Fuzzy Clustering Algorithms and their Application to Medical Image Analysis

A dissertation submitted in partial fulfilment of the requirements
for the degree of **Doctor of Philosophy** of the **University of London**.

AHMED ISMAIL SHIHAB

Department of Computing
Imperial College of Science, Technology and Medicine
University of London, London SW7 2AZ.

DECEMBER 2000.

Abstract

The general problem of data clustering is concerned with the discovery of a grouping structure within a finite number of data points. Fuzzy Clustering algorithms provide a fuzzy description of the discovered structure. The main advantage of this description is that it captures the imprecision encountered when describing real-life data. Thus, the user is provided with more information about the structure in the data compared to a crisp, non-fuzzy scheme.

During the early part of our research, we investigated the popular Fuzzy c-Means (FCM) algorithm and in particular its problem of being unable to correctly identify clusters with grossly different populations. We devised a suite of benchmark data sets to investigate the reasons for this shortcoming. We found that the shortcoming originates from the formulation of the objective function of FCM which allows clusters with relatively large population and extent to dominate the solution. This led to a search for a new objective function, which we have indeed formulated. Subsequently, we derived a new so-called Population Diameter Independent (PDI) algorithm. PDI was tested on the same benchmark data used to study FCM and was found to perform better than FCM. We have also analysed PDI's behaviour and identified how it can be further improved.

Since image segmentation is fundamentally a clustering problem, the next step was to investigate the use of fuzzy clustering techniques for image segmentation. We have identified the main decision points in this process. Furthermore, we have used fuzzy clustering to detect the left ventricular blood pool in cardiac cine images. Specifically, the images were of the Magnetic Resonance (MR) modality, containing blood velocity data as well as tissue density data. We have analysed the relative impact of the velocity data in the goal of achieving better accuracy. Our work would be typically used for qualitative analysis of anatomical structures and quantitative analysis of anatomical measures.

Dedication

To my parents and sisters, with much love, appreciation, and affection.

Acknowledgments

While a thesis has a single author by definition, many people are responsible for its existence. Dr Peter Burger, my supervisor, is perhaps the most important of these people. Peter provided me with regular weekly meetings and many ideas. I wish to thank him sincerely for being very supportive and friendly throughout the whole of my PhD. I would also like to thank my examiners: Professor Michael Fairhurst of the Electronic Engineering Department, University of Kent, Canterbury and Professor Xiaohui Liu of the Department of Computer Science, Brunel University. I am grateful to Dr. Daniel Rückert for commenting extensively on an earlier draft of this thesis. I would also like to acknowledge Dr Guang-Zhong Yang for his help in my mock viva.

During my PhD journey I met a number of excellent people with whom I have become good friends and therefore made the journey particularly enjoyable. I would hope that we remain friends after we have all gone separate ways. Moustafa Ghanem: thank you for your wise and light-hearted chats. Daniel Rückert and Gerardo Ivar Sanchez-Ortiz: thank you for being special friends with whom boundaries faded. Ioannis Akrotirianakis: thank you for our many shared magical moments. Tarkan Tahseen: thank you for your friendship, inspiration, and all those netmaze sessions. Khurrum Sair: thank you for putting up with all sorts of inconveniences from me and for our new friendship. Outside of College, I would like to thank Atif Sharaf and Walid Zgallai for being my good (half-Egyptian!) Arab friends with whom I shared many a good time.

On a more personal level, I would like to thank my parents, Amaal and Ismail — their hard work gave me the opportunity to choose the path that led here — and my sisters, Fatima and Iman, for their continuous support and encouragement.

Last but not least, I must thank the Department of Computing, Imperial College for kindly allowing me to use its facilities even after the expiry of my registration period.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

This dissertation contributes to the subject area of Data Clustering, and also to the application of Clustering to Image Analysis. Data clustering acts as an intelligent tool, a method that allows the user to handle large volumes of data effectively. The basic function of clustering is to transform data of any origin into a more compact form, one that represents accurately the original data. The compact representation should allow the user to deal with and utilise more effectively the original volume of data. The accuracy of the clustering is vital because it would be counter-productive if the compact form of the data does not accurately represent the original data. One of our main contributions is addressing the accuracy of an established fuzzy clustering algorithm.

In this introductory Chapter, we provide brief descriptions of the subjects of our research, and establish the motivations and aims of the research we conducted. Section 1.5 provides a summary of the main research contributions presented in this dissertation. The Chapter concludes with an outline of the remainder of the dissertation.

1.1 Clustering

Research on Clustering is well-established; it dates back to the 1950s and is widely reported in various current journals. The research problem is concerned with discovering a grouping structure within a number of objects.

Typically, a set of numeric observations, or features, are collected of each object.[1] The collected feature-sets are aggregated into a list which then acts as the input to a chosen computational clustering algorithm. This algorithm then provides a description of the grouping structure which it has discovered within the objects. The description typically consists of a list containing, for every object, the cluster to which it has been assigned. The clusters would be identified by labels usually supplied by the user. In this way, a large number of seemingly disparate objects, once a number of features are extracted of them, can be organised into groups of approximately shared features.

Data clustering gained initial formal treatment as a sub-field of statistics. Systematic methods of clustering were required to be developed because the data may be large in size and therefore cumbersome to analyse and visualise. The computing revolution of the sixties and seventies gave momentum to this new field because, for the first time, computers enabled the processing of large amounts of data and took the burden of the very large amounts of computation generally involved. The field can, however, trace its origins to further back in time.

The Swedish botanist Carolus Linnaeus, who was concerned with classification in the plant and animal kingdom, wrote in his seminal 1737 work *Genera Plantarum* [Everitt, 1974]:

> All the real knowledge which we possess, depends on methods by which we distinguish the similar from the dissimilar. The greater number of nat-

[1] This is called *feature selection* and is studied in its own right, separately from clustering. Naturally, the selection of features strongly influences the effectiveness of whatever process takes place after the extraction of the features, be it clustering or otherwise.

> ural distinctions this method comprehends the clearer becomes our idea of things. The more numerous the objects which employ our attention the more difficult it becomes to form such a method and the more necessary.
>
> For we must not join in the same genus the horse and the swine, tho' both species had been one hoof'd nor separate in different genera the goat, the reindeer, and the elk, tho' they differ in the form of their horns. We ought therefore by attentive and diligent observation to determine the limits of the genera, since they cannot be determined *a priori*. This is the great work, the important labour, for should the Genera be confused, all would be confusion.

If translated to modern formalisms, Linnaeus's quotation is very relevant to the clustering problem. Linnaeus uses the term *natural distinction*; this is the much sought after goal of clustering — finding an "intrinsic classification" or an "inherent structure" in data. He states that the better we are at finding an inherent structure in data, the more knowledge we shall therefore possess about it. Furthermore, he states that the bigger the volume of data is (more numerous objects), the more necessary it is to develop better clustering methods. He mentions a key aspect of all clustering methods: little information is available *a priori*. Interestingly, the quotation emphasises the importance of *feature selection,* e.g., being one-hoofed doesn't put the horse and swine in the same genera. However, feature selection is considered to be out of the scope of the clustering problem in all modern studies.[2]

1.1.1 Clustering Applications

The explosion of sensory and textual information available to us today has caused many data analysts to turn to clustering algorithms to make sense of the data (thereby heeding Linnaeus's warning on "confusion"). It has become a primary tool for so-called knowledge discovery [Fayyad *et al.*, 1996a; Fayyad *et al.*, 1996b], data mining,

[2] There is a case though for its inclusion back into the clustering domain especially in concept-forming and machine learning applications, [Mirkin, 1999]. Our research, however, has followed the established distinction between *feature selection* and *clustering*.

and intelligent data analysis [Liu, 2000]. In fact, the massively-sized data sets of these applications have placed high demands on the performance of the computationally expensive clustering algorithms.

Clustering is used in various applications. In general, it can assist in [Backer, 1995]:

1. Formulating hypotheses concerning the origin of the data (e.g., evolution studies).

   Investigating clustering behaviour at various scales of measurement provides a hierarchical description of the data. The hierarchical description captures the early formation of clusters and how they break down to smaller ones and so on. This can aid in formulating hypotheses about the system generating the data, particularly in biological taxonomy applications. (See Chapter 2 for more details of hierarchical clustering.)

2. Describing the data in terms of a typology (e.g., market analysis).

   Profiles of consumers, including their purchasing behaviour, may cluster around a small number of "consumer types", this is then used to improve marketing performance.

3. Predicting the future behaviour of types of this data (e.g., modelling economic processes).

   If the temporal data tends to cluster, the predictive process can be simplified by identifying patterns of temporal behaviour based on clusters. This can be then generalised to similar types of data.

4. Optimising a functional process (e.g., information retrieval).

   Identifying clustering behaviour in demand-driven environments can help in optimising access to the resources under demand so that improved responsiveness is achieved.

1.1.2 Clustering Paradigms

Above, we mentioned one way of describing clustering structures within a number of objects, which is: assign a cluster label for every object. We can then, perhaps, do a search on a specific label to find out which objects belong to it. However, there are other ways of describing the discovered structure and this depends on the clustering paradigm being followed. These paradigms reflect the different assumptions and approaches taken by researchers in the field.

In Table 1.1 we list the five main clustering paradigms. We describe the main feature of each paradigm and give recent examples from the literature. Each of these paradigms is not exclusive and considerable overlap exists between them. In Chapter 2, we will concentrate on only the hierarchical and partitional paradigms.

1.1.3 Fuzzy Clustering

Our research has used the paradigm of fuzzy clustering which is based on the elements of fuzzy set theory. Fuzzy set theory employs the notion that for a given universe of discourse, every element in the universe belongs to a varying degree to all sets defined in the universe. In *fuzzy clustering*, the universe of discourse is all the objects and the sets defined on the universe are the clusters. Objects are not classified as belonging to one and only one cluster, but instead, they all possess a degree of membership with each of the clusters.

The most widely used fuzzy clustering algorithm is called fuzzy c-means, or FCM. In the five years between January 1995 to December 1999 there were 124 journal papers containing “fuzzy c-means” in their titles or abstracts. The subject areas of the journals were many and included Process Monitoring, Soil Science, and Protein Engineering. The papers were split between those reporting on an application of FCM and those reporting on improving its performance in some way. Being so widely used,

Paradigm	Description and Recent Literature
Hierarchical	Produces a tree-like description of the clustering structure. The tree is constructed by recursively merging similar objects to form clusters, then merging the clusters to form new super-clusters, this ends when all clusters have merged into one super-cluster. Cutting the tree at any level provides a partition of the objects. [Bajcsy & Ahuja, 1998], [ElSonbaty & Ismail, 1998]
Graph-theoretic	Views the objects as nodes in a weighted network, or graph. This is very helpful for two-dimensional dot patterns. The weight between one node and another is the distance between them using an appropriate metric. The problem, thus, becomes a graph-theoretic one where, for example, a minimal spanning tree is constructed on the dot pattern. This can help illustrate the clustering structure. [Brito *et al.*, 1997], [Pacheco, 1998], [Shapiro, 1995]
Mixture Models	Assumes the objects were generated by a mixture of probability distributions. Determination of the parameters of each distribution defines the clusters. [McLachlan & Basford, 1988], [Fraley & Raftery, 1998], [Banfield & Raftery, 1993]
Partitional	Clusters are disjoint partition of objects. An object belongs to only one cluster; crisp membership. Usually employs notion of prototypes around which objects cluster, and an objective function to assess a given partition. [Lin & Lin, 1996], [AlSultan & Khan, 1996]
Fuzzy	An object possesses varying degrees of membership with more than one cluster. Extends partitional paradigm, but extensions for all other paradigms are being proposed. [Bezdek, 1981], [Hoppner *et al.*, 1999], this dissertation

Table 1.1: Clustering Paradigms with examples of recent literature.

FCM is known to have certain shortcomings. We will be discussing these shortcomings and proposing our own solution to a particular but important shortcoming later in this thesis.

1.2 Image Analysis

Today, *imaging* plays an important role in medical diagnosis, and in planning, executing, and evaluating surgical and radiotherapeutical procedures. The information extracted from images may include functional descriptions of anatomical structures, geometric models of anatomical structures, or diagnostic assessment.

Most medical imaging modalities provide data in two spatial dimensions (2D) as well as in time (2D + time cine sequence). Data in three spatial dimensions (3D) as well as in time (3D + time, so-called 4D) are also becoming common. The large amount of data involved necessitates the identification or *segmentation* of the objects of interest before further analysis can be made. The result of this segmentation process is the grouping or labelling of pixels into meaningful regions or objects. Currently, segmentation is often carried out manually by experienced clinicians or radiologists.

There is a very strong intuitive similarity between clustering and segmentation; both processes share the goal of finding accurate classification of their input. Fuzzy clustering, therefore, has been used for image segmentation for the past twenty years [Pal & Pal, 1993; Bezdek *et al.*, 1993; Bezdek *et al.*, 1997]. The process of using clustering in image analysis is generally flexible and therefore a lot of decisions are taken ad-hoc. We will explore this process in Chapter 5 of this thesis. Also, in Chapter 6, we describe a specific application of fuzzy clustering to cardiac MR image analysis.

1.3 General Framework and Motivation

The ability to learn is an outstanding human faculty. This faculty allows us to interact and deal successfully with new situations and to improve our performance at whatever task we are performing. A simplified model of learning is that it is a process over time that uses its percepts, or perceptive input from sensors, to add continuously to and refine knowledge about its environment [Rumelhart *et al.*, 1986; Russel & Norvig, 1995].

