ball and Hall [26], Jones and Jackson [25], Haralick and Dinstein [24], Sebestyen [21,27], and Fukunaga [23]. Large sets of data are handled by initially grouping the data into subgroups based merely on the order of their occurrence, and finding the modes in each of the subgroups and finally combining the modes between the subgroups.

CHAPTER 3. A MODE SEEKING ALGORITHM

In order to fully understand the usefulness of the mode seeking algorithm in pattern recognition, the algorithm was applied to a set of two hundred computer generated points. There were two classes with 100 points each. The data has two features (x,y), and is Gaussian with a standard deviation equal to 1. Class 1 is centered at (x,y) = (0,0) and class 2 is centered at (x,y) = (0,4).

3.1 Mode Seeking

A computer program was written to group the data into modes. To ensure that the mode-seeking process terminates itself at an appropriate time it is necessary to specify one of the following two: 1) the maximum value of the cluster threshold, or 2) the maximum number of modes allowed for the given set of data. Of course, the numerical values for the above two specifications depend upon the nature and type of data distribution. In practical problems, a prior knowledge of the ground truth information will definitely aid in deciding a good value for the maximum number of clusters to be allowed. Numerical magnitudes of the observations will influence the choice for the maximum value for the cluster threshold. If it is approximately known that the data represent N different classes of objects then the number of clusters allowed should be more than or equal to N, since it is necessary to distinguish the N classes from one another. Note that a mode in its entirety belongs to only one class and it can never

represent more than one class, whereas a class can be represented by one or more modes.

The actual application of the clustering algorithm on data points was written as a separate subroutine. This subroutine was given a title "ASSC", an acronym for adaptive sample set construction. The flow diagram for the main program as well as for the subroutine are shown in Fig. 3.1 and Fig. 3.2,respectively. Since the data of the trial problem was known to consist of only two classes, the maximum number of clusters allowed was arbitrarily chosen to be 4. The starting value for the cluster threshold was taken to be 0.2.

The clustering algorithm is the significant part of the computer program. To start with, the first data point is the center of the cluster number 1. The square of the distance from this center to the next point, in the euclidean sense, is calculated. If this value is less than or equal to the cluster threshold, the point is assigned to the first cluster, if not, a second cluster is initiated with that point as the cluster. Similarly, all other observations or samples are treated. Whenever an observation is under consideration, the square of the distance from the sample to the centers of all the existing clusters is calculated and the smallest of these values is selected. If the smallest value is less than or equal to the cluster threshold, the sample is assigned to the nearest cluster. Whenever a sample is assigned to an existing cluster, the center as well as the grade of membership (measure of the number of points in the cluster) of that cluster is updated. If

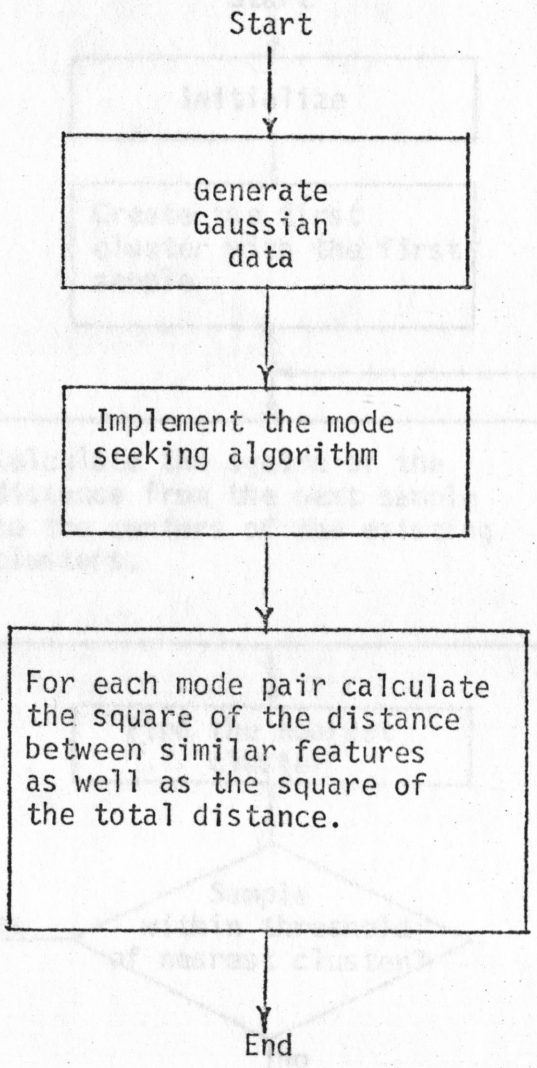


Fig. 3.1 - Flow chart of the main program for mode seeking.

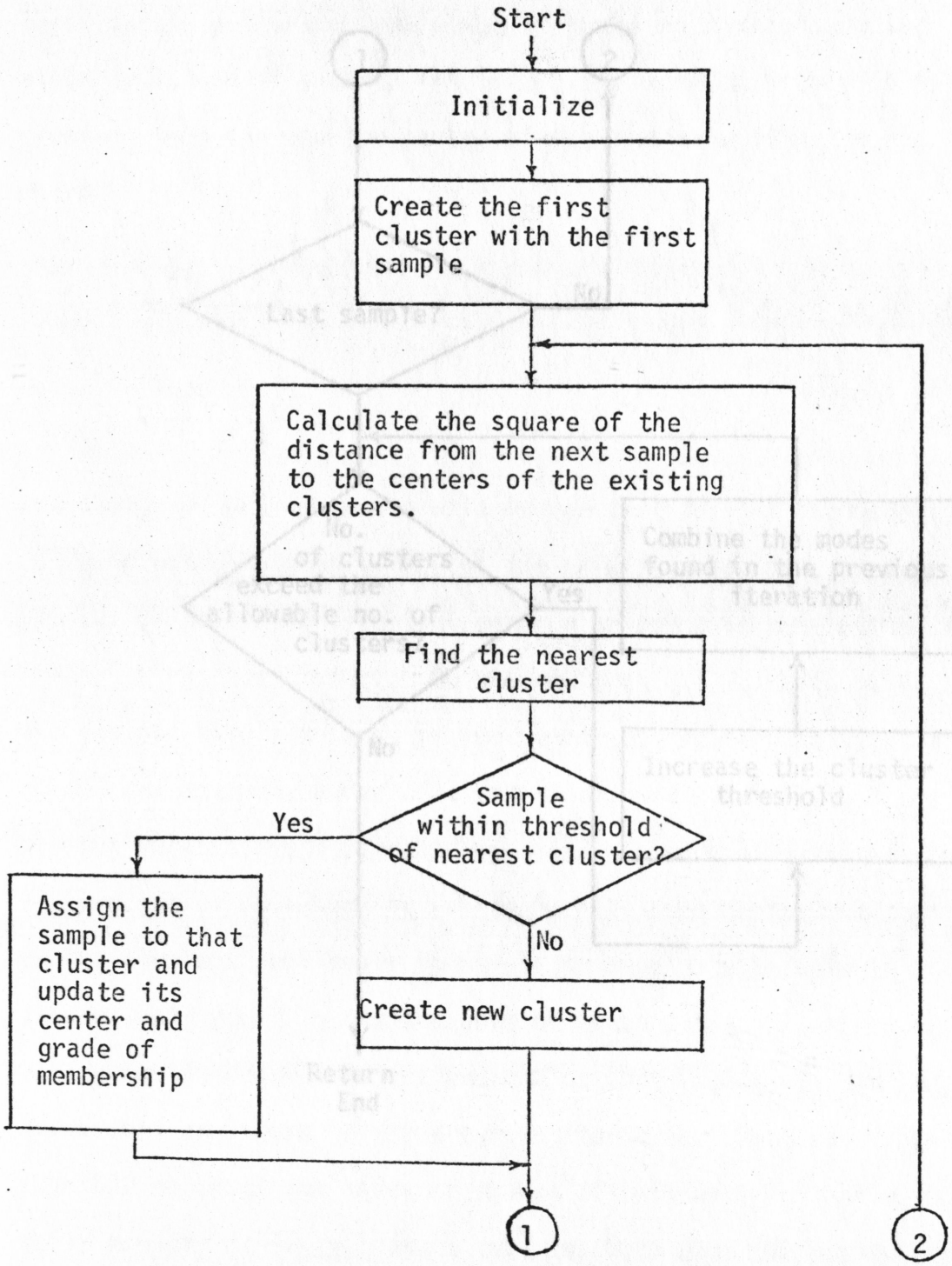


Fig. 3.2 - Flow chart of the mode seeking algorithm
(Continued on the next page)

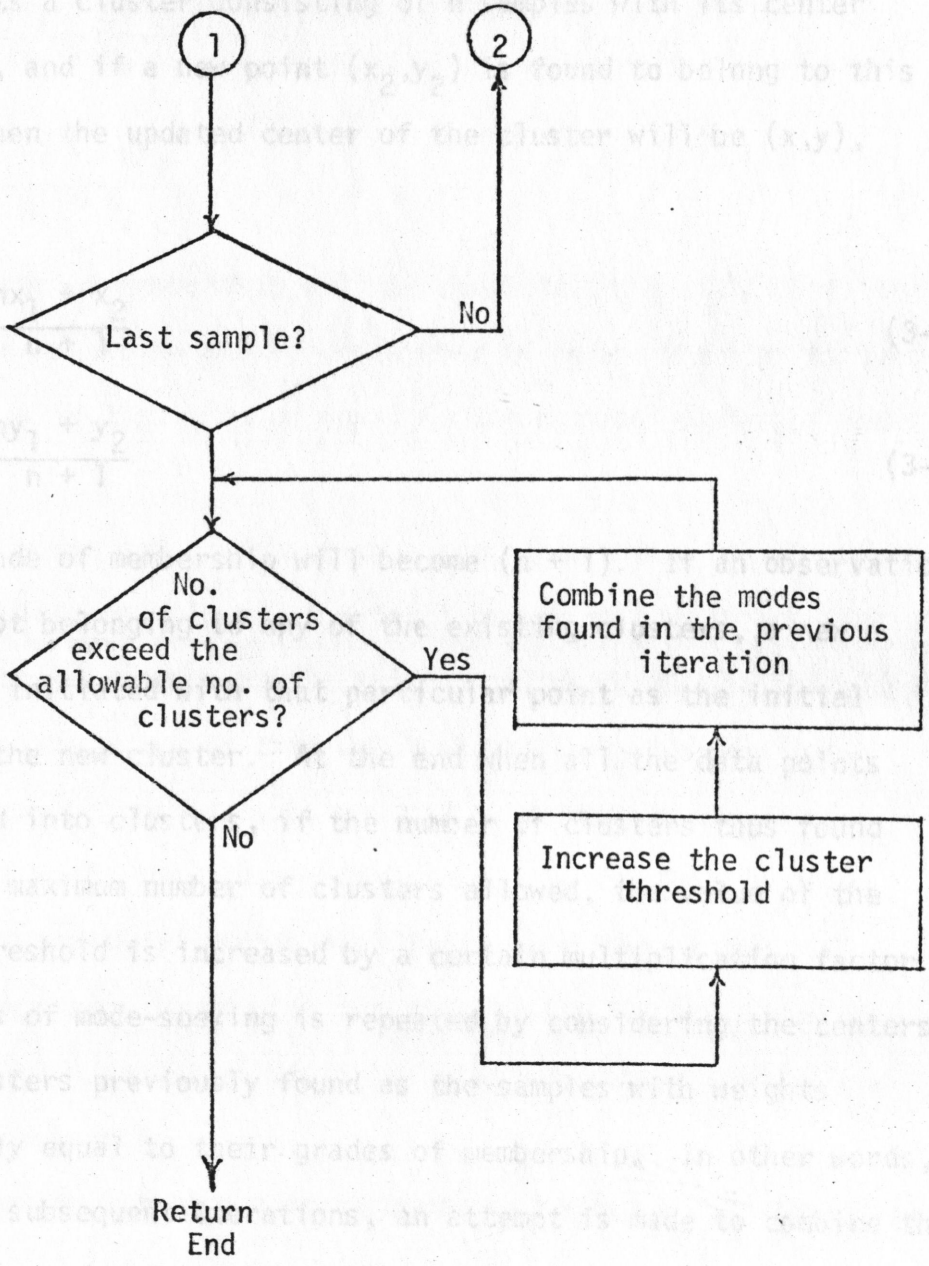


Fig. 3.2 - Flow chart of the mode seeking algorithm
(Continued from the previous page)

there exists a cluster consisting of n samples with its center at (x_1,y_1), and if a new point (x_2,y_2) is found to belong to this cluster, then the updated center of the cluster will be (x,y), where

$$x = \frac{nx_1 + x_2}{n + 1} \tag{3-1}$$

$$y = \frac{ny_1 + y_2}{n + 1} \tag{3-2}$$

and the grade of membership will become $(n + 1)$. If an observation is found not belonging to any of the existing clusters, a new cluster is initiated with that particular point as the initial center of the new cluster. At the end when all the data points are grouped into clusters, if the number of clusters thus found exceed the maximum number of clusters allowed, the value of the cluster threshold is increased by a certain multiplication factor and the process of mode-seeking is repeated by considering the centers of the clusters previously found as the samples with weights respectively equal to their grades of membership. In other words, during the subsequent iterations, an attempt is made to combine the modes that are found in the previous iteration. Whenever it becomes possible to merge two modes of grades of membership n_1 and n_2 with their centers at (x_1,y_1) and $x_2,y_2)$, respectively, the combined mode will have its center at (x,y), where

$$x = \frac{n_1 x_1 + n_2 x_2}{n_1 + n_2} \tag{3-3}$$

$$y = \frac{n_1 y_1 + n_2 y_2}{n_1 + n_2} \tag{3-4}$$

and its grade of membership will be equal to $(n_1 + n_2)$. The iterative procedure comes to an end if the number of modes found at the end of an iteration is less than or equal to the maximum number of modes allowed.

For a given set of data with a specified value for the maximum number of clusters allowed, the total number of iterations that are to be performed depends on the initial value chosen for the cluster threshold. The computer time required and hence the cost of operation can be considerably reduced by selecting the correct initial value for the cluster threshold. In the trial problem the cluster threshold incremental factor was taken to be 1.2. Four clusters were found with the final value for the threshold being 5.58268. Table 3-1 lists the clusters, their centers and grades of membership.

Separation between pairs of modes in terms of individual features as well as all the features taken at one time, are calculated in the main program. Table 3-2 lists these separations for the four clusters found in the trial problem. This table shows the mode separations and whether it is possible to combine any of the modes. From Table 3-2 one can see that the total separation between the modes 1 and 2 is not much different from the value for the cluster

TABLE 3-1. MODES FOUND IN A SET OF 200 COMPUTER GENERATED GAUSSIAN DATA WITH TWO FEATURES.

Cluster	Center		Grade of Membership
	x	y	
1	0.028	0.114	102
2	-2.346	-0.582	3
3	-0.033	4.100	92
4	-2.557	4.561	3

TABLE 3-2. FEATUREWISE AS WELL AS TOTAL SEPARATION BETWEEN THE MODES FOUND IN THE TRIAL PROBLEM.

