Clustering Concepts in Automatic Pattern Recognition

1975

Patricia J. DeFilipps

University of Central Florida

Find similar works at: https://stars.library.ucf.edu/rtd

University of Central Florida Libraries http://library.ucf.edu

Part of the Engineering Commons

STARS Citation

https://stars.library.ucf.edu/rtd/145

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.
CLUSTERING CONCEPTS IN AUTOMATIC PATTERN RECOGNITION

BY

PATRICIA J. DeFILIPPS
B.S., University of Florida, 1974

RESEARCH REPORT

Submitted in partial fulfillment of the requirements for the degree of Master of Science in the Graduate Studies Program of Florida Technological University

Orlando, Florida
1975
CLUSTERING CONCEPTS IN AUTOMATIC PATTERN RECOGNITION

ABSTRACT

During the past decade and a half, there has been a considerable growth of interest in problems of pattern recognition. Contributions to the growth have been from many of the disciplines including statistics, control theory, operations research, biology, linguistics, and computer science. One of the basic approaches to pattern recognition is cluster analysis, in which various methodologies may be successfully employed. It is the purpose of this research report to investigate some of the basic clustering concepts in automatic pattern recognition.

[Signature]
Director of Research Report
ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Dr. Klaus Lindenberg, whose enthusiasm originally interested me in this subject and who has provided valued support and direction. I would also like to express my gratitude to Dr. George Schrader and Dr. David Clapp for the guidance they have provided. Many thanks goes to my parents, Anne and Fred DeFilipps for the support they have extended.
# TABLE OF CONTENTS

| LIST OF TABLES | vii |
| LIST OF FIGURES | viii |

## SECTIONS

1. PATTERN RECOGNITION SYSTEMS ........................................ 1
   1.1. Description of Components ..................................... 1
   1.2. Design Concepts for Automatic Pattern Recognition .............. 2
   1.3. Applications of Pattern Recognition .............................. 4

2. THE CLUSTERING CONCEPT ........................................... 8
   2.1. Fundamentals ................................................. 8
   2.2. Similarity Measures .......................................... 10
   2.3. Criterion Functions ........................................... 11
   2.4. Iterative Approaches .......................................... 12
   2.5. Two Clustering Algorithms .................................... 13
      2.5.1. K-Means Algorithms ...................................... 13
      2.5.2. Maximin Distance Algorithms ............................... 15

3. A COMPUTER GENERATED EXAMPLE .................................... 18
   3.1. Introduction ................................................... 18
   3.2. Description of the Program ..................................... 18
   3.3. Analysis of Results ............................................ 22
   3.4. Extensions ...................................................... 24
APPENDIX ......................................................... 28
REFERENCES ..................................................... 57
BIBLIOGRAPHY .................................................... 58
LIST OF TABLES

1. List of Cluster Centers for Data Representing Corn, Soy beans, and Wheat ............... 22
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Samples in Two Dimensional Feature Space for Corn, Soy beans and Wheat</td>
<td>6</td>
</tr>
<tr>
<td>2.</td>
<td>Clustering Properties of Data for Professional Jockeys and Wrestlers</td>
<td>9</td>
</tr>
<tr>
<td>3.</td>
<td>Block Diagram for the K-Means Clustering Algorithm</td>
<td>20</td>
</tr>
<tr>
<td>4.</td>
<td>Block Diagram for the Maximin Clustering Algorithm</td>
<td>23</td>
</tr>
</tbody>
</table>
1. PATTERN RECOGNITION SYSTEMS

1.1. Description of Components

A typical pattern recognition system consists of three components: (a) a sensing device, (b) a feature extractor and (c) a classification component. The physical world can be represented by a continuum of parameters, hence is infinite in dimensionality. The sensing device senses part of the physical world, thereby generating finite data for input into the measurement or pattern space, hence the pattern space is finite in dimension. If we assume that the sensing device has $n$ elements, and the measurements yield information in the form of real numbers, then the measurements can be arranged in the form of a $n$-tuple pattern vector $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ where $x_i$ is the $i$th measurement corresponding to the $i$th element of the sensing device. This pattern vector can be thought of as a point in $n$-dimensional Euclidean space.

The feature extractor or preprocessing component transforms the $n$-dimensional pattern vector into a $K$-dimensional feature vector, $K$ less than $N$, while still maintaining the discriminatory attributes inherent in the data for classification purposes. The feature vector may be viewed as a $k$-tuple vector $\mathbf{y} = (y_1, y_2, \ldots, y_K)$. Thus a feature space is postulated of dimensionality $K$ less than $N$ in which classification rules can be applied in reasonable amounts.
of time. This provides a method of reducing processing cost by reducing the number of parameters to be analyzed.

