DEVELOPMENT OF AN ADAPTIVE FUZZY SHELL CLUSTERING ALGORITHM AND VALIDITY MEASURES

by

Kurra Bhaswan.

Thesis submitted to the Faculty of the Graduate School of
the New Jersey Institute of Technology in partial fulfillment of
the requirements for the degree of
Master of Science in Mechanical Engineering

1991

APPROVAL SHEET

Title of Thesis: Development of an Adaptive Fuzzy Shell Clustering Algonthm and Validity Measures.

Name of the Candidate: Kurra Bhaswan

Thesis and Abstract Approved:

Dr. R. N. Dave　　Date
Mechanical Engineering Department
New Jersey Institute of Technology

Signature of other members of the thesis committee:

Dr. A. D. Rosato　　Date

Dr. I. S. Fischer　　Date

ABSTRACT

Title of Thesis: Development of an Adaptive Fuzzy Clustering Algorithm and Validity Measures

Kurra Bhaswan, Master of Science in Mechanical Engineering, 1991.

Thesis directed by: Dr. R. N. Dave, Assistant Professor in Mechanical Engineering department, New Jersey Institute of Technology.

In objective functional based fuzzy clustering algorithms the weighted sum of the distances of the feature vectors from cluster prototype are minimized. The fuzzy memberships are utilized as weighing factors. The cluster prototype can be a point or a line or a plane, etc. This work extends the recent concept of using curved prototypes by utilizing the Adaptive Norm Theorem proposed by Dave[1] to develop an algorithm for the detection of fuzzy hyper-ellipsoidal shell prototypes. The Objective functional associates a norm for each cluster in which to measure the proximity of the shell prototype adaptively. The resulting implementation necessitates solving a set of non-linear equations through the application of the Newton's method which requires good starting values as a pre-requisite for convergence. A robust initialization scheme is presented to obtain good starting values for the partition and the prototype. The kind of substructures encountered are isolated and categorized into two groups and appropriate strategies suggested. Two schemes are suggested for the initial partition and the spatial properties of the domain are used to generate starting values for the prototypes. An iterative algorithm is outlined to obtain the starting

[1] R. N. Dave and K. J. Patel, FCES Clustering Algorithm and detection of ellipsoidal shapes, *Proc. of the SPIE Conf. on Intelligent Robots and Computer Vision IX*, Boston, pp. 320-333, Nov.

guesses for the Newton's method. The algorithm is coerced to find a good initial guess by the use of different prototypes at different phases during its operation. Examples typifying substructures commonly encountered are shown to demonstrate the combined results of the initialization and the subsequent application of the AFCS algorithm. The problem of validating the number of subsets present in the data and the evaluation of the resulting substructure is also addressed. The existing validity measures for fuzzy clustering are surveyed and are shown to be partition based. Three new measures specifically designed to validate the shell substructure are introduced. Several examples are included to demonstrate the superiority of the new measures over the existing measures.

VITA

Name: Kurra Bhaswan.

Permanent Address: 261, Kearny Ave, Kearny, NJ, 07032.

Degree and date to

be conferred: M. S. M. E., May 1991.

Date of birth:

Place of Birth:

Secondary Education: Kendriya Vidyalaya, Madras, India.

Collegiate institutions attended	**Date**	**Degree**	**Date of Degree**
New Jersey Institute of Technology	Aug. 1989- May 1991	MSME	May. 1991
J. N. T. U. College of Engineering	Aug. 1984- May 1988	BSME	June 1988

Major: Mechanical Engineering

Publications: **"Adaptive Fuzzy c-shells Clustering and Detection of Ellipses"** by R. N. Dave and K. Bhaswan (submitted to) in IEEE Trans. Neural Networks 1991.

"New measures for evaluating fuzzy partitions induced through c-shells clustering" by R.N. Dave and K. Bhaswan in Intelligent Robots and Computer Vision IX: Algorithm and Techniques, an SPIE conference, 12-17 November 1991, Boston, MA.

“Adaptive c-shells clustering”, by R.N. Dave and K. Bhaswan in Proceedings of the North American Fuzzy Information Processing Society Workshop, Columbia, Missouri, pp. 195-199, 1991.

ACKNOWLEDGEMENTS

I take this opportunity to express my deep gratitude to Dr. Rajesh Dave, Assistant Professor, Mechanical Engineering Department of N. J. I. T. for his valuable guidance and help throughout the course of this work. It was an enlightening experience working under his guidance.

I am also very thankful to Dr. A. D. Rosato and Dr. I. S. Fischer for devoting their precious time to reviewing my work and providing me with constructive suggestions.

CONTENTS

3 Adaptive Norm FCS Algorithm

4 Cluster Validity

CHAPTER 1

Introduction

1.1 Classification

Decision making problems occur in numerous scientific and engineering applications ranging from basic measurements of the environment to machine-intelligent operations such as vision and speech recognition. Systems that perform some sort of decision making surround us daily. Examples of these systems are the burglar alarms that protect homes, offices, and automobiles; remote control devices that control home entertainment systems, and readers of UPC bar codes that expedite pricing and inventory of retail merchandise. In other words, decision making involves a classification of the real world data into one of a number of possible groups, the data variable can be assigned based on some objective criterion. It seems apparent that for the most part, unless the classification is obvious and trivial we still depend on human expertise to classify on the basis of observations.

Even before computer use became common, statisticians and others developed fairly simple methods of objective classification based on standard probability theory. However, as the classification problem has proved so important in so many different fields of application it has suffered from being re-solved very many times. Each time a discipline has re-invented the subject of classification it has introduced its own jargon, its own notation and its own favorite methods. For example classification is known as pattern recognition, discriminate analysis, decision theory, assignment analysis etc. Perhaps the most recent and most important re-use of classification analysis is in the area of 'expert systems', programs which seek directly to replace expert reasoning using AI (artificial intelligence) techniques. The particular type of classification referred to will be that of Pattern Recognition.

1.2 Pattern Recognition

The goal of Pattern Recognition is to classify objects of interest into one of a number of categories or classes. The objects of interest are generically called patterns and may be printed letters or characters, biological cells, electronic waveforms or signals, "states" of a system or any of a number of other things that one may desire to classify. Diday and Simon[15] give a definition of classification in the context of Pattern Recognition as follows:

Let X be an object defined by n parameters or variables $(x_1, x_2, \ldots, x_n)$. Let X be the "space" of these variables. Let $E = \{X_1, \ldots, X_m\}$ be the set of m of these objects. In a natural language, $\boldsymbol{X}$ is a variable object; $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}$ are the results of measures on this object. Usually a measure has the properties of a "*quantitative* value"; length, weight, amplitude etc. In an abstract machine language, the interpretation of X is a set of variable input data: $x_1, \ldots, x_n$ are the atomic variable input data. X is the name of a data; $x_1, \ldots x_n$ are the states of the data of global name X.

The mapping that defines the equivalence of the pattern classes C_i on the space X is given by

$$C_i = \{ X \mid \varepsilon : X \rightarrow \omega_i \}.$$

All the objects of the same C_i having the same name ω_i are usually interpreted as different occurrences of the same object. The practical problem of classification is to find a constructive identification function (or program or operator) which will perform the above mapping. A detailed discussion of the form such a function can take is carried out in Chapter2.

1.3 Fuzzy Clustering

The main difficulties in a pattern recognition problem are:

- to define the semantics of the pattern recognition, in other words to define what properties the pattern recognition function must have.
- to find a constructive function which will satisfy the above semantics.

Even while this is so in many decision making problems there is little prior information about the data and the decision maker wishes to make as few assumptions about the data as possible. This restricts one to studying the interrelationships among the data points to make a preliminary assessment of their structure. Cluster analysis is one tool of exploratory data analysis that attempts to assess the interaction among patterns by organizing the patterns into groups or clusters such that patterns within a cluster are more similar to each other than are patterns belonging to different clusters. This work will be concerned with fuzzy clustering which involves the application of fuzzy set theory to clustering.

Zadeh[34] proposed the fuzzy set theory in which he defines a fuzzy set as a class of objects with a continuum of grades of membership. Such a set is characterized by a membership (characteristic) function which assigns to each object a grade of membership ranging between zero and one. The theory of fuzzy sets is applied to clustering. Generally speaking, the clustering problem is formulated in the following way. Consider a finite set of elements $X = \{x_1, x_2, \ldots, x_n\}$ being elements of a p-dimensional Euclidean space R^n. Perform a partition of this collection of elements (points) into "c" fuzzy sets, c fixed, with respect to a given criterion, i.e. assign membership grades to each element of X equal to u_{ik}, $i = 1, 2, \ldots, c$, $k = 1, 2, \ldots, n$. Usually the methods use a notion of distance or dissimilarity measure to classify the objects, which lead to metric or non-metric methods of clustering. The results of a clustering are represented in a convenient way in the form of a partition matrix U indicating the detected structure of the studied data set. It consists of c rows and n columns, $U = [u_{ik}]$, $i = 1, 2, \ldots, c$, $k = 1, 2, \ldots, n$. The rows correspond to the clusters obtained. The (i,k)th element of U indicates a belongingness of the kth object to the ith cluster. Two additional constraints are also introduced; their meaning being self-evident. Firstly, a total membership of the element (object) $x_k \in X$ to all classes is equal to 1.0,

$$\forall_{1 \le k \le n} \sum_{i=1}^{c} u_{ik} = 1.$$

Secondly, every cluster constructed is non-empty, and different from the entire set. This requirement is set down as follows,

$$\underset{1 \le i \le c}{\forall} \quad n > \sum_{k=1}^{n} u_{ik} > 0$$

One of the widely used clustering methods, the fuzzy c-means(FCM) has been studied and developed in detail by Bezdek[4]. The Objective functional takes the form

$$J = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^{m} \|x_k - v_i\|^2, m > 1$$

Minimization is obtained by means of an iterative procedure.

1.4 Statement of the Problem

Dave[7] introduced the idea of hyperspherical fuzzy shell clustering that measures distances from a "shell" prototype. In an attempt to extend the concept of shell clustering to ellipsoidal shells and higher order derivatives, Patel[27] employed the theory of residuals to solve the resulting non-linear system of equations. But his representation is supposititious in that the orientation is calculated in isolation from the prevalent formulation and thus suffers from becoming entrapped in local optima. The multipartite environment resulting from the utilization of fuzzy memberships compounded by the choice of a poor numerical model renders the algorithm's behaviour as unpredictable with even slight changes in the orientation. Thus there is clearly a need for a more well-founded approach to this problem.

Dave[11] proposed an adaptive norm theorem for the detection of hyperellipsoidal shells that is based on the extraction of the optimum resulting from an adaptive realization of the norm associated with each substructure being detected . The objective of this thesis is to develop an algorithm that uses the idea of an adaptive norm as proposed by Dave[11] and applying it to the detection of hyper-ellipsoidal shells. The numerical method chosen is the Newton's method. The need for a good initialization is exigent with the Newton's method. As a result, the investigation and development of an initialization scheme forms a significant proportion of the goal being sought. In order to evaluate the performance of the resulting algorithm and the ascertainment of the validity of the structure being detected, there clearly exists the need for a figure of merit. In other words, some quantitative measures of significance are needed for evaluating the cluster substructure. Thus a concommitant objective was to investigate the problem of cluster validity and to develop a set of indices to evaluate the substructure detected.

1.5 Outline of the remaining Chapters

Chapter 2 gives an insight into the semantics behind the clustering phenomenon by presenting the concept. A brief survey of the various clustering techniques available in the literature is given. In doing so, the approach has been one of gaining a perspective on the diversity of algorithms present in the field of clustering as also the kind of data they are specifically suited to operating upon. A discussion of three particularly interesting and innovative fuzzy clustering algorithms is also presented.

Chapter 3 discusses the importance of shell clustering as opposed to the traditional approach of searching for solid prototypes taken by most partitional clustering algorithms. It discusses the limitations of the FCS algorithm and presents the AFCS Norm algorithm as proposed by Dave in which the algorithm induces in each of its clusters a norm for it. A good deal of this chapter is devoted to discussing the implementation and the constraints of initialization imposed by the nature of complexity of the solution space of the theorem proposed. A robust initialization scheme is presented along with results of the AFCS algorithm.

Chapter 4 discusses the need for performance measures as a result of applying a clustering technique. It gives a critique of the various schemes suggested by researchers in the past on validating the results of clustering and discusses the attendant problem of determining the number of subsets present in a given data. A number of numerical examples are given along with the graphical and tabular representations of the various measures.

Finally Chapter 5 concludes the work with a summary of the various approaches presented within and briefly points to directions for future research.

The appendices supplement most of the theory given in Chapters 3 and 4. Appendix A discusses some basic operations upon random vectors and the technique used in this work to solve systems of equations viz., factorization by LU decomposition. Appendix B presents some formal theorems on metrics, the norm and normed linear spaces. More specifically, the approach is one of attempting to portray through depiction in the two-dimensional Euclidean co-ordinate space, some of the critical properties of normed linear spaces in higher dimensions. Appendix C gives a pseudo code for the initialization scheme presented in chapter 3 to help gain a good starting guess for the Newton's method given the skewness of the solution space. Appendix D gives the proof for the AFCS Norm theorem given in chapter 3 employing constrained optimization through the Lagrange multiplier method. Finally Appendix E concludes the appendices section by giving the term by term expansion of the Jacobian.

CHAPTER 2

Background

2.1 Introduction

The process of recognition and classification in its simplest sense is an activity that consists of sorting like items into groups. These groups are described by patterns and what is performed is the act of recognition of certain patterns and then classification of them into groups. The word "pattern" follows the root of the word "patron" and reflects the concept of an ideal model of a set of objects or structures. Thus when an object gets classified as belonging to what we call a class, we render it classified. If there exists some set of patterns, the individual classes of which are already known, the phenomenon is known as supervised pattern recognition.

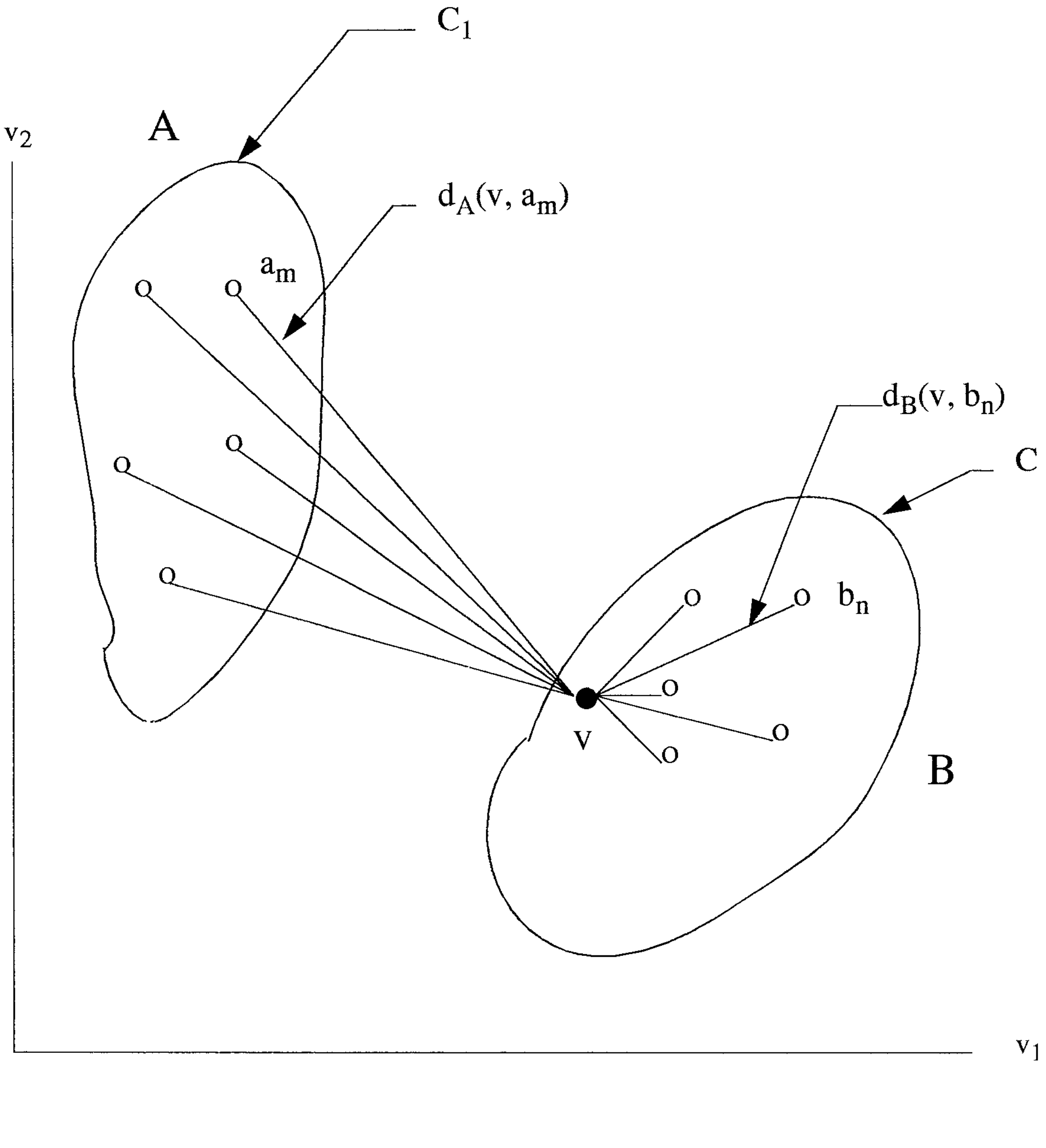
2.2 Unsupervised Pattern Classification

If all of the available patterns are unknown, and perhaps even the number of classes is unknown, then one has a problem in unsupervised pattern recognition or clustering. In such classification and decision-making problems, there is little prior information available about the data and the decision-maker wishes to make as few assumptions about the data as possible. In such problems, one attempts to find classes of patterns with similar properties where sometimes even these properties may be undefined. This restricts one to studying the interrelationships among the data points to make a preliminary assessment of their structure. Such situations come under the domain of exploratory data analysis that attempts to assess the interaction among patterns by organizing the patterns into groups or clusters such that patterns within a cluster are more similar to each other than are patterns belonging to different clusters. An object is described either by a set of measurements or by a set of relationships between the object and other objects. Cluster analysis does not use category labels that tag objects with prior identifiers. The objective of cluster analysis is simply to find a convenient and valid organization of the data, not to establish rules for separating future data into categories. Clustering algorithms are geared towards finding structure in data.

2.3 Meaning of Clustering

In order to classify an object into one of several sets of patterns, we will need quantized information about the object so that it can be used by some mathematical method. Since mathematical methods cannot deal with electrical signals, physical objects and optical images or any other such raw information directly, first a mathematical model of the physical world must be constructed. Thus these signals, images or events to be recognized are considered to be represented by points or vectors in an N-dimensional space. Each dimension expresses a property of the event, a type of statement that can be made about it. The entire signal that represents all the information available about the event is a vector $v = (v_1, v_2, ..., v_N)$ the coordinates of which have numerical values that correspond to the amount of each property the event has. For instance the photographic image of an object may be represented as a vector by scanning the photograph with a television raster and band limiting the resulting video signal. This, in effect divides the photograph into N rectangular cells. The set of cell intensities, the N equally spaced sample heights of the television video, forms an N-dimensional vector representation of the photograph. In an N-dimensional space, therefore, the entire picture can be thought of as a single point.

More generally, however, an object or event can be represented by the numerical values of a set of descriptors, by the numerical outcomes of N quantitative tests performed on the input. Each type of test is a descriptor of the physical world, and the set of descriptors may be likened to a vocabulary of finite size to which our communication between the physical world and the machine is restricted. In this representation the set of events belonging to the same class corresponds to an ensemble of points scattered within some region of the object space. One might expect that the set of points representing different events that belong to the same class would cluster in the N-dimensional space in the sense that "distances" between members of the same class would be small, on the average. One might also expect that members of another class would also cluster, but that the two clusters representing the two classes would remain separated from one another. A simple illustration of this idea in a two-dimensional space is shown in the figure 2.1, where the ensemble of points, A represents different samples of class A, and those labelled B represent samples of class B. As is evident from the illustration, distances between points within A are smaller than those between two points, of which one is in A and the other in B. Unfortunately this state of affairs cannot be expected to exist. Therefore the problem becomes one of developing functions from sets of finite samples of the classes so that the functions will partition the space into regions each containing the sample points belonging to one class. The practical result to be achieved would seem to lie in the automatic con-



MEMBERS OF 'A' LIE WITHIN C_1 ,
THOSE OF 'B' LIE WITHIN C_2

Fig 2.1 **Separation of Classes**

struction of many dimensional templates that optimally define the region of the N-space in which members of different classes are contained. In the two dimensional illustration of Fig 2.1, the areas enclosed by contours C1 and C2 are such templates. These templates specify how each point in the vector space should be classified. Thus the templates specify the decision rule with which membership in A or B is determined. These are areas within which members of A and B, respectively are densely distributed.