Modes		Features	Square of Feature Distance	Square of Total Features
1	2	1	5.63684	5.63684
1	2	2	0.48451	6.12135
1	3	1	0.00372	0.00372
1	3	2	15.89121	15.89492
1	4	1	6.68315	6.68315
1	4	2	19.77681	26.45995
2	3	1	5.35103	5.35103
2	3	2	21.92528	27.27629
2	4	1	0.04451	0.04451
2	4	2	26.45229	26.49680
3	4	1	6.37160	6.37160
3	4	2	0.21228	6.58388

threshold. Also the separation between modes 1 and 3 and modes 1 and 4 are large compared to the separation between modes 1 and 2. As such, if necessary, it is possible to combine the modes 1 and 2. Similarly, modes 3 and 4 can be combined.

3.2 Labeling the Data Point

In order to identify the cluster that the data point belongs to, a second computer program was written. Fig. 3.3 shows the flow chart for the program logic.

The four clusters found during mode seeking process were numbered as 1, 2, 3, and 4,respectively. The same 200 Gaussian data points were generated in a two dimension measurement space. The first 100 of them have a mean of (0,0) and the latter 100 have (0,4) as their mean. For each point, the program calculates its distance from the center of cluster 1. If the square of this distance is less than or equal to the final value found for the cluster threshold during mode seeking then the point is classified as belonging to the cluster 1 and a label of 1 is attached to that point. If the point is found not belonging to the cluster 1, then it is tested for its belonging to cluster 2, then with cluster 3, 4, and so on. At the end, if it is found that the point does not belong to any of the four clusters, then a label of 5 is attached to that point meaning that it is not possible to classify that point into any of the four clusters of interest.

A computer printout of the data points in terms of their cluster labels was obtained in a matrix form. It was found that 96 points had

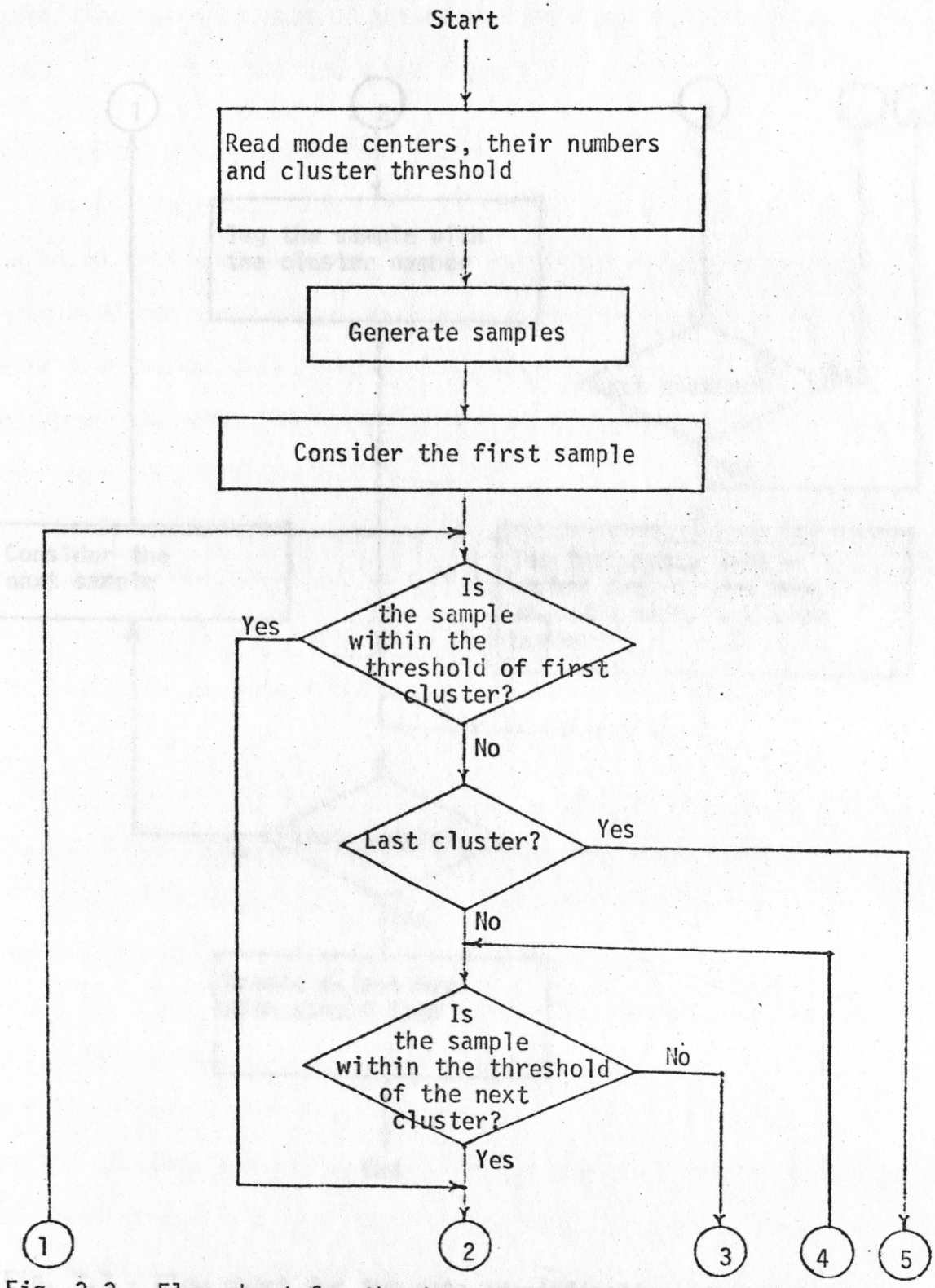


Fig. 3.3 - Flow chart for the data identification program logic
(Continued on the next page)

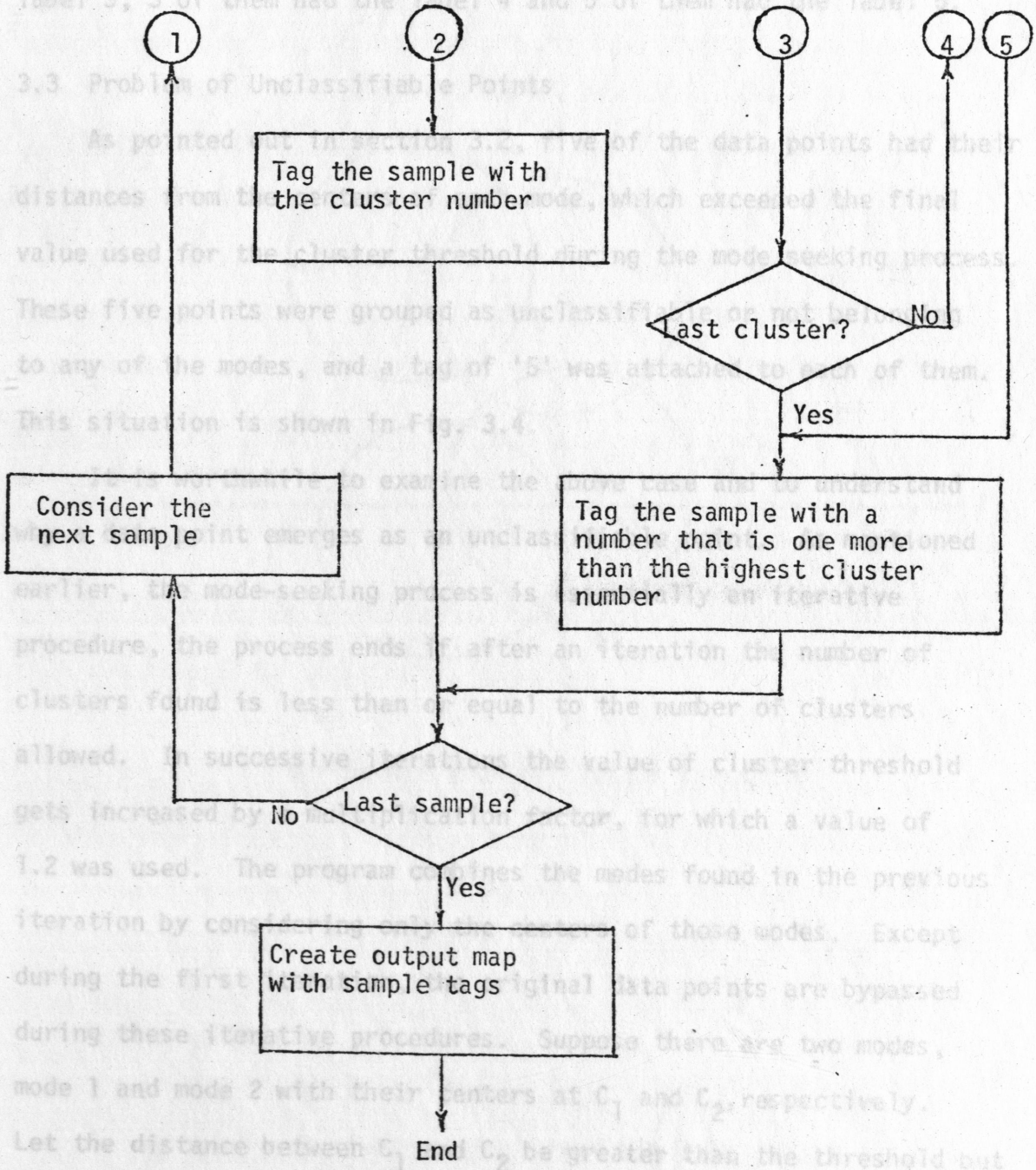


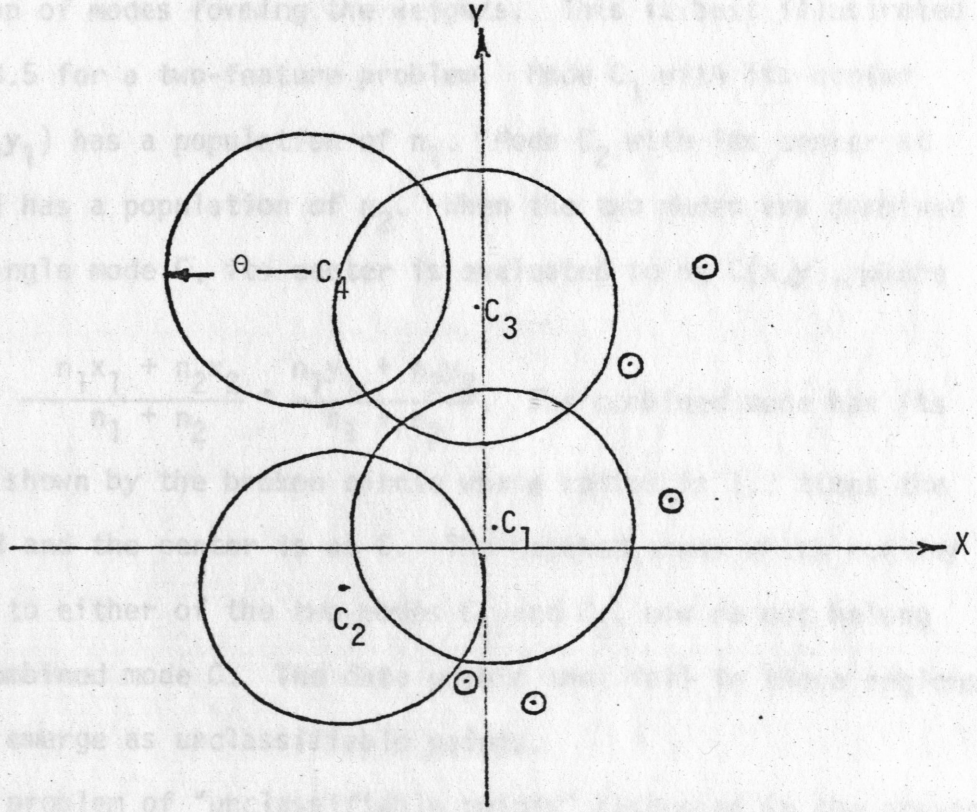
Fig. 3.3 - Flow chart for the data identification program logic (Continued from the previous page)

label 1 attached to them, 4 points had the label 2, 92 points had the label 3, 3 of them had the label 4 and 5 of them had the label 5.

3.3 Problem of Unclassifiable Points

As pointed out in section 3.2, five of the data points had their distances from the centers of each mode, which exceeded the final value used for the cluster threshold during the mode seeking process. These five points were grouped as unclassifiable or not belonging to any of the modes, and a tag of '5' was attached to each of them. This situation is shown in Fig. 3.4.

It is worthwhile to examine the above case and to understand why a data point emerges as an unclassifiable point. As mentioned earlier, the mode-seeking process is essentially an iterative procedure, the process ends if after an iteration the number of clusters found is less than or equal to the number of clusters allowed. In successive iterations the value of cluster threshold gets increased by a multiplication factor, for which a value of 1.2 was used. The program combines the modes found in the previous iteration by considering only the centers of those modes. Except during the first iteration, the original data points are bypassed during these iterative procedures. Suppose there are two modes, mode 1 and mode 2 with their centers at C_1 and C_2, respectively. Let the distance between C_1 and C_2 be greater than the threshold but less than or equal to 1.2 times the threshold. Hence, during the next iteration these two modes will be combined into one mode with



θ = square root of cluster threshold

Fig. 3.4 - Unclassifiable points; the points are shown as θ. C_1, C_2, C_3 and C_4 are the centers of the modes 1, 2, 3, and 4 respectively.

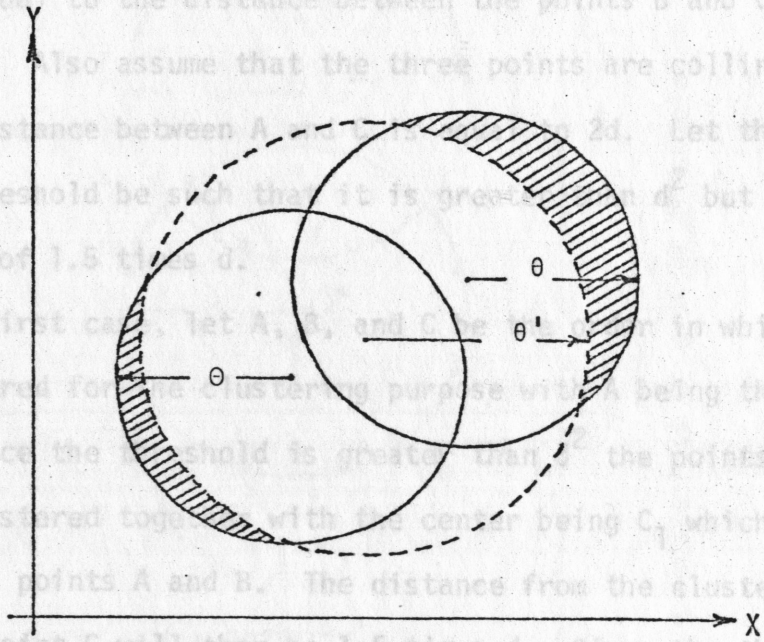
its center evaluated to be at C, where the coordinates of C are given by the weighted averages of the coordinates C_1 and C_2 with the population of modes forming the weights. This is best illustrated in Fig. 3.5 for a two-feature problem. Mode C_1 with its center at $C_1(x_1,y_1)$ has a population of n_1. Mode C_2 with its center at $C_2(x_2,y_2)$ has a population of n_2. When the two modes are combined into a single mode C, its center is evaluated to be C(x,y), where

$$C(x,y) = \frac{n_1x_1 + n_2x_2}{n_1 + n_2}, \frac{n_1y_1 + n_2y_2}{n_1 + n_2}.$$ The combined mode has its boundary shown by the broken circle whose radius is 1.2 times the threshold and the center is at C. The hatched areas which earlier belonged to either of the two modes C_1 and C_2, now do not belong to the combined mode C. The data points that fall in these regions will now emerge as unclassifiable points.

