ABSTRACT

ROBUST FUZZY CLUSTERING IN OBJECT RECOGNITION AND CLASSIFICATION OF RELATIONAL DATA

by
Sumit Sen

Prototype based fuzzy clustering algorithms have unique ability to partition the data while detecting multiple clusters simultaneously. However since real data is often contaminated with noise, the clustering methods need to be made robust to be useful in practice. This dissertation focuses on robust detection of multiple clusters from noisy range images for object recognition. Dave's noise clustering (NC) method has been shown to make prototype-based fuzzy clustering techniques robust. In this work, NC is generalized and the new NC membership is shown to be a product of fuzzy c-means (FCM) membership and robust M-estimator weight (or possibilistic membership). Thus the generalized NC approach is shown to have the partitioning ability of FCM and robustness of M-estimators. Since the NC (or FCM) algorithms are based on fixed-point iteration technique, they suffer from the problem of initializations. To overcome this problem, the sampling based robust LMS algorithm is considered by extending it to fuzzy c-LMS algorithm for detecting multiple clusters. The concept of repeated evidence has been incorporated to increase the speed of the new approach. The main problem with the LMS approach is the need for ordering the distance data. To eliminate this problem, a novel sampling based robust algorithm is proposed following the NC principle, called the NLS method, that directly searches for clusters in the maximum density region of the range data without requiring the specification of number of clusters.

The NC concept is also introduced to several fuzzy methods for robust classification of relational data for pattern recognition. This is also extended to non-Euclidean relational data.

The resulting algorithms are used for object recognition from range images as well as for identification of bottleneck parts while creating desegregated cells of machine/ components in cellular manufacturing and group technology (GT) applications.

ROBUST FUZZYCLUSTERING FOR OBJECT RECOGNITION AND CLASSIFICATION OF RELATIONAL DATA

by
Sumit Sen

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Department of Mechanical and Industrial Engineering

May 1998

APPROVAL PAGE

ROBUST FUZZY CLUSTERING IN OBJECT RECOGNITION AND CLASSIFICATION OF RELATIONAL DATA

Sumit Sen

Dr. Rajesh N Dave, Dissertation Advisor — Date
Associate Professor of Mechanical Engineering, NJIT

Dr. Sunil Dhar, Committee Member — Date
Associate Professor of Mathematics Department, NJIT

Dr. Rong-Yaw Chen, Committee Member — Date
Professor of Mechanical Engineering
and Associate Chairperson for Graduate Studies, NJIT

Dr. Zhiming Ji, Committee Member — Date
Associate Professor of Mechanical Engineering, NJIT

Dr. Ming C. Leu, Committee Member — Date
Professor of Mechanical Engineering, NJIT
Director, Manufacturing, Machines & Equipment Program, NSF

BIOGRAPHICAL SKETCH

Author: Sumit Sen

Degree: Doctor of Philosophy in Mechanical Engineering

Date: May 1998

Undergraduate and Graduate Education:

- Doctor of Philosophy in Mechanical Engineering,
  New Jersey Institute of Technology,
  Newark, New Jersey, 1998

- Master of Science in Mechanical Engineering,
  New Jersey Institute of Technology,
  Newark, New Jersey, 1992

- Bachelor of Mechanical Engineering,
  Jadavpur University,
  Calcutta, India, 1985

Major: Mechanical Engineering

Presentations and Publications:

- R. N. Davé and S. Sen, "On generalizing the noise clustering algorithm," in *Proceedings Seventh International Fuzzy Systems Association World Congress: IFSA '97*, Prague, Czech Republic, pp. 205-210, June 1997.

- R. N. Davé and S. Sen, "Robust Fuzzy Relational Clustering," submitted to *IEEE Transaction on Fuzzy Systems*.

- S. Sen and R. N. Davé, "Clustering of relational data containing noise and outliers," accepted, *IEEE Conference on Fuzzy Systems*, Anchorage, Alaska, May 1998.

- R. N. Davé and S. Sen, "Noise Clustering Algorithm Revisited," in *Proceedings of NAFIPS 97*, Syracuse, New York, pp. 199-204, September 1997.

This dissertation is dedicated to my
beloved family.

ACKNOWLEDGMENT

I wish to express my sincere gratitude to my advisor, Dr. Rajesh N. Dave, for his expertise, guidance, and support throughout this research.

Special thanks to Drs. Rong Chen, Sunil Dhar, Zhiming Ji and Ming Leu for serving as committee members.

I acknowledge the contributions of my colleagues Dr.Tianxiong Fu and Girish Phansalkar. I also acknowledge Dr. Richard J. Hathaway of Georgia South University and Dr. Peter J. Rousseeuw of Vrije Universiteit Brussel for providing important information.

Finally, I gratefully acknowledge the financial support received from Department of Mechanical Engineering at New Jersey Institute of Technology during the entire period of study.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

Table **Page**

LIST OF TABLES
(Continued)

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Purpose of Research

Clustering techniques are an important part of many engineering and scientific applications. They have been extensively used in pattern recognition and computer vision tasks. Research in classification has laid significant emphasis on the development of variety of fast and efficient algorithms that can recognize clusters of various geometric shapes (e.g. lines, planes, circles, ellipses etc.). Key aspect of these clustering methods is their potential to obtain a neat and compact mathematical shape description of a geometric prototype. The input data to such algorithms is the spatial information of geometric objects, described by two-dimensional intensity or three dimensional range image. Range data are often produced in the form an array of numbers, referred to as a range image. The numbers quantify the distances from the origin of a global coordinate frame to object surface within the field of view along rays emanating points on a regularly spaced grid. However, in a real application, the data is often contaminated with noise and outliers. Therefore clustering algorithms need to be robust to useful in practice. One emphasis of this research is to formulate new methods for robust recognition of geometric models from noisy two dimensional intensity images as well as three dimensional range images. Second, these robust algorithms have also been validated in the relational domain where the data is represented by mutual similarity or dissimilarity of objects.

The results of this research are applicable to areas like computer aided graphic design (CAGD), rapid prototyping, computer aided inspection (CAI) through coordinate measuring machine (CMM) for quality control in manufacturing industry or reverse engineering. The applications of robust classification in relational data space is often required in areas like management and social science. The application of these robust algorithms in group technology and specifically to solution of desegregated cells and identification of bottleneck parts in cellular manufacturing has been discussed in detail in this research.

1.2 Background and Motivation

Identifying the features that promote good representation is the key to the progress in computer vision [28, 44]. Although the choice of representation obviously depends on the context, task and the form of image data. There is a traditional dichotomy between object- and viewer-centered representations in computer vision [28, 44]. While object-centered representation relies heavily on an explicit three-dimensional description of the object of interest in some world coordinate frame; viewer-centered schemes generally depend on possible abstraction of collection of images of objects as the implicit description of its shape[44, 79].

Since recent approaches may store photometric, color, or texture information from image data, the whole body of viewer-based techniques are referred as "appearance-based" methods [79]. These techniques impose a structure on the data by first extracting features into aspects (or "view class") or by explicitly identifying invariants of these features. These approaches differ in the type of features used and the criteria utilized for grouping or clustering the features into invariant sets [79]. A relatively new approach getting lot of attention is the physics-based approach. This approach typically models shape as a mechanical system submitted to forces that reflect material properties as well as smoothness and image constraints. However the difficulty of these models due to involvement of complex differential equations makes them hard to use in abstract visual recognition task [79].

Constructing an object description from image data requires solving of two mutually complementary problems: first, the object of interest must be isolated from the background (figure / ground discrimination). Second, reliable features must be found and clustered together into significant structure (based on geometric connectivity) [28]. Due to unavailability of explicit three-dimensional information for segmentation in top-down manner; appearance-based techniques rely substantially on bottom-up feature grouping and figure / ground discrimination. Bottom-up segmentation is intrinsically very difficult; which limits the general applicability of appearance-based methods [79].

Most object-centered approaches rely on one type or another of shape primitives. Primitives as diverse as parametric surfaces, volume patches, generalized cylinders, point or curve sets with various invariant properties that have been used to support recognition [14, 15, 17, 20, 52, 53, 54, 55, 57, 58]. These approaches can be classified according to the degree of generality of the underlying primitives and the method of detection. Primitive shapes are explicitly defined by numerical parameters. As prototypes with more parameters are used, it may take fewer of them to describe the complicated object. It may take hundreds of triangles or quadric patches to capture a shape, which can very well be approximated by a handful of super quadrics or algebraic surfaces. More generality can be achieved by a representation that applies to a wide variety of shapes while still capturing the intended object structure. Considering this, the application of quadric surfaces is more sensible [3, 4, 44, 79].

The primary emphasis of this research is on the primitive based robust object detection through fuzzy clustering. Selecting the fuzzy clustering method has certain merits. The range image of an object is a large array of data points at specific temporal location and the algorithm attempts to agglomerate them together to fit the clusters representing the shape of the prototype [3, 71]. Fuzzy clustering is an ideal tool in this regard [4, 8]. Fuzzy partitioning the data into few components with each containing one cluster, simultaneously extracting the reliable features and combining them together into a significant structure for concurrent detection of multiple clusters are key features of the fuzzy clustering technique [21, 22, 23, 24, 25, 26, 27, 31]. The segmentation capability of fuzzy clustering is a built-in feature, hence only the solution technique to detect the actual shape needs to be inter-linked with prototype representation. However the number of clusters needs to be specified a priori. Prototypes include multiple shapes such as cylinders, spheres, cones and planer surfaces, from noisy intensity and range image data, typically found in mechanical and manufacturing engineering environment. Recent robustification of fuzzy clustering [18, 26, 55] has enabled the method to identify the good clusters from background noise. By robustness we mean that the performance of a technique should not be affected significantly due to small deviations from an assumed model and it should not deteriorate drastically due to noise and outliers [37, 38, 46, 47, 65, 66]. Fuzzy clustering technique is a prototype based fixed point iteration scheme, which requires initialization of the prototype at the start. This initialization issue often complicates the robustness aspect of the algorithm.

1.3 Fuzzy Clustering Techniques

The Fuzzy c-means (FCM) algorithm [4] is one of the earliest fuzzy clustering methods proposed that is generally used to detect blob type shapes. They are based on objective function optimization technique and generate a fuzzy partition of the data. It is the most widely used standard algorithm that might be changed to different variations by using different norm inducing matrices for distance measurement in it. Bezdek et al [4, 6, 8, 13] generalized FCM by allowing the prototype to be linear manifolds of arbitrary and different dimensions. The Fuzzy c-Shells (FCS) algorithm is a novel generalization of FCM that uses hyper spherical shells as cluster prototypes. It has proved to be a great success as a method for detecting and representing circular and elliptical sub-structure in two dimensional data-sets [22, 31]. Based on the "shell" concept a series of algorithms have been developed, Fuzzy Ellipsoidal-Shell Clustering algorithm [17, 52], Adaptive Fuzzy C-Shell Clustering algorithm [20], Fuzzy C Quadric Shells Clustering algorithm [53, 55, 56] etc.

Fuzzy clustering techniques partition data point based on membership values between 0 and 1, and this value indicates the degree to which a point belongs to different cluster. The clustering methods with hard memberships partition a point with only one of two values, either 0 or 1. In other words, a point can only be classified to one cluster. Certainly, the hard clustering method can be taken as a special case of fuzzy method. Following the notations and definitions of Bezdek [6] the basic Fuzzy c-Means algorithm can be described as follows:

The FCM minimizes an objective functional J_m: $M_{fc}\times R^{cp} \rightarrow R^+$

$$J_m(U,v) = \sum_{i=1}^{c}\sum_{k=1}^{n}\left(u_{ik}\right)^m\left(d_{ik}\right)^2 \tag{1.1}$$

where

M_{fc} is the fuzzy partition space

R^{cp} are the c p-tuples of the real numbers.

$U \in M_{fc}$ is a fuzzy c-partition of data set $\mathbf{X}$

$(v=(v_1,v_2,..v_c))$ is the cluster center vectors

d_{ik} is given by

$$(d_{ik})^2 = \|x_k\text{-}v_i\|^2 \tag{1.2}$$

and $\|\bullet\|$ is any inner product induced norm on R^p; and $m \in [1,\infty)$ is the weighting exponent or the fuzzifier.

In the above equations, index i denotes one each of c number of clusters, p the dimension of R space from which the data is derived, and index k one each of n number of data-points. It can be realized that J_m is a squared error clustering functional, and solutions that minimize J_m are least-squared error stationary points of J_m. The basic procedure to find the solution is Picard iteration, consisting of following steps.

FCM Algorithm

Fix c, $2\leq c<n$; choose any inner product norm; fix m, $1\leq m<\infty$; initialize the membership matrix $U^{(0)}$.

Calculate c fuzzy cluster centers v_i as

$$v_i = \sum_{k=1}^{n}\left(u_{ik}\right)^m x_k \Bigg/ \sum_{k=1}^{n}\left(u_{ik}\right)^m \tag{1.3}$$

Update membership function $U^{(i)}$ to $U^{(i+1)}$ as

$$u_{ik} = 1 \Big/ \left[\sum_{j=1}^{c} \left(d_{ik} / d_{jk} \right)^{2/(m-1)} \right] \tag{1.4}$$

Compare the change in the membership values using a convenient norm; stop if

$$| U^{(i+1)} - U^{(i)} | < \varepsilon \tag{1.5}$$

Else $i = i+1$ and return to 2.

This algorithm tends to find clusters that are spherical in shape. Although it may not be directly used for plane or quadric surface detection in an image space, it may be used in a range image segmentation [31]. Although there are numerous new development of FCM, the family of fuzzy clustering methods follow the same fixed point iteration scheme.

1.4 Fuzzy Classification of Relational Data

While clustering of object data using fuzzy technique has been a very active field of research, clustering of relational data has received much less attention. This may in part due to the fact that most engineers and mathematicians usually deal with object data, and rarely encounter purely relational data. However in fields like management and social sciences, relational data are frequently encountered. The relational data comes from measure of similarity or dissimilarity between objects, and in some cases actually based on object data [7, 34, 40, 41, 42, 49]. For example, in problem of building of a supply warehouse, one often considers the distance from various plants as a basis for generating relational data [73]. On the other hand, when the data is purely relational, and does not easily fit into any metric axioms, one cannot efficiently apply any object based clustering method, and must employ a relational clustering technique. For n objects, the relational data is usually a n x n matrix (if the relational measure R_{ij} = R_{ji} then only a lower triangular portion of n x n matrix is required. In such cases, there is no explicit knowledge of the "location" of objects in the real space, but such information may be implicit in relational matrix, and could be made explicit through use of techniques based on multidimensional scaling, albeit at a significant computational cost. The concept of noise or outlier is hard to visualize in relational data space. Currently there is no robust fuzzy classification algorithm available to handle the noise and outliers in relational data. This problem has been explored herein details.

1.5 Robust Classification in Group Technology

A problem from group technology and cellular manufacturing has been identified to demonstrate the application of robust clustering method. In group technology or cellular manufacturing, one needs to assign n parts and p machines into c fixed cells in order to form disconnected and self-sufficient cells. Typically it is a classification problem. Since it requires simultaneous detection of multiple cells, fuzzy clustering is an ideal candidate for solution of this problem. Often, some exceptional parts require processing by machines from more than one cell and thereby create bottleneck in production planning and necessitates inter-cell material flow. Industrial engineers often recommend identifying these parts and handling them as special cases, like subcontracting [59]. Here in this research it is shown how NC method can automatically identify these parts as well as correctly classify the rest of the parts.

1.6 The Outline of Thesis

The thesis has seven chapters, primarily dealing with the development of robust clustering algorithms for detection of various geometric cluster sub-structures. Secondly, robust classification of relational data has been explored along with its application to identify the noise and outliers in group technology problem.

Chapter 2 describes how Dave's noise clustering algorithm has been generalized so that each point can take different value of δ. It is shown that NC membership is a product of two terms, original FCM membership, and the generalized possibilistic membership.

Chapter 3 concentrates on the robust least median square (LMS) algorithm and extends it to fuzzy c-LMS method for simultaneous detection of *c* clusters. The number of minimum random sampling necessary has been predicted probabilistically. Further the concept of 'repeated evidence in sampling' like RHT has been incorporated along with fuzzy c-LMS algorithm for faster and more precise estimates.

Chapter 4 introduces the new concept of sampling based noise clustering algorithm. The new robust algorithm, noise least square (NLS), has been formulated by combining the good features of other robust fitting and sampling based algorithms like NC, LMS and RHT. All results are illustrated with examples.

In chapter 5, considers the application of robust fuzzy clustering to relational data. Several fuzzy models of relational data has been surveyed and extended with the Dave's noise concept to handle noise and outliers in data. A new algorithm has been formulated to handle non-euclidean relational data.

Chapter 6 illustrates how we can apply the fuzzy model and noise clustering concept in group technology problem of machine-component cell formation and identification of the exceptional parts and bottle-neck machines.

Chapter 7 describes the overall summary of current dissertation and presents the direction of future research for robust clustering and possibility of new applications.

Appendix A describes the 'hybrd' subroutine in the Minpack library of non-linear functions.

Appendix B shows algebraic equivalence of objective function of FANNY and FCM algorithm.

Appendix C shows a few intermediate steps for derivation of memberships for non-Euclidean relational data.

CHAPTER 2

GENERALIZED NOISE CLUSTERING

2.1 Introduction

Although the minimization of least squared error has been a popular technique to fit models to data for many decades, it is well known that this approach is not very robust[37, 38, 46, 47, 66]. Since the classical fuzzy c-means (FCM) clustering algorithm [4, 6] and many of its derivatives are based on the minimization of least squared error, they are also not very robust against noisy data. In particular, the presence of outliers in the data may drastically affect their performance. The noise clustering (NC) algorithm has been proposed to overcome this major deficiency of the FCM algorithms [18, 25]. It has been shown to be very successful in detecting a variety of cluster shapes in noisy[18, 21, 23, 24, 25]. In this chapter, the NC algorithm is revisited. Several issues regarding the NC algorithm are considered. First, it is argued that the original NC algorithm [18] is made restrictive due to the basic definition of the distance of a feature vector from the noise prototype. It is not necessary to make this distance, called the noise distance, a constant value. Second, it is shown that the individual membership generated by the NC algorithm is comprised of a product of two terms, the first of this is the ordinary FCM membership, while the second one is the robust component of membership. This component is also comparable to the membership in the possibilistic clustering (PC) algorithm [55, 58], as well as the weights in robust M-estimators [23, 24]. Next, it is shown that based on this fact, the NC algorithm can be considered a C-class generalization of the PC algorithm.

2.2 Noise Clustering Technique

Davé [18] considered noise to be a separate class, and represented it by a prototype that has the same distance, d, from all the feature vectors. His definition was:

> Noise prototype: Noise prototype is an universal entity such that it is always at the same distance from every point in the data-set. Let β^* be the noise prototype, and x_j be the point in feature space. Then the noise prototype is such that the distance $d(x_j, \beta_i)$ which is the distance of point x_j from β^*, is a constant value δ.

This definition does not specify what the distance is, but it states that all the points have equal apriori probability of belonging to the noise cluster. Although this makes sense, it does not allow different clusters to view the outliers in different ways. This makes the NC algorithm somewhat restricted. The membership u_{*j} of a data point x_j in the noise cluster is defined as,

$$u_{*j} = 1 - \sum_{i=1}^{c} u_{ij} \tag{2.1}$$

Here C is the number of clusters and u_{ij} denotes the grade of membership (belonging) of point x_j in the i th fuzzy subset of X. Since (2.1) is used to define the membership u_{*j} in the noise class, the usual membership constraint of FCM algorithms is not required. Thus, the membership constraint for the good clusters is effectively relaxed to

$$\sum_{i=1}^{C} u_{ij} < 1. \tag{2.2}$$

This allows noise points to have arbitrarily small membership values in good clusters. The objective function is given as

$$J(B,U;X)=\sum_{i=1}^{C}\sum_{j=1}^{N}\left(u_{ij}\right)^{m}d^{2}\left(x_{j},\beta_{i}\right)+\sum_{j=1}^{N}\delta_{ij}^{2}\left(1-\sum_{i=1}^{C}u_{ij}\right)^{m} \qquad (2.3)$$

In (2.3), $d^2(x_j,\beta_i)$ is the distance from a feature point x_j to the prototype β_i.

The above functional can be optimized with respect to the prototypes and the memberships in a manner similar to FCM functional. The resulting equations for the prototype parameters are very similar, however, the equation for the memberships is different, and is given as,

$$u_{ij}=\frac{\left(\frac{1}{d_{ij}^{2}}\right)^{1/(m-1)}}{\sum_{k=1}^{C}\left(\frac{1}{d_{kj}^{2}}\right)^{1/(m-1)}+\left(\frac{1}{\delta^{2}}\right)^{1/(m-1)}} \qquad (2.4)$$

In the above, d_{ij} is equivalent to $d^2(x_j,\beta_i)$. It can be easily seen that the membership for FCM does not have the second term in the denominator, and thus the NC memberships are different.

2.3 Generalizing the Noise Distance

The first modification one can make to the noise clustering technique is by allowing the noise distance to be different for different feature points. This simply means that each point views the noise prototype in a different perspective. This makes better sense because points belonging to different clusters or different regions in the space may have different proximity to noise cluster prototype. To understand the effect of the noise distance, consider the simplified case of a single cluster (C = 1) and the fuzzifier m = 1, representing the hard partition. Then, all the points having the distance from the prototype less than the noise distance are classified into the good cluster, while all the points having the distance from the prototype more than the noise distance are classified into the noise cluster. Thus the noise distance becomes a switching criteria. For m ≠ 1, the case is fuzzy, and one can visualize the points belonging to the cluster in terms of its fuzzy membership. In such cases too, the noise distance becomes a switching criteria, but the switch is fuzzy. When C ≠ 1, the situation becomes much harder to visualize, but the effect is essentially same.

According to the first modification, the noise distance is defined as δ_j, so that each point j has a different noise distance. This is a rather simple and obvious generalization, but further examination reveals that it eliminates some of the criticism of NC approach as compared to other techniques such as possibilistic clustering. The possibilistic clustering technique has been compared in some detail with the NC method in [24], and the two were shown to be identical for the case of a single cluster (C = 1). Although there are disadvantages of PC technique when C ≠ 1, as will be discussed later [2], when C = 1, the PC method [55] has one advantage that it allows the noise distance (or the possibilistic distance) to be different for different clusters. However, since in this chapter the noise distance is allowed to be different for different points, this original disadvantage of the NC can be eliminated. Before discussing this further, the basics of the PC technique are presented.

The approach, called the Possibilistic C Means (PCM) algorithm [55], was introduced to overcome the relative membership problem of the FCM by specifying the objective function as follows. Here, relative membership means that the membership of a point in one cluster is related to its membership in other clusters.

$$J(B,U;X)=\sum_{i=1}^{C}\ \sum_{j=1}^{N}\ (u_{ij})^{m}+\sum_{i=1}^{C}\eta_i\sum_{j=1}^{N}\left(1-u_{ij}\right)^{m} \tag{2.5}$$

In (2.5), the η_i are suitable positive numbers. The first term in (2.5) requires that the distance from the feature vectors to the prototypes be as low as possible, whereas the second term forces u_{ij} to be as large as possible, so that it avoids the trivial solution. The membership due to the PCM functional can be found by the following,

$$u_{ij}=\frac{1}{1+\left[d^2\left(x_j,\beta_i\right)/\eta_i\right]^{1/(m-1)}} \tag{2.6}$$

It can be easily observed from (2.3) and (2.5) that when C = 1, both NC and PCM are identical for $\delta^2=\eta$. When C ≠ 1, the PCM has a different η_i for each cluster i. In case of the original NC technique, there is only one noise distance, however in its generalized version, the following equation allows for utilizing a different noise class for each cluster through setting up δ_j for each j such that

$$\delta_j^2=\eta_i,\ \text{for } i \text{ such that } \max_{1\le i\le C}\left\{u_{ij}\right\} \tag{2.6a}$$

In the above, η_i can be chosen in the manner similar to PCM clustering, see for example [24]. In fact, they may be computed in a robust way as proposed in [24]. The above modification allows for having one noise class per cluster through such specification of δ_j as the algorithm progresses.

As can be observed from equation (2.6), the PCM membership is independent of other points and/or clusters, and depends only on the parameter η_i. This makes the resulting memberships "possibilistic", as they just show the degree of typicality. As mentioned before, when C = 1, both NC and PCM are identical. When C > 1, however, there are many subtle and not so subtle differences between these two methods. The major difference is that in NC, there is one noise class, while PCM can be interpreted as having C noise classes. Although it is preferable to have one noise class per cluster as in PCM, it is easy to recognize that since the PCM functional in (2.5) is separable in C independent functional, it is equivalent to C separate NC functional, each looking for a single cluster. Moreover, without proper initialization in PCM, it is quite possible to obtain C identical clusters (all finding the same single cluster) as was observed and reported in [24], thus missing out the other C-1 clusters. In this light, the number of clusters C is somewhat irrelevant and arbitrary for the PCM algorithm, and it may be more meaningful to use it for C = 1. Despite the number of arguments and "insights" presented in [58], the fact remains that possibilistic C-means is a misnomer, and it should just be called possibilistic clustering, and be considered a mode seeking algorithm, best suitable for finding a single highest density cluster. However, in that disguise, it is really a special case of NC algorithm. This can be further examined by deriving a new interpretation of the noise membership.

Further insight or new interpretation of noise membership can be gained by recognizing the appearance of the harmonic mean distance in equation (2.4). One can compute harmonic mean distance, η_i , for each feature point x_j over all C clusters as,

$$h_j^2 = \frac{1}{\sum_{i=1}^{C} \frac{1}{\left(d_{ij}\right)^2}} \tag{2.7}$$

This distance, as defined above, is not a true harmonic distance, because it is not divided by the number of clusters C. However, in the rest of the chapter, this modified definition will be used. As defined here, it has a very unique relation to the ordinary Euclidean distance of a point from all the clusters. For instance, when there is only one cluster, this distance is same as the distance of the point from the cluster center, but when there are multiple clusters, it tends to attain a value closer to the smallest of all the distances of the point from cluster prototypes. This property is further addressed in the next section. When the above definition is substituted in (2.4), the following is obtained.

$$u_{ij} = \frac{\left(\frac{1}{d_{ij}^2}\right)^{1/(m-1)}}{\left(\frac{1}{h_j^2}\right)^{1/(m-1)} + \left(\frac{1}{\delta_j^2}\right)^{1/(m-1)}} \tag{2.8}$$

This can be expressed as a product of two terms as follows,

$$u_{ij} = \left(\frac{\left(\frac{1}{d_{ij}^2}\right)^{1/(m-1)}}{\left(\frac{1}{h_j^2}\right)^{1/(m-1)}}\right)\left(\frac{\left(\frac{1}{h_j^2}\right)^{1/(m-1)}}{\left(\frac{1}{h_j^2}\right)^{1/(m-1)} + \left(\frac{1}{\delta_j^2}\right)^{1/(m-1)}}\right) \tag{2.9}$$

In the above, the first term is same as the membership of FCM, while the second term can be shown to be a possibilistic membership through rearrangement as below,

$$u_{ij} = \left({}^{FCM}u_{ij} \right) \left(\frac{1}{1 + \left(\frac{h_j^2}{\delta_i^2} \right)^{1/(m-1)}} \right) \tag{2.10}$$

The first term is denoted to be FCM membership by a pre-superscript FCM. The second is a possibilistic membership with the harmonic mean distance, η_i, as the basis for the distance in evaluating the membership. This equation is comparable to equation (2.6) of membership for PCM. The pre-superscript for the second term is PCM* to denote its similarity with PCM, but the "*" indicates that it utilizes the harmonic mean distance (defined as in this chapter) as its basis. Thus finally, the noise membership can be written as,

$${}^{NC}u_{ij} = {}^{FCM}u_{ij} \; {}^{PCM^*}u_{ij} \tag{2.11}$$

The above shows that the NC is a generalized algorithm for which both the FCM and PCM are its special cases.

The PCM algorithm has been called a mode seeking algorithm [58], and in contrast, the FCM is termed as a partitioning algorithm. Such description is very true, however, it does not make the disadvantages of PCM go away. The major disadvantage is that without good initialization, which ironically is usually provided by the FCM, the PCM may have a difficult time reaching a desired solution. In some cases, it may even find same clusters many times, and miss the other clusters. As discussed elegantly in [58], this could happen for a number of reasons, including improper choice of fuzzifier, or η_i, or more commonly when there is one dominant cluster along with other less prominent clusters. When the latter occurs, it may be a blessing in disguise in some cases. However, for many other clustering applications, such a result is highly undesirable. For most cases, one needs the FCM's power of partitioning, coupled with the PCM's power of mode seeking. When viewed in the light of equation (2.11), the NC has a power to provide both these abilities simultaneously. Use of NC, however, may not allow one to obtain the membership values which represent typicality or possibility, but there are a number of simple ways to overcome that. This will be addressed later in this chapter. Regarding the interpretation of equation (2.11), the only question remains is that of the use of the harmonic mean distance versus the Euclidean distance in PCM membership computation. Next, that issue is examined.

2.4 Harmonic Mean Distance

Harmonic mean distance has many interesting features in FCM type clustering. It is the distance that gives FCM its partitioning capability, and in fact, it may be used to eliminate the membership term altogether from the FCM functional through an alternate formulation [39], such that the FCM functional is modified as follows,

$$J(FCM) = \sum_{j=1}^{N} h_j^2 \tag{2.12}$$

In the above, the harmonic mean distance is defined as in equation (2.7), so that the summation on the right hand side is not divided by C as would be the case if true harmonic distance were used [39]. Implicit in the above equation is the fact that the fuzzifier m is used in computing the memberships, see [39] or [51] for details.

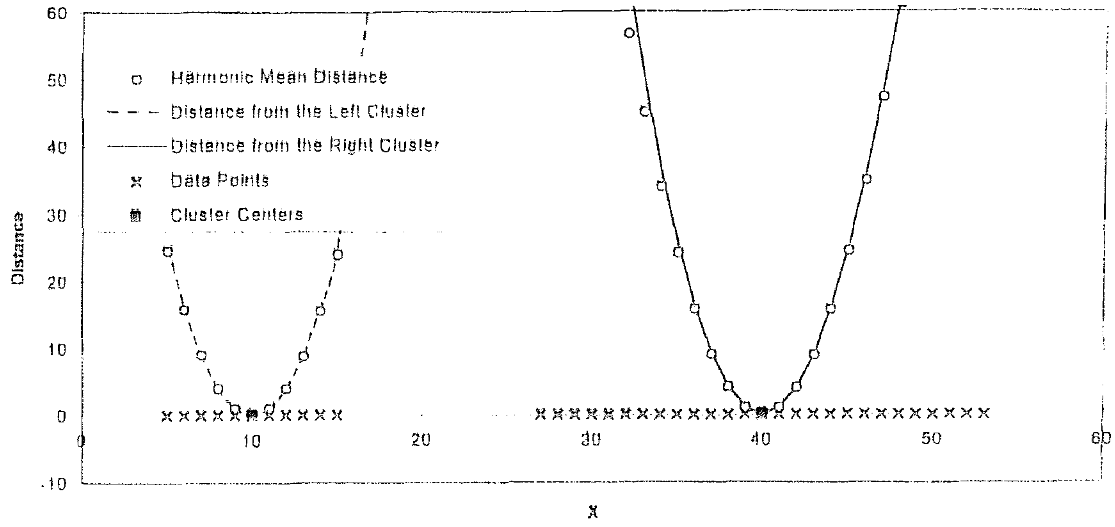

Figure 2.1 Show two clusters and harmonic distance

When (2.12) is compared with a single cluster case (or the ordinary minimization of least squared error fit), one can notice that the harmonic distance takes the role of the Euclidean distance in the ordinary least squared error minimization. This fact was used by Kim etal. [51] to derive a least trimmed squared error clustering algorithm, where the trimming was performed on harmonic mean distance, rather than the Euclidean distance as in a single cluster case. One can gain useful insight into the harmonic mean distance by considering a simple, one dimensional, two class problem. Consider Figure 2.1, where a one dimensional data-set is shown with two clusters. In this figure, all the distances are "squared" distances. It can be seen clearly in this figure that the distances from either left or right clusters, plotted as curves, become very large as one moves away from the data point, while the harmonic distance, shown as open circles, is always closer to the smaller of the two distances. In possibilistic clustering, the outliers as well as the points of the two distances. In possibilistic clustering, the outliers as well as the points belonging to other clusters are treated as noise points for computing the membership of a particular cluster, while in NC, only the real outliers would be treated as noise points when computing the possibilistic component of the membership using the harmonic mean distance. It is interesting to observe that for computing the membership in right cluster, the points belonging to left cluster will not be treated as outliers. However, real outliers (not shown in this figure) will have a high harmonic mean distance, and they will be treated as outliers by both the clusters. It is true that if ones objective is to only allow for the memberships to represent the degree of typicality, such as what is realized by the PCM technique, then the use of harmonic distance as shown here is not proper. It is true that the PCM technique achieves what it planned to achieve, which is eliminating the problem of relative membership so that the sum of memberships across the classes is not one. However, in doing that, it really throws away any proper meaning to the value of C, as that becomes arbitrary. Basically, it turns out that if one wants the typicality as in PCM, then one loses the partitioning capability of FCM, and vice versa. Although it may seem from the previous paragraph that the NC approach cannot have the degree of typicality in its possibilistic component of membership. A clever manipulation of the definition of allows one to mimic the effect of PCM in that component.