The discipline of science concerned with designing computer programs that learn, so-called Machine Learning, concentrates on supervised learning methods [Niyogi, 1995; Mitchell, 1997]. These methods must be presented with prior training examples so that they can perform in a successful manner when dealing with new data. The training examples consist of a finite number of input-output pairs. From this training set, the learning agent must discover the learning function so that when it is presented with unencountered data, it produces a “reasonable” output. The learning function represents the knowledge gained by the learning agent. Supervised methods, thus, assume the existence of a training set for the percepts of the learning agent. What about when there is no training set, as is often the case in early learning experiences?

Here, unsupervised learning methods must be used. These methods operate on only the input percepts because no training examples are available. They must work on the basis of minimal assumptions about the data. Thus, it is these methods that capture the formative part of learning most [Michalski & Stepp, 1983; Stepp & Michalski, 1986]. Unsupervised learning acts as an exploratory tool, a tool by means of which a preliminary model may be generated.

One of the primary unsupervised learning methods is clustering. Thus, the research we carried out was motivated by the desire to improve and develop clustering methods further so that better learning agents may be built.

Our research was also motivated by another interest. Human beings can by seeing a picture recognise things in it as well as learn new things about the scene. Studies of the human visual sytem suggest that one of the primary operations carried out is clustering of visual sensory data [Ahuja & Tuceryan, 1989; Mohan, 1992; Li, 1997]. The research we undertook, particularly in the application of clustering to image analysis, was motivated by the similarities between clustering and perceptual grouping in the human visual system.

1.4 Research Aims

The aims of our research are:-

1. To investigate the main fuzzy clustering algorithms and to identify their stengths and weaknesses.
2. To study the process of using clustering for image segmentation and analysis.
3. To apply the results of our research in a medical image analysis problem.

1.5 Main Research Contributions

Our main research contributions can be summarised as:-

1. We studied and investigated the FCM algorithm thoroughly and identified its main strengths and weaknesses.
2. We developed a systematic method for analysing FCM's classification accuracy when it is used to cluster data sets that contain clusters of very different sizes and populations.

3. We proposed a new algorithm, based on FCM, which performs far more accurately than FCM on data sets like those described above. We also investigated performance properties of our new algorithm.

4. We identified the main decision points encountered when applying clustering methods to image analysis.

5. We carried out a case study in which we applied fuzzy clustering as the main image analysis tool for a novel type of image in cardiac Magnetic Resonance Imaging (MRI).

We believe these contributions provide new understanding and methods in regard to our Research Aims.

1.6 Outline of this Dissertation

This dissertation can be viewed as constituting two parts: the first part is concerned with the clustering of data of any type, whereas the second part is concerned with the clustering of data extracted from images. Chapters 2, 3, and 4 focus on the first part, and Chapters 5 and 6 focus on the second part.

Chapter 2, *The Basics of Data Clustering*, furnishes the reader with the general framework of the data clustering problem. The nomenclature that we used throughout the dissertation is presented. Examples of data typically used in clustering papers are shown. Hierarchical and Partitional clustering are described. A brief outline of two well-established clustering algorithms is given in order to familiarise the reader with the approaches used in solving the clustering problem. Finally, a brief commentary on new ideas in the clustering literature is presented.

Chapter 3, *Fuzzy Clustering*, presents a critical review of the fuzzy clustering field, but particularly algorithms based on an objective function model and relating to FCM.

First, the FCM algorithm is examined in detail. Second, extensions and developments on FCM are briefly reviewed. The Chapter concludes with an overview of the weaknesses of FCM.

Chapter 4, *A New Algorithm for Fuzzy Clustering*, presents the Population Diameter Independent (PDI) algorithm. This is an algorithm we propose that alleviates one of the important weaknesses of the FCM algorithm which is its tendency to mis-classify a data set containing smaller clusters located close to larger ones. An experiment is presented to analyse FCM's shortcoming and to motivate the new algorithm, PDI. The name Population-Diameter Independent is given to the algorithm because its performance remains more accurate than FCM and independently from the populations and diameters of clusters involved. The Chapter concludes with a review of some of PDI's performance parameters.

Chapter 5, *Clustering of Medical Images*, discusses the application of fuzzy clustering algorithms to image analysis, particularly segmentation. We break the analysis process into feature extraction, clustering, and post-processing, giving our experiences with the decisions involved in each stage. Within this framework, we give examples of successful applications of this process. We also carry out a comparison between FCM and PDI on synthetic medical images and demonstrate PDI's strength in this regard.

Chapter 6, *Application to Medical Image Analysis*, presents the results of our work to analyse Magnetic Resonance cardiac images. The work aims to track the left ventricle in cine images of the heart. The types of image we used contain velocity data as well as tissue density data. We followed the framework we outlined previously and conclude by reporting our results on this novel application.

Chapter 7, *Conclusions and Further Work*, summarises the conclusions of our research and outlines several ideas for further work based on the results we achieved.

CHAPTER 2

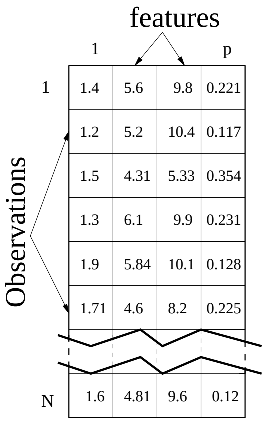
The Basics of Data Clustering

Stated simply, the clustering problem is:

> Given a collection of N objects, each of which is measured on each of p features, devise a grouping scheme for grouping the objects into c classes. The number of classes and the characteristics of the classes are unknown and should be determined.

In this Chapter, we expand on this definition and provide an introduction to the field. We defer the subject of Fuzzy Clustering to the next Chapter. Definitions of the nomenclature used for the remainder of the dissertation are provided in Section 2.1, and examples of dot patterns encountered in clustering literature are presented in Section 2.2.

Classically, clustering algorithms have been divided into Partitional and Hierarchical. In Section 2.3, hierarchical and partitional algorithms are described with the specific examples of the Single Link hierarchical algorithm and the Hard c-Means partitional (HCM) algorithm. The Chapter concludes with a brief review of new directions in clustering.



1	1.4	5.6	9.8	0.221
	1.2	5.2	10.4	0.117
	1.5	4.31	5.33	0.354
	1.3	6.1	9.9	0.231
	1.9	5.84	10.1	0.128
	1.71	4.6	8.2	0.225
N	1.6	4.81	9.6	0.12

Figure 2.1: An example of a data set.

2.1 Notation and Terminology

In general, we seek to cluster N objects, or observations. An observation may consist of a set of p numeric attributes or features. If that is the case, we name the collection of $N \times p$ values the **data set**. Figure 2.1 illustrates this concept.

Let the data set to be clustered be defined as $\mathcal{X}$. The set $\mathcal{X}$ consists of N **feature vectors** or **data points**, $\mathbf{x}$. Each $\mathbf{x}$ consists of p features such that $\mathbf{x} \in \mathcal{R}^p$.

$$\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}.$$

Assume we wanted to find c clusters in $\mathcal{X}$, where $2 \leq c < N$. In *crisp* clustering, the goal would be to partition $\mathcal{X}$ into the disjoint non-empty partitions $S_1, \cdots, S_c$ defined by:

$$\mathcal{X} = S_1 \cup S_2 \cup \ldots \cup S_c$$

where

$$S_i \cap S_j = \phi \qquad\qquad i, j \in \{1, \ldots, c\}, i \neq j$$

and

$$S_i \neq \phi \qquad\qquad i \in \{1, \ldots, c\}$$

In *fuzzy* clustering (described in detail in Chapter 3), the goal would be to find the **partition matrix**, $\mathcal{U}$. The partition matrix is a real $N \times c$ matrix that defines membership degrees for each feature vector. $\mathcal{U}$ is defined by:

$$\mathcal{U} \in \mathcal{R}^{N \times c} = [u_{ik}] \qquad\qquad i \in \{1, \ldots, c\}, k \in \{1, \ldots, N\}$$

where u_{ik} is the **degree of membership** of $\mathbf{x}_k$ in cluster i,

$$u_{ik} \in [0, 1] \qquad\qquad \forall i, k.$$

Clusters should contain feature vectors relatively similar to one another. In the general case, therefore, the results of a given clustering method very much depend on the **similarity measure** used. The similarity measure will provide an indication of proximity, likeness, affinity, or association. The more two data objects resemble one another, the larger a similarity index and, conversely, the smaller a dissimilarity index. In this sense, the Euclidean distance between two data vectors is a dissimilarity index, whereas the correlation is a similarity index.

Data sets may not always contain only numeric data. Many feature observations, especially data collected from humans, are binary. These would require an appropriate similarity measure like: *matching coefficients*. In some cases, feature observations would have been obtained from a time-series. An appropriate similarity measure should then take account of the temporal nature of these data. Furthermore, there are situations where the features would be of mixed types, or when data observations are missing. We refer the reader to [Backer, 1995] for an introduction to common ways of extracting similarity measures for binary, mixed, and missing data.

In some applications dissimilarity data are collected directly in the form of a **dis-**

similarity matrix. The dissimilarity matrix occurs as an $N \times N$ matrix whose entries measure dissimilarity between all pairs of N observations. It is also sometimes called the proximity data. This type of data set is commonly used as input to hierarchical clustering algorithms (described in the next Section). The dissimilarity matrix can in general be derived from the feature-vector data set by using, for example, Euclidean distance. The reverse transformation is not always possible and requires special ordination techniques [Everitt, 1978]. We do not use dissimilarity data in this dissertation.

Researchers in *Pattern Recognition* usually make a distinction between *clustering* and *classification*. This distinction is that clustering is an unsupervised process where no, or little, prior information is given on the classes in the data. On the other hand, the classification problem [Bishop, 1995; Mitchell, 1997] utilises pre-classified training data which is then used to deal with previously encountered data. For the rest of the dissertation, we will not consider classification, only clustering, but we will use the words *cluster* and *class* interchangeably.

Most partitional clustering methods utilise the concept that: for a given cluster i, there exists an ideal point $\mathbf{p}_i$, such that $\mathbf{p}_i \in \mathcal{R}^p$, which best represents cluster i's members. This point is called the **prototype** of the cluster. Thus, the clustering problem becomes that of finding a set of c prototypes,

$$\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_c\} \qquad \text{where} \qquad \mathbf{p}_i \in \mathcal{R}^p \quad \forall i \in \{1, \ldots, c\}$$

that best represent the clustering structure in $\mathcal{X}$.

We note that in the general case, prototypes are not restricted to points. This is so that they can better represent any possible cluster shape. For example, a ring-shaped cluster would be best represented with a circle-prototype. Further, a prototype may be composed of a set of points instead of a single point. However, choosing non-single-point prototypes renders the clustering problem harder. We do not delve into this in this dissertation. For now, we assume a set of single-point-prototypes as defined above.

The clustering algorithms we investigate in this thesis work on entirely numerical data and employ a **distance metric** to measure similarities between points, or between points and prototypes. For a given point $\mathbf{x}_k$ and a given prototype $\mathbf{p}_i$, by far the most common of all metrics is the Euclidean, or squared error one:

$$||\mathbf{x}_k - \mathbf{p}_i|| = \sqrt{(x_{k1} - p_{i1})^2 + \cdots + (x_{kp} - p_{ip})^2} \tag{2.1}$$

Another common, computationally simple, metric is the Manhattan (or taxicab) one.

$$||\mathbf{x}_k - \mathbf{p}_i|| = |(x_{k1} - p_{i1})| + \cdots + |(x_{kp} - p_{ip})| \tag{2.2}$$

The Mahalanobis metric is sometimes preferred to the Euclidean one because it is invariant to linear transformation of the data:

$$||\mathbf{x}_k - \mathbf{p}_i|| = (\mathbf{x}_k - \mathbf{p}_i)^T C_x^{-1} (\mathbf{x}_k - \mathbf{p}_i) \tag{2.3}$$

where C_x is the covariance matrix of $\mathcal{X}$. The price to pay for the scale-invariance of the Mahalanobis metric is the determination of covariance matrix and added computational complexity.

Clustering methods often employ an **objective function** to provide a numeric evaluation of a tentative clustering of the data set. Usually, this is employed within an iterative scheme where tentative solutions are evaluated to obtain progressively better partitions. This type of clustering methods is known as: objective-functional, or objective-function-based, and is strongly related to optimisation theory. Objective functions are usually formulated on the basis of distances. So, for example, the familiar “sum of squared errors” criterion may be translated to the clustering framework. More detail will be provided on this in the following Chapters.

2.2 Examples of Data Sets

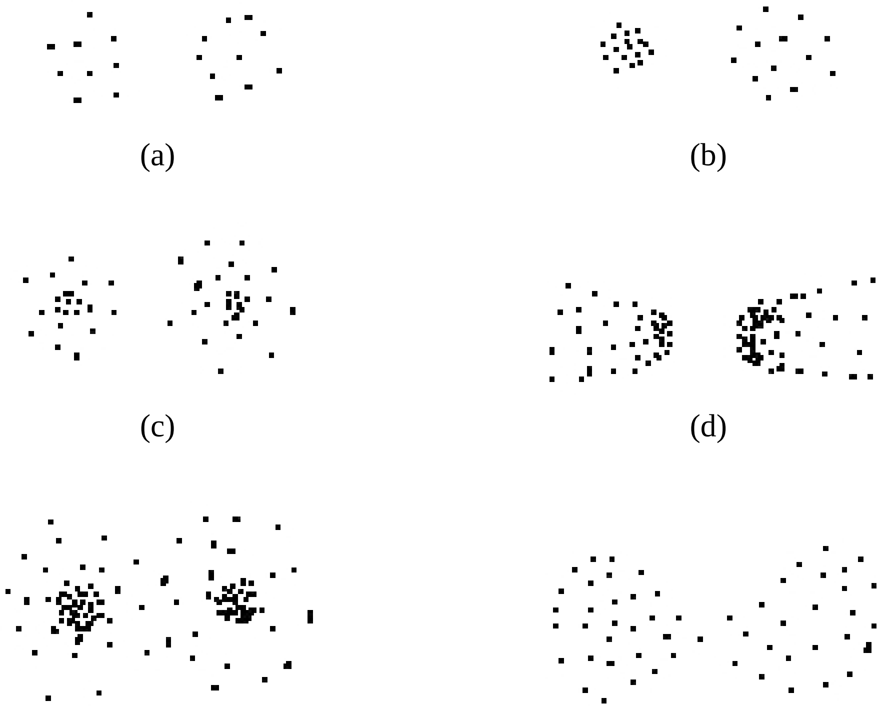

(a) (b) (c) (d) (e) (f)

Figure 2.2: Sample dot patterns of two clusters of varying densities and separation — reproduced from [Zahn, 1971].