The classification component of the system identifies the pattern class to which these data belong. When the measurements are similar in nature they form a group of similar patterns which may be considered members of the same class. The set of patterns belonging to the same class represent an arrangement of points scattered within some region of the pattern space. Hence these classes \((C_1, C_2, \ldots, C_M)\) may be thought of as sets of Euclidean space containing the pattern vectors. The classification space is the decision space in which one of the \(M\) classes has been selected, and hence is of dimension \(M\). Typically the classification algorithms which define the space are designed to implement some optimizing criterion function.

1.2. Design Concepts for Automatic Pattern Recognition

Three basic design concepts for automatic pattern recognition are: (a) membership roster, (b) common property and (c) clustering. The concept selection depends upon the characterization and definition of the pattern classes (1):

Pattern classes characterized by a roster of its members suggests automatic pattern recognition by matching techniques which are the foundation of this process. The patterns belonging to the established pattern classes are stored in the system. When an unknown pattern is presented to the system it is compared to the
stored patterns. If it matches one of the stored patterns, then it is classified as belonging to the pattern class. A simple example is character recognition where standard character fonts are stored, and various incoming characters are compared (2). As long as the incoming characters are not distorted by noise due to smear, bad inking, porous paper, etc., satisfactory results may be obtained.

Pattern classes characterized by common properties shared by all of its members suggest automatic pattern recognition by the detection and selection of similar features. The primary assumption underlying this technique is that patterns belonging to the same pattern class have certain similar attributes common to all patterns belonging to that particular class. These common properties can be stored in the system. When an incoming pattern is presented, its features are extracted and compared to the stored features in the system. The storage requirement for this method is considerably less than for the membership roster described above.

A pattern class can be characterized by its clustering properties in the pattern space when the patterns of a class are vectors whose components are real numbers. This suggests pattern recognition via the clustering concept. When the classes are relatively far apart, simple clustering algorithms may be effectively implemented. Two such clustering algorithms will be presented in the next section of this report. However, if the clusters overlap for various reasons such as the presence of measurement noise or a deficiency in observed information, more sophisticated methodologies
for partitioning the pattern or feature space must be employed (3).

1.3. Applications of Pattern Recognition

Pattern recognition has numerous applications in the medical field. For example, (a) automatic classification of photomicrographs of tissue cells which are used in blood tests, cancer tests, brain tissue studies, (b) recognition of chromosome properties for genetic studies and (c) analysis of clinical data from electrocardiograms and electroencephalograms can be classified using pattern recognition techniques (4).

Another application that has received a great deal of attention is the automatic classification of remotely sensed data. Remote sensing means measuring an object from a distance. It essentially concerns the measurement of electromagnetic energy which is emitted, reflected, or scattered by objects. Different objects return different types and amounts of energy and detecting these differences enables identification of the objects under consideration. One popular remote sensing device is a multispectral scanner which may be mounted in either a spacecraft or an aircraft. The multispectral scanner can be built to cover the optical wave length portion ranging from 0.3 to 15 micrometers. The wavelengths shorter than 0.4 micrometers are the ultra violet region. The visible region to which the human eye is sensitive, is 0.4 to 0.7 micrometers. The portion above the visible range is the infrared region which is broken up into two sections; the reflective infrared ranging from
0.7 to approximately 3.0 micrometers and the emmissive or thermal infrared ranging from 3.0 to 15.0 micrometers. In this emmissive infrared region of the spectrum, energy is emitted from the body due to its thermal activity or heat rather than reflected from it. The scanner responds to light in selected wave-length bands. The optics of the system refract this beam of energy separating it into components according to the wave length. The response in each wave length band may be stored on magnetic tape. Consider, for purposes of illustration a multi-spectral scanner that responds to light in the 0.58 - 0.62, 0.66 - 0.72 and .80 - 1.00 micrometers wavelength bands over a field of corn, soy beans and wheat. These ranges are in the green, red, and infrared bands respectively. The ground region produces three images simultaneously, one in each color range, hence each point in the region is characterized by three color components. The data for each point can be expressed as the pattern vector \( \mathbf{x} = (x_1, x_2, x_3) \). After this data has been preprocessed suppose the dimensionality has been reduced by one component and the data can be represented by the feature vector \( \mathbf{y} = (y_1, y_2) \). Figure 1 below represents the data for the field of corn, soybeans, and wheat at these two wavelengths plotted with respect to one another.
It is readily apparent that the three different crops, whose response as a function of wavelength are different, will lie in different portions of the two dimensional feature space (5). We can refer to these data points as possessing clustering properties since a feature vector belonging to one of the three classes (crops) tends to cluster near points with similar attributes. This is an example of crop identification. Other applications in this field are crop disease detection, land use, geology, forestry, and many other applications of environmental significance. In the following section,
two clustering algorithms will be presented which manifest themselves in the concept of pattern classification by distance measures.
2. THE CLUSTERING CONCEPT