Before that is investigated, we consider Figure 2.2, where for those same clusters, the FCM and PCM component of memberships are plotted, where the value of m is chosen as 2, and the δ_j are selected through equation (2.6a), with η_i values of 5 and 9 respectively for the left and right clusters.

Notice in this figure that use of two different η_i to effectively obtain two different levels of values for δ_j gives rise to different width of the PCM membership component around two cluster centers. As seen in the plot, the FCM components behave quite predictably, while the PCM component only takes care of outliers, and not the so-called typicality of individual clusters that the PCM algorithm is concerned with. Having said that, one can further exploit the power of equation (2.11), by noticing that the PCM component can be a function of both j (i.e. the point), and i (i.e. the cluster). So far, because of the straight forward definition used from equations (2.6a) and (2.10), the PCM component is made same for both the clusters for each data point, as evidenced by a single curve in Figure 2.2 One could further generalize the NC technique by specifying the noise distance which is a function of both i and j such that the PCM component is also

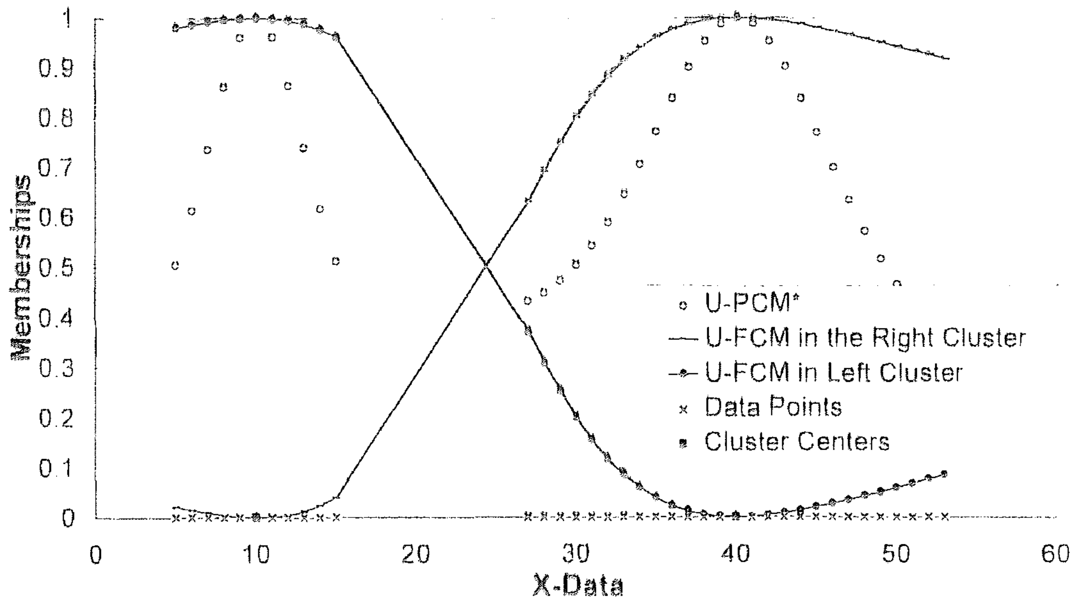

Figure 2.2 Plot show two clusters and memberships

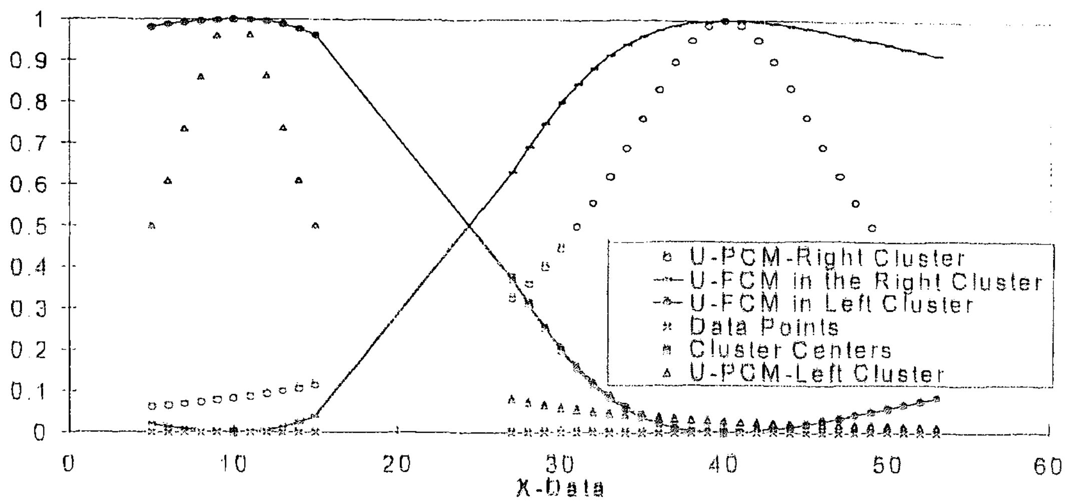

Figure 2.3 Show two clusters and new PCM memberships.

a function of both i and j in term ${}^{PCM*}u_{ij}$. This can be done by defining the noise distance as δ_{ij} for the purpose of evaluating the NC membership. It is noted that equation (2.4) and its subsequent modifications are not valid if δ_{ij} is used as a noise distance in the original functional of equation (2.3). However, for the sake of derivation, the noise distance is considered to be independent of both i and j (and also the cluster-point distances), but for the sake of membership computation it may become a function of both i and j (and also the cluster-point distances). This type of dual treatment is standard practice for example in computing weights in weighted least squared approach. In this respect, one can even specify a value for each δ_{ij} such that

$$\delta_{ij}^2 = \eta_i\left({}^{FCM}u_{ij}\right) \tag{2.13}$$

where ${}^{PCM}u_{ij}$ is exactly same as in equation (2.6), thus the NC algorithm can be made to behave like a generalized combination of FCM and PCM algorithms. Such version would have a combination of partitioning and mode seeking capabilities. Thus it is a true C-class extension of PC approach. This can be illustrated by re-plotting Figure 2.2 as Figure 2.3, to show how ${}^{PCM}u_{ij}$ for two clusters look for all the data points.

Substituting above in (2.10) gives the following for the NC membership,

$${}^{NC}u_{ij} = {}^{FCM}u_{ij}\;{}^{PCM}u_{ij} \tag{2.14}$$

Once again, it is reminded that the above plot, and equation (2.14) are obtained by treating δ_{ij} independent of i and j during optimization. If the noise distance is considered a function of both i and j during optimization, one needs to modify the NC functional in (2.3) to the following.

$$J(B,U;X) = \sum_{i=1}^{C}\sum_{j=1}^{N}\left(u_{ij}\right)^{m} d^{2}\left(x_{j},\beta_{i}\right) + \sum_{i=1}^{C}\sum_{j=1}^{N}\delta_{ij}^{2}\left(1-\sum_{i=1}^{C}u_{ij}\right)^{m} \tag{2.15}$$

Based on the above, the following is obtained for the NC membership.

$$u_{ij} = \frac{\left(\dfrac{\delta_{ij}^{2}}{d_{ij}^{2}}\right)^{1/(m-1)}}{\displaystyle\sum_{k=1}^{C}\left(\dfrac{\delta_{kj}^{2}}{d_{kj}^{2}}\right)^{1/(m-1)} + 1} \tag{2.16}$$

The above equation is very similar to equation (2.4), but one cannot simplify it directly to obtain equation (2.14) in conjunction with (2.13). It shows, however, that the power of typicality in (2.16) comes through scaling of the distances by δ_{ij} and the power of partitioning comes from the first term in the denominator. Thus the generalized NC membership has benefits of FCM as well as PCM.

2.5 Results and Conclusion

The original NC algorithm considered noise to be a separate class, and represented it by a prototype that has the same distance δ, from all the data points. Although this concept has been successful in developing a class of NC algorithms to detect a variety of cluster shapes in noisy data, use of the same constant value of the noise distance δ, for all the feature vectors in the data-set makes it somewhat limited in its scope.

Generalization of the NC technique is achieved through relaxing the original definition of the noise distance. By letting the noise distance be different for each data point, and even for each cluster, a variety of interesting effects can be obtained. It is also shown that the original NC membership is a product of two terms, one which is the FCM component responsible for data partitioning, while the other is the possibilistic component that achieves a mode seeking effect, and imparts robustness. In this light, it is shown that the NC technique is a generalization of the possibilistic clustering technique. The aspect of robustness are not discussed here, but is obtainable from reference [21, 23, 24, 25] where a detailed discussion that shows the similarities between NC technique with robust statistical techniques such as M-estimators [37, 38, 46, 47, 66]. While the discussion in [23, 24] consider only the NC for a single cluster, it is easy to see from equation (2.11) or (2.14) that the NC membership is a product of the FCM term and a robust term that can be equivalent to a robust M-estimator. Comparison with the PC technique shows that the NC can be considered its generalization, and it can allow one to combine the partitioning capability of FCM with the typicality of PCM.

The issue of needing partitioning capability versus mode seeking capability is debatable, and perhaps one can find equally strong arguments for either case. In practice, however, the nature of application and the desired outcome would decide which one to select. One should realize that PC is a combination of C independent functional having no connection to each other, and thus should be viewed as a single cluster seeking algorithm. While that may be an advantage that one does not need to know the exact number of clusters, it becomes susceptible to many other problems related to single cluster seeking algorithms, such as the problem of scale and initialization and/or need for validity measures for removal of clusters. For instance, the PC is very similar to the potential function approach, which suffers from extreme sensitivity to the choice of scale parameters [24]. It is useful to know that the NC approach combines both partitioning and mode seeking traits, and may be a better overall choice. Also, if one wants it to be a purely mode seeking, single cluster finding algorithm, then too it works by letting C = 1, and in that form it inherits all the good as well as bad qualities of the PC algorithm. Moreover, when used in a manner similar to equation (2.13) and (2.14), it becomes an extension of an ordinary robust M-estimator to a C-class M-estimator. In fact, this version can be called a Fuzzy C-Robust-M-estimator clustering algorithm. Through a judicious choice of δ_{ij}, one can produce a variety of M-estimators.

The last issue is regarding the appearance of harmonic mean distance in the PCM component of basic NC membership (see equation (2.10)). This is indeed a fortunate occurrence, because this distance has a great significance in terms of the extension of single cluster robust estimators to multiple clusters. As mentioned before, Kim etal. [51] were perhaps first to exploit this to achieve a rather easy extension of robust Least Trimmed Squared (LTS) technique [66] to the case of multiple clusters, and thus avoided the usual paradox related to the fraction of contaminated data, and the upper limit on breakdown point of robust techniques. This paradox occurs for multiple clusters, because in the traditional sense, all the points outside a cluster, including the points in other clusters are considered to be outliers, and in that sense, any robust method should have a breakdown point of greater than 50%, which is considered a theoretical maximum [66]. In case of NC, the harmonic mean distance allows one to discriminate between true outliers and good points that belong to one of the clusters. In particular, if the scale of each cluster is estimated properly, than the NC will discriminate between outliers and good points even when the clusters differ much in size. It is noted that just like η_i in the original PCM, selection of proper δ_{ij} is crucial and nontrivial for NC, albeit somewhat less sensitive due to the built-in partitioning capability.

CHAPTER 3

FUZZY C-LMS CLUSTERING ALGORITHM

3.1 Introduction

Cluster analysis deals with the problem of partitioning the data-set into a number of subsets. In partitioning an unlabelled data-set, there have been two fundamental strategies are in practice, namely (a) search for the evidence of a good cluster (b) fitting the clusters based on minimization of squared error. The search based clustering techniques, include Hough transform (HT) [45], generalized HT [1], randomized HT or commonly known as (RHT) [77], often converting a difficult global detection problem in image space into a more easily solved local peak detection problem, within limited search, in parameter space to collect evidence of a good cluster. The resulting computational and storage complexity is directly proportional to the accuracy of results and the number of parameters of the cluster to be detected. While fitting the clusters by minimizing the squared errors, *e.g.* the family of fuzzy clustering algorithms [4, 6, 8, 13, 18 etc.], are based on objective function minimization is truly C-class fuzzy generalization of squared error criterion following a fixed point iteration scheme. Algorithms of both types are highly sensitive to noise and outliers.

Robustness against noise and outliers is an indispensable characteristic for any clustering algorithm to be useful in practice [24, 25]. Since noise and outliers are common phenomenon in both real world sensors and low-level image processing algorithms, so robust clustering techniques are strongly mandated for real world applications. Of late, several robust estimators were developed in statistical field being able to handle noise and outlier problem in the context of single cluster with considerable success [37, 38, 46, 47, 64, 65, 66]. According to Huber [46, 47], robustness of an estimator from statistical view point should have following characteristics:- (1) a reasonable good efficiency at the assumed model, (2) small deviations from the assumed model should alter the performance of the algorithm by small amount and the larger deviations should not cause a catastrophe, (3) should reject those outliers beyond a finite limit, (4) the estimate should be resistant against small fluctuation of intermediate residual values, (5) the largest fraction of contamination that can be tolerated by the estimator, commonly known as breakdown point, should be high. The objective is to describe the structure of the cluster best fitting the bulk of the data and identify the deviating data points ("outliers") and highly influential data points ("leverage points"). New clustering algorithms are therefore inspected in the perspective of the robustness. Here we show that the robust LMS method [65, 66] can be upgraded to detect concurrent multiple clusters to produce noise-resistant fuzzy c-LMS algorithm or FCLMS method.

3.2 The Robustness of LMS Algorithm

There has been a variety of robust algorithms available in robust statistics for detecting single cluster in digital image [37, 38, 46, 47, 64, 65, 66]. Robustness of an estimator is enhanced by reducing the influence of the outliers and noise data. The Least Median Square (LMS), Repeated Median (RM), Andrews's, Tukey's and Theil's median based estimators and their extensions are some of the search based robust algorithms with break down point between 0.25 to 0.5 [65, 66]. Of all robust estimators with their relative merits and demerits, the LMS has the highest break-down point value of 0.5. This indicates, if data has up to 50% of outliers and noise, the algorithm can still estimate correctly. As we know, the theoretical maximum breakdown point achievable is 0.5 since if more than half of the data are bad, they may "conspire" to look better than the correct fit. LMS algorithm achieves its robustness by considering the median value of the residual, since the median of any ordered distribution always lies most dense region of the data and the addition of one extreme outlier shifts the "middle" order statistics move in that direction. LMS does not reject any outlier while evaluating the median residual.

The quantitative information about robustness is provided by influence curve IC or influence function IF and derived quantities. It is more precisely called as 'influence curve' since most of the robustness information are derived from its shape. The contamination by one extreme residual has at the most a small finite influence $\sqrt{\pi/2}$ or

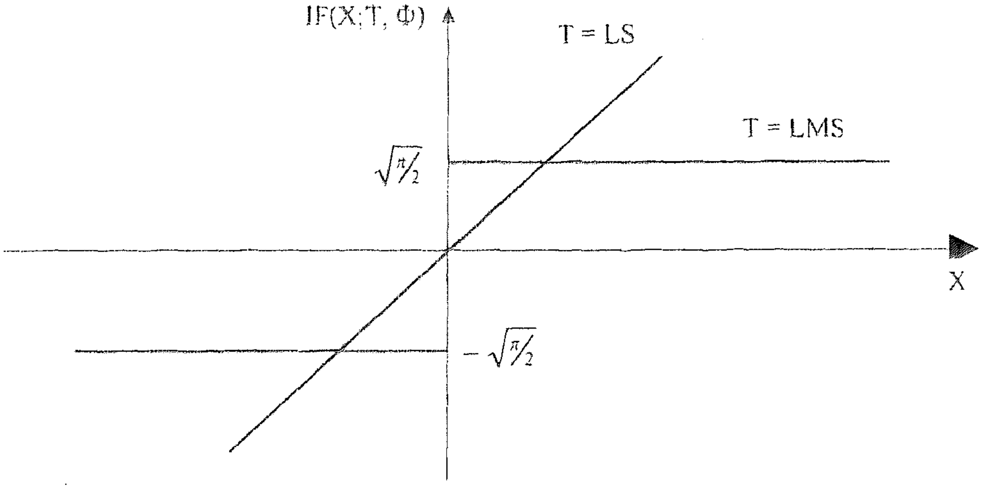

Figure 3.1 Influence Function of Least Square, LS and Least Median Square, LMS

1.253 on the median of residual distribution Figure 3.1 [Figure 1, page 90, 66]. Therefore irrespective of the position of the outlier the worst (approximate) influence which a small amount of contamination of fixed size can have on the value of the estimator is very low. So the median is less sensitive against extreme outliers. However 'Influence Function' has a jump at zero indicates median is sensitive to 'wiggling' near the center of symmetry. So the estimates based on minimization of median of the residuals are robust fit to majority of the data but sometimes not very accurate. The LMS is computationally very expensive since there is no closed form expression for LMS estimator. Theoretically it is a process of exhaustive sampling over the whole data-set.

The objective function for LMS is expressed as

$$Minimize(\underset{i=1,\ldots,n}{med}(y_i - x_i\theta_j)^2) \tag{3.1}$$

The trial estimate for which this value is minimal is the LMS solution of θ_j for one cluster. How many times we need to search over the data space so that the probability of the found bandwidth being correct depends upon the number of parameters (p) of the cluster to be estimated.. If we consider all possible subsamples of size p, there are C_n^p in total. As we can understand C_n^p increases very fast with n and p. In fact the computational complexity is inherent to all known high breakdown point regression or search based robust estimators. Here Rousseeuw [65, 66] suggested to perform a certain number of random selections, such that the probability P that at least one out of m subsamples or picked points is "good" is almost 1 [66]. He expressed this P assuming that n/p is large, as

$$P = 1 - \left(1 - (1 - \varepsilon)^{p+1}\right)^m \tag{3.2}$$

where the data-set contains a fraction of ε of outliers and noise and p is the degree of freedom i.e. the number of parameters of the cluster to be estimated. By requiring that this probability must be near 1, one can determine m for any given value of p and ε has been tabulated by Rousseeuw [66]. Clearly if we try to evaluate a reliable estimate with probability of 0.995 while detecting the line $(p = 2)$ from a data-set that has say 0.5 fraction outliers, we will require at least 11 picking. This is the suggested minimum trials necessary for detecting one cluster. Obviously with the increasing number of parameters p of the cluster to be estimated and the fraction of contamination ε, the number of random sampling m requiring to detect one good cluster with a probability of 0.95 or above rises astronomically. If data contain multiple clusters, conventional LMS neither been able to handle the data-set nor suggests the number of sampling should we consider. It is therefore necessary to upgrade LMS to tackle c-clusters where c is the number of clusters. The number of minimum sampling necessary for that is also discussed [27].

3.3 Fuzzy c-LMS (FCLMS) Clustering Algorithm

Simultaneous detection of c clusters by multiple partitioning of the dataset is one of the key features of fuzzy clustering. However the segmentation into multiple clusters is obtained here by the weighted averaging method where the squared membership is the weight. This contradicts with the LMS approach, where the desired value is obtained from the middle position of an ordered distribution. It may be possible to formulate the objective function of fuzzy C-LMS algorithm, altering the FCM functional by substituting $\ell 1$ distance, where the median appears as the solution of cluster parameters as in [50]. If we follow the fixed point iteration scheme, the objective function needs to be differentiated w.r.t. the cluster parameters. This leads to derivative of a distance function $f(x) = |x|$ $w.r.t.$ x. It is not defined at $x = 0$ since $f(x)$ is not continuous at $x = 0$. So the fixed point iteration scheme is not appropriate here. We therefore apply a global search based technique through fuzzy partition of data to formulate the objective function.

3.3.1 Least Median Square in Fuzzy Clustering

Strong points of fuzzy clustering is its ability to partition the data set into multiple high density regions, as well as detect multiple clusters simultaneously. To get a better understanding of fuzzy clustering let us analyze the fuzzy c-means or FCM algorithm [4, 6]. The mathematical formulation of FCM algorithm is the minimization of the weighted sum of the squared distance between the points and the cluster prototypes are minimized. Following the notations of Bezdek [4, 6], the minimization functional is

$$J(U,v) = \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^{m}(d_{ik})^{2} \tag{3.3}$$

since in our examples line is consider as prototype, here the distances are defined as

$$(d_{ik})^{2} = \frac{\left(y_{k} - m_{i}x_{k} - c_{i}\right)^{2}}{\left(m_{i}^{2} + 1\right)} \tag{3.4}$$

with y_k and x_k being the coordinate of point k (k=1 through n number of points), and m_i, c_i the cluster prototype parameters (i =1 through c). Here m is the exponent, 1<m< α, and u_{ik} is the membership of point k in cluster i. The following restriction apply to the membership u_{ik}

$$\sum_{i=1}^{c} u_{ik} = 1 \tag{3.5}$$

If the memberships u_{ik} are hard, i.e. 0 or 1, then the exponent $m>1$ has no meaning, hence $m=1$ is chosen. The membership value is 1 when an object belongs to a class, and 0 when it does not. The memberships can also be fuzzy, *i.e.* it can also take the value between 0 and 1. If the memberships are fuzzy, the following equation for u_{ik} can be obtained using lagrange multiplier technique [4, 6] is given below:

$$u_{ik} = d_{ik}^{2/(1-m)} \Big/ \sum_{j=1}^{c} d_{jk}^{2/(1-m)} \tag{3.6}$$

Careful observation of the above formulation shows that the constraint imposed by the equation (3.3) causes the bias due the outliers which affects the prototype parameters estimate. Since the sum of the memberships of a point k must be equal to unity, means that the point must be assigned to one of the classes. Thus the outliers must be identified and treated separately from good data as an objective of robust clustering. As we know the fuzzy clustering technique creates a c-partition of the data while minimizing the objective function. The local minima is achieved if individual clusters have been detected with its associated noise.

Hathaway et.al. [39] reformulated this FCM algorithm by substituting equation (3.6) into (3.3) and assume $m=2$, we obtain

$$J(U,v)_{FCM} = \sum_{k=1}^{N} \left(\sum_{i=1}^{c} d_{ik}^{2/(1-m)} \right)^{1-m} = \sum_{k=1}^{N} h_k^2 \tag{3.7}$$

where h_k^2 is the modified harmonic mean of the distances d_{ik}^2, $i = 1,\ldots,c$ of a point k. Since c is a constant, it can be ignored in (3.7). Let us investigate the characteristics of modified harmonic mean of the distances (h_k^2) of a point k. The harmonic mean-distance of a point is defined as

$$h_k^2 = \frac{1}{\sum_{j=1}^{c} \frac{1}{d_{jk}^2}} \tag{3.8}$$

It is observed, the value of h_k^2 is always smaller than the smallest d_{ik}^2 for $i = 1,\ldots,c$. If a data point k is very close to only one cluster and far away from others, then h_k^2 value is marginally less than its distance from the nearest cluster $d_{nearest,k}^2$. Therefore all good data, close to any cluster will have smaller h_k^2 value, may be even zero if on the prototype, compared to the outliers and noise, as those points are usually far away from all the clusters. Consider the example in Figure 3.2, with three clusters and 25 noise points in the inter-cluster space. There are 150 points in data. If we assume, all three

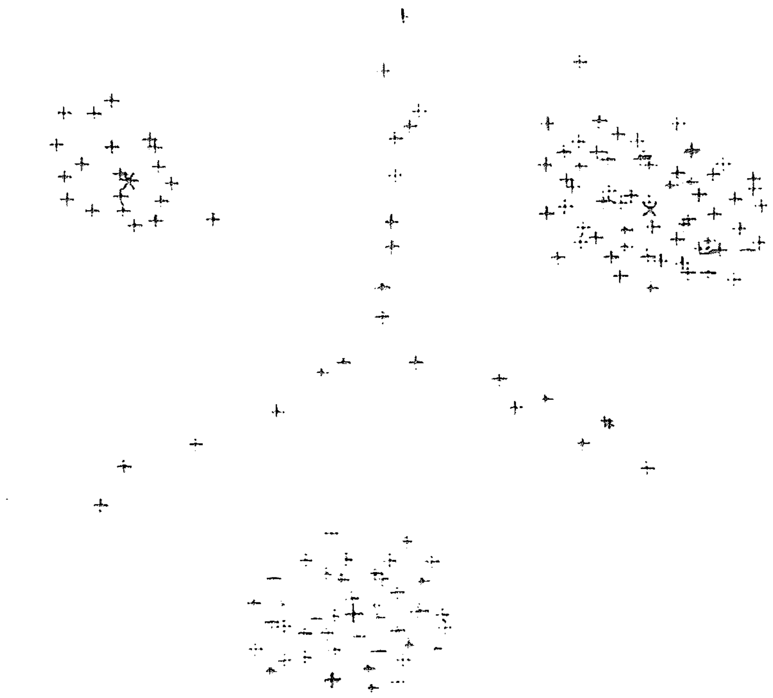

Figure 3.2 Three blob clusters constituted of 125 points and added 25 noise points

clusters has been correctly detected. the fraction of noise and outliers ε, in the ordered h_k^2 distribution will appear at the far end, as evident in the Figure 3.3. Even the h_k^2 of outliers and noise are relatively much less than its distances to the nearest cluster. Figure 3.4 and figure 3.5 shows the 3-dimensional plot of h_k^2 and d_{ik}^2 along with h_k^2. Due to the scale of the data. the h_k^2 profile has flattened out in figure 3.5. This indicates that h_k^2 of all points gives a measure of proximity of the points to its nearest cluster.

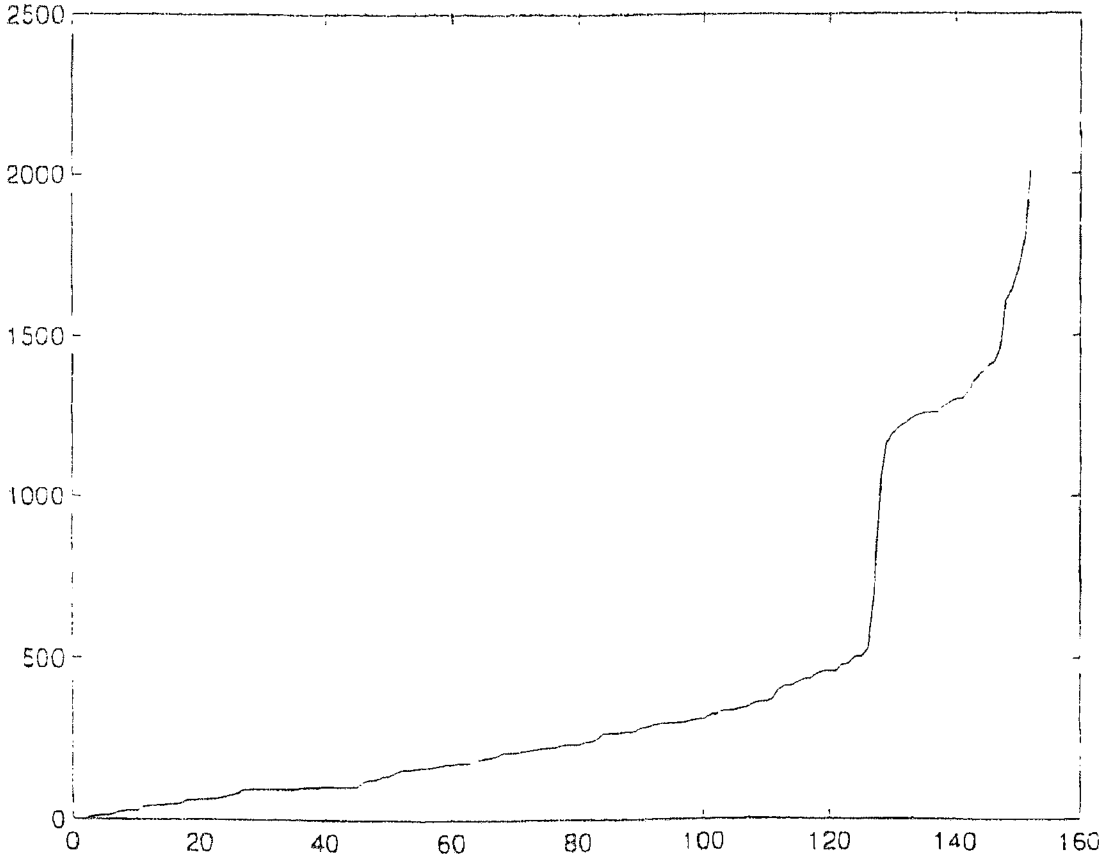

Figure 3.3 The ordered h_k^2 distribution of 150 points of the Figure 3.2

It may be observed that if a point k is close to ac cluster i the h_k^2 is approximately same as d_{ik}^2. To get a better understanding of the relation between h_k^2 and d_{ik}^2, it may also be useful to investigate the following function

$$f(d_{ik}^2) = (d_{ik}^2 - h_k^2) = d_{ik}^2\left(1 - \frac{1/d_{ik}^2}{1/h_k^2}\right) = d_{ik}^2\left(1 - u_{ik}\right) \qquad (3.9)$$

Since $u_{ik} \leq 1$ so (3.9) is always positive. The point k local ed far away from the cluster i, the distance d_{ik}^2, is high and the corresponding membership u_{ik} is low; overall $f\left(d_{ik}^2\right)$ will be high. For points close to the clusters, the reverse occurs. This observation will be used in the formulation of objective function of fuzzy C-LMS algorithm in later section.

3.3.2 Objective Function of Fuzzy c-LMS Algorithm

Considering the data set has as high as 50% contamination, the median h_k^2 will correspond to the limit beyond that the k will correspond to a noisy data point and all good data will have h_k^2 less than the median value. From this observation it may appear logical to conclude that minimizing the median h_k^2 should be the objective of fuzzy c-LMS algorithm. However, for less than 0.5 contamination in data set, it will not partition good and bad data in the ordered h_k^2 distribution and the above minimization may detect only $c - c1$ clusters containing 50% 'good' data. Since the points corresponding to $c1$ clusters may have h_k^2 above median are free to have any higher value, as a result $c1$ clusters may remain undetected in spite of the objective of minimum median h_k^2 has been achieved. We can overcome this problem by trimming a fraction from the beginning so that the median value is at the boundary of good and bad data of ordered h_k^2 distribution. The trimming percentage depends on the degree of contamination ε, can be found out as $t = 2 * (0.5 - \varepsilon)$ and is an unknown to this problem.

Therefore we can formulate the objective function of fuzzy c-LMS as given by

$$J(U,v)_{FCLMS} = \min(h_{k:med}^{2})_{t} \tag{3.10}$$

Here, $h_{1:N}^{2} \leq \cdots \leq h_{N:N}^{2}$ are the ordered harmonic mean distances w.r.t. all the feature points $x_{j:N}$. The $(h_{k:med}^{2})_{t}$ is the median harmonic mean distance after trimming away t% of data from the beginning. The key Issue in this algorithm is therefore the determination of the trimming ratio t so that the minimum median harmonic mean distance is at the boundary of 'good' data and noise points. Since h_{k}^{2} is function of cluster parameters we can apply a strategy of globally searching for the clusters to achieve the minimum of the above functional. Since the exact value of t needs to be evaluated by gradually increasing the trimming fraction. It is therefore necessary to use some type of validity criterion to measure the 'goodness of fit' of the data to the clusters at each step and complete the cycle of gradual increase of t in order to determine optimum trimming fraction for best performance. But this technique should not be recommended as a good practice. Since we are actually solving the same problem repeatedly at each step of trimming, results in high computational burden. More over the optimum trimming percentage t is based on validity criterion. Decision based on only validity measures is questionable because a single validity criterion is not appropriate for all kinds of data set.

To eliminate the problem of repeated estimation of the same clusters we propose an alternate strategy, to partition the data set into c regions through fuzzy segmentation and compute the minimum median residual (i.e. LMS) in each zone and formulate the objective function accordingly to sum them over all partitions. Since the partition has already been established it is practical to considered only those points within each partition and compute the regular LMS estimate only with in each region. The partitioning of the data set is obtained by assigning the point k to that partition i for which $(d_{ik}^2 - h_k^2)$ is minimum and subsequently minimize the median value. This has similarity with α-cut of the i th partition and where no link exists of the data among regions [55]. The objective function of fuzzy C-LMS algorithm can be written by slightly altering the minimization criterion as

$$J(U,v) = \min \sum_{i=1}^{c} med(\min_{i}(d_{ik}^2 - h_k^2))_{k=1..N_i} \tag{3.11}$$

where N_i is the number of data points in the partition i.

The distribution of $f(ik) = (d_{ik}^2 - h_k^2)$ is shown Figure 3.4. It is observed, if all the estimates are picked correctly in one iteration, the ordering of this function of distance of point k shows a behavior of arranging the points assigned to region i at the top and its associated noise points at the far end. The correct estimate of all cluster parameters will ensure minimum value of the above functional. We therefore globally search for the c clusters by sampling $c \times p$ points at a time. Now the question is how many sampling is at least necessary to find all the clusters simultaneously.