In 1971, Charles Zahn wrote an influential paper on clustering using minimum spanning tree techniques, [Zahn, 1971]. In Figures 2.2-2.6, we show the same scatter plots of the examples of two-dimensional data sets that he presented in his paper. These data sets continue to present a challenge to researchers in clustering. In almost thirty years, no single clustering algorithm has been developed capable of identifying successfully the same clusters humans perceive in *all* of these plots.

The six dot patterns of Figures 2.2(a),(b), (c), (d), (e), and (f) show a pair of clusters but with different varying point densities and varying degrees of separation. In (a), the two clusters have approximately the same density. In (b), they have unequal densities. In (c), the densities vary proportionally to the distance from the mean. In (d), the clusters have smoothly varying densities, and the separation appears nearest the points of highest densities of the two classes. In (e) and (f), the separation between clusters becomes almost non-existent, as the clusters touch each other. These six dot patterns should not pose a problem to a lot of the established algorithms available today. However, in certain situations the accuracy of detected clustering structure may be compromised. Our research has examined this issue in detail and we shall describe our results in Chapter 4.

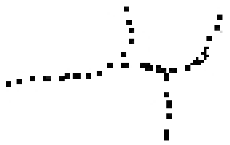


Figure 2.3: Sample dot pattern of linear, branch-like clusters.

Figure 2.3 shows a plot of clusters of linear fragments with a branch-like structure. Here, humans might themselves be unable to agree on whether there is any clustering and if so, what it is. However, given the information that plots of this kind consist of linear fragments, most of us would not have problems identifying the clustering structure. On the other hand, clustering algorithms that are specifically designed to detect linear cluster structures might fail.

Figure 2.4 shows a plot of two well defined clusters, but in a different type of pattern than that of Figures 2.2(a), (b), or (c). Here, the performance of many algorithms would be ad-hoc, depending on the length of each "string" of points and how far the strings are apart.

Figure 2.5 shows a plot of clusters with one class enclosed by the other, but both

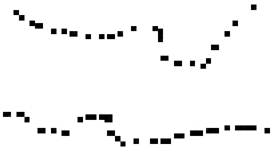


Figure 2.4: Sample dot pattern of linear, string-like clusters.

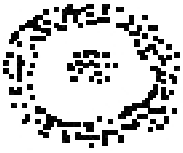


Figure 2.5: Sample dot pattern of ring-like and circular clusters.

well-defined. With the exception of the shell clustering algorithms, no other algorithms would be capable of handling ring-like patterns like this. Shell clustering is a recent development in cluster analysis and suffers from the fact that it looks for only shells. The non-ring shaped cluster within the shell in Figure 2.5 may confuse such algorithms.

Figure 2.6 shows a point pattern that may have been extracted from an image processing application. Region- or edge-based operators may have been applied to the original image and the resulting image then thresholded. Most clustering algorithms we know would fail with this point pattern because of the containment of one group of points within another.

Having realised the limited success achieved in Clustering so far, we should hasten to add that with regards to the point patterns of Figure 2.6 our expectations are mainly

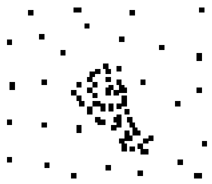


Figure 2.6: Sample dot pattern possibly extracted from an image after some image-processing stages.

dictated by human perception, rather than machine learning or knowledge discovery. It is debatable whether identifying the structure with a pattern like that of Figure 2.6 could be of use within such contexts.

2.3 Hierarchical and Partitional Clustering

Clustering methods tend to be divided in the literature into *hierarchical* and *partitional* methods. In hierarchical clustering (the older of the two), a tree-structured partitioning of the data set is produced. The tree is either constructed top-down or bottom-up, with an all-inclusive cluster at the top of it and the individual data points at the bottom of it. Different partitions may be suggested according to where we "cut" the tree.

In partitional clustering, only one suggested partition is produced. Partitional methods also usually produce prototypes, or typical representatives, of the clusters. These methods have become prevalent mainly due to their low computational cost.

2.3.1 Hierarchical Clustering

Hierarchical clustering algorithms transform a proximity data set into a tree-like structure which for historical reasons is called a *dendogram* [Jardine & Sibson, 1971]. The

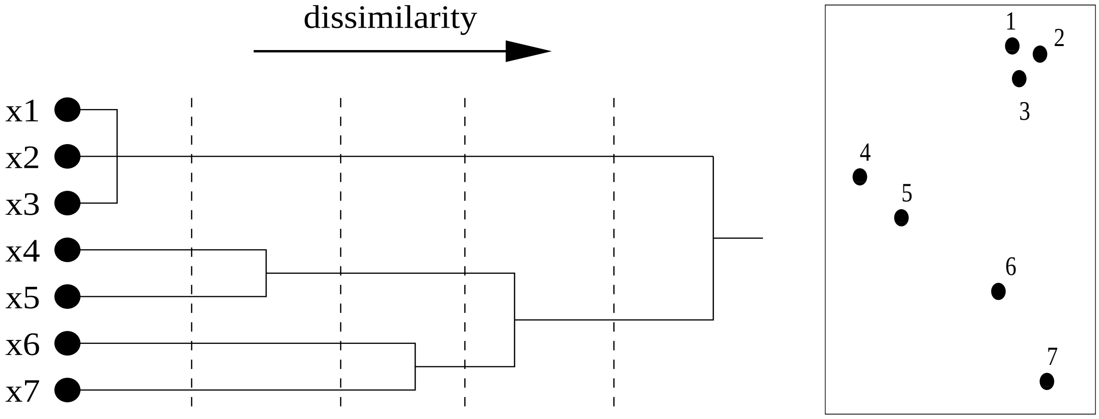


Figure 2.7: An example of the dendogram that might be produced by a hierarchical algorithm from the data shown on the right. The dotted lines indicate different partitions at different levels of dissimilarity.

dendogram is constructed as a sequence of partitions such that its root is a cluster covering all the points and the leaves are clusters containing only one point. In the middle, child clusters partition the points assigned to their common parent according to a dissimilarity level. This is illustrated in Figure 2.7. (We remark that the dendogram is not a binary tree.) The dendogram is most useful up to a few levels deep, as the clustering becomes more trivial as the tree depth increases.

Agglomerative clustering is a bottom-up way of constructing the dendogram. The hierarchical structure begins with N clusters, one per point, and grows a sequence of clusterings until all N observations are in a single cluster. *Divisive* clustering on the other hand is a top-down way of constructing the dendogram. The structure begins with one cluster containing all N points and successively divides clusters until N clusters are achieved.

Agglomerative hierarchical clustering is computationally less complex and, for this reason, it is more commonly used than divisive hierarchical clustering. A generic agglomerative hierarchical clustering technique would consist of the steps shown in Figure 2.8. Various algorithms can be constructed depending on the way in which the

```
1 Assign each data vector to a cluster
2 Find the smallest entry in the dissimilarity matrix and merge the corresponding
  two clusters
3 Update the dissimilarities between the new cluster and other clusters
4 Return to step (2) until all vectors are in one cluster
```

Figure 2.8: A generic agglomerative hierarchical algorithm.

dissimilarities are updated in step (3). In Section 2.3.2 we describe one such way.

Both agglomerative and divisive techniques suffer from the fact that if, say, at one point during the construction of the dendogram, a misclassification is made, it is built on until the end of the process. At some point of the dendogram's growth an observation may be designated as belonging to a cluster in the hierarchy. It remains associated with the successors of that cluster till the dendogram is finished. It is impossible to correct this misclassification while the clustering process is still on. Optimization of clusterings is then called for [Fisher, 1996].

After the tree has been produced, a multitude of possible clustering interpretations are available. A practical problem with hierarchical clustering, thus, is: at which value of dissimilarity should the dendogram be cut, or in other words, at which level should the tree be cut. One heuristic commonly used is to choose that value of dissimilarity where there is a large "gap" in the dendogram. This assumes that a cluster that merges at a much higher value of dissimilarity than that at which it was formed is more "meaningful". However, this heuristic does not work all the time [Jain, 1986].

2.3.2 Example: Single link algorithm

The Single Link Algorithm is an instantiation of the generic agglomerative clustering procedure and is one of many possible agglomerative algorithms. Its steps are outlined

1 The dissimilarity matrix, if unavailable, is calculated at first.

2 The smallest entry in the matrix is chosen, and the two points, a and b are fused together as one group.

3 The dissimilarity matrix is updated by reducing its size by one and recalculating the distances using the nearest neighbour rule. Thus for observation k and the newly formed (ab) cluster:

$$d_{(ab)k} = \min(d_{ak}, d_{bk})$$

4 Go back to step (2) until the matrix is 1x1.

Figure 2.9: The Single Link Algorithm

in Figure 2.9. For a description of other possible algorithms see [Everitt, 1974].

2.3.3 Partitional Clustering

Most partitional clustering algorithms assume *a priori* a number of clusters, c, and partition the data set into c clusters. Obviously, there can be many partitions of a given data set, but there will be only a few which identify the clustering in the data set. To arrive at a correct partition, an objective function can be formulated that measures how good a partition with respect to the data set is. If a given partition minimises the objective function (or maximises, depending on the formulation of the objective function), we assume that the optimal partition has been found. A generic partitional clustering technique would probably operate as in Figure 2.10.

Most objective function-based algorithms use c cluster prototypes to facilitate the evaluation of a given partition. Each prototype is assumed to be a typical representative of the group of points in that cluster. In the ideal case, each prototype will take the general shape of its cluster. In practice, however, most algorithms assume point prototypes because this simplifies the mathematics. Since objective functions are typically

```
1 Fix $c$, $2 \leq c < N$, choose the objective function you wish to minimise, and
  initialise the partition matrix
2 Evaluate the objective function, and modify the partition matrix accordingly
3 If consecutive partitions have not changed by a fixed threshold, stop, otherwise,
  return to step (2)
```

Figure 2.10: A generic partitional clustering algorithm.

non-linear, the optimal partition will usually have to be searched for algorithmically. The initial placement of the prototypes, thus, is important since there can be many suboptimal solutions that will trap the prototypes and terminate the algorithm.

Objective functions are specified using the data set, $\mathcal{X}$, a distance metric, d, the partition matrix, $\mathcal{U}$, and the set of cluster prototypes $\mathcal{P}$. The data set $\mathcal{X}$ and the metric d are fixed and act as input. $\mathcal{U}$ and $\mathcal{P}$ are variables whose optimal values are being sought. This can be represented mathematically as:

$$\min \quad [J(\mathcal{P},\mathcal{U};\mathcal{X},d,\ldots)]$$

where J is a generic objective function whose minimum value is being sought. The dots after D indicate that a given formulation of the objective function can use its own set of parameters. The squared error criterion, which minimises offsets between a prototype and its nearest points, is the most common formulation of the objective function.

2.3.4 Example: Hard c-Means (HCM)

The HCM algorithm has appeared in different equivalent versions over the years since its first appearance in the sixties. It was given the name *Hard* because it produces a crisp, or hard, partition (as opposed to fuzzy, or soft, partition, as described before).

Further, HCM shares the c-*means* part of its name with many prototype-based partitional algorithms. The reason is because they search for c prototypes, which intuitively are the means, or centroids, of the clusters they represent. The objective function minimised in this algorithm is:

$$J = \sum_{i=1}^{c} \left(\sum_{k,\mathbf{x}_k \in S_i} d_{ik}^2 \right) \; = \; \sum_{i=1}^{c} \left(\sum_{k,\mathbf{x}_k \in S_i} \|\mathbf{x}_k - \mathbf{p}_i\|^2 \right)$$

where S_i is the partition of $\mathcal{X}$ corresponding to cluster i, and d_{ik}^2 is a norm metric, usually the Euclidean distance, measuring the distances between the cluster prototypes and those data vectors belonging to it: $\mathbf{x}_k \in S_i$. In this way, J is the overall or total within-cluster sum of squared errors, and thus, carries a geometrically appealing rationale. The equation for determination of the prototypes is given by:

$$\mathbf{p}_i = \sum_{k=1}^{N} u_{ik}\mathbf{x}_k / \sum_{k=1}^{N} u_{ik} \tag{2.4}$$

where u_{ik} is either 0 or 1 depending on membership of $\mathbf{x}_k$ in S_i. From the equation we can see that the prototypes are the geometrical centroids of their respective cluster's data members. Equation 2.4 is arrived at by setting the gradient of J with respect to each $\mathbf{p}_i$ equal to zero. The derivation is similar to the one in Appendix C which is explained in detail later.