2.4 Fuzzy Clustering

The concept that plays a central role in clustering is the notion that, in the object space, the ensemble of points which represents a set of non-identical events of a common category, must be close to each other, as measured by some - as yet unknown - method of measuring distance. Transformations of the vector space or measures of distance must be developed that increase the clustering of points within a class and increase the separation between classes. This proximity requirement is significant because the points represent events that are close to each other in the sense that they are members of the same categories. Mathematically speaking, the fundamental notion is that similarity (closeness in the sense of belonging to the same class or category) is expressible by a metric (a method of measuring distance) by which the points representing samples of the category that are to be recognized are found to lie close to each other. In order to give credence to this idea let us consider what we mean by the abstract concept of a class. According to one of the possible definitions, a class is a collection of things that have common properties. By a modification of this thought, a class could be characterized by the common properties of its members. A metric by which points representing samples of a class are close to each other must operate chiefly on the common properties of the samples and must ignore to a large extent, those properties not present in each sample. As a consequence of this argument if a metric were found that called samples of the class close, somehow it would have to exhibit their common properties.

In order to present this fundamental idea in a slightly different way, we can restate that a transformation on the object space which is capable of clustering the points representing samples of a class must operate primarily on the common properties of the samples. A simple illustration of this idea is shown in Fig2.2, where the ensemble of points is spread out in the object space (only a two-dimensional space is shown for ease in illustration), but where a transformation T of the space is capable of clustering the points of ensemble. In the example above, neither the object's property represented by coordinate v_2 is sufficient to describe the class, for the spread in each is large over the ensemble of points. Some function of the two coordinates, on the other hand, would exhibit the common property that the ratio of the value of coordinate v_2 to that of coordinate v_1 of each point in the ensemble is nearly one. In this specific instance, of course, simple correlation between the two coordinates would exhibit this property; but in more general situations simple correlation will not suffice.

If the object space shown in Fig 2.2 were flexible (like a rubber sheet), the transformation T would express the manner in which various portions of the space must be stretched or compressed in order to bring the points together most closely. A mathematical technique is presented in chapter 3 that will find automatically the best metric or norm or transformation to achieve clustering according to a specified criterion which is in some sense the best. We present below some definitions on clustering proposed by researchers in the area of cluster analysis.

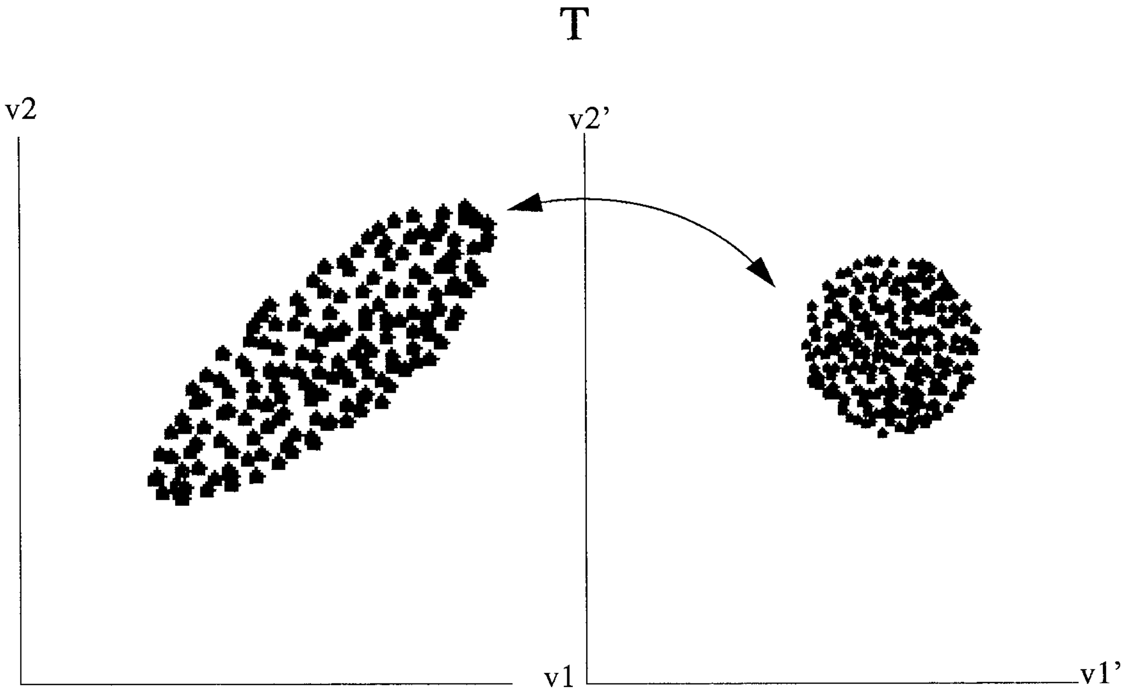


Fig 2.2 **Clustering by transformation**

Diday and Simon[15] give a workable definition of clustering.

Clustering is a classification technique for which

1) the semantic of the classification problem is given by similarities between the objects X.

2) Usually all the objects X to be classified are known by their measurements.

3) Usually no training set is given a priori.

4) The set of Classes has to be determined by the process.

Clustering algorithms organize data, and data are collected in several formats, sizes, and shapes. There are two popular forms of data representations: the pattern matrix and the proximity matrix. Proximity data occurs as an *n* x *n* proximity matrix whose columns and rows both represent patterns and whose entries measure proximity (similarity or dissimilarity) between all pairs of patterns. A pattern matrix is an *n* x *n* matrix, where each row is a pattern and each column denotes a feature. The *p* features are viewed as a set of orthogonal axes, and each pattern is represented as a point in *p*-dimensional space, called the pattern space. A proximity matrix can be derived from a pattern matrix using Euclidean distance techniques, but ordination techniques are needed to create a pattern matrix from a proximity matrix.

The choice of variables or measurements or features to describe the patterns is very important. While this is dictated by the application area and the perception and prior experience of the investigator, it is important to keep the number of features small for ease in computation and interpretation of the results. Another important consideration in cluster analysis is whether or not the data should be normalized. Often the features are measured on different units, and the patterns need to be normalized so that no single feature overwhelms the data merely because of scale. The most commonly used normalization replaces each feature with a new one whose mean value is zero and standard deviation is one. This

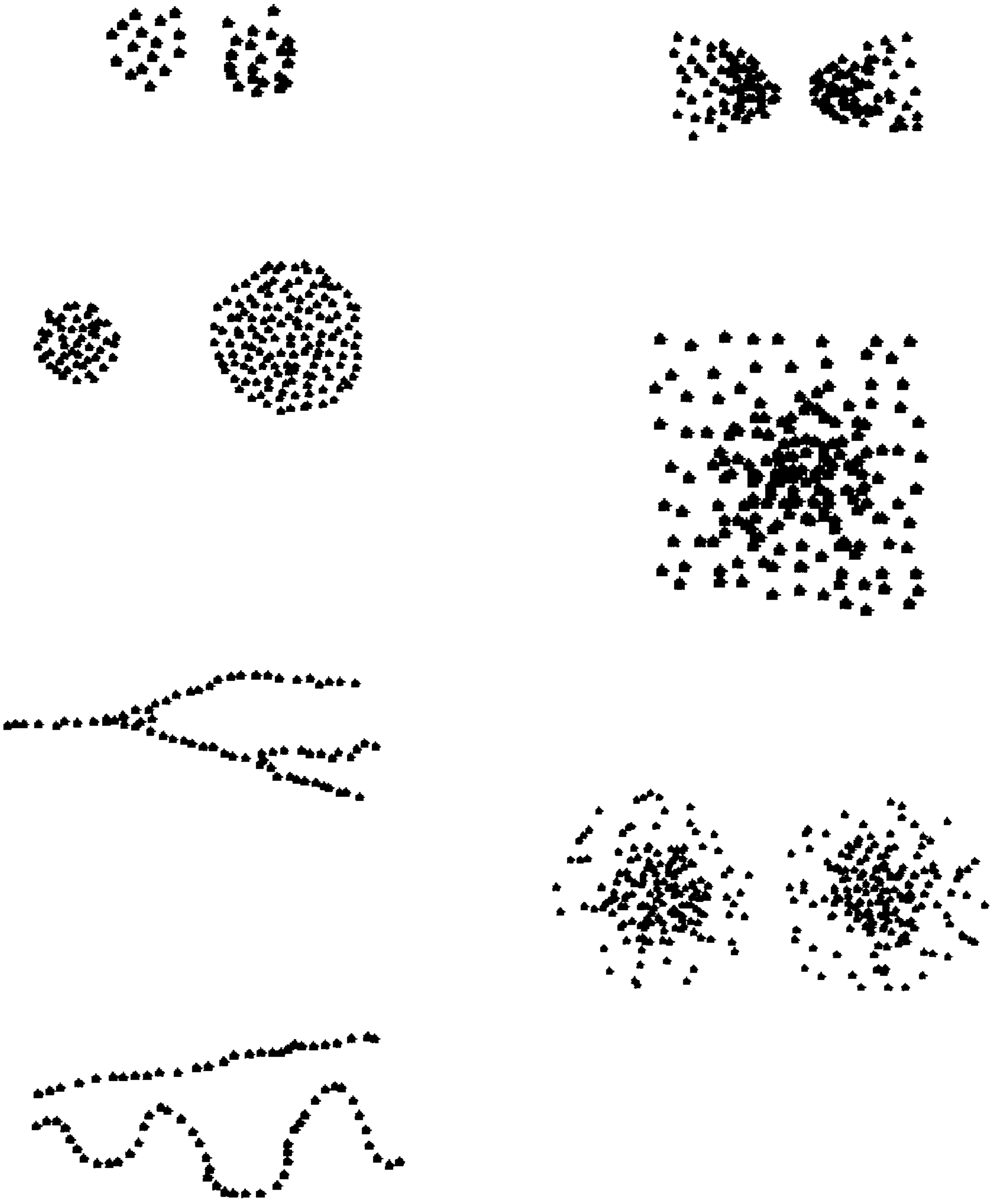


Fig 2.3 **Sample Cluster Problems**

normalization, however, should be used with caution since it can distort the clustering structure present in the data.

Clustering algorithms group patterns based on some measure of similarity or dissimilarity between the patterns. Everitt[20] correctly points out that the output of a clustering algorithm will be only as meaningful as the input distances among the patterns. Euclidean distance is the most popular distance measure. In many situations, Mahalanobis distance is preferred because it takes into account the correlation among features and is unaffected by a change of scale of the feature. However, it is important to note that different distance measures can lead to different partitions of the same data.

A number of definitions for a cluster have been proposed, but no single definition of cluster is adequate, as seen in Fig 2.3. A cluster is comprised of a number of *similar* objects collected or grouped together. Everitt [20] documents some of the following definitions of a cluster:

1. "A cluster is a set of entities which are *alike*, and entities from different clusters are not alike."
2. "A cluster is an aggregation of points in the test space such that the *distance* between any two points in the cluster is less than the distance between any point in the cluster and any point not in it."
3. "Clusters may be described as connected regions of a p-dimensional space containing a relatively *high density* of points, separated from other such regions by a region containing a relatively low density of points."

These traditional definitions are not well accepted because they do not take into consideration any "Gestalt" concepts which people seem to use in grouping objects.

It is clear that the user's prior conception determines what a cluster means and sets the goal for a clustering method. One can formulate an idea of a cluster from an assumed mathematical model for data generation or from prior work in the subject matter. For example one can picture each cluster as a single point to which measurement noise has been added in the direction of each feature. A reasonable idea of a cluster would then be a spherical or hyperellipsoidal swarm of patterns. Several partitional clustering methods proposed in the literature are based on idealized clustering structures of this type and essentially fit a mixture of Gaussian distributions to the given data. Some clustering algorithms always place the two patterns which are the closest in the same cluster.

The last two definitions assume that the objects to be clustered are represented as points in the measurement space. Usually "similarity" is defined as the proximity of the points according to some distance function, but measures of similarity could be based on other properties such as the direction of the vectors in the measurement space. The method for finding the clusters may have a heuristic basis or may be more rigorously dependent on minimization of a mathematical clustering criterion. In either case, iterative procedures are generally used to find the clusters. We recognize a cluster when we see it in the plane, although it is not clear how we do it. While it is easy to give a functional definition of a cluster, it is very difficult to give an operational definition of a cluster. This is due to the fact that objects can be grouped into clusters with different purposes in mind.

2.5 Clustering Techniques

Clustering techniques offer several advantages over a manual grouping process. First a clustering program can apply a specified objective criterion consistently to form the groups.

Second, a clustering algorithm can form the groups in a fraction of the time required by a manual grouping, particularly if a long list of descriptors or features is associated with each object. The speed, reliability, and consistency of a clustering algorithm in organizing data together constitute an overwhelming reason to use it. A clustering algorithm relieves a scientist or data analyst of the treacherous job of "looking" at a pattern matrix or a similarity matrix to detect clusters. Clustering is also useful in implementing the "divide and conquer" strategy to reduce the computational complexity of various decision-making algorithms in pattern recognition. In the clustering paradigm, no expert is available to define the categories. Cluster analysis is one component of exploratory data analysis, which means sifting through data to make sense out of measurements by whatever means are available. Cluster analysis is a child of the computer revolution and frees the analyst from time honored statistical models and procedures conceived when the human brain was aided only by pencil and paper.

The user of a clustering technique is trying to understand a set of data and to uncover whatever structure resides in the data. Clustering techniques are tools for discovery rather than ends in themselves. Their application and their interpretation are subjective, depending on the experience and perspicacity of the user. The subjective nature of the clustering problem precludes a realistic mathematical comparison of all clustering techniques.

The large number of clustering algorithms available in the literature can be broadly classified into one of the two types: (i) hierarchical or (ii) partitional. A hierarchical clustering technique imposes a hierarchical structure on the data which consist of a sequence of clusterings.

A partitional clustering technique organizes the patterns into a small number of clusters by labeling each pattern in some way. Unlike hierarchical techniques which give a sequence of partitions, a partitional clustering technique gives a single partition. A pattern matrix is usually clustered in this way, which explains the popularity of these techniques in pattern recognition and image processing. Partition techniques make use of criterion functions (square error), density estimators (mode seeking), graph structures, and nearest neighbors. Fuzzy partitional clustering deals with the overlapping case in which each pattern is allowed to belong to several classes with a measure of "belongingness" to each.

It is appropriate at this point to mention that there exists a clear distinction between clustering techniques or methods and clustering algorithms or programs. The same clustering technique can be implemented differently, resulting in several clustering algorithms.

2.6 Partitional Clustering

Partitional clustering techniques partition the given set of n patterns into C clusters, where $C << n$. The desired number of clusters, C is usually specified by the user. Partitional techniques usually operate on a pattern matrix and result in a single partition. Unlike hierarchical techniques, partitional techniques allows patterns to move from one cluster to the other so that a poor initial partition can be corrected later. A majority of partitional techniques obtain that partition which maximizes some criterion function.

2.7 Hierarchical Clustering

Hierarchical clustering techniques begin with a triangular dissimilarity matrix whose rows and columns correspond to patterns and whose entries measure dissimilarity between patterns; the larger the entry, the more dissimilar the patterns. The entries are the Euclidean distance between the patterns in the pattern space. The output of a hierarchical clustering program is a dendrogram, which is a tree showing a sequence of nested clusterings. The graphical output is the outstanding feature of such programs since several clusters are represented on the same picture. The number of patterns is limited by computational considerations.

2.8 Graph Theoretical clustering

Not all natural groupings of patterns are globular, or hyperellipsoidal in shape. For example, patterns that are spaced along a straight line or on a sheet in pattern space are well structured. The squared error methods force a Gaussian-based model on such structures and as a result may fail. Graph-theoretic methods provide one means for uncovering unconventional data structures.

Zahn[35] gives an overview of graph-theoretic methods. The basic idea is to generate a minimum spanning tree for the complete graph whose nodes are patterns and whose edge weights are Euclidean distances in the pattern space. Cutting all edges having weights greater than a user-specified threshold creates subtrees, each of which represents a cluster. The threshold is actually computed as the sample mean of the edge weights in the tree plus a user-specified number of sample standard deviations. This procedure is equivalent to cutting the dendrogram generated by the single-link hierarchical clustering procedure at a level equal to the threshold.

Zahn[35] suggests several heuristic tactics for uncovering various arrangements of patterns. One finds the longest path in the minimum spanning tree and computes a measure of density for each node in the middle half of this path. An edge connected to the node at which the density is minimum is removed before the cutting process described above is begun. The density of a node is the reciprocal of the average of the edge weights over all edges connected to the clusters. This tactic is designed to separate touching clusters. As we have mentioned before, our emphasis will be on partitional clustering techniques. We describe below three interesting partitional clustering algorithms.

2.9 Gath and Geva's UFP-ONC Algorithm

Gath and Geva[21] developed a two layer partitional clustering strategy in order to obtain a satisfactory solution to the problem of large variability in cluster shapes and densities, and to the problem of unsupervised tracking of classification prototypes. During the first step, a modification of the fuzzy K-means algorithm is carried out. There are no initial conditions on the location of cluster centroids, and classification prototypes are identified during a process of unsupervised learning. Using these prototypes, the second step involves the utilization of a second clustering algorithm in order to achieve optimal fuzzy partition. This scheme is iterated for increasing the number of clusters in the data set, computing performance measures in each run, until partition of an optimal number of subgroups is obtained.

For hyperellipsoidal clusters, as well as in the presence of variable cluster densities and unequal numbers of data points in each cluster, an "exponential" distance measure $d_e^2(X_j,V_i)$, based on maximum likelihood estimation is defined. This distance is used in the calculation of $h(i|X_j)$ posterior probability (the probability of selecting the ith cluster given the *j*th feature vector):

$$h(i|X_j) = \frac{1/(d_e^2(X_j,V_k))}{\sum_{k=1}^{K} 1/(d_e^2(X_j,V_k))} \qquad (2.1)$$

$$d_e^2(X_j,V_i) = \frac{[det(F_i)]^{1/2}}{P_i} exp[(X_j - V_i)^T F_i^{-1} (X_j - V_i)/2] \qquad (2.2)$$

where F_i is the fuzzy covariance matrix of the *i*th cluster, and P_i, the *a priori* probability of selecting the *i*th cluster. The main difference in the fuzzy *K*-means algorithm is that of the calculation of centroids and of the memberships. The memberships are updated using the expression for the posterior probability as given above. In addition to the computation of the new centroid, calculation of P_i, the a priori probability of selecting the *i*th cluster is also needed as shown below:

$$P_i = \frac{1}{N}\sum_{j=1}^{N} h(i|X_j) \qquad (2.3)$$

and of F_i, the fuzzy covariance matrix of the *i*th cluster:

$$F_i = \frac{\sum_{j=1}^{N} h(i|X_j)(X_i - V_j)(X_i - V_j)^T}{\sum_{j=1}^{N} h(i|X_j)} \qquad (2.4)$$

Due to the "exponential" distance function incorporated in this algorithm it seeks an optimum in a narrow local region. It therefore does not perform well, and might be even unstable during unsupervised identification of classification prototypes. Its major advantage is obtaining good partition results in cases of unequally variable features and densities, but only when starting from "good" classification prototypes. A goal directed approach to the cluster validity problem is chosen, where the goal is classification in the sense of minimization of the classification error rate. This criterion is discussed in Chapter 4.