2.1. Fundamentals

Clustering is a data analysis technique the purpose of which is to determine the natural or "inherent" relationships in a set of data points. Clustering procedures yield a description of the data in terms of "clusters" or groups of data points that possess strong internal similarities. Clustering techniques are usually employed when classification of a set of unlabeled samples (data points) is desired. This is sometimes referred to as unsupervised classification since the classification of each point into a "class" is established by the analysis procedure based on the data itself. This is in contrast with supervised classification in which the class label is known for each data point. In order to illustrate what is meant by inherent relationships in a set of data points, a simple example is considered. If the weight and height of a random sampling of professional jockeys and wrestlers is plotted (weight versus height) it is very likely that two relatively distinct clusters would result. One cluster would represent the wrestlers (taller and heavier) and the remaining cluster would represent the jockeys (shorter and lighter) (see Figure 2). Similarly, if the spectral reflectance of vegetation in a visible wave band is plotted against the reflectance in an infrared wave band, dry and green vegetation could be expected to form discernible clusters.
The clustering problem involves the determination of the
natural groupings in a set of data. This problem can be broken down
in two components: a) the determination of the measure of
similarity between the sample points, and b) the partitioning of a
set of sample points into clusters which is usually accomplished by
employing a criterion function that measures the clustering quality
of any portion of the sample data points. These two concepts are
discussed in the following paragraphs.
2.2. Similarity Measures

One of the most simple approaches to the problem of pattern classification is the concept of pattern classification by distance functions. The motivation for using distance functions as a classification tool follows from the fact that the most intuitive way of establishing a measure of similarity between pattern vectors is by determining their proximity to one another. It is expected that the distance between data points in the same cluster will be significantly less than the distance between data points in the different clusters. The concept of pattern classification by distance functions will yield practical results if the patterns tend to have clustering properties.

The most common distance measure is the familiar Euclidean point to point distance. This is defined for two n-dimensional points or vectors $\bar{x}$ and $\bar{z}$ by the following relation.

$$ D = \| \bar{x} - \bar{z} \| $$

$$ = \left( \sum_{i=1}^{n} (x_i - z_i)^2 \right)^{1/2} $$

Another familiar point to point distance measure is $L_1$ distance. This is defined for two N-dimensional points or vectors, $\bar{x}$ and $\bar{z}$, by the following relation

$$ D = \sum_{n=1}^{n} |x_i - z_i| $$

where $|x_i - z_i|$ denotes the usual absolute value function.
The algorithms presented in this report are based on the two distance measures above. There are, however, other meaningful distance measures which are also useful. For example the Mahalanobis distance is a useful measure of similarity when the statistical properties are being considered. This is given below:

\[ D = (\bar{x} - \bar{\mu})^t \sum^{-1} (\bar{x} - \bar{\mu}) \]

where \( \sum \) is the covariance matrix of a cluster, \( \bar{x} \) is a variable data point and \( \bar{\mu} \) is the mean vector of the cluster. The mean vector locates the center of the cluster and the covariance matrix tells us how well the sample mean describes the data in terms of the amount of scatter that exists in various directions (6).

2.3. Criterion Functions

The partitioning of a set of sample points into clusters is usually accomplished by using criterion functions. These functions measure the clustering qualities of a portion of the total given data points. That is, assume there are \( n \) sample data points given and it is desired to partition these points in \( M \) disjoint classes. Each class is to represent a cluster, with data points in a given cluster sharing more similar attributes than data points belonging to another cluster. It is the purpose of the criterion function to measure the quality of a particular cluster. There are numerous ways to accomplishing this. Some of the most common approaches are the sum-of-error criterion, the related minimum variance criterion
and the scattering criterion (6).

2.4. Iterative Approaches

Once the measure of similarity and the criterion function are established, clustering becomes a well defined problem in discrete optimization. The problem is to determine those partitions of the set of sample data points that optimize the criterion function.