Most versions of HCM operate in the same way as the oldest and frequently cited algorithm of Forgy [Forgy, 1965] which is given in Figure 2.11. Its intuitive procedure is: guess c hard clusters, find their centroids, reallocate cluster memberships to minimise squared errors between the data and current prototypes; stop when looping ceases to lower J. Since the memberships are discrete, either 0 or 1, the notion of local minimum is not defined for J, and likewise convergence would be undefined.

There are probably hundreds of papers detailing the theory and applications of HCM (other names like ISODATA, k-means, etc, have also been used); [Duda & Hart, 1 c cluster prototypes (centroids), or equivalently, an initial partition is randomly generated

2 Each feature vector is assigned to the cluster of the nearest prototype

3 If no change of cluster memberships for all feature vectors, stop. Otherwise, calculate the centroids for the c new clusters according to equation 2.4, and go to (2)

Figure 2.11: The Hard c-Means Algorithm.

1973] surveys some of this literature. HCM suffers from the weakness of producing spurious solutions because the algorithm's iterative steps may not converge [Bezdek, 1980; Selim & Kamel, 1992]. It also does not provide the wealth of information fuzzy clustering provides.

2.4 Remarks

In this Chapter, we reviewed Clustering terminology and described our nomenclature. We also described some hierarchical and partitional clustering algorithms. In applying any clustering method, some issues need to be addressed — these include:

1 studying the raw data in terms of processing it, dealing with missing values in it, or deciding on the features to use,

2 determining the similarity measure that will be incorporated in the clustering process,

3 studying the parameter list of the algorithm and setting the parameters to appropriate values, perhaps revisting this step a number of times to experiment with different values,

4 for algorithms that require iteration, or consist of an optimisation procedure, reruns may be required to discover if different solutions will surface,

5 finally, some form of validation, or quantified cross-checking, of different solutions may be used to decide on the best solution.

Before moving to the main theme of our research, fuzzy clustering, we conclude this Chapter with examples of recent novel clustering approaches.

As was mentioned before, most partitional algorithms utilise a cluster prototype in their calculations. In general, it is not effective to describe a cluster using a single prototype if the cluster has an elongated or nonconvex shape. Examples of recent work to tackle this problem include [Chaudhuri & Chaudhuri, 1997] where more than one seed was used to describe a cluster if it passes a nonconvexity test, and [Tyree & Long, 1999] where linked line segments based on density linkage were used.

The notion of scale space was used for hierarchical clustering in [Roberts, 1996], producing good results. However, the problem of where to cut the resulting tree still persists. Scale space was also used in [Kothari & Pitts, 1999] to find and validate clustering results.

In conclusion, as was discussed in this Chapter, hierarchical methods are computationally costly and always suffer from the problem of not knowing where to cut the generated tree. Crisp partitional methods, while computationally inexpensive are notorious for getting trapped in spurious solutions. But both paradigms possess a further underlying shortcoming; this is their inability to describe effectively data sets with a clustering structure that is not crisp. It was this shortcoming that motivated the introduction of fuzzy clustering methods. Naturally, these open new research problems, as we see in the next two Chapters.

CHAPTER 3

Fuzzy Clustering

In the previous Chapter we described the general clustering problem and gave examples of crisp hierarchical and partitional algorithms.

In this Chapter, we describe fuzzy clustering algorithms particularly those related to the fuzzy c-means (FCM) algorithm. FCM's objective function has been generalised and extended as well as changed in several ways. For this reason, FCM is sometimes described as a model for fuzzy clustering. Our aim in this Chapter will be to define and describe the FCM model. We then describe several algorithms that are based on this model. We also highlight the strengths and shortcomings that these various algorithms have. In the next Chapter, we propose our own modification to FCM.

The concept of "fuzziness" underpins fuzzy clustering. By fuzziness is meant imprecision as to the exact class of an object. When we "fuzzy cluster" a data set we allow for data points to belong with varying degrees to more than one cluster. We briefly introduce Fuzzy Set Theory in the first Section of this Chapter in order to familiarise the reader with it.

3.1 Fuzzy Set Theory

Fuzzy Set Theory was developed by Lotfi Zadeh [Zadeh, 1965] in order to describe mathematically the imprecision or vagueness that is present in our everyday language. Imprecisely defined classes play an important role when humans communicate and learn. Despite this imprecision, humans still carry out sensible decisions. In order to deal with these classes, Zadeh introduced the concept of a *fuzzy set*. Fuzzy sets parallel ordinary mathematical sets but are more general than them in having a continuum of grades, or degrees, of membership.

Let X be a space of points, or objects. Let us denote any element of X by x. A fuzzy set A in X is now defined by a *membership function*, $f_A()$, which associates with each point in X a real number in the interval $[0, 1]$, with the value of $f_A(x)$ representing the "degree of membership" of x in A. The nearer the value of $f_A(x)$ to unity, the higher the degreee of membership of x in A.

Based on the above definition for the fuzzy set, extensions for definitions involving ordinary sets like *empty*, *equal*, *containment*, *complement*, *union*, and *intersection* have been proposed. We refer the reader here to the wide literature available on this matter [Kosko, 1993; Zadeh & Klir, 1996; Klir *et al.*, 1997; Cox, 1998].

In the fuzzy clustering setting, a cluster is viewed as a fuzzy set in the data set, $\mathcal{X}$. Thus each feature vector in the data set will have membership values with all clusters — membership indicating a degree of belonging to the cluster under consideration. The goal of a given fuzzy clustering method will be to define each cluster by finding its membership function.

In the general case, the fuzzy sets framework provides a way of dealing with problems in which the source of imprecision is the absence of sharply defined criteria of class membership rather than the presence of random variables. Fuzzy clustering fits well with the rest of the fuzzy sets and systems applications. It has been used with success in, for example, optimising membership functions for forming fuzzy inference rules, [Chi *et al.*, 1996; Chen & Wang, 1999].

Fuzzy set theory is widely used as a modeling tool in various Pattern Recognition and Image Analysis problems, [Rosenfeld, 1979; Philip *et al.*, 1994] for example, because of the relative ease with which it can be applied to a problem and the robustness of the resulting solution.

For a discussion of the future directions of fuzzy logic as seen by its founder see [Zadeh, 1995; Zadeh, 1996; Zadeh, 1999]. Fuzzy logic is seen ultimately as a methodology for *computing with words* (CW) in which words are used in place of numbers for computing and reasoning. The rationale for CW is that words become a necessity when the available information is too imprecise to justify the use of numbers. And also when there is a tolerance for imprecision which can be exploited to achieve tractability, robustness, low solution cost, and better human-computer interaction.

3.2 The Fuzzy c-Means Algorithm

The FCM algorithm took several names before FCM. These include Fuzzy ISODATA and Fuzzy k-Means. The idea of using fuzzy set theory for clustering is credited to Ruspini [Ruspini, 1969; Ruspini, 1970]. Dunn is credited with the first specific formulation of FCM, [Dunn, 1973], but its generalisation and current framing is credited to Bezdek, [Bezdek, 1981]. A collection of influential papers in the development of fuzzy clustering methods can be found in [Bezdek & Pal, 1992]. The FCM objective function and its generalisations are the most heavily studied fuzzy model in Pattern Recognition.

As mentioned in Section 2.1, we expect FCM to be a clustering algorithm that provides a fuzzy partition of the input data set. However, there is an infinite range of possible fuzzy partitions. Therefore, an optimisation model or objective function must be devised to search for the optimal partition according to the chosen objective function. FCM is, thus, first and foremost an objective function. The way that most researchers have solved the optimisation problem has been through an iterative locally-optimal technique, called the FCM algorithm. This is not the only way to solve the FCM objective function, for example, in [AlSultan & Selim, 1993] it is solved by the Simulated Annealing optimisation technique; in [Hathaway & Bezdek, 1995] the problem is reformulated and general optimisation methods are suggested for its solution; in [Al-Sultan & Fedjki, 1997] it is solved by a combinatorial optimisation technique called Tabu Search; in [Hall *et al.*, 1999] it is solved by the *genetic algorithm* which is an optimisation technique based on evolutionary computation; and in [Runkler & Bezdek, 1999] it is solved within an alternate optimisation framework. In fact, it is not impossible that an exact solution to the problem may be formulated.

3.2.1 FCM Optimisation Model

The formulation of the FCM optimisation model is :-

$$\text{Minimise } J_{FCM}(\mathcal{P},\mathcal{U};\mathcal{X},c,m) = \sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ik})^m d_{ik}^2(\mathbf{x}_k,\mathbf{p}_i) \tag{3.1}$$

$$\text{subject to the constraint } \sum_{i=1}^{c} u_{ik} = 1 \qquad \forall k \in \{1\ldots N\}, \tag{3.2}$$

where $\mathcal{P}$ and $\mathcal{U}$ are the variables whose optimal values are being sought. $\mathcal{X}$, c, and m are input parameters of J_{FCM}, where :-

- c is the number of clusters assumed to exist in $\mathcal{X}$.
- $m \geq 1$ is a fuzzification exponent that controls how fuzzy the result will be. The larger the value of m the fuzzier the solution. At $m = 1$ FCM collapses to HCM, giving crisp results. At very large values of m, all the points will have equal memberships with all the clusters.

- u_{ik} describes the degree of membership of feature vector $\mathbf{x}_k$ with the cluster represented by $\mathbf{p}_i$. $\mathcal{U} = [u_{ik}]$ is the $c \times N$ fuzzy partition matrix satisfying the constraint stated in Equation 3.2.

- N is the total number of feature vectors.

- d_{ik}^2 is the distance between feature vector $\mathbf{x}_k$ and prototype $\mathbf{p}_i$. The original formulation of FCM uses point prototypes and an inner-product induced-norm metric for d_{ik}^2 given by

$$d_{ik}^2(\mathbf{x}_k, \mathbf{p}_i) = \parallel \mathbf{x}_k - \mathbf{p}_i \parallel_A^2 = (\mathbf{x}_k - \mathbf{p}_i)^T A (\mathbf{x}_k - \mathbf{p}_i).$$

  A is any positive definite matrix which in the case of Euclidean distance is the identity matrix.

3.2.2 Conditions for Optimality

Let the minimisers of $J_{FCM}(\mathcal{P},\mathcal{U})$ be called $(\mathcal{P}^*,\mathcal{U}^*)$. The necessary conditions for $(\mathcal{P}^*,\mathcal{U}^*)$ are defined below. These conditions are derived in [Bezdek, 1981] and are similarly derived for the PDI algorithm in Appendix C.

$$\mathbf{p}_i^* = \frac{\sum_{k=1}^{N} u_{ik}^m \mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^m} \tag{3.3}$$

and

$$u_{ik}^* = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}^2}{d_{jk}^2}\right)^{1/(m-1)}} \tag{3.4}$$

The FCM algorithm is a sequence of iterations [1] through the equations above, which are referred to as the update equations. When the iteration converges, a fuzzy

[1] This is referred to as Picard iteration in [Bezdek, 1981]; Picard iteration [Greenberg, 1998] is a successive approximation scheme commonly used to solve differential equations, which starts with initial guesses of the variables and by means of successive substitution arrives at a solution.

c-partition matrix and the pattern prototypes are obtained. A proof of the convergence of the iterations to a local minimum can be found in [Bezdek, 1980; Selim & Kamel, 1992].

3.2.3 The Algorithm

See Figure 3.1.

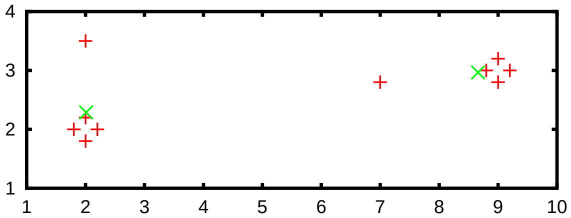
1. Fix c, $2 \leq c < N$; choose any inner product norm; fix m, $1 \leq m < \infty$; initialize the fuzzy membership matrix, $\mathcal{U}$.
2. Calculate c fuzzy cluster centers $\mathcal{P}$ as per Equation 3.3
3. Update memberships $\mathcal{U}$ as per Equation 3.4
4. Compare the change in the membership values using a appropriate norm; if the change is small, stop. Else return to 2.

Figure 3.1: The FCM algorithm

3.2.4 An Example

Let us give an example of FCM in action. Figure 3.2 shows the data set that we used as input to FCM ($c = 2$). The table on the right of Figure 3.2 tabulates the found partition matrix.

Whereas the solution is an approximately correct one, the locations of the found prototypes are not satisfactory since they should be at the centres of the diamond-like patterns. It is clear that the points located away from the diamond patterns have influenced FCM's solution in that they have "pulled" the prototypes away from the ideal locations. We note that, as expected, the membership values per each point add up to one.



Data		Memberships	
x	y	Cluster 1	Cluster 2
1.8	2	0.997	0.003
2.0	2.2	1.000	0.000
2.0	1.8	0.995	0.005
2.2	2	0.997	0.003
2.0	3.5	0.968	0.032
8.8	3	0.000	1.000
9.0	3.2	0.003	0.997
9.0	2.8	0.003	0.997
9.2	3	0.006	0.994
7	2.8	0.100	0.900

Figure 3.2: A 10-point data set with two clusters and two outlying points. Input data points are marked with a + and the prototypes found by FCM are marked with x. Membership values provided by FCM are tabulated on the right hand side. The found prototypes are at $(2.0, 2.2)$ and $(8.7, 3.0)$ instead of ideal placement at $(2.0, 2.0)$ and $(9.0, 3.0)$.