2.10 Gustafson and Kessel's Algorithm

Gustafson and Kessel[22] proposed an interesting modification of the fuzzy c-means algorithms which attempts to recognize the fact that different clusters in the same data set X may have differing geometric shapes. Since the norm controls the basic shape of all clusters identified with the functional $J_m(U, v)$ via the topological structure of open sets in the norm metric it induces, perhaps local variation of the norm would allow a modified objective function to identify clusters of various shapes which are locally compatible with different topological structures in the same data set. Mathematical realization of this idea is accomplished by considering the class of inner product norms induced on R^p by symmetric, positive-definite matrices in V_{pp}. Let us denote by A a c-tuple of such matrices, $A=(A_1,A_2,...,A_c)$ and let the weighted inner product induced on R^p by A_i be $\langle x, x \rangle_{Ai} = \| x \|^2_{Ai} = x^T A_i x$; the distance between x, y ∈ R^p in the weighted norm is $\| x\text{-}y \|_{Ai}$ The Functional will now take the form

$$J_m(U, v, A) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m \|x_k - v_i\|^2_{A_i} \qquad (2.5)$$

The functional defined above is the same as before. The basic difference is that all distances $\{d_{ik}\}$ are measured by a pre-specified norm; whereas (possibly) c different norm metrics-one for each $u_i \in U$ - are being sought for the functional. To render the minimization of the functional with respect to A tractable, each A_i is constrained by requiring the determinant of A_i, $\det(A_i)$ to be fixed. Specification of $\det(A_i)=\rho_i > 0$ for each i=1 to c amounts to constraining the of cluster u_i along the ith axis. Allowing A_i to vary while keeping its determinant fixed thus corresponds to seeking an optimal cluster shape fitting the x_ks to a fixed volume for each u_i.

2.10.1 Gustafson and Kessel's Adaptive Norm Theorem:

Let $\eta : \mathrm{PD}^c \rightarrow \Re, \eta(A) = J(U, v, A)$ where $(U, v) \in M_{fc} X R^{cp}$ are fixed. If m>1 and for each j,$\det(A_j)=\rho_i$ is fixed, then A^* is a local minimum of η only if

$$A_i^* = [\rho_i det(S_{fi})]^{(1/\rho)} (S_{fi}^{-1}), 1 \leq j \leq c \qquad (2.6)$$

where

$$S = \sum_{k=1}^{n} (u_{ik})^m (x_k - v_i)(x_k - v_i)^T \qquad (2.7)$$

is the fuzzy scatter matrix of u_i.

2.11 Dave's Fuzzy C-Shells Clustering

In objective functional based fuzzy clustering algorithms the weighted sum of the distances of the feature vectors from cluster prototypes are minimized. The fuzzy memberships are utilized as weighting factors. The cluster prototype can be a point or a line or a plane, etc. The fuzzy c-shells clustering(FCS) method as introduced by Dave[7] assumes a cluster structure that is of some p-dimensional hyperspherical shells which are simply circles when p=2. Hyperspheres refer to boundaries (surfaces for p>2). The prototypes do not include

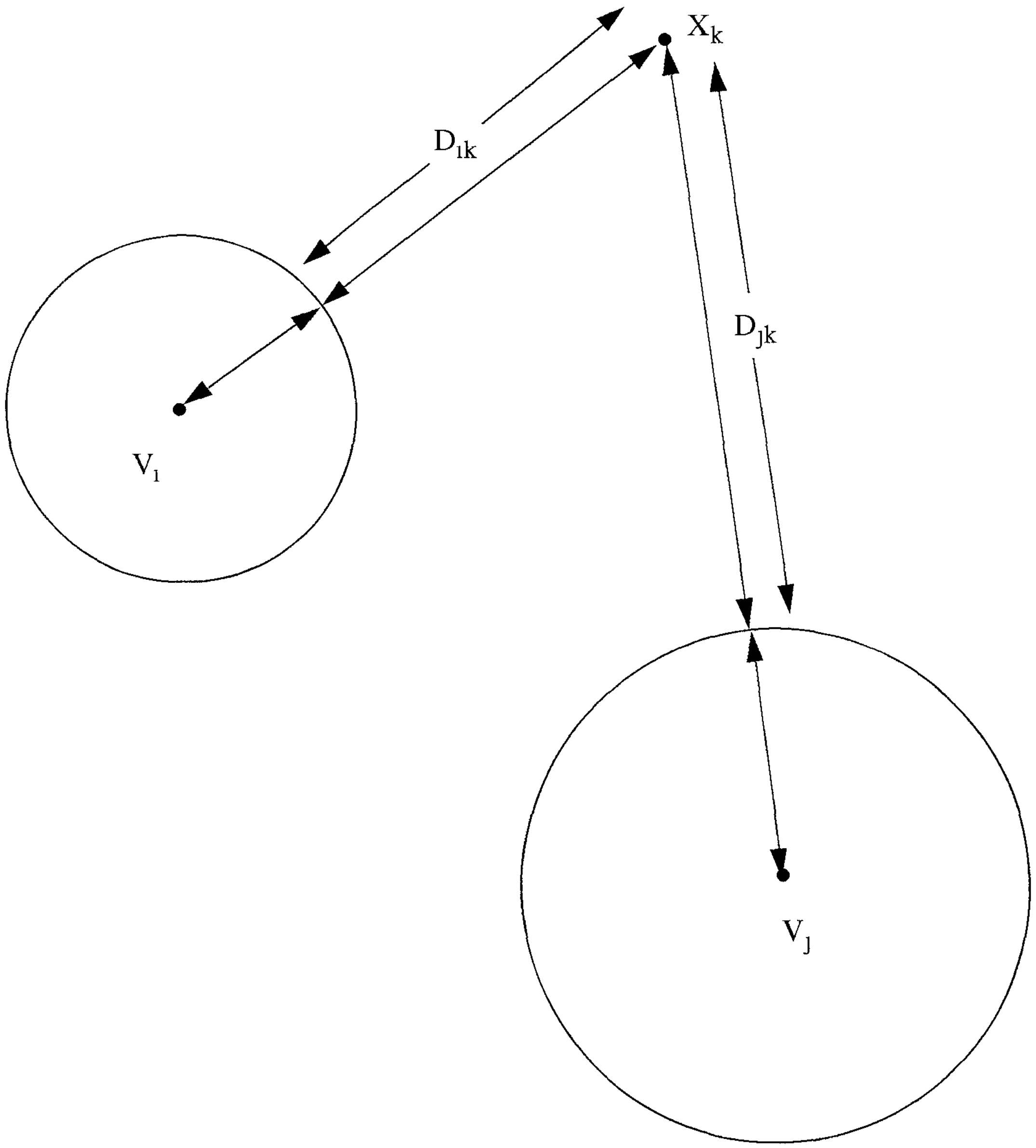


Fig 2.4 **The FCS Prototype**

interiors - whence the word "shells" to describe the cluster prototypes $\{\mathbf{P_i}\}$. Thus, the i^{th} prototypical shell $(\mathbf{P_i})$ is parameterized by a shell center $\mathbf{v_i} \in \Re^S$ and a shell radius $\mathbf{r_i} \in \Re$: $\mathbf{P_i} = (\mathbf{v_i}, \mathbf{r_i}) \in \Re^{\mathbf{S+1}}$. Note that $\mathbf{P_i}$ is used here both to denote the general prototypical entity (e.g., $\mathbf{P_i}$ is a line, plane, hyperplane, sphere, etc.); and the specific parameters that identify a member of the general family, $\mathbf{P_i} = (\mathbf{v_i}, \mathbf{r_i})$. For a given set of data **X** and choice of constant **m>1**, the corresponding FCS objective function $\mathbf{J_m}:\mathbf{M_{fcn}}\ \mathbf{X}\ \Re^{\mathbf{CS}}\ \mathbf{X}\ \Re^{\mathbf{C}} \rightarrow \Re$ is defined as

$$J(U,P;X) = \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m(D_{ik}), \qquad \text{where} \qquad U \in M_{fcn} \qquad (2.8)$$

Figure 2.4 depicts the geometry of the measure of (dis)similarity used by fuzzy c-shells for s=2. D_{ik} is the (squared) Euclidean distance between data point $\mathbf{x_k}$ and (a line tangent to) circle $\mathbf{P_i}$; more generally, this is the minimum (squared) Euclidean distance to the plane tangent to hypersphere $\mathbf{P_i}$ at the point of intersection of $\mathbf{P_i}$ with the line connecting $\mathbf{x_k}$ to center $\mathbf{v_i}$ in any number of dimensions. The general form of the FCS function in (2.8) is that of the fuzzy c-varieties (FCV) objective functions given in Bezdek[4]. The major difference between FCV and FCS is the nature of the cluster prototypes $\{\mathbf{P_i}\}$. FCV fits linear varieties to clusters, whereas FCS uses hyperspheres for the class paradigms.

The following description of the FCS algorithm uses the main iteration described below as soon as the distance between successive iterates is less than a certain threshold.

2.11.1 Fuzzy C-Shells (FCS) Algorithm

1. Choose **c**, $\mathbf{2 \leq c < n}$ and fix **m**, $\mathbf{1 < m} < \infty$.

2. Set the iteration counter **q**=0. Initialize a fuzzy c-partition **U**(0).

3. Calculate centers $\mathbf{v_i}^{(q+1)}$ and radii $\mathbf{r_i}^{(q+1)}$ by solving (2.1) and (2.2) simultaneously

$$\sum_{k=1}^{n} (u_{ik})^{m} (1 - r_i / \|x_k - v_i\|_E) (x_k - v_i) = 0 \tag{2.9}$$

$$\sum_{k=1}^{n} (u_{ik})^{m} (\|x_k - v_i\|_E - r_i) = 0 \tag{2.10}$$

4. Calculate the $\mathbf{D_{ik}}$ using equation

$$D_{ik} = |\|x_k - v_i\| - r_i| \tag{2.11}$$

5. Update memberships at $\mathbf{q}^{th}$ iteration to $\mathbf{U}^{(q+1)}$ using (2.14) and (2.15): For every **k**, define

$$I_k = \{i|\ 1 \le i \le c\}\ D_{ik} = 0; and \qquad 0 \tag{2.12}$$

and

$$\bar{I}_k = \{1, 2, \ldots, c\} - I_k \tag{2.13}$$

$$If \qquad I_k = \phi, then \qquad u_{ik} = 1/(\sum (D_{ik}/D_{jk})^{1/(m-1)}); \tag{2.14}$$

$$If \qquad I_k \neq \phi, then \qquad u_{ik} = 0 \qquad \forall i \in \bar{I}_k; and \sum u_{ik} = 1 \tag{2.15}$$

6. Terminate if $\mathbf{U}^{(q)}$ is close to $\mathbf{U}^{(q+1)}$ in a convenient matrix norm. If $\|\ \mathbf{U}^{(q)} - \mathbf{U}^{(q+1)}\ \| < \varepsilon$, then stop; else set **q** = **q** + 1 and go to step 3.

3

CHAPTER 3

Adaptive Norm FCS Algorithm

3.1 Introduction

The concept of linear or piecewise linear separation plays an important role in pattern recognition and neural networks. Many existing methods use a single hyperplane or a number of hyperplanes as separators for classification. One of the main advantages in using hyperplanes as separators is that there exist training algorithms for determining these separators. The problem with these methods is that in general one may have to use a large number of hyperplanes to separate a set of points belonging to different classes. For instance, to form a bounded region in a *d*-dimensional Euclidean space R^d, at least $d+1$ hyperplanes are needed. In contrast, if hyperspheres are adopted as separators, many classification problems can be simplified. In the worst case, one needs to a maximum of *m* hyperspheres to separate the *m* points because there are at most *m* different classes.

Dave proposed the FCS (Fuzzy *C*-Shells Clustering) algorithm using the shell as the prototype to partition the data into hyperspheres. The FCS algorithm measures the distances in a Euclidean norm. This introduces a certain measure of error in measuring distances that are non-Euclidean. Dave also proposed an adaptive norm theorem in which a positive definite matrix induces a norm for points "in" its cluster. The adaptive fuzzy *c*-shells clustering algorithm incorporates the adaptive norm theorem and is shown to provide with an optimal classification.

3.2 Fuzzy C-Shells Clustering and its limitations

The fuzzy *c*-shells clustering(FCS) method assumes a cluster structure that is of some p-dimensional hyperspherical shells which are simply circles when p=2. In the case of a sub-structure that has a norm other than the Euclidean norm or in a case where the data has a tendency to cluster with variable norm, a significant amount of error will result from measuring distances in a norm that is not Euclidean. Thus there is a need to develop an algorithm that will induce for each cluster a norm "in" it. In other words, a positive definite matrix A_i is assumed to induce a norm for points "in" its cluster. The hyperellipse resulting from such a formulation for each cluster derives its center from the data, and its orientation and axial stretch in the principal directions from the eigen structure of A_i. It is clear that one cannot specify *a priori* possible starting norms for a given set of data. Thus one would expect that an algorithm would adaptively adjust the shape of the shells during the computation much like the adaptive version of FCM discussed by Gustafson and Kessel[19]. This requires making the matrices $\{A_i\}$ variables of the optimization problem itself. Dave proposed the Adaptive Norm theorem that makes the norm a variable of the optimization as given below.

3.3 The Adaptive Norm Theorem

Let $(\mathbf{U},\mathbf{v},\mathbf{r}) \subset \mathbf{M}_{\mathbf{fc}} \times \Re^{\mathbf{cp}} \times \Re^{\mathbf{c}}$ be fixed. For $m > 1$ and for each ı, $\det(A_i) = \rho_\iota$ fixed, the A^* ıs a local mınımum of the functional only if

where

$$A_i = [\rho_\iota det(S_{sfi})]^{(1/\rho)} (S_{sfi})^{-1} \qquad 1 \le i \le c \tag{3.1}$$

$$S_{sfi} = \sum_{k=1}^{n} (u_{ik})^m \frac{D_{ik}}{d_{\iota k}} (x_k - v_i)(x_k - v_\iota)^T \tag{3.2}$$

The above theorem provides the necessary condıtions for mınımızatıon wıth respect to A_i's. The proof for the above theorem can be obtaıned through the Lagrange multiplıer technique and ıs given in Appendix D.

3.4 Implementation

Standard Picard ıteratıon is used to set up an algorıthm for the method. To solve the non-linear equatıons that result from the constraıned optımızation of the objectıve functional, the Newton's method ıs used. Bezdek and Hathway [12] show that an exact solution of (3.1) and (3.2) ıs not requıred at every step of the fixed poınt iteration. They recommend performing only a sıngle step of Newton's method, this may save computer tıme, since every step of Newton's method requires sıgnıfıcant amount of computatıons. On the other hand, doing a single step may require a greater overall number of ıteratıons. Both the strategies, namely, single step and full step are tried for the above algorithm. The overall fixed point iteration scheme using the above algorithm is not guaranteed to converge to a global minima. Due to the complexity of the solution space, a global minima can be achieved only if the initial partition is close enough to the expected partition. If the initial partition is *close* enough to the expected partition, the chances of convergence to the solution are improved. This is a problem that is common to all such algorithms. Furthermore, the non-linearity of the resulting equations imposes the constraint of a good initial guess.

In order to ensure a good initialization, a two layer strategy was developed. Two methods of Initialization of the fuzzy c-partitions were fashioned to deal with the two classes of substructures that arise as described below. After the initial fuzzy c-partition is obtained in such a manner, an iterative initialization scheme to be described later is employed.

3.4.1 Initialization of the fuzzy c-partitions:

One could classify the commonly encountered range of substructures occurring in the data into one of the following two categories:

1. ***Type 1***: those that are either well separated or contain a mix of well separated and overlapping clusters or

2.***Type 2:*** those that contain either concentric clusters or those that contain predominantly overlapping clusters

In the first case, we hope to obtain good partitions by using a quick partition algorithm whose only objective will be to do a single pass through the objects and provide a favorable starting partition based on a proximity heuristic. Appendix C describes the quick partitioning algorithm. In the second case, we let the data assume an arbitrary or random partition. The utility of such an approach lies in the propensity of such an initialized sub-structure towards convergence. In both the cases, the starting values for the prototypes are computed from

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_i}{\sum_{k=1}^{n} (u_{ik})^m} \tag{3.3}$$

$$r_i = \frac{\sum_{k=1}^{n} (u_{ik})^m [(x_k - v_i)^T I (x_k - v_i)]^{1/2}}{\sum_{k=1}^{n} (u_{ik})^m} \tag{3.4}$$

We fix the norm by using the Euclidean norm. We then proceed to apply an iterative initialization algorithm as described below to provide us with a good partition.

3.4.2 Iterative Initialization Scheme:

Once we have the initial fuzzy c-partition as obtained in the manner discussed in the previous section and the corresponding starting values for the prototypes from equations (3.3) and (3.4), we can apply an iterative algorithm to help us in achieving good starting guesses for the Newton's method to work on. Essentially the aim is to apply a specific technique depending on the kind of data set one encounters. The process of automating the choice can be carried out by picking the partition with a lower functional value at the point of application of the AFCS algorithm.

The iterative scheme we have in mind assumes an iterative form very similar to that of the FCM algorithm. However, rather than measure the distances from a point prototype, we will let the algorithm take a course that will involve the measurement of the distance from a point prototype and the measurement of the distance from a shell prototype at different times. The scheduling of the points of entry of the measurement of distance from a specific prototype are pre-determined empirically. The proportion of the algorithm to be used for a specific prototype is varied between two values one each for the two classes of sub-structures we are most likely to encounter. For the first type, we allow the point prototype to be used for the first four iterations and the shell prototype to be used for the next six iterations. For the second type of structure we let all the iterations use the measurement from the shell prototype. As this scheme is primarily meant to provide with good starting guesses, we restrict the maximum number of iterations to 10. As we have said, the norm used is Euclidean at this point and is fixed. The other two parameters *viz.*, the centers and radii are computed from the fuzzy mean as given by (3.3) and the fuzzy radius as given by (3.4). The reasoning behind the usage of two different prototypes during the initialization scheme lies in the fact that the FCM type algorithm has a tendency to split the data and hence its applicability to the data of type1 during the early stages. The FCS prototype on the other hand can detect concentric shapes and is thus applicable to the kind of data described by the structure in type2.

More formally, we can describe the initialization scheme as follows:

1. Choose $\mathbf{c}$, $\mathbf{2} \le \mathbf{c} < \mathbf{n}$ and fix $\mathbf{m}$, $\mathbf{1} < \mathbf{m} < \infty$.

2. Set the iteration counter $\mathbf{q=0}$. Initialize a fuzzy *c*-partition $\mathbf{U(0)}$ as explained in the above(i.e., either by using the quick partitioning algorithm or through an arbitrary choice).

3. Calculate centers $\mathbf{v_i}^{(\mathbf{q+1})}$ and radii $\mathbf{r_i}^{(\mathbf{q+1})}$ by solving (3.3) and (3.4) sequentially.

4. Calculate the distance using D_{ik} as follows:

If type1 then

$$\left.\begin{array}{l}\text{for } 0 < q \ \le 4, \text{ use the FCM prototype.}\\ \text{for } 4 < q \le 10, \text{ use the FCS prototype.}\end{array}\right\} \qquad (3.5)$$

```
else if type2 then
        use the FCS prototype.                                  (3.6)
5. Update memberships at $\mathbf{q^{th}}$ iteration to $\mathbf{U^{(q+1)}}$ using equations given by 2.12 - 2.15
```

Once the initialization is done, the Newton's method is used to solve the resulting non-linear equations. The Jacobian constructed to apply the Newton's method is symmetric and so the numerical method is well behaved. The Jacobian is reproduced in its entirety in Appendix E. To render the minimization of the functional with respect to A tractable, each A_i is constrained by requiring the determinant to be fixed. We denote this fixed value by ρ_1. It is given the value unity. This results in the conservation of the volume even after the transformation which seems reasonable enough. Appendix B gives some formal theorems along with a brief discussion on the norm and what it represents. The fixed point iteration solves for the centers and radii for fixed A_i and then solves for A_i keeping the centers and radii fixed. Convergence is tested by using the max norm.

3.5 Adaptive Norm Fuzzy C-Shells (FCS) Algorithm

The algorithm consists of two phases. In the first phase we use an iterative initialization scheme as described in the previous section to obtain good initial guesses following which we apply the AFCS algorithm in its actual form.

Phase I:

Iterative Initialization algorithm (refer to the previous section).