Theoretically, this can be accomplished by classical enumeration methods, since a finite number of points is involved. However, enumeration is almost completely infeasible in practical terms except in problems dealing with a very small set of points to be classified. This rests on the fact that there are approximately $\frac{n}{c}$ ways of partitioning a set of $n$ elements into $c$ subsets. For example a search for the best set of 5 clusters in 100 samples would require considering approximately $10^{67}$ partitionings (6). It is evident that due to the exponential growth of the above relation as $n$ increases, this approach is beyond practicality.

An alternate approach to the problem of finding optimal partitions of the feature space is iterative optimization. The basic procedure is to find some reasonable initial partition of the given data samples. In the following iterations the samples are assigned to another cluster domain if this transaction increases the value of the criterion function. However, like hill climbing procedures in general, these algorithms guarantee only local optimization. Another limiting factor is that different starting
partitions can yield significantly different results. But, despite these limitations, these approaches are popular since they can be implemented efficiently.

2.5. Two Clustering Algorithms: "K-Means" and "Maximin"

Two cluster-seeking algorithms which are based on the iterative optimization concept will be presented (1). The performance of a given algorithm is dependent on a number of factors. As noted previously, the initial starting point usually yields different solutions. Another factor involved is the chosen measure of similarity which is employed by the algorithm. The type of data being analyzed also has a significant effect on the performance aspects. The algorithms given below are simple procedures based on the concept of pattern classification by distance functions. The primary assumption is that the sample data points form relatively disjoint sets. The description of the algorithms is presented using the Euclidean distance notation for simplicity.

2.5.1. K-Means Algorithm

This algorithm is a simple clustering algorithm based on the minimization of a performance criterion $P$. This performance index may be defined as the sum of the squared distances from all points in a given cluster domain to the cluster center. The following steps outline this procedure, often referred to as K Means.
The following notation is applicable:

\[ K \] = number of initial cluster centers

\[ \hat{\mathbf{z}}_j^{(k)} \] = the jth cluster center at the kth iterative step

\[ D_j^{(k)} \] = the jth cluster domain at the kth iterative step

\[ N_j^{(k)} \] = the number of points in the cluster domain, \( D_j^{(k)} \), at the kth iterative step

\[ \bar{x}_j \] = the jth data point that is to be classified

**Step 1:** K initial cluster centers are arbitrarily chosen, \( \hat{\mathbf{z}}_1^{(1)} \), \( \hat{\mathbf{z}}_2^{(1)} \), ..., \( \hat{\mathbf{z}}_K^{(1)} \). These are usually the first K points of the given data.

**Step 2:** At the kth iterative step, distribute the sample points, \( \bar{x} \), among the K cluster domains. This is accomplished by the relation

\[ \bar{x} \in D_j^{(k)} \text{ if } ||\bar{x} - \hat{\mathbf{z}}_j^{(k)}|| < ||\bar{x} - \hat{\mathbf{z}}_i^{(k)}|| \]

for all \( i = 1, 2, 3, ..., K, \neq j \). \( D_j^{(k)} \) represents the set of data points whose cluster center is \( \hat{\mathbf{z}}_j^{(k)} \). Ties in the above expression are arbitrarily resolved.

**Step 3:** From the results of step two, K cluster domains exist. For each cluster domain, \( D_j^{(k)} \), new cluster centers are computed, \( \hat{\mathbf{z}}_j^{(k+1)} \), \( j = 1, 2, ..., K \), such that the sum of the squared distances from all points in \( D_j^{(k)} \) to the new cluster center is minimized. That is, the new cluster center \( \hat{\mathbf{z}}_j^{(k+1)} \) is calculated such that the performance index \( P_j \) is minimized.

\[ P_j = \sum_{\bar{x} \in D_j^{(k)}} ||\bar{x} - \hat{\mathbf{z}}_j^{(k+1)}||^2, \quad j = 1, 2, ..., K \]

The new cluster center \( \hat{\mathbf{z}}_j^{(k+1)} \) which minimizes this performance index, \( P_j \), is the mean of the domain \( D_j^{(k)} \). Hence the new cluster
Center is given by

$$\bar{z}_{(k+1)} = \frac{1}{N_j(k)} \sum_{\bar{x} \in D_j(k)} \bar{x}^j, \quad j = 1, 2, \ldots, K$$

where $N_j(k)$ is the number of points the domain $D_j(k)$. That is, $K$ new cluster centers are computed.

Step 4: The new cluster center is compared to the old cluster center for each domain $D_j(k)$. If the new cluster center is the same as the old, the algorithm has converged, and the procedure is terminated. If this criterion is not met then go to step two.