Outliers and Noise Points

We remark here on our definition of outlier and noise points. There is a lot of literature on outlier detection and rejection (see [Millar & Hamilton, 1999] for a recent review). In this dissertation, we took the view that every outlier point can be associated with one cluster in the data in the sense that it would be lying close to that cluster. Also, we took the view that the few points in a data set that cannot be said to be close to any cluster, be considered noise points. We recognise that a dense collection of outliers could become, at some scale, a "small" cluster of its own, but we operate on the assumption that the number of outliers is insignificant and that we already know the correct number of clusters.

In general, we perceive that outliers should be recognised as "satellite" points to a given cluster and given an appropriately high degree of membership with that cluster. However, their presence should not affect the accuracy of determining the location of the clusters. For noise points, we perceive that they should not receive significant memberships with any of the clusters.

3.2.5 Analysis of FCM Model

Let us first start by describing the HCM (hard c-means) model. The optimisation approach to the clustering problem uses an objective function to measure how good a suggested partition is compared to an ideal, generalised one. This is facilitated by using the concept of cluster prototypes; by introducing them, the formulation of the objective function is made easier. In the ideal scenario, the prototypes are located within very tightly packed clusters of points so that the distances between every cluster of points and its prototype would be almost zero. Deviations from this model can then be formulated, in squared-error fashion, as:

$$\sum_{i=1}^{c} \sum_{k,\mathbf{x}_k \in S_i} d_{ik}^2(\mathbf{x}_k, \mathbf{p}_i)$$

where S_i would be the cluster of points belonging to prototype i. To decide on the membership of a point with a prototype, a crisp decision is made; it belongs to the prototype it is closest to.

FCM generalised the notion of membership to emulate the fuzzy clustering structures found in the real-world. The FCM objective function weighted the distance between a given data point and a given prototype by the corresponding degree of membership between the two (the respective entry in the fuzzy partition matrix). Thus, partitions that minimise this function are those that weight small distances by high membership values and large distances by low membership values. This was formulated as per Equation 3.1. To visualise this, consider Figure 3.3. If point 6 is given a high membership value with prototype B as compared to points 2 and 3, the overall objective function score will be minimal compared to any other membership scheme involving those three points and that prototype.

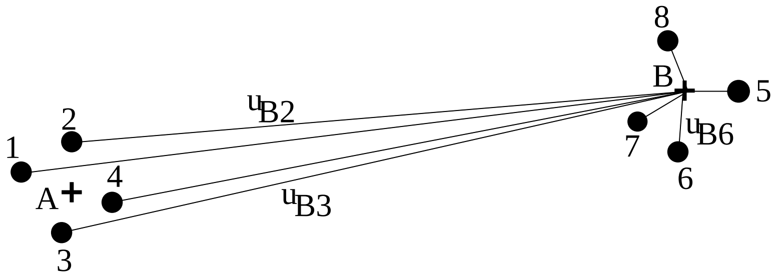


Figure 3.3: The distances between points $1 \cdots 8$ and prototypes A and B are weighted by the degrees of memberships. Here, the distances and memberships concerning only prototype B are shown.

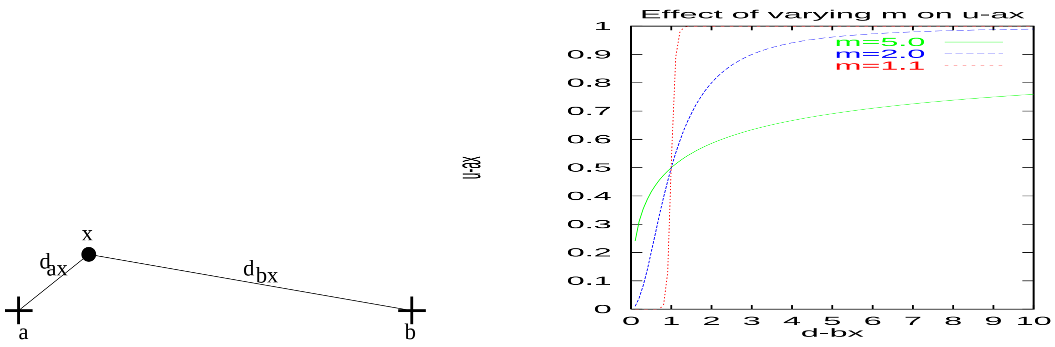


Figure 3.4: Fixing d_{ax} at 1, u_{ax} is plotted versus d_{bx} for $m = 1.1, 2.0$ and 5.0. This clearly illustrates that u_{ax} changes in value depending on the location of the prototype b. Note that as m approaches 1 the membership decision becomes a crisp one.

However, if things were left at the objective function formulation, without the constraint of Equation 3.2, all the u_{ik}'s would take the value of zero as this would set J to the absolute minimal value of zero, which is a trivial solution. In order to force the u_{ik}'s to take values greater than zero, the constraint was imposed. This way, degrees of membership must take non-trivial values.

Looking now at the minimisers of the objective function, Equations 3.3 and 3.4, we see that the prototypes are the fuzzy centroids, or means, of their respective membership function. This is an intuitively-pleasing result.

Further, we see that a point's membership with a given prototype is affected by how far that same point is to the other prototypes. This is illustrated in Figure 3.4, where

$$u_{ax} = \frac{1}{1 + (\frac{d_{ax}^2}{d_{bx}^2})^{(1/m-1)}}.$$

This may cause counter-intuitive behaviour in real-world data. For example, in the case of a “noise” point lying far outside of two clusters but equi-distantly to both centroids, such a point would be given a membership of 0.5 with each cluster (see Figure 3.4: all curves pass through 0.5 for $d_{bx} = 1$). The intuitive solution would be to award such points equal but small membership degrees with each cluster. However, such a solution would violate the constraint of equation 3.2 (memberships must add to 1). If we observe Figure 3.4, we notice that a point’s membership degree is not a function of anything but its relative distances to each prototype. The presence of many points close to one prototype which is our (human) cue to the “noiseness” of a point, is not included. Later in this Chapter, we will present brief summaries of some ideas proposed to alleviate this counter-intuitive behaviour.

3.2.6 Notes on Using FCM

Several investigations have been made on the best value to choose for the fuzzification exponent, m, which is chosen a priori. A recent study [Pal & Bezdek, 1995] concludes empirically that $m = 2.0$ is a “good” value. This value for m has the further advantage of simplifying the update equations and can therefore speed up computer implementations of the algorithm.

Many investigations have been made on the convergence properties of FCM, for example, [Bezdek, 1980; Selim & Kamel, 1992]. The conclusion is that the constraint of Equation 3.2 is a necessary condition for the proof of convergence to a local minimum of the FCM algorithm.

Investigations have also been made on speeding up the implementation of FCM

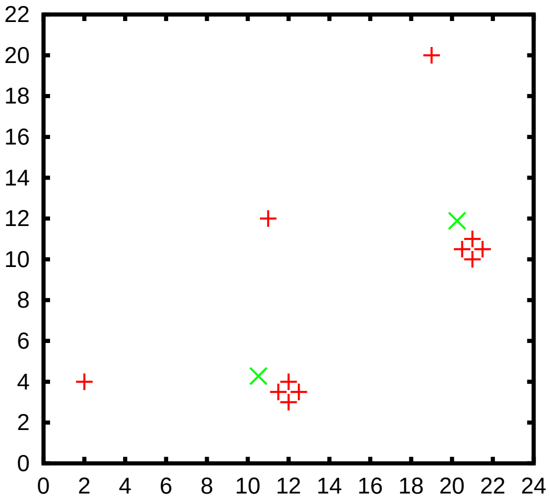
[Cannon *et al.*, 1986]. Recent examples of such studies are geared towards image analysis applications [Cheng *et al.*, 1998; Smith, 1998], and report orders of magnitude speed-ups.

3.2.7 Strengths and Weaknesses

The FCM algorithm has proven a very popular method of clustering for many reasons. In terms of programming implementation, it is relatively straightforward. It employs an objective function that is intuitive and easy-to-grasp . For data sets composed of hyper-spherically-shaped well-separated clusters, FCM discovers these clusters accurately. Furthermore, because of its fuzzy basis, it performs robustly: it always converges to a solution, and it provides consistent membership values.

The shortcomings of FCM, as we have assessed them independently, are:

1. It requires the number of clusters to look for to be known a priori.
2. Initialisation
   (a) It requires initialisation for the prototypes, good initialisation positions are difficult to assess.
   (b) If the iterative algorithm commonly employed for finding solutions of the FCM objective function is used, it may find more than one solution depending on the initialisation. This relates to the general problem of local and global optimisation.
3. It looks for clusters of the same shape (hyper-spheres if using the Euclidean metric); different cluster shapes cannot be mixed.
4. Its objective function is not a good clustering criterion when clusters are close to one another but are not equal in size or population. This is studied comprehensively in Chapter 4.



Data		Memberships	
x	y	Cluster 1	Cluster 2
12.0	3.0	0.975	0.025
12.0	4.0	0.983	0.017
11.5	3.5	0.989	0.011
12.5	3.5	0.967	0.033
21.0	10.0	0.028	0.972
21.0	11.0	0.009	0.991
20.5	10.5	0.014	0.986
21.5	10.5	0.021	0.979
2.0	4.0	0.845	0.155
19.0	20.0	0.174	0.826
11.0	12.0	0.588	0.412

Figure 3.5: A data set containing noise points. The prototypes found by FCM are also plotted. Membership values provided as output are shown on the right hand side. The presence of noise points strongly affected the positions of the found prototypes, furthermore, the noise points' membership values may be consistent but they are not intuitive.

5. Its accuracy is sensitive to noise and outlier points (as demonstrated in Figure 3.2 and also again in Figure 3.5 where the placement of the prototypes was affected by the outlying points). This is so because it squares the "error" between a prototype and a point, thus, the effect of outlier and noise points is emphasised.

6. It gives counter-intuitive membership values for noise points. Noise points are those that do not belong to any cluster, thus, their type of memberships should not necessarily sum to one. In Figure 3.5, for example, the far points to the top and bottom of the plot should have low memberships with both clusters. However, FCM gives each of them a membership value of more than 0.8 with their respective nearest cluster. The probabilistic constraint of Equation 3.2 causes this behaviour.

3.3 Extensions of FCM

Despite its weaknesses, the strengths of FCM have led researchers to generalise and extend it further. In fuzzy covariance clustering, covered in Section 3.3.1, hyperellipsoids can be detected instead of only hyperspheres. In elliptotype clustering, covered in Section 3.3.2, lines or planes can be detected by means of looking for hyperellipsoids with a flat thickness. In shell clustering, covered in 3.3.3, boundaries of spheres and ellipsoids are detected. All these extensions cannot mix cluster shapes, *i.e.*, they cannot look for a line and a circular shell simultaneously. Furthermore, they are all very sensitive to initialisation and much more computationally expensive than FCM. However, they must be considered as necessary evolutionary steps in the development of better fuzzy clustering algorithms. This view also underlies our own work in Chapter 4.

A generalisation was made by Bobrowski and Bezdek [Bobrowski & Bezdek, 1991] of the distance metric norm. For generalisations and extensions relating to handling non-numeric data see [Hathaway *et al.*, 1996; Huang, 1998]. For generalisations and comparisons with switching regression models see [Hathaway & Bezdek, 1993], and linear vector quantisation models see [Bezdek, 1992; Karayiannis *et al.*, 1996].

3.3.1 Fuzzy Covariance Clustering

Gustafson and Kessel [Gustafson & Kessel, 1979] introduced a new variation on the FCM functional given by Equation 3.1 by allowing the inner product inducing matrix $\mathbf{A}$ used in the distance metric to vary per each cluster. In other words, they allowed each cluster to have its own A-norm with which to measure distances from its prototype. This allows different clusters to have differing ellipsoidal shapes. Thus, their modified objective function becomes:

$$\mathbf{J}(\mathcal{P},\mathcal{U},\mathcal{A};\mathcal{X},c,m) = \sum_{k=1}^{N}\sum_{i=1}^{c} u_{ik}^{m} \| \mathbf{x}_k - \mathbf{p}_i \|_{A_i}^{2} = \sum_{i=1}^{c} u_{ik}^{m} \Big((\mathbf{x}_k - \mathbf{p}_i)^T A_i (\mathbf{x}_k - \mathbf{p}_i) \Big) \tag{3.5}$$

where A_i is a positive definite symmetric matrix. An additional constraint to the constraint of equation 3.2 was imposed. This is:

$$||\mathbf{A}_i|| = \rho_i = constant \tag{3.6}$$

1. Fix c. Fix m. Initialise all $\mathbf{p}_i$. Initialise all A_i.
2. Calculate fuzzy partition matrix $\mathcal{U}$ by $u_{ik} = \frac{1}{\sum_{j=1}^{c} (\frac{d_{ik}}{d_{jk}})^{2/(m-1)}}$
3. Update prototypes $\mathcal{P}$ by $\mathbf{p}_i = \frac{\sum_{k=1}^{N} u_{ik}^{m}\mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^{m}}$
4. Calculate A's by

   $$A_i^{-1} = (\frac{1}{\rho_i|\mathbf{C}_i|})^{1/p}\mathbf{C}_i$$

   where $\mathbf{C}_i$, the fuzzy covariance matrix, is given by:

   $$\mathbf{C}_i = \frac{\sum_{k=1}^{N} u_{ik}^{m}(\mathbf{x}_k - \mathbf{p}_i)(\mathbf{x}_k - \mathbf{p}_i)^T}{\sum_{k=1}^{N} u_{ik}^{m}}$$

5. If termination condition not achieved, return to step 2.

Figure 3.6: The Gustafson-Kessel Algorithm

The resulting optimality conditions remain the same with the addition of an update equation for the A_i's. The modified algorithm is described in Figure 3.6. Allowing A_i to vary for each cluster enables the detection of ellipsoidal-shaped clusters each with a differing orientation. The new constraint above limits the volume within which an A-norm metric can have influence. The new constraint may have been placed to simplify deriving update equations that would allow implementation of the method as an algorithm. Adding this constraint, however, causes the G-K algorithm, as it is commonly referred to, to look for hyper-ellipsoids of equal volume and this may limit its accuracy, [Krishnapuram & Kim, 1999]. Note that unlike FCM there is no proof of convergence for this algorithm. Furthermore, the algorithm is very sensitive to initialisation.