The problem of "unclassifiable points" discussed in the previous two paragraphs can be avoided by proper modification of the subroutine to determine the modes. This modification is discussed in detail in the next chapter. It is in order at this stage to consider the following two aspects of the mode seeking process: 1) effect of first data point on the resulting modes, and 2) distance between modes.

3.4 Effect of First Data Point on the Outcome of Modes

The computer program for the mode seeking process lets the first data point in the set of data be the center of the first cluster. This has a definite effect on which of the data points are



Θ = square root of cluster threshold

θ' = square root of 1.2 times cluster threshold

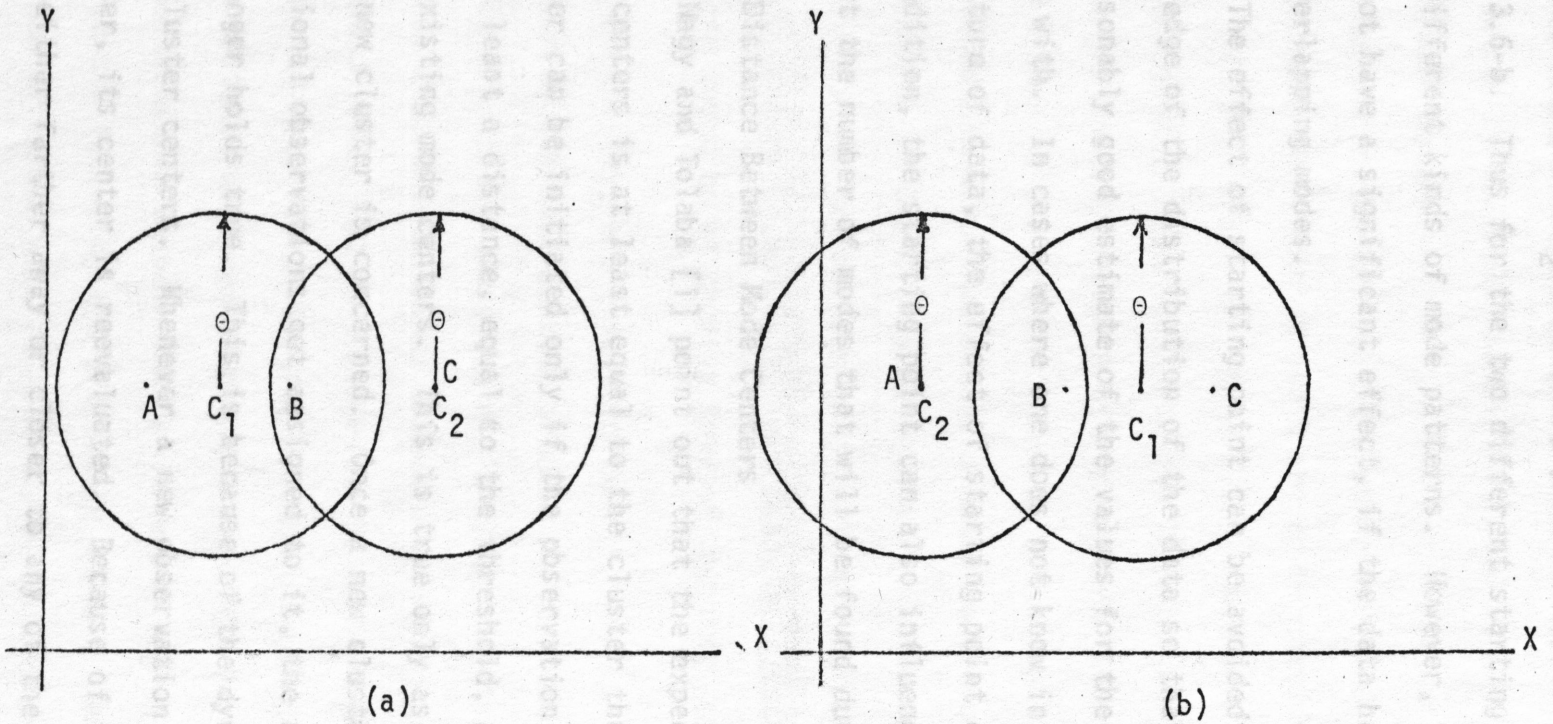
Fig. 3.5 - Emergence of unclassifiable points.

clustered together and as such will affect the outcome of modes. This can be best understood by considering an example in a two dimensional space which is illustrated in Fig. 3.6.

Points A, B, and C represent three typical points in the given set of data. Suppose that the distance between the points A and B is equal to the distance between the points B and C and each equal to d. Also assume that the three points are collinear so that the distance between A and C is equal to 2d. Let the value of cluster threshold be such that it is greater than d^2 but less than the square of 1.5 times d.

As a first case, let A, B, and C be the order in which the points are considered for the clustering purpose with A being the starting point. Since the threshold is greater than d^2 the points A and B will be clustered together with the center being C_1 which is midway between the points A and B. The distance from the cluster center C_1 to the point C will then be 1.5 times d. Since the threshold is less than the square of 1.5 times d, the point C will give rise to a second cluster to be formed whose center C_2 will be the point C itself. Thus we have two clusters: 1) cluster 1 centered at C_1 containing the points A and B, and 2) cluster 2 centered at C_2 containing the point C. This is shown in Fig. 3.6-a.

As a second case, let C, B, A be the order in which the three points are considered with point C being the starting point. By reasoning in the same way as in the first case we will come up with two modes, mode 1 centered at C_1 containing the points B and C, and



Θ = square root of cluster threshold

Fig. 3.6 - Effect of starting point on the resulting modes, (a) modes obtained with point A as the starting point, (b) modes obtained with point C as the starting point.

mode 2 centered at C_2 containing point A. This is shown in Fig. 3.6-b. Thus for the two different starting points we have two different kinds of mode patterns. However, the starting point may not have a significant effect, if the data has distinct nonoverlapping modes.

The effect of starting point can be avoided if one has a prior knowledge of the distribution of the data so that he can specify a reasonably good estimate of the values for the mode centers to start with. In cases where one does not know in advance the structure of data, the effect of starting point cannot be overcome. In addition, the starting point can also influence to a certain extent the number of modes that will be found during an iteration.

3.5 Distance Between Mode Centers

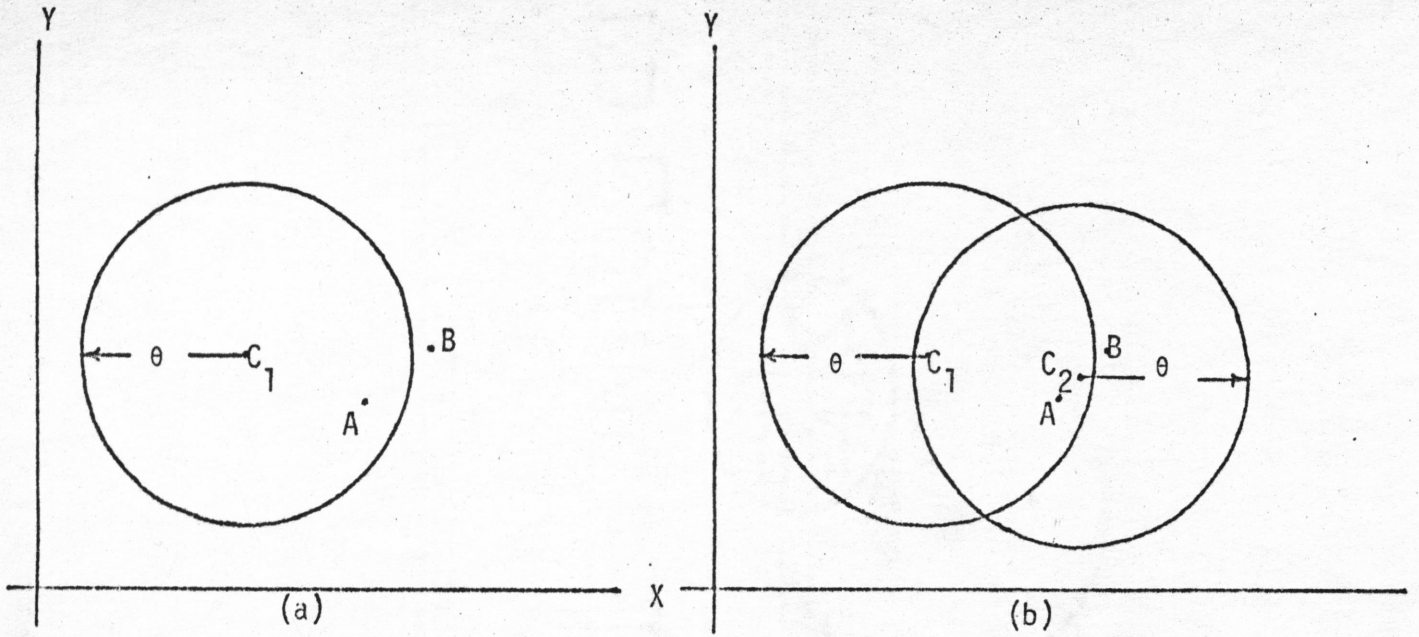
Nagy and Tolaba [1] point out that the expected distance between mode centers is at least equal to the cluster threshold because a new cluster can be initiated only if the observation under consideration is at least a distance, equal to the threshold, away from all of the existing mode centers. This is true only as far as the initiation of a new cluster is concerned. Once a new cluster is formed and additional observations get assigned to it, the above condition no longer holds true. This is because of the dynamic nature of the cluster centers. Whenever a new observation is assigned to a cluster, its center is reevaluated. Because of this the center can move either farther away or closer to any of the centers, though to start with it was at least a distance, equal to the threshold, away

from the centers of the clusters then existing. This fact can be illustrated by a simple example depicted in Fig. 3.7 in a two-dimensional space.

Suppose that there is one cluster with its center at C_1 and that the two points A and B are to be considered in that order. This is shown in Fig. 3.7-a. Since point A is at a distance greater than the threshold from the center C_1, a second cluster is initiated with A as its center. Point B is closer to point A than to the center C_1 and as such it will be assigned to the second cluster. The second cluster center will now be evaluated to be C_2; midway between points A and B as shown in Fig. 3.7-b. It can be seen that the distance between the two cluster centers C_1 and C_2 is less than the cluster threshold. This has been found to be true in some of the trial problems.

3.6 Nearest Neighbor Classification

During the mode seeking process an observation is assigned to a particular cluster whose center is closest to the observation, provided that this shortest distance is less than or equal to the threshold value. It is appropriate to apply the same criterion during the data identification process also, with the exception that the least distance need not necessarily be less than or equal to the threshold value. This will enable one to classify all the data points including the previously defined "unclassifiable" ones. This new criterion will be used during data identification with modes along with the modified mode seeking process.



θ = square root of cluster threshold

Fig. 3.7 - Case of two modes with separation less than cluster threshold; (a) before considering points A and B, (b) after considering points A and B.

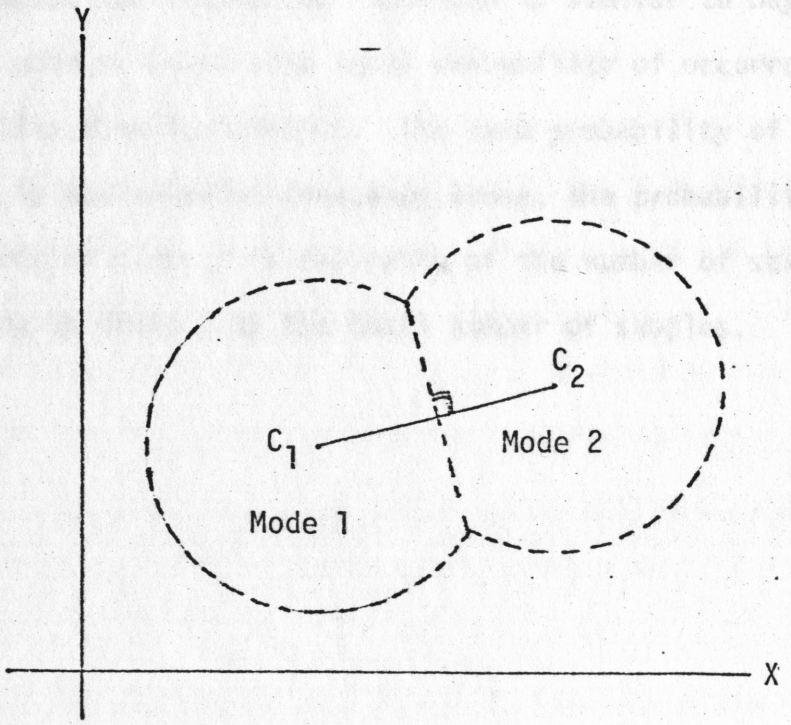


Fig. 3.8 - Boundary between two overlapping modes according to nearest neighbor classification.

The concept of the nearest neighbor classification enables one to draw a boundary between two overlapping modes. In a two-feature problem this boundary is given by the perpendicular bisector of the line joining the centers of the two modes as shown in Fig. 3.8. In this respect the clustering algorithm is similar to Baye's classifier for the case of modes with equal probability of occurrence and normal probability density function. The term probability of occurrence is used in the relative frequency sense; the probability of occurrence of class j is the ratio of the number of samples belonging to class j to the total number of samples.

CHAPTER 4. MODIFIED MODE SEEKING ALGORITHM

4.1 Modified Mode Seeking Algorithm

As mentioned in the previous chapter, the problem of "unclassifiable" points can be avoided by suitably modifying the original mode seeking algorithm. As reasoned earlier, these "unclassifiable" points arose because of the fact that the successive iterations in the mode seeking process considered only the centers of the modes found in the previous iteration and not the original set of data points. Accordingly, the mode seeking algorithm was modified such that every iteration will go through the original data set. The time required, in terms of computer operation, to implement the modified mode seeking algorithm, as compared to that of implementing the original mode seeking algorithm, is more from one point of view and is less from another point of view. These two aspects are discussed in the following two paragraphs. The increase and decrease in computer time practically cancel each other, as evidenced in some of the trial problems.

The modified mode seeking algorithm requires an increased amount of time in terms of computer operation and hence an increase in the cost of processing the data. This is because, in the original mode seeking algorithm, the feature of combining the modes of previous iteration would reduce the number of points to be processed by the successive iterations. Hence, the time required for successive iterations decreases while the time remains the same for each iteration in the modified algorithm.

Another distinct feature of the modified mode seeking algorithm is the following: whenever a new cluster is formed during an iteration, the total number of clusters found is compared with the maximum allowable number of clusters specified. If the number of clusters exceeds the limit, then the iteration is immediately terminated. The threshold is then increased by a factor 1.2 and the next iteration is performed. Thus the duration of each iterative cycle in terms of computer time is very short to start with and it increases gradually with the successive cycles as we are able to combine or cluster more and more points before the number of clusters formed exceeds the maximum allowable number of clusters. It is not possible to introduce this feature in the original mode seeking algorithm since we have to go through all the modes of the previous iterative cycle irrespective of the resulting number of modes. The saving in computer time and hence the reduction in the cost of processing the data practically completely off-sets the increase in the cost mentioned in the previous paragraph.