Phase II:

The AFCS Algorithm

1. Choose $\mathbf{c}$, $\mathbf{2 \leq c < n}$ and fix $\mathbf{m}$, $\mathbf{1 < m < \infty}$.

2. Set the iteration counter **q=0**. Initialize a fuzzy *c*-partition **U(0)** as explained in the above.

3. Calculate centers $\mathbf{v}_\mathbf{i}^{(\mathbf{q+1})}$ and radii $\mathbf{r}_\mathbf{i}^{(\mathbf{q+1})}$ by solving (3.7) and (3.8) simultaneously:

$$\sum_{k=1}^{n} (u_{ik})^m (1 - r_i/\|x_k - v_i\|_{A_i}) (x_k - v_i) = 0; and \qquad 0 \tag{3.7}$$

$$\sum_{k=1}^{n} (u_{ik})^m (\|x_k - v_i\|_{A_i} - r_i) = 0 \tag{3.8}$$

4. Calculate the $\mathbf{D_{ik}}$ using equation (3.9)

$$D_{ik} = \left| \|x_k - v_i\|_{A_i} - r_i \right|$$

5. Update memberships at $\mathbf{q}^{\mathbf{th}}$ iteration to $\mathrm{U}^{(\mathbf{q+1})}$ using equations 2.12 - 2.15.

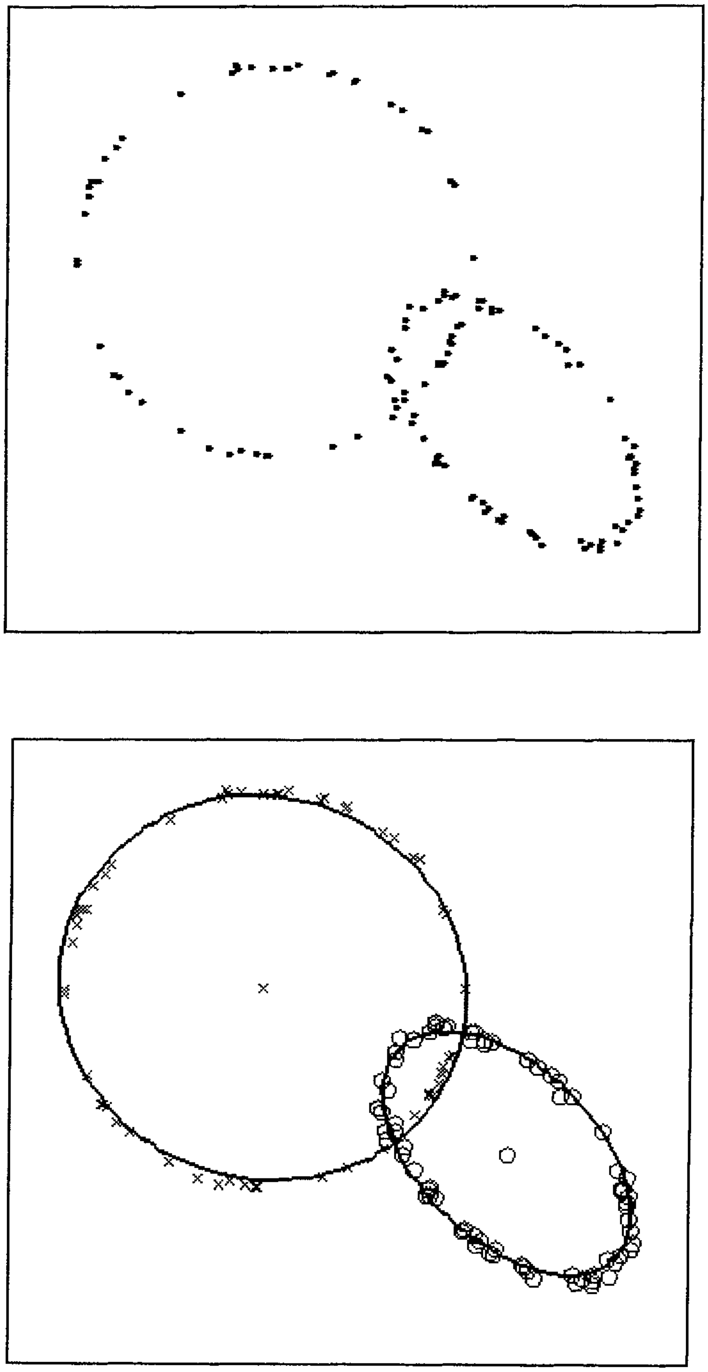
6. Terminate if $\mathrm{U}^{(\mathrm{q})}$ is close to $\mathrm{U}^{(\mathbf{q+1})}$ in a convenient matrix norm. If $\| \mathrm{U}^{(\mathrm{q})} - \mathrm{U}^{(\mathrm{q+1})} \| < \varepsilon$, then stop; else set **q=q+1** and go to Step3

3.6 Numerical Examples

The Adaptive fuzzy c-shells algorithm was tested out on two dimensional data-sets. The coordinates of the points thus constitute the components of the feature vectors thus obtained. The number of clusters is given by *c*. The initialization in all of the cases was good. Fig 3.1 shows an ellipse and a circle each slightly overlapping the other. This data set could be treated as being of type1. Fig 3.2 shows two ellipses barely touching each other, again an example of the type1. Figs 3.3 and 3.4 show examples of type 2. Fig 3.3 constitutes a set of concentric ellipses while fig 3.4 shows three overlapping ellipses. The last example shows three contiguous variable eccentricity ellipses. We can classify them as belonging to type 1. As can be seen the substructures detected by the AFCS algorithm are optimal for the examples shown. Convergence in all of the examples occurred within 5 iterations of the AFCS algorithm.

3.7 Conclusions

The algorithm has performed very well for the examples shown. Previous fuzzy clustering algorithms were not crafted to deal with concentric structures such as the data set containing concentric ellipses. The initialization scheme described in this chapter is specifically designed to detect the presence of such structures. The fact that the algorithm has been able to achieve global convergence with wide ranging differences in structures such as those shown in the examples can be attributed to the ability of the initialization scheme to provide with good starting values. The efficiency of the intialization scheme is conveyed by the small number of iterations(usually within five iterations for most of the data sets the algorithm was tested on), that is needed for convergence. A comparison with FCES for similar data sets shows the AFCS algorithm to be superior and it's ability to detect circular as well as elliptical data in two dimensions is clearly of great significance. In the current algorithm an implicit form of the norm gets entrapped in the expression for the distance from the prototype. It therefore remains to be seen if the norm can be solved for explicitly. It would also be interesting to explore the consequences of choosing a value other than unity for the restraint imposed on the determinant of the norm.



A Circle and an Ellipse

Fig 3.1

Two Touching Ellipses

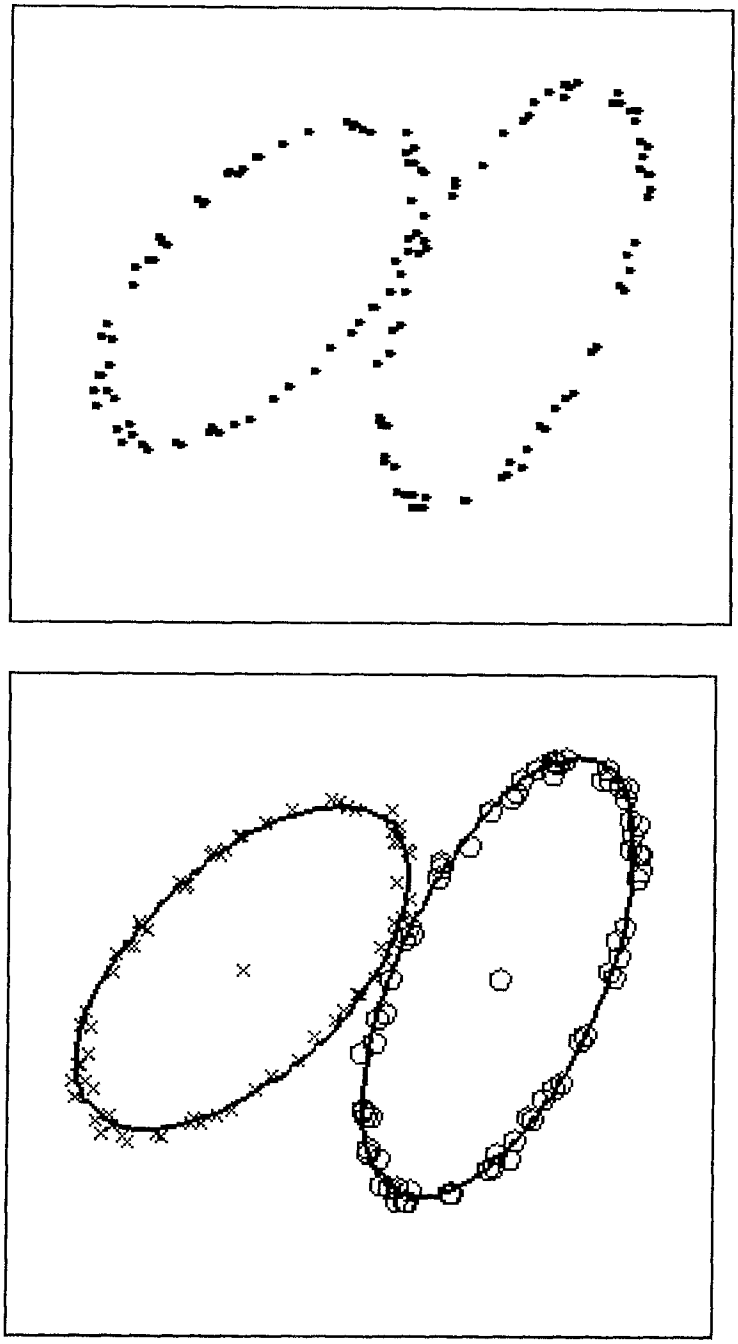
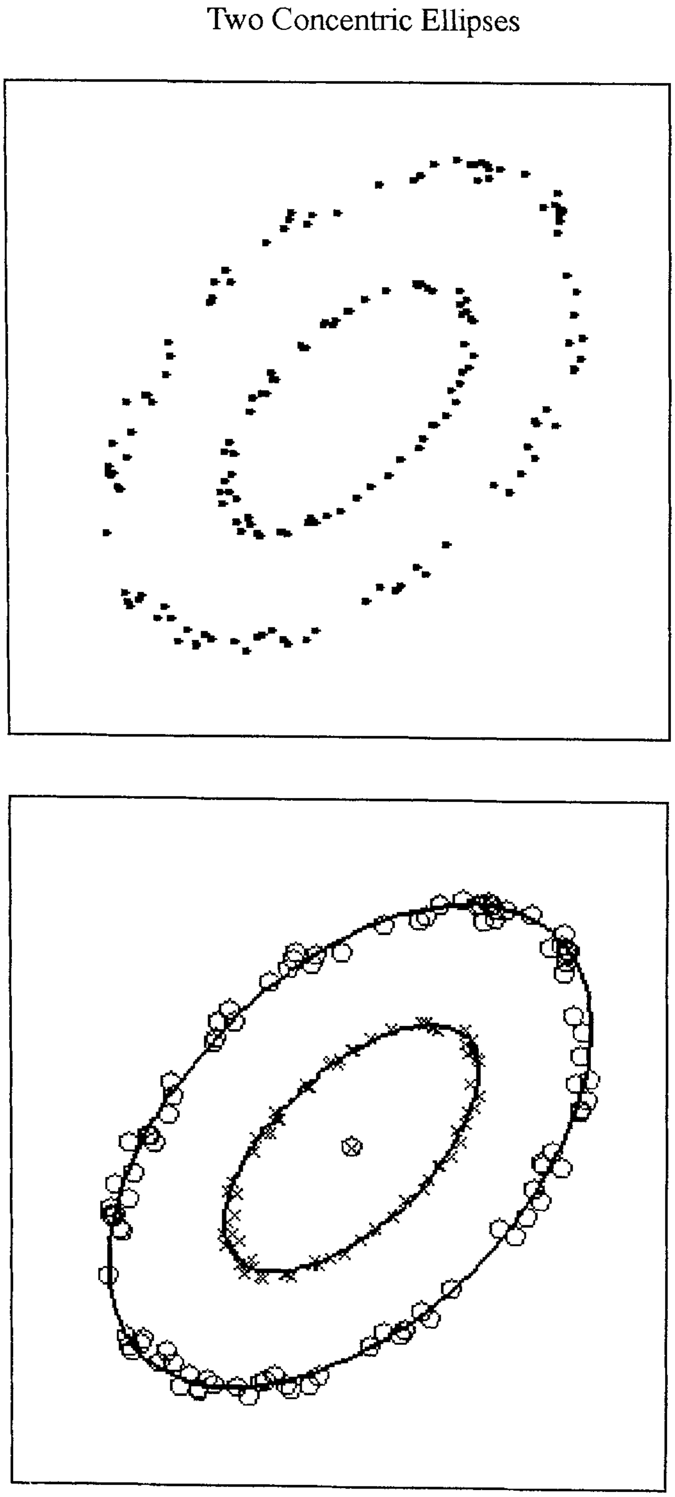


Fig 3.2



Two Concentric Ellipses

Fig 3.3

Two ellipses and a circle

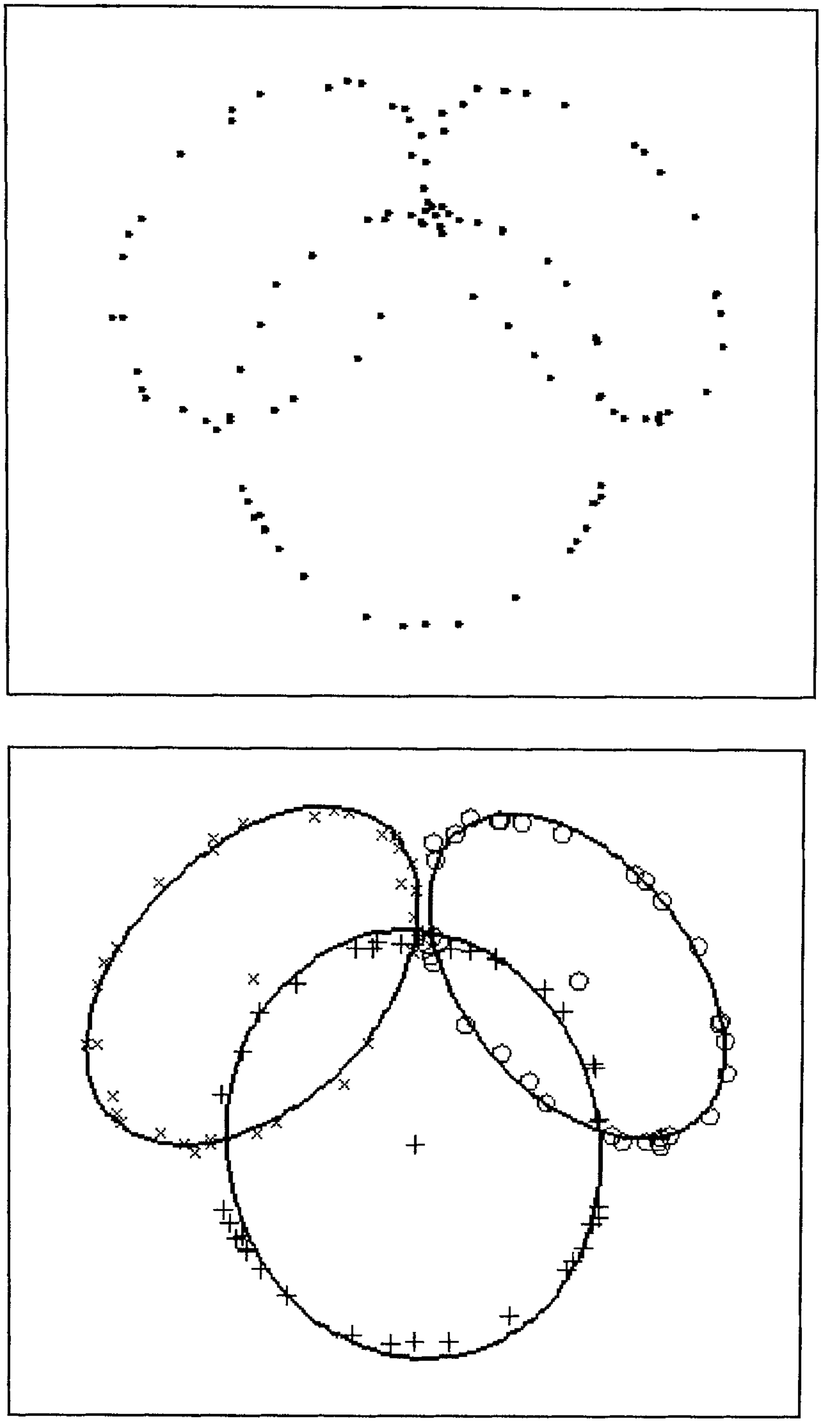


Fig 3.4

Three Touching Ellipses

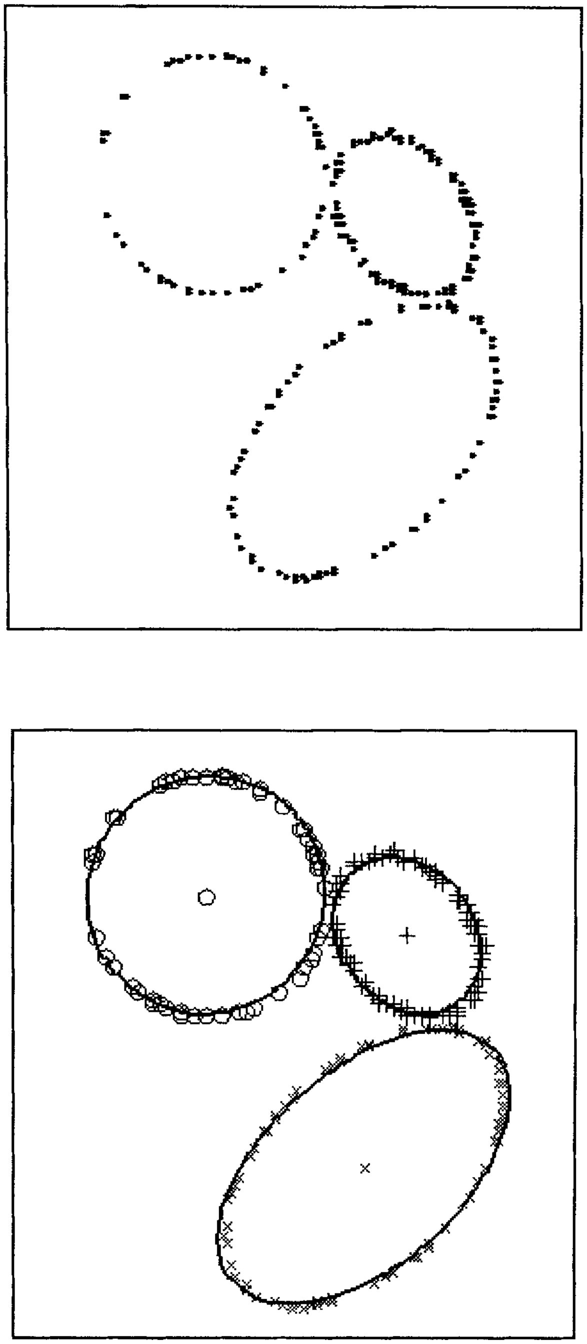


Fig 3.5

4

CHAPTER 4

Cluster Validity

4.1 Introduction

In the previous chapter several numerical examples involving two-dimensional data sets were used to illustrate the effectiveness of the AFCS algorithm. It was assumed that the number of clusters present was known and that the structure resulting from the application of the algorithm was somehow the "best". In practice, it may be plausible to expect a "good" partition at more than one value of c, the number of subsets. In the absence of quantitative indicators, the evaluation of the resulting structure becomes suspect. In the case of two-dimensional data sets our ability to visualize the geometrical properties of the clusters can help us evaluate the results. But even such a vindication can prove to be subjective in the case of some prototypes. Thus it is natural to formulate a criterion or quantitative measure that will help in realizing an objective evaluation of the cluster substructure. The dilemma of deciding the number of clusters as well as the evaluation of the resulting substructure fall in the domain of what has come to be called cluster validity.

The Validity problem can be restated from Dubes and Jain[16] as follows:
"Are the clusters achieved in a particular clustering significant enough to provide evidence for hypotheses about the underlying structure of the data? In other words is the clustering "real" or merely an artifact of the clustering algorithm? The difficulty of validating clusters lies in our inability to agree on the definition of the difficulty of determining the statistical distribution of the validity measure."