3.3.3 Number of Minimum Random Sampling (m)

Obviously it is computationally very expensive to achieve the global minimum by fuzzy C-LMS algorithm In many applications this would become unfeasible. In such cases we can perform certain number of random selections, such that the probability $(\Pr ob)$ that at least one of the m iterations will result in detection of all good clusters simultaneously is almost 1. An iteration is "good" c samples of p points each correspond to all the correct clusters. According to Dr. Dhar [27], assuming n/p is "large" from the theory of multinomial and multivariate hyper geometry, the probability (P) of detecting c clusters in one iteration can be expressed as

$$P = \frac{C_p^{g_1} \cdots C_p^{g_c}}{C_{c \times p}^{n}} \tag{3.12}$$

If g_i represents the number points in cluster i then for large n if g_i/n converges to a constant we may approximate the above expression as [27]

$$P \approx \frac{(cp)!}{(p!)^c} \left({g_1}/{n} \right)^p \left({g_2}/{n} \right)^p \ldots \left({g_c}/{n} \right)^p \tag{3.13}$$

If all the clusters has nearly equal number of points we can approximate $g/n = (1-\varepsilon)/c$. Therefore the probability of detecting c clusters at least once in m iterations can be expressed as

$$\Pr ob = 1 - \left(1 - \frac{(cp)!}{(p!)^c} \left(\frac{1-\varepsilon}{c} \right)^{cp} \right)^m \tag{3.14}$$

based on the number of clusters c, the degree of contamination ε and the number of parameters of the cluster p we can determine m. Clearly if $p=1$ corresponds to equation (3.2). In fact we tabulated the minimum iterations m required for $p \le 5$, with ε between 0.1 and 0.5 and $c=1, 2, 3$. This permits us to avoid the extensive search and reduce computations substantially while data contains up to 50% noise. As we can find the number of iterations computed from equation (3.11) becomes tremendous for large p and c, at least in the most extreme case of $\varepsilon=0.5$. In fact after a certain stage, the computation time become astronomically high and is therefore impractical to calculate.

This problem has been solved in a faster approach based on gathering repeated evidence of the clusters as described later in a later section.

Table 3.1a Number m of random iterations, determined in function of p, c and ε by requiring that the probability of at least one good sampling 95%. Number of cluster $c=1$

Dimension	Fraction ε of contaminated data				
p	0.1	0.2	0.3	0.4	0.5
1	2	2	3	4	5
2	2	3	5	7	11
3	3	5	8	13	23
4	3	6	11	22	47
5	4	8	17	38	95

Table 3.1b Number m of random iterations, determined in function of p, c and ε by requiring that the probability of at least one good sampling 95%. Number of cluster $c=2$

Dimension	Fraction ε of contaminated data				
p	0.1	0.2	0.3	0.4	0.5
1	6	8	11	16	23
2	11	18	32	61	126
3	17	36	80	204	613
4	24	64	189	651	2804
5	34	112	430	2012	12464

Table 3.1c Number m of random iterations, determined in function of p, c and ε by requiring that the probability of at least one good sampling 95%. Number of cluster $c=3$

Dimension	Fraction ε of contaminated data				
p	0.1	0.2	0.3	0.4	0.5
1	17	25	38	61	107
2	45	92	205	519	1552
3	90	261	869	3482	17969
4	162	668	3319	21107	188196
5	179996	1.0533E+06	7.80603E+06	7.88192E+07	Inf

3.3.4 The Fuzzy c-LMS Method

We assume c number of clusters being detected from a data-set containing edge data points within m times of sub-sampling. The data-set is assumed to contain ε fraction of contamination. The distance of point k to cluster i is d_{ik}^2, its harmonic mean distance is h_k^2 and the fuzzy membership of the point to cluster i is $u_{i,k}$.

(1) Pick $c \times p$ number of distinct different points randomly; ensure the set of points were not picked in earlier iterations.

(2) Calculate the parameters of c clusters, p picked points for each cluster.

(3) Calculate all d_{ik}^2 of each point k from c estimates of clusters and the h_k^2.

(4) Calculate h_k^2 of N points, create fuzzy partition by minimum $f(ik) = d_{ik}^2 - h_k^2$ for $i = 1,..,c$ clusters.

(5) Order all points $k = 1..N_i$ in each partition i according to $f(ik) = d_{ik}^2 - h_k^2$ and sum the median values for all c regions.

(6) Sample iterativly 'maxiter' times, the 'maxiter' obtained from Table 3.1.

(7) While sampling, the estimates produce the minimum value of equation (3.10) are the correct results.

3.4 Blending FCLMS with RHT

To get a better understanding the role of repeated evidence in random sampling let us analyze the randomized Hough transform or RHT algorithm [77]. RHT is a kind of probabilistic method, aims at gathering evidence of a good cluster directly from image space. It randomly picks p points in image space for an p-parameter curve, and maps them into one point in the parameter space at each step of the iterative procedure by solving p joint equations. Since any RHT related algorithm is a local search based estimation process, uses a parameter data set with each element containing both a real valued vector and an integer score to implicitly represent the parameter space. RHT detects one cluster at a time, only based on repeated evidence level (n_t) of cluster parameters within closed tolerance δ. For convenience, if we assume there are c clusters within an image without any outlier and p parameters to be detected for each cluster and number of data points n, total possible number of sub-sampling required to extract all clusters is

$$N_{RHT} = (c^p + (c-1)^p + \ldots + 2^p + 1^p) \times n_t = \sum_{i=1}^{c} (i)^p \times n_t \tag{3.15}$$

where n_t represents the threshold number for searching a possible cluster. In [31] n_t has been suggested as 2 or 3.

The number of computations required to can be obtained from the above equation (3.14) is astronomically high. In presence of noise and outliers the number of mapping for extracting all clusters is given by

$$N'_{RHT} = ((c+1)^p + c^p + \ldots + 3^p + 2^p) \times n_t = \sum_{i=1}^{c} (i+1)^p \times n_t \tag{3.16}$$

where all the noise points may be considered another cluster and n_t represents the threshold number for searching a possible cluster. Here n_t should be larger than 2 or 3 as degree of contamination increases. Clearly the number of computations become even higher. It may be observed that to apply RHT we need the apriori information of the values of two parameters, tolerance δ and threshold n_t. Both these parameters are very important for accurate estimation of the cluster parameters, are difficult to evaluate before hand with out any earlier information of the data set. Consequently the usefulness of this extensive computation in practice is questionable. It is therefore necessary to probabilistically predict the limited iterations necessary to come up with a list of distinctly different possible estimates that will include all the correct ones. RHT also has several other limitations. The problems encountered in clustering through RHT are, ability to search only one cluster at a time leads to higher computations for repeated search in data-set for multiple clusters and validating the detected cluster.

In sampling based fuzzy c-LMS cluster detection algorithm, while randomly picking the points in each iteration it is hard to predict the location of picked points relative to the cluster position and orientation. In fact it is impossible to know while sampling whether the picked points are 'good' or belong to the noise and outliers. Since the fraction of 'good' points belonging to the clusters are generally more or at least equal compared to the that of the outliers, it is more likely that while sampling, the picked points will represent the correct estimate (at least within close tolerance δ) repeatedly assuming that the dataset is large enough and there is no repetition of previous sampling. If there is biasing noise resembling the cluster shape, it may generate the evidence of spurious clusters. However, as we increase the number of random sampling, it will exhibit more evidence of the correct estimates. In fact if we conduct an exhaustive search by sampling all the points in all possible combinations, the correct estimates will have the maximum number of evidences.

The tolerance δ needs to be specified apriori, is introduced primarily to reduce storage of estimates. The repetition of evidence is directly related to the value of δ. It is difficult to select without any information about scale of data. It also depends on goodness of fit of the points assigned to the clusters. The problem of specifying tolerance δ is similar to the problem of specifying the 'noise distance δ' in noise clustering [18, 26]. In fact both the approaches eliminate the influence beyond a certain distance δ while estimating the cluster parameters, while RHT does it in parameter space but noise clustering requires it in the image space. If we consider clusters with parameters $p=1$, i.e. image space and parameter space are identical, then this δ in both RHT and noise clustering are equal. Essentially noise clustering tries to fit a prototype that contains maximum number of points within a pre-specified distance δ from the estimate. As shown in [24, 25], noise clustering algorithm can be reformulated as a robust M-estimator [46]. Similarly the evidence gathering technique can enhance robustness, while estimating through sampling, by elimination of those estimates beyond a distance δ in the parameter space. In fact the averaging of those candidates in parameter space within a distance δ should be a better estimate of the cluster.

As we know in fuzzy C-LMS the number minimum sampling is as shown Table 3.1c is very high, we therefore propose an optimal blending of the concept of repeated evidence while searching for the correct clusters there by reduce the number of sampling. Since all those cluster parameters with higher evidences are more probable to be the correct estimate therefore one combination of such probable estimates of clusters, results in minimum sum of median $f(ik)$ in all partitions, is concluded as the correct estimates of the clusters. Since the performance of this algorithm rely heavily on probable estimates obtained through repeated evidence while sampling, it may be wise to use RHT alone. But here we can eliminate the requirement of specifying the fixed threshold n_t

since only high evidence level estimates are considered as possible candidates. These estimates can be grouped based on clustering algorithm.

3.4.1 Modified Fuzzy c-LMS Method

(1) The first two steps are same as fuzzy c-LMS algorithm of sampling the cluster parameters.

(2) Increase the counter by comparison with the previous estimates' of the cluster parameter values, if they are within tolerance limit, as a measure of the collected evidence.

(3) One of the c cluster found at least 3 times while searching iteratively is considered as possible estimate.

(4) Complete picking iteratively 'max' times, the 'max' depends upon and generate the list of possible estimates of c clusters.

(5) Try all possible combination of sets amongst the probable estimates of the clusters and apply fuzzy c-LMS algorithm. The one combination, that results in minimum value of sum of median $f(ik)$ in all partitions, is concluded as the correct estimates of the c clusters.

3.5 Experimental Results and Discussion

To test the algorithm we considered line as a prototype. Since a line can be described by only two points, therefore a few outliers and noise may "conspire" to generate more evidences for a fictitious line, is more than just possibility. From [7, 10] it is known that if the data-set contains only one line the LMS estimate will be in the middle of a narrowest 'strip' or a 'band' that will contain 50% of the data points. Instead of performing a minimization of median residual as in LMS, here we minimized the median $f(ik) = d_{ik}^2 - h_k^2$ in each partition i. Figure 3.6 shows a noisy data with three clusters al lines and the algorithm works fine. The data contain 30% random noise and estimates found within 56 sub-sampling. Also the Figure 3.7 shows contaminated data of three clusters but here noise is 50%. The correct estimates were found in 95 sampling.

In order to increase the probability from 0.95 to 0.995 that the estimates has been found at least once in m iterations, obviously the value of m will rise tremendously. So if we keep on increasing this probability although this results in more computations, it will be highly probable that the estimates are closer to the global minimum. In fact if the sampling has been performed following all possible combinations of the data points *i.e.* C_n^p it will guarantee that the global minimum of the objective has been reached. But C_n^p usually results in a very large number as n and p increases and is not at all useful in practice. However if the precision of estimate is more critical in an application and the computation time is not a constraint, this exhaustive search process should not be inappropriate.

This is a search based algorithm, so it does not involve the problem of good initialization. Since the correct estimate will have one cluster of good points along with its associated noise in each of the c components of the partition, minimization of the objective in each component will ensure estimation of each cluster perfectly. It is important to mention that component wise the number of good points in the cluster should be more or at least equal to its associated noisy data. Otherwise the 'bad' points may appear to look like a good estimate. Therefore the algorithm achieves component wise breakdown point of 0.5 and thereby can tolerate 50% noisy in the whole data set. It will find it difficult to estimate if component wise noise exceeds above 50%. However an improvement is suggested by setting the partition so that more noise points are assigned to the larger cluster thereby smaller clusters will have less noise points associated with it. One way to reduce this problem is to incorporate the concept of collecting repeated evidence of the clusters, similar to RHT, while sampling. While sampling, there is no way to estimate the position of the picked points. Since the fraction of good points is more in the data set, obviously there will be more sampling from good points. This will lead to repetition of the same estimate in subsequent samplings. Thus we can generate a list of possible estimates with repeated evidences and apply fuzzy c-LMS to these estimates only. So this will reduce the number of false iterations and the computation cost will also reduce substantially. This leads to the development of a fast and robust clustering algorithm, named as fast fuzzy c-LMS algorithm. The strategy has been tested and proved successful. For the example in Figure 3.6 with 30% noise in data, the regular FCLMS require 55 sampling while fast FCLMS completed the task in only 20 sampling. The same estimates were obtained for Figure 3.7 with 50% noise in data, by Fast-FCLMS in 35 iterations while regular FCLMS method required 95 sampling. As the fraction of contamination in data increases, algorithms require more sampling. Although, not shown here, many other examples were tested and similar good results were obtained. However, when a typical range image of quadric surfaces is considered, e.g. Figure 3.8, even Fast-FCLM could not detect the clusters. For each cluster, ten points were required to be picked. Here good sampling requires each set of ten points to be picked from each individual cluster. This requirement apparently proved to be quiet stringent. So the quadric surfaces could not be detected even after substantial amount of sampling.

3.6 Conclusion

LMS, a robust estimator can detect the single cluster, with high break down point 0.5, if the data has only one cluster. It is extended here to concurrent multiple cluster detection through fuzzy partition. We show that the concept of evidence gathering technique, similar to randomized Hough transform can be incorporated along with fuzzy c-LMS algorithm for faster and improved estimation. It is possible to apply this technique to detect other type of prototype shapes e.g. circles, ellipses, spheres ellipsoids, cylinders. As such the computation of distance of each point from quadric surface prototype is computationally very expensive. More over the algorithm requires ordering of points of all points based on residual in each iteration has turned out to be big task. So in our problem of detecting of quadric surfaces from range image, since the number of data points are very high, often more than 5000, the ordering of points in each sampling is a big computational burden.. Though FCLMS is a robust method, it is not very appropriate for the current problem. To overcome this problem a similar sampling based robust algorithm has been introduced in next chapter.

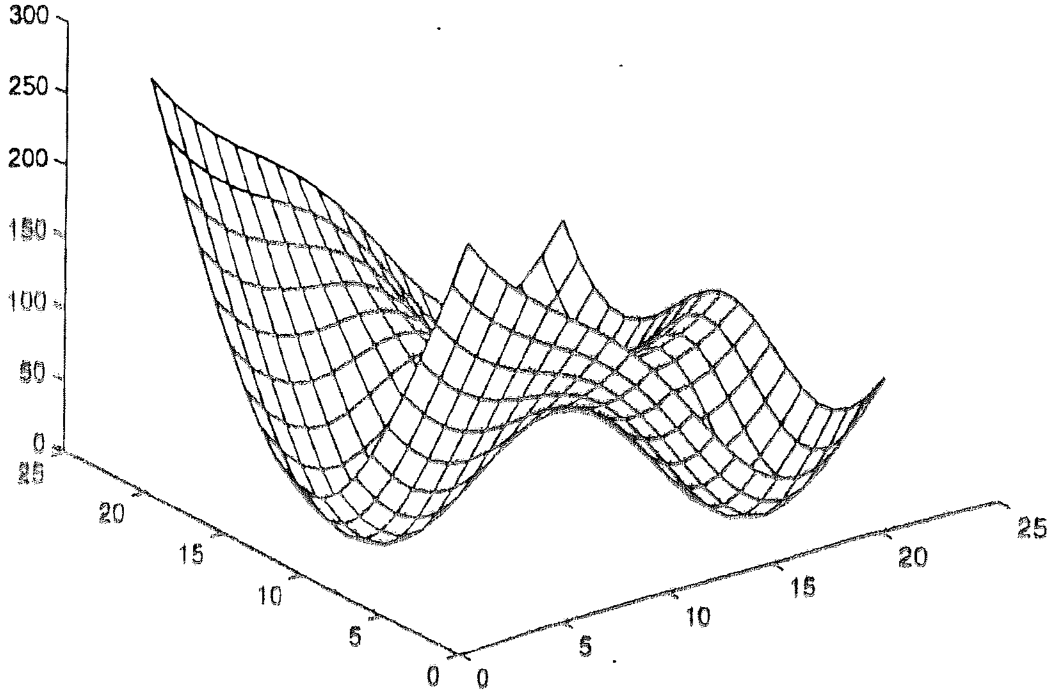

Figure 3.4 The distribution of h_k^2 of the points of the three clusters in Figure 3.2

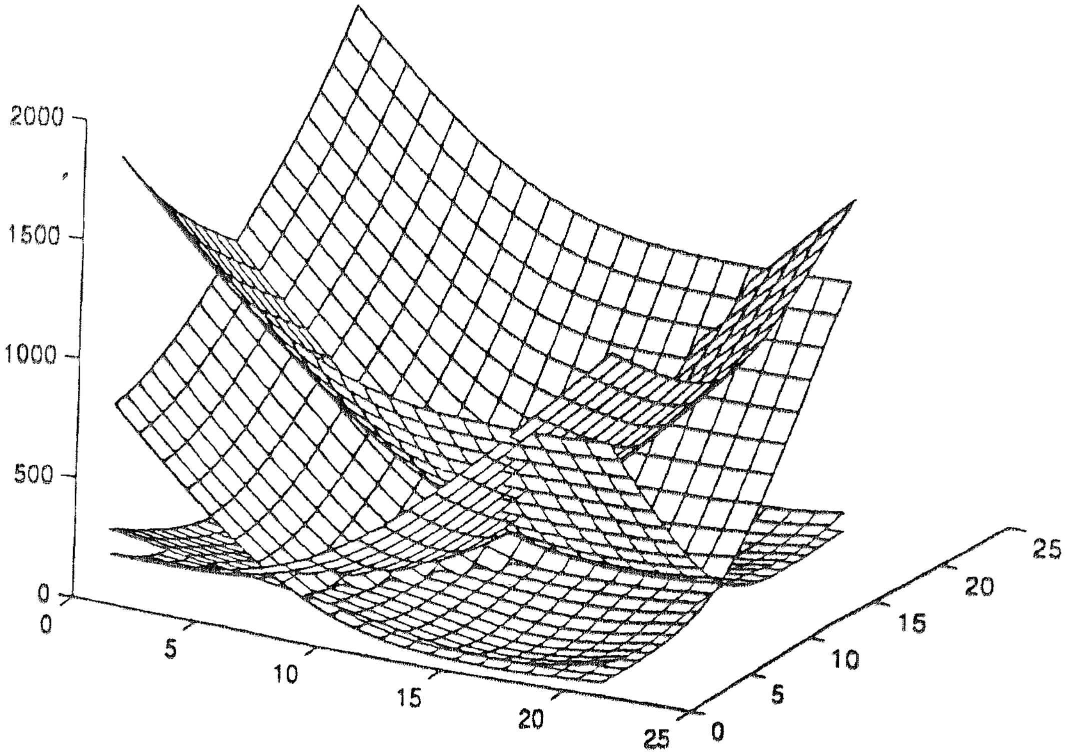

Figure 3.5 The distribution of h_k^2 and d_{ik}^2 of the points of three clusters in Figure 3.2

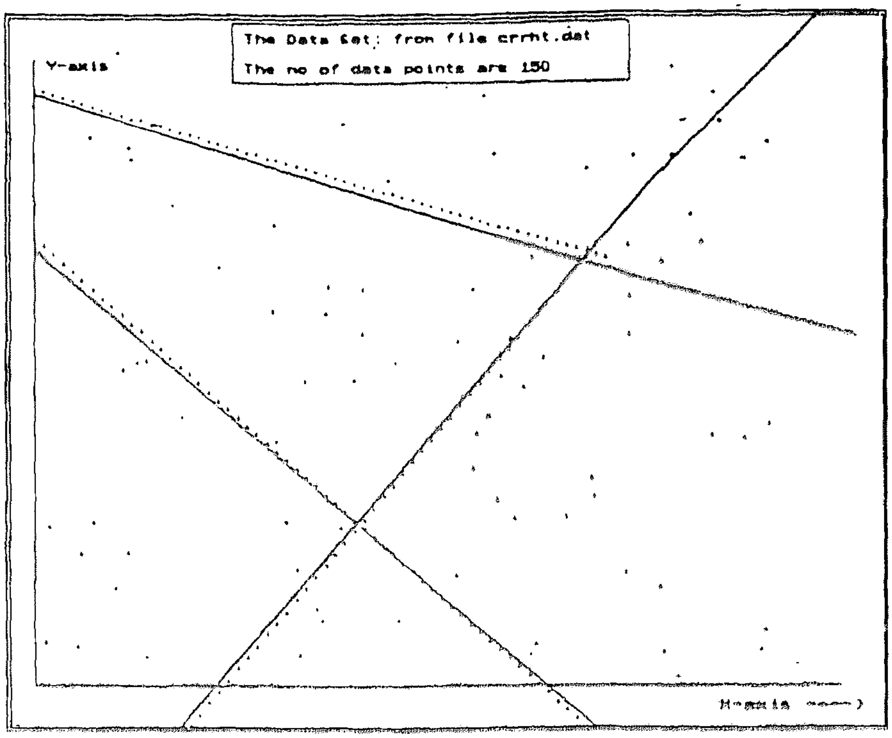

Figure 3.6 Three clusters detected as lines by FCLMS method with 30% noise in data

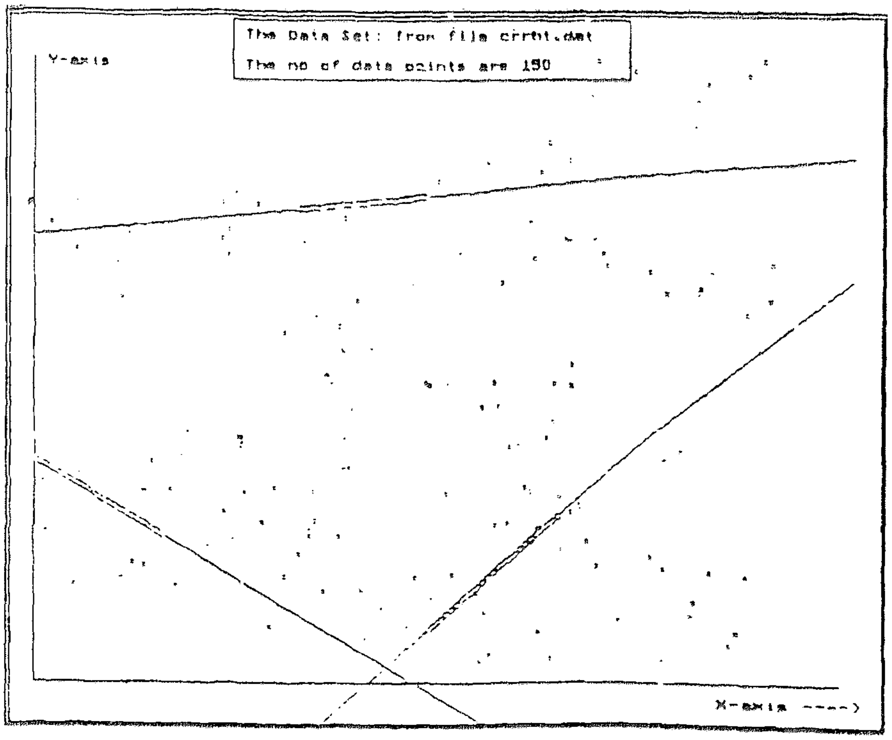

Figure 3.7 Three clusters detected as lines by FCLMS method with 50% noise in data

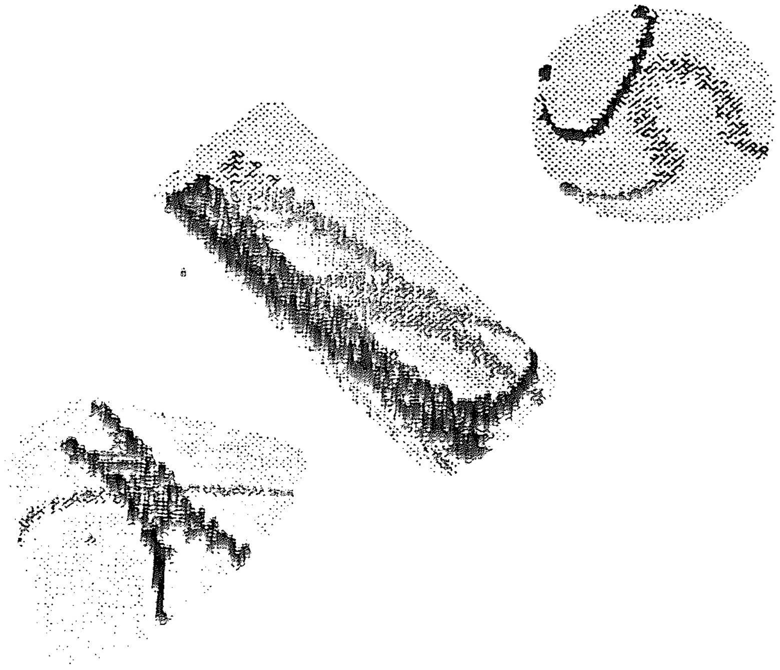

Figure 3.8 Three quadric surfaces could not be detected by FCLMS method

CHAPTER 4

SAMPLING BASED NOISE CLUSTERING

In Chapter 2 we generalized the NC algorithm so that points may have variable noise distance, however the problem of initialization of cluster prototype is still a disadvantage. In Chapter 3, we discussed the sampling based robust LMS algorithm to avoid the problem of initialization and extended it to detect *c* clusters simultaneously. Considering the requirement of exhaustive search, a limited probabilistic search was attempted, and we tried to further reduce the number of computations. A faster version of the algorithm has been presented by blending LMS with RHT. Although it performed well in detection of multiple clusters from 2 dimensional edge images, its applicability in object detection from range images was not very successful. Since the number of data points are usually very high in range images of quadric surfaces, the ordering of all points based on the distance from the prototype in each sampling is an impractical task.

4.1 Comparative Analysis of NC, LMS and RHT

Cluster analysis handles the task of classifying the data into a number of subsets. There are two fundamental approach of classification, namely (a) solving the model fitting problem based on minimization of squared error criterion through analytical technique such as gradient descent, and (b) model fitting by search based technique that finds the best fitting cluster through sampling the data through an evidence collection based approach. Both areas have received considerable attention in research, which resulted in development of number of clustering algorithms. However each method has a limitation in its robustness features. The analytical approach is primarily derived from least square (LS) error minimization whose performance degrades drastically with noise and outliers. Since, fitting the clusters according to prototype-based fuzzy c-means (FCM) algorithms is such an approach; it is not very robust. The noise clustering (NC) algorithm has been proposed to robustify the FCM-type algorithms by making it less sensitive to noise and outliers through relaxing the constraint on the memberships. Due to this, the sum of the memberships of a bad point to all good clusters is not forced to be equal to one. The noise clustering (NC) considers noise to be a separate class and represents it by a prototype that has same distance δ from all points. In an earlier chapter, section 2.3, noise clustering has been generalized to detect variable size clusters and shown to be equivalent to a *c*-class robust M-estimator, where the modified harmonic mean distance is used as a residual. While M-estimator are very popular due to their high efficiency and good robustness, their major shortcoming is that they do not have high break down point.

The low breakdown point of an M-estimator based approach may be even serious disadvantage when coupled with a sensitivity to initialization for FCM and even NC techniques. The problem arises due to use of re-weighted least squared error minimization approach. The iterative minimization scheme is not guaranteed to converge to a global minimum. Therefore incorrect initialization in presence of biasing noise in data may result altogether in poor classification.

The sampling based methods on the other hand does not suffer from this problem of initialization, where the objective is to find strong or repeated evidence of the correct estimate. It is possible to search for a cluster directly in image space, *e.g.* least median square (LMS), minimum volume ellipsoid (MVE) etc. [66]. Alternatively we can search the cluster in parameter space through randomized Hough transform (RHT), which searches the parameter space for the peaks or highest evidence of the cluster [77]. One of the advantage of sampling based approaches is that it guarantees that the global minimum may be achieved, but at the cost of the high amount of computations required due to the need for an almost exhaustive search. Though it is argued that based on probabilistic reasoning as suggested by Rousseeuw and shown in earlier chapter section 3.4, the search may be made limited for specific problem. The number of random sampling necessary to guarantee that a good cluster may be detected depends on the fraction of contamination in data and the number of parameters of the estimate of the cluster. The RHT method has been discussed in chapter 3 and also in [31] shows that as the dimension of the estimate or the threshold increases, the number of computation increases astronomically. The robustness aspects of least median square (LMS) algorithm and *c*-class extension has been discussed in details. Although it guarantees high breakdown point, up to 0.5, it requires ordering of data in each sampling. If the number of data points is very high, as in our case of classifying clusters in the range image of quadric surfaces, the ordering of all the data points in each sampling is a big computational task. These difficult limits the application of the algorithm in range images with large number of data points. Comparatively, the main advantage of RHT technique is that it does not require specifying the number of clusters apriori. One cluster at a time is removed from data set until not enough data points are left to be classified. While searching, RHT requires to maintain an accumulated array of the parameters of the prototypes that determined in earlier sampling thereby it necessitates continually expanding runtime memory. This may often exhaust the runtime memory available. It requires user to specify the tolerance δ_p

in the parameter space and the threshold value of iteration $k_{\max}$. The selection δ_p and $k_{\max}$ is often very difficult since the scale of data in the parameter space is almost impossible to predict from image data.

In order to circumvent these problems a new sampling based clustering algorithm is proposed that produces robust solution by combining the good features of other search based algorithms. This includes high robustness or breakdown point of least median square (LMS) or the capability to detect clusters without the knowledge of number of clusters of RHT as well as faster solution of fitting based techniques. The method basically attempts to search for one cluster at a time directly from image space following the similar principle as NC algorithm. The algorithm is detailed in the next section.

4.2 Sampling Based NC Algorithm for Single Cluster

The noise clustering technique is detailed in an chapter 2 section 2.4. In the original algorithm, Davé [18] considered noise to be a separate class, represented it by a prototype that has same distance δ from all feature vectors. The definition of noise prototype and related material is repeated below specifically for a single cluster problem.

<u>Noise prototype</u>: Noise prototype is an universal entity such that it is always at the same distance from every point in the data-set. Let $\beta *$ be the noise prototype, and x_j be the point in feature space. Then the noise prototype is such that the distance $d\left(x_j, \beta_i\right)$, which is the distance of point x_j from $\beta *$, is a constant value δ.

This definition does not specify what the distance is, but it states that all the points have equal apriori probability of belonging to the noise cluster. Considering a hypothetical situation that the data contains only one cluster with some outlying elements, the membership of the point x_j is u_j and to the noise cluster as u_{*_j} is defined as,

$$u_{*_j} = 1 - u_j \tag{4.1}$$

Since (4.1) is used to define the membership u_{*_j} in the noise class, the usual membership constraint of FCM algorithms is not required. Thus, the membership constraint for the good clusters is effectively relaxed to

$$u_j \leq 1 \tag{4.2}$$

This allows noise points to have arbitrarily small membership values in good clusters. The objective function is given as

$$J(B,U;X) = \sum_{j=1}^{N} u_j^m d^2(x_j,\beta) + \sum_{j=1}^{N} \delta^2 (1 - u_j) \tag{4.3}$$

In (4.3). $d^2(x_j,\beta)$ is the distance from a feature point x_j to the prototype β.

The above functional is optimized with respect to the prototypes and the memberships in a manner similar to FCM functional as shown in Ref. [4, 6]. The resulting equations for the prototype parameters are very similar to that of FCM, however, the equation for the memberships reduces to as given,

$$u_j = \frac{1}{1 + \left(d_j^2 \big/ \delta^2 \right)^{1/(m-1)}} \tag{4.4}$$

For $m = 1$ this gives a better understanding of the effect of the noise distance. All the points having the distance from the prototype less than the noise distance δ are classified into the good cluster, while all the points having the distance from the prototype more than the noise distance are classified into the noise cluster. Thus the noise distance becomes a switching criteria. For $m \neq 1$ the case is fuzzy, and one can visualize the points belonging to the cluster in terms of its fuzzy membership. In such cases too, the noise distance becomes a switching criteria, but the switch is fuzzy. When there are more than one cluster, the situation becomes much harder to visualize, but the effect is essentially same which is discussed in Chapter 2. It is even reported in Ref. [18, 24, 25] that if the scale of data is known, the selection of δ is not problem. In fact in many in scientific and engineering situation the value of δ is known apriori. Under this circumstance it is possible to globally search for one cluster at a time by sampling p points from feature space to estimate the cluster with p parameters, similar to LMS or RHT. The noise distance δ acts like a switch and actually determines whether a point belong to a cluster or not. All points beyond δ from the estimated prototype are considered as noise. It is clear, the specific sampling that result in maximum number of points in the cluster as per the membership equation (4.4), can be used to calculate the cluster parameters. As we know if data contains only one cluster (i.e. $c = 1$), fuzzy c-means (FCM) reduces to least square (LS) algorithm. It has been discussed in detail in Ref. [18, 21, 23, 24, 25], NC is truly a robust version of FCM, therefore for $c = 1$ case this sampling based algorithm may appropriately be called as Noise Least Square or NLS method. Here after this algorithm is called as NLS algorithm. Since the algorithm is framed in global search mode directly in image space, the method is not plugged with the problem of initialization. The sampling based noise clustering for single cluster, maintains only the coordinate of p sampled points that produce maximum number of points within distance δ from the estimated prototype. This method substantially reduces memory requirement compared to RHT where the runtime memory usage gradually increases. In fact, since clusters are removed one at a time the computation load gradually reduces and it does not require prior knowledge of number of clusters. Here the algorithm is continually sampling for a better estimate directly in the image space from the region with maximum density of good points to estimate the cluster parameters.

NLS Algorithm

(1) Fix the minimum number of points in a cluster N_l and the minimum number of sampling N_s of the cycle before selecting the cluster parameter.

(2) Randomly select p points from the image space and calculate p parameter of the prototype by solving p joint equations.

(3) Calculate the distance of all the points from the calculated prototype and the number of points N_k within the specified distance δ.

(4) The value of maximum N_k is updated in each sampling. The estimate that produce maximum N_k is considered as the most probable estimate.

(5) At the end of sampling cycle all the points within distance δ from the most probable estimate is removed. If the remaining number of points is less than N_k then stop otherwise go to step (2).