3.3.2 Fuzzy c-Elliptotypes Clustering

The Fuzzy c-Elliptotypes (FCE) algorithm was proposed by Bezdek *et al.* to detect clusters that have the shape of lines or planes [Bezdek, 1981]. Its main idea is to discount Euclidean distances for points lying along the main eigenvector directions of a cluster (like those lying on a line) while taking the Euclidean distance in full for other points. This is achieved by means of using a distance measure which is a weighted combination of two distance measures:

$$d^2(\mathbf{x}_k, \mathbf{p}_i) = \alpha d^2_{Vik} + (1 - \alpha) d^2_{Eik}. \tag{3.7}$$

Here, d^2_{Eik} is the Euclidean distance and d^2_{Vik} is defined as:

$$d^2_{Vik} = \parallel \mathbf{x}_i - \mathbf{p}_i \parallel^2 - \sum_{j=1}^{r} ((\mathbf{x}_k - \mathbf{p}_i) \cdot \mathbf{e}_{ij})$$

where $r \in [1, p]$, and $\mathbf{e}_{ij}$ is the j^{th} eigenvector of the covariance matrix $\mathbf{C}_i$ of cluster i. (The $\cdot$ operator denotes the dot product of the two vectors.) The eigenvectors are assumed to be arranged in descending order of the corresponding eigenvalues. Thus, the first eigenvector describes the direction of the longest axis of the cluster. When $r = 1$, d^2_{Vik} can be used to detect lines, and when $r = 2$ it can be used to detect planes. The value of α in Equation 3.7 varies from 0 to 1 and needs to be specified a priori, but there is a dynamic method commonly used in the algorithm's implementations (see [Davé, 1992]). It has been shown [Krishnapuram & Kim, 1999] that by allowing this dynamic variation the FCE algorithm avoids the G-K algorithm's shortcoming of looking for clusters of equal volumes. However, since it looks for only linear structures, it would therefore fit these structures onto data that may not contain them. The update equations for this algorithm can be shown to be equivalent to those of fuzzy covariance clustering.

3.3.3 Shell Clustering

The main application of shell clustering algorithms is in image processing. Images are pre-processed for edge detection and the edge pixels are then fed to these algorithms for boundary detection. There are several variants of shell clustering algorithms and a full review of them can be found in [Hoppner *et al.*, 1999].

The main innovation behind every shell clustering algorithm is the distance measure it uses. In the Fuzzy c-shells algorithm by Davé, the prototype for a circular shell cluster is described by its centre point and radius, $\mathbf{p}_i$ and r_i, respectively. The distance measure is:

$$d^2(\mathbf{x}_k, (\mathbf{p}_i, r_i)) = (\parallel \mathbf{x}_k - \mathbf{p}_i \parallel - r_i)^2$$

In the fuzzy c-spherical shells algorithm the distance measure used instead is:

$$d^2(\mathbf{x}_k, (\mathbf{p}_i, r_i)) = (\parallel \mathbf{x}_k - \mathbf{p}_i \parallel^2 - r_i^2)^2$$

This distance measure is more sensitive to points on the outside of the shell than on the inside but has the advantage of simplifying the update equations. In the adaptive fuzzy c-shells algorithm, shells in the shapes of ellipses are detected by means of the distance measure:

$$d^2(\mathbf{x}_k, (\mathbf{p}_i, A)) = (\sqrt{(\mathbf{x}_k - \mathbf{p}_i)^T A (\mathbf{x}_k - \mathbf{p}_i)} - 1)^2$$

where A is a positive definite matrix that contains the axes and orientations of the ellipse. A more complex distance measure for shell ellipsoids is described in [Frigui

& Krishnapuram, 1996].

Shell clustering algorithms are computationally expensive because their update equations require solving a system of non-linear equations which require iteration. Thus, within each clustering iteration, several iterations take place. Data set sizes of more than two dimensions or of lengths more than a few thousand are impractical.

3.4 Modifications to the FCM Model

Several attempts have been made to remedy one or more of the shortcomings we mentioned in Section 3.2.7. In Possibilistic Clustering, covered in Section 3.4.1, the membership value of a point with a cluster does not depend on the location of other cluster prototypes. In High Contrast Clustering, covered in 3.4.2, mixtures of the hard and fuzzy c-means algorithms will be formulated. In Competitive Agglomeration, covered in 3.4.3, the requirement for specifying c is overcome by means of starting with a large value for it and subsequently letting bigger clusters compete for the smaller ones. In Credibilistic Clustering, covered in 3.4.4, noise points are identified first as not credible representatives of the data set and awarded membership values that do not sum up to 1.

3.4.1 Possibilistic C-Means (PCM) Clustering

Krishnapuram and Keller [Krishnapuram & Keller, 1993a] removed what they termed the probabilistic constraint of Equation 3.2 by allowing the degrees of membership, u_{ij}, to take on any value within the $[0 - 1]$ range. Their argument for the removal of the constraint was: *the membership function of a set should not depend on the membership functions of other fuzzy sets defined in the same domain of discourse*. The u_{ij}'s were therefore allowed to take on any value within the $[0 - 1]$ range, but in order to avoid the trivial solution, the following *possibilistic* constraint was added:

$$\max_i u_{ij} > 0 \quad \forall j \tag{3.8}$$

Thus, the memberships values generated are taken as absolute, not relative, and denote degrees of belonging or typicality of the cluster in question.

The new objective function proposed was:

$$\mathbf{J}(\mathbf{U}, v; \mathcal{X}) = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^m d_{ik}^2 + \sum_{i=1}^{c} \eta_i \sum_{k=1}^{N} (1 - u_{ik})^m \tag{3.9}$$

where η_i are positive numbers. The first term is the normal FCM objective function which is minimised for compact and well-separated clusters, whereas the second term forces the u_{ik}'s to be as large as possible, thus avoiding the trivial solution. This formulation of the objective function leads to the update equation of u_{ik} to be modified to

$$u_{ik} = \frac{1}{1 + (\frac{d_{ik}^2}{\eta_i})^{\frac{1}{m-1}}} \tag{3.10}$$

The value of η_i determines the distance at which the membership value of a point in a cluster becomes 0.5. If all clusters are expected to be similar in size, this value could be the same for all of them. In the objective function we notice that the value of η_i determines the relative importance of the second term and the authors observe that it should therefore be of the same range as d_{ik}^2 if equal weighting to both terms is desired.

This definition of possibilistic clustering can be applied to the other fuzzy clustering algorithms. So, if we use the FCM algorithm but update u_{ik} according to Equation 3.10 above (plugging in the suitable values for the η_i's), the algorithm becomes the Possibilistic c-Means algorithm (PCM). Likewise we may have the Possibilistic Gustafson-Kessel algorithm, Possiblistic c-Spherical Shells algorithm, and so on.

The success of PCM is very much dependent on the initialisation, as it may not converge or the prototypes may even conincide [Barni *et al.*, 1996; Krishnapuram & Keller, 1996]. The values of η_i to use are probably the most difficult choice to make when using this algorithm. The authors themselves recommend running FCM once and estimating η_i from its output, then running PCM and adjusting η_i in its first few iterations in order to provide the most meaningful values of u_{ik} while bypassing the danger of not converging to a stationary point. The main advantage of PCM is that it is more resilient to noise by comparison to FCM, and after taking the above guidelines into consideration, the membership values it finds are more intuitive by human perception standards.

3.4.2 High Contrast

Except in the case where a data point coincides with the location of a prototype, degrees of membership found by FCM are never either 0 or 1. This is so even when a point is very close to a prototype. The reason for this is the "sharing" constraint of Equation 3.2 imposed on the FCM optimisation problem. This constraint leads to update Equation 3.4 from which we can see that a membership value will never be zero, since it is a ratio of distances. This peculiarity causes core points of a cluster to receive membership values of less than one, even though we would clearly see them as being typical of the cluster.

Approaches of the "High Contrast" kind, though not developed fully in [Rousseeuw *et al.*, 1995; Pei *et al.*, 1996], aim to classify clear-cut, core, points in a crisp manner, while leaving other points to still be classified in a fuzzy manner.

In [Rousseeuw *et al.*, 1995], the u_{ik}^2 term in the objective function is replaced by $f(u_{ik}) = cu_{ik} + (1-c)u_{ik}^2$ where $0 < c < 1$ is termed a contrast factor. When $c = 0$, $f(u_{ik}) = u_{ik}^2$ which gives a fuzzy solution identical to standard FCM. When $c = 1$, $f(u_{ik}) = u_{ik}$ which gives a crisp solution identical to standard HCM. Varying c between 0 and 1 changes the "contrast" of the clustering results from none whatsover (fuzzy) to full (crisp). Rousseeuw *et al.* conclude empirically that $c = 0.3$ is a good value to set the contrast factor. However, the general case of $m \neq 2.0$ was not mentioned in the paper, nor were the differences between their approach and dynamically varying m stated.

3.4.3 Competitive Agglomeration

The CA algorithm [Frigui & Krishnapuram, 1997] was proposed as a robust successor to FCM attempting to remedy several of its shortcomings. First, it requires only a maximum number of clusters as input rather than the exact number, it will then find the "correct" number of clusters itself. It does so by first partitioning the data set into the given (large) number of small clusters. As the algorithm progresses, adjacent clusters compete for data points and the clusters that lose the competition gradually become depleted and vanish.

The CA algorithm minimises the following objective function, noting that c is dynamically updated by the algorithm:

$$J(\mathcal{P},\mathcal{U}) = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2 \| \mathbf{x}_k - \mathbf{p}_i \|_A^2 - \alpha \sum_{i=1}^{c}[\sum_{k=1}^{N} u_{ik}]^2 \tag{3.11}$$

subject to

$$\sum_{i=1}^{c} u_{ik} = 1 \qquad \forall k \in \{1, \ldots, N\} \tag{3.12}$$

The objective function has two components. The first component is similar to the FCM objective function ($m = 2.0$) while the second component is the sum of squares of the fuzzy cardinalities [2] of the clusters. The global minimum of the first component is achieved when the number of clusters c is equal to the number of samples N, *i.e.*,

[2] The cardinality of a cluster is the number of points belonging to it; the fuzzy cardinality of a cluster is the sum of the memberships of all points with it.

each cluster contains a single point. The global minimum of the second component is achieved when all points are grouped into one cluster, and all other clusters are empty. Based on the premise that α is chosen properly the final partition resulting from this algorithm will find compact clusters while at the same time finding the smallest possible number of clusters.

3.4.4 Credibilistic Clustering

Noise points, *i.e.*, points that do not lie close to any particular cluster are not distinguished as such by FCM. They share memberships with all clusters just like all points even though we may clearly identify them as not belonging to any cluster. Noise points affect the accuracy of the FCM algorithm.

The credibilistic fuzzy c-means algorithm was proposed by Chintalapudi and Kam [Chintalapudi & Kam, 1998] to combat FCM's sensitivity to noise points. Their requirement was to assign to noise points low membership values with all clusters. In this way, noise points will not affect the location of the prototypes.

The probabilistic constraint of Equation 3.2 was replaced by:

$$\sum_{i=1}^{c} u_{ik} = \psi_k \qquad\qquad \forall k$$

where ψ_k is the *credibility* of point $\mathbf{x}_k$. It represents the typicality of $\mathbf{x}_k$ to the entire data set and not to any specific cluster. Thus, if $\psi_k > \psi_j$, then $\mathbf{x}_k$ is more typical to $\mathcal{X}$ than $\mathbf{x}_j$. Two alternative formulations for the credibility of a point are given, both are measures of the relative isolation of the point. The first formulation compares the point's average distance to its σ nearest neighbours to the average intra-point distance of $\mathcal{X}$, while the second formulation compares it to the harmonic second moment of $\mathcal{X}$. After estimating the ψ_k's, their values are plugged into the slightly modified update equations of the algorithm. This approach has also introduced its own share of parameters. However, the authors suggest that their algorithm performs well in most cases using the default values for the parameters.

3.5 Remarks

In this Chapter, we reviewed in detail the Fuzzy c-Means clustering model, and we also briefly reviewed some of its extensions and modifications. We explained that a lot of the algorithms mentioned in this Chapter were motivated by one or more of the shortcomings we listed in Section 3.2.7. In the next Chapter, we will focus only on FCM's inability to perform accurately on data sets containing clusters close to one another but not equal in size or population. Since we have only mentioned a few of the large body of algorithms based on FCM, we conclude with a quick look at two threads of fuzzy clustering research we did not include in our review.

The first thread of research is concerned with finding the optimal number of clusters in the data. This problem is continually being addressed in the literature. The first approach is to validate fuzzy partitions obtained at different values of c by means of an index, and then selecting the value of c corresponding to the partition that scored best on the index. In comparison to many indices, the Xie-Beni index [Xie & Beni, 1991] performs best (as studied in [Pal & Bezdek, 1995]), though there are some new competitors [Kwon, 1998; Rezaee *et al.*, 1998]. Further, there have been attempts to integrate the validation step into the FCM clustering process such as the validity-guided clustering method of Bensaid [Bensaid *et al.*, 1996]. The second approach is to fuse an agglomeration process with the clustering process, starting at a reasonably high value for c. Section 3.4.3 already described an algorithm of this type. Another recent algorithm is that of [Geva, 1999] which fuses hierarchial clustering with fuzzy clustering.