The modified mode seeking algorithm is shown in the form of block diagram in Fig. 4.1.

4.2 Handling a Large Set of Data

The amount of data that can be processed at one time is limited by the size of the computer memory available. When the size of the data set exceeds this limit, means have to be found to overcome this limitation. Gitman and Levine [20] have

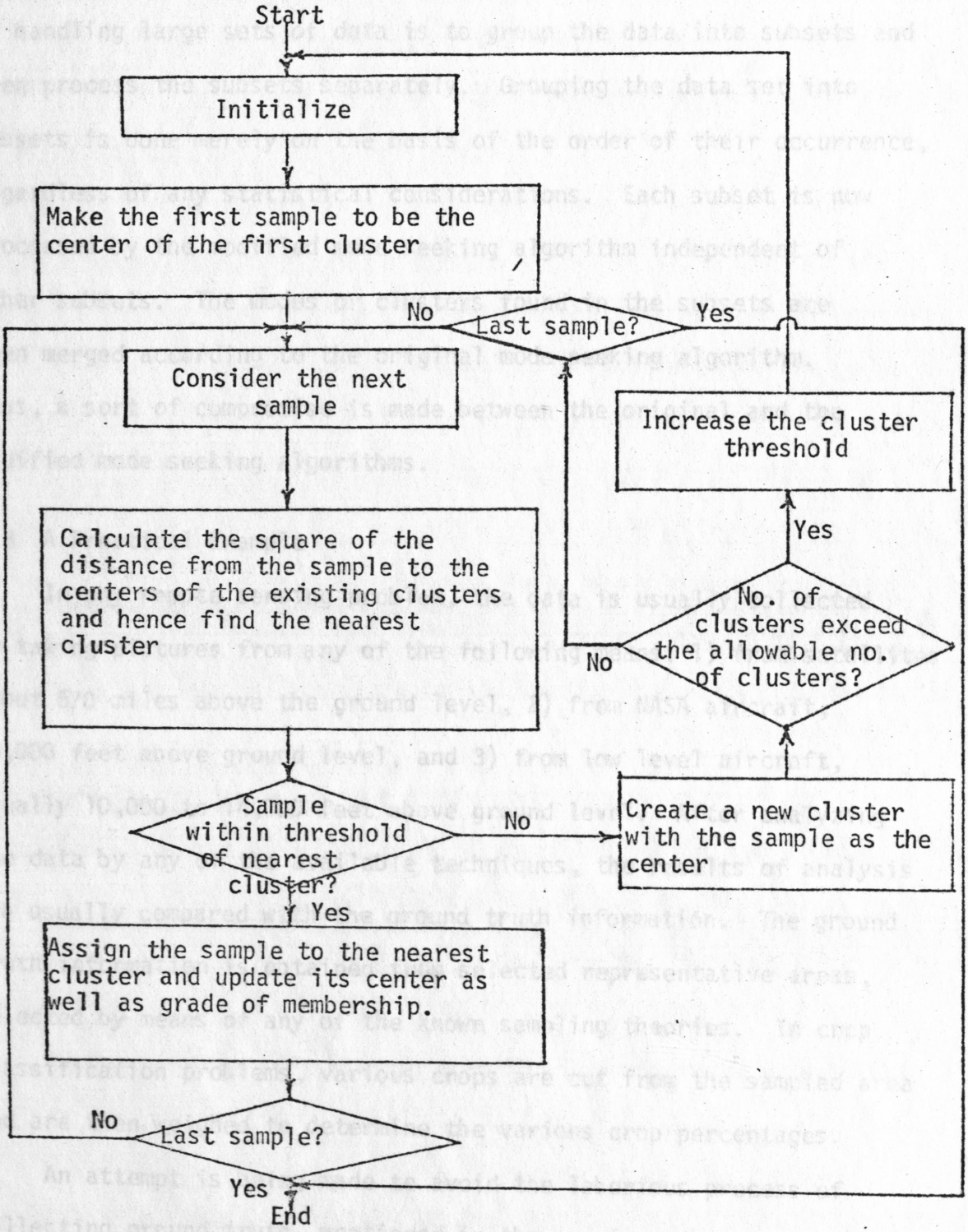


Fig. 4.1 - Flow chart for the modified mode seeking algorithm.

suggested a method known as threshold filtering. One simple way of handling large sets of data is to group the data into subsets and then process the subsets separately. Grouping the data set into subsets is done merely on the basis of the order of their occurrence, regardless of any statistical considerations. Each subset is now processed by the modified mode seeking algorithm independent of other subsets. The modes or clusters found in the subsets are then merged according to the original mode seeking algorithm. Thus, a sort of compromise is made between the original and the modified mode seeking algorithms.

4.3 A Practical Example

In any remote sensing problem, the data is usually collected by taking pictures from any of the following means: 1) from satellites about 570 miles above the ground level, 2) from NASA aircraft, 60,000 feet above ground level, and 3) from low level aircraft, usually 10,000 to 14,000 feet above ground level. After analyzing the data by any of the available techniques, the results of analysis are usually compared with the ground truth information. The ground truth information is obtained from selected representative areas, selected by means of any of the known sampling theories. In crop classification problems, various crops are cut from the sampled area and are then weighed to determine the various crop percentages.

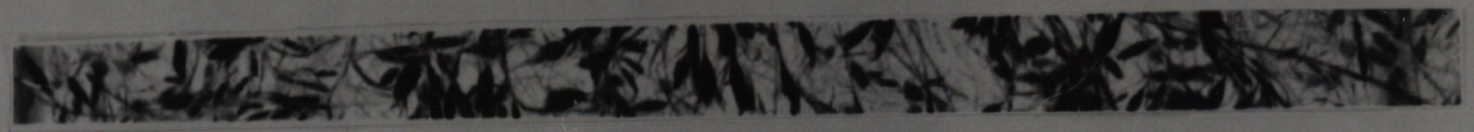
An attempt is being made to avoid the laborious process of collecting ground truth, mentioned in the previous paragraph. The idea here is to take pictures, at the ground level, of the sampled

area. Data is then derived from these pictures and is analyzed. It is hoped that such an analysis would give good results which can be used to train the analyzer that is being used to analyze the data obtained through low level aircraft.

A small strip of western range land about 3 feet in length was analyzed by the mode seeking process. Three main classes were apparent; 1) Bromus japonicus, 2) western wheat grass, and 3) short grass crowns. Six modes were allowed in the mode seeking analysis. Three frames of pictures were taken of the strip under consideration. Fig. 4.2 shows the original black and white pictures of the three frames. Frame 1 is the picture of the strip containing mainly Bromus japonicus with a background of short grass crowns and bare soil. Frame 2 is the picture containing western wheat grass with a background of short grass crowns and bare soil. Frame 3 consists of short grass crowns.

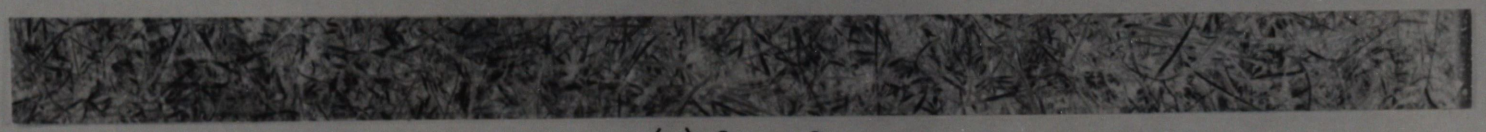
Photographic transmission measurements formed the data to be analyzed. Each frame had 26 scan lines of 360 points each. Thus there were 9,360 samples per frame or a total of 28,080 samples in all three frames. Each sample had four features, which are the photographic transmission measurements through neutral (no filter), red, green, and blue filters. Transmission readings were digitized before they were analyzed. Figs. 4.3, 4.4, and 4.5 show the digitized version of the frames 1, 2, and 3, respectively as seen through neutral, red, green, and blue filters.

All three frames of data were analyzed twice. First each frame



(a) frame 1

(b) frame 2



(c) frame 3

Fig. 4.2 - Black and white picture of the area under investigation in example 1.

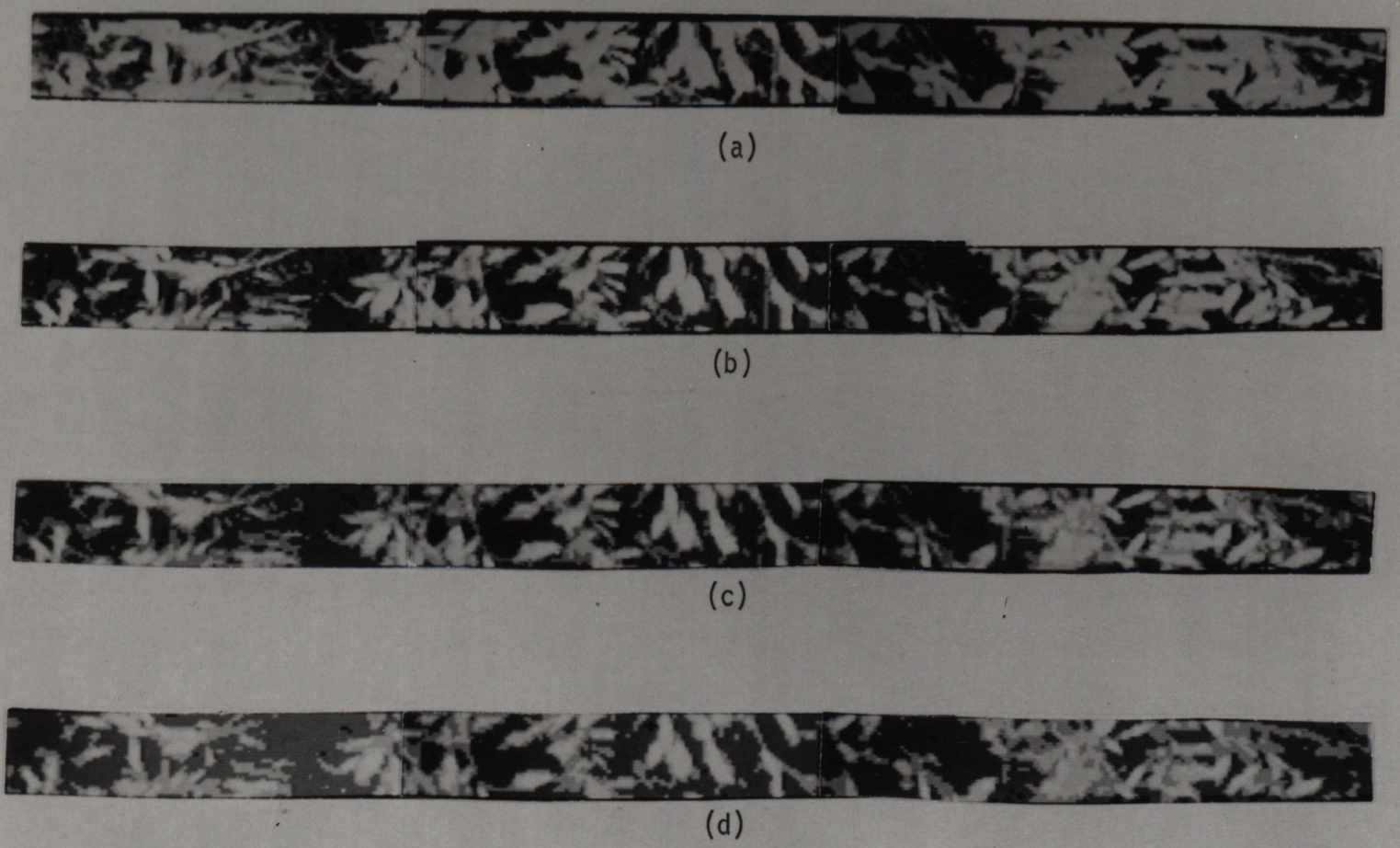


Fig. 4.3 - Digitized version of frame 1 in example 1, as seen through (a) no filter, (b) red filter, (c) green filter, and (d) blue filter.

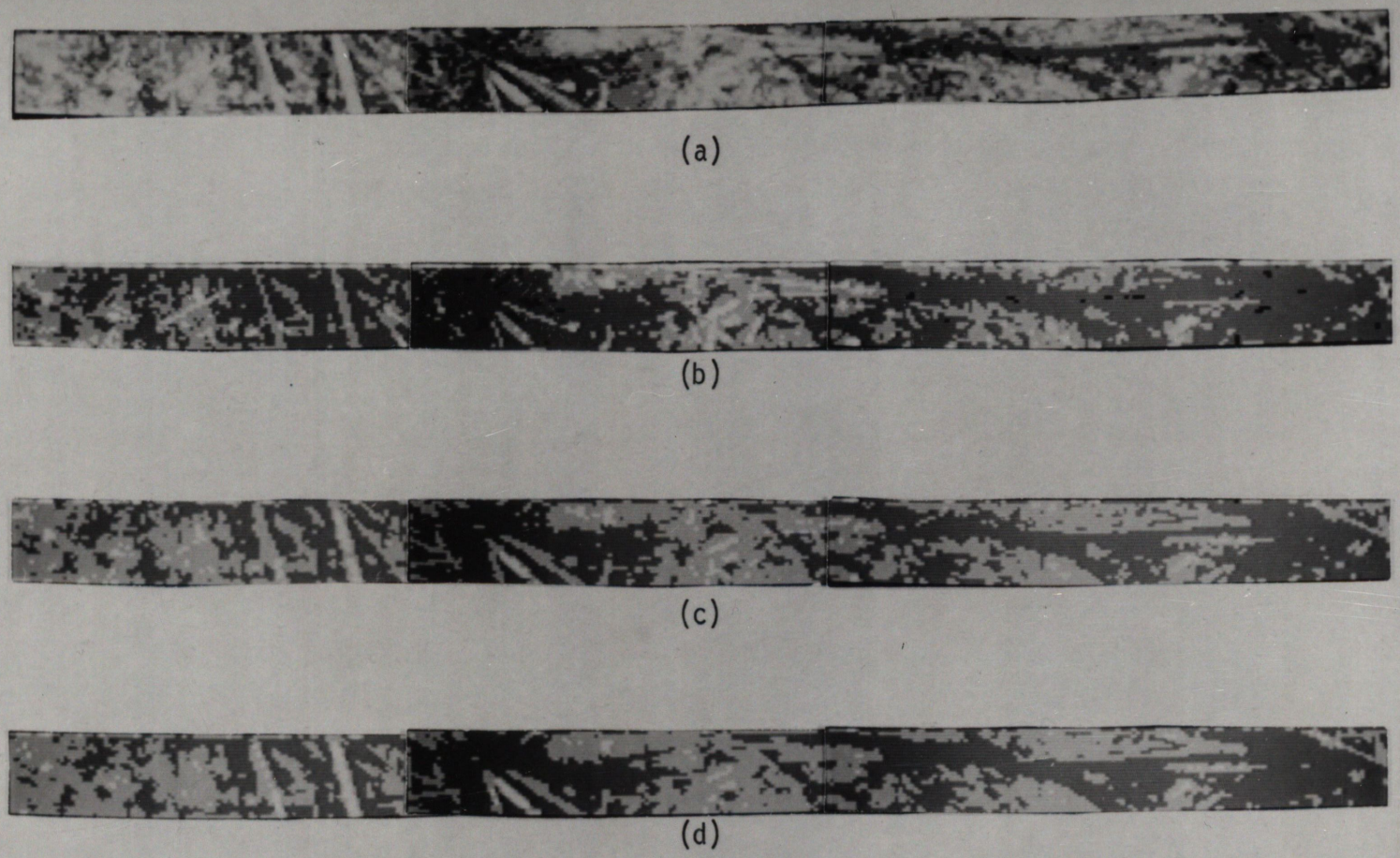


Fig. 4.4 - Digitized version of frame 2 in example 1, as seen through (a) no filter, (b) red filter, (c) green filter, and (d) blue filter.