Since one cluster at a time is detected and removed from original data, it is not necessary to know the number of clusters in the data. Moreover as the points are removed the original data, also the number of distance computations in each cycle of sampling gradually reduces. This increases the speed of solution.

The NLS algorithm can be constructed so as to determining c clusters simultaneously. The method is described below. Similar to regular NLS, prior specification of δ is mandatory

C-NLS Algorithm:

(1) Fix the minimum number of points in a cluster N_t and the minimum number of sampling N_s of the cycle before selecting the cluster parameter.

(2) Randomly select $c \times p$ points from the image space and calculate p parameters of each of the c prototype by solving cq joint equations.

(3) Calculate the distance of each point from c prototypes and the number of points c within the specified distance δ for each cluster N.

(4) Maintain and update the maximum N_{tk} in each sampling. Ensure all the c prototypes are not identical with each other.

(5) Continue sampling until all c prototypes are detected.

One of the disadvantage of C-NLS algorithm is, it requires prior knowledge of the number of clusters in the data. This somewhat limits its applicability in situations when we have no knowledge of the image. Moreover the method has to ensure, clusters detected in each sampling are not identical with each other. This later problem often forces large number of random sampling and thereby reduces the speed of detection. From the algorithm it is clear, unless each set of N_k points is sampled from each individual cluster. C-LMS algorithm won't generate correct result. This is relatively a restrictive condition compared to regular NLS, which poses to be a better candidate for detection quadric surfaces from range image.

4.3 Mathematical Analysis of NLS Algorithm

The key aspects of NLS algorithm are: (1) high breakdown point of up to 0.5 (2) precision and accuracy of estimation (3) increased speed of detection, thereby reduced computational load.

The high breakdown point of NLS algorithm has been derived from its close resemblance of its procedure with LMS algorithm. Usually the good points in the data are close to each other, forms the cluster sub-structure and noise and outliers are spread all over the image space. Often the outliers combine together to form biasing noise and may even conspires to look like the prototype. Since the estimates obtained by NLS method will be in the region that contains maximum density of points, it can tolerate high level of contamination in data. As long as the δ for the particular data set is selected correctly, the NLS algorithm can tolerate 50% noise in data *i.e.* the theoretical maximum. If there is more contamination in data, the noise and outliers can form denser region and then NLS algorithm may evaluate incorrectly that as an estimate. However, if noise is spread evenly all over the image and the good points forming the cluster as the densest region in image, as in some practical situations, NLS algorithm may tolerate more than 50% noise in data. The mathematical basis for this observation is required to be checked.

The issue of selection of δ in correct range is very important. Over and above, it determines the precision of the prototype parameters evaluated. Most often the δ will depend on the scale of data. The value of δ also influences the number of sampling in each cycle N_s. It is also very important issue in successful detection of the cluster. If δ is relatively small N_s will be required to be comparatively high or vice versa.. Because if N_s is low the sampling may not be able to detect the right cluster whereas if N_s is high there will be large number of false iterations. The false iterations will obviously increase computational burden. By false iteration we mean that the sampling in which all the randomly selected p points are not from the same cluster.

In order to improve the sampling a better practical method has been suggested. This idea has also been tested in Ref. [31]. The objective of this technique is to restrict the search only within a specified region so the randomly selected points are more probable to be from the same cluster. Though this does not restrict the distance calculation of all the points from every other region. Although the number of clusters in an image is invariant, we can arbitrarily divide the whole image into multiple regions so that each small region contains approximately one cluster. This is schematically shown in Figure 4.1a. The left image is an ideal situation with one cluster per partition. Where as the right image has one cluster in partition A that spreads over other three (B-C-D) regions. Searching in the smaller region will more likely ensure that the sampled points are from the same cluster. Each smaller windowed region (A, B etc.) will be sampled sequentially. If one partition contain more than one cluster, as in right image, the largest image is more likely to detected and removed faster. The remaining clusters will be detected in subsequent cycle of sampling over the same region. If one cluster spreads over several regions and it is detected while sampling one region where it is first encountered, e.g. the cluster in region A in right image. Once found, the algorithm will remove the cluster from entire image. This may reduce some of the regions empty and thereby requiring no sampling. The sampling cycle N_s for each partition may be restricted to a smaller value thereby minimize the overall computational load.

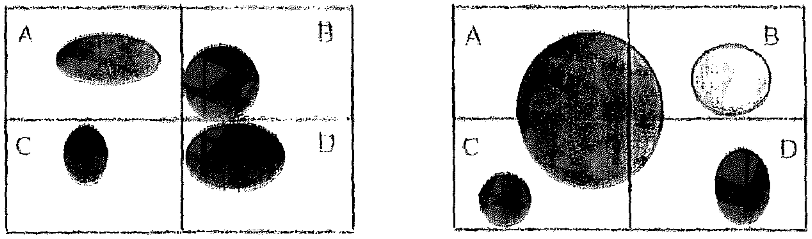

Figure 4.1 Dividing an image into multiple regions

4.4 Distance from Quadric Surface

For a general quadric curved surface, the model can be expressed as

$$a_1x^2 + a_2y^2 + a_3z^2 + a_4xy + a_5xz + a_6yz + a_7x + a_8y + a_9z + a_{10} = 0 . \qquad (4.5)$$

Obviously, there are ten parameters for each surface, and nine of them are independent. When some of the parameters are set to zero, Equation (4.5) represents a specific case, for instance, if $a_1 = a_2 = a_3 > 0, a_{10} < 0$ and $a_4 = a_5 = a_6 = a_7 = a_8 = a_9 = 0$, then the surface represents a sphere. Without any prior information about the detected surface, a general equation of (4.5) is always taken as the default model.

4.4.1 Exact or Euclidean Distance

How to determine the distance of a point from a quadric surface may appear to be simple as it could be decided just by checking whether the values of the point satisfy the expression of the detected cluster. In reality, it is difficult to satisfy the expression even with the points on the cluster because of different types of errors, such as the errors from data acquisition and computation. This problem may be solved by setting a tolerance ε, i.e., if the values of a point (x_i, y_i, z_i) are substituted into the expression of the cluster and satisfy

$$\left\| a_1 x_i^2 + a_2 y_i^2 + a_3 z_i^2 + a_4 x_i y_i + a_5 x_i z_i + a_6 y_i z_i + a_7 x_i + a_8 y_i + a_9 z_i + a_{10} \right\| < \varepsilon, \quad (4.6)$$

then the point can be considered to lie on the cluster. However, a proper value for ε can not be easily selected without prior information about the cluster as it does not have any physical meaning. Therefore, it is suggested to use the actual distance between a point and a cluster to determine whether a point belongs to a cluster. The distance D is defined as the shortest distance between the point (x_i, y_i, z_i) and the cluster. Let (x_c, y_c, z_c) be the point on the cluster, which has the shortest distance to point (x_i, y_i, z_i), then

$$D = \min(\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2}), \text{ subject to}$$

$$a_1 x_c^2 + a_2 y_c^2 + a_3 z_c^2 + a_4 x_c y_c + a_5 x_c z_c + a_6 y_c z_c + a_7 x_c + a_8 y_c + a_9 z_c + a_{10} = 0. \quad (4.7)$$

The Lagrangian method and the routines from MINPACK (refer to the Appendix A for more details) can be used to calculate this distance. In Equation (4.6), nine of ten parameters of the model are independent and the other one can be set as a constant in actual applications. With the Euclidean distance, it is easy to set a threshold value δ to classify a point to a cluster. If the distance D between a point and a cluster satisfies $D < \delta$, then the point is classified to the cluster. There is an alternative method of calculating the distance of a point from a quadric surface.

4.4.2 Approximate Distance

Considering the general quadric curved surface in (4.6), the hyper-spheres and hyper-ellipsoids are extreme cases of it. The definition for the quadric prototype in 3-D space is given as below.

Definition 1: The *i*th quadric curved surface prototype in 3-D space is the set

$$p_i = \{v \in \Re^3 | v^T A_i v + v^T B_i + C_i = 0\}, \qquad (4.8)$$

where

$$v^T = [x, y, z], \qquad (4.9)$$

and

$$A_i = \begin{bmatrix} a_{i1} & a_{i1}/\sqrt{2} & a_{i5}/\sqrt{2} \\ a_{i1}/\sqrt{2} & a_{i2} & a_{i6}/\sqrt{2} \\ a_{i5}/\sqrt{2} & a_{i6}/\sqrt{2} & a_{i3} \end{bmatrix}, B_i = \begin{bmatrix} a_{i7} \\ a_{i8} \\ a_{i9} \end{bmatrix}, C_i = a_{i10}. \qquad (4.10)$$

Equation (4.8) is different from the above defined prototype, where the constant $\sqrt{2}$ is introduced for simplifying the following derivation. Correspondingly, the distance from a point to a cluster can be defined as below.

Definition 2: The distance d_{ik} between a point v_k and a cluster p_i is defined as

$$(d_{ik})^2 = (v_k^T A_i v_k + v_k^T B_i + C_i)^2. \qquad (4.11)$$

Because the computation of Euclidean distance is very time consuming, it is required to be improved. Fortunately, some approaches using approximate distance to replace exact distance have been performed [31]. What interests us is the first-order approximation of the exact distance by which the approximate distance of a point from a surface can be represented as:

$$(d_{Aik})^2 = \frac{(d_{ik})^2}{\left|\nabla d_{ik}\right|^2} \tag{4.12}$$

where ∇d_{ik} is the gradient of the algebraic distance functional d_{ik} evaluated at point k. Undoubtedly, the approximate distance is still a kind of nonlinear distance. Equation (4.12) brings a closed form of distance solution to our algorithms so the computation speed might be extremely raised, and correspondingly, the improved NLS algorithm with approximate distances can be implemented.

4.5 Numerical Results

The NLS algorithm has been tested on various examples of two dimensional edge data and range images. Both exact distance and approximate distance of the points to the prototype has been used in the NLS method.

Consider the range image in Figure 4.2. It is an artificially generated range image of 200x200 pixels, with three clusters, one truncated cone, one cylinder and a sphere. There are all together 8122 points. Apparently this data does not contain any outlying noise. Since only one cluster at a time is detected by NLS algorithm, other clusters are considered as outliers while the method is implemented. If depth values are recorded floating point numbers, NLS could detect all the clusters with δ = 0.001. The results are graphically displayed in Figure 4.3. The strategy to improve speed paid off. The number of iterations have reduced from 110 to 20 while the image is partitioned into 3x3 regions. Each cycle consist of 10 samplings and iterations continued until not enough points are remained to classify. While the depth values are integers, the required δ = 1, since the image is coarse or noisy. NLS detected the clusters in 40 iterations. The performance of course varies with strategy of partition of the range image.

The next range image in Figure 4.4 is generated by adding some noise to an image containing real images, one cylinder and one sphere. The original images were obtained from the Michigan State University, PERP Lab. The total number of points is 16225. NLS method identified the clusters with δ = 0.1 and is displayed in Figure 4.5. It required 160 iterations in regular method and reduced to 50 iterations by the improved version of 2×2 windowing the image. The results clearly indicate NLS as highly robust clustering method. The strategy to improve the speed of detection has been successful.

Figure 4.6 shows an computer generated synthetic image of a lamp shade. There are 13065 points in the image and depth values are recorded in integer numbers. The NLS algorithm could find all the clusters successfully with a δ = 1.8. To implement the faster algorithm the image was partitioned horizontally into three rows. The clusters were detected in 60 sampling and is displayed in Figure 4.7.

All these examples clearly demonstrate that the NLS is a robust algorithm that does not require any specification of number of clusters. Only requirement is specification of δ in correct range. If δ is comparatively smaller, NLS will require more iterations. However if δ is large, the points from other clusters may wrongly get classified and estimate may be wrong all together.

To improve speed of detection the image is partitioned so that each smaller windowed region has only one cluster was successful in all the above examples. The method increased the possibility of sampled points being picked from the same cluster. However determination of the number of partitions is done by trial and error.

4.6 Conclusion

The NLS algorithm has demonstrated very good performance for detection of quadric surface from range image. However if the data contains several adjacent clusters then removal points as a good cluster may pose a problem. The points on the edge of any two intersecting quadric profiles belong to both the clusters. In fact almost every geometric objects have intersecting profiles. Therefore removal of those points by the first detected cluster often lead to incorrect estimates. This problem requires to be addressed with in the algorithm.

The presence of noise and outliers has been handled adequately. While searching for one cluster, points on other cluster appear as noise. In view of that, algorithm is highly robust against high fraction of contamination in data. Therefore the algorithm apparently have a high breakdown point. However a mathematical proof of that is due. The other robustness characteristics of the algorithm are needed to be evaluated from the Influence function curve.

The strategy to improve the speed by partitioning the image was very useful. However for complex objects with multiple adjacent profiles, the simple partitioning strategy may not work. It will need to be reformulated. The algorithm is needed to be checked by using the range image of solid objects in used engineering. Application.

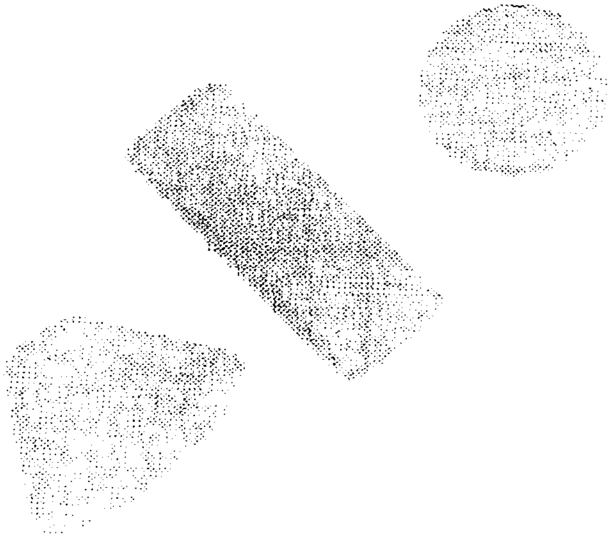

Figure 4.2 The range image of three quadric surfaces with 8122 points

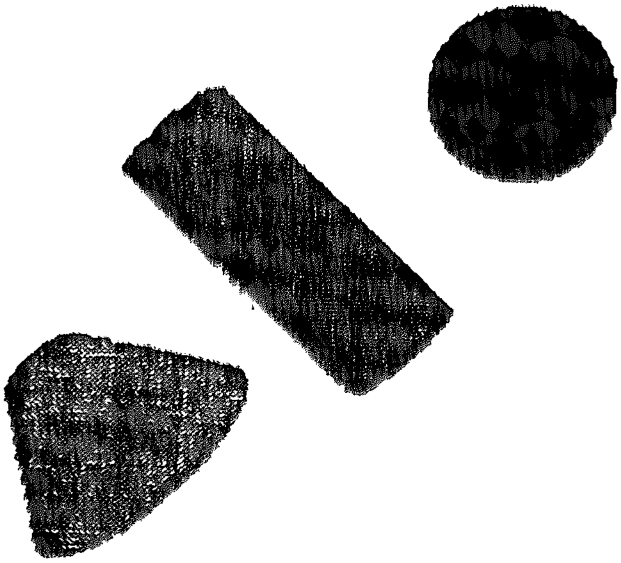

Figure 4.3 Results of the range image in Figure 4.2 by NLS method with $\delta = 0.001$

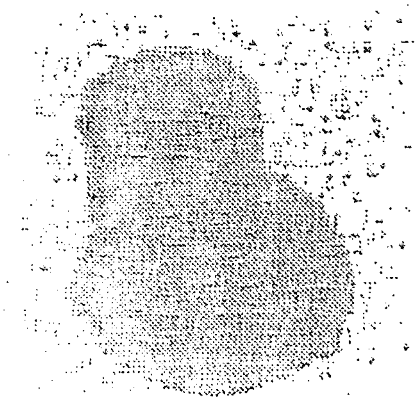

Figure 4.4 Typical noise range image with two real images and some noise points

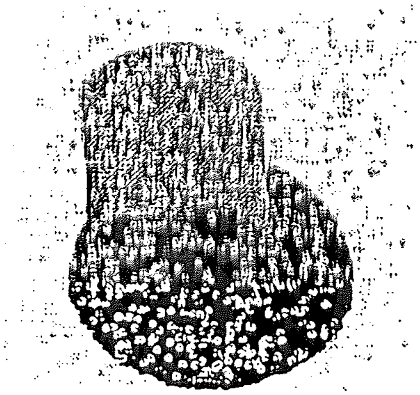

Figure 4.5 Results of the range image shown in Figure 4.4 by NLS method with $\delta = 0.1$

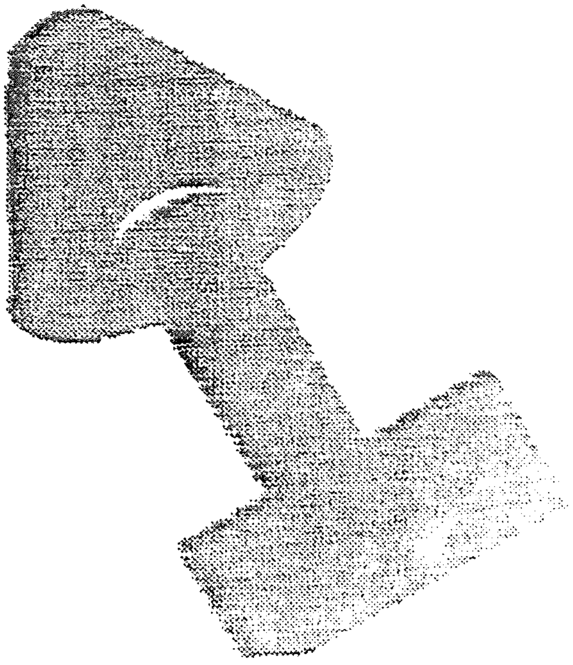

Figure 4.6 The range image of a lampshade with 13065 integer coordinate points

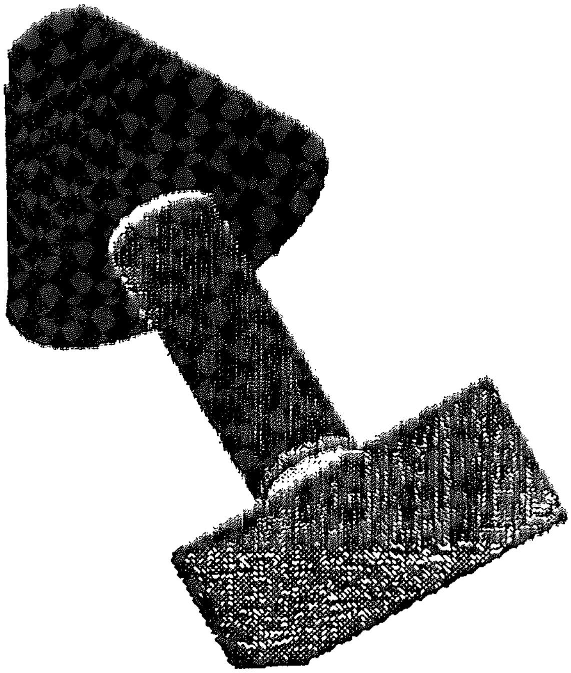

Figure 4.7 Results of the range image in Figure 4.6 by NLS method with $\delta = 0.1$

CHAPTER 5

ROBUST FUZZY CLUSTERING OF RELATIONAL DATA

5.1 Introduction

While clustering of object data using fuzzy techniques has been a very active field of research (see for example, [8]), clustering of relational data has received much less attention. This may partly due to the fact that most engineers and mathematicians usually deal with object data, and rarely encounter purely relational data. However, in fields like management and social sciences, relational data are frequently encountered. The relational data comes from the measure of dissimilarity (or similarity) between objects, and in some cases it is actually based on the object data. For example, in problem of building of a supply warehouse, one often considers the distance from various plants as a basis for generating relational data (see for example, Vinod [73]). Relational data can also be based on subjective expert knowledge, see for example microcomputer data in Gowda and Diday [34], or subjective dissimilarity between countries in Kaufman and Rousseeuw [49]. In general, however, if the object data is available, direct clustering of object data may be more efficient (computationally) than generating relational data from the object data and then partitioning through relational clustering methods. On the other hand, when the data is purely relational, and does not easily fit into any metric axioms, one cannot efficiently apply any object based clustering method, and must employ a relational clustering technique. For n objects, the relational data is usually a $n \times n$ matrix (if the relational measure R_{ij} between objects i and j, is reflective, i.e. $R_{ij} = R_{ji}$ then only a lower triangular portion of $n \times n$ matrix is required). In such cases, there is no explicit knowledge of the "location" of objects in the real space, but such information may be implicit in the relational matrix, and could be made explicit through use of techniques based on multidimensional scaling, albeit at a significant computational cost.

Noteworthy fuzzy techniques for clustering relational data include the methods by Ruspini [67], Roubens [63], Windham [76], Hathaway etal [42], and Kaufman and Rousseeuw [49]. Majority of the relational clustering techniques mentioned above are based on minimization of two types of functionals. For example, techniques by Hathaway etal [42] and Kaufman and Rousseeuw [49] are based on the minimization of essentially the same functional, while the techniques by Roubens [63] and Windham [76] are based on another very similar functional. Although Hathaway etal [42] and Kaufman and Rousseeuw [49] start from almost the same functional, the algorithms for minimization are somewhat different. Not too surprisingly, Hathaway etal [42] cast the minimization as what is called the Relational Dual of FCM (RFCM). In that, clustering of relational data is performed in the relational space using an algorithm that is based on the classical FCM algorithm of Bezdek [6]. As will be discussed later, Hathaway etal [42] use clever arguments to derive this algorithm. On the other hand, the minimization algorithm of Kaufman and Rousseeuw [49] is based on direct application of Lagrange multiplier approach with Kuhn-Tucker conditions to derive the algorithm called FANNY. The approach used by FANNY appears to be a better choice, and it is utilized in this paper to derive new algorithms. Roubens [63] and Windham [76] start from a very similar functional, but Windham defines two different types of memberships instead of a single one in Roubens, and derives a more stable minimization procedure.

One of the objectives of this chapter is to consider increased robustness of clustering techniques for relational data. Several techniques have been introduced to increase robustness of algorithms for clustering of object data, see for example Davé [18], and a review in Davé and Krishnapurum [25]. Davé [18] proposed the concept of noise clustering for making all the fuzzy c-means type object data algorithms robust against noise. Consequently, use of such technique in all the derivatives of FCM type object data clustering algorithms would make those algorithms robust against noise. However, the relational clustering techniques mentioned above are not very robust, although Kaufman and Rousseeuw [49] claim that since the dissimilarity measure R_{ij} appears as an L_1 norm, it finds "medoids" (median based centroids) instead of ordinary centroids, and hence is more robust. However, it is not clear if the use of L_1 norm in this context is equivalent to having a robust location estimator, see for example the treatment of breakdown point of various least-squared error based techniques in Rousseeuw and Leroy [66]. Recently, Hathaway etal [41] have suggested that incorporation of the concept of noise clustering (Davé [18]) in their RFCM would make it robust. However, no results or specific algorithm have been presented. Here, application of the concept of noise clustering is considered to specifically address the problem of robustness in all the popular relational clustering techniques. This also includes robustification of the techniques by Hathaway etal [42] (RFCM) and Kaufman and Rousseeuw [49] (FANNY) through an approach which is based on directly converting the original functional to a noise clustering functional.

Another objective of this paper is to address issues regarding the restrictions imposed on the dissimilarity relation R_{ij} in order to derive different algorithms. Original

RFCM [42] requires that the relation R_{ij} must be derived from Euclidean distance between two objects, and hence is a more restrictive assumption regarding R_{ij}, as compared to the constraints imposed by methods of Roubens [63], Windham [76], and Kaufman and Rousseeuw [49]. The latter group only requires that,

$$R_{ij} \geq 0, R_{ii} = 0, \text{ and } R_{ij} = R_{ji}. \tag{5.1}$$

The restriction in RFCM occurs due to the algorithm being a relational dual of FCM. Later, Bezdek etal. [7] compared the performance of Windham's [76] assignment prototype (AP) algorithm with the RFCM to show that while AP has a less restrictive set of assumptions than RFCM, its performance on several synthetic examples that include non-Euclidean distance measures is not any better than (in fact worse, in some cases) RFCM. To overcome the restriction regarding the Euclidean metric, Hathaway and Bezdek [40] proposed NERFCM (non-Euclidean relational fuzzy c-means) clustering technique. In NERFCM, the original RFCM algorithm is modified by adding a step that involves an innovative technique to expand the original non-Euclidean relational data to make it Euclidean. Here we show that an algorithm can be constructed in a manner similar to FANNY that does not require any restrictions besides (5.1) to handle the RFCM functional, and the resulting algorithm can be applied to non-Euclidean relational data in a computationally more efficient manner than NERFCM. This new algorithm is simply named the Fuzzy Relational Clustering (FRC) algorithm. Both ordinary and robust versions of this algorithm are proposed.

Ref. [27] describes the current work, is presently under peer review for publication. In what follows, in section 5.2, the terminology and definitions for clustering relational data are presented, followed by the comparison of popular objective functions of fuzzy relational classification methods. This section also describes Roubens, AP, RFCM, and FANNY algorithms. Through generalization of FANNY, a new relational clustering algorithm (FRC) is also introduced. Section 5.3, the noise clustering concept [18] is described, deals with the extension of Roubens, AP, RFCM and FRC techniques to noise clustering. FRC is generalized using the noise clustering concepts to develop a Robust version of Fuzzy Relational Clustering algorithm (R-FRC). Qualitative equivalence of all these algorithms with noise fuzzy c-means (NFCM) [18] is considered to show comparison of noise distance for relational clustering and object clustering. In section 5.4, the derivation of R-FRC includes inequality constraint for the fuzzy memberships, and by using Kuhn-Tucker conditions, the resulting algorithm is shown to systematically handle non-Euclidean data. This version, called Noise Resistant, non-Euclidean Fuzzy Relational data Clustering (NR-NE-FRC), is the most general form of relational clustering algorithm presented in this research. In section 5.5, several examples are presented to demonstrate the differences and similarities of different approaches, followed by the conclusions of this chapter in section 5.6.

5.2 Fuzzy Relational Clustering Methods

Similarity (or dissimilarity) data are usually found in social sciences, marketing, and management information systems. Vinod [73] may have been the first to introduce a non-hierarchical clustering method based on an optimization model to classify inter-point distances and dissimilarity data. The objective was to minimize total dissimilarity amongst all objects and their corresponding most representative objects. This idea was further discussed by Rao [63], and Malvey and Cowder [61]. The optimization model of Vinod [73] can be stated as below.

$$\text{minimize} \quad \sum_{j=1}^{n}\sum_{k=1}^{n} z_{jk} R_{jk} \tag{5.2}$$

subject to

$$\sum_{j=1}^{n} z_{jk} = 1, \quad k = 1,2,....,n \tag{5.3}$$

$$z_{jk} \leq y_j, \quad j, k = 1,2,....,n \tag{5.4}$$

$$\sum_{j=1}^{n} y_j = c, \quad c = \text{number of clusters} \tag{5.5}$$

$$z_{jk}, y_j \in \{0,1\}, \quad j,\ k = 1,2,....,n \tag{5.6}$$

In the above, for a given k, only one z_{jk} is equal to 1 and all others are zero, so along with constraint in (5.4), an object k can only be assigned to an object j if object j has been selected as a representative object. For j to be a representative object, y_j must be one, and only c of n objects can be representative objects, as imposed by (5.5). Thus the minimization of (5.2) using the constraints (5.3) through (5.6) can generate a c-partition of the objects, where exactly c objects are chosen as prototypical objects. There are two types of "crisp" (or "hard") memberships in the above, designated through z_{jk} and y_j. This algorithm is comparable to the approach by Windham [76] as shown later.

Fuzzy algorithms to handle relational data include methods of Ruspini [67] and Roubens [63]. According to Windham [76], and Bezdek [6] criteria on which Ruspini's algorithms are based are considered difficult to interpret, and numerically complex, and although simpler, Roubens numerical procedure is unstable. Based on these observations, Windham [76] proposed a functional, that resembles like a slightly modified version of Roubens functional, and proposed a stable numerical scheme to derive the AP algorithm. The technique due to Roubens is described first, followed by Windham's AP algorithm.

$$F_R = \sum_{i=1}^{c}\sum_{j=1}^{n}\sum_{k=1}^{n} u_{ik}^2 u_{ij}^2 R_{jk} \tag{5.7}$$

subject to constraint

$$\sum_{i=1}^{c} u_{ik} = 1, \quad k = 1,2,\ldots,n \tag{5.8}$$

as well as the constraint from (5.1). The functional in equation (5.7) has "R" as the subscript denoting that it is Roubens algorithm (similar subscripts are used in subsequent functional equations). The minimization of F_R can be easily done through Lagrange multiplier approach, and an iterative procedure can be utilized to create a partition. The resulting algorithm is described in Roubens [63], and can be used to generate fuzzy c-partitions. Our limited experience with this algorithm, however, indicated instabilities in convergence, thus corroborating the observation of Windham [76].

The AP algorithm utilizes two types of memberships, assignment type, and prototype weight type. The functional to be minimized by AP algorithm is as below.

$$F_{AP} = \sum_{i=1}^{c}\sum_{j=1}^{n}\sum_{k=1}^{n} u_{ik}^2 v_{ij}^2 R_{jk} \tag{5.9}$$

subject to constraints

$$\sum_{i=1}^{c} u_{ik} = 1, \quad k = 1,2,\ldots,n \tag{5.10}$$

$$\sum_{j=1}^{n} v_{ij} = 1, \quad i = 1,2,\ldots,c \tag{5.11}$$

as well as the constraint from (5.1). As can be seen, this is similar to Roubens[63], but Windham [76] has two different types of memberships instead of one in Roubens. Although not easy to recognize, AP can be seen as a fuzzy extension of Vinod's method [73], by recognizing that y_j in Vinod's method are like the prototype weights in AP. In other words, only the objects that are prototypes have the weight 1 in Vinod's method, like weights v_{ij} in AP algorithm if the memberships in AP are hard. Moreover, the z_{jk} in Vinod's method are like u_{ik} in AP, because j in z_{jk} now stands for only those cases when the jth object is a prototype, thus making j equivalent to i in AP. It is clear that Windham [40, 76]) does not recognize that the AP algorithm is a fuzzy extension of Vinod's method. Another interesting observation that can be made is that since Vinod utilizes only hard memberships, his c prototypes are exactly c distinct objects, while due to the utilization of fuzzy memberships in AP, each of the c prototypes of AP is a weighted average of n prototypes as seen in the constraint (5.11).

Hathaway etal. [42] proposed the RFCM algorithm, based on the functional which is a further extension of Roubens functional in equation (5.7). This functional, shown below, originally appeared in Tucker [72], albeit in a different context.

$$F_{RFCM} = \sum_{i=1}^{c} \frac{\sum_{j=1}^{n}\sum_{k=1}^{n} u_{ik}^{m} u_{ij}^{m} R_{jk}}{2\sum_{t=1}^{n} u_{it}^{m}} \qquad (5.12)$$

where the fuzzifier exponent $m \geq 1$. A very important restriction of RFCM is that the dissimilarities R_{jk} must be derived from Euclidean distances between the objects in the object space as

$$R_{jk} = d_{jk}^2 = \left\| x_j - x_k \right\|^2 \text{ for } j, k = 1, \ldots\ldots, n \tag{5.13}$$

As can be seen, apart from this restriction and the use of fuzzifier m, the functional in (5.12) is a normalized version of the Roubens functional (equation (5.7)), through dividing by twice the fuzzy cardinality of each cluster. It can be noticed that (5.13) guarantees that R_{jk} satisfy (5.1), but the restriction in (5.13) is more severe than in (5.1). It is noted that the constraint on the memberships from (5.10) also applies here, and the additional membership constraint shown below also applies.

$$u_{ik} \geq 0, \quad i = 1,2,\ldots\ldots,c; \quad k = 1,2,\ldots,n \tag{5.14}$$

Tucker [72] derived (5.12) from the FCM functional, and thus established equivalence of two algorithms. This equivalence could provide the directions for deriving the minimization algorithm for RFCM. In fact, the algorithm for RFCM is developed through arguments that cast it as a relational dual of FCM. Since the new membership u_{ik} in the fixed point iteration scheme for FCM is computed through d_{ik}, the distance of objects from cluster prototypes. Hathaway etal. propose the following equations to compute d_{ik}, from memberships and the relational data. The first equation is,

$$V_i = (u_{i1}, u_{i2}, \ldots\ldots, u_{in})^T \Big/ \sum_{k=1}^{n} u_{ik} \tag{5.15}$$

where the V_i represents a mean (i.e. averaged) unit vector of memberships for the ith cluster. These are then used to obtain object to cluster distances, d_{ik}, as following.

$$d_{ik}^2 = (RV_i)_k - (V_i^T R V_i)/2 \quad \text{with} \quad R = [R_{jk}] = [d_{jk}^2], \tag{5.16}$$

In the above, j and k are two objects, and index i represents the ith cluster. It is emphasized that (5.15) and (5.16) are valid only when the relational data comes from

Euclidean measure in (5.13). Now since d_{ik}, the distance of objects from cluster prototypes, are available, one can use standard membership equation from FCM algorithm. Therefore, the new memberships are,

$$u_{ik} = \frac{\left(\frac{1}{d_{ik}^2}\right)^{1/(m-1)}}{\sum_{w=1}^{c}\left(\frac{1}{d_{wk}^2}\right)^{1/(m-1)}} \tag{5.17}$$

Thus the RFCM algorithm is very similar to FCM algorithm, and thus inherits all its well-established properties. The only problem remaining is that the constraint on memberships from (5.14) is not explicitly satisfied, and unlike FCM. the u_{ik} from (5.17) are not guaranteed to be non-negative because d_{ik} from (5.16) may not be always non-negative. However, if constraint from (5.13) is satisfied, then u_{ik} will be always non-negative because it can be shown that it is the square of Euclidean distance from prototype to the object. The RFCM algorithm, when $m > 1$, is presented below.