Mathematically, that is

If $\bar{z}_{j(K+1)} = \bar{z}_{j(k)}, \quad j = 1, 2, \ldots, K$

stop, otherwise go to step two.

2.5.2. Maximin - Distance Algorithm

The maximin (maximum - minimum) distance algorithm is another simple clustering algorithm used to classify data points. This procedure utilizes the maximum - minimum concept in determining a candidate for a new cluster center. This procedure is given below.

The following notation is applicable:

- $NC$ = the number of cluster centers determined by the algorithm
- $\bar{z}_j(k)$ = the $j$th cluster center at the $k$th iterative step
- $D_j(k)$ = the $j$th cluster domain at the $k$th iterative step
- $N_j(k)$ = the number of points in the cluster domain, $D_j(k)$, at the $k$th iterative step
- $\bar{x}_j$ = the $j$th data point that is to be classified
Step 1: One initial cluster center is arbitrarily chosen. $\bar{z}_1(1)$ at the first iterative step. This is usually the first data point of the given data. The furthest point from $\bar{z}_1(1)$ is determined and becomes the second cluster center $\bar{z}_2(1)$.

Step 2: At the $k$th iterative step, for each remaining point, the distance from the point to each cluster center is calculated. The minimum distance is saved. Mathematically, that is

$$\min || \bar{x} - \bar{z}_j(k) ||, j = 1, 2, \ldots, NC$$

for each data point $\bar{x}$, where $NC$ is the number of cluster centers.

Step 3: The maximum of the minimum distances (calculated in Step 2) is selected. If this distance is greater than the average of the previous maximum distances, then the corresponding point becomes the new cluster center $\bar{z}(k+1)$, and this procedure is repeated (go to step 2). However, if this criterion is not met, the iterative procedure is terminated and no new cluster centers are established.

Step 4: The number of cluster centers is now established and the remaining data points are distributed among the cluster centers $\bar{z}_j(k), j = 1, 2, \ldots, NC$ using the following relation

$$\bar{x} \in D_j(k) \text{ if } || \bar{x} - \bar{z}_j(k) || < || \bar{x} - \bar{z}_i(k) ||$$

for all $i = 1, 2, \ldots, NC$, $i \neq j$, and for all $\bar{x}$.

Step 5: In order to obtain a more representative cluster center for each domain $D_j(k)$, the mean vector is calculated using the relation

$$\bar{z}'_j(k) = \frac{1}{N_j(k)} \sum_{\bar{x} \in D_j(k)} \bar{x}, j = 1, 2, \ldots, NC$$

for all $\bar{x}$, where $N_j(k)$ is the number of points in domain $D_j(k)$. 
A computer program written in Fortran IV, is presented in the appendix of this report. This program is designed to classify data points possessing clustering properties into classes or clusters. This is accomplished by utilizing the two clustering algorithms above, and the Euclidean and L1 distance measures defined previously. The following section deals with a description of the program.
3. A COMPUTER GENERATED EXAMPLE

3.1. Introduction

For purposes of illustration, reconsider the example presented in section one, dealing with data for a field of corn, soy beans and wheat (see Figure 1). Since their response as a function of wavelength is different, these data points representing the three different crops, lie in different portions of the two dimensional feature space. A computer generated representation for these data is presented in the Appendix, page 32. It is the purpose of the program presented in this report to partition these sample data points into separate classes representing the three crops.

3.2. Description of the Program

The program presented in the appendix may be divided into three sections:

1) Main Program
2) Plotting Subroutine
3) Clustering Algorithm Subroutines:
   a) K-means algorithm, Euclidean distance
   b) Minimax algorithm, Euclidean distance
   c) K-means algorithm, L1 distance
   d) Minimax algorithm, L1 distance
The primary function of the main program (Appendix p.29) is to generate data points similar to those shown in Figure 1 for corn, soy beans and wheat. These points, listed in the Appendix, page 31, are utilized by the different clustering algorithms for classification purposes.

The plotting subroutine (Appendix, page 30) plots the two dimensional points generated in the previous section of the program. This subroutine plots the "raw data points" before they are classified. (Appendix, page 32) In addition, it is called upon by the various clustering algorithm subroutines to plot the points after they have been classified.