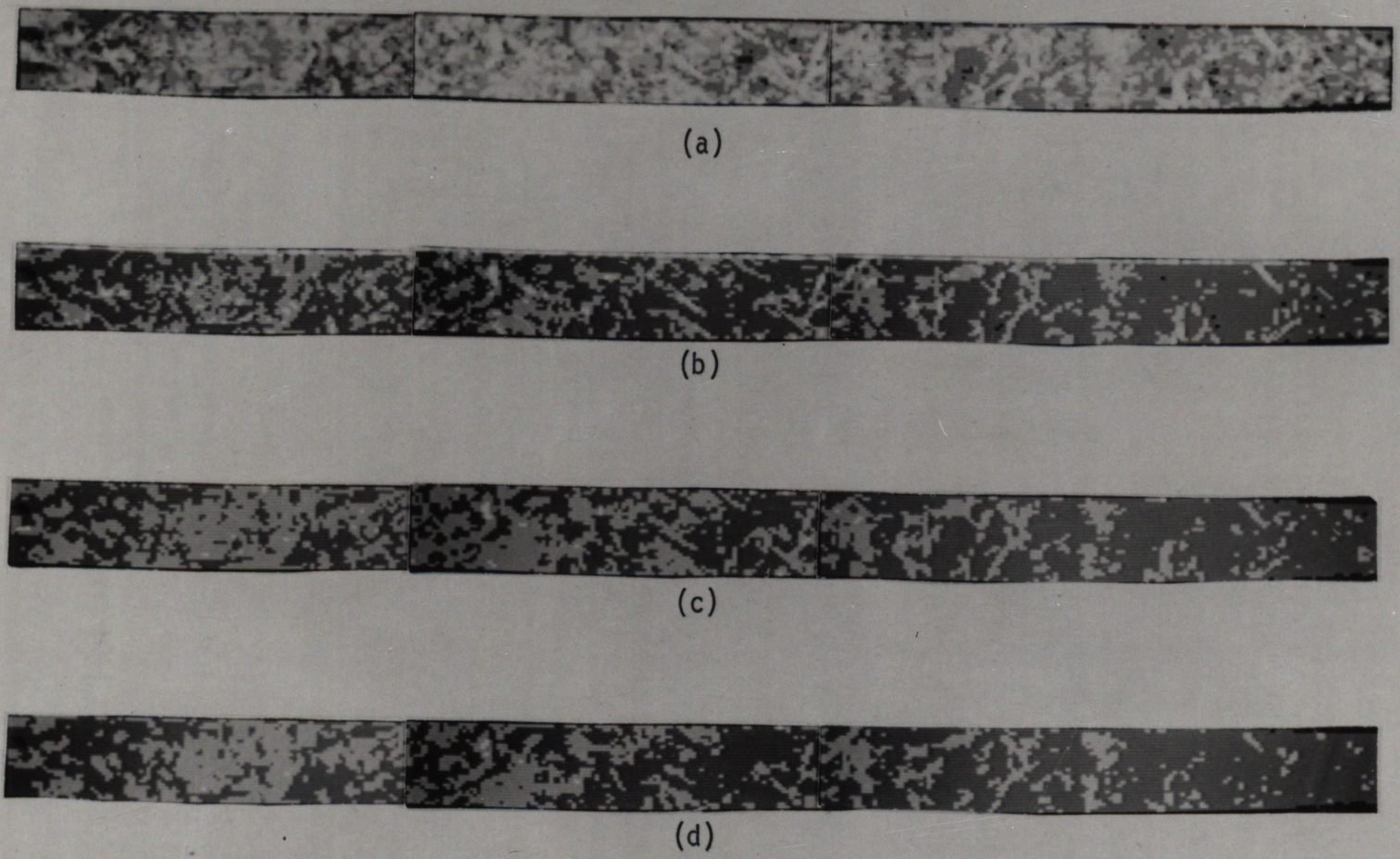
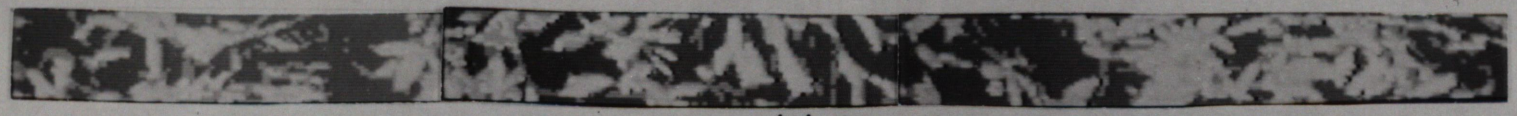


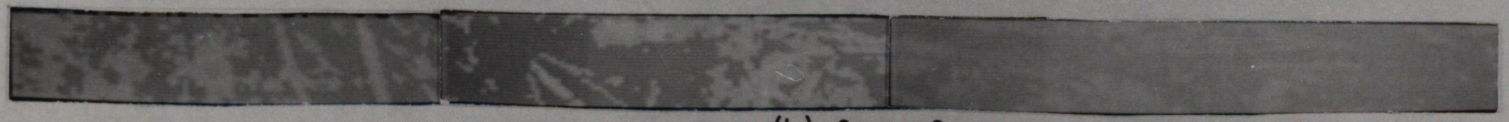
Fig. 4.5 - Digitized version of frame 3 in example 1, as seen through (a) no filter, (b) red filter, (c) green filter, and (d) blue filter.

of data is treated as a complete data set independent of the other two frames. Data set was divided into subsets with each scan line forming a subset. Modes were sought for each scan line and these modes were stored in the computer until all the scan lines were analyzed. Modes of individual scan lines were then merged as explained in section 4.2. Once the modes for the complete frame were obtained, every sample of observation was then identified as belonging to a particular mode. The identification codes or labels of every sample are then converted into an appropriate grey level for the purpose of display on a TV screen. The classification result of each frame was displayed on a TV screen and a picture of the same was taken. Fig. 4.6 is a block diagram showing the various processes the data has undergone. Fig. 4.7 shows the classification results for the three frames. Next, the three frames were analyzed by using the data for all the three frames together as if the three frames were interdependent. This kind of analysis is important in airborne missions, where, in order to bring out as many details as possible, it may be necessary to take photographs from different altitude levels. Fig. 4.8 shows the classification results for the three frames based on the combined data for the three frames.

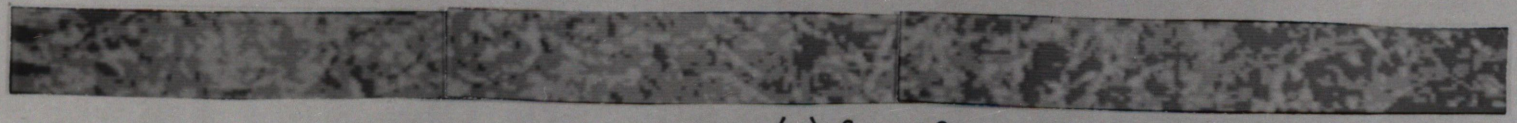
It was found that the mode seeking process classified the data into three major classes. The classification was correct to a reasonably good extent. The three main classes found were Bromus japonicus, western wheat grass and short grass crowns.



(a) frame 1



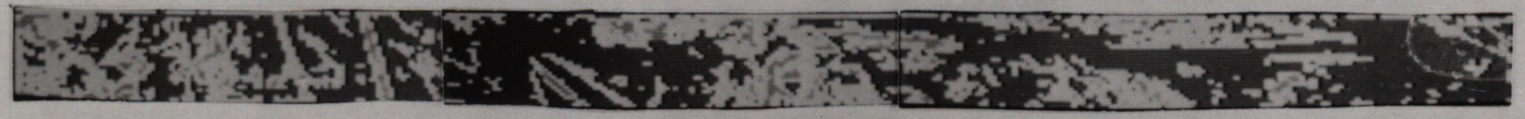
(b) frame 2



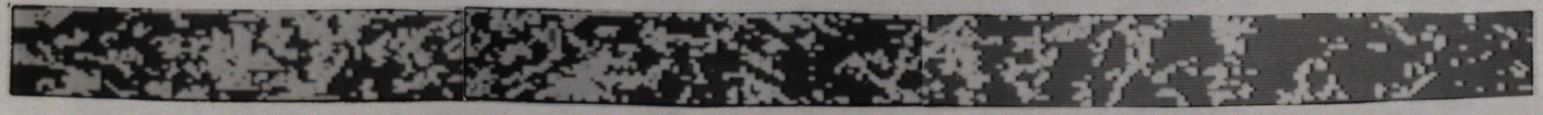
(c) frame 3

Fig. 4.7 - Classification results by mode seeking for example 1, treating each frame independent of others.

(a) frame 1



(b) frame 2



(c) frame 3

Fig. 4.8 - Classification results by mode seeking for example 1, using the data from all three frames at one time.

However, it could not separate the dying crown leaves from the dying upper wheat grass, as they were of the same color. This is expected because the only feature that formed the data was the transmission measurement which depends on the reflectance of the object, which in turn depends on the color of the object. It is also seen that analyzing each frame independently of the others gives a better classification. Nevertheless, the second approach is important for the reason already mentioned. In all, mode seeking gives fairly good classification results.

Fig. 4.9 shows the boundaries between the six modes, obtained during the mode seeking analysis of frame 1. The boundaries are shown in a two feature space for every possible pair of two features out of the total of four features. It is possible to reduce this diagram to a single feature space by simply projecting the mode centers to the respective base axis and selecting the midpoint between any two modes as the boundary between those two modes, as is done in Fig. 4.9-a. This clearly shows the nonlinear nature of the mode seeking algorithm.

4.4 Example 2

As a second practical application of mode seeking algorithm in pattern recognition, the algorithm was applied to a single feature, three class crop identification problem. The three crop categories were corn, soybeans, and fallow, designated respectively as class 1, class 2, and class 3. There were 448 samples, the signature being the transmission through the green

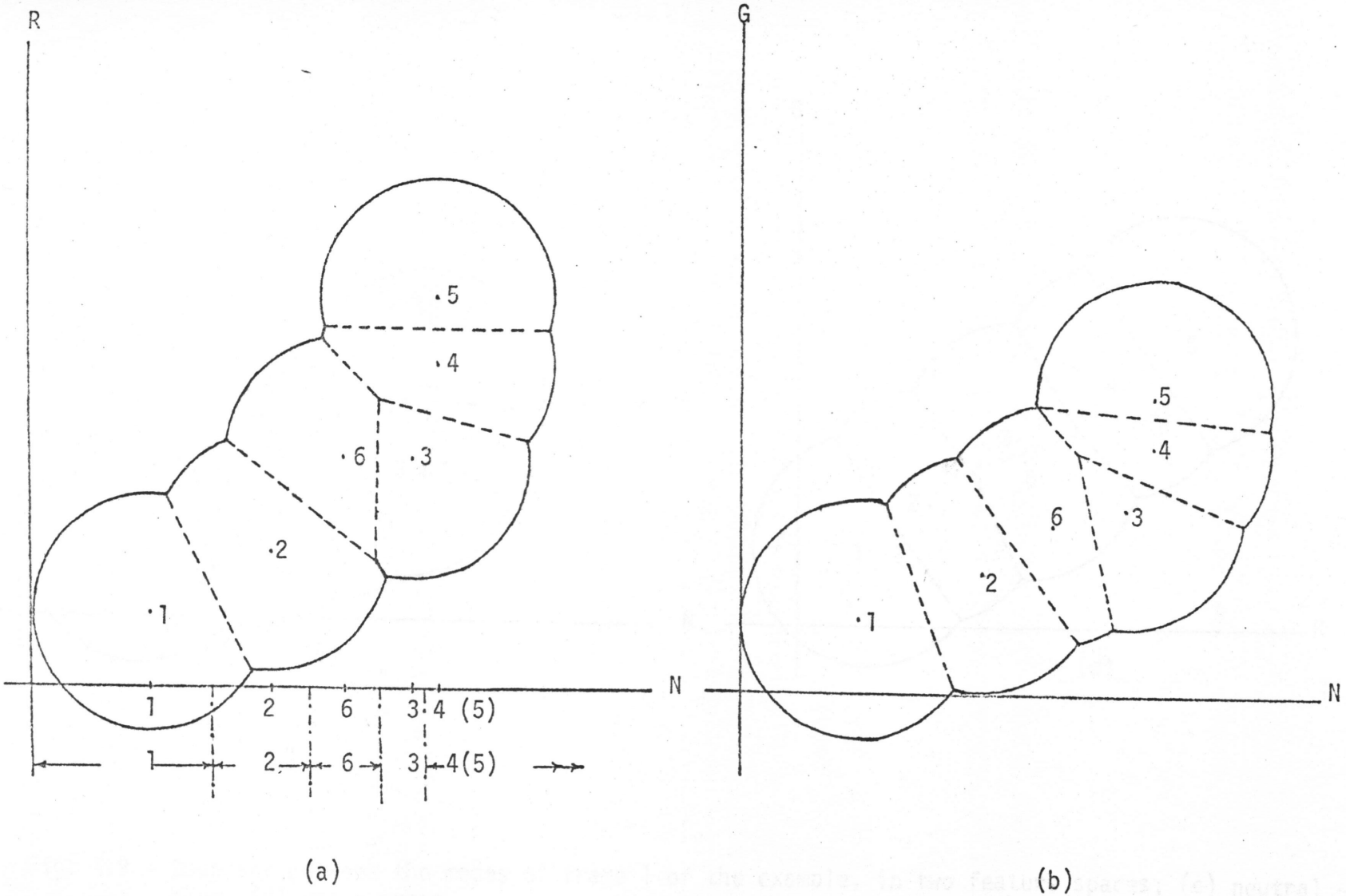


Fig. 4.9 - Boundary between the modes of frame 1 of the example, in two feature spaces; (a) neutral-red, (b) neutral - green.