Dubes and Jain[18] contend that validating the results of imposing a structure on data with a clustering method requires clear definitions of the following four structural criteria. These criteria are not independent and must be blended into a workable methodology.

(1) Compactness criterion: measures the inner strength, or concentration or cohesion or uniqueness of an individual cluster with respect to its environment.

(2) Isolation criterion: measures the distinctiveness or separation or gaps between a cluster and its environment.

(3) Global fit criterion: measures the accuracy with which the structure describes the relationships between clusters, as well as the extent to which all the clusters are individually valid.

(4) Intrinsic dimensionality criterion: determines the "shape" of a cluster and provides information about representing the patterns in a cluster.

They give an excellent review of the various problems related to cluster validity as applied to measuring the clustering tendency and the fit of hierarchical and partitional structures.

In standard classification problems, a correct classification exists against which to evaluate, and often a measure of the "goodness" of such schemes involves a simple count of misclassified points or a normalized percentage error of such a count. In the clustering situation, the operating algorithm's *modus operandi* is dictated by the data themselves and as such an absolute scheme cannot be readily constructed. As Bezdek[4] points out, the problem of formulating a cluster validity index for a "good" cluster rests on an even more delicate issue - that of what is meant by a "cluster". The principal difficulty is that the data X and every fuzzy partition $U \in M_{fc}$ of X are separated by the algorithm generating U (and defining "cluster" in the process). We will however focus our attention on evaluating structures given by the shell paradigm.

We will give a brief discussion of the validity indices proposed for fuzzy clustering algorithms following which we will introduce the new measures for the validation of the FCS algorithms. Our emphasis will be on understanding the basic approach of the old measures rather than analyzing how they were constructed or why they fail when they do.

4.2 Validity Functionals

It would seem that in the case of an objective functional based clustering algorithm, the functional could double as a performance measuring index. Duda and Hart[19] discuss formalizing such a notion by advancing a null hypothesis whose acceptance becomes contingent on the existence within limits of the corresponding probability of false rejection. They express the problem of formalizing the use of a criterion function to evaluate the structure as one of determining the practicality of computing the sampling distribution for the criterion function.

4.2.1 Partition Coefficient

Bezdek[5] attempted to define a performance measure based on minimizing the overall content of pairwise fuzzy intersection in U the partition matrix.

He defined an index, the partition coefficient as

$$F(U;c) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^{2}/n$$

where U is the partition matrix.

Fc(U) indicates the average relative amount of membership sharing done between pairs of fuzzy subsets in **U** the partition matrix by combining into a single number, the average contents of pairs of fuzzy algebraic products. The range of variation of **Fc(U)** is [1/c,1]. This would seem to suggest the reliance of the measure upon *c*. The disadvantages of the partition coefficient as stated by Bezdek[4] are its monotonic tendency and lack of direct connection to some property of the data themselves.

4.2.2 Fuzzy Set Decomposition Measure

Backer and Jain[1] address the problem of Validity through a performance measure based on Fuzzy set decomposition which we shall call the Fuzzy set decomposition measure (or FDM for short). The first step involves obtaining a c-collection of induced fuzzy sets. The partition is then characterized as follows. If the amount of induced fuzziness is high it means that the collection of induced fuzzy sets is reasonably separable and that the inducing partition reflects the real data structure reasonably well. On the other hand, if the amount of induced fuzziness is low, it means that the inter-fuzzy set separability is low and that either the inducing partition does not reflect the real structure well, or that almost no structure is present in the data. Thus their performance measure should measure the fuzziness in the gaps between fuzzy sets (along the fuzzy boundaries) and therefore, should be based on the notion of intersection of fuzzy sets. They adopt the definition of the intersection of fuzzy sets as given by the following expression.

$$f_{i \cap j}(u) = f_i(u) \cdot f_j(u)$$

Thus the performance measure is defined as follows:

$$FDM = 1 - \frac{2c}{c-1} \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} \frac{1}{N} \sum_{u \in M_{fc}} f_i(u) \cdot f_j(u)$$

The value of the performance measure lies between 0 and 1 corresponding to no-fuzziness and maximum fuzziness.

Like the PC measure, this measure also utilizes the fuzzy algebraic product, but it's formulation by eliminating any dependence upon c induces a transformation. This results in a stretched range between 0 and 1 that should be able to better distinguish any nuances between partitions. Even then, it might be inadequate for the detection of the FCS substructure and has been shown to fail. The intersection of two fuzzy sets can also be expressed as follows:

$$f_{i \cap j}(u) = |f_i(u) - f_j(u)|$$

This shows that the performance measure can be looked upon as a cluster membership distance measure. They use it to express the utility of the results of different clustering algorithms applied to a single database and rank them. The comparative analysis puts an order on the utility of different clustering results as a consequence of the performance measure producing unique values. They contend however that the subjective nature of the utility demands for a realistic goal-directed relationship to the application domain. The use is made of the classification error rate of nearest-neighbor (NN) classifier to corroborate the values given by the performance measure. Their goal is stated as one of obtaining a set of design samples for a classifier and they show the performance measure to be somehow related with the classification error probability experimentally. The use of a goal directed comparison is shown to yield lesser number of errors in the testing phase. Choosing the "best" clustering is inferred to become meaningful if a goal (classification) is defined. In essence the method employed uses fuzzy set theory to estimate the overlap of the fuzzy sets and gives an estimate of the fuzzy cluster separability. They suggest the measurement of gaps between fuzzy clusters as well served if classification were the ultimate goal.

4.2.3 Classification Entropy(CE)

The concept of association based on entropy has been discussed in statistical literature and has been applied to fuzzy sets. Thus a scheme that uses an entropy measure based on fuzzy sets should acquire a minimum for a hard partition. Shannon[30] required that the measure of two dependent schemes be additive under conditioning of one scheme's measure by dependence on the other. Bezdek[4] defines the Classification Entropy(CE) of any fuzzy c-partition U as

$$H(U;c) = -\sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}\log_a(u_{ik})/n$$

where logarithmic base $a \in (1,\infty)$ and $u_{ik}\log_a(u_{ik})=0$ whenever $u_{ik} = 0$. It is based on the fact that the closer the probability of success of an event the closer one is to evaluating the solution. The purpose of the logarithm is to make the value additive. This would then give us an estimate at the step being considered. In order to estimate the question itself the product as shown in the above expression is taken. The classification entropy which should have a minimum for a good classification. H is a scalar measure of the amount of fuzziness in a given $U \in M_{fc}$. According to Bezdek[4], the limitations of the classification entropy can be attributed to its apparent monotonicity and to an extent, to the heuristic nature of the rationale underlying its formulation.

4.2.4 Proportion Exponent(PE)

Windham[33] proposed a measure that attempts to overcome the sensitivity to parameters that F and H tend to suffer from and can be defined as follows:

$$P(U;c) = -\log_e \prod_{j=1}^{n} \left[\sum_{k=1}^{I_j} (-1)^{k+1} \binom{c}{k} (1 - k\mu_j)^{c-1} \right]$$

where μ_j is defined as $\mu_j = \max_i(u_{ij})$ and I_j is defined to be the greatest integer in $1/\mu_j$. The key building block of the proportion exponent(PE) is the proportion of membership functions whose maximum exceeds a given value. The proportion exponent attempted to overcome the difficulty that the previous measures had of suffering from sensitivity to parameters by looking not at the values of measure of quality directly, but by looking at how they compared to a standard. The higher the maximum, the better the point is classified. The maximum value itself was not used as an indicator of quality but rather the probability that one could do better by selecting the memberships for the data at random.

It should be noted that the proportion exponent may not be defined for a particular membership matrix U. If one of the columns describes a hard cluster, the proportion exponent is undefined since the proportion of functions whose maximums exceed one is zero. One assumes that such an occurrence is rare in the case of fuzzy clustering algorithms. The proportion exponent also suffers from monotonicity as shown by Windham[33].

4.2.5 Hypervolume and Partition Density

The fuzzy volume and the fuzzy density of the cluster may also be used as indices of cluster validity. It can be argued that a good partition should yield a high value for the fuzzy partition density and a low value for the fuzzy hypervolume. Such measures have been used in the past, for example by Gath and Geva[21].

4.3 Cluster Validity for Shell Clustering

In the previous section, some of the popular cluster validity measures were presented. It was shown that all these measures suffered from some kind of problem or the other. In this section, new measures which are specifically designed for the c-shells type clusters are presented. The behavior of the schemes on which the older measures were based is unraveled.

4.3.1 Need for new measures

The criteria described above are inadequate because of the following:

1. They rely heavily on the information in the partition matrix U.

2. Though the measures described above use the basic heuristic (and in some sense a goal) that good clusters are "not fuzzy", they rely solely upon the memberships. It is debatable if the memberships alone can convey in total the nature of the underlying substructure detected.

3. The functionals constructed above are monotonic in most cases over the number of clusters.

4. Most of the measures described above are designed for round or other blob like prototypes and may not be used in the same sense for the c-shells clustering prototype that do not possess any interior.

4.3.2 Fuzzy Hypervolume

The volume of the clusters in the fuzzy partition is obtained using the formula

$$F_{HV} = \sum_{k=1}^{n} [\det(F_i)]^{1/2}$$

where F_i denotes the ith Fuzzy covariance matrix. For c-shells clustering, this matrix is defined as below in terms of the fuzzy shell scatter matrix introduced by Dave[11].

$$F_i = \frac{S_{(shell)\,i}}{\sum_{k=1}^{n} (u_{ik})^m} = \frac{\sum_{k=1}^{n} (u_{ik})^m \left(\frac{D_{ik}}{d_{ik}}\right) (x_k - v_i)(x_k - v_i)^T}{\sum_{k=1}^{n} (u_{ik})^m}$$

where $S_{(shell)i}$ stands for the ith shell-scatter matrix. Thus the value of F_{HV} gives a measure of the volume. A partition can be expected to have a low value for this measure if the partition is really tight. In other words if the points tend to cluster very close to the prototype, the volume will be minimal. Thus an extremum for this index would ideally indicate a good partition.

4.3.3 Average Partition Density

The average partition density measures the average of the density per cluster taken over all clusters, given by the following equation.

$$D_{PA} = \frac{1}{n} \sum_{k=1}^{n} \frac{S_{ui}}{[\det(F_i)]^{1/2}}$$

where S_{ui} is the “sum of central members”. is given by:

$$S_{ui} = \sum_{k=1}^{n} (u_{ik})^{m},$$

$$\forall x_k \in \{x_k : |\|x_k - v_i\| - r_i| < \sigma\}$$

where σ stands for standard deviation, thus taking into account only those members that lie within the shell whose radii are the standard deviations of the clusters features.

4.3.4 Partition Density

The partition density is calculated from

$$P_D = \frac{S}{F_{HV}}$$

where

$$S = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik} \qquad \forall x_k \in \{x_k : |\|x_k - v_i\| - r_i| < \sigma\}$$

where σ stands for standard deviation. By considering only the central members of the cluster, only the points contributing to the core of the cluster will be taken into account. This emphasizes the fuzzy environment present and gives meaning to the term dense partition. Thus for a given data set a dense partition with a low fuzzy hypervolume will in effect be indicative of an optimum partition. In other words, if the points around all the prototypes cluster tightest for a given partition, we will in all likelihood have the densest partition density and the lowest fuzzy hypervolume.

4.3.5 Shell thickness(ST) measure

Dave[11] proposed a validity index for measuring the thickness of the shell. By measuring the scatter of the points about the prototype, the new index measures the thickness or in other words the closeness of the points in the partition to the data. The thickness is normalized to the scale of the data-set through dividing by the average cluster radius. The normalized measure has a very interesting connotation. By being able to give a measure of the thickness of the shell for a given partition, this index should be able to forewarn the user of the lack of existence of shell sub-structure in the data. Thus in addition to being able to validate a given partition, its absolute value should give us an indication of the kind of substructure we can expect in the data. The actual absolute value will be dependent on the average shell thickness the user might expect in the data. In the absence of such a value, one can take an empirical standard and allow a 5-10 percent variation about this value. Clustering algorithms will generate a clustering whether one really exists or not. This index should indeed be very useful in being able to predict if the algorithm is trying to impose a structure on the data. The thickness for the shell is defined as follows.

$$T = \frac{\sum_{i=1}^{c}\sum_{k=1}^{n} (u_{ik})^{m}(\|x_k - v_i\|_{A_i} - r_i) / \sum_{k=1}^{n} (u_{ik})^{m}}{\sum_{i=1}^{c} (r_i)/c}$$

4.4 Numerical Results

In order to test the performance of the validity criteria, synthetic data in 2 dimensions were used. By choosing the distances between cluster prototypes to be near each other, and controlling the variance in the two dimensions, overlapping between clusters could be obtained, resulting in a fuzzy environment. The number of subgroups in the data, their density, and number of data points in each subgroup were subject to variation. We present three datasets pertaining to structures that represent typical variations in densities and numbers and one data set containing uniformly distributed points. The datasets on which the indices were used are shown towards the end of this chapter followed by the results in a table format.

In order to depict the effectiveness of the validity criteria in a graphical manner, the following plots are constructed:

1) Fuzzy Hypervolume as a function of subgroups in the data.

2) Partition density as a function of the number of the subgroups and

3) Shell thickness as a function of the number of subsets present in the data.

For each example, the figures containing the data and the partitions for the correct number of subsets present are shown on one page, while the plots and the table containing the values obtained for the measures considered are shown on another page. Accordingly we label the page containing the figures by A and the page containing the plot by B. The tables are plotted along with the plots.

In each of the examples considered, the validity measures were computed as a function of the number of subsets in the data. On the limiting side there can be at least two clusters in any given data. Thus we begin with 2 clusters and increase the number by one until we have a change in the gradient of the index being measured. In practice it is possible to have more than one partition for a specified number of subsets. The best partition achievable by the algorithm is considered in evaluating the performance of the validity measures. In other words, for a specified number of subsets, the partition that had the lowest value for the objective functional was used. In the event of multiple optima within the range of the number of subsets over which the measures were employed, the first optima were considered in the case of all the measures except for the FDM. The FDM measure is an absolute measure and so the optimum with the highest value was chosen when multiple contiguous optima occurred.

The first example, shown in Fig. 4.1A consists of the data in form of two ellipses having low eccentricity and one overlapping ellipse having high eccentricity as well as high point density. The figure shows three correct ellipses plotted over the partition depicted by use of different markers. The plot of the three measures viz., FHV, PD and ST and the table of the numerical values of all the validity measures are shown in Fig. 4.1B. The shaded cell values in the table refer to the optima detected for the corresponding measures while the values in boldface refer to the optima for the correct number of subsets. All the validity measures except the PE concur on choosing the number of subsets as three. The number of subsets chosen by PE is four. This seems to substantiate the apparent weakness of the scheme on which the PE is based. The heuristic on which the PE is based, *viz.,.* that of estimating the proportion of membership functions exceeding a given value seems to fail in detecting the number of sub-sets. For examples such as these, there is hardly any doubt about the existence of three subsets, and the failure of PE clearly indicates its poor performance.

The second example, shown in Fig. 4.2A shows five circular clusters. The validity measures for this example in a tabular form and the plot is shown in Fig 4.2B. Only the measures ST and FDM correctly identify the presence of 5 clusters. If one were to use the global optima for the PC and CE, it would result in the correct identification of five clusters by these measures. But with the first optimum, the PC and CE choose two as the number of subsets. The PE shows a little monotonicity at first, followed by indeterminate values for 4, 5, and 6 number of subsets. This confirms our belief about the PE becoming undefined upon encountering a point that has a hard membership in a particular cluster. The FHV and PD just miss choosing five and instead choose six as the correct number of clusters. This seems to suggest that these measures might need to be normalized in the sense of nullifying the discrepancies arising out of the formation of clusters from only a few points and in the process causing these measures to have nearly equal adjacent extrema as is evident from the table.

The third example, shown in Fig. 4.3A, consists of four ellipses. One can readily perceive the presence of two well separated pairs of contiguous ellipses all of which possess a fair degree of eccentricity. As can be seen from the table in fig 4.3B, all the validity indices based solely on the memberships, viz., the FDM, PC, CE and the PE pick two as the correct number of subsets present in the data. The corresponding fit is shown in Fig 4.3A (iii). All the new measures introduced, i.e. FHV, PD, and ST correctly identify three as the number of sub-sets. This particular example provides one with an insight into why even a relatively better measure such as the FDM can fail. The failure of the FDM, PC, CE, and PE measures can be traced back to the apparent presence of two very well separated partitions which have ensconced within them the four prototypes we are looking for. This seems to indicate that at least with the shell prototype, the construction of a measure based on partitions alone need not necessarily reflect the underlying structure in the data.

The fourth example shown in Fig 4.4A is similar to Fig 4.3A, but has 3 ellipses, with variable densities and sizes. This example has been included to show that the results are identical for examples with well partitioned structures even if they vary to a good degree in size and density. The data and the fit are shown in Fig 4.4A while the plot and the table are shown in Fig 4.4B.

Finally an example of a data set containing uniformly distributed points is shown in fig4.5A. This dataset does not contain any shell substructure. The partition obtained for 11-partition is shown in Fig 4.5A as this was the partition for which most of the measures considered here seemed to have an optimum. Only the FDM and the ST measures can be examined for the absolute values. Even while this is so, the FDM measure picks 11 as the number of subsets present in the data. It must be noted that the FDM value by itself does qualify the exact nature of the substructure, while the ST can be used to tell us if a shell substructure exists by examining its absolute value. The optimum obtained for the ST is 0.33892 which clearly shows that the data has no shell substructure in the data. All the other measures seem to have varied results for this dataset. The optimum for the FHV is 918.52320 which is quite high. Even though the FHV is not a normalized measure and cannot be examined for pointers to the substructure present, the high value clearly is indicative of the lack of a shell substructure.

4.5 Conclusions

The examples reveal interesting features of the measures discussed here. The measures based on the partitions alone have been shown to fail invariably if not in one example, at least in the other. Although none of these measures succeeded in the last example, it was observed that the FDM measure was correct in most of the cases. It is clear that the measures that are exclusively geared towards finding good partitions can loose out on the real issue which is the validity of the prototypes and are therefore inadequate as was demonstrated by the last example. The new measures, namely ST, FHV, and PD, seem to perform consistently for these and several other examples that were tried. Although this paper considers only fuzzy memberships, the new measures are also applicable for hard c-shells clustering. Amongst the three new measures, the one introduced by Dave seems to be the best, as it always picks the correct partition. The other advantage of the measure ST is that it also gives a more or less absolute measure of the goodness of fit for the shell prototype. This measure is also invariant to the scale of the data, unlike FHV and PD. For the examples having only circles, this measure attains values less than 0.1 for a good partition, and for the elliptical examples, it attains values less than 0.25. Thus it seems to provide a good indication of not only the correct number of sub-sets, but also indicates how good the partition itself is.

In summary, new measures, which are specifically designed for the shell clustering are introduced. The measure based on the shell thickness as proposed by Dave is shown to be the best. For the purpose of obtaining more evidence about the partition, the use of the hypervolume and the partition density is highly recommended.