The clustering algorithm K-means, based on iterative optimization, provides a simple procedure for classifying data points (see Figure 3). The behavior of the K-means algorithm is influenced by the number of cluster centers specified, the choice of initial cluster centers, and the geometrical properties of the data. In most practical cases, the application will require experimenting with various values of K as well as different choices of starting configurations. In this example the first three data points are arbitrarily chosen as three initial cluster centers. The algorithm is presented twice, once using the Euclidean distance measure and once the L1 distance measure. (Appendix, pp.33 to 38, and pp. 39 and 44 respectively).
Assign K initial cluster centers

Distribute points among cluster centers

Calculate mean vector for each cluster domain

\[ Z_{k}(K) = Z_{j}(K+1) \]
for all \( j \)

STOP

Fig. 3.—Block diagram for K means clustering algorithm
The clustering algorithm, Maximin, is another procedure based on iterative optimization. One initial cluster center is arbitrarily chosen and the algorithm locates the remaining cluster centers. The behavior of the Minimax algorithm is influenced by the choice of the initial cluster center as well as the geometrical properties of the data. In this example the first sample data point is arbitrarily chosen as the first cluster center. The algorithm is presented twice, using the Euclidean and L1 distance measures. Appendix, pages 45 to 50 and 51 to 56). For the purposes of this report, an additional step was introduced. After experimentation, using different sample points and starting points, it became apparent that the algorithm had not converged to a stable set of cluster centers. Pages 42, and 54 of the appendix depicts the results obtained by the algorithm using the Euclidean and L1 distances, respectively, before the revision. This inaccuracy is due to the fact that after the cluster centers are established using the maximin criteria, the data points are assigned to the cluster domains on the minimum distance concept. That is, a point is assigned to a cluster domain if it is closest to the corresponding cluster center. In the last step, the mean vector (cluster center) for each cluster domain is calculated and the algorithm is terminated. Since the points are assigned to each cluster domain using the cluster centers established by the program, these cluster centers may not be a true representation of the data. The additional step is simply to reassign the data points to the cluster domains using the updated cluster centers
calculated in the previous step until the cluster centers converge. In other words, follow steps 2 thru 4 of the K-means clustering algorithm, explicitly (see Figure 4).

3.3. Comparison of Results

As noted previously, it is the purpose of the program described above to partition the computer generated data points into separate classes representing three crops; corn, soy beans, and wheat. The final results of this example are given in the appendix of the report. (Appendix pp. 38,44,50,56) Examination of the above reveals that each of the four combinations of algorithm and distance measure yield identical results. In other words, in each of the four combinations it is apparent that each point is assigned to the same respective cluster thereby resulting in four sets of identical cluster centers. These are given below.

TABLE 1

LIST OF CLUSTER CENTERS FOR DATA REPRESENTING CORN, SOY BEANS, AND WHEAT

<table>
<thead>
<tr>
<th>Cluster Number</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Number 1</td>
<td>58.11</td>
<td>63.89</td>
</tr>
<tr>
<td>Cluster Number 2</td>
<td>93.05</td>
<td>17.69</td>
</tr>
<tr>
<td>Cluster Number 3</td>
<td>23.27</td>
<td>24.10</td>
</tr>
</tbody>
</table>
Assign one initial cluster center; determine the farthest point from it, this becomes cluster center number two.

For each point, compute distance from the point to each cluster center; save the minimum distance.

Pick the maximum of the minimum distances.

Average of previous < Max. Max. distances distance

Distribute points among cluster centers.

Calculate mean vector for each cluster domain.

If $z_j(K) < z_j(K+1)$ for all $j$ then NO.

STOP
Since there is no reasonable approach to compare the true execution time for a given algorithm and distance measure combination, no attempt will be made to do so. The time for convergence of a given algorithm depends strongly on the initial starting point. However, it should be noted that the L1 distance measure is more efficient, since it requires less computation than the Euclidean distance measure. Additionally, the K means algorithm is more efficient since all of the initial cluster centers are chosen at the beginning of the algorithm. This is in contrast with the maximin algorithm in which one initial cluster center is initially chosen and the remaining centers are established by the program based on predefined criteria.

In an effort to illustrate the above clustering algorithms, a two dimensional example was selected for simplicity. The results obtained are precisely what we would expect. The two dimensional classification problem is an intuitive process since visual analysis may be successfully performed. However, beyond two or possibly three dimensions, visual analysis is impossible. For such cases it is necessary to have a computer perform the cluster analysis and report the results in a useful fashion.

3.4. Extensions

The material in this report exemplifies the principal concepts underlying pattern recognition systems. The clustering concept is discussed and two algorithms based upon the distance function concept
and iterative optimization are introduced. A computer program is presented illustrating how significant clusters may be identified in a given set of data. The remainder of this report deals with extensions to the algorithms presented thus far.