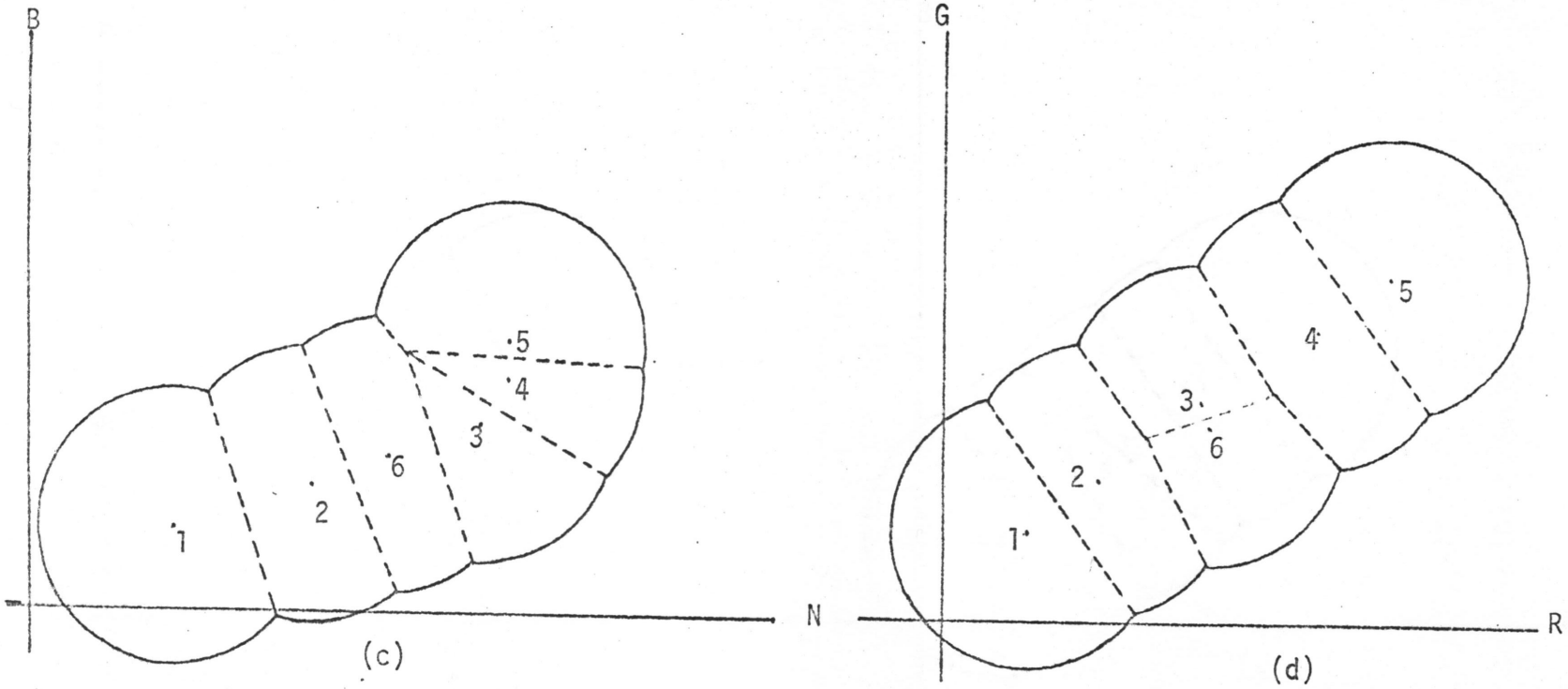


Fig. 4.9 - Boundary between the modes of frame 1 of the example, in two feature spaces; (c) neutral - blue, (d) red - green.

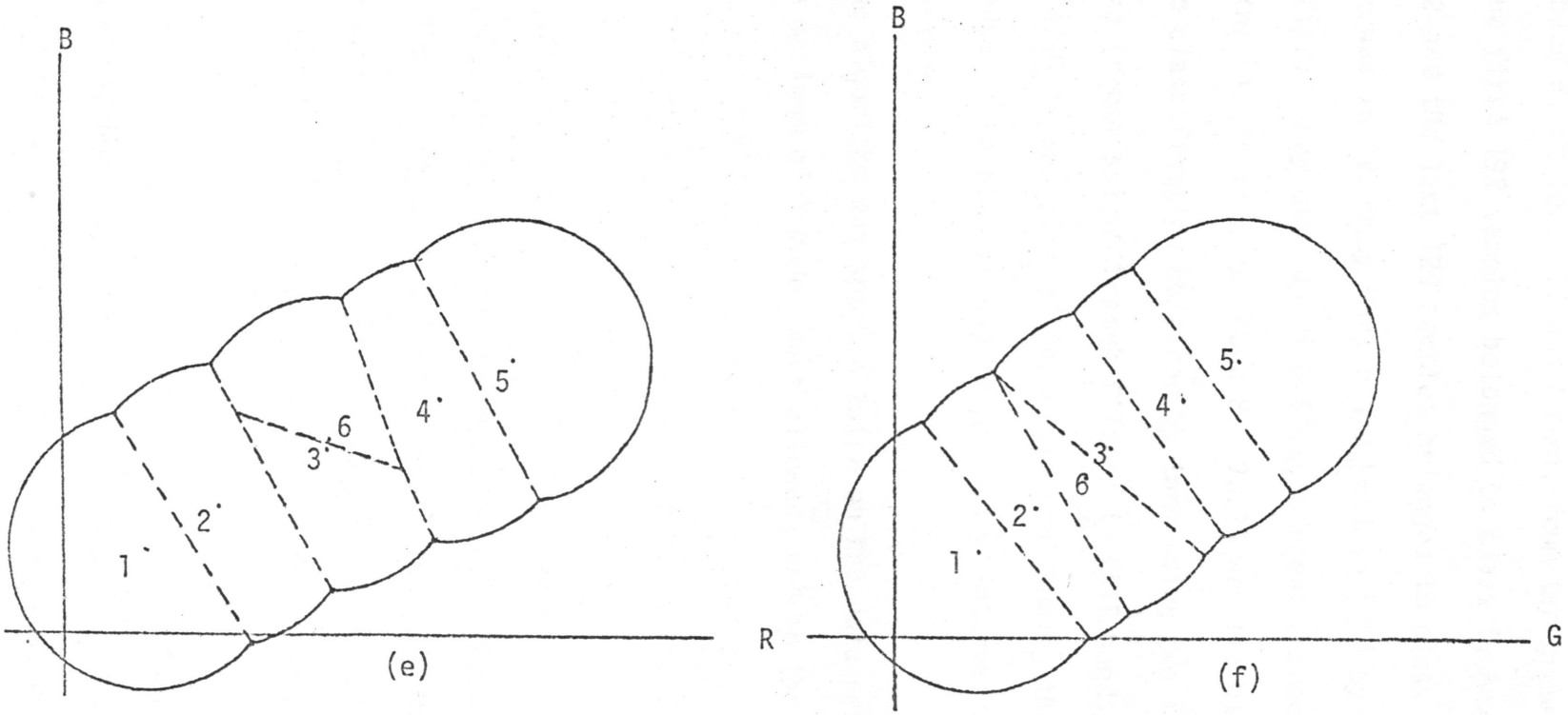


Fig. 4.9 - Boundary between the modes of frame 1 of the example, in two features spaces; (e) red - blue, (f) green - blue.

(ERTS band 6) filter. It was known, from the ground truth information, that the first 192 samples belonged to class 1, next 128 belonged to class 2 and the last 128 samples belonged to class 3. This very problem was analyzed by Horton and Heilman [28] by means of K-class classifier. They came up with 87.28 percent correct classification using the single feature, band 6. 93.75 was the highest percentage correct classification obtained by them using two features which were the transmission through filters in ERTS bands 5 and 6. Figure 4.10 is the plot of the histogram showing the frequency of occurrence of samples in any transmission interval against the transmission.

The algorithm was applied twice on the data set. In the first case, a maximum of 3 modes were allowed, and in the second case, a maximum of 6 modes were allowed. In both cases, the samples were classified into classes. In the first case, the algorithm came up with three modes with a classification tally of 190, 128, and 130 samples, respectively, in mode 1, mode 2, and mode 3. The modes 1, 2, and 3 correspond to the classes 1, 2, and 3, respectively. Fig. 4.11 shows the mode centers, which also form the class means and the boundary between the modes or classes. Note that the boundary between the two modes is nothing but the midpoint between the two modes. The classification results revealed that two samples belonging to class 1 were classified as belonging to class 3. In the histogram plot, these two points lie in the region where the two classes 1 and 3 overlap. The classification result is shown

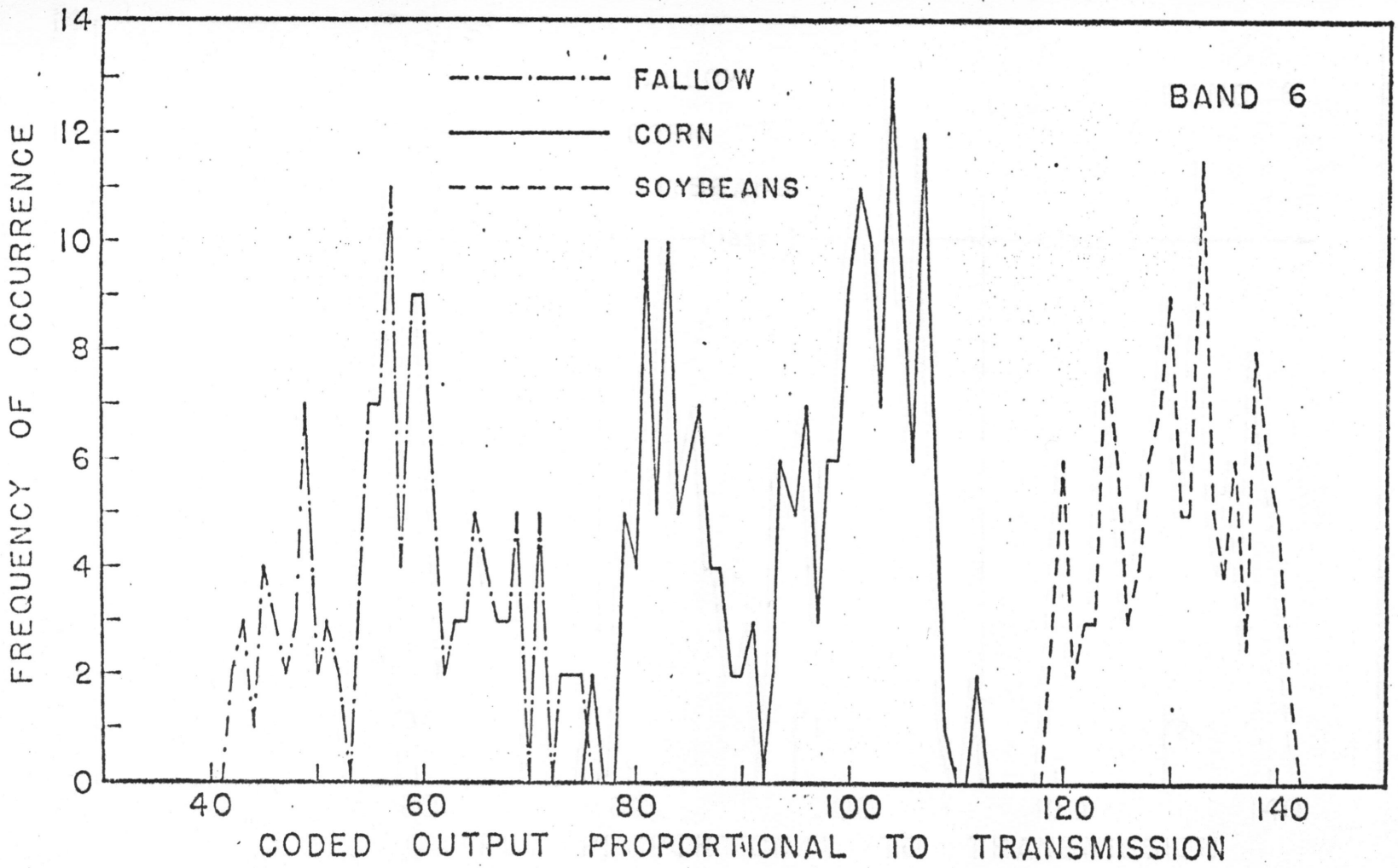


Fig. 4.10 - Histogram for the problem in example 2..

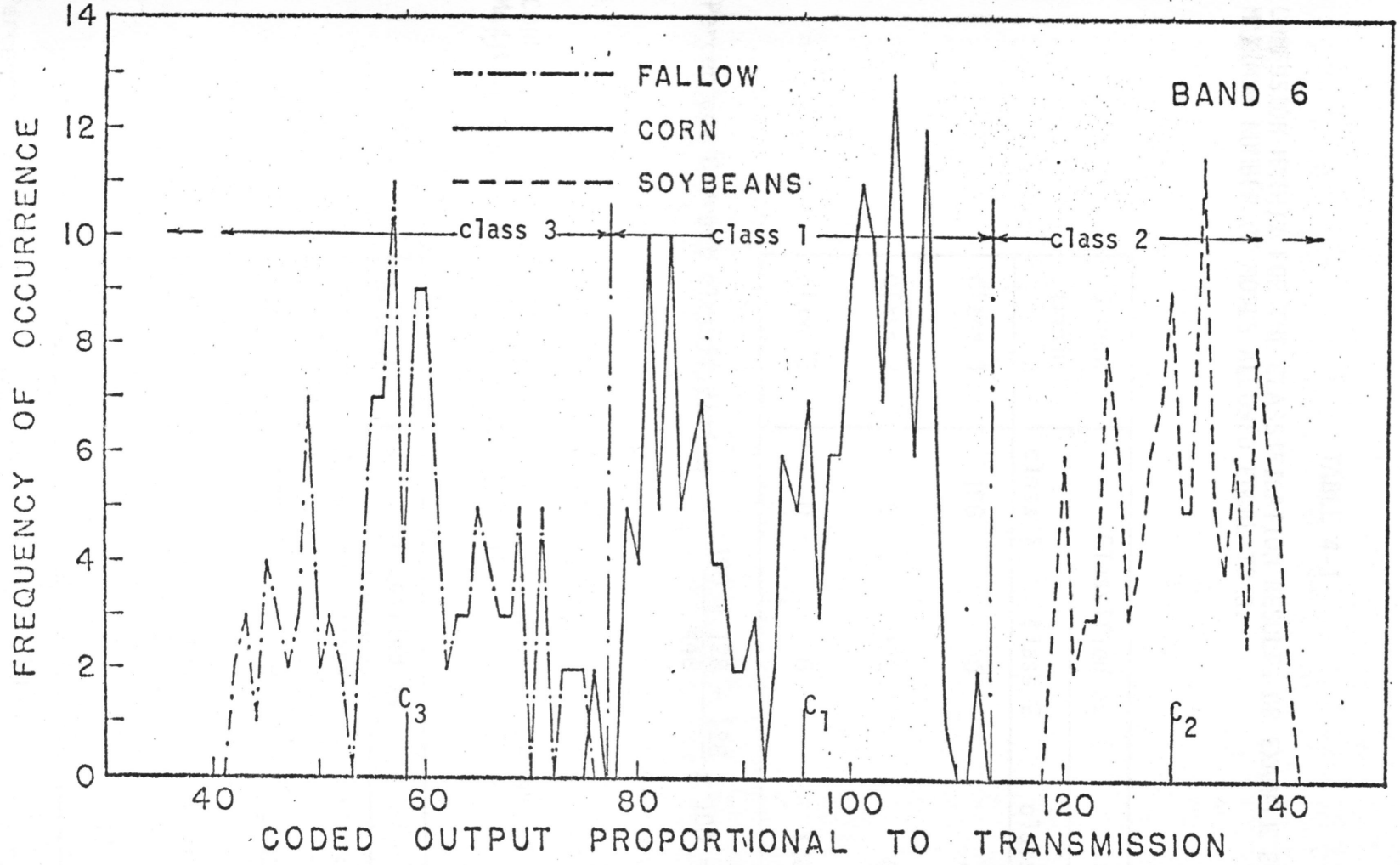


Fig. 4.11 - Diagram showing the mode centers C_1, C_2, C_3 and the boundary between classes, for case 1 in example 2 with 3 modes allowed.