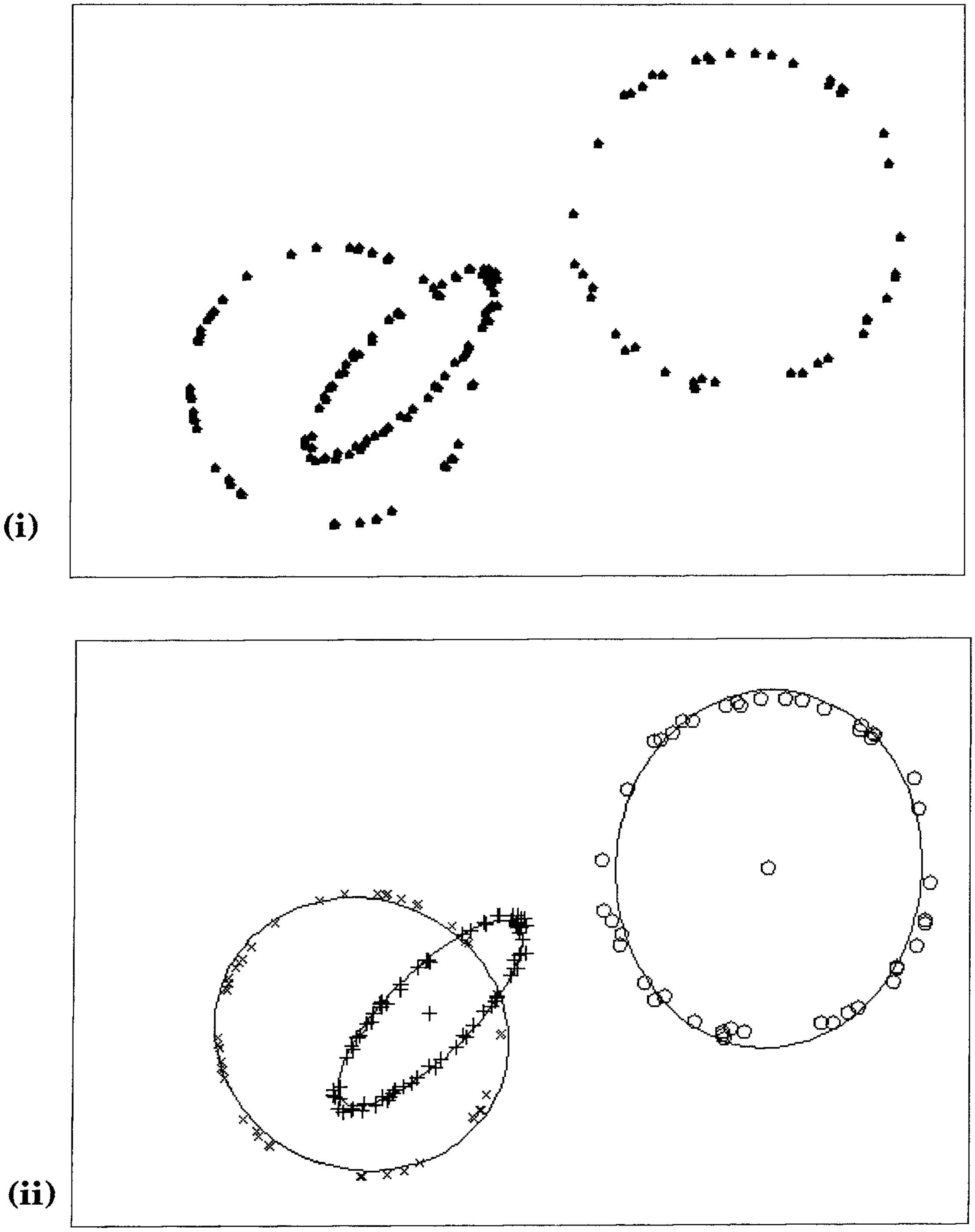


Fig 4.1A

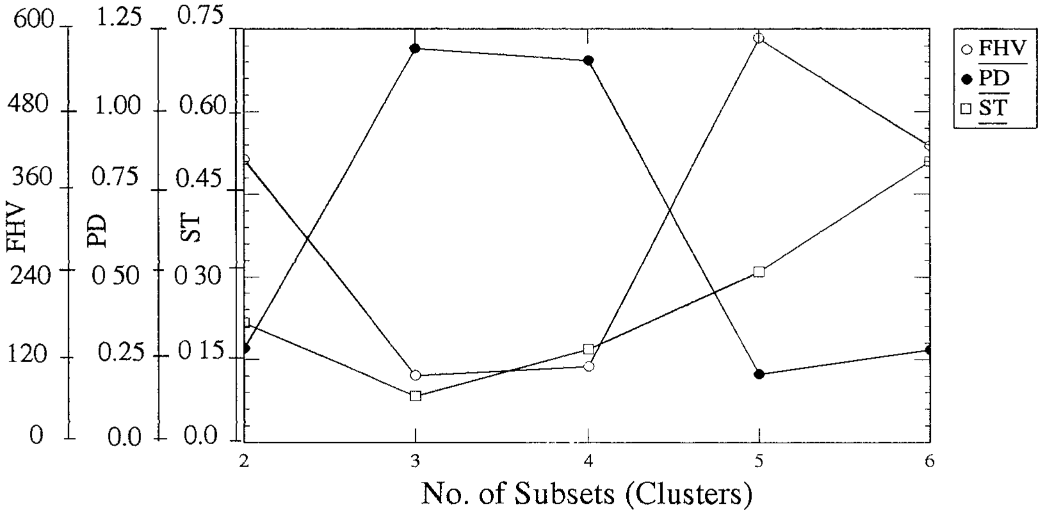


Fig 4.1B

Table 1:

No of Clusters	FDM	PC	CE	PE	FHV	PD	ST
2	0.73200	0.86664	0.223529	507.150	417.013	0.26242	0.27681
3	**0.97400**	**0.98315**	**0.037809**	**2029.28**	**98.3371**	**1.10201**	**0.10701**
4	0.91600	0 93708	0.122144	2621.80	111 872	1 06896	0.21609
5	0.77000	0 81658	0.342473	2119 56	595.829	0 19025	0.39556
6	0.82700	0 85688	0.277274	2938 33	437.028	0.25805	0 65078

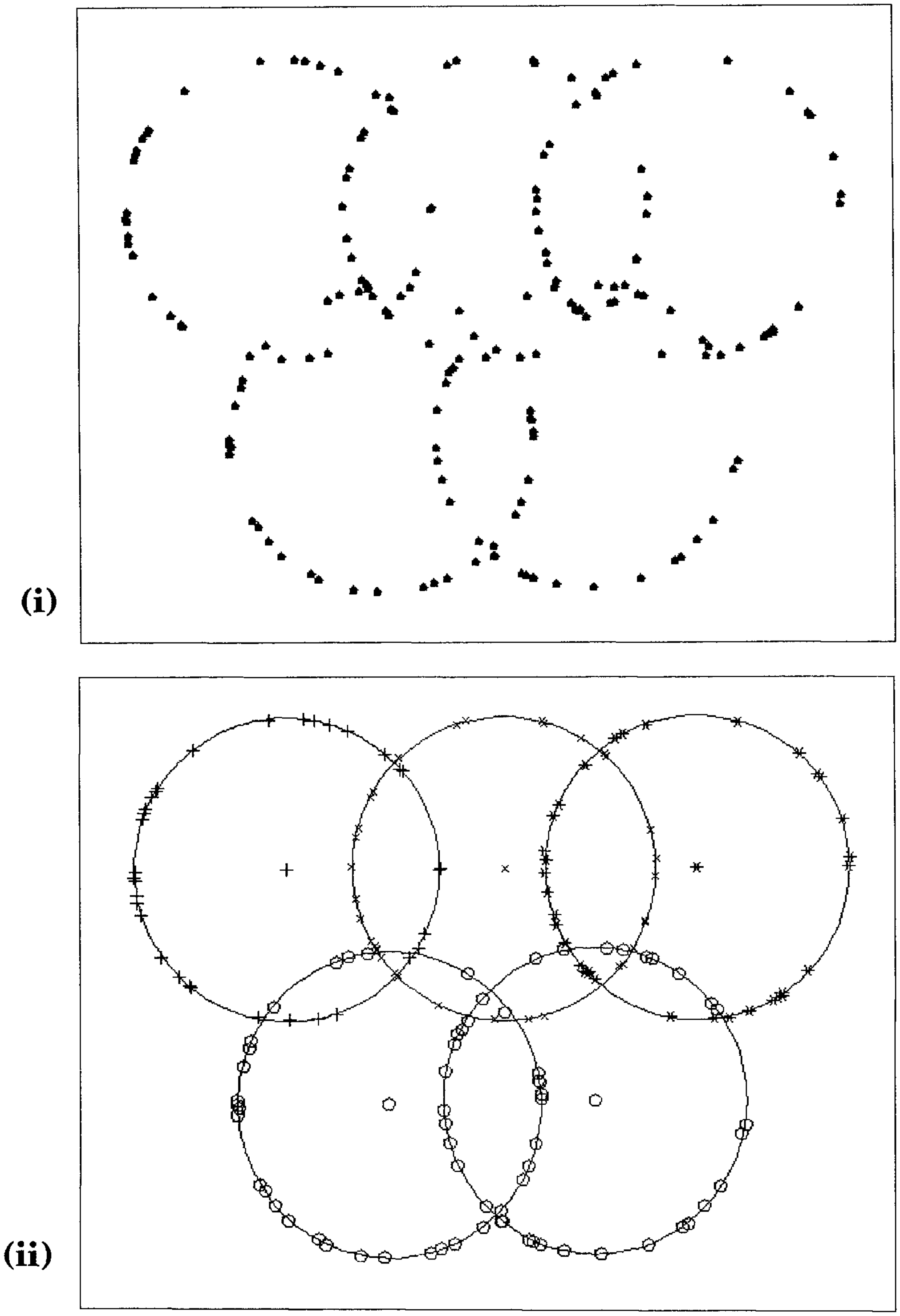


Fig 4.2A

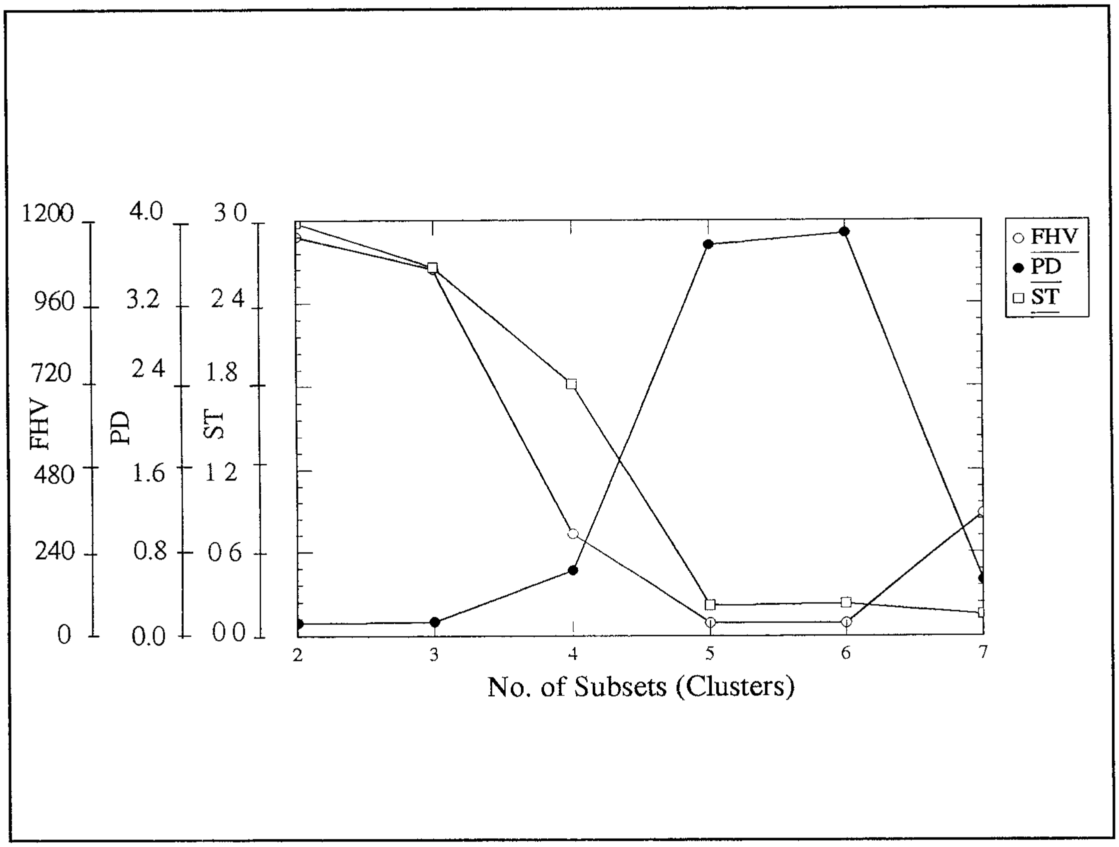


Fig 4.2B

Table 2:

No. of Clusters	FDM	PC	CE	PE	FHV	PD	ST
2	0 660000	0.830912	0.273482	462 9507	1001 503	0.112751	0.299608
3	0.679000	0.786420	0.379868	851 3699	919.0403	0.124395	0 267278
4	0.871000	0 903570	0.183165	∞	253.9715	0 569013	0 182834
5	**0.985000**	**0.988774**	**0.024481**	∞	**31.80968**	**3.436796**	**0.021660**
6	0.983000	0 986435	0.029067	∞	31.66899	3.544494	0.023247
7	0.925000	0.936027	0.132195	6775 835	308.4085	0.482212	0.015371

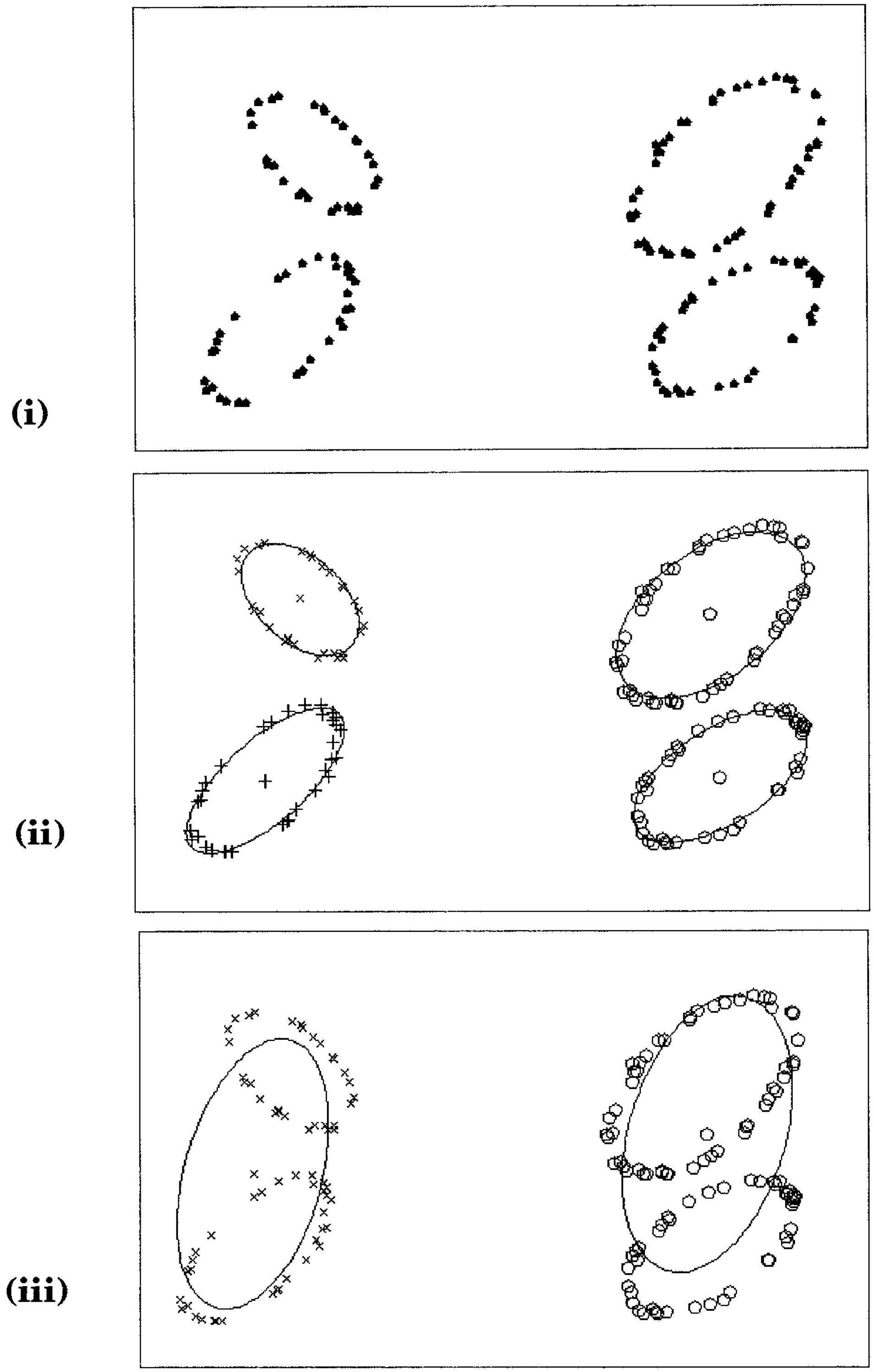


Fig 4.3A

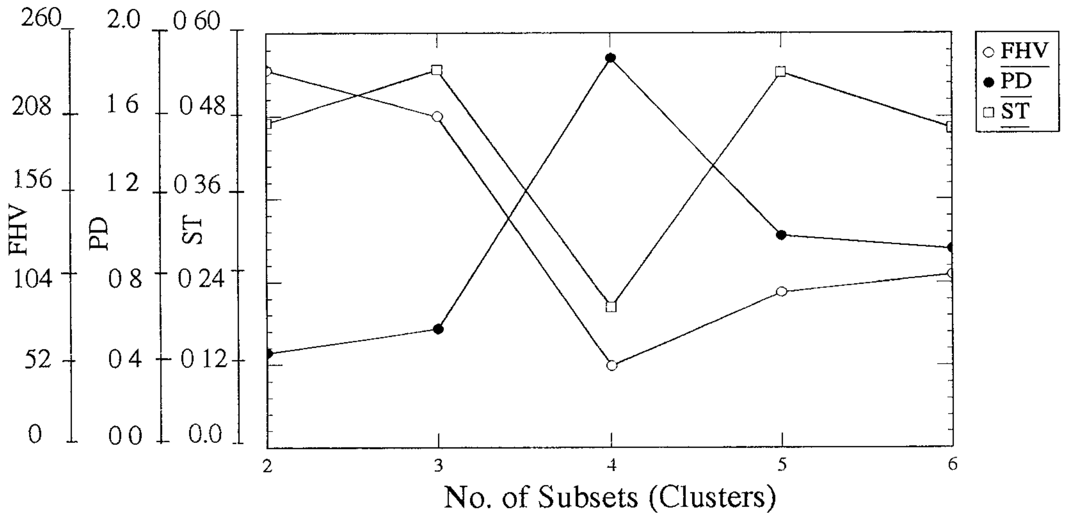


Fig 4.3B

Table 3:

No. of Clusters	FDM	PC	CE	PE	FHV	PD	ST
2	0 984575	0.992287	0,022491	∞	254 9259	0.442250	0.459806
3	0.944406	0.962711	0.089928	1591 883	224.4481	0.556712	0.535520
4	**0.973178**	**0.979883**	**0.015069**	**3244.964**	**59.10288**	**1.713435**	**0.207288**
5	0 950791	0.960632	0.090098	3544.83	105.325	0.991243	0.531263
6	0.937788	0 948156	0 105096	4244 395	117.3116	0 930702	0 453234