RFCM Algorithm

1. For relational data satisfying (5.13), fix c, $2 \le c \le n$, and $m > 1$, and initialize fuzzy c-partition, u_{ik}.

2. Compute c mean vectors, V_i, from (5.15) and then compute distances, d_{ik}, from (5.16)

3. Update memberships, u_{ik}, from (5.17)

4. Check for convergence using some convenient norm on u_{ik} and if converged stop, else go to step 2.

One can notice a strong resemblance of the above with the FCM algorithm.

Next the FANNY (Fuzzy Analysis) algorithm of Kaufman and Rousseeuw [49] is considered. The original FANNY technique starts with essentially the same functional as RFCM in (5.12), except that it has a fixed value of fuzzifier $m = 2$. Moreover, FANNY does not impose (5.13) on R_{jk} but has the usual restriction of (1). The FANNY functional is similar to (5.12),

$$F_{FANNY} = \sum_{i=1}^{c} \frac{\sum_{j=1}^{n} \sum_{k=1}^{n} u_{ik}^2 u_{ij}^2 d(j,k)}{2\sum_{t=1}^{n} u_{it}^2} \tag{5.18}$$

with the membership constraints from (8) and (14). In the above, $d(j, k)$ is the distance or dissimilarity between objects j and k, and it is usually implied to be the L_1 distance. The reader is referred to [49] for details of derivation of an algorithm that is based on application of Lagrange multiplier and Kuhn-Tucker conditions to directly minimize (5.18) subject to the constraints (5.8) and (5.14).

If the fuzzifier exponent m, is used in (18) along with R_{jk} as to denote any dissimilarity measure, then one obtains a functional shown below that looks exactly same as RFCM functional in (5.12).

$$F_{FRC} = \sum_{i=1}^{c} \frac{\sum_{j=1}^{n} \sum_{k=1}^{n} u_{ik}^m u_{ij}^m R_{jk}}{2\sum_{t=1}^{n} u_{it}^m} \tag{5.19}$$

In the above, the subscript *FRC* stands for Fuzzy Relational Clustering, which is an extension of FANNY technique. To reiterate, the difference between the two are; (a) the fuzzifier m, which makes the fuzzy memberships more general, and (b) the use of R_{jk} instead of $d(j, k)$ in the original FANNY [49], implying that while the relational data in FANNY usually comes from L_1 norm, in *FRC* it could be from any dissimilarity measure. The difference due to the use of the fuzzifier m becomes an important issue when *FRC* is made robust using the concept of noise clustering [18] in section 5. Thus hereafter, the version in (5.19) is referred to as *FRC*. To derive the necessary conditions for the minimization of (5.19), a Lagrangian is constructed based on the constraint (5.8), while the inequality constraint in (5.14) is ignored with a hope that it may be automatically satisfied. This treatment is similar to the derivation of original FCM algorithm, where the inequality constraint was not directly included in the optimization problem. Thus this derivation is different from the exact derivation in Kaufman and Rousseeuw [49]

$$L = \sum_{i=1}^{c} \frac{\sum_{j=1}^{n} \sum_{k=1}^{n} u_{ik}^{m} u_{ij}^{m} R_{jk}}{2 \sum_{j=1}^{n} u_{ij}^{m}} - \sum_{k=1}^{n} \lambda_k \left(\sum_{i=1}^{c} u_{ik} - 1 \right) \qquad (5.20)$$

The above can be minimized with respect to u_{ik} and through eliminating the Lagrange multipliers λ_k, one can obtain the following for the memberships.

$$u_{ik} = \frac{\left(\frac{1}{a_{ik}} \right)^{1/(m-1)}}{\sum_{w=1}^{c} \left(\frac{1}{a_{wk}} \right)^{1/(m-1)}} \qquad (5.21)$$

where the terms a_{ik} are given by,

$$a_{ik} = \frac{m \sum_{j=1}^{n} u_{ij}^{m} R_{jk}}{\sum_{j=1}^{n} u_{ij}^{m}} - \frac{m \sum_{h=1}^{n} \sum_{j=1}^{n} u_{ij}^{m} u_{ih}^{m} R_{jh}}{2 \left(\sum_{j=1}^{n} u_{ij}^{m} \right)^{2}} \qquad (5.22)$$

Thus by direct application of Lagrange multiplier technique to derive constrained minimization of (5.19), we obtain the solution for the c-partition from (5.21) and (5.22). It is noted that in deriving the above, the only constraint on R_{jk} has been (5.1). Thus this derivation has an advantage over the derivation in RFCM. A few observations regarding (5.21) and (5.22) are in order. First, equation (5.21) is a transcendental equation in u_{ik}, and second, the constraint (5.14) is not explicitly satisfied. To solve for u_{ik} from (5.21), one can use a gradient descent technique such as Newton's method, or simply use a successive substitution method, in which one can repeatedly use old values of u_{ik} in (5.22) to obtain a_{ik} and then solve for new values of u_{ik} from (5.21) till convergence. In practice, one can improve the order of convergence of this method by using the Seidel iteration scheme, where in solving for a_{ik} one utilizes all the new available membership values. In other words, when computing the a_{ik}, the membership values u_{ij} when $j < k$ are all newly computed (or from current iteration), while for $j \geq k$ they are old (or from previous iteration) values. This can be illustrated in the following algorithm for FRC, which is similar to FANNY algorithm.

FRC Algorithm:

1. For relational data satisfying (5.1), fix c, $2 \leq c \leq n$, and $m > 1$, and initialize fuzzy c-partition, u_{ik}. Initiate a counter $p = 0$

2. Compute for each $k = 1, \ldots.., n$,

   a) Compute for each $i = 1, \ldots.., c$: a_{ik} from equation (5.22), using memberships ${}^{(p+1)}u_{ij}$ for $j < k$ and ${}^{(p)}u_{ij}$ for $j \geq k$. (here the pre-superscript denotes iteration number).

b) Compute membership ${}^{(p+1)}u_{ik}$ using (5.21).

3. Check for convergence using some convenient norm on u_{ik} and if converged stop, else set $p = p + 1$, and go to step 2.

As mentioned before, there is no guarantee that constraint (5.14) will be satisfied as a result of the above algorithm. In fact, when any of the a_{ik} becomes negative, then a corresponding u_{ik} also becomes negative. A close examination of (5.22) requires to determine the conditions under which a_{ik} are non-negative. In fact, one can rewrite (5.22) as

$$\frac{a_{ik}}{m} = (RV_i)_k - (V_i^T RV_i)/2 \tag{5.23}$$

This reveals that (5.22) is indeed comparable to the right hand side of (5.16) in derivation of RFCM. Hence the equations (5.21) and (5.17) are also equivalent, as the factor m will drop out in (5.21). It is noted that in rewriting (5.22) as (5.23), no further assumptions are necessary. Therefore, this result points out that although the condition (5.13) was required in derivation of RFCM, the actual algorithm may not be as restrictive, since the same equations can be also obtained without requiring (5.13) as in FRC derivation shown here. This may explain why RFCM works for many non-Euclidean examples as reported in Bezdek etal [7]. We will come back to the results from Bezdek etal. [7] later in this chapter. When the relational data is derived from Euclidean distance as in (5.13), then (5.23) indicates that a_{ik} are indeed related to the Euclidean distance, because now,

$$a_{ik} = m(d_{ik})^2 \tag{5.24}$$

hence, for FRC, if the relational data is Euclidean, it will automatically satisfy the constraint (5.14) that the memberships are positive. However, when the relational data is non-Euclidean, neither RFCM nor FRC will automatically satisfy (5.14). The only way to make sure that (5.14) is satisfied, one must employ that constraint also in the minimization procedure. However, this form of FRC algorithm is derived here to (a) obtain a simple relational clustering algorithm that is based on first principles of optimization, and (5.2) explain observations in Bezdek etal [7] regarding why RFCM worked for many non-Euclidean examples. Later (5.14) will be also included in derivation for the robust version of the FRC in section 5.5. The first step towards robustification of relational clustering comes through consideration of noise clustering technique, that is described in next section.

5.3 Noise Clustering Applied to Relational Clustering Techniques

The equivalence of RFCM and FCM algorithms indicate that RFCM inherits the major deficiency of FCM *e.g.* poor performance in presence of noise and outliers. Similarly, FANNY and the FRC presented here are not highly robust. In FANNY, the authors claim that since the relational data can be derived from L_1 norm, it will be less susceptible against evenly distributed random noise. This may not be true for a general case of relational data, including the relational data derived from Euclidean measure. Thus all the relational clustering algorithms described in section 5.2 will not be robust against noisy data. The main reason for this is that all these algorithms must create a c-partition of the data, and thus must even include noisy data in this c-partition. Therefore, the noisy data may bias the resulting partition. Davé's noise clustering (NC) [18] was proposed to alleviate this major drawback of such partitioning algorithms by creating a separate noise class. The idea of having a noise class is not new, but the concept of noise as a prototype, and how to determine distance of an object from noise prototype was a novel idea proposed in [18]. In this section, this concept is briefly described.

The concept of noise clustering works well for object data clustering methods such as FCM and FCS (fuzzy c-shells clustering) [15], as the definition of the noise distance δ has a direct physical meaning. In the object data clustering, there are object prototypes, and hence there is also a noise prototype. The extension of noise clustering to relational data clustering techniques is not obvious, because in a strict sense, there are no cluster (and hence noise) prototypes in relational clustering, and there is only a need to generate a partition, need for detecting cluster prototypes is not explicit. Since Hathaway etal. [42] developed their relational clustering method (RFCM) as a relational dual of FCM, robustification of RFCM though use of noise clustering is somewhat easy. However, such extension for other techniques is not that obvious. In this section, robustification of all the relational clustering methods discussed in section 5.2 is considered. First the Roubens algorithm [63] is considered. Although the Roubens algorithm is found to be unstable and thus not useful, it is considered for this exercise, because the extension of Roubens algorithm also paves way for the extension of RFCM, FANNY, and FRC techniques due to the similarities in the memberships. Considering the Roubens functional from (5.7), one needs to add a noise class, thus making the number of clusters to become $c + 1$. When this is done, the new functional becomes,

$$ {}^{N}F_{R} = \sum_{i=1}^{c+1} \sum_{j=1}^{n} \sum_{k=1}^{n} u_{ik}^{2} u_{ij}^{2} R_{jk} \tag{5.25} $$

In above equation, the pre-superscript "N" denotes extension to noise clustering. In (5.25), it is not obvious how to introduce the noise distance, since the noise distance in the original paper is defined as a distance from the noise prototype to the object data point. In relational data, since there is no explicit object data available, one must modify the definition of the noise distance. For this purpose, (5.25) is rewritten as follows.

$$ {}^{N}F_{R} = \sum_{i=1}^{c} \sum_{j=1}^{n} \sum_{k=1}^{n} u_{ik}^{2} u_{ij}^{2} (R_{jk})_{i} + \sum_{j=1}^{n} \sum_{k=1}^{n} u_{*k}^{2} u_{*j}^{2} (R_{jk})_{*} \tag{5.26} $$

In the above, the first term on RHS (right hand side) is same as the original Roubens functional, while the second term is the extra term due to the extension to noise clustering. Another modification here is the extra subscript to the dissimilarity distance $(R_{jk})_i$ denoting that this is the "amount" of dissimilarity between objects j and k as viewed by class i. Under ordinary circumstances, the dissimilarity should be independent of the class, thus $R_{jk} = (R_{jk})_i$ for all i. However, when introducing the noise class, we must make a distinction that it is a special class and it imposes its own bias (or lack there of) to determine the "amount" of dissimilarity. Then analogous to the original noise clustering, we specify that the noise class views all dissimilarities as equal. Thus, $(R_{jk})_i = \delta$, the dissimilarity noise distance. This noise distance, δ can be the same for all cases, or similar to the generalized noise clustering [26], it could take different values for different pairs of points as well as clusters. In this chapter, we restrict this to be a constant value, and thus (26) is written as,

$$ {}^{N}F_{R} = \sum_{i=1}^{c}\sum_{j=1}^{n}\sum_{k=1}^{n} u_{ik}^{2} u_{ij}^{2} R_{jk} + \sum_{j=1}^{n}\sum_{k=1}^{n} u_{*k}^{2} u_{*j}^{2} \delta \qquad (5.27) $$

To understand the physical meaning of the dissimilarity noise distance, δ, in the above equation, let us consider the following. If all dissimilarities R_{hk} are greater than the dissimilarity noise distance, δ, then in order to minimize the objective functional, the membership of object h in the noise class must be made higher, and thus all other u_{ih} would be made very small, and consequently the multipliers to larger R_{hk} values will be much smaller as compared to the multiplier for the dissimilarity noise distance, δ. Thus the hth object is classified as an outlier. However, a more restrictive interpretation for the dissimilarity noise distance, δ would be that within a given cluster, none of the object pairs will have a dissimilarity larger than the dissimilarity noise distance, δ. Based on this interpretation, it is clear that if there exists an object h in the data-set such that its dissimilarity with even one object in every cluster is larger than the dissimilarity noise distance, δ, then it will be classified as an outlier. On the other hand, if for object h in the data-set, its dissimilarity with at least one object of any one *good* cluster is less than dissimilarity noise distance, δ, then it will belong to that cluster, and not to the noise class. This interpretation is similar to object data noise distance, where the distances (or dissimilarities) are measured always from the cluster prototype and not from cluster members. Thus for object data clustering, if a given object is farther to all the cluster centers as compared to object noise distance, then that object is classified as a noise point, and on the other hand, if an object is closer than the object noise distance to any one cluster prototype then it will belong to that cluster, and not to the noise class.

The algorithm to solve for the partition based on the above equations can be derived to obtain the following:

5.3.1 Robust Version of Roubens (R-Roubens) Algorithm

(1) For relational data satisfying (5.1), fix c, $2 \leq c \leq n$, and initialize fuzzy (c+1)-partition, u_{ik}. Select noise distance, $\delta > 0$.

(2) Compute terms D_{ik} defined as below

$$D_{ik} = \sum_{j=1}^{n} u_{ij}^2 R_{jk} \tag{5.28}$$

and the noise term,

$$D_{*k} = \sum_{j=1}^{n} u_{*j}^2 \delta = \delta N_{nc}\,, \tag{5.29}$$

where N_{nc} is the equivalent fuzzy clardinality of noise class, i.e.

$$N_{nc} = \sum_{j=1}^{n} u_{*j}^2\,. \tag{5.30}$$

Note that all these terms are ≥ 0.

(3) Compute memberships by solving the new minimization problem that resembles the original noise clustering for FCM formulation:

$$\min_{u_{ik}} \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^2 D_{ik} + \sum_{k=1}^{n} u_{*k}^2 D_{*k} \tag{5.31}$$

to obtain the memberships as

$$u_{ik} = \frac{\left(\dfrac{1}{D_{ik}}\right)}{\displaystyle\sum_{j=1}^{c}\left(\frac{1}{D_{jk}}\right) + \left(\frac{1}{\delta N_{nc}}\right)}\,. \tag{5.32}$$

(4) Check for convergence using some convenient norm on u_{ik} and if converged stop, else go to step

In the above, it is easy to see how the noise distance appears in the solution procedure. It is noted that the term D_{*k} in (5.29) is a product of the noise distance δ and the equivalent fuzzy cardinality of noise class defined in (5.30). In step 3, one can also easily compute the membership in the noise class as shown in (5.32).

Now one can proceed with other relational clustering methods. Since the functional of RFCM is basically a normalized version of Roubens functional, it can be extended to noise clustering in a similar way as follows:

$$ {}^{N}F_{RFCM} = \sum_{i=1}^{c} \frac{\sum_{j=1}^{n}\sum_{k=1}^{n} u_{ik}^{m} u_{ij}^{m} R_{jk}}{2\sum_{t=1}^{n} u_{it}^{m}} + \frac{\sum_{j=1}^{n}\sum_{k=1}^{n} u_{*k}^{m} u_{*j}^{m} \delta}{2\sum_{t=1}^{n} u_{*t}^{m}} \tag{5.33} $$

The equation (5.15) for the membership vector can still be used for computing the first c vectors V_i, and similarly, the equation (5.16) can be used to compute the object to class distances for the first c classes. However, the noise membership vector (membership of objects to the noise class) is computed as below.

$$ V_* = (u_{*1}, u_{*2}, \dots\dots, u_{*n})^T \Big/ \sum_{k=1}^{n} u_{*k} \tag{5.34} $$

These are used to obtain object to noise cluster distances, d_{*k}, as follows using (5.16).

$$ d_{*k}^2 = (RV_*)_k - (V_*^T RV_*)/2 \quad \text{with} \quad R = [R_{jk}]_* = \delta, \tag{5.35} $$

In the above, j and k are two objects, and index * represents the noise cluster. Although it may not be apparent, in this noise cluster extension. the dissimilarity distance $(R_{jk})_i$ is viewed differently by each class, thus the dissimilarity distances in (5.35) are all same as

δ because this equation is written specifically for the noise class. Knowing that the expansion of (5.16) is related to (5.22) as shown in (5.23), using (5.35), we obtain that the object to noise class distance is directly related to δ as

$$d_{*k}^2 = \delta / 2, \tag{5.36}$$

It is clear that (5.34) is not even required for evaluating object to noise cluster distances, d_{*k}, as those terms drop out from (5.35). Subsequently, (5.17) for memberships can be modified to obtain memberships in good classes as

$$u_{ik} = \frac{\left(\dfrac{1}{d_{ik}^2}\right)^{1/(m-1)}}{\sum\limits_{w=1}^{c}\left(\dfrac{1}{d_{wk}^2}\right)^{1/(m-1)} + \left(\dfrac{2}{\delta}\right)^{1/(m-1)}} \tag{5.37}$$

and the memberships in the noise class is

$$u_{*k} = \frac{\left(\dfrac{2}{\delta}\right)^{1/(m-1)}}{\sum\limits_{w=1}^{c}\left(\dfrac{1}{d_{wk}^2}\right)^{1/(m-1)} + \left(\dfrac{2}{\delta}\right)^{1/(m-1)}} \tag{5.38}$$

Thus the robust version of RFCM algorithm can be written down using above equations.

5.3.2 Robust RFCM (R-RFCM) Algorithm

(1) For relational data satisfying (5.13), fix c, $2 \le c \le n$, and $m > 1$, and initialize fuzzy $(c+1)$ -partition, u_{ik} Select noise distance δ, and compute object to noise cluster distances, d_{*k}, from (5.36).

(2) Compute c mean vectors, V_i, from (5.15), then compute distances, d_{ik}, from (5.16).

(3) Update membership, u_{ik}, from (5.37) the noise membership (if require) from (5.38).

(4) Check for convergence using some convenient norm on u_{ik} and if converged stop, else go to step 2.

A careful observation of the above algorithm shows that the only major difference between this and the original RFCM is in equation (5.37). It may be seen that one does not require explicit computation of noise memberships, and thus there is only a minor modification necessary from the old algorithm to new one, which is in terms of changing (5.17) to (5.37). This indicates simplicity of this approach. More understanding for the physical meaning of the noise distance for the relational data (as compared to the object data) can be made from the results shown in (5.37). Comparing (5.37) with (2.4), it is seen that the noise distance for the relational data would be twice that of noise distance for object data, if the relational data is computed as Euclidean measure of distance between two objects. This can be illustrated in Figure 5.1, where for the sake of convenience, a single round cluster (of two-dimensional object data) is shown, along with an outlier, just outside the cluster. Since the outlier is at a distance larger than δ (radius of the circle) it does not belong to the good cluster, and will be classified as a noise point by use of NC-FCM algorithm (see equation (2.4)). However, if one were to use NC-RFCM to do this classification, the largest dissimilarity of this object from any good object must be less than the diameter of the circle, i.e. twice the radius of the circle. Thus as shown in Figure 5.3, the noise object (i.e. outlier) is at a distance greater than twice the radius from object A, which is the farthest object from outlier in Euclidean sense.

Next, the noise clustering is incorporated into Windham's AP algorithm. Since AP requires two types of memberships (or weights), the formulation becomes somewhat more complex. Let u_{*k} be the membership assignment of point k in noise class, and v_{*k} be the noise prototype weight of object k. These quantities are constrained through the following equations.

$$u_{*k} = 1 - \sum_{i=1}^{c} u_{ik}, \quad k = 1,2,\ldots,n \tag{5.39}$$

$$\sum_{j=1}^{n} v_{*j} = 1 \tag{5.40}$$

Now the functional is modified in the usual fashion where the extra subscript to the dissimilarity distance $(R_{jk})_i$ denotes that this is the "amount" of dissimilarity between objects j and k as viewed by class i.

$${}^{N}F_{AP} = \sum_{i=1}^{c}\sum_{j=1}^{n}\sum_{k=1}^{n} u_{ij}^{2} v_{ik}^{2} (R_{jk})_{i} + \sum_{j=1}^{n}\sum_{k=1}^{n} u_{*j}^{2} v_{*k}^{2} (R_{jk})_{*} \tag{5.41}$$

In the above, inter-point dissimilarity viewed by the noise class is the constant noise dissimilarity, δ, and hence the functional becomes,

$${}^{N}F_{AP} = \sum_{i=1}^{c}\sum_{j=1}^{n}\sum_{k=1}^{n} u_{ij}^{2} v_{ik}^{2} (R_{jk})_{i} + \sum_{j=1}^{n}\sum_{k=1}^{n} u_{*j}^{2} v_{*k}^{2} \delta \tag{5.42}$$

The algorithm to solve for the partition based on the above is much more involved as compared to Roubens algorithm, and can be derived to obtain the following:

5.3.3 Robust Version of AP (R-AP) Algorithm

(1) For relational data satisfying (5.1), fix c, $2 \le c \le n$, and initialize fuzzy $(c+1)$-partition, u_{ik}. Select noise distance, $\delta > 0$.

(2) Compute terms D_{ik} defined as in (5.28) and the noise term as in (5.29), along with the equivalent fuzzy cardinality of noise class as in (5.30). Note that all these terms are ≥ 0.

(3) Compute new prototype weights by solving the minimization problem that resembles the original noise clustering for FCM formulation with a small modification due to the constraint (5.11):

$$\min_{v_{ik}} \sum_{i=1}^{c}\sum_{k=1}^{n} v_{ik}^2 D_{ik} + \sum_{k=1}^{n} v_{*k}^2 \delta N_{nc} \tag{5.43}$$

to obtain the prototype weights for regular clusters as

$$v_{ik} = \frac{\left(\frac{1}{D_{ik}}\right)}{\sum_{j=1}^{n}\left(\frac{1}{D_{ij}}\right)} \tag{5.44}$$

and the noise class as

$$v_{*k} = \frac{1}{n} \tag{5.45}$$

(4) Now that the prototype weights are determined, compute the class memberships by first computing terms B_{ik} defined as below

$$B_{ik} = \sum_{j=1}^{n} v_{ij}^2 R_{jk} \tag{46a}$$

and the noise term through using the result from (5.45),

$$B_{*k} = \sum_{j=1}^{n} v_{*j}^2 \delta = \delta \tag{46b}$$

Note that all these terms are ≥ 0.

(5) Using results from above, compute memberships by solving the new minimization problem, along with constraint (5.39), that resembles the original noise clustering for FCM formulation:

$$\min_{u_{ik}} \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{2} B_{ik} + \sum_{k=1}^{n} u_{*k}^{2} B_{*k} \tag{5.47}$$

to obtain the memberships as

$$u_{ik} = \frac{\left(\dfrac{1}{B_{ik}}\right)}{\sum_{j=1}^{c}\left(\dfrac{1}{B_{jk}}\right) + \left(\dfrac{1}{\delta}\right)} \tag{5.48}$$

(6) Check for convergence using some convenient norm on u_{ik} and if converged stop, else go to step 2.

The results obtained above for noise clustering version of AP algorithm further show the power of the NC approach. As seen in (5.45), the prototype weights for the noise class are all equal to value $(1/n)$ indicating that all the points have equal probability to be a noise prototype, and thus when the noise distance is a constant value, then the noise class is truly an universal entity. In other words, the noise class is spread throughout the space, and each object data contributes equally towards the noise prototype. The results for the class memberships, u_{ik}, are similar to the NC-FCM type algorithm, as shown in (5.48), where the AP dissimilarity noise distance δ is directly equivalent to the object noise distance in NC algorithms. Next, let us reconsider the functional for FRC, with one extra cluster for noise class. Using the same treatment for the dissimilarity distance by adding the extra subscript to $(R_{jk})_i$ and then incorporating the noise distance, we obtain,

$$^{N}F_{FRC}=\sum_{i=1}^{c}\frac{\sum_{j=1}^{n}\sum_{k=1}^{n}u_{ik}^{m}u_{ij}^{m}R_{jk}}{2\sum_{t=1}^{n}u_{it}^{m}}+\frac{\sum_{j=1}^{n}\sum_{k=1}^{n}u_{*k}^{m}u_{*j}^{m}\delta}{2\sum_{t=1}^{n}u_{*t}^{m}} \tag{5.49}$$

To derive the necessary conditions for the minimization of (5.49), a Lagrangian is constructed based only on the constraint (5.8)

$$L=\sum_{i=1}^{c}\frac{\sum_{j=1}^{n}\sum_{k=1}^{n}u_{ik}^{m}u_{ij}^{m}R_{jk}}{2\sum_{t=1}^{n}u_{it}^{m}}+\frac{\sum_{j=1}^{n}\sum_{k=1}^{n}u_{*k}^{m}u_{*j}^{m}\delta}{2\sum_{t=1}^{n}u_{*t}^{m}}-\sum_{k=1}^{n}\lambda_{k}\left(\sum_{j=1}^{c+1}u_{jk}-1\right) \tag{5.50}$$

From the above, equation for membership can be obtained just as for robust RFCM as,

$$u_{ik}=\frac{\left(\frac{1}{a_{ik}}\right)^{1/(m-1)}}{\sum_{w=1}^{c}\left(\frac{1}{a_{wk}}\right)^{1/(m-1)}+\left(\frac{2}{m\,\delta}\right)^{1/(m-1)}} \tag{5.51}$$

where the terms a_{ik} are given by,

$$a_{ik}=\frac{m\sum_{j=1}^{n}u_{ij}^{m}R_{jk}}{\sum_{j=1}^{n}u_{ij}^{m}}-\frac{m\sum_{h=1}^{n}\sum_{j=1}^{n}u_{ij}^{m}u_{ih}^{m}R_{jh}}{2\left(\sum_{j=1}^{n}u_{ij}^{m}\right)^{2}} \tag{5.52}$$

In a manner similar to how the FRC was derived, the algorithm for robust version of FRC (R-FRC) is presented below.

5.3.4 Robust Version of FRC (R-FRC) Algorithm

(1) For relational data satisfying (5.1), fix c, $2 \le c \le n$, and $m > 1$, and initialize fuzzy $(c+1)$-partition, u_{ik}. Initiate a counter $p = 0$. Select noise distance, $\delta > 0$.

(2) Compute for each $k = 1, \ldots\ldots, n$,

a) Compute for each $i = 1, \ldots\ldots, c$: a_{ik} from equation (5.52), using memberships ${}^{(p+1)}u_{ij}$ for $j < k$ and ${}^{(p)}u_{ij}$ for $j \geq k$. (here the pre-superscript denotes iteration number)

b) Compute membership ${}^{(p+1)}u_{ik}$ using (5.51)

(3) Check for convergence using some convenient norm on u_{ik} and if converged stop, else set $p = p + 1$, and go to step 2.

The above algorithm is very similar to the FRC algorithm, except for the minor differences in the equations. In this section, conventional relational clustering algorithms are extended to noise clustering. The results obtained for doing that show consistency of the NC approach. In the next section, the FRC algorithm is considered again for extension to noise clustering. However, while doing that, the issue of the membership constraint from equation (5.14) is also explicitly handled in objective function minimization.

5.4 Robust Non-Euclidean FRC Algorithm

The functional for FRC, with one extra cluster for noise class is considered again, as shown in (5.49). To derive the necessary conditions for the minimization of (5.49), a Lagrangian is constructed based on the constraint (5.8) as well as the inequality constraint in (5.14).

$$L = \sum_{i=1}^{c} \frac{\sum_{j=1}^{n}\sum_{k=1}^{n} u_{ik}^{m} u_{ij}^{m} R_{jk}}{2\sum_{t=1}^{n} u_{it}^{m}} + \frac{\sum_{j=1}^{n}\sum_{k=1}^{n} u_{*k}^{m} u_{*j}^{m} \delta}{2\sum_{t=1}^{n} u_{*t}^{m}} - \sum_{k=1}^{n} \lambda_k \left(\sum_{j=1}^{c+1} u_{jk} - 1 \right) - \sum_{j=1}^{n}\sum_{i=1}^{c+1} \psi_{ij} u_{jij} \qquad (5.53)$$

In the above, λ_k and ψ_{ij} are the Lagrange multipliers. The above can be minimized with respect to u_{ik} to obtain

$$\frac{\partial L}{\partial u_{ij}} = \frac{m u_{ij}^{m-1} \sum_{k=1}^{n} u_{ik}^{m} R_{ik}}{\sum_{k=1}^{n} u_{ik}^{m}} - \left(\frac{m}{2} \right) \frac{u_{ij}^{m-1} \sum_{k=1}^{n} \sum_{l=1}^{n} u_{ik}^{m} u_{lk}^{m} R_{kl}}{\left(\sum_{k=1}^{n} u_{ik}^{m} \right)^{2}} - \lambda_j - \psi_{ij} = 0 \qquad (5.54)$$

and then the corresponding Kuhn-Tucker conditions are as follows.

$$\psi_{ij} \geq 0 \qquad (5.55)$$

$$\frac{\partial L}{\partial u_{ij}} = 0 \qquad (5.56)$$

$$u_{ij} \psi_{ij} = 0 \qquad (5.57)$$

One can simplify (5.54) to write the following

$$a_{ij} u_{ij}^{m-1} - \lambda_j - \psi_{ij} = 0 \qquad (5.58)$$

where the terms a_{ij} are given by equation (5.22) as in the case of FRC. However, to eliminate Lagrange multipliers λ_j, it is more convenient to define another quantity as

$$b_{ij} = a_{ij} u_{ij}^{m-2} \qquad (5.59)$$

and then, one can obtain the following for the memberships.

$$u_{ij} = \frac{1/b_{ij}}{\sum_{w=1}^{c+1} 1/b_{wj}} + \frac{\psi_{ij}}{b_{ij}} - \frac{\sum_{w=1}^{c+1} \left(\psi_{wj} / b_{wj} \right)}{b_{iv} \sum_{w=1}^{c+1} \left(1/b_{wj} \right)} \qquad (5.60)$$

In the above equation, keep in mind that clusters sum to $c+1$, due to the extra noise cluster. Now the same type of arguments as in the derivation of FANNY [49] can be applied here to take care of the remaining Kuhn-Tucker conditions. First, observing (5.55), for each object j, we can have one of the following two conditions.

1. $\psi_{ij} = 0$ for all i, so that (5.57) becomes

$$u_{ij} = \frac{1/b_{ij}}{\sum_{w=1}^{c+1} 1/b_{wj}} \tag{5.61}$$

Considering that all memberships must be non-negative as per (5.14), the solution of (5.61) is valid for each object j, if the right hand side of (5.61) is also non-negative. If not, then (5.61) is not valid. In other words, one cannot accept the memberships from (5.61) if they turn out to be negative.

2. $\psi_{ij} > 0$ for some i, so according to (5.57), for those i, the membership will be zero.

$$u_{ij} = 0 = \frac{1/b_{ij}}{\sum_{w=1}^{c+1} 1/b_{wj}} + \frac{\psi_{ij}}{b_{ij}} - \frac{\sum_{w=1}^{c+1}\left(\psi_{wj}/b_{wj}\right)}{b_{ij}\sum_{w=1}^{c+1}\left(1/b_{wj}\right)} \tag{5.62}$$

However, the membership constraint from (5.8) dictates that (5.62) is not valid for all i for the same object j. Thus we can define the partition of clusters as:

$$I^- = \left\{i; u_{ij} = 0\right\} \tag{5.63}$$

$$I^+ = \left\{i; u_{ij} > 0 \Rightarrow \psi_{ij} = 0\right\} \neq \varnothing \tag{5.64}$$

Note that in order to satisfy (5.8), the set I^+ cannot be a null set. Assuming that the term b_{ij} is finite, for set defined by (5.63), we obtain from (5.62),

$$\psi_{ij} = \frac{\sum_{w=1}^{c+1}\left(\psi_{wj}/b_{wj}\right)}{\sum_{w=1}^{c+1}\left(1/b_{wj}\right)} - \frac{1}{\sum_{w=1}^{c+1} 1/b_{wj}} \text{ for } i \in I^{-} \tag{5.65}$$

Noting that the right-hand side of (5.65) is independent of i, after some manipulations (see Appendix C), one can obtain the following.

$$\psi_{ij} = -\frac{1}{\sum_{w \in I^{+}}\left(1/b_{wj}\right)} \text{ for } i \in I^{-} \tag{5.66}$$

For the set in (5.64), equation (5.60) becomes

$$u_{ij} = \frac{1/b_{ij}}{\sum_{w=1}^{c+1} 1/b_{wj}} - \frac{\sum_{w=1}^{c+1}\left(\psi_{wj}/b_{wj}\right)}{b_{ij}\sum_{w=1}^{c+1}\left(1/b_{wj}\right)} \tag{5.67}$$

and then using (5.66) in (5.67), after some manipulations (Appendix C), one can obtain the following.

$$u_{ij} = \frac{1/b_{ij}}{\sum_{w \in I^{+}}\left(1/b_{wj}\right)} \text{ for } i \in I^{+} \tag{5.68}$$

Based on the above equation, one obtains following conditions for local minima of (5.49).

$$u_{ij} = 0 \text{ for } i \in I^{-}; \text{ and } u_{ij} = \frac{1/b_{ij}}{\sum_{w \in I^{+}}\left(1/b_{wj}\right)} \text{ for } i \in I^{+} \tag{5.69}$$

where the cluster sets are defined as

$$I^{-} = \left\{ i;\ \frac{1/b_{ij}}{\sum\limits_{w=1}^{c+1}\left(1/b_{iw}\right)} \leq 0 \right\} \tag{5.70}$$

$$I^{+} = \left\{ i;\ \frac{1/b_{ij}}{\sum_{w=1}^{c+1}\left(1/b_{iw}\right)} > 0 \right\} \tag{5.71}$$

The above conditions do not take care of the situations when any b_{ij} is zero. However, in the limit, that makes the corresponding membership u_{ij} to go to 1. In all the above equations, the noise class is implicitly included and thus is taken care of.