The second thread of research is concerned with making FCM more robust by enhancing its response to noise points. We have already discussed one such algorithm in Section 3.4.4 which addressed that point, however the recent and still developing work by [Davé & Krishnapuram, 1997; Frigui & Krishnapuram, 1999] should also be highlighted. These aim to use statistical methods such as the M estimator and weighted least-squares technique to supplement the objective function.

CHAPTER 4

A New Algorithm for Fuzzy Clustering

In the previous Chapter we described the FCM algorithm and detailed several algorithms based on it.

In this Chapter, we investigate a well-known behavioural shortcoming of FCM, namely that it mis-classifies a small cluster of data lying close to a large one. We formulate a new objective function (OF), based on FCM's, that redresses this shortcoming. We will accordingly derive a new algorithm, that we named the Population Diameter Independent (PDI) algorithm. We will evaluate PDI's effectiveness by comparing its results with those of FCM.

We first start by describing a framework to evaluate the behavioural performance of objective-function-based clustering algorithms. Focusing only on the small-cluster shortcoming, we identify the factors that cause it. In correspondence to the factors we identified, we will then generate a suite of benchmarks consisting of two-dimensional data sets of incrementally varying properties. Tabulating the output of the FCM algorithm, we will demonstrate the extent of the shortcoming and analyse how to overcome it. We will then develop PDI and evaluate its behaviour on our chosen benchmark.

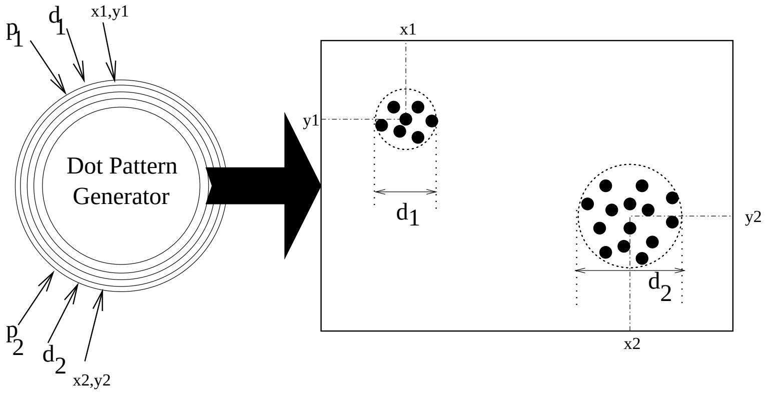


Figure 4.1: In this two-clusters example, the inputs to the dot pattern generator are: the populations, p_1 and p_2, the diameters, d_1 and d_2, of each cluster, and the central locations of each cluster, (x_1, y_1) and (x_2, y_2). A clustering algorithm should now attempt to match this description of the clusters by examining only the dot pattern.

4.1 The Experimental Framework

Assume that we have a dot pattern generator that generates clusters of points in a given p-dimensional feature space, $\mathcal{R}^p$. Assume, further, that the points of every cluster are distributed uniformly around that cluster's centre-point. This assumed generator will require as input a number of parameters. First, the number of clusters we want to have in the dot pattern. Second, the central location of each cluster. Finally, for each cluster its population and diameter. We define the *diameter* of a cluster as the diameter of a hyper-sphere (or a circle in 2D) that contains the entire population of the cluster. This is illustrated in Figure 4.1.

The test for any clustering algorithm would be to produce an accurate description of the clusters present in the dot pattern, given only the dot pattern and no other information. Since the clustering structure of the dot pattern is already known, accuracy of the clustering can be computed by comparing the known description to the one discovered by the clustering algorithm. This is illustrated in Figure 4.2. Thus, for the example in Figure 4.1, we would ideally like any clustering algorithm to output the in-

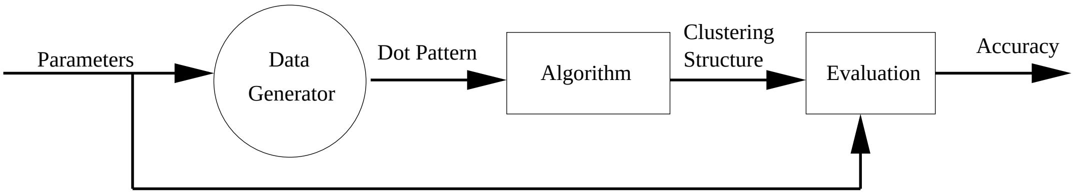


Figure 4.2: A block diagram of our framework.

formation: number of clusters is two; the locations of the prototypes of the two clusters are (x_1, y_1) and (x_2, y_2); the diameters of the two clusters are d_1 and d_2 respectively; as well as a classification of the points from which we can calculate the populations p_1 and p_2.

The generator we have described above is ideal for objective-function (OF) methods that minimise the aggregate distances between data points and suggested prototypes. This type of methods, as discussed earlier, search for hyper-spherical clusters of points (assuming the Euclidean distance metric). Prototype-based, sum-of-squared-error objective function methods like FCM should perform with maximum accuracy because the generated data consists of hyper-spherical clusters.

In general, clustering algorithms provide different types of results, e.g., fuzzy, crisp, or hierarchical. These different ways of providing a description of the clustering structure will necessitate different types of accuracy measures for evaluation. Algorithms like FCM produce their results in the form of prototype locations and the fuzzy partition matrix. To evaluate this output, one accuracy measure could be the average offsets of the FCM-found prototypes from the known central locations. Another accuracy measure could be to use the FCM-found partition matrix to calculate the fuzzy cardinalities of the clusters [1] and then to compare these values with the known population values. In a similar manner, accuracy measures for the diameters of the found clusters can be devised.

[1] the fuzzy cardinality of a cluster is the sum of all its membership values

A further test for any clustering algorithm is to find the correct solution every time it is run. Since the results of some clustering algorithms may depend on the initialisation, the correct solution should be found irrespective of the initialisation. Otherwise, the algorithm would not be suitable for non-expert use. This challenge, however, can be assumed to be of less priority than the other challenges since it depends also on the optimisation scheme used.

4.1.1 Tests for Clustering Algorithms

Within our framework, a clustering algorithm is required to find reliably:-

I the correct number of clusters,

II the correct locations of the prototypes, and

III populations of the clusters, and also

IV diameters of the clusters,

Realistically, we know that most clustering algorithms will not be able to pass all these tests successfully. For example, most objective-function-based algorithms require the number of clusters, c, as an input parameter beforehand. They thus fail test I. This shortcoming is not addressed in this dissertation, as instead we assume that the correct number of clusters has been estimated beforehand. If no such estimate exists, the common way of handling this shortcoming is validating the solutions resulting from different values of c and choosing the best one [Windham, 1982; Gath & Geva, 1989; Pal & Bezdek, 1995].

Objective-function-based methods can deliver on tests II, III, and IV. Their performance on these tests, however, may seem ad-hoc, for they can find the correct solution in one run but fail to do so in another. The reason is that OF-based algorithms are iterative and locally optimal, and therefore produce results that depend on their initialisation. Unless an exhaustive or an (as-yet undiscovered) analytical solving method is used, different solutions may be found. Therefore, short of initialising these algorithms identically each time they are run, obtaining the same correct solution should not be expected. Thus, in order to measure an algorithm's accuracy on any of the three tests above, we need to use identical initialisation. This initialisation should be favourable to finding the correct solution by being close to it. If an algorithm now fails a test, we will know that it cannot ever find the correct solution starting from a near-correct one.

Turning our attention now to tests II, III, and IV, we observe that within our dot pattern generator framework, we can vary three sets of variables:

1. the centre-points of the clusters,
2. the populations of the clusters, and
3. the diameters of the clusters.

In the next Section we will describe how we used these three variables in a two-dimensional two-cluster setting to generate a suite of synthetic data sets. Our aim is to construct a benchmark covering many of the data sets that could be encountered within this setting and then to see if an OF algorithm like FCM will pass tests II, III, and IV on each of the data sets in the benchmark suite. We should note here that while the framework as described above is ideal for squared-error-type prototype-based methods, its basic structure is valid for other types of methods.

4.1.2 Designing the Synthetic Data

We now use our dot pattern generator to generate a suite of two-dimensional two-cluster data sets. Thus, 6 variables must be set: the two centre-points at (x_1, y_1) and (x_2, y_2), the diameter and population of cluster 1 (the lhs cluster) d_1 and p_1, and the diameter and population of cluster 2 (the rhs cluster) d_2 and p_2. If we do not consider overlapping clusters and sample the range of possibilites, a suite of data sets that covers many cluster configurations can be generated.

The Centre-Points

We first fix the two centre-points at $(0, 0)$ and $(1, 0)$. This is a valid assumption not only because it reduces the number of possible data sets but also because we can always transform any two given locations in 2-D space to the designated coordinates of $(0, 0)$ and $(1, 0)$. The transformations will consist of translation, rotation, and scaling transformations applied sequentially. By fixing the central locations, we can now concentrate on varying the remaining variables.

The Populations

We now have to consider that p_1 and p_2 can vary. Our approach has been to fix a minimum value for the population of any cluster in any of the data sets: p_{min}. Then, to use configurations where clusters have populations that are whole number multiples of p_{min}. Using this new scale, we rename p_1 and p_2 to P_1 and P_2 respectively. We chose to limit the range of both P_1 and P_2 to 1 to 20. A configuration with $P1 : P2 = 1 : 20$ indicates that the lhs cluster has the minimum population while the rhs cluster has twenty times that population. To reduce the number of data sets generated, we sampled the range of $P1$ and $P2$ at 1, 10, and 20 only. Thus, there will be $3^2 = 9$ population configurations. These are (in the form of $P_1 : P_2$): $1 : 1$, $1 : 10$, $1 : 20$, $10 : 1$, $10 : 10$, $10 : 20$, $20 : 1$, $20 : 10$, and $20 : 20$.

The Diameters

With regards to d_1 and d_2, both can have values from zero to infinity — a range that has to be sampled. Let us choose to sample the distance between $(0, 0)$ and $(1, 0)$ 20 times and to restrict the diameters to those 20 levels. Thus, a cluster with diameter-level 10 will touch the point $(0.5, 0)$ and one with diameter-level 20, will touch the other's centre-point. Using this normalised scale, d_1 and d_2 are renamed to D_1 and D_2 respectively, where the latter set has discrete-level units.

In order to not lose focus of our goals, we leave further details of the generation of the data suite to Appendix A. We will say at this juncture that all in all 900 data sets were generated in correspondence to the various combinations of populations and diameters available.

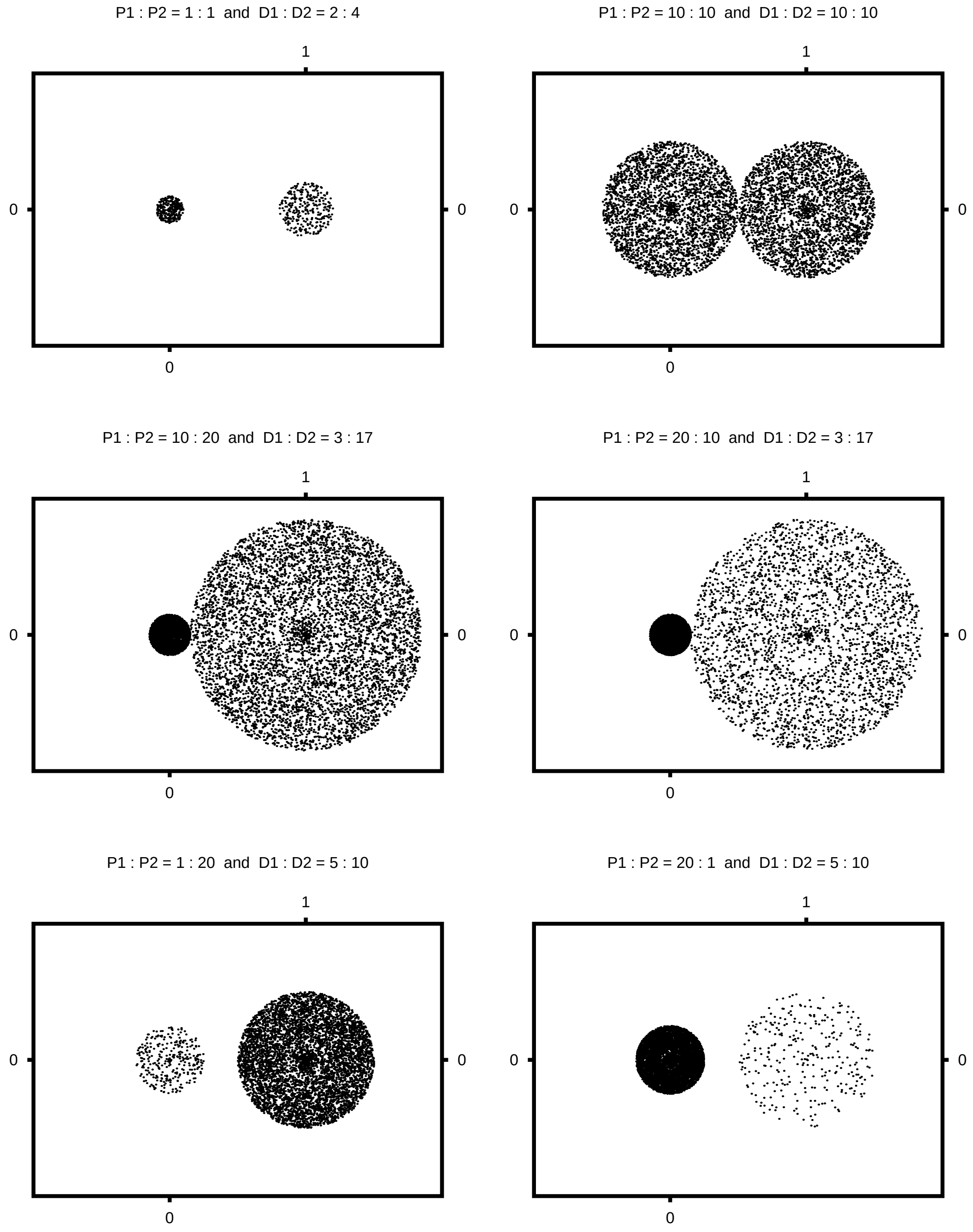
Samples of the Benchmark Data Suite

Table 4.1 shows the values we used in our actual generated suite of data sets. Figure 4.3 illustrates some examples of the 900 data sets generated.

x_1	y_1	p_{min}	d_{min}
0	0	300	2×0.05
x_2	y_2	p_{max}	d_{max}
1	0	6000	2×0.95

Table 4.1: Parameters used to generate the suite of data sets.