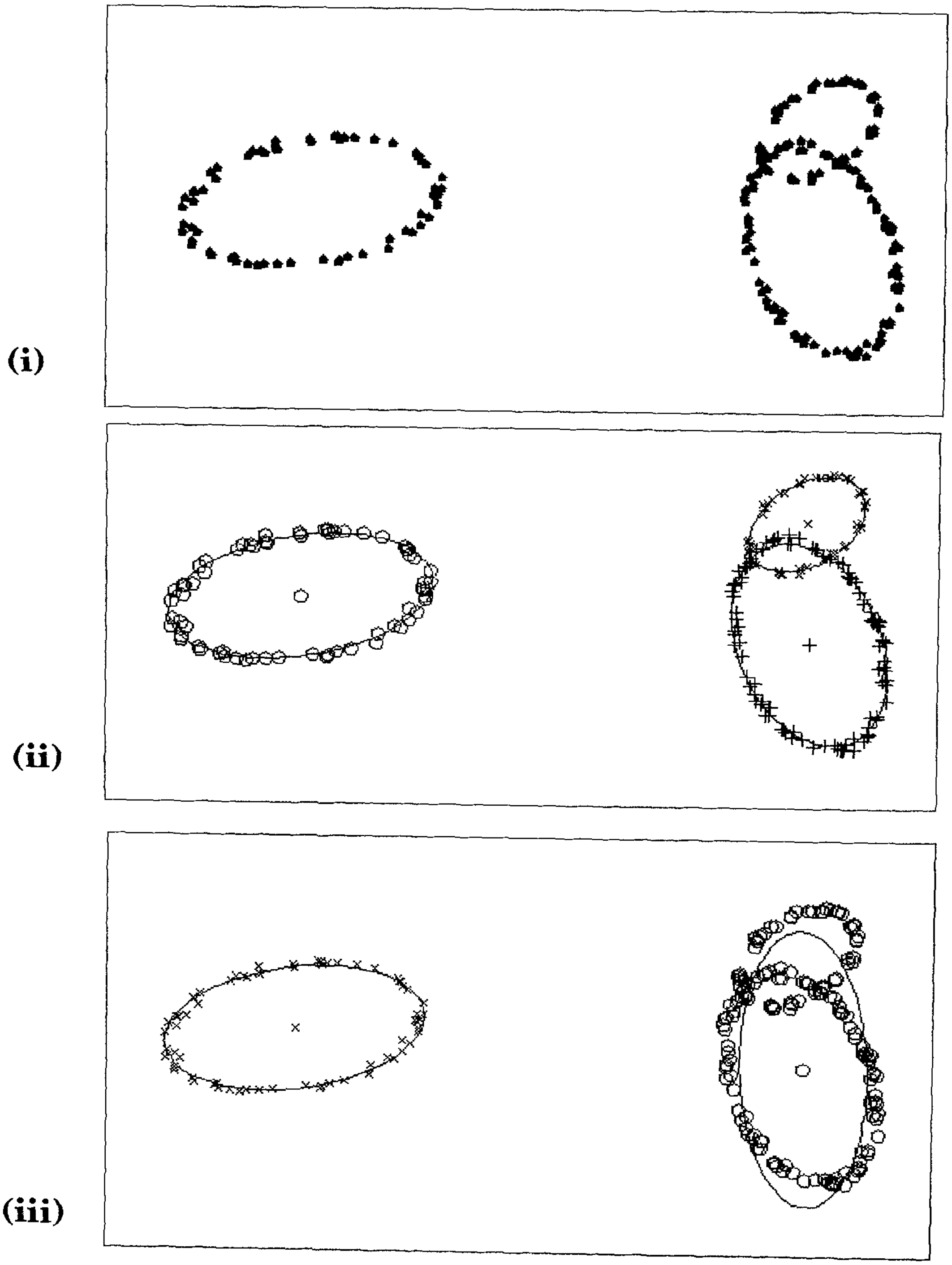


Fig 4.4A

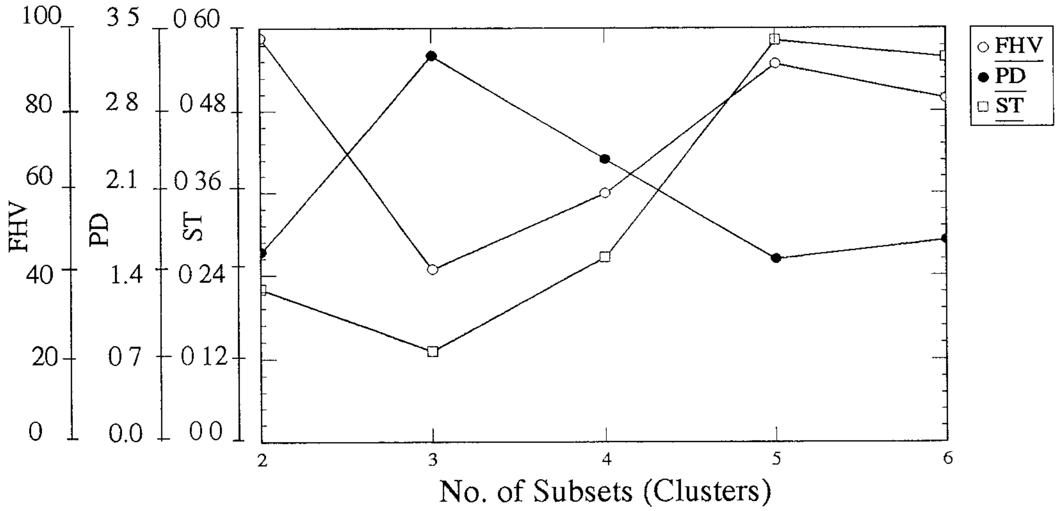


Fig 4.4B

Table 4:

No. of Clusters	FDM	PC	CE	PE	FHV	PD	ST
2	0,985372	0,992686	0,020845	∞	98.42145	1.661784	0.214708
3	**0.945929**	**0.963952**	**0.063477**	∞	**41,65735**	**3.415997**	**0.128176**
4	0.953180	0 964885	0.068954	∞	60 32405	2.491247	0 259557
5	0.929728	0 943781	0.120971	4150.444	92.14151	1 602122	0.569531
6	0 908895	0.924078	0.158056	5072.723	83 75744	1.771680	0.545642

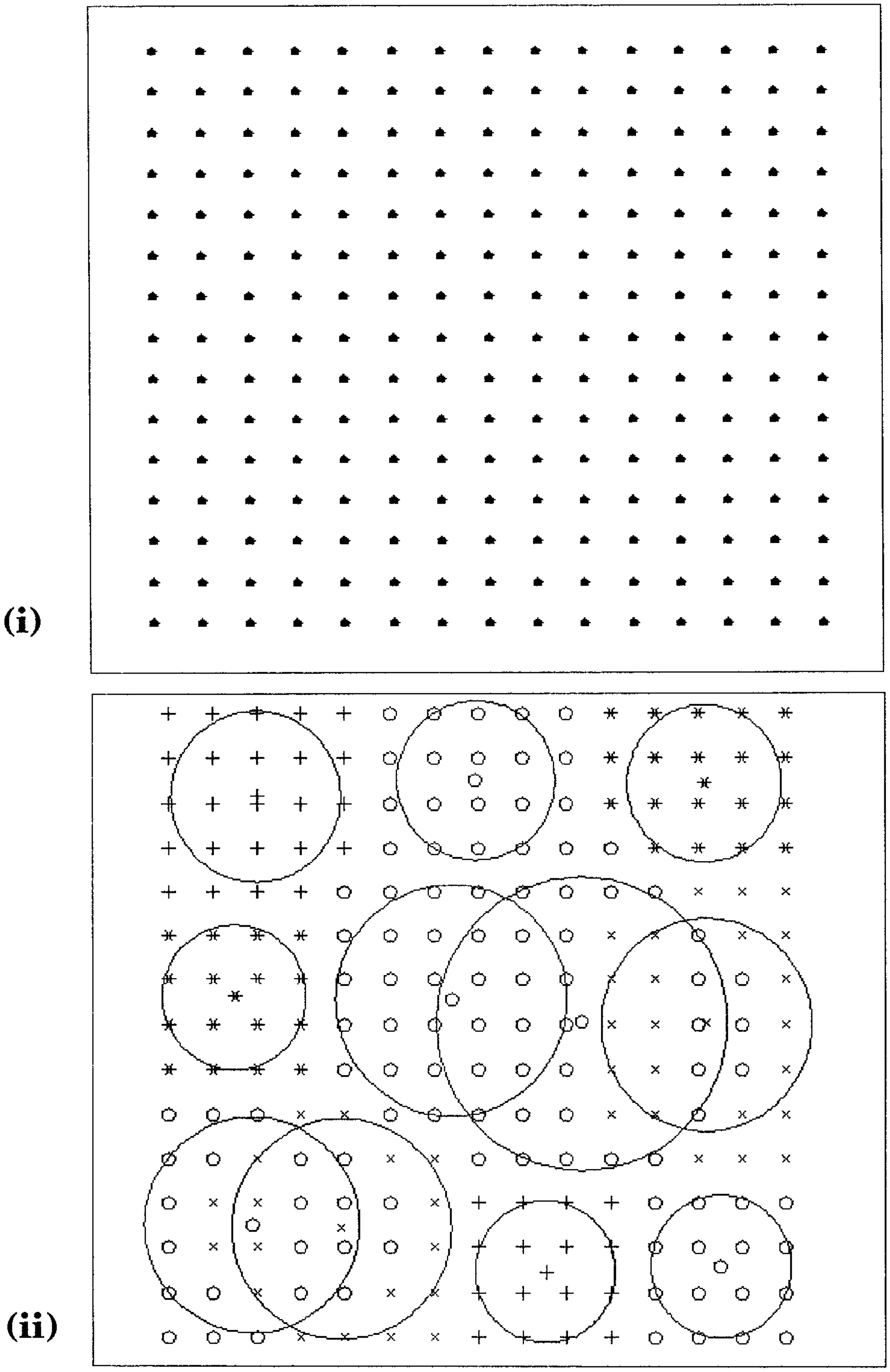


Fig 4.5A

Table 5:

No. of Clusters	FDM	PC	CE	PE	FHV	PD	ST
2	0.570055	0.785028	0 341393	449.3146	1300.013	0.111092	0.338299
3	0 566006	0 710671	0.507405	∞	1257 190	0 114156	0.358952
4	0576124	0.682092	0.590570	1358.393	1208.572	0.119851	0.359552
5	0.635888	0.708710	0 567891	1872.692	1163.267	0.124858	0.860800
6	0 652072	0 710060	0.583928	2452 831	1078.027	0.132037	0.909667
7	0 650872	0 700746	0.617616	2942.793	1087.384	0 132067	1 242963
8	0.633018	0.678888	0 666360	3362 465	1106.534	0.127780	1.159356
9	0.653018	0.691625	0.691625	3927.296	1035.124	0.138096	1.063280
10	0.666385	0.699744	0.653810	4542 154	998.5767	0.144376	1.698176
11	0.696592	0 724172	0.623478	∞	918.5233	0.160073	1.873478
12	0 609486	0.642026	0.774128	∞	1155.020	0 114643	1.688895

CHAPTER 5

Conclusion

The algorithm based on the adaptive norm theorem has been shown to be remarkably successful in detecting shapes characterized by hyperellipsoidal or spherical shells. This is especially significant because previous fuzzy clustering algorithms such as the FCES were inaccurate and would only work under a given set of constraints. The algorithm's success can be attributed to the soundness of the theorem on which it is based and the reliability of the numerical method used, viz., the Newton's method. The global convergence achieved by the algorithm for diverse structures demonstrates that the initialization scheme introduced here has worked. The initialization scheme outlined here can be used as a front end for other algorithms with similar goals.

The new measures for validating the shell substructure detected by the algorithm have proved to be effective. The measures based entirely on extracting the partitions have been shown to fail in detecting the shell substructure. The non-dependence of the validity of the shell prototype upon the partition is shown to explain the inability of the existing measures in validating the shell sub-structure. All the three new measures are shown not only as needed but also as being superior to the existing measures. The examples shown clearly demonstrate the superiority of the new measures. In summary a powerful new algorithm with a robust initialization scheme and an evaluation measure have been presented here.

APPENDIX A

Operations Upon Random Vectors

Inner Products, Vector Spaces, and Bases

Let x and y be vectors with real-valued components x_i and y_i, i=1, 2, . . . , n. The inner product (sometimes called the scalar product or dot product in physics) of x and y is defined by

$$\langle x,y \rangle = x^T y = y^T x = \Sigma x_i y_i$$

If $x^T y$ =0, then x and y are said to be orthogonal. The magnitude of a vector is defined by

$$|x| = \sqrt{x^T x} = [\Sigma x_i^2]^{1/2}$$

If a vector u = x/|x| is defined, this vector has unit magnitude and direction the same as x. The inner product can be written as

$$\langle x,y \rangle = (y^T u).\ |x|$$

and represents the orthogonal projection of y on u multiplied by the magnitude of x

The n-dimensional space in which the vectors x reside will be called the vector space and denoted by X If there exists a set of vectors u_1, u_2, . . . , u_n in X such that an arbitrary vector x can be represented by a linear combination.

$$x = a_1u_1 + a_2u_2 + \ldots + a_nu_n$$

where the a_i are real numbers, and any vector x in X has such a representation, then the $\{u_i\}$ are said to form a basis for X. The $\{a_i\}$ are called the components of x with respect to the basis $\{u_i\}$. To form a basis, it is necessary and sufficient that the $\{u_i\}$ be linearly independent; that is, no member of the set can be written as a linear combination of the other vectors. Furthermore, if the $\{u_i\}$ satisfy

$$u_i^T u_j = \{ \begin{matrix} 1 & i = j \\ 0 & i \neq j \end{matrix}$$

then the basis vectors are said to be orthonormal and can be interpreted as defining the axes of a cartesian coordinate system.

Distance Functions

The usual way to define distance is as the magnitude of the vector difference between the points. That is, if x and y are vectors representing points in the same n-dimensional vector space, we define the distance dE(x, y) between the two points as

$dE(x, y) = |x - y| = [\Sigma\ (x_i - y_i)]^{1/2}$

This is called the Euclidean distance.

More generally, a distance function d(x,y) is any scalar-valued function that satisfies the following conditions

$$\begin{aligned} d(x,y) &> 0 \quad x \neq y \\ &= 0 \quad x = y \end{aligned}$$

$$d(x,y) = d(y,x)$$

$$d(x,y) + d(y,z) \geq d(x, z)$$

The last condition is known as the triangular inequality and is particularly strong constraint. Functions meeting only the first two conditions can be useful in many analyses, although they are not true distance functions.

General Linear Transformations

A linear transformation is a mapping from a vector space X to another vector space Y is represented by a matrix. If x is a vector in X and y is the corresponding (mapped) vector in Y, then one can write

$y=Ax$

where A is the matrix that defines the linear transformation. The transformation is said to be one to one if given vector y in Y can be derived from one and only one vector x in X. The transformation is said to be onto if every vector y in Y can be derived by applying the transformation to some vector x in X. In other words, the set of all transformed vectors from the X space is the entire Y space. If the linear transformation is both one-to-one and onto, then an inverse transformation A-1 exists and one can write

$x= A^{-1}y$

Note that if A is to be invertible, it is necessary (but not sufficient) that the dimension of y is the same as the dimension of x, A is a square matrix. If this were not the case, the linear transformation would not be one-to-one.

Positive Definite Matrix

To determine if the matrix

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

is positive definite, we form the quadratic product $x^T Ax$

$$x_1{}^2a + x_1x_2(b+c) + x_2{}^2d$$

If this expression is greater than zero for all values of x_1 and x_2 except $x_1=x_2=0$, the matrix is positive definite.

Orthonormal Transformations

If S is a square matrix that satisfies the relation

$$SS^T = S^TS = I$$

then S is said to be an orthonormal transformation. This leads to S-1=ST

An orthonormal transformation preserves the magnitude of vectors because if y is given by

$$y = SX$$

then

$$|y| = \sqrt{y^Ty} = \sqrt{x^TS^TSex} = \sqrt{x^Tx} = |x|$$

One can also say that the magnitude of the vector is invariant under an orthonormal transformation. It will be shown that an orthonormal transformation can be thought of as a rotation of the basis or the coordinate frame in which the vector is represented. In other words, if x is a column matrix of the vector components with respect to the original coordinate frame, then y is a column matrix of the vector components with respect to the rotated coordinate frame.

Let $r_i{}^T$ represent the *i*th row of S. Thus y can be represented as

$$\begin{bmatrix} \leftarrow r_1^T \rightarrow \\ \leftarrow r_2^T \rightarrow \\ \\ \leftarrow r_n^T \rightarrow \end{bmatrix}^T$$

so that $\quad y_i = r_i^T x$

In addition, since $S^T = S^{-1}$, one can write

$$x = \begin{bmatrix} r_1 \; r_2 \quad r_n \end{bmatrix}$$

or

$$x = \sum_{i=1}^{n} y_i r_i$$

One can observe from the above that the $\{r_i\}$ form a set of orthonormal basis vectors and that $\{y_i\}$ are the components of x with respect to that basis.

Diagonalization by LU Decomposition

Let A be a nonsingular square matrix such that all of the submatrices formed by taking only the first k rows and columns of A are also nonsingular. Then A can be expressed as a product

$$A = LU$$

where L is lower triangular (i.e., all elements above the main diagonal are zero) and U is upper triangular (all elements below the diagonal are zero). If we require that the diagonal elements of L are all ones, that is, that L is unit lower triangular, then this factoring is unique and can be effected by Gaussian elimination of the elements in the lower triangular portion of A.

We can further note that if D a diagonal matrix consisting of the diagonal elements of U then the equation can be written as

$$A = LDU^T$$

where U is unit upper triangular (ones on the diagonal). If a matrix A is symmetric, then clearly $U^T = LT$

APPENDIX B

The Norm

We present formal definitions for the metric and the norm followed by an illustration of a normed linear space and some theorems for the quadratic representaion of the norm.

A metric space is a non-empty set where the distance between any two points is specified. The notion of distance has to retain those properties of distance which are evidently vital for the development of a sensible analysis.

Definition:

Metric:

Given a non-empty set X, a distance function d on X, called a metric for X, is a function which assigns to each pair of points a real number, (or formally, $d: X \times X \rightarrow \Re$), satisfying the following properties:

For all x, y $\in$ X

(i) $d(x, y) \geq 0$

(ii) $d(x, y) = 0$ if and only if $x = y$

(iii) $d(x, y) = d(y, x)$,

and for all x, y, z $\in$ X

(iv) $d(x, y) \leq d(x, z) + d(y, z)$, (the triangle inequality)

A non-empty set X with a metric d si denoted by (X, d) and is called a metric space. Different metrics could be defined on the same set giving rise to different metric spaces.

Definition:

Norm:

Given a linear space X over $\Re$, a norm $\| \bullet \|$ for X is a function on X which assigns to each element a real number, (or formally, $\| \bullet \| : X \rightarrow \Re$), satisfying the following properties:

For all $x \in X$

(i) $\| x \| \geq 0$

(ii) $\| x \| = 0$ if and only if $x = 0$

(iii) $\| \lambda x \| = | \lambda | \| x \|$ for any scalar λ,

and for all $x, y \in X$

(iv) $\| x + y \| \leq \| x \| + \| y \|$ (the triangular inequality)

A linear space X with a norm $\| \bullet \|$ is denoted by $(X, \| \bullet \|)$ and is called a normed linear space. Again different norms could be defined on the same linear space giving rise to different normed linear spaces.

Given a normed linear space $(X, \| \bullet \|)$, it is clear that the function $d: X \times X \rightarrow \Re$ defined by

$$d(x, y) = \| x - y \|$$

is a metric for X, and we call this the metric generated by the norm $\| \bullet \|$, So then every normed linear space is a metric space under the metric.

To develop a feel for a metric or normed linear space we need to explore the geometry associated with certain fundamental sets in the space related to the metric or norm functions.

Definition:

In a metric space (X, d), given $x_0 \in X$ and $r > 0$, the set

$$S(x_0; r) \equiv \{ x \in X : d (x_0, x) = r \}$$

is called the sphere center x_0 and radius r, the set

$$B[x_0; r] \equiv \{ x \in X : d (x_0, x) \leq r \}$$

is called the closed ball center x_0 and radius r, and the set

$$B(x_0; r) \equiv \{ x \in X : d (x_0, x) < r \}$$

is called the open ball center x_0 and radius r.

The Euclidean motivation for drawing attention to such sets is clear enough. However in some metric spaces the shape of such sets hardly accords with our Euclidean intuition. In a normed linear space the linear structure brings some orderliness.

Definition:

In a normed linear space $(X, \|\bullet\|)$, the set

$$S(\underline{0};1) \equiv \{ x \in X : \| x \| = 1 \}$$

is called a unit sphere, the set

$$B[\underline{0}; 1] \equiv \{ x \in X : \| x \| \leq 1 \}$$

is called the closed unit ball and the set

$$B(\underline{0};1) \equiv \{ x \in X : \| x \| < 1 \}$$

is called the open unit ball.

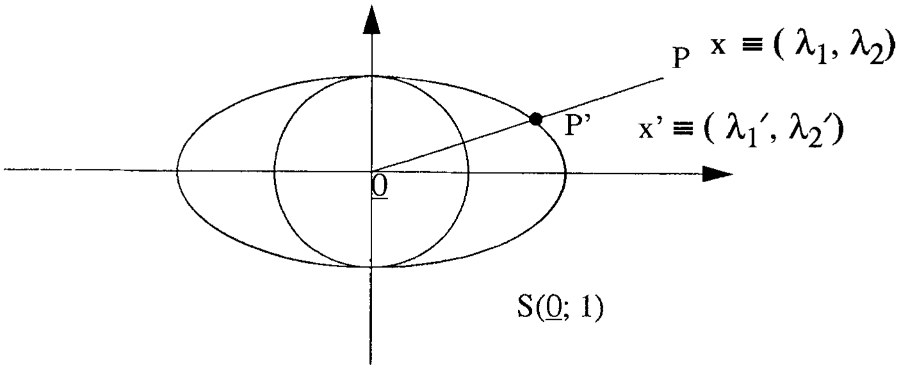
Theorem:

In a normed linear space $(X, \|\bullet\|)$, given $x_0 \in X$ and $r > 0$,

$S(x_0; r) = x_0 + rS(0; 1)$;

That is, every sphere of the space is a translate of a strictly positive multiple of the unit sphere. With Coordinate space examples of normed linear spaces we can assume a Euclidean background and gain some insight into the role of the unit sphere in the measurement of distance.