Thus by direct application of Lagrange multiplier technique to derive constrained minimization of (5.49), we obtain the solution for the $c+1$-partition from (5.69) through (5.71). It is noted that in deriving the above, the only constraint on R_{jk} has been (5.8). Thus this derivation has an advantage over the derivation in RFCM. It should be clear that (5.69) can be used to find memberships in good clusters as well as noise class, while from (5.22), the quantity a_{*k} can be obtained by the following simplified equation.

$$a_{*k} = m\frac{\delta}{2} \tag{5.72}$$

from which one can obtain b_{*k} using (5.59).

It is clear that equation (5.69) is a transcendental equation in u_{ik}. To solve for u_{ik} from (5.69), one can use a gradient descent technique such as Newton's method, or simply use a successive substitution method. As in FRC, one can improve the order of convergence of this method by using the Seidel iteration scheme. This is shown in the following algorithm.

5.4.1 Noise Resistant, Non-Euclidean FRC (NR-NE-FRC) Method

(1) For relational data satisfying (5.1), fix c, $2 \le c \le n$, and $m > 1$, and initialize fuzzy c+1-partition, u_{ik}. Select value of δ. Initialize $I^- = I^+ = \varnothing$. Set a counter $p = 0$.

(2) Compute for each $k = 1,\dots, \text{n}$,

a) Compute for each $i = 1,\dots, c$: b_{ik} from equation (5.59), using memberships ${}^{(p+1)}u_{ij}$ for $j < k$ and ${}^{(p)}u_{ij}$ for $j \ge k$. (here the pre-superscript denotes iteration number)

b) Compute for each $i = 1,\dots, c$: the quantities

$$B_i = \frac{1/b_{ij}}{\sum_{w=1}^{c+1}\left(1/b_{iw}\right)} \tag{5.73}$$

and define the following cluster sets:

$$\text{if } B_i \le 0 \text{ then } I^- = I^- \cup \{i\}\text{, and if } B_i > 0 \text{ then } I^+ = I^+ \cup \{i\} \tag{5.74}$$

c) Compute membership ${}^{(p+1)}u_{ik}$ for $i \in I^-$ as ${}^{(p+1)}u_{ij} = 0$

d) Compute membership ${}^{(p+1)}u_{ik}$ for $i \in I^+$ from second part of (5.78).

e) Set $I^- = I^+ = \varnothing$ and go to (a) with next value of k.

(3) Check for convergence using some convenient norm on u_{ik} and if converged stop, else set $p = p + 1$, and go to step 2.

The above algorithm is further improvement over the FRC algorithm in two ways. First, it is able to handle noisy data (like R-FRC), and second, it handles situations when for FRC or R-FRC the memberships become negative, through use of Kuhn-Tucker conditions that include the non-equality constraint (5.14). When the relational data is such that robustness to outliers in not an important issue, one can "disable" the NR feature of this algorithm either by explicitly discarding the noise class from the algorithm, or by setting the noise distance δ to a very large value. That is called NE-FRC algorithm.

5.5 Numerical Results

In this section we consider several numerical examples for the relational clustering. The results are organized in several parts. First, we consider several classical examples of relational data from literature. In most cases, the results are presented for FRC and its improved versions. In terms of the classic examples from literature, there are three cases. First, a data set from Kaufman and Rousseeuw (Table 5, Chapter 2) [49] called "countries data" (CD). In this data set, dissimilarities between 12 countries in obtained by averaging the results of a survey among political science students. Second, from Table VII of Gowda and Diday [34], example of similarity matrix for fat-oils, called FAT. This example is also used by Hathaway and Bezdek [40] as an example of non-Euclidean relational data. Third, is also from Table X of Gowda and Diday, [34], example of similarity matrix for microcomputers, called COMP. The original similarity matrices for these examples are shown in Tables 5.1 to 5.3, for CD, FAT, and COMP, respectively.

When the original RFCM and FRC algorithms are used, the results for CD are shown in Table 5.4, and are very similar to that reported in using FANNY. These results show for a three class partition, USA, Belgium, France, and Israel as one group (cluster #

1, developed countries); Cuba, China, former USSR, and Yugoslavia as second group (cluster # 0, communist countries); and Brazil, Egypt, India, and Zaire (cluster # 2, developing countries) as third group. However, further analysis of the fuzzy partition reveals that Egypt is unlike any of the three typical groups, and its membership in cluster # 2 and cluster # 1 are very close to each other. In [49], their silhouette plot (page 176) also indicated that Egypt was "worst clustered" object. This example will be later analyzed using robust version of FRC. It is noted that using RFCM, the number of iterations for convergence was 43 and while for FRC, it was 15 for the same termination criteria. This indicates a modest increase in convergence due to the use of Seidel iteration technique. If the Siedal iteration scheme is not implemented it takes 35 cycles to converge.

Next, the FAT example is considered. Once again, the final results are similar for FANNY, FRC and RFCM. In Table 5.5, results using FRC are shown for two different cases of converting similarity measures to dissimilarity measures, as in Hathaway and Bezdek [40]. Unlike what is reported in Hathaway and Bezdek [40], the first case of computing dissimilarity measure did not require the computation of β in the NERFCM implementation that was used from MATLAB code downloaded from Hathaway's home page. And likewise, during the computation in FRC, the condition of obtaining negative a_{ij} was never encountered, and the memberships were always positive. Thus despite the dissimilarity being non-Euclidean, the FRC algorithm worked just fine. Again, we note that the number of iterations for NERFCM for case 1 and 2 were 13 and 10 respectively, while that of FRC for case 1 and 2 were 8 (9 without Siedal iteration) and 7 (8 without

Siedal iteration) respectively. This once again shows an advantage of using Seidel iteration. When the results in Table 5.5 are examined carefully, it is seen that the 2-partition looks fine. However, when the similarity matrix is examined, it is seen that the similarity of all oils in class 0 to Linseed oil is rather low as compared with the other members of that class. Thus when this example is run under FRC with 3 classes, we obtain results showed in Table 5.6. These results indicate that indeed Linseed oil may be classified into a class of its own. The results shown here are for case 2 dissimilarity measure. These results of FRC show that it is a good partitioning algorithm.

Next the COMP data is considered. However, during the computations, the FRC does run into getting negative memberships. Thus this data set appears to pose difficulties for the regular FRC algorithm, which does not explicitly satisfy constraint (5.14). This requires the use of either NR-NE-FRC or NE-FRC algorithm. The results using NE-FRC are shown in Table 5.7. The NE-FRC algorithm works well on this example, as shown. The classification obtained here is slightly different from that reported in [34], as the microcomputer "Ex. Sorcerer" and "O.S.II Series" are classified differently. However, a careful examination of the data in Table 3, and the original properties listed in Table IX of [34], may explain that this difference is in fact quite reasonable. For instance, the four-prototypical microcomputers according to the classification in Table 5.7, are, HP-85, Zenith H89, O.S. Challenger, and Atari 800. From Table 5.3, one can see that Ex. Sorcerer has higher similarity to Atari 800, than Zenith H89, and O.S.II Series has higher similarity to O.S. Challenger than Atari 800. Thus the classification shown in Table 5.7, is also an acceptable classification. When this example was tried with RFCM, it failed, and then it was tried with NERFCM. Our results for NERFCM were not good, as the algorithm yielded two identical clusters out of four, indicating either a problem of convergence or of coding. This issue was not further investigated.

Now we consider the CD again. In that example, it was discussed that Egypt is a difficult country to classify, as it is poorly classified. For this purpose, we may consider it as an outlier object, as it does not belong very well in the class of developing countries due to its simultaneous low dissimilarity with for example USA as well as India. Here, we then consider application of noise clustering, namely, R-FRC. The results using R-FRC are shown in Table 5.8, where class # 3 corresponds to the noise class. As seen from these results, Egypt is classified as an outlier, while the rest of the classification remains as before. One may argue that original FANNY or FRC could have been tried with a 4-class partition, and similar results could have obtained. That may work for a situation such as this where there is only one outlier, as one can see in Table 5.9, where ordinary FRC is applied to find a 4 class partition. As seen in Table 5.9, Egypt is in a class by itself, while the previous classification remains the same. However, when the outliers are well dispersed, such strategy will not work. This leads to an example, discussed in the next paragraph, which is derived from directly converting two dimensional object data into dissimilarity data using Euclidean distance measure.

We consider a modification of example from [49], page 165. Here, as compared to 22 data points in the original example, we have added 3 points to show that the noise clustering can handle dispersed biasing outliers. This example is shown in Figure 5.2. It is easy to see through human eye that in the object space, there are three distinct clusters and several outliers. Since the data is from Euclidean measure, R-FRC can be applied. The results of R-FRC are shown in Figure 5.4. Similar results are found through robust version of RFCM and FANNY. As can be seen, the outliers, plotted as "o" are clearly identified, and a perfect three good class partition is obtained. This shows that the noise clustering also works for relational data. When the original FANNY is run on this example, the results are shown in Figure 5.2, where FANNY classification is not perfect. As seen, three outliers belong to cluster on left, while one outlier each belongs to the other two clusters. Examination of the membership matrix (not shown here) reveals that all 5 outliers do have more fuzzy memberships as compared to the good points. However, with R-FRC, one can get automatic outliers identification. To illustrate the point made in the previous paragraph regarding the use of ordinary FANNY or FRC with an extra class, the results of FRC with four classes are shown in Figure 5.3. Here, it is clear that mere use of an extra class does not take care of outliers.

A textbook example shown in [18] is also considered in [69] using R-FRC. Here, a similar example is considered to see how robust FANNY is as compared to RFCM or FRC based on L_2 norm. In Figure 5.5, two clusters and one biasing outlier, at a medium distance from the right cluster, are shown. The results of partition of FANNY, and FRC with L_1 norm (i.e. using distances induced through such norm as measures of dissimilarity) are shown. Unlike the results in Figure 5.5, which shows outcome due to FRC with L_2 norm, the outlier does not completely bias the classification. While, in Figure 5.6, as in [18] and [69], the outlier itself becomes one class, and two good cluster become another class. To further test robustness of FANNY, one more outlier is added to this example, as shown in Figure 5.7. Here the classification results for FANNY, and FRC with L_1 norm, L_2 norm or $\sqrt{L_2^2}$ norm are the same, and the outliers are put into one class, and the two good clusters are put into another one. Thus for higher bias, FANNY is not as robust. Next, the R-FRC is applied, and as shown in Figure 5.8, the outliers are classified as noise points, plotted using "+" symbol, and perfect two good classes are identified. Thus, though this simple example, it is shown that explicit use of noise clustering technique is better than relying on L_1 norm to take care of bias due to outliers.

Although not shown here, many other examples were tested using R-FRC, and good results were obtained. While it is clear from these examples that use of L_1 norm may be better than L_2 norm, detection of outliers is not automatic through use of FANNY. For this type of examples, Kaufman and Rousseeuw [49] argue that use of FANNY versus non-fuzzy, i.e. hard clustering technique, is better, as hard clustering would automatically classify outliers (object 6 and 13 in their original example), to good classes, while the fuzzy memberships indicate that these objects are not well-classified. However, even with that knowledge, one cannot easily determine that these are outliers, while the R-FRC can do that automatically. Moreover, the claim by Kaufman and Rousseeuw [49] that use of L_1 norm in FANNY makes the result more robust is also not entirely valid as seen from previous example.

Nest, we consider cases of data that is far from being Euclidean. In this segment of the results, our aim is to show how the NR-NE-FRC or NE-FRC take care of such situations, without needing complex scheme as in NERFCM. As evident by now, in some of these cares, the regular FRC fails, as it runs into situations when the memberships turn negative, thus violating (5.14). The object data from example X of [7] is considered here, as shown in Figure 5.9. In this figure, each data point is labeled by a number (in bold, italic), used as the object number in the following results. As done in [7, 40], this object data is converted to relational data using several different norms, most of which are non-Euclidean. Euclidean norm (i.e. L_2^2) is contaminated (or distorted) by adding different fixed values, ranging from 10 to 40. Other norms such as Mahalanobis and sup norms are also used as in [7, 40]. The results for one of the two classes for this example for a 2-class partition are shown in Table 10. In all cases except the last one, which is the L_1^2, required use of NE-FRC, instead of FRC. For that case, FRC resulted in negative memberships. These results are consistent with those reported using RFCM and NERFCM. The β-spread expansion was needed for that case only, when using NERFCM. The membership results shown for the L_1^2 norm are interesting, since the two points closest to each cluster center have perfect hard memberships. This indicates that each of these points is seen as either being right at the center or "within" the perceived center or prototype for each cluster. If it is the latter situation, it explains why the ordinary FRC runs into negative memberships due to negative distances.

As shown before, the microcomputer data required application of NE-FRC instead of FRC, while the countries data did not. Both are examples of true relational data, and are far from being derived using the Euclidean norm. Yet when FRC (or RFCM or FANNY) are applied, the behavior is different. This is in line with the observations made in Table 5.10. Thus it is clear, being non-Euclidean alone is not a reason for the data to require

NE-FRC or NERFCM. Next, we briefly examine the effect of the fuzzifier, m. The FANNY algorithm uses a fixed value of 2 as the membership exponent. The FRC and other algorithms derived here use a variable exponent. In some cases, use of variable m may become a significant issue. Consider the example of Figure 5.9 again. We apply FRC for Euclidean dissimilarity for different values of m. The results for values of m varying from 1.4 to 2.5 are shown in Table 5.11. As shown, there is a gradual variation from very crisp to quite fuzzy memberships, as value of m becomes higher. This information is plotted in Figure 5.10 to show this variation. Thus it is clear to see the effect of m on the memberships. When the robust versions of these algorithms are used, the exponent m plays a great role in how the outlier rejection is done in the sense of weights of an equivalent robust M-estimator. This is seen in Figure 2 of [25], where the rejection smoothness changes as the exponent m changes. Since m does play a crucial role in fuzziness of memberships as well as outlier rejection, it is important to allow the user to select a proper value according to the application. Thus use of variable m in FRC as compared to a constant value in FANNY is more appropriate for wide range of applications.

Table 5.1 Countries data (CD): dissimilarity to other countries [49]

Countries	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
C1:Belgium	0.00	5.58	7.00	7.08	4.83	2.17	6.42	3.42	2.50	6.08	5.25	4.75
C2:Brazil	5.58	0.00	6.50	7.00	5.08	5.75	5.00	5.50	4.92	6.67	6.83	3.00
C3:China	7.00	6.50	0.00	3.83	8.17	6.67	5.58	6.42	6.25	4.25	4.50	6.08
C4:Cuba	7.08	7.00	3.83	0.00	5.83	6.92	6.00	6.42	7.33	2.67	3.75	6.67
C5:Egypt	4.83	5.08	8.17	5.83	0.00	4.92	4.67	5.00	4.50	6.00	5.75	5.00
C6:France	2.17	5.75	6.67	6.92	4.92	0.00	6.42	3.92	2.25	6.17	5.42	5.58
C7:India	6.42	5.00	5.58	6.00	4.67	6.42	0.00	6.17	6.33	6.17	6.08	4.83
C8:Israel	3.42	5.50	6.42	6.42	5.00	3.92	6.17	0.00	2.75	6.92	5.83	6.17
C9:USA	2.50	4.92	6.25	7.33	4.50	2.25	6.33	2.75	0.00	6.17	6.67	5.67
C10:USSR	6.08	6.67	4.25	2.67	6.00	6.17	6.17	6.92	6.17	0.00	3.67	6.50
C11:Y_slavia	5.25	6.83	4.50	3.75	5.75	5.42	6.08	5.83	6.67	3.67	0.00	6.92
C12:Zaire	4.75	3.00	6.08	6.67	5.00	5.58	4.83	6.17	5.67	6.50	6.92	0.00

Table 5.2 Fat-Oil data (FAT): similarity to other types of fat [34]

Type of Fat	F1	F2	F3	F4	F5	F6	F7	F8
F1: Linseed Oil	0.00	4.98	3.66	3.77	3.84	3.24	0.86	1.22
F2: Perilla Oil	4.98	0.00	5.70	5.88	4.70	5.30	2.78	3.08
F3: Cotton-Seed Oil	3.66	5.70	0.00	7.00	6.25	6.68	4.11	4.44
F4: Sesame Oil	3.77	5.88	7.00	0.00	5.90	6.37	3.61	3.97
F5: Camelia Oil	3.84	4.70	6.25	5.90	0.00	6.24	3.48	3.89
F6: Olive Oil	3.24	5.30	6.68	6.37	6.24	0.00	4.28	4.68
F7: Beef-tallow	0.86	2.78	4.11	3.61	3.48	4.28	0.00	6.74
F8: Lard	1.22	3.08	4.44	3.97	3.89	4.68	6.74	0.00

Table 5.3 Microcomputer data (COMP): similarity measure of microcomputers[34]

Microcompute	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
M1 Apple II	0.00	5.27	4.40	4.32	3.29	3.42	1.15	3.68	4.31	5.02	5.56	4.09
M2 Atari 800	5.27	0.00	4.65	4.98	3.54	3.67	1.35	4.35	4.23	4.94	4.98	4.34
M3 VIC 20	4.40	4.65	0.00	3.64	3.55	3.71	2.36	3.03	4.75	4.04	3.64	4.47
M4 Sorcerer	4.32	4.98	3.64	0.00	4.07	4.06	1.24	4.74	4.12	4.83	5.22	4.66
M5 Zenith H8	3.29	3.54	3.55	4.07	0.00	5.81	1.18	4.71	2.65	3.35	3.74	5.00
M6 Zenith H89	3.42	3.67	3.71	4.06	5.81	0.00	1.30	4.83	2.77	3.48	3.87	5.13
M7 HP-85	1.14	1.35	2.36	1.24	1.18	1.30	0.00	0.61	1.90	1.20	1.23	2.12
M8 Horizon	3.68	4.35	3.03	4.74	4.71	4.83	0.61	0.00	3.06	3.77	4.16	4.02
M9 Challenger	4.31	4.23	4.75	4.12	2.65	2.77	1.90	3.06	0.00	5.23	4.77	3.44
M10 O.S.II Ser	5.02	4.94	4.04	4.83	3.35	3.48	1.20	3.77	5.23	0.00	5.48	4.15
M11 TRS-80 I	5.56	4.98	3.64	5.22	3.74	3.87	1.23	4.16	4.77	5.48	0.00	4.61
M12TRS-80 III	4.09	4.34	4.47	4.66	5.00	5.13	2.12	4.02	3.44	4.15	4.61	0.00

Table 5.4 Results of FRC for countries data

Country	Membership Cluster 0	Membership Cluster 1	Membership Cluster 2	Cluster
Belgium	0.162085	0.596511	0.241404	1
Brazil	0.211748	0.261950	0.526302	2
China	0.493488	0.223969	0.282543	0
Cuba	0.634196	0.158889	0.206915	0
Egypt	0.250809	0.331799	0.417392	2
France	0.169511	0.597835	0.232654	1
India	0.288711	0.253162	0.458127	2
Israel	0.214003	0.491960	0.294037	1
USA	0.145707	0.628702	0.225591	1
USSR	0.599672	0.182350	0.217978	0
Yugoslavia	0.507477	0.234015	0.258508	0
Zaire	0.215261	0.252782	0.531957	2

Table 5.5 FRC results for FAT dissimilarity data, two class partition

Type of Fat	Membership of Cluster 0 $D_{ij} = 1/S_{ij} - \min\{1/S_{rt}\}$	Membership: Cluster 0 $D_{ij} = \max\{S_{rt}\} - S_{ij}$	Cluster
Linseed Oil	0.919404	0.703728	0
Perilla Oil	0.931349	0.818098	0
Cotton-seed Oil	0.898728	0.934605	0
Sesame Oil	0.924036	0.923698	0
Camelia Oil	0.852621	0.815740	0
Olive Oil	0.770371	0.833682	0
Beef Tallow	0.010047	0.036179	1
Lard	0.003215	0.028367	1

Table 5.6 FRC result, FAT dissimilarity data, three class partition, $D_{ij} = \max\{S_{rt}\} - S_{ij}$

Type of Fat	Membership: Cluster 0	Membership: Cluster 1	Membership: Cluster 2	Cluster
Linseed Oil	0.978027	0.014389	0.007584	0
Perilla Oil	0.374620	0.481974	0.143405	1
Cotton-seed Oil	0.006855	0.985786	0.007359	1
Sesame Oil	0.045338	0.915294	0.039368	1
Camelia Oil	0.153041	0.715315	0.131644	1
Olive Oil	0.063652	0.851195	0.085153	1
Beef Tallow	0.012414	0.023604	0.963983	2
Lard	0.011594	0.023580	0.964827	2

Table 5.7 FRC results for COMP dissimilarity data, $D_{ij} = 1/S_{ij} - \min\{1/S_{rt}\}$

Type of Fat	Membership: C0	Membership: C1	Membership: C2	Membership: C3	Cluster
M1: Apple II	0.015833	0.103860	0.327705	0.552602	3
M2: Atari 800	0.012020	0.081054	0.204650	0.702276	3
M3: Com. VIC 20	0.058804	0.156431	0.534898	0.249867	2
M4: Ex. Sorcerer	0.012808	0.141872	0.166658	0.678663	3
M5: Zenith H8	0.002547	0.964177	0.012164	0.021113	1
M6: Zenith H89	0.000631	0.991320	0.002948	0.005101	1
M7: HP-85	0.999936	0.000017	0.000028	0.000019	0
M8: Horizon	0.010957	0.523816	0.139712	0.325515	1
M9: O.S. Cha_ger	0.012817	0.026393	0.858304	0.102485	2
M10: O.S.II Series	0.008838	0.057624	0.475860	0.457677	2
M11: TRS-80 I	0.013582	0.118326	0.315086	0.553005	3
M12: TRS-80 III	0.035551	0.526332	0.152710	0.285407	1

Table 5.8 R-FRC results for countries data

Country	Membership: C0	Membership: C1	Membership: C2	Membership: NC	Cluster
Belgium	0.125768	0.463865	0.193747	0.216621	1
Brazil	0.148109	0.184149	0.390133	0.277609	2
China	0.331957	0.156315	0.198473	0.313255	0
Cuba	0.492317	0.114395	0.150414	0.242875	0
Egypt	0.178903	0.229919	0.294235	0.296942	3
France	0.127283	0.473710	0.177866	0.221141	1
India	0.193075	0.169230	0.324208	0.313487	2
Israel	0.151614	0.364953	0.211434	0.271999	1
USA	0.103371	0.543078	0.163064	0.190487	1
USSR	0.488580	0.125214	0.150562	0.235644	0
Yugoslavia	0.384422	0.159538	0.177857	0.278184	0
Zaire	0.144097	0.166744	0.426731	0.262428	2

Table 5.9 FRC results for countries data for a four class partition

Country	Membership: Cl 0	Membership: C1	Membership: C2	Membership: C3	Cluster
Belgium	0.117684	0.279481	0.412446	0.190389	2
Brazil	0.153291	0.275696	0.210442	0.360571	3
China	0.376661	0.215909	0.177566	0.229864	0
Cuba	0.544052	0.161583	0.125879	0.168486	0
Egypt	0.168994	0.297609	0.244644	0.288753	1
France	0.123400	0.267822	0.421788	0.186990	2
India	0.196259	0.266691	0.191019	0.346031	3
Israel	0.145553	0.289336	0.347941	0.217170	2
USA	0.106570	0.255417	0.458249	0.179763	2
USSR	0.489838	0.182214	0.146281	0.181667	0
Y_slavi	0.364596	0.231649	0.186632	0.217123	0
Zaire	0.157387	0.276484	0.203468	0.362661	3

Table 5.10 Memberships in class 0 using FRC or NE-FRC for objects of X from [7]

	Norm (E is the Euclidean, $(L_2)^2$)								
Object	E	Mahalanobis	L_1	Sup	E+10	E+20	E+30	E+40	$(L_1)^2$
1	0.931	0.900	0.931	0.943	0.859	0.787	0.715	0.631	0.920
2	0.905	0.679	0.905	0.919	0.819	0.745	0.678	0.605	0.906
3	0.999	0.992	0.999	0.981	0.879	0.785	0.703	0.619	1.000
4	0.905	0.679	0.905	0.919	0.818	0.745	0.677	0.605	0.906
5	0.810	0.842	0.810	0.842	0.703	0.641	0.595	0.553	0.792
6	0.498	0.496	0.498	0.499	0.500	0.499	0.500	0.500	0.499
7	0.188	0.153	0.188	0.156	0.296	0.358	0.404	0.447	0.206
8	0.095	0.320	0.095	0.081	0.181	0.255	0.323	0.396	0.094
9	0.001	0.008	0.001	0.019	0.121	0.214	0.297	0.383	0.000
10	0.095	0.320	0.095	0.081	0.181	0.255	0.323	0.396	0.094
11	0.069	0.100	0.069	0.057	0.141	0.212	0.287	0.373	0.080

Table 5.11 Memberships in class 0 using FRC for objects of X from [7, 40] with $(L_2)^2$ norm, and different values of *m*.

No	*m*=1.4	*m* =1.5	*m* =1.6	*m* =1.7	*m* =1.8	*m* =1.9	*m* =2.0	*m* =2.1	*m* =2.2	*m* =2.3	*m* =2.4	*m* =2.5
1	0.982	0.960	0.934	0.904	0.874	0.846	0.819	0.794	0.771	0.750	0.731	0.714
2	0.970	0.940	0.907	0.874	0.843	0.814	0.787	0.764	0.743	0.724	0.707	0.693
3	0.996	0.987	0.974	0.957	0.937	0.915	0.893	0.871	0.849	0.828	0.808	0.790
4	0.970	0.940	0.907	0.874	0.843	0.814	0.787	0.764	0.743	0.724	0.707	0.693
5	0.909	0.861	0.817	0.781	0.751	0.726	0.703	0.687	0.673	0.660	0.648	0.638
6	0.496	0.499	0.499	0.499	0.500	0.500	0.498	0.500	0.500	0.500	0.500	0.500
7	0.088	0.138	0.181	0.218	0,249	0.274	0.294	0.312	0.327	0.340	0.352	0.362
8	0.030	0.059	0.092	0.126	0.157	0.186	0.213	0.236	0.257	0.276	0.293	0.307
9	0.004	0.013	0.026	0.043	0.063	0.085	0.107	0.129	0.151	0.172	0.192	0.210
10	0.030	0.059	0.092	0.126	0.157	0.186	0.213	0.236	0.257	0.276	0.293	0.307
11	0.018	0.040	0.066	0.096	0.126	0.154	0.182	0.207	0.229	0.250	0.269	0.286

5.6 Conclusions

Several existing fuzzy relational data clustering techniques are examined and the issue of robustness to outliers and noise is considered in this chapter. The existing techniques are shown to have several commonalties, such as similar objective functional, and in some cases, direct connection with prior work in hard clustering of relational data. One of the main contributions here is to show why RFCM is quite successful for many non-Euclidean examples, despite having the restriction from equation (5.13). Further, it is shown that a generalization of FANNY, called FRC, has an identical objective function as RFCM, and one can minimize it without the restriction from (5.13). Next, the issue of robustness is addressed through application of concepts of noise clustering of object data [18]. The robustness of NC approach through its connection to robust M-estimators from robust statistics [24, 25] is shown before its extension to relational clustering. Although the extension of NC approach to relational data is not trivial, based on the new interpretations of noise class and its perception of dissimilarities, all the relational clustering techniques are converted to robust versions. In doing show, it is shown that the NC concept is consistently well behaved, and in each algorithm, the noise dissimilarity distance and associated noise memberships are well defined. Even in case of Windham's AP algorithm [76], its dual types of memberships are handled nicely through application of NC concept.

Based on the NC approach, and optimization techniques from [49], several versions of FRC algorithm are introduced. The FRC and R-FRC are useful in particular when the relational data is derived from Euclidean dissimilarities, and in some cases, these algorithms can handle the non-Euclidean data. The NE-FRC, and NR-NE-FRC can handle all types of non-Euclidean data through direct incorporation of the membership inequality constraint (5.14) in the minimization procedure. This versions compare well with NERFCM, which utilizes clever β-spread expansion of non-Euclidean data. Our results clearly show that it is unnecessary to perform such conversion of data at an added computational expense. Moreover, through the use of NC concept and extra noise class, these algorithms also handle noise and outliers in the data.

Examples, both real relational data and relational data derived from Euclidean distance of synthetic object data, are considered to show how the new algorithms behave. The series of examples presented show that FRC and its different versions have good convergence for these examples, and produce good classification. They handle noisy data as well as highly non-Euclidean data. The NR-NR-FRC should prove to be an excellent choice when analyzing large relational data sets, for which the expert prior knowledge is not available. Through varying the fuzzifier exponent, m, various grades of fuzzy memberships can be generated, and thus useful information about the data complexity could be generated.

In summary, a class of very practical fuzzy relational data clustering algorithms are introduced, that allows for robust classification of all types of relational clustering problems. Such relational data may be obtained from a variety of management, decision science, and other engineering applications. If the data is based on Euclidean dissimilarities, then use of FRC is recommended, while if such data is suspected to contain much noise, it is recommended that the R-FRC should be used, with the relational noise distance estimated through schemes similar to those for object data clustering [18, 25, 26]. When the data is expected to be highly non-Euclidean, it is recommended that the NE-FRC be used when noisy data is not expected, else, NR-NE-FRC should be used. In all cases, the user must select a proper value of the fuzzifier exponent, *m*.