The cluster seeking problem has been treated in a variety of ways. The literature abounds with methods and procedures designed to identify clusters in a given set of data. Some of the more sophisticated concepts currently being used are now discussed. These approaches generally encompass some of the concepts utilized by the two algorithms presented in this report, along with more elegant approaches designed to further analyze the data under consideration.

Two basic concepts currently employed are cluster splitting and cluster merging. Specific parameters or thresholds are established and splitting and merging criteria are based on these parameters. Examples of a few of the most common are listed below (7):

\[ P_1 = \text{minimum number of points allowed in a cluster} \]
\[ P_2 = \text{parameter for cluster splitting, units of standard deviation} \]
\[ P_3 = \text{parameter for cluster splitting, maximum number of clusters allowed} \]
\[ P_4 = \text{parameter for cluster merging, minimum distance between cluster centers} \]
\[ P_5 = \text{parameter for cluster merging, maximum number of pairs of clusters which can be merged} \]

The basic procedure is relatively straightforward. Cluster means (centers) are established and data points are assigned to a
particular cluster domain on the minimum distance concept similar to the algorithms originally presented. At the end of each pass thru the data, those clusters whose number of points is less than the corresponding parameter, $P_1$, are eliminated. The cluster splitting or cluster merging phase is now entered.

If certain criteria are met, a cluster is allowed to split. Usually this is accomplished by determining the standard deviation vector $\sigma_j = (\sigma_{1j}, \sigma_{2j}, \ldots, \sigma_{nj})$ for a specific cluster domain, $D_j$. The maximum component of $\sigma_j$ is determined. If the standard deviation $\sigma_j$ exceeds the standard deviation parameter, $P_2$, then the cluster is split along the component corresponding to the maximum component in $\sigma_j$. Other clusters may be allowed to split depending on the value of the parameter $P_3$. Typically, the iteration is terminated followed by another pass thru the data points. Two or more clusters may be allowed to merge, if various criteria are met. Normally, this is accomplished by determining the distance between the means (cluster centers) of the particular cluster domains under consideration. If this distance is less than the corresponding parameter, $P_4$, the clusters are allowed to merge into one cluster. The new cluster mean is simply the average of the two respective means. Other clusters may be allowed to merge with this cluster depending on parameter $P_5$.

The primary difficulty underlying cluster analysis is the fact we may be exploring an unknown set of data. Thus, a constantly recurring problem is that of deciding on reasonable parameters to
be utilized by the various clustering algorithms. Although various statistical approximate analysis are currently being employed, the problem of testing cluster validity is still essentially unsolved.
THE MAIN PROGRAM GENERATES THE DATA POINTS TO BE CLASSIFIED.

INTEGER X(99), Y(99), Z(99), POINT
DATA POINT /1, 1/
DO 1 I=1, 99
2(I)=POINT
1 CONTINUE
IX=6444055
AM=.25
S=50./6.
DO 100 I=20, 60
CALL RANDU(IX, IY, YFL)
IX=IY
X(I)=YFL*40
CALL GAUSS(IX, S, AM, V)
100 Y(I)=V
AM=.90
DO 200 I=61, 99
CALL RANDU(IX, IY, YFL)
IX=IY
X(I)=YFL*19.49
CALL RANDU(IX, IY, YFL)
Y(I)=YFL*19.49
200 X(I)=V
DO 300 I=1, 19
CALL RANDU(IX, IY, YFL)
IX=IY
X(I)=YFL*19.49
CALL RANDU(IX, IY, YFL)
Y(I)=YFL*19.49
300 Y(I)=YFL*15.58
WRITE(6,403)
DO 400 I=1, 49
400 J=I+49
WRITE(6,401) X(I), Y(I), X(J), Y(J)
401 FORMAT(T25, T24, T20, T22)
WRITE(6,401) X(99), Y(99)
CALL PATTY(X,Y)
CALL KMEANS(X, Y)
STOP
END

MAIN PROGRAM
SUBROUTINE PATTY PLots THE DATA POINTS.