Consider $(\Re^n, \|\bullet\|)$ with unit sphere $S(\underline{0}; 1)$. For any $x \equiv (\lambda_1, \lambda_2, \dots, \lambda_n)$ we will determine $\| x \|$ in terms of the Euclidean norm $\|\bullet\|_2$ on $\Re^n$:


P $x \equiv (\lambda_1, \lambda_2)$
P' $x' \equiv (\lambda_1', \lambda_2')$
$\underline{0}$
$S(\underline{0}; 1)$

The measurement of distance in $(\Re^2, \|\bullet\|)$ described by measurement of distance in $(\Re^2, \|\bullet\|_2)$.

Consider a ray drawn from 0 through point P with co-ordinates x $\equiv (\lambda_1, \lambda_2, \ldots, \lambda_n)$. Suppose this ray OP cuts S(0; 1) at the point P' with co-ordinates x' $\equiv (\lambda_1', \lambda_2', \ldots, \lambda_n')$. Now in the linear space $\mathfrak{R}^n$ there exists an $\alpha > 0$ such that x = αx'. We have from norm property (iii) that

$$\frac{\|x\|}{\|x'\|} = \alpha = \frac{\|x\|_2}{\|x'\|_2}.$$

But $\| x' \| = 1$ so $\| x \| = \| x \|_2 / \| x' \|_2$;

that is, $\| x \| = | OP | / | OP' |$, the ratio of two Euclidean line segments.

In Euclidean space the complete symmetry of the unit sphere tells us that measurement of distance is invariant under rotation. This is not so with other normal linear spaces where the measurement of distance is dependent on the direction in which the distance is measured.

Orthogonalization Theorem:

Let $x_1, x_2 \ldots$ be a finite or infinite sequence of elements in a Euclidean space V, and let L $(x_1, x_2, \ldots x_n)$ denote the subspace spanned by the first n of these elements. Then there is a corresponding sequence of elements $y_1, y_2, \ldots y_n$ in V which has the following properties for each integer k.

(a) The element y_k is orthogonal to every element in the subspace L $(y_1, y_2, \ldots y_{k-1})$.

(b) The subspace spanned by $y_1,\ldots,y_k$ is the same as that spanned by $x_1,\ldots,x_k$.

$L(y_1,\ldots,y_k)=L(x_1,\ldots,x_k)$

(c) The sequence $y_1,\ldots y_k$ is unique, except for scalar factors. That is if $y_1,y_2,y_3,\ldots$ is another sequence of elements in V satisfying properties (a) and (b) for all k, there is a scalar c_k such that $y_k=c_ky_k$.

Reduction of a real quadratic form to a diagonal form:

A real symmetric matrix A is Hermitian. Therefore it is similar to the diagonal matrix

$$\Lambda = diag\,(\lambda_1, \lambda_2, \ldots \lambda_n)$$

of its eigenvalues. Moreover, we have $\Lambda = C^tAC$ where C is an orthogonal matrix. Now we show that C can be used to convert the quadratic form XAX^t to a diagonal form.

Theorem: Let XAX^t be the quadratic form associated with a real symmetric matrix A, then let C be an orthogonal matrix that converts A to a diagonal matrix $\Lambda = C^tAC$. Then we have

$$XAX^t = Y\Lambda Y^t = \sum_{i=1}^{n} \lambda_i y_i^2$$

where $Y = [\, y_1, \ldots y_n]$ is the row matrix $Y=XC$, and $\lambda_1 \ldots \lambda_n$ are

the eigenvalues of A.

Since C is orthogonal we have $C\text{-}1 = C^t$.

Therefore the equation $Y=XC$ implies $X=YC^t$, and we obtain

$$\begin{aligned} XAX^t &= (YC^t)A(YC^t)^t \\ &= (YC^t)A(CY^t) \\ &= Y(C^tAC)Y^t \\ &= Y\Lambda Y^t \end{aligned}$$

Note: The above theorem is described by saying that the linear transformation $Y=XC$ reduces the quadratic form XAX^t to a diagonal form $Y\Lambda Y^t$.

Eigenvalues of a symmetric transformation obtained as values of its quadratic form.

Let T:V->V be a symmetric transformation on a real Euclidean space V, and let Q(x) = (T(x)). Then the eigenvalues of T (if any exist) are to be found among the values that Q takes on the unit sphere in V. Let $V = V_2(R)$ with the usual basis (i,j) and the usual dot product as inner product. Let T be the symmetric transformation with matrix A, then the quadratic form of T is given by

$$Q(x) = \sum_{i}\sum_{j} a_{ij} x_i x_j$$

The smallest and largest eigen values (if they exist) are always the minimum and maximum values which Q takes on the unit sphere.

APPENDIX C

Ordered Initialization:

A quick partitioning algorithm is outlined below and is applied to obtain an initial partition. We have used pseudo-code with the key words of the language we use being underlined. The key words used are either relational operators or control statements like if, while etc with a corresponding end marker such as end_while, end_if to mark the scope of the control. The key words referred to here are slight variations of the control and looping constructs that can be readily found in any common programming language, such as FORTRAN or PASCAL.

All Variable names contain a mix of lower and uppercase or entirely lowercase letters with an optional underscore, but they don't contain all uppercase letters. There are several array variables and their purpose is described below:

Point: Holds the coordinate information of the point.

Rec_Classified: Will tell us if a point has been classified.

The functions used in the pseudo code are described below:

1. INCREMENT(<variable_name>):

Increments the value of the <variable_name> by one.

2. DISTANCE(<var1>, <var2>):

Computes the Euclidean distance between <var1> and <var2>.

3. LABEL(<var1>, <var2>)

Labels <var2> as belonging to <var1>

EQUAL_IN_ALL (<var>)

Distributes the membership of the point given by <var> evenly in all the clusters.

This must be done whenever a point does not meet any threshold requirements.

Let Nc denote the number of clusters.

Let Num denote the total number of points.

Note: Calculate the average inter-point distance and multiply it by a suitable probability value such as 0.7 in order to employ it as a preliminary threshold for partitioning purposes. For practical purposes, if large data sets are involved, only a subset (a random sampling of the data) need be considered in evaluating the threshold.

Two hash symbols at the beginning of the line indicate that the rest of the line will contain comments.

Quick Partitioning Algorithm

```
##   It is assumed that the average interpoint distance is computed.
##   Threshold holds the value of the threshold.
BEGIN_MAIN_PROGRAM

Threshold IS_ASSIGNED 0.7 * average interpoint distance.
##    Initialize the Rec_Classified array
WHILE Points_Counter IS_LESS_THAN Num BEGIN
    This_Point IS_ASSIGNED points_counter
    Rec_Classified[This_Point] IS_ASSIGNED FALSE
    INCREMENT(points_counter)
END_WHILE
```

```
## Initialize the cluster_counter
## Use the first point as the first prototype by assigning it to Next_Seed
cluster_counter IS_ASSIGNED 1.
Next_Seed IS_ASSIGNED Point[1]

  WHILE cluster_counter IS_LESS_THAN Nc BEGIN
  This_Cluster IS_ASSIGNED cluster_counter
  points counter IS_ASSIGNED 1.
  This_Seed IS_ASSIGNED Next_Seed
  Next_Seed_Flag IS_ASSIGNED FALSE

      WHILE points_counter IS_LESS_THAN Num BEGIN
      This_Point IS_ASSIGNED points_counter

        IF REC_CLASSIFIED[This_Point] IS_EQUAL_TO FALSE

          IF DISTANCE(This_Point, This_Cluster) IS_LESS_THAN Threshold
            LABEL(This_Cluster, This_Point)
            Rec_Classified[This_Point] IS_ASSIGNED TRUE
          ELSE
            IF Next_Seed_Flag IS_EQUAL_TO FALSE
            Next_Seed IS_ASSIGNED POINT[This_Point]
            Next_Seed_Flag IS_ASSIGNED TRUE
            END_IF
          END_IF
```

```
                END_IF


            INCREMENT(points_counter)
            END_WHILE


    INCREMENT(cluster_counter)
    END_WHILE


point_counter IS_ASSIGNED 1


                WHILE points_counter IS_LESS_THAN Num
                    IF REC_CLASSIFIED[This_Point] IS_EQUAL_TO FALSE
                    EQUAL_IN_ALL(This_Point)
                    INCREMENT(points_counter)
                    END_IF
                END_WHILE



END_MAIN_PROGRAM
```

Appendix D

The Proof for the AFCS Norm Theorem

The Adaptive Norm Theorem:

Let $(\mathbf{U},\mathbf{v},\mathbf{r}) \subset \mathbf{M}_{\mathbf{fc}} \times \Re^{cp} \times \Re^{c}$ be fixed. For m > 1 and for each i, det(A_i) $= \rho_i$ fixed, the A* is a local minimum of the functional only if

where

$$A_i = [\rho_i det(S_{sfi})]^{(\frac{1}{\rho})} (S_{sfi})^{-1} \qquad 1 \le i \le c \qquad (1)$$

$$S_{sfi} = \sum_{k=1}^{n} (u_{ik})^m \frac{D_{ik}}{d_{ik}} (x_k - v_i)(x_k - v_i)^T \qquad (2)$$

The above theorem provides the necessary conditions for minimization with respect to Ai's. The proof for the above theorem can be obtained through the Lagrange multiplier technique and is given as follows:

The Lagrangian is formed as,

$$F(\lambda, A) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m (D_{ik})^2 - \sum_{i=1}^{c} \lambda_i [\det(A_i) - \rho_i] \qquad (3)$$

To obtain (1) and (2), the gradients of F with respect to λ_i's and Ai's are set to zero. At the zeros (λ*, A*), it is necessary that

$$\nabla_{\lambda} F(\lambda^*, A^*) = \{-([\det(A_1^*) - \rho_i], \ldots, -[\det(A_c^*) - \rho_c])\} = 0 \in \Re^c (4)$$

and for the gradients woith respect to each Aj,

$$\nabla_{\lambda} F(\lambda^*, A^*) = \sum_{k=1}^{n} (u_{ik})^m \frac{\{[(x_k - v_j)^T A_j^* (x_k - v_j)]^{1/2} - r_i\}}{\{[(x_k - v_j)^T A_j^* (x_k - v_j)]^{1/2}\}} (x_k - v_j)(x_k - v_j)^T$$

$$-\lambda_j^* \det(A_j^*)(A_j^*)^{-1} = S_{sfi} - \lambda_j^* \det(A_j^*)(A_j^*)^{-1} = 0 \qquad (5)$$

Conditions $\nabla A_j(x^T A_j x) = xx^T$; and $\nabla A_j[\det(A_j)] = \det(A_j) \cdot A_j^{-1}$, are used to obtain(5). Since $\det(A_i) = \rho_i$ for each $i = 1$ to c, and utilizing the definitions of D_{ik} and d_{ik} from equations

$$(d_{ik})^2 = (x_k - v_i)^T I (x_k - v_i) \qquad (6)$$

and

$$(D_{ik})^2 = ([(x_k - v_i)^T A_i (x_k - v_i)]^{1/2} - r_i)^2 \qquad (7)$$

equation (5) can be simplified by using S_{sfi} as,

$$S_{sfi} = \lambda_j^* \rho_j (A_j^*)^{-1} \Rightarrow A_j^* = \lambda_j^* \rho_j (S_{sfi})^{-1}; \Rightarrow \lambda_j^* I = (\rho_i)^{-1} (S_{sfi} A_j^*) \qquad (8)$$

In the above, I is an identity matrix. Taking the determinant pf the last equation in (8), we can find λj and then eliminate it from the middle equation in (8) to obtain Aj explicitly in the form shown in equation (1). This completes the proof.

Q. E. D.

APPENDIX E

The Jacobian

The notation followed to represent the term by term expansion of the jacobian is as follows:

The norm is represented as follows:

$$A_i = \begin{bmatrix} a_{1i} & a_{2i} \\ a_{2i} & a_{3i} \end{bmatrix}$$

x_{1k} represents the x coordinate of the *k*th object x_k

x_{2k} represents the y coordinate of the *k*th object x_k

v_{1i} represents the x coordinate of the *i*th center v_i

v_{2i} represents the y coordinate of the *i*th center v_i

r_i represents the radius of the *i*th prototype

u_{ik} represents the membership of the *k*th point in the *i*th cluster

m represents the fuzzifier

D_{ik} represents the distance measured in the normed space

The Jacobian is represented by the symbol J_{ij} where *i* and *j* correspond to the *i*th row and *j*th column respectively in the Jacobian.

$$D_{ik} = [a_{1i}(x_{1k} - v_{1i})^{2.0} + (2.0)\, a_{2i}(x_{1k} - v_{1i})\ (x_{2k} - v_{2i})\ + a_{3i}(x_{2k} - v_{2i})^{2.0}]$$

$$\Psi_{ik} = \left(1.0 - \frac{r_i}{\sqrt{D_{ik}}}\right)$$

$$\Phi_{ik} = (r_i/2.0)\ (\sqrt[3]{(D_{ik})^2})$$

$$\chi_{ik} = (-2.0)\ (a_{1i}(x_{1k} - v_{1i}) + a_{2i}(x_{2k} - v_{2i}))$$

$$\Upsilon_{ik} = (-2.0)\ (a_{2i}(x_{1k} - v_{1i}) + a_{3i}(x_{2k} - v_{2i}))$$

$$\xi_{ik} = (-1.0)\ (\sqrt{D_{ik}})$$

$$\mu_{ik} = (u_{ik})^m$$

$$J_{11} = (\mu_{ik})\ ((-2.0)\ (a_{1i})\ (\Psi_{ik}) + (\chi_{ik})^{2.0}(\Phi_{ik}))$$

$$J_{12} = (\mu_{ik})\ ((-2.0)\ (a_{2i})\ (\Psi_{ik}) + (\chi_{ik})\ (\Upsilon_{ik})\ (\Phi_{ik}))$$

$$J_{13} = (\mu_{ik})\ (\chi_{ik})\ (\Upsilon_{ik})\ (\xi_{ik})$$

$$J_{21} = (\mu_{ik})\ ((-2.0)\ (a_{2i})\ (\Psi_{ik}) + (\chi_{ik})\ (\Upsilon_{ik})\ (\Phi_{ik}))$$

$$J_{22} = (\mu_{ik})\ ((-2.0)\ (a_{3i})\ (\Psi_{ik}) + (\Upsilon_{ik})^{2.0}(\Phi_{ik}))$$

$$J_{23} = (\mu_{ik})\ (\Upsilon_{ik})\ (\xi_{ik})$$

$$J_{31} = (\mu_{ik})\,(\chi_{ik})\,(\xi_{ik})$$

$$J_{32} = (\mu_{ik})\,(\Upsilon_{ik})\,(\xi_{ik})$$

$$J_{33} = (2.0)\,(\mu_{ik})$$

REFERENCES

1. E. Backer and A. K. Jain, "A clustering performance measure based on fuzzy set decomposition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, no. 1, pp. 66-74, 1981.

2. D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, pp. 111-122, 1981.

3. J.C. Bezdek, "Fuzzy mathematics in pattern classification", Ph.D. dissertation, Cornell Univ., Ithaca, NY, 1973.

4. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum, New York, 1981.

5. J. C. Bezdek, "Cluster Validity with fuzzy sets", *J. Cybernetics*, vol. 3, 58-73, 1974.

6. J. C. Bezdek and R. Hathway, "Accelerating convergence of the fuzzy c - shells clustering algorithm", *Proceedings of International Fuzzy Systems Assoc. Congress*, Brussels, Vol.Math., pp. 12-15, 1991.

7. R. N. Dave, "Fuzzy-shell clustering and applications to circle detection in digital images", *International Journal of General Systems*, vol. 16, pp. 343-355, 1990.

8. R. N. Dave, "Generalized fuzzy c-shells clustering and detection of circular and elliptical boundaries", to appear in *Pattern Recognition.*

9. R.N.Dave and S.K. Bhamidipati, "Application of the fuzzy-shell clustering algorithm to recognize circular shapes in digital images", *Proceedings of the International Fuzzy Systems Association Congress*, Seattle, pp. 238-241, 1989.

10. R. N. Dave and K. J. Patel, "Progressive Fuzzy Clustering Algorithms for Characteristic Shape Recognition", *Proceedings of NAFIPS'90*, pp 121-124, 1990.

11. R. N. Dave and K. J. Patel, “Fuzzy ellipsoidal-shell clustering algorithm and detection of ellipsoidal shapes”, *Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision IX:Algorithms and Techniques*, Boston, pp. 320-333, Nov. 1990.

12. R. N. Dave and K. Bhaswan, “Adaptive c-shells clustering”, *Proceedings of the North American Fuzzy Information Processing Society Workshop*, Columbia, Missouri, pp. 195-199, 1991.

13. R. N. Dave and K. Bhaswan, “Adaptive Fuzzy c-shells clustering and detection of ellipses”, submitted to *IEEE Trans. Neural Networks.*

14. J. E. Dennis Jr and R. B. Schnabel, *Numerical Methods for unconstrained optimization and Nonlinear equations*, Prentice Hall series in Computational Mathematics, 1983.

15. E. Diday and J. C. Simon, “Clustering Analysis”, *Digital Pattern Recognition*, edited by K. S. Fu, Springer Verlag, N.Y. 1976.

16. R. Dubes and A. K. Jain, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, 1988.

17. R. Dubes and A.K.Jain, “Clustering Techniques - the users dilemma”, *Pattern Recognition*, vol. 8, 247-260, 1976.

18. R. Dubes and A.K. Jain, “Validity studies in clustering methodologies,” *Pattern Recognition*, vol. 11, pp. 235-253, 1979.

19. R.O.Duda and P.E.Hart, *Pattern Classification and Scene Analysis*, John Wiley, New York, 1973.

20. B. S. Everitt, *Cluster Analysis*, Heinemann, London, 1974.

21. I. Gath and A. B. Geva, “Unsupervised optimal fuzzy clustering”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11, 773-781, July 1989.

22. E. E. Gustafson and W. C. Kessel, “Fuzzy Clustering with a Fuzzy Covariance Matrix,” in *Proc. IEEE CDC*, San Diego, Calif., pp. 761-766, 1979.

23. J. Illingworth and J. Kittler, “A survey of Hough transforms,” *Computer Vision,*

Graphics and Image Processing, pp 87-116, 1988.

24. M. James, *Classification Algorithms*, John Wiley and Sons, New York, 1985.

25. A. Kandel, *Fuzzy Techniques in Pattern Recognition*, John Wıley, New York, 1982.

26. J. M. Ortega and Rheınboldt, *Iterative Solution of Nonlinear equations in several variables*, Academıc Press, New York, 1970.

27. K. J. Patel, "*Characteristic Shape Recognition with Fuzzy Clustering*", M. S. Thesis, N. J. I. T., Newark, NJ, 1990.

28. J. T. Tou and R. C. Gonsalez, *Pattern Recognition Principles*, Addison-Wesley Publıshıng Co.

29. G. Sebestyen, *Decision Making Processes in Pattern Recognition*, The Macmillan Company, New York, 1962.

30. C. E. Shannon, "A Mathematical Theory of Communication", *Bell Syst. Tech. J.*, vol. XXVII-3, pp. 379-423, 1948.

31. H. Spath, *Cluster Analysis Algorithms*, John Wıley and Sons, New York, 1980.

32. C. W. Therrien, *Decision Estimation and Classification*, John Wıley and Sons, New York, 1989.

33. M. P. Wındham, "Cluster Validıty for fuzzy clustering algorıthms," *Fuzzy sets Syst.*, vol. 5, pp 177-185, 1981.

34. L. A. Zadeh, "Fuzzy Sets", *Inform. and control*, **Vol 8**, pp 338-353.

35. C. T. Zahn, "Graph Theoretical methods for detectıng and describıng Gestalt clusters", *IEEE trans. on computers*, **C 20**, Jan.-June, 1971, pp 68-86.