TABLE 4-1

CONFUSION MATRIX FOR THE CLASSIFICATION RESULTS OF EXAMPLE 2 WITH MAXIMUM NUMBER OF MODES ALLOWED EQUAL TO 3.

Ground truth	Classified as		
	class 1	class 2	class 3
class 1	190	0	2
class 2	0	128	0
class 3	0	0	128

Percentage Correct classification $= \frac{190 + 128 + 128}{448} \times 100 = 99.6\%$

TABLE 4-2

CONFUSION MATRIX FOR THE CLASSIFICATION RESULTS OF EXAMPLE 2 WITH MAXIMUM NUMBER OF MODES ALLOWED EQUAL TO 6

Ground truth	Classified as		
	class 1	class 2	class 3
class 1	192	0	0
class 2	2	126	0
class 3	5	0	123

Percentage correct classification $= \frac{192 + 126 + 123}{448} \times 100 = 98.5\%$

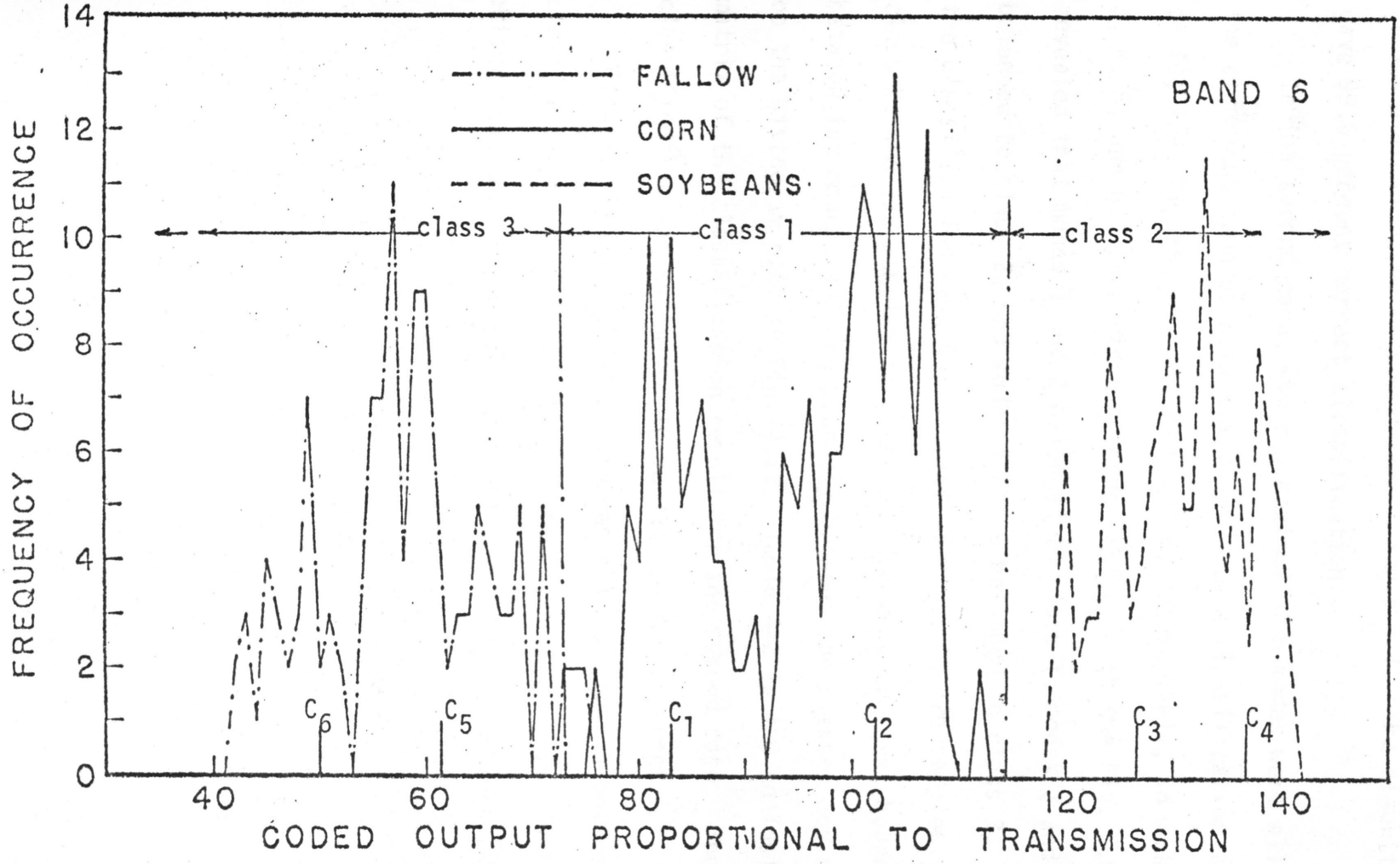


Fig. 4.12 - Diagram showing the mode centers $C_1, C_2, C_3, C_4, C_5, C_6$ and the boundary between classes, for case 2 in example 2 with 6 modes allowed.

in the form of a confusion matrix in Table 4-1. The algorithm gave 99.6 percent correct classification.

In the second case, where a maximum of 6 modes were allowed, the algorithm came up with six modes with a classification tally of 70, 129, 71, 55, 88, and 35 samples, respectively, in modes 1, 2, 3, 4, 5, and 6. Comparison of mode centers with the Fig. 5.9 revealed that modes 1 and 2 belonged to class 1, modes 3 and 4 belonged to class 2, and modes 5 and 6 belonged to class 3. Thus, the classification was, 199 samples to class 1, 126 samples in class 2, and 123 samples in class 3. The mode centers, boundary between the modes, and the boundary between the classes are shown on the histogram plot in Fig. 4.12. Table 4-2 is the confusion matrix of the classification result for the second case. The classification was estimated to be 98.5 percent correct.

Case 1, with a maximum of 3 modes allowed, gave a better classification result. Results of both case 1 and case 2 were better than the best result obtained by Horton and Heilman [28].

CHAPTER 5. MODE SEEKING USED IN K-CLASS CLASSIFIER

5.1 K-Class Classifier

Classification is defined as the assignment of an event to one of K mutually exclusive subsets of events, called classes [29]. An event is characterized by a set of measurable parameters, "features" or "attributes", and by an abstract set of parameters not amenable to direct measurement, the "class parameters". An event is then described by (d,x), where d is the class vector and x is an attribute vector. All events belonging to class i will then have as a description (d_i, x_i).

Classification may now be defined as the determination of the class vector, d given only the original measurements, (x) for a given event. This requires the estimation of a matrix B which transforms an event from the N-dimension measurement space into a K-dimension decision space as shown in Fig. 5.1. The classifier is envisioned as a least squares mapping of the attribute vector, x^e, in feature space, toward the class vector d_e, in the decision space. An event is then assigned to class i if d_e is closer in a Euclidean sense to d_i than to any other class vector.

The determination of the operator B, which results in a least squares mapping toward the orthonormal class vectors, is made by selecting B such that

$$\Delta_B \overline{[d_e - BX_A^e]^T \; [d_e - BX_A^e]} = 0 \qquad (5\text{-}1)$$

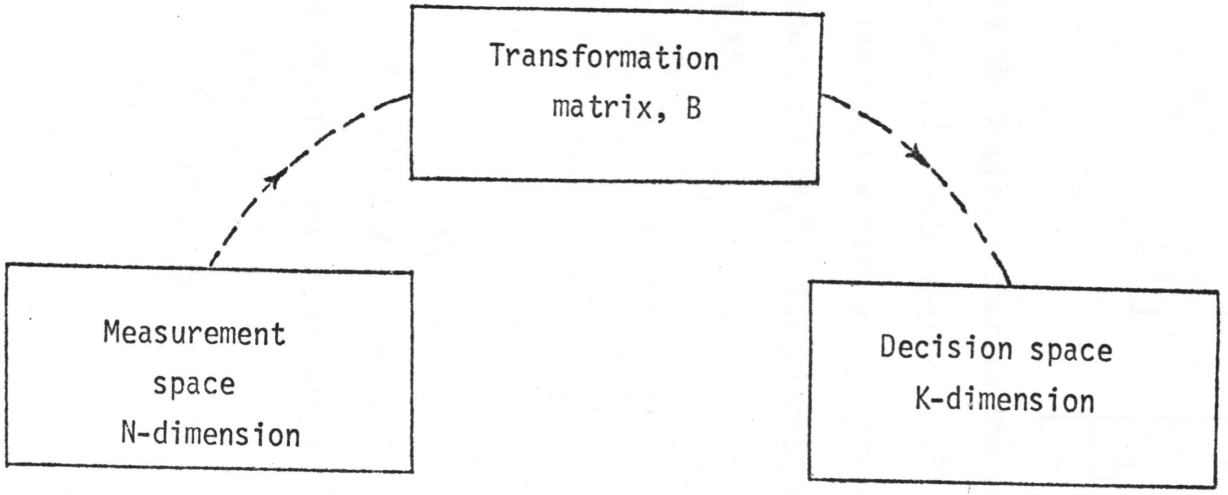


Fig. 5.1 - Transformation of an event from the measurement space to the decision space.

The horizontal bar denotes average over all events, and Δ_B is the gradient with respect to the operator B. X_A is the augmented attribute vector,

$$X_A = \begin{bmatrix} X \\ \hline -1 \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \\ \hline -1 \end{bmatrix} \qquad (5\text{-}2)$$

and $d = BX_A^e$ is the mapped vector in the decision space corresponding to the event whose attribute vector is x^e. An event will be assigned to class i if the class vector d_i is closest in an euclidean sense to the mapped vector, $d = BX_A^e$. The classification rule [29] is to choose class i if

$$[d_i - d]^T [d_i - d] < [d_j - d]^T [d_j - d] \qquad (5\text{-}3)$$

$$\text{for all } j \neq i,\ j = 1, 2, \ldots k$$

which reduces to selecting class i if

$$d_i > d_j \qquad (5\text{-}4)$$

$$\text{for all } j \neq i,\ j = 1, 2, \ldots k$$

where d_i is the decision vector or the estimated class vector for the i^{th} class and specifically

$$d_i = P_i [\overline{X}^i - \overline{X}]^T \phi^{-1} [X - \overline{X}] + P_i \qquad (5\text{-}5)$$

where,

P_i = a priori probability of class i

$\overline{X}^i$ = mean attribute vector for class i

$\overline{X}$ = mean attribute vector obtained by averaging over all classes

ϕ^{-1} = inverse of covariance matrix of attributes,

$$([\phi] = [X - \overline{X}]\,[X - \overline{X}]^T = [\overline{XX^T} - \overline{X}\,\overline{X}^T])$$

$[\ ]^T$ denotes transpose

x = attributes vector of the event being classified

Note that the above decision rule is equivalent to the rule: select class i, if,

$$P\ (\text{class } i/(x_m)) > P\ (\text{class } j/(x_m)) \qquad (5\text{-}6)$$

all $j \neq i$, $j = 1, 2, \ldots k$

Where $P\ (\text{class } 1/(x_m))$ is the least squares estimate of the a postriori probability of class 1, and is given by

$$P\ (\text{class } 1/(x_m)) = d_1 = P_1\,[\overline{X}^1 - \overline{X}]^T\,\phi^{-1}\,[X - \overline{X}] + P_1 \qquad (5\text{-}7)$$

Thus d_i can be thought of as a measure of a postriori probability of class i given an event of attributes x^e. Consequently, the classifier is an approximate a postriori probability computer.

5.2 Implementation of the Classifier

It must be emphasized that the only statistics learned for implementation of the classifier are the mean of class attributes, the mean attributes, and the average covariance matrix. No assumptions as to form of the joint probability distributions of the attributes are used in influencing the classifier, and in that sense it may be considered a non-parametric classifier.

Before the classifier can attempt to classify an event into one of the K-classes, it needs to be trained by means of a training set of data. This involves the availability of the following:

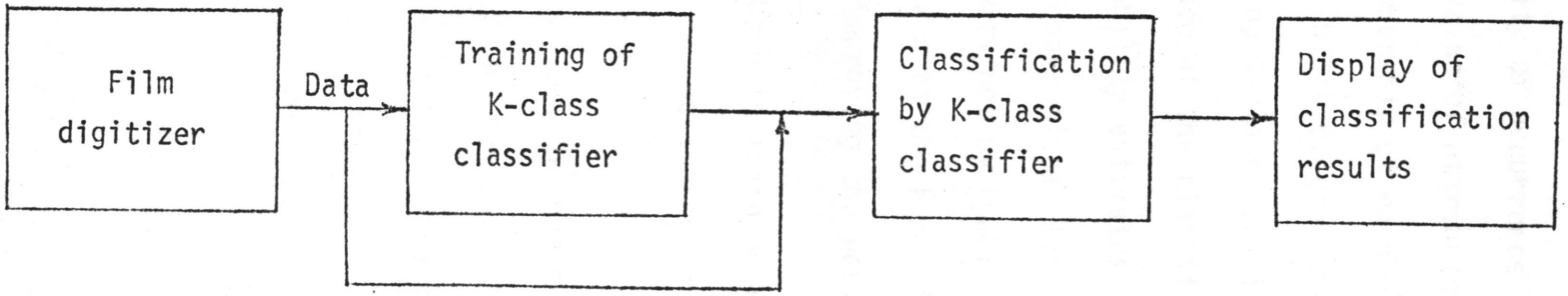


Fig. 5.2 - Data classification by K-class.

1. A reasonably good approximate value for the a priori probability of occurrence for each class.
2. Average value of attributes for each class. This means that class membership of each event in the training set is known.

Fig. 5.2 is a block diagram showing the various steps in data classification using K-class classifier.

The performance of the classifier depends on the accuracy of the a priori probability estimates and the identification of the training set. In conventional types of classifiers these two requirements are obtained by human judgement. The use of mode seeking can do away with the human intervention task normally required, thereby improving the performance of the classifier.

5.3 Mode Seeking Used in Training K-Class

Recall that mode seeking selects a number of modes in a given data set, calculates the centers of each mode and the number of points in each mode. Let the number of modes equal the number of classes, mode centers be the mean class attributes. The number of points in any class divided by the total number of points is the a priori probability of occurrence of that class. The training of the K-class classifier can use the results from the mode seeking algorithm. The following training steps are proposed:

1. Calculate the a priori probability P_i of occurrence of class i, where

$$P_i = \frac{\text{Number of points in mode } i}{\text{Total number of points}}; \quad i = 1, 2, \ldots K \text{ classes.}$$

2. Calculate the average of the attributes over all classes

$$\bar{X}_j = \frac{\sum_{i=1}^{\text{K-class}} (X_j \text{ of mode } i)\ (\text{Number of points in mode } i)}{\text{Total number of points}}$$

for $j = 1, 2, \ldots, N$ features.

3. Calculate the attributes covariance matrix

$$\phi = [\overline{XX^T} - \bar{X}\bar{X}^T]$$

4. Calculate ϕ^{-1}, the inverse of the covariance matrix

5. Calculate the B matrix

$$[B] = [\bar{X}^i - \bar{X}]^T \phi^{-1}$$

6. Calculate the C matrix

$$[C] = [B]\ [\bar{X}]$$

7. Calculate the elements d_i of the class vector d for the attribute vector [X], where

$$d_i = [BX - C_i + 1]\ P_i;\ i = 1, 2, \ldots, \text{K-class}$$

8. Assign the event [X] to class i for which d_i is maximum.

Fig. 5.3 is the block diagram form of data classification by K-class classifier, that is being trained by mode seeking results.