INTEGER LINE(115), ALANK, POINT, X(99), Y(99), PRT(23), Z(99)

DATA BLANK, POINT / 1, 1, 1 /
DATA ONE, TWO, THREE / '3', '2', '1' /
WRITE(6,10)
10 FORMAT(1l, 115('---), (1l)
   100, 11=1, 70
   I=71-1
   NO, 20 IA=1, 115
20 LINE(IA)=ALANK
   NO, 30 JJ=1, 115
   J=JJ-1
   NO 30 TJ=1, 99
   IF(Y(TJ).EQ.1.AND.X(IJ).EQ.J) LINE(J+1)=Z(IJ)
   CONTINUE
   WRITE(6,40) I, LINE
40 FORMAT(1l, 115A1, '1')
100 CONTINUE
   J=0
   NO, 60 I=1, 115+5
   J=J+1
   PRT(J)=I
   WRITE(6,50) PRT
50 FORMAT(1l, 115('---), I',/,' ', 23I5,/) RETURN
END

PLOTTING SUBROUTINE
### DATA POINTS

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>60</td>
</tr>
<tr>
<td>53</td>
<td>66</td>
</tr>
<tr>
<td>64</td>
<td>68</td>
</tr>
<tr>
<td>60</td>
<td>65</td>
</tr>
<tr>
<td>61</td>
<td>66</td>
</tr>
<tr>
<td>60</td>
<td>65</td>
</tr>
<tr>
<td>60</td>
<td>65</td>
</tr>
<tr>
<td>58</td>
<td>67</td>
</tr>
<tr>
<td>57</td>
<td>67</td>
</tr>
<tr>
<td>56</td>
<td>66</td>
</tr>
<tr>
<td>59</td>
<td>66</td>
</tr>
<tr>
<td>58</td>
<td>67</td>
</tr>
<tr>
<td>50</td>
<td>62</td>
</tr>
<tr>
<td>59</td>
<td>66</td>
</tr>
<tr>
<td>61</td>
<td>66</td>
</tr>
<tr>
<td>66</td>
<td>61</td>
</tr>
<tr>
<td>60</td>
<td>58</td>
</tr>
<tr>
<td>55</td>
<td>56</td>
</tr>
<tr>
<td>49</td>
<td>50</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>24</td>
<td>23</td>
</tr>
<tr>
<td>17</td>
<td>20</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>23</td>
<td>34</td>
</tr>
<tr>
<td>27</td>
<td>33</td>
</tr>
<tr>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>37</td>
<td>24</td>
</tr>
<tr>
<td>32</td>
<td>22</td>
</tr>
<tr>
<td>25</td>
<td>21</td>
</tr>
<tr>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>37</td>
<td>36</td>
</tr>
<tr>
<td>32</td>
<td>35</td>
</tr>
<tr>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>39</td>
<td>28</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>168</td>
<td>36</td>
</tr>
</tbody>
</table>

**LIST OF DATA POINTS TO BE CLASSIFIED**
K MEANS ALGORITHM, EUCLIDEAN DISTANCE
PROGRAM FOR K MEANS ALGORITHM, EUCLIDEAN DISTANCE
PROGRAM FOR K MEANS ALGORITHM, EUCLIDEAN DISTANCE
ITERATIVE PLOTS FOR K MEANS ALGORITHM, EUCLIDEAN DISTANCE
ITERATIVE PLOTS FOR K MEANS ALGORITHM, EUCLIDEAN DISTANCE
ITERATIVE PLOTS FOR K MEANS ALGORITHM, EUCLIDEAN DISTANCE
K MEANS ALGORITHM, L1 DISTANCE
PROGRAM FOR K MEANS ALGORITHM, L1 DISTANCE
PROGRAM FOR K MEANS ALGORITHM, L1 DISTANCE
ITERATIVE PLOTS FOR K MEANS ALGORITHM, L1 DISTANCE
ITERATIVE PLOTS FOR K MEANS ALGORITHM, L1 DISTANCE
ITERATIVE PLOTS FOR K MEANS ALGORITHM, L1 DISTANCE
MAXIMIN ALGORITHM, EUCLIDEAN DISTANCE
PROGRAM FOR MAXIMIN ALGORITHM, EUCLIDEAN DISTANCE
PROGRAM FOR MAXIMIN ALGORITHM, EUCLIDEAN DISTANCE
RESULTS OF ORIGINAL MAXIMIN ALGORITHM, EUCLIDEAN DISTANCE
ITERATIVE PLOTS OF REVISED MAXIMIN ALGORITHM, EUCLIDEAN DISTANCE
MAXIMIN ALGORITHM, L1 DISTANCE
PROGRAM FOR MAXIMIN ALGORITHM, L1 DISTANCE
PROGRAM FOR MAXIMIN ALGORITHM, L1 DISTANCE
RESULTS OF ORIGINAL MAXIMIN ALGORITHM, L1 DISTANCE
ITERATIVE PLOTS OF REVISED MAXIMIN ALGORITHMS, L1 DISTANCE
REFERENCES CITED