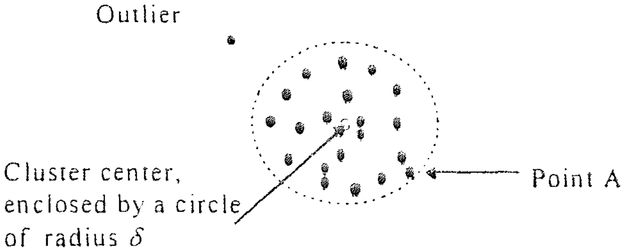

Figure 5.1 The noise distance δ in object data space

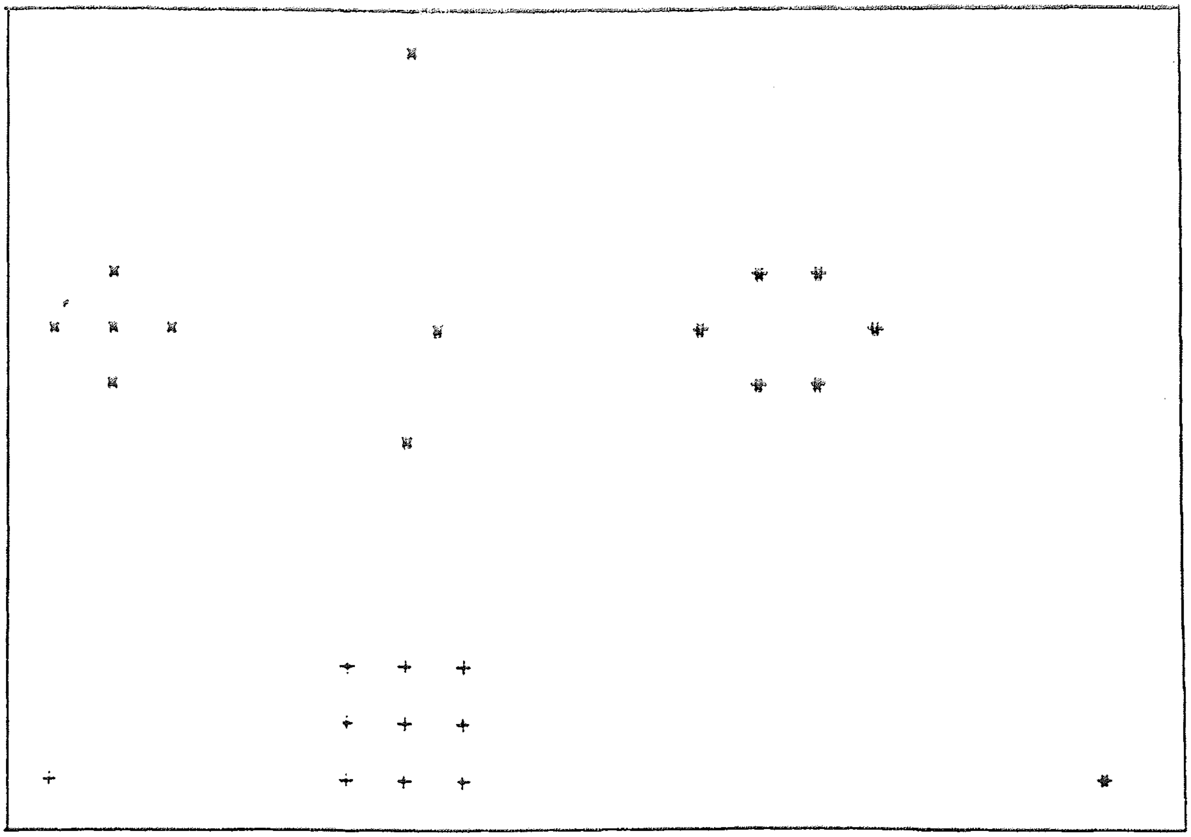

Figure 5.2 The euclidean data set with 25 points including 5 outliers

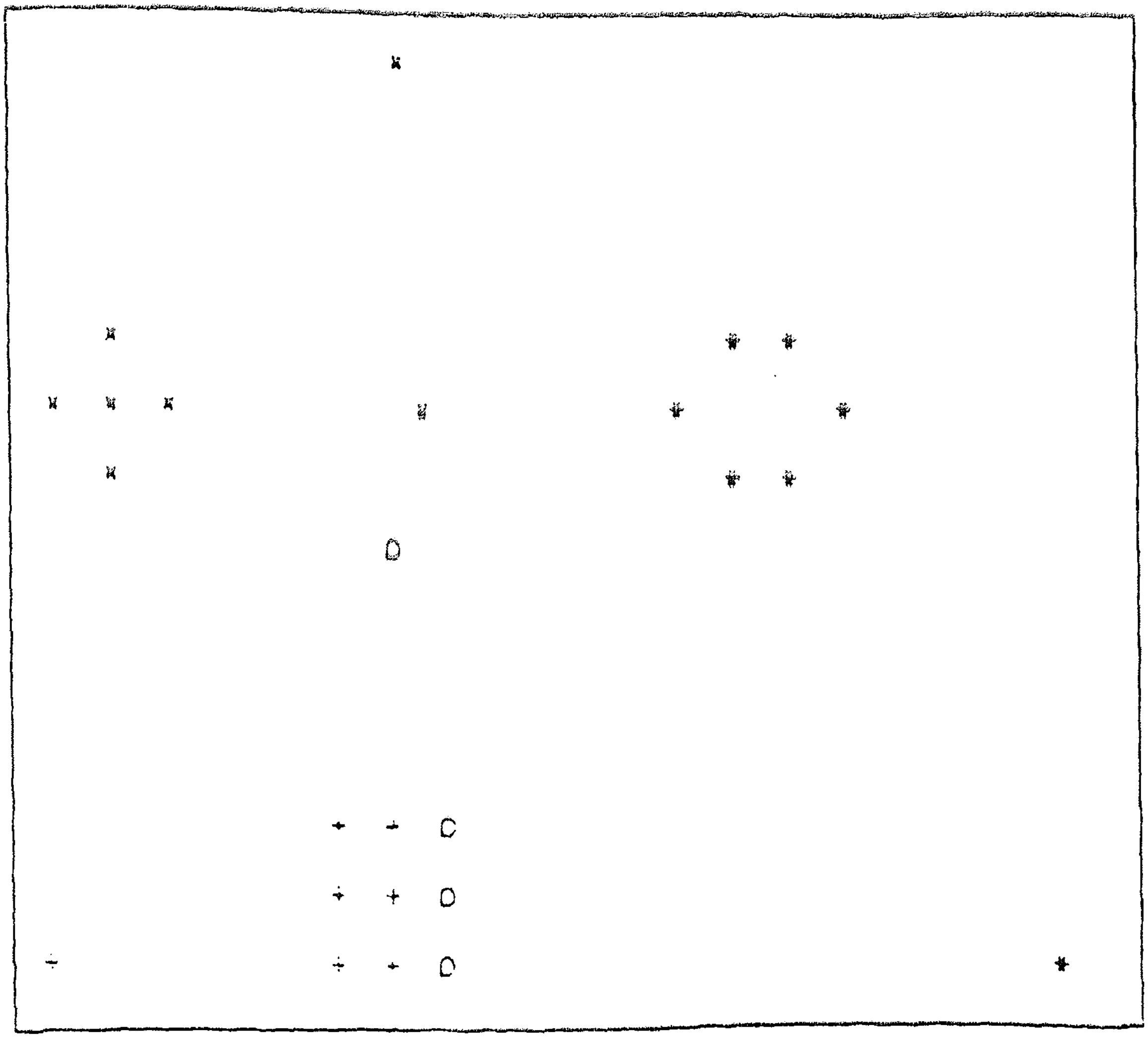

Figure 5.3 The result of FANNY and FRC on Figure 5.2 with number of clusters (c = 4)

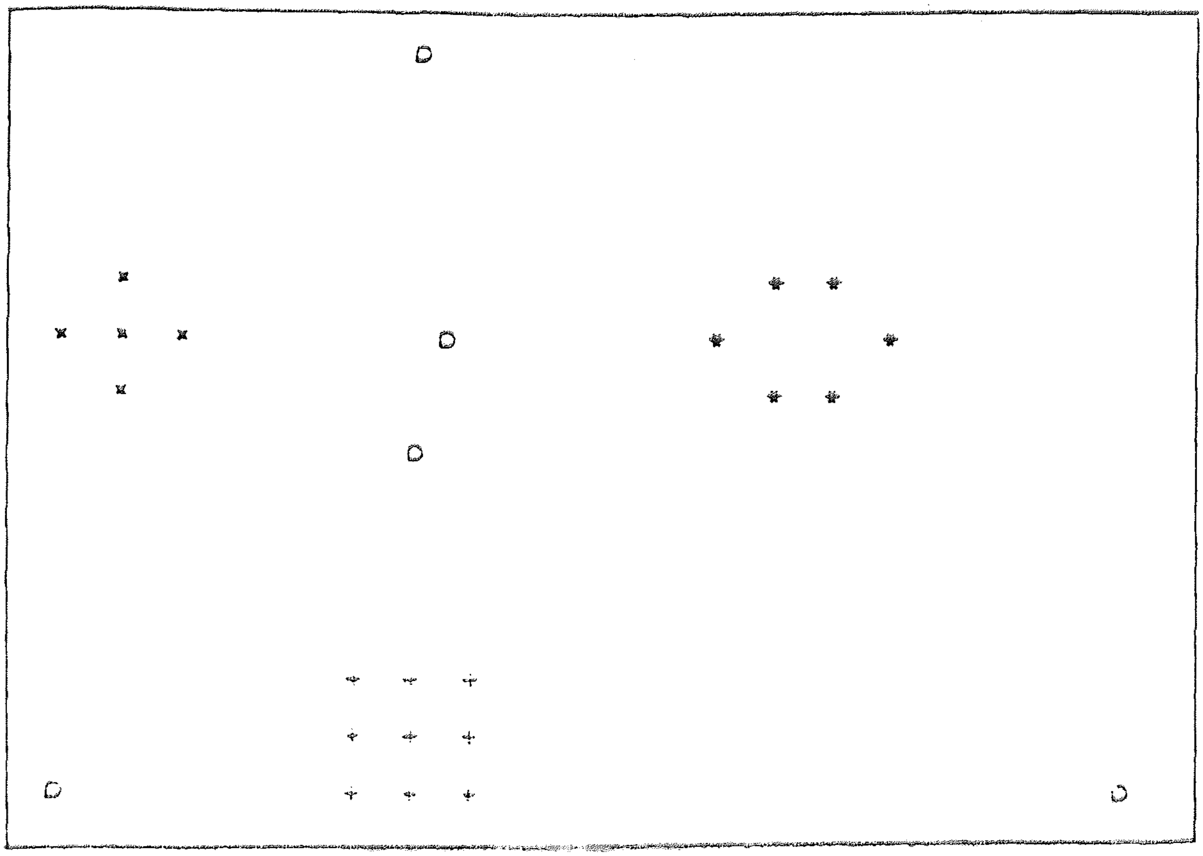

Figure 5.4 The results of R-FRC on Figure 5.2, number of clusters ($c = 3$)

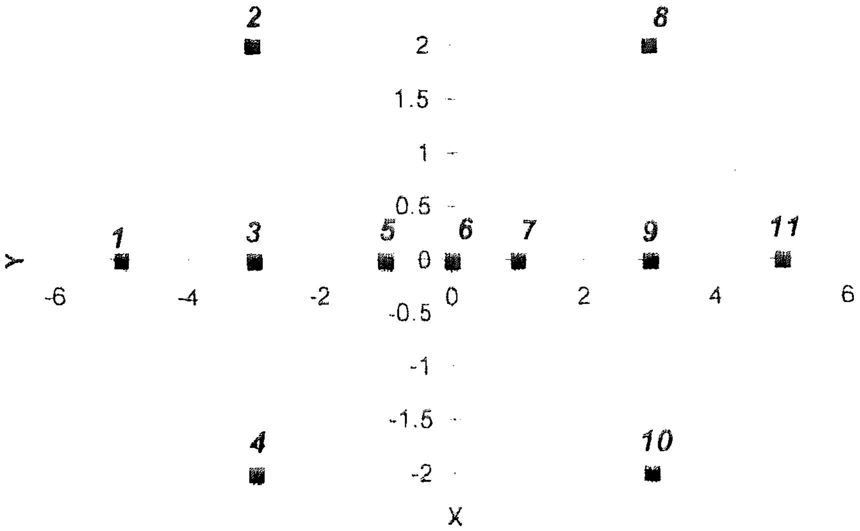

Figure 5.9 The geometry of 11 points data

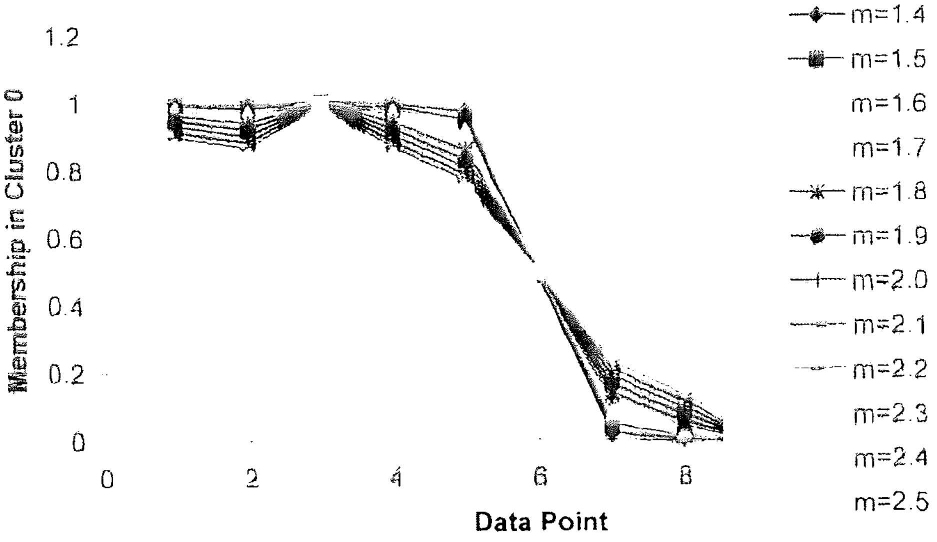

Figure 5.10 The variation of membership with change of m = 1.4 to 2.5

CHAPTER 6

APPLICATION OF NOISE CLUASTERING IN GROUP TECHNOLOGY

6.1 Introduction

Simplicity of the process layout is one of the key issues that determine the productivity of any manufacturing facility. The smooth component-flow through process layout directly leads to simplified management, reduction of throughput times, reduction in investment in inventory and reduced handling cost. Since the components usually require processing by well-defined set of machine tools, the flexibility of adding machines to the flow lines is often restricted due to high investment cost. Therefore it is advantageous to re-structure the production factory into a number of desegregated manufacturing cells such that each unit includes a number of dissimilar machines constitute a self-sufficient unit to process a family of parts grouped together based on the commonality between parts and manufacturing processes. This leads to the concept of cellular manufacturing [9, 33, 59]. The main aim is to improve the flexibility and manufacturing productivity. Cell formation, the most important problem faced in designing cellular manufacturing systems, is to group parts with similar geometry, function, material and/or requiring a similar production process into part families and corresponding machines are organized as independent cells. Therefore the initial step in designing a cellular manufacturing system is the identification of part families and formation of machine cells. More often the flexibility in machine cells formation is limited compared to the part-family formation since machine cell formation is dictated by economic constraints of resources and limitations in manufacturing facility. Therefore the problem boils down to assignment of

n parts into c fixed cells in order to maximize the objective of manufacturing. Typically it is a classification problem [9, 10, 11].

Group technology is a manufacturing philosophy which seeks to exploit underlying similarity between components in order to achieve improved planning, operation of manufacturing system and increased productivity and efficiency. Obviously the GT principles constitutes the framework of cellular manufacturing concept[59] since the parts that require processing by similar machines are grouped together. Grouping algorithms are traditionally based on assuming a well-defined (preferred) routing of each component. This dictates the preferred set of machine tools to be used for its processing. Compared to the conventional analytical methods such as array based clustering, hierarchical clustering (or similarity coefficient-based) and considering other alternatives, fuzzy clustering has been advocated as an appropriate methodology for part family formation in cases where no clear division between component groups can be achieved and hence crisp logic of family formation does not seem appropriate [11]. Therefore fuzzy clustering provide a more realistic environment for decision making. So fuzzy mathematics is employed in this research. The employment of fuzzy membership functions allows us to include uncertainty inherent in part features and thus produce more realistic results. In recent past fuzzy clustering has also started receiving considerable attention for solution of part classification problem in cellular manufacturing and FMS. In [11] Chu and Hayya first applied fuzzy c-means clustering algorithm for solution of group formation. In fact work load balancing among different cells has been able to be achieved using fuzzy method. In [78] new approaches based on fuzzy classification and fuzzy equivalence are introduced in the process of part family formation. A dynamic part-family assignment procedure is presented using the methodology of fuzzy pattern recognition to assign new parts to existing PFs. The problem of number of groups in GT is equivalent to the problem of cluster validity in fuzzy clustering. In [33] the grouping algorithms has been extended to develop the criteria for partitioning the components into an 'optimum number' of groups. Clearly the fuzzy clustering method has evolved to be an extremely valuable tool in the classification type of problems in GT.

Overall GT is an organizational technique to improve manufacturing productivity focusing on factors like delivery speed, quality, design flexibility, delivery reliability, as well as manufacturing costs. One of the main weakness of the conventional grouping methods, including the fuzzy clustering technique [4, 6], is that they implicitly assume that the components belong to one of the part families. In reality, some parts often require processing by machines from multiple cell and thereby belong to more than one part families and appear as bottleneck. As a result the cells ceases to be self-sufficient units. This requires material handling between cells, leading to complex material flow, increased lead time and escalation of investment in inventory. It is therefore necessary to identify these bottleneck parts while grouping, and may process alternatively by subcontracting. Truly the identification of bottleneck parts is equivalent to the isolation of noise and outliers in robust fuzzy classification task. So to solve this problem we propose to apply Dave's noise model of fuzzy [18] clustering.

The rest of the chapter proceeds as follows. In the next section the fuzzy clustering formulation of the problem is described with its inability to handle outlier elements or exceptional parts. How Dave's noise model of fuzzy clustering is implemented to solve the problem is outlined in section 6.3 with example. In section 6.4 the grouping efficiency of this method is compared with other analytical methods in the context of examples from the literature. The paper is concluded by the discussion of the results and summary on this approach in section 6.5.

6.2. Fuzzy Clustering in Group Technology

In 1965 Zadeh [80] first proposed the theory of fuzzy sets was the fuzzy generalization of the ordinary mathematical concept of sets. In a universe of discourse U, a fuzzy subset A of U is defined by a membership function $\mu(x)$ representing the degree of membership of each element x_i in a fuzzy subset A. The value of $\mu(x)$ lies between 1 and 0 depending on full, partial or no membership. In 1981 Bezdek[6] incorporated to the fuzzy concepts in k-means clustering and proposed the classical theory of fuzzy c-means (FCM) clustering. Applications in various field have proved FCM as a powerful classification tool. The fuzzy clustering method [4, 6] was first introduced in group technology by [11].

Assume that there are n parts and p machines to be grouped c part families and corresponding machine cells. Conventional grouping methods implicitly assume that disjoint part families exist in data set; therefore a part can only belong to one part family. We can represent it as a binary matrix.

$$T = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & \dots & 0 \\ . & . & . & \dots & . \\ . & . & . & \dots & . \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \tag{6.1}$$

and that

$$u_{ik} = 0 \text{ or } 1,\ i = 1,2,\ldots,c \text{ and } k = 1,2,\ldots,n \tag{6.2}$$

$$\sum_{i=1}^{c} u_{ik} = 1, k = 1,2,\ldots,n, \tag{6.3}$$

$$0 < \sum_{k=1}^{n} u_{ik} \leq n\ \ i = 1,2,\ldots,c \tag{6.4}$$

The constraint (6.2) ensures that that u_{ik} equals to 1 if the kth part belongs to the i th part family. Constraint (6.3) ensures that each part exactly belongs to one part family. Constraint (6.4) ensures that each part family consists of at least one part.

But in many cases, while classifying real parts, some families are not completely disjoint, rather separation of part families and machine cells is rather fuzzy. Under the circumstances Chu and Hayya [11] applied the fuzzy clustering technique i.e. FCM for better representation of the classification problem where the degree or the grade of membership of a part to each family is expressed by a value between 0 and 1. Consequently each element of the above membership matrix T is represented by decimal number. While the constraint (6.3) and (6.4) remains unaltered, the constraint (6.2) is only modified to represent to represent fuzzy membership of the parts to the family, such that

$$0 \leq u_{ik} \leq 1, i = 1,2,\ldots,c,\ k = 1,2,\ldots,n \tag{6.2a}$$

So the parts now belong to various part families with different degree of membership. The application of FCM in machine cells / part family formation in GT is simple.

Though there may be infinite number of classifications possible it is obvious to group the parts into C families that results in most compact clusters. If we represent the prototype part of the i th part family V_i as $V_i = \left(v_{i1}, v_{i2}, \ldots, v_{ip}\right)$ and

$$v_{ik} = \frac{\sum_{j=1}^{N} u_{ij} * \mu\left(x_{jk}\right)}{\sum_{j=1}^{N} u_{ij}} \text{ where } i = 1,2,\ldots,c;\; k = 1,2,\ldots,p \tag{6.5}$$

the distance of the j th part from the prototype of i th family can calculated as

$$\left[\sum_{k=1}^{p}\left(\mu_k\left(x_{jk}\right) - v_{ik}\right)^2\right]^{1/2} \tag{6.6}$$

and the weighted sum of squares of the distance of the j th part from C prototype part of the respective families is expressed as

$$\sum_{i=1}^{C} u_{ij} \sum_{k=1}^{p}\sum_{k=1}^{p}\left(\mu_k\left(x_{jk}\right) - v_{ik}\right)^2 \tag{6.7}$$

The total weighted sum of squares of the distance of N parts from C reference pattern is

$$J(U,V) = \sum_{j=1}^{N}\sum_{i=1}^{C} u_{ij}^{m} \sum_{k=1}^{p}\left(\mu_k\left(x_{jk}\right) - v_{ik}\right)^2 \tag{6.8}$$

Here m refers to the fuzzifier has a value > 1; used primarily to reduces the effective membership value. The minimization of the above functional under the constraint (6.2), (6.3) and (6.4) Bezdek (1983) consist of following few steps

(1)Choose desired number of part families C, $1 < C < N$ and a value of $m > 1$, usually 2

(2) Choose an initial classification matrix $U^{(0)}$ and the value of ξ for stopping criterion

(3) For iteration $r = 0,1,2,\ldots$, calculate the mean vector

$$v_{ik}^{(r)} = \frac{\sum_{j=1}^{N} \left(u_{ij}\right)^{m} \mu\left(x_{jk}\right)}{\sum_{j=1}^{N} u_{ij}^{m}} \text{ for } i = 1,2,\ldots,c \text{ and } k = 1,2,\ldots,p$$

(4) Update $U^{(r)}$ using $v_{ik}^{(r)}$ such that

$$u_{ij} = \frac{1}{\sum_{l=1}^{C} \left(\frac{\sum_{k=1}^{p} \left(\mu_{k}\left(x_{jk}\right) - v_{ik}\right)^{2}}{\sum_{k=1}^{p} \left(\mu_{k}\left(x_{jk}\right) - v_{lk}\right)^{2}} \right)^{1/(m-1)}} \text{ where } i = 1,2,\ldots,c \text{ and } j = 1,2,\ldots,n$$

(5) Compare $U^{(r)}$ to $U^{(r+1)}$. If $\left|u_{ik}^{(r+1)} - u_{ik}^{r}\right| \leq \xi$ stop otherwise go to step (3)

Clearly since the number of machines p is quite high, even comparable with the number of parts n, therefore data is often very sparse. The numerical implication is, the memberships tend to be more fuzzy. Since it is very high (p) dimensional dataset, the visualization of it is a difficult task.

The algorithm being applied to a data set shown in Table 6.1 and the resulting grouping in The grouping is perfect if all the parts to be grouped clearly belong to one of the cells. However some exceptional components that require processing by multiple cells while desegregating the production of the factory into disconnected cells, leads to noise or outlier problem. Noise or the outliers are those data that do not belong to any cluster.

Table 6.1 The example of 18 parts and 9 machines

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1			1	1			1			1	1		1					1
2	1	1		1		1									1	1	1	
3	1	1				1									1	1	1	
4				1	1				1			1	1	1				
5				1	1				1			1	1	1				1
6			1			1	1			1	1							1
7			1				1			1	1							1
8				1	1	1			1			1	1	1				
9	1	1				1							1		1	1	1	

Table 6.2 Classification of parts by FCM of example in Table 6.1

	3	7	10	11	18	1	2	6	15	16	17	4	5	8	9	12	13	14
1	1	1	1	1	1							1					1	
6	1	1	1	1	1			1				1						
7	1	1	1	1	1													
2						1	1	1	1	1	1							
3						1	1	1	1	1	1							
9						1	1	1	1	1	1						1	
4												1	1	1	1	1	1	1
5					1							1	1	1	1	1	1	1
8								1				1	1	1	1	1	1	1

Unable to distinguish between a well-classified component and a bottleneck part is the main limitation for applying FCM type algorithm in group technology problems. An ideal solution would be one where these bottleneck elements get automatically identified and removed from the pool of parts. The concept of having an approach where one cluster or cell, defined as noise cluster or noise cell where all the bottleneck parts could be dumped. Next it is required to define the similarity or dissimilarity measure for the noise cell. Based on this argument Davé proposed the algorithm of Noise clustering [18].

6.3 Segregation of Outliers by Noise Clustering Technique

Davé [18] considered noise to be a separate class, and represented it by a prototype that has the same distance, δ, from all the feature vectors. His definition does not specify what the distance is, but it states that all the parts have equal apriori probability of belonging to the noise cell. This makes sense, since given no prior information, all parts should have an equal probability of falling in noise cell. As the algorithm progress, the parts belonging to good machine cells will increase their probability to be classified. The membership u_{*j} of a part x_j in noise cell is defined as,

$$u_{*j} = 1 - \sum_{i=1}^{C} u_{ij} \tag{6.9}$$

Here C is the number of clusters and u_{ij} denotes the grade of membership (belonging) of part x_j in the i th fuzzy subset of X. Since (6.1) is used to define the membership u_{*j} in the noise class, the usual membership constraint of FCM algorithms is not required. Thus, the membership constraint for the good clusters is effectively relaxed to

$$\sum_{i=1}^{C} u_{ij} \leq 1 \tag{6.10}$$

This allows bottleneck part to have arbitrarily small membership values in good clusters. The objective function being optimized by noise clustering algorithm is given as

This allows bottleneck part to have arbitrarily small membership values in good clusters. The objective function being optimized by noise clustering algorithm is given as

$$J(B,U;X)=\sum_{i=1}^{C}\sum_{j=1}^{N}\left(u_{ij}\right)^{m}d^{2}\left(x_{j},\beta_{i}\right)+\sum_{j=1}^{N}\delta_{ij}^{2}\left(1-\sum_{i=1}^{C}u_{ij}\right)^{m}. \qquad (6.11)$$

In (6.11), $d^{2}\left(x_{j},\beta_{i}\right)$ is the distance from a feature point x_{j} to the prototype β_{i}.

The above functional can be optimized with respect to the part family prototypes and the memberships in a manner similar to FCM functional as shown in section 6.2. The resulting equations for the prototype parameters are very similar, however, the equation for the memberships is different, and is given as,

$$u_{ij}=\frac{\left(\frac{1}{d_{ij}^{2}}\right)^{1/(m-1)}}{\sum_{k=1}^{C}\left(\frac{1}{d_{kj}^{2}}\right)^{1/(m-1)}+\left(\frac{1}{\delta^{2}}\right)^{1/(m-1)}} \qquad (6.12)$$

In the above, d_{ij} is equivalent to $d^{2}\left(x_{j},\beta_{i}\right)$. It can be easily seen that the membership for FCM does not have the second term in the denominator, and thus the NC memberships are different. Here, by specifying the value of δ , we can separate out the dissimilar parts. The noise distance δ is acts like a "limit" or "boundary" that eventually determines whether the part belong to any group or an outlying bottleneck element. It is interesting to note that in group technology problem of cell formation the bottleneck parts that require processing by additional machines from other cells will lead to generate intra-cellular outliers and noise element. If a part require processing by the machines not in demand by any other cell, that will appear as an extra-cellular outlier during classification. However this type of biasing outlier are rarely encountered. To apply this the δ is described in detail [18, 25, 26]. Applying NC algorithm with δ = 0.3 to the problem in Table 6.1 the resulting classification is shown in Table 6.3.

Table 6.3 Classification of parts by NC algorithm with δ = 0.3

	3	**7**	**10**	**11**	**1**	**2**	**15**	**16**	**17**	**5**	**8**	**9**	**12**	**14**	**4**	**6**	**13**	**18**
1	1	1	1	1											1		1	1
6	1	1	1	1												1		1
7	1	1	1	1														1
2					1	1	1	1	1						1	1		
3					1	1	1	1	1							1		
9					1	1	1	1	1							1	1	
4										1	1	1	1	1	1		1	
5										1	1	1	1	1	1		1	1
8										1	1	1	1	1	1	1	1	

6.4 Numerical Results

The fuzzy c-means (FCM) clustering algorithm [4, 6] has unique ability of simultaneous detection of multiple clusters by segmenting data into multiple groups as well extracting their cluster prototype centers at the same time. Since noise clustering was introduced primarily to robustify the FCM-family of algorithms against noise and outliers in data, it inherits the partitioning capability into multiple clusters over and above identify the outlying elements as well. This makes it an ideal candidate for identification of bottleneck parts while forming the cells following the GT principles. The first example in Table 6.1 shows each of the 18 parts that require processing by several of the 9 machines. The result obtained by FCM C = 3 is shown in Table 6.2. Though it produced classification, clearly it could not create totally disconnected cells. After applying noise clustering algorithm with same number of clusters $C = 3$ and noise distance $\delta = 0.3$, it is possible to identify the bottleneck parts at the same time while forming the disconnected machine cells or part families. The results are shown in Table 6.3. However selection of correct value of δ is a very important issue for correct results.

Table 6.4 The example of 20 parts and 8 machines from [11]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		1	1					1	1		1		1	1		1	1		1	
2			1	1		1	1							1				1		1
3		1						1	1		1		1	1		1	1		1	
4			1	1		1	1			1								1		1
5	1				1	1				1		1			1		1			
6	1				1				1	1		1			1					1
7			1	1		1	1				1	1						1		1
8			1	1		1	1											1		1

The example in Table 6.4 is obtained from [11] has 20 parts 8 machines. Though application of FCM produces identical part families as reported in [11], is shown in Table 6.5. That explains the FCM as a useful classification tool for cell formation in GT applications. It was evident that there are few bottleneck parts in the data set. It is therefore attempted to classify the good parts into groups and identify the bottleneck parts as outlying noise. While using the NC algorithm with $\delta = 0.9$ we can identify all exceptional parts as well as classify the good parts in groups and forms disconnected cells as displayed in Table 6.6.

Table 6.5 Classification of parts by FCM of example in Table 6.4

	2	8	9	11	13	14	16	17	19	3	4	6	7	18	20	1	5	10	12	15
1	1	1	1	1	1	1	1	1	1	1										
3	1	1	1	1	1	1	1	1	1											
2						1				1	1	1	1	1	1					
4										1	1	1	1	1	1			1		
8										1	1	1	1	1	1					
7				1						1	1	1	1	1	1				1	
5								1				1				1	1	1	1	1
6			1												1	1	1	1	1	1

Table 6.6 Classification of example in Table 6.4 obtained by NC algorithm with δ=0.9

	2	8	13	16	19	4	7	18	1	5	15	3	6	9	10	11	12	14	17	20
1	1	1	1	1	1							1		1		1		1	1	
3	1	1	1	1	1									1		1		1	1	
2						1	1	1				1	1					1		1
4						1	1	1				1	1		1					1
8						1	1	1				1	1		1					1
7						1	1	1				1	1		1	1	1			1
5									1	1	1		1				1		1	
6									1	1	1			1			1			1

The Table 6.7 is collected from the literature [59]. Here the application of NC could not identify bottleneck parts directly, instead classified them among in detected cells. Due to constraint of space the machine component process Table 6.7 is displayed in reverse order with machines in columns and components along the rows.

Since there are only 41 parts to be classified in 30 dimensional space, since there are 30 machines, visualization proves that data is very sparse. The resulting classification

Table 6.7 The example of 30 machines and 41 components from [59]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1								1											1	1									1	1
2										1													1							
3									1										1	1									1	1
4														1											1					
5					1									1																
6						1										1														
7				1													1													
8								1																			1	1		
9								1																					1	
10	1										1										1	1								
11		1										1																		
12	1	1	1							1		1									1	1	1							
13									1																				1	
14								1										1												
15				1																							1			
16							1										1	1								1				
17					1										1															
18		1												1																
19												1	1																	
20										1		1																		
21									1										1	1									1	1
22																			1	1										1
23			1							1		1										1	1							
24				1								1																		
25				1									1																	
26						1								1																
27							1										1	1								1				
28				1												1														
29								1																				1		
30																			1										1	
31	1		1																		1	1	1							
32	1	1																			1	1								
33	1	1									1										1									
34							1											1								1				
35				1																								1		
35					1		1											1								1				
37														1	1															
38													1											1						
39	1	1	1							1	1	1										1	1							
40			1									1									1	1								
41	1	1									1																			

produced will be too fuzzy, as evidenced by all membership coefficients have approximate value 0.33. Though, after assignment to the cells, the parts produced nice groups but not totally disconnected cells as shown in Figure 6.8.

Table 6.8 Classification of 30 machines and 41 components by NC with $\delta = 2.1$

	1	21	11	2	22	12	3	23	10	14	25	15	5	6	18	17	26	7	24	13	4	16	27	28	8	29	9	19	20	30
10	1	1	1	1																										
41	1		1	1																										
32	1	1		1	1																									
33	1	1	1	1																										
31	1	1			1		1	1																						
40		1			1	1	1																							
11				1		1																								
23					1	1	1	1	1																					
39	1		1	1	1	1	1	1	1																					
12	1	1		1	1	1	1	1	1																					
20						1			1																					
2								1	1																					
4										1	1																			
37										1		1																		
5										1			1																	
26										1				1																
17												1	1																	
36													1		1		1	1												
27															1	1	1	1												
16															1	1	1	1												
34															1		1	1												
38																			1	1										
25																				1	1									
28																					1	1								
15																					1		1							
35																					1			1						
8																							1	1	1					
29																								1	1					
6															1							1								
7																1					1									
14															1										1					
18				1						1																				
19						1														1										
24						1															1									
1																									1	1		1	1	1
3																										1	1	1	1	1
9																									1	1				
13																										1	1			
21																										1	1	1	1	1
22																												1	1	1
30																										1		1		

To identify the bottleneck parts we implement the following novel technique on the classification results. The uses of the machines in each cell are calculated from the classified parts in different part families. It is termed as Utilization coefficient $U[machine][cell]$ of the machine. The machine will be assigned to the cell that has highest utilization coefficient for it. As we know if a specific machine is used by more than one cell, it will be termed as bottleneck machine. The parts in different cells that use the same machine are exceptional parts. Therefore those parts in other cells should be identified as bottleneck parts. It is therefore necessary these bottleneck machines. Since the number of machines in any manufacturing facility is less than the number of parts, due to obvious reason of process requirement or financial limitation, it is therefore comparatively easier to identify these machines. The visualization of utilization coefficient $U[machine][cell]$ is obtained in Table 6.9. Clearly the machine # 2, 8 and 12 are used by more than one cell. So all the parts using these machines i.e. 8, 14, 18, 19, 24, 29 are considered as the exceptional parts causing bottleneck in processing. However it is mandatory to have good classification before applying this method. By good result, it is implied that the bias due the bottleneck parts has no drastically altered the classification.