An existing computer program for the implementation of K-class classifier [30] was modified such that the results of mode seeking could be used to train the classifier. Note that for any event of attributes [X], the sum of the estimated elements d_i of the class vector d over all classes is equal to 1. From this it follows that

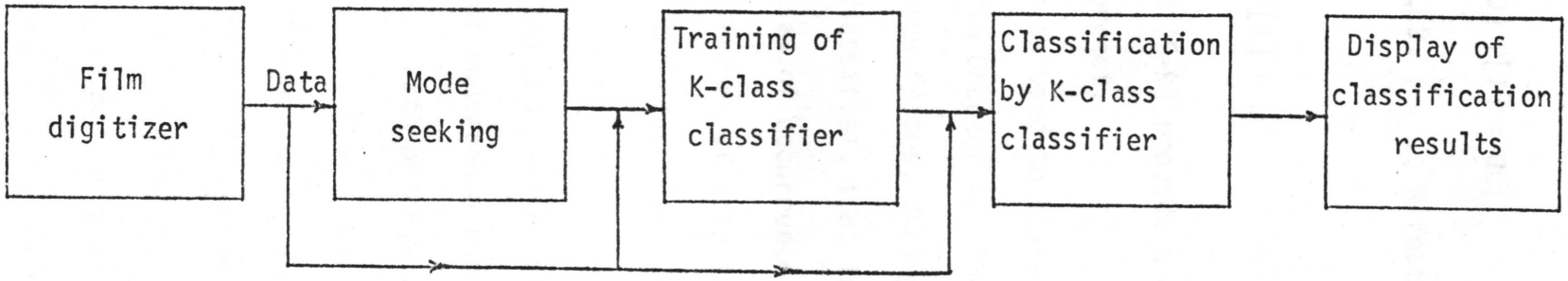


Fig. 5.3 - Data classification by K-class using mode-seeking results.

$$\sum_{I=1}^{K \text{ class}} B[I,J]\, P[I] = 0 \qquad (5\text{-}8)$$

for all J = 1, 2, . . ., N features

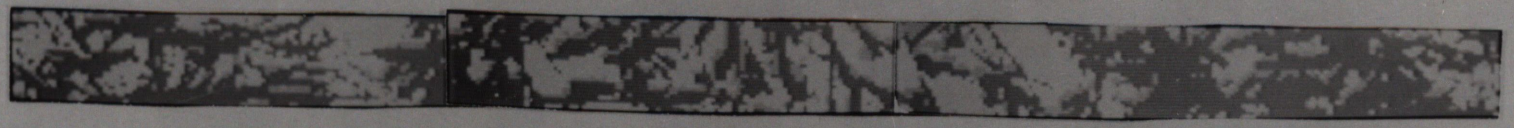
and

$$\sum_{I=1}^{K \text{ class}} C[I]\, P[I] = 0 \qquad (5\text{-}9)$$

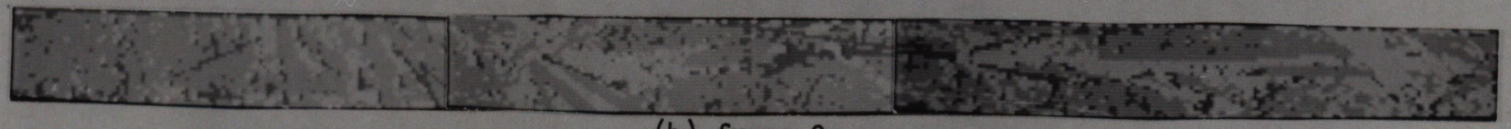
Equations (5-8) and (5-9) provide a good check on whether the calculations are correct.

The classifier is used to classify the three frames of data that were analyzed in Chapter 4. The results obtained, for the combined data of three frames, during mode seeking is used to train the K-class classifier, i.e. to calculate the B-matrix and the a priori probabilities of occurrence of each class. Based on this training, the three frames of data are classified.

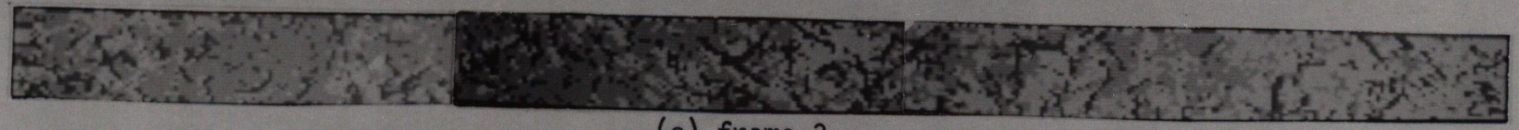
Fig. 5.4 shows the classification results by the K-class classifier without the use of mode seeking. The classifier is trained by assuming that there are three classes of equal a priori probability of occurrence and that frame 1 data belongs to class 1 alone, frame 2 data belongs to class 2 alone, and frame 3 data belongs to class 3 alone. Fig. 5.5 shows the K-class classification results based on training by use of the mode seeking algorithm. Comparison of Fig. 5.4 and 5.5 shows that K-class coupled with mode seeking does a better classification job than K-class by itself. This is because, the assumption that data from frame 1, frame 2, and frame 3 belong entirely to class 1, class 2, and class 3, respectively, was in error. For example, in addition to class 1, frame 1 also



(a) frame 1

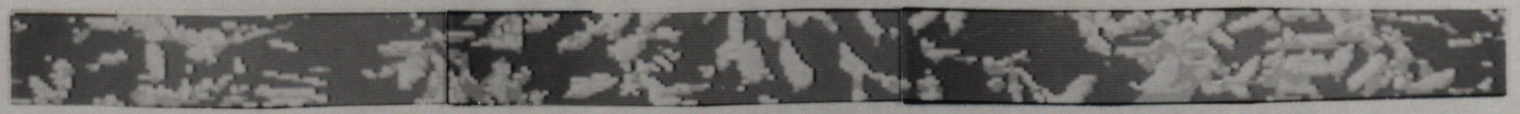


(b) frame 2

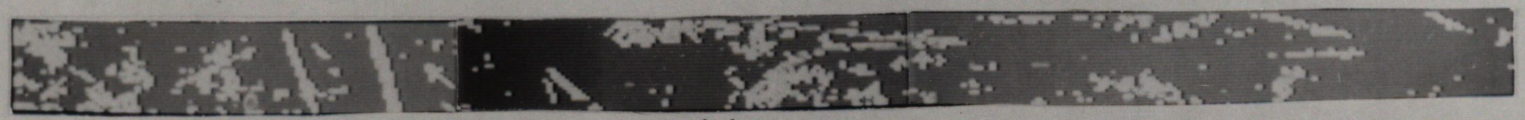


(c) frame 3

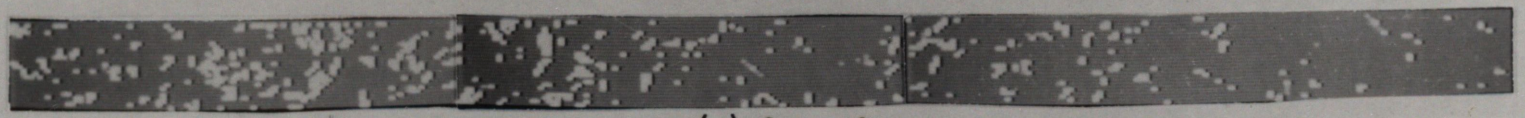
Fig. 5.4 - Classification results for example 1 by K-class classifier assuming that frame 1 is class 1, frame 2 is class 2, and frame 3 is class 3.



(a) frame 1



(b) frame 2



(c) frame 3

Fig. 5.5 - Classification results for example 1 by K-class classifier, which was trained by mode seeking results.

contained class 2 and class 3 in the background. Overall, the classification of three frames by K-class classifier is not very satisfactory. This may be due to 1) the K-class is designed to minimize the error introduced in mapping the samples from the measurement space into the decision space; it does not minimize the classification error, and 2) training the linear K-class classifier by the results of the nonlinear mode seeking algorithm.

CHAPTER 6. CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

Mode seeking algorithm gives good classification results for data with nonoverlapping modes. However, it does recognize the overlapping modes in a given set of data. It is necessary to exercise some caution in applying the mode seeking algorithm on any given set of data. For example, if it is desired to classify a given set of data into various crop categories, and if the ground region from which the data is collected contains other types of objects also, and if the major contribution in forming the data is from these other objects, then the algorithm will fail to come up with the desired classification. Such an application might lead one to draw wrong conclusions about the usefulness of the algorithm.

6.2 Recommendations

While performing this research, the value of cluster threshold was the same for all the modes. This amounts to assuming that each mode has the same variance. This may not be true in most practical problems. It is conceivable that different categories can have different variances in spectral readings about their respective means. It would be worthwhile, if possible, to allow different thresholds to different modes. Of course, this would require a prior knowledge of the categories involved and the structure of the data.

In addition, for a set of data with N features, mode seeking algorithm should be applied to various possible feature combinations,

such as, taking one feature at a time, two features at a time and so on. This enables one to find the best feature combination giving optimum classification results. New features can be created by taking the difference, sum, quotient of the various feature pairs.

6.3 Research Contribution

1. The mode seeking algorithm is modified such that each iteration will consider all the samples on the given data set instead of the centers of the modes found in the previous iteration.
2. A computer program is developed to classify the samples into modes and to identify the samples with the cluster codes to which they belong.
3. A method of training the K-class classifier by the results of mode seeking algorithm is presented.
4. The mode seeking algorithm is successfully applied, to a set of data collected by a camera at the ground level, by combining both the multiband approach and the multistage photographs approach.
5. The mode seeking algorithm is used to classify a set of ERTS data representing crops.
6. A set of rangeland data, collected through a camera at the ground level, is classified by K-class classifier being trained by the results of mode seeking algorithm.

BIBLIOGRAPHY

[1] G. Nagy and J. Tolaba, "Nonsupervised Crop Classification Through Airborne Multispectral Observations," IBM Journal of Research and Development, March 1972, pp. 138-151.

[2] G. Ball, "Data Analysis in the Social Sciences," Proceedings of Fall Joint Computer Conference, Vol. 27, Part I, 1965, p. 553.

[3] K. Fukunaga and L. G. Koontz, "A Criterion and an Algorithm for Grouping Data," IEEE Transactions on Computers, Vol. C-19, No. 10, October 1970, pp. 917-923.

[4] G. N. Lance and W. T. Williams, "Computer Programs for Monoletic Classification (association analysis)," Computer Journal, Vol. 8, October 1965, pp. 246-249.

[5] R. L. Mattson and J. E. Dammann, "A Technique for Determining and Coding Subclasses in Pattern Recognition Problems," IBM Journal of Research and Development, July 1965, pp. 294-302.

[6] M. Wirth, G. Estabrook and D. Rogers, "A Group Theory Model for Systematic Biology, With an Example for the Oncidunea," Systematic Zoology, Vol. 15, No. 1, March 1966, pp. 59-69.

[7] H. P. Friedman and J. Rubin, "On Some Invarient Criteria for Grouping Data," Journal of American Statistical Association, Vol. 62, December 1967, pp. 1159-1179.

[8] R. E. Bonner, "On Some Clustering Techniques," IBM Journal, Vol. 8, January 1969, pp. 22-32.

[9] H. Teicher, "Identifiability of Mixtures of Product Measures," Annal of Mathematical Statistics, Vol. 38, 1967, pp. 1300-1302.

[10] H. Teicher, "Identifiability of Mixtures of Product Measures," Annal of Mathematical Statistics, Vol. 34, 1963, pp. 1265-1269.

[11] H. Teicher, "Identifiability of Mixtures of Product Measures," Annal of Mathematical Statistics, Vol. 32, 1961, pp. 244-248.

[12] S. J. Yakowitz, "Unsupervised Learning and The Identification of Finite Mixtures," IEEE Transactions on Information Theory, Vol. IT-16, May 1970, pp. 330-338.

[13] S. J. Yakowitz, "A Consistent Estimator for The Identification of Finite Mixtures," Annal of Mathematical Statistics, Vol. 40, 1969, No. 5, pp. 1728-1735.

[14] S. J. Yakowtiz and J. D. Spragins, "On Identifiability of Finite Mixtures," Annal of Mathematical Statistics, Vol. 39, 1968, No. 1, pp. 209-214.

[15] D. F. Stanat, "Unsupervised Learning of Mixtures of Probability Functions," Pattern Recognition, L. Kanal Ed. Washington, DC: Thompson, 1968, pp. 357-389.

[16] S. C. Fralick, "Learning to Recognize Patterns Without a Teacher," IEEE Transactions on Information Theory, Vol. IT-13, January 1967, pp. 57-64.

[17] J. Spragins, "Learning Without a Teacher," IEEE Transactions on Information Theory, Vol. IT-12, April 1966, pp. 223-230.

[18] E. A. Patrick and J. C. Hancock, "Nonsupervised Sequential Classification and Recognition of Patterns," IEEE Transactions on Information Theory, Vol. IT-12, July 1966, pp. 362-372.

[19] E. A. Patrick, "On a Class of Unsupervised Estimation Problems," IEEE Transactions on Information Theory, Vol. IT-14, May 1968, pp. 407-415.

[20] I. Gitman and M. D. Levine, "An Algorithm for Detecting Unimodal Fuzzy Sets and Its Application as a Clustering Technique," IEEE Transactions on Computers, Vol. C-19, No. 7, July 1970, pp. 583-593.

[21] G. Sebestyen, Decision-making Process in Pattern Recognition, The Macmillan Company, New York.

[22] M. W. Blasgen, "Pattern Recognition and The Estimation of Modes," IBM Research, TC 3155, November 1970.

[23] L. G. Koontz and K. Fukunaga, "A Nonparametric Valley-Seeking Technique for Cluster Analysis," IEEE Transactions on Computers, February 1972, pp. 171-178.

[24] R. M. Haralick and I. Dinstein, "An Iterative Clustering Procedure," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-1, No. 3, July 1971, pp. 275-289.

[25] K. S. Jones and D. Jackson, "Current Approaches to Classification and Clump-Finding at the Cambridge Language Research Unit," Computer Journal, Vol. 10, No. 1, 1967, pp. 27-31.

[26] G. H. Ball and D. J. Hall, "ISODATA, A Novel Method of Data Analysis and Pattern Classification," Sanford Research Institute, Menlo Park, California, Technical Report, 1965.

[27] G. Sebestyen and J. Edie, "An Algorithm for Non-Parametric Pattern Recognition," IEEE Transactions on Electronic Computers, Vol. EC-15, No. 6, December 1966, pp. 908-915.

[28] M. L. Horton and J. L. Heilman, "Crop Identification Using ERTS Imagery," Technical Report Presented at the Symposium on Significant Results Obtained From ERTS-1, March 5-9, 1973.

[29] N. R. Zagalsky, "A New Formulation of a Classification Procedure," M.S. Thesis, University of Minnesota, March, 1968.

[30] D. V. Serreyn and G. D. Nelson, "The K-Class Classifier," SDSU, RSI Interim Technical Report, RSI-73-08, April 1973.