The data points of each cluster were generated within a circle centred at the points stated above. The area of each circle is divided into 10 shells of equal areas. The population of a shell, *i.e.*, the number of points inside it, is the result of dividing the total population of the cluster by the number of shells. For each point, two polar coordinates (r, θ) were picked from a random number generator of uniform distribution. These coordinates were then transformed to Cartesian coordinates.



P1 : P2 = 1 : 1 and D1 : D2 = 2 : 4

P1 : P2 = 10 : 10 and D1 : D2 = 10 : 10

P1 : P2 = 10 : 20 and D1 : D2 = 3 : 17

P1 : P2 = 20 : 10 and D1 : D2 = 3 : 17

P1 : P2 = 1 : 20 and D1 : D2 = 5 : 10

P1 : P2 = 20 : 1 and D1 : D2 = 5 : 10

Figure 4.3: Samples of the benchmark data suite. The population and diameter settings for a pattern are located at the top of its plot.

4.2 The Behaviour of FCM

Having established a framework and designed our suite of benchmark data we will now examine the behaviour of the FCM algorithm. We will first present the results of FCM clustering of the benchmark data, then, we will discuss the performance of FCM and its behaviour. In Section 4.3, we will present our new Population Diameter Independent (PDI) algorithm and its results on the same benchmark data.

4.2.1 FCM's Results on the Synthetic Data

FCM was run on the full 900 data sets described in Section 4.1.2. In Figures 4.4 and 4.5 samples of the 900 clustered sets are shown. The prototypes found by FCM are marked out with arrows. Also, the points are classified according to the max rule which specifies that a point is classified according to its maximum degree of membership. FCM was run with m set at 2.0 and the initial prototypes placed at $(-0.05, 0.05)$ and $(1.05, 0.05)$, *i.e.,* at positions which are very close to the ideal positions. It is not our aim here to test FCM's shortcoming of getting entrapped in local solutions. Our aim is to see if the ideal solution can indeed be an FCM solution.

We can clearly see from Figures 4.4 and 4.5 that FCM's performance *is* affected by the relative widths of the clusters and by their relative populations. We can also see that in some cases gross misclassification has occured. Since the prototype initialisation was very favourable (by being very close to the correct locations), we can deduce that in these cases placing the prototypes at the correct locations is not a minimal solution for the OF of FCM.

Let us first provide a summary of the FCM results. To achieve this, we need to decide on our accuracy measures. There are potentially three different measures of a given clustering algorithm's accuracy within our framework. They are:-

1. how well it performs in finding the correct centre-points of the clusters,

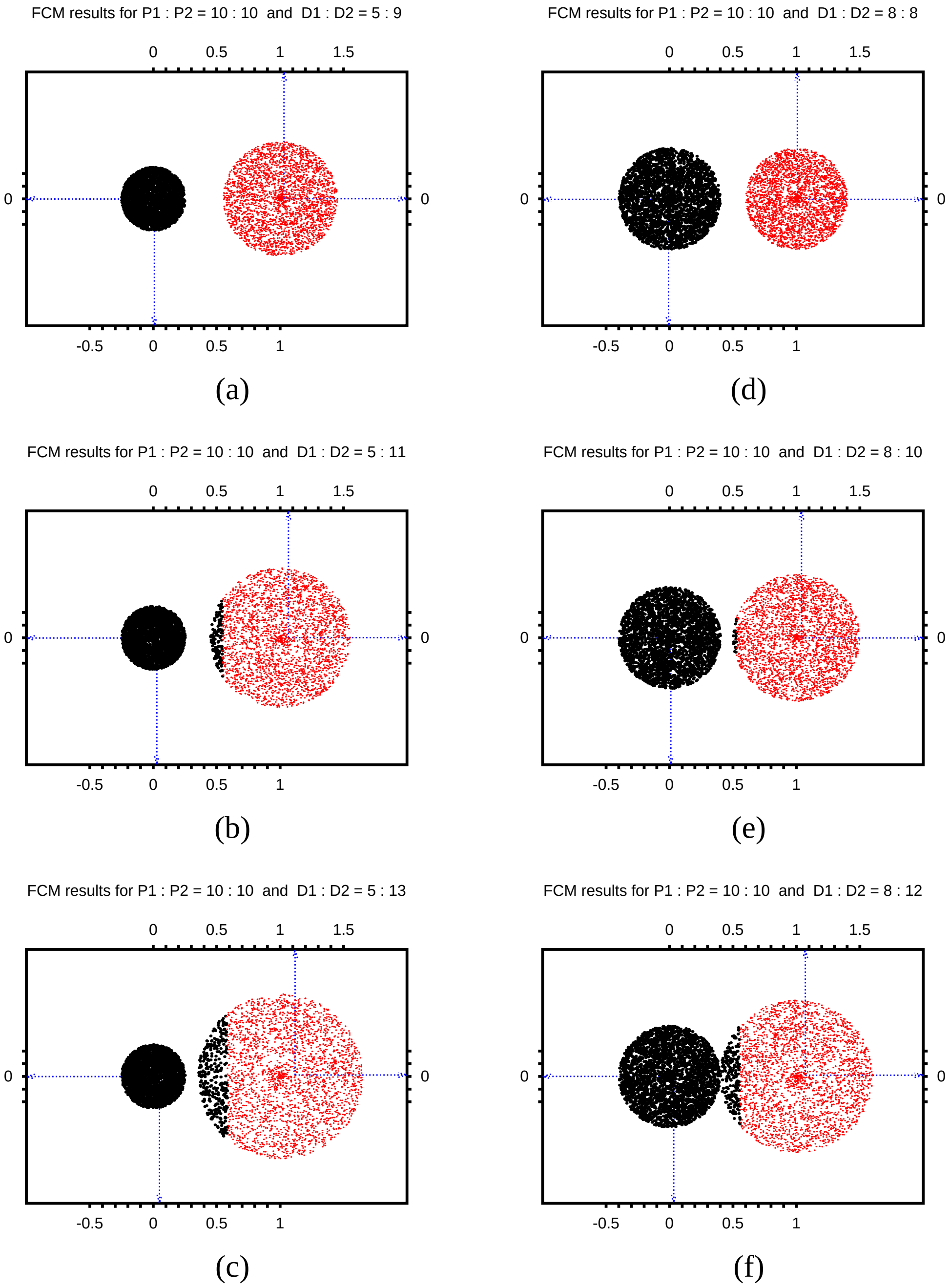


Figure 4.4: FCM clustering of synthetic dot patterns with two colours representing the two found clusters. Prototypes are marked out by the dotted blue lines. P1:P2 ratio fixed at 10:10. D2 is varied while D1=5 for column (a), (b), and (c), and D1=8 for column (d), (e), and (f).

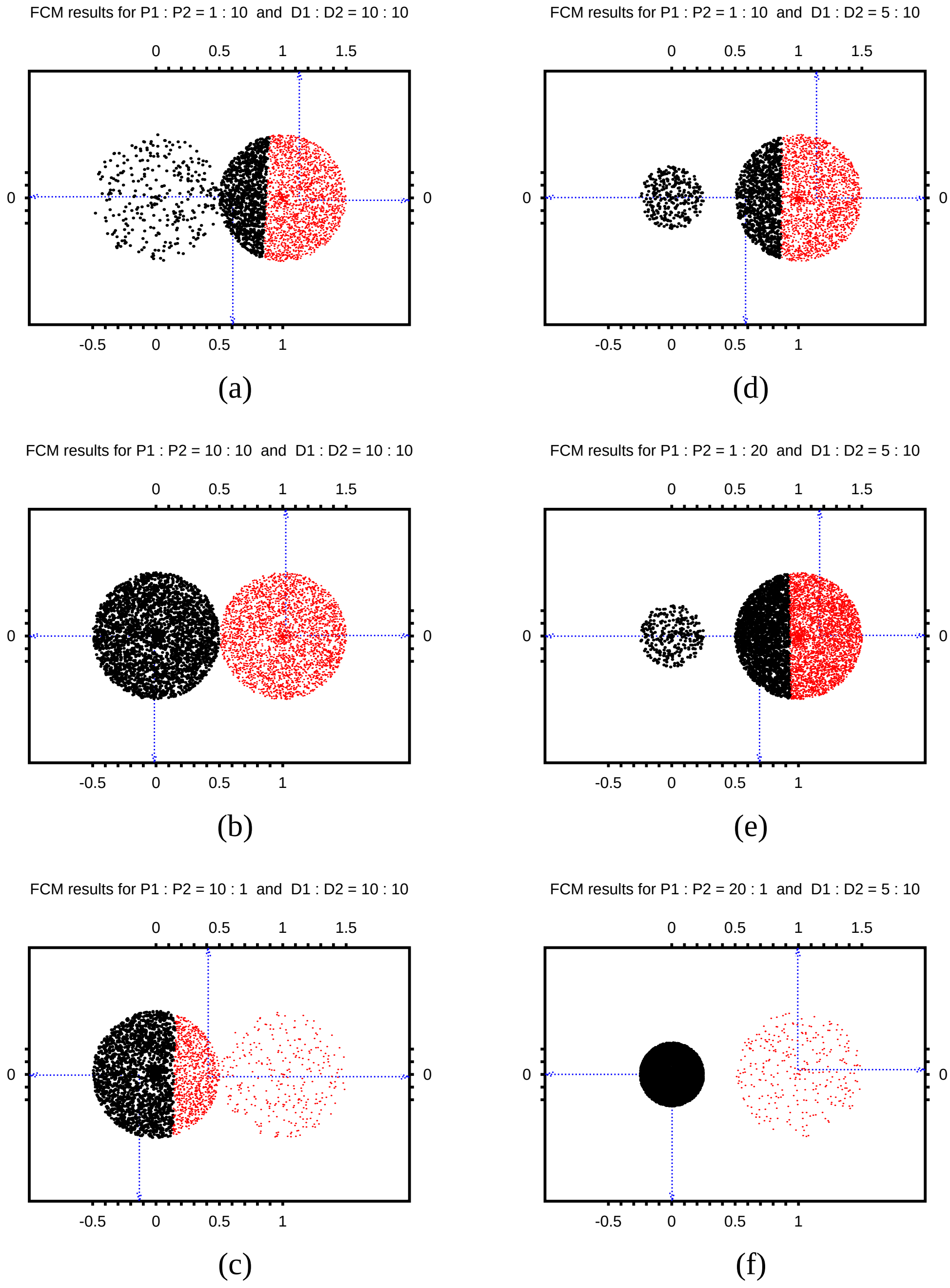


Figure 4.5: FCM clustering results. D1:D2 fixed at 10:10 for column (a), (b), and (c), and 5:10 for column (d), (e), and (f). The P1:P2 ratio is varied.

2. how well it performs in finding the correct diameters of the clusters,
3. and, also how well it performs in finding the correct populations of the clusters.

In FCM's case and with this type of symmetrical cluster, the three measures are not all required. If FCM fails in finding the centre-point, the other two measures become misleading. So, our first priority will be, for every data set, to see how far off each of the found prototypes is from its correct location. We decided that, for a given data set, the maximum of the two prototype offsets will be our measure of accuracy.

Defining e_1 as the distance between $\mathbf{p}_1$ and (x_1, y_1) (where $\mathbf{p}_1$ is the closest found-prototype to (x_1, y_1)), and e_2 similarly, we can define the maximum prototype offset, e, as:

$$e = \max(e_1, e_2)$$

$$\text{where } e_1 = \parallel \mathbf{p}_1 - (x1, y1) \parallel \quad \text{and} \quad e_2 = \parallel \mathbf{p}_2 - (x2, y2) \parallel$$

In Figure 4.6 we plotted e against $D2$ for all nine population configurations, while fixing $D1$ at 1. Each curve represents a constant ratio of proportions. We note that apart from population configurations where P1:P2 = 1:10 and 1:20, the curve proceeds in a somewhat uniform upward trend. However, for the aforementioned configurations, the curve takes a very steep upward climb and then slowly falls afterwards. In both these configurations cluster 2 becomes very large by comparison to cluster 1. This largeness is twofold: both in diameter and in population. Thus, cluster 1's prototype moved toward cluster 2 while cluster 2's prototype moved towards the right