Table 6.9 The utilization co-efficient $U[machine][cell]$ of 30 machines in 3 cells

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
7	7	5	0	0	0	0	0	0	5	4	6	0	0	0	0	0	0	0	0	6	6	5	0	0	0	0	0	0	0
0	1	0	6	3	1	4	3	0	0	0	2	3	5	2	2	3	0	0	0	0	0	0	1	1	4	2	3	0	0
0	0	0	0	0	0	0	2	3	0	0	0	0	0	0	0	0	0	5	4	0	0	0	0	0	0	0	0	6	4

6.4.1. Grouping Efficiency

To compare the result of classification by different algorithms instead of subjective consideration, Chandrasekharan and Rajagopaln in [10] provided a quantitative measure of how well the parts are grouped together with a set of machines while forming the cells. As aptly termed as grouping efficiency, defines the concentration of 1's in diagonal matrix is considered as an indicator of within-cell utilization of machines, whereas 1's in nondiagonal locations are indicators of intercellular movements. It is desirable that each part should visit all the machines in their respective cell to utilize the entire capacity of the machines and minimize the intercellular trips. The within-cell work load factor for all components as a fraction of total work load in all the cells and the number of intracellular moves determine the overall goodness of the solution and reflects the level of independence achieved as a result of cell formation. Therefore, we get

$$\eta = q\eta_1 + (1-q)\eta_2 \tag{6.13}$$

where η_1 is the total intercellular moves as a fraction of total moves (intercellular and intracellular moves combined), η_2 is the total within-cell work load as a fraction of total work load when combined for all the cells, and q is any user specified weight which reflects the user consideration of η_1 and η_2. Usually the q is 0.5 selected.

6.5 Conclusion

The traditional machine-component grouping algorithms in cellular manufacturing including fuzzy clustering method, based on similarity analysis of production process, assign all the parts to one of the machine cells. Those components, require processing by machines from multiple cells are the exceptional parts, lead to inter cellular movement, create bottleneck in manufacturing process planning. This leads to more complex material flow, increased inventory and overall throughput time. While desegregating the production process into disconnected manufacturing cells, it is essential to isolate the bottleneck parts thereby each cell achieve self sufficiency and improve grouping efficiency.. Here we propose a strategy based on Dave's noise model of fuzzy clustering [18] approach to handle the task. The method is flexible enough of letting the designer choose the number of cells. Results are shown in the Table 6.1 to 6.8. Table 6.10 shows the grouping performance. Clearly, it establishes NC is an efficient tool in identification of bottleneck parts in GT application.

Table 6.10 Comparison of grouping efficiency FCM and NC algorithm

	Fuzzy C Means Algorithm	Noise Clustering Algorithm
1	0.75	0.98
2	0.66	0.99
3	0.57	0.99

CHAPTER 7

CONCLUSION AND FUTURE DIRECTION

7.1 Conclusion

The main objective of this dissertation was the robust detection of multiple clusters from noisy range image. This was accomplished by considering improvement of NC approach for both the conventional gradient-descent based and the probabilistic search based methods that attempt to minimize objective function.

For improving robustness, two most popular fuzzy clustering algorithms, namely NC and PCM are considered. It was shown that while PCM being a mode seeking algorithm, is capable of detecting clusters in noisy data, it lacks the partitioning capability of FCM. On the other hand, NC technique does have the partitioning capability of FCM, and can detect clusters in noisy data. Further NC algorithm was generalized by allowing the noise distance to take different values for different points. This generalization enables the NC technique to successfully detect clusters of varying sizes in noisy data. It was also shown that the NC membership is a product of FCM membership with partitioning capability and a type of robust possibilistic membership based on harmonic mean distance. Moreover, through judicious choice of variable noise distance, the generalized NC can be made equivalent to a Fuzzy c-M-estimator clustering algorithm, capable of mimicking various robust M-estimators. Hence, the generalized NC algorithm can have the desirable robustness aspects of M-estimators, and is thus used in subsequent work.

To eliminate the problem initialization of NC method, we considered sampling based robust LMS estimator. LMS has the highest break down point 0.5, (the theoretical maximum) which is applicable if the data contains only one cluster. In view of our problem the LMS algorithm required to be extended to fuzzy c-LMS algorithm for detecting multiple clusters simultaneously. Since LMS family of algorithms are sampling based methods, they require exhaustive search or probabilistic limited search over the data set. To reduce the number of sampling, the concept of repeated evidence of RHT has been blended to develop a faster fuzzy c-LMS. The algorithm is tested to be highly robust for two dimensional edge data. However the algorithm requires ordering of all points in each sampling. So its applicability for object detection from range image often becomes computationally a big task.

Comparative analysis of NC, LMS and RHT methods is shown with relative merits and demerits. A new sampling based robust algorithm is proposed, based on noise clustering principle, called noise least square or NLS method, that combine the good features of NC, LMS and RHT. The method basically searches for one cluster at a time directly from image space instead of parameter space. The noise distance δ is required to be specified similar to NC. This algorithm basically fits the prototype where the density of points in image is maximum. It is shown to be a very robust method for detection of quadric surface from range image. To improve the speed of detection, the range image is partitioned into several regions to restrict the sampling within single cluster thereby reduce the number of false iterations. The strategy is shown to be very successful.

Several existing fuzzy clustering methods for relational data were examined. Commonality and differences were discussed. A new algorithm, which combine the good features of RFCM and FANNY, called FRC was introduced. The concept of noise prototype and the distance it for relational data was also developed. Applying this FRC is extended to a robust method, named NFRC. Several popular fuzzy methods also have been robustified accordingly. Since relational data often does not conform to Euclidean metric, NFRC was further modified to handle non-Euclidean data by including inequality constraint (5.55) through Kuhn Tucker condition. This method was shown to be less computational compared to NERFCM.

As an example application of fuzzy clustering in solution of engineering problems, we identified the machine/component cell formation in cellular manufacturing following the group technology (GT) principles. The problem of identification of exceptional parts to create desegregated cells for better manufacturing efficiency is detailed.

Overall the robustness aspect of fuzzy classification has been addressed in this research. Apart from introducing new algorithms, both for object data and relational data space, new application has been tested in the field of group technology. Several interesting results do encourage continuing further investigation in future.

7.2 Future Direction

There are several interesting issues that evolved during current research that requires being resolved. NLS algorithm has proved to be a very robust algorithm. Closer scrutiny of the algorithm reveals that it has very high breakdown point. It is therefore necessary to evaluate the other robustness features e.g. gross error sensitivity, local shift sensitivity, efficiency, asymptotic variance, finite outlier rejection etc. The influence function (IF) of the estimator is required to be evaluated to determine all these robustness properties.

In this research, the objects from range image have been recognized in the form of several clusters represented by quadric surfaces. These surfaces are then required to be combined to create the final solid model of single identity. In real application, objects with varied shape will obviously require complex combination of clusters of quadric surfaces. It is therefore necessary to formulate more reliable method cluster merging while developing the real entity.

Only difficulty encountered in application of NLS algorithm is to classify the data points at the intersection of two clusters represented by quadric surfaces. Typically the points in a range image, located on the edge of an object are often get removed by the cluster detected first, which should evenly belong to two or more clusters. Although it is suggested to applying one step noise c-quadric shell, NCQS algorithm on NLS results, however a better strategy which can be incorporated to the sampling scheme of NLS is necessary.

One of the primary requirements of fuzzy clustering method is that it requires the number of clusters required being specified apriori. In [29] a new competitive agglomerative clustering method has been derived for object data that automatically determines the optimum number of clusters during classification. This concept requires to be extended to FRC and N-FRC algorithm to solve the problem of number clusters in relational data space.

As we know clustering is an essential tool for various areas of scientific and engineering activity. There has been considerable research in fuzzy clustering in the last two decades that has produced a number of algorithms to the solution of several research issues.

APPENDIX A

DSCRIPTION OF SUBROUTINE HYBRD IN MINPACK

```
/*   subroutine hybrd */

/*   the purpose of hybrd is to find a zero of a system of */
/*   n nonlinear functions in n variables by a modification */
/*   of the powell hybrid method. the user must provide a */
/*   subroutine which calculates the functions. the jacobian is */
/*   then calculated by a forward-difference approximation. */

/*   the subroutine statement is */

/*     subroutine hybrd(fcn,n,x,fvec,xtol,maxfev,ml,mu,epsfcn, */
/*                      diag,mode,factor,nprint,info,nfev,fjac, */
/*                      ldfjac,r,lr,qtf,wa1,wa2,wa3,wa4) */

/*   where */

/*     fcn is the name of the user-supplied subroutine which */
/*       calculates the functions. fcn must be declared */
/*       in an external statement in the user calling */
/*       program, and should be written as follows. */

/*       subroutine fcn(n,x,fvec,iflag) */
/*       integer n,iflag */
/*       double precision x(n),fvec(n) */
/*       ---------- */
/*       calculate the functions at x and */
/*       return this vector in fvec. */
/*       --------- */
/*       return */
/*       end */

/*       the value of iflag should not be changed by fcn unless */
/*       the user wants to terminate execution of hybrd. */
/*       in this case set iflag to a negative integer. */

/*     n is a positive integer input variable set to the number */
/*       of functions and variables. */

/*     x is an array of length n. on input x must contain */
```

```
/*	  an initial estimate of the solution vector. on output x */
/*	  contains the final estimate of the solution vector. */

/*	fvec is an output array of length n which contains */
/*	  the functions evaluated at the output x. */

/*	xtol is a nonnegative input variable. termination */
/*	  occurs when the relative error between two consecutive */
/*	  iterates is at most xtol. */

/*	maxfev is a positive integer input variable. termination */
/*	  occurs when the number of calls to fcn is at least maxfev */
/*	  by the end of an iteration. */

/*	ml is a nonnegative integer input variable which specifies */
/*	  the number of subdiagonals within the band of the */
/*	  jacobian matrix. if the jacobian is not banded, set */
/*	  ml to at least n - 1. */

/*	mu is a nonnegative integer input variable which specifies */
/*	  the number of superdiagonals within the band of the */
/*	  jacobian matrix. if the jacobian is not banded, set */
/*	  mu to at least n - 1. */

/*	epsfcn is an input variable used in determining a suitable */
/*	  step length for the forward-difference approximation. this */
/*	  approximation assumes that the relative errors in the */
/*	  functions are of the order of epsfcn. if epsfcn is less */
/*	  than the machine precision, it is assumed that the relative */
/*	  errors in the functions are of the order of the machine */
/*	  precision. */

/*	diag is an array of length n. if mode = 1 (see */
/*	  below), diag is internally set. if mode = 2, diag */
/*	  must contain positive entries that serve as */
/*	  multiplicative scale factors for the variables. */

/*	mode is an integer input variable. if mode = 1, the */
/*	  variables will be scaled internally. if mode = 2, */
/*	  the scaling is specified by the input diag. other */
/*	  values of mode are equivalent to mode = 1. */

/*	factor is a positive input variable used in determining the */
/*	  initial step bound. this bound is set to the product of */
/*	  factor and the euclidean norm of diag*x if nonzero, or else */
```

```
/*        to factor itself. in most cases factor should lie in the */
/*        interval (.1,100.). 100. is a generally recommended value. */

/*      nprint is an integer input variable that enables controlled */
/*        printing of iterates if it is positive. in this case, */
/*        fcn is called with iflag = 0 at the beginning of the first */
/*        iteration and every nprint iterations thereafter and */
/*        immediately prior to return, with x and fvec available */
/*        for printing. if nprint is not positive, no special calls */
/*        of fcn with iflag = 0 are made. */

/*      info is an integer output variable. if the user has */
/*        terminated execution, info is set to the (negative) */
/*        value of iflag. see description of fcn. otherwise, */
/*        info is set as follows. */

/*        info = 0   improper input parameters. */

/*        info = 1   relative error between two consecutive iterates */
/*                   is at most xtol. */

/*        info = 2   number of calls to fcn has reached or exceeded */
/*                   maxfev. */

/*        info = 3   xtol is too small. no further improvement in */
/*                   the approximate solution x is possible. */

/*        info = 4   iteration is not making good progress, as */
/*                   measured by the improvement from the last */
/*                   five jacobian evaluations. */

/*        info = 5   iteration is not making good progress, as */
/*                   measured by the improvement from the last */
/*                   ten iterations. */

/*      nfev is an integer output variable set to the number of */
/*        calls to fcn. */

/*      fjac is an output n by n array which contains the */
/*        orthogonal matrix q produced by the qr factorization */
/*        of the final approximate jacobian. */

/*      ldfjac is a positive integer input variable not less than n */
/*        which specifies the leading dimension of the array fjac. */
/*      r is an output array of length lr which contains the */
/*        upper triangular matrix produced by the qr factorization */
/*        of the final approximate jacobian, stored rowwise. */

/*      lr is a positive integer input variable not less than */
/*        (n*(n+1))/2. */

/*      qtf is an output array of length n which contains */
/*        the vector (q transpose)*fvec. */

/*      wa1, wa2, wa3, and wa4 are work arrays of length n. */

/*     subprograms called */

/*      user-supplied ...... fcn */

/*      minpack-supplied ... dogleg,dpmpar,enorm,fdjac1, */
/*                           qform,qrfac,r1mpyq,r1updt */

/*      fortran-supplied ... dabs,dmax1,dmin1,min0,mod */

/*     argonne national laboratory. minpack project. march 1980. */
/*     burton s. garbow, kenneth e. hillstrom, jorge j. more */
```

APPENDIX B

THE RELATION BETWEEN FRC AND FCM ALGORITHM

The fuzzy clustering technique was proposed by Bezdek [4] is primarily known as *fuzz c-means* or FCM method where c is the pre-specified number of clusters is attempted to be detected. This algorithm is actually the fuzzy generalization of the classical *c-means* approach of hard clustering. This method is based on the minimization of the objective function as stated below

$$J=\sum_{i=1}^{n}\sum_{v=1}^{k}u_{iv}^{2}\|x_i-m_v\|^2=\sum_{i=}^{n}\sum_{v=1}^{k}u_{iv}^{2}\sum_{f=1}^{p}\left(x_{if}-m_{vf}\right)^2 \qquad \text{(B.1)}$$

where p is the number of variables in the data, u_{iv} represents the fuzzy membership i th data point to the v th cluster and the cluster centers are represented as

$$m_{vf}=\frac{\sum_i u_{iv}^2 x_{if}}{\sum_i u_{iv}^2} \qquad \text{(B.2)}$$

It is possible to replace the expression of m_{vf} from (B.2) into (B.1). Hence

$$J=\sum_{i=1}^{n}\sum_{v=1}^{k}u_{iv}^{2}\sum_{f=1}^{p}\left(x_{if}-\frac{\sum_{j=1}^{n}u_{jf}^{2}\cdot x_{jf}}{\sum_{j=1}^{n}u_{jf}^{2}}\right)^2$$

by arranging the terms, we get

$$=\sum_{v=1}^{k}\sum_{f=1}^{p}\left[\sum_{i=1}^{n}u_{iv}^{2}-2\frac{\sum_{i=1}^{n}u_{iv}^{2}x_{if}\sum_{j=1}^{n}u_{jf}^{2}x_{jf}}{\sum_{j=1}^{n}u_{jf}^{2}}+\frac{\left(\sum_{j=1}^{n}u_{jf}^{2}\right)^2}{\sum_{j=1}^{n}u_{jf}^{2}}\right]$$

Combining the suffix of *i* and *j* inside the parenthesis, we get

$$= \sum_{v=1}^{k}\sum_{f=1}^{p}\left[\sum_{j=1}^{n} u_{jv}^{2} x_{jf}^{2} - \frac{\left(\sum_{j=1}^{n} u_{jf}^{2} x_{jf}\right)^{2}}{\sum_{j=1}^{n} u_{jv}^{2}}\right]$$

By linearizing the terms inside the parenthesis, it results in

$$= \sum_{v=1}^{k}\sum_{f=1}^{p}\frac{\sum_{j=1}^{n} u_{jv}^{2}\sum_{i=1}^{n} u_{iv}^{2} x_{if}^{2} - \sum_{i=1}^{n}\sum_{j=1}^{n} u_{iv}^{2} u_{jv}^{2} x_{if} x_{jf}}{\sum_{j=1}^{n} u_{jv}^{2}}$$

Since the summation of *i* and *j* series terms from 1 to *n*, we rearrange the terms as

$$= \sum_{v=1}^{k}\sum_{f=1}^{p}\frac{\sum_{i=1}^{n}\sum_{j=1}^{n} u_{iv}^{2} u_{jv}^{2}\left(x_{if} - x_{jf}\right)^{2}}{2\sum_{j=1}^{n} u_{jv}^{2}}$$

Finally, we can get

$$J = \sum_{v=1}^{k}\frac{\sum_{i=1}^{n}\sum_{j=1}^{n} u_{iv}^{2} u_{jv}^{2}\left\|x_{i} - x_{j}\right\|^{2}}{2\sum_{j=1}^{n} u_{jv}^{2}} \qquad (B.3)$$

The last expression is exactly the objective function of FANNY algorithm [49]. If the measurement consist of position of object in p dimensional space, it is possible implement L_2 norm distance. Then the above derivation proves the equivalence of result by both FANNY and FCM algorithm.

APPENDIX C

DERIVATION OF EQUATIONS (5.66) AND (5.68)

First, consider (5.65) rewritten as below.

$$\psi_{ij} = \frac{\sum_{w=1}^{c+1}\left(\psi_{wj}/b_{wj}\right) - 1}{\sum_{w=1}^{c+1}\left(1/b_{wj}\right)} \text{ for } i \in I^{-} \tag{C-1}$$

Next we split the summation terms into two sets, I^{+} and I^{-}.

$$\psi_{ij} = \frac{\sum_{w \in J^{-}}\left(\psi_{wj}/b_{wj}\right) + \sum_{w \in J^{+}}\left(\psi_{wj}/b_{wj}\right) - 1}{\sum_{w \in J^{-}}\left(1/b_{wj}\right) + \sum_{w \in J^{+}}\left(1/b_{wj}\right)} \tag{C-2}$$

Since $\psi_{ij} = 0$ for $i \in I^{+}$, by simplification we get the following.

$$\psi_{ij} \sum_{w \in I^{-}}\left(1/b_{wj}\right) + \psi_{ij} \sum_{w \in I^{+}}\left(1/b_{wj}\right) = \psi_{ij} \sum_{w \in I^{-}}\left(1/b_{wj}\right) - 1 \tag{C-3}$$

Terms cancel out from right hand and left hand sides, and thus (C-3) can be further simplified to obtain the following which is same as equation (5.66).

$$\psi_{ij} = -\frac{1}{\sum_{w \in I^{+}}\left(1/b_{wj}\right)} \text{ for } i \in I^{-} \tag{C-4}$$

Next, we consider derivation of (5.68). Reconsider (5.67) as shown below.

$$u_{ij} = \frac{1/b_{ij}}{\sum_{w=1}^{c+1} 1/b_{wj}} - \frac{\sum_{w=1}^{c+1}\left(\psi_{wj}/b_{wj}\right)}{b_{ij}\sum_{w=1}^{c+1}\left(1/b_{wj}\right)}$$

This can be simplified as below.

$$u_{ij} = \frac{1 - \sum_{w=1}^{c+1}\left(\psi_{iw}/b_{iw}\right)}{b_{ij}\sum_{w=1}^{c+1}\left(1/b_{iw}\right)} \tag{C-5}$$

Once again the terms can be split into two sets, I^+ and I^-.

$$u_{ij} = \frac{1 - \sum_{w \in I^-} \left(\psi_{iw} / b_{iw} \right) - \sum_{w \in I^+} \left(\psi_{iw} / b_{iw} \right)}{b_{ij} \sum_{w=1}^{c+1} \left(1/b_{iw} \right)} \tag{C-6}$$

Noting again that $\psi_{ij} = 0$ for $i \in I^+$, (C-6) can be simplified as,

$$u_{ij} = \frac{1 - \psi_{ij} \sum_{w \in I^-} \left(1/b_{iw} \right)}{b_{ij} \sum_{w=1}^{c+1} \left(1/b_{iw} \right)}. \tag{C-7}$$

substituting the value of ψ_{ij} from (C-4), the following is obtained.

$$u_{ij} = \frac{1 + \dfrac{\sum_{w \in I^-} \left(1/b_{wj} \right)}{\sum_{w \in I^+} \left(1/b_{wj} \right)}}{b_{ij} \sum_{w=1}^{c+1} \left(1/b_{iw} \right)} \tag{C-8}$$

By arranging the terms of the numerator and through further simplification the following which is same as (5.68) is obtained.

$$u_{ij} = \frac{1/b_{ij}}{\sum_{w \in I^+} \left(1/b_{wj} \right)} \text{ for } i \in I^+ \qquad \text{(C-9}$$

REFERENCES

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

2. M.Barni, V. Cappellini, and A. Mecocci, "Comments on a possibilistic approach to clustering," *IEEE Trans. on Fuzzy Systems*, vol. 4, pp. 393-396, 1996.

3. J. Besl, and R. C. Jain, "Invariant surface characteristics for 3D object recognition in range images," *Computer Vision, Graphics and Image Processing*, vol. 33, pp. 33-80, 1986.

4. J. C. Bezdek, "Fuzzy mathematics in pattern classification," Ph.D. Thesis, Applied Math. Center, Cornell University, Ithaca, NY, 1973.

5. J. C. Bezdek, "Cluster validity with fuzzy set," *J. Cybernetics*, vol. 3, pp. 58-73, 1974.

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

7. J. C. Bezdek, R.. J. Hathaway, and M. P. Windham, "Numerical comparison of the RFCM and AP algorithms for clustering relational data," *Pattern Recognition*, vol. 24, pp. 783-791, 1991.

8. J. C. Bezdek and S. K. Pal, (eds) *Fuzzy Models for Pattern Recognition*, IEEE Press, New York, 1992.

9. M. P. Chandrasekharan and R. Rajagopalan, "An ideal seed non-hierarchical clustering algorithm for cellular manufacturing," *International J. of Production Research*, vol. 24, pp. 451-464, 1986.

10. M. P. Chandrasekharan and R. Rajagopalan, "GROUPABILITY: an analysis of the property of binary data metrices for group technology," *International J. of Production Research*, vol. 27, pp. 1035-1052, 1989.

11. C. H. Chu and J. C. Hayya, "A fuzzy clustering approach to manufacturing cell formation," *International J. of Production Research*, vol. 29, pp. 1475-1487, 1991.

12. M. Cohen and G. T. Toussaint, "On the detection of structures in noisy pictures," *Pattern Recognition*, vol.9, pp. 95-98, 1977.

13. R. N. Davé, "Use of the adaptive fuzzy clustering algorithm to detect lines in digital images," in *Proceedings of Intelligent Robots and Computer Vision VIII*. 1192, pp. 600-611, May 1989.

14. R. N. Davé and S. Bhamidipati, "Application of fuzzy shell-clustering algorithm to recognize circular shapes in digital images,"in *Proceedings of Third Annual IFSA World Congress*, pp. 238-241, 1989.

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

16. R. N. Davé and K. J. Patel, "Progressive fuzzy clustering algorithms for characteristic shape recognition,"in *Proceedings of NAFIPS 90*, pp. 121-124, 1990.

17. R. N. Davé and K. J. Patel, "Fuzzy ellipsoidal-shell clustering algorithm and detection of elliptical shapes," in *Proceedings of Intelligent Robots and Computer Vision IX: Algorithms and Techniques*, vol. 1381, pp. 320-333, 1990.

18. R. N. Davé, "Characterization and detection of noise in clustering," *Pattern Recognition Letters*, vol. 12, pp. 657-664, 1991.

19. R. N. Davé, "New measures for evaluating fuzzy partitions induced through c-shells clustering," in *Proceedings of the SPIE Conference on Intelligent Robot and Computer Vision X: SPIE* 1607, pp. 406-414, 1991.

20. R. N. Davé and K. Bhaswan, "Adaptive fuzzy c-shells clustering and detection of ellipses," *IEEE Trans. on Neural Networks*, vol. 3, pp. 5-12, 1992.

21. R. N. Davé, "Robust fuzzy clustering algorithms," in *Proceedings of Second IEEE International Conference on Fuzzy Systems*, San Francisco, CA, pp. 1281-1286, 1993.

22. R. N. Davé and T. Fu, "Robust shape detection using fuzzy clustering: practical applications," *Fuzzy Sets and Systems*, vol. 65, pp. 161-185, 1994.

23. R. N. Davé and R. Krishnapurum, "M-estimators and robust fuzzy clustering," in *Proceedings of 1996 Biennial Conference of the NAFIPS*, Berkeley, CA, June 19-22, pp. 400-404, 1996.

24. R. N. Davé and R. Krishnapurum, "Robust clustering methods: A unified view," *IEEE Trans. Fuzzy Systems*, vol. 5, pp.270-293, 1997.

25. R. N. Davé and S. Sen, "On generalizing the noise clustering algorithm," in *Proceedings. Seventh International Fuzzy Systems Association World Congress: IFSA '97*, Prague, Czech Republic, pp. 205-210, June, 1997.

26. R. N. Davé and S. Sen, "Robust fuzzy relational clustering," submitted to *IEEE Transaction on Fuzzy Systems*.

27. S. Dhar, Associate Professor, Mathematics Dept., NJIT, Newark, NJ, Private communication 1996-1997.

28. R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.

29. H. Frigui and R. Krishnapuram, "A robust competetive clustering algorithm with applications in computer vision," to appear in *IEEE Trans. PAMI* in 1998.

30. T. Fu and R. N. Davé, "Segmenting curved surfaces from range images," in *Proceedings of Ninth International Conference on CAD/CAM, Robotics, and Factories of the Future, ISPE*, August 1993.

31. T. Fu, "Robust clustering for object recognition," Ph.D. Dissertation, dept. of Mechanical Engineering, New Jersey Institute of Technology, Newark, NJ 1995.

32. I.Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. PAMI*, vol. 11, pp. 773-781, 1989.

33. N. N. G. Gindy, T. M. Ratchet, and K. Case, "Component grouping for GT applications a fuzzy clustering approach with validity measure," *International J. Production Research*, vol. 33, pp. 2493-2509, 1995.

34. K. C. Gowda and E. Diday, "Symbolic clustering using a new similarity measure," *IEEE Trans. System Man. Cybernetics*, vol. 22, pp. 368-378, 1992.

35. M. P. Groover and E. W. Zimmers, Jr., *CAD/CAM: Computer-Aided Design and Manufacturing*, Prentice-Hall, New Jersey, 1984.

36. J. J. De Gruijter and A. B. McBratney, "A modified fuzzy k-means method for predictive classification," in Classification and Related Methods of Data Analysis, H. H. Bock (editor), Elsevier Science Publishers B. V., North Holland, 1988.

37. F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Function*, John Wiley, New York, 1986.

38. F. R. Hampel, "Beyond location parameters: robust concepts and methods," *Bull. Int. Stat. Inst.*, vol. 46, pp. 375-382, 1975.

39. R. J. Hathaway and J.C. Bezdek, "Optimization criteria by reformulation," *IEEE Trans. Fuzzy Systems*, vol. 3, pp. 241-245, 1995.

40. R. J. Hathaway and J. C. Bezdek, "NERF *c*-means: non-euclidean relational fuzzy clustering," *Pattern Recognition*, vol. 27, pp. 429-437, 1994.

41. R. J. Hathaway, J. C. Bezdek, and J.W. Devenport, "On relational data versions of *c*-means algorithms," *Pattern Recognition Letters*, vol. 17, pp. 607-612, 1996.

42. R. J. Hathaway, J.W. Devenport, and J. C. Bezdek, "Relation duals of the *c*-means clustering algorithms," *Pattern Recognition*, vol. 22, pp. 205-212, 1988.

43. R. Hoffman and A. K. Jain, "Segmentation and classification of range images," *IEEE Trans. PAMI.*, vol. 9, pp. 608-620, 1987.

44. B. K. P. Horn, *Robot Vision*, MIT Press, Cambridge, Massachusetts, 1979.

45. P. V. C. Hough, "A method and means for recognizing complex patterns," U.S. Patent No. 3069654, 1962.

46. P. J. Huber, *Robust Statistics*, John Wiley & Sons, New York, 1981.

47. P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. A (6), pp. 789-794, 1964.

48. J. Jolion and A. Rosenfeld, "Cluster detection in background noise," *Pattern Recognition*, vol. 22, pp. 603-607, 1989.

49. L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data*, John Wiley & Sons, New York, 1990.

50. P. R. Kersten, "The fuzzy median and the fuzzy MAD," in *Proceedings ISUMA*, College Park, Maryland, pp. 85-88, 1995.

51. J. Kim, R. Krishnapuram, and R. Dave, "Application of the least trimmed square technique to prototype-based clustering," *Pattern Recognition Letter*, vol. 17, pp. 633-641, 1996.

52. R. Krishnapuram, H. Frigui, and O. Nasraoui, "A fuzzy clustering algorithm to detect planar and quadric shpaes," in *Proceedings of the North American Fuzzy Information Processing Society Workshop*, Puerto Vallarta, Mexico, December 1992.

53. R. Krishnapuram, O. Nasraoui, and H. Frigui, "The fuzzy c spherical shells algorithms: A new approach," *IEEE Trans. on Neural Networks*, vol. 3, pp. 663-671, Sept. 1992.

54. R. Krishnapuram, H. Frigui, and O. Nasraoui, "Quadratic shell clustering algorithm and the detection of second-degree curves," *Pattern Recognition Letters*, vol. 14, pp. 545-552, July 1993.

55. R. Krishnapuram and J. M. Keller, "A Possibilistic approach to clustering," *IEEE Trans. on Fuzzy Systems*, vol. 1, pp. 98-110, 1993.

56. R. Krishnapuram, H. Frigui, and O. Nasraoui, "Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation: part I," *IEEE Trans. on Fuzzy Systems*, vol. 3, pp. 29-43, 1995.

57. R. Krishnapuram, H. Frigui, and O. Nasraoui, "Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation: part II," *IEEE Trans. on Fuzzy Systems*, vol. 3, pp. 44-60, 1995.

58. R. Krishnapuram and J. M. Keller, "The Possibilistic c-means algorithm: insights and recommendations," *IEEE Transactions on Fuzzy Systems*, vol. 4, pp. 385-393, 1996.

59. K. R. Kumar and A. Vannelli, "Strategic subcontracting for efficient desegregated manufacturing," *International J. of Production Research*, vol. 25, pp. 1715-1728, 1987.

60. A. Leroy and P. J. Rousseeuw, "PROGRESS: A program for robust regression analysis," in *Technical Report 201, Center for Statistics and O.R.*, University of Brussels, Belgium, 1984.

61. J. Mulvey and H. Cowder, "Cluster analysis: An application of lagrangian relaxation," *Management Sci.*, vol. 25, pp. 329-340, 1979.

62. M. R. Rao, "Cluster analysis and mathematical programming," *J. American Statistical Assoc.*, vol. 66, pp. 622-626, 1971.

63. M. Roubens, "Pattern classification problems and fuzzy sets," *Fuzzy Sets and Systems*, vol. 1, pp 239-253, 1978.

64. P. J. Rousseeuw, "Multivariate estimation with high breakdown point," in *Fourth Pannonian Symposium on Mathematical Statistics and Probability*, Bad Tazmannsdorf, Austria, Sept. 4-9, 1983.

65. P. J. Rousseeuw, "Least median of squares regression," *J. American Statistical. Assoc.*, vol. 79, pp. 871-880, 1984.

66. P. J. Rousseeuw, and A. M. Leroy, *Robust Regression and Outlier Detection*, John Wiley & Sons, New York, 1987.

67. E. Ruspini, "Numerical methods for fuzzy clustering," *Information Science*, vol. 2, pp. 319-350, 1970.

68. H. C. Schildt, *C - The Complete Reference*, McGraw-Hill, New York, 1988.

69. S. Sen and R. N. Davé, "Clustering of relational data containing noise and outliers," accepted, *IEEE Conference on Fuzzy Systems*, Anchorage, Alaska, May 1998.

70. M. S. Shunmugam, "On assessment of geometric errors," *International J. of Production Research*, vol. 24, pp. 413-425, 1986.

71. G. Taubin, "Estimation of planer curves, surfaces, and nonplaner space curves defined by implicit equations with application to edge and range segmentation, " *IEEE Trans. on PAMI*, vol. 13, pp. 1115-1138, 1991.

72. W. T. Tucker, "Counterexamples to the convergence theorem for the fuzzy c-means clustering algorithms," in *Analysis of Fuzzy Information, J. Bezdek (ed.)*, vol. III, CRC Press, pp. 109-121, 1982.

73. H. Vinod, "Integer programming and theory of grouping," *J. American Statististical Assoc.*, vol. 64, pp. 506-517, 1969.

74. S. Viswanath, "A new approach for solving the P-median problem in group technology," *International J. Production Research*, vol. 34, pp. 2691-2700, 1996.

75. M. P. Windham, "Cluster validity for fuzzy clustering algorithms," *Fuzzy Sets and Systems*, vol. 5, pp. 177-185, 1981.

76. M. P. Windham, "Numerical classification of proximity data with assignment measures," *J. of Classification*, vol. 2, pp. 157-172, 1985.

77. L. Xu, E. Oja and P. Kultanen, "A new curve detection method: randomized Hough transform (RHT)," *Pattern Recognition Letters*, vol. 11, pp. 331-338, May 1990.

78. H. Xu, and H. P. Wang, "Part family formation for GT applications based on fuzzy mathetics," *International J. Production Research*, vol. 27, pp. 1637-1651, 1989.

79. L. Xu, Abstract report NSF/ARPA workshop on 3D object representation for computer vision, CUNY Graduate School and University center, Dec. 5-7 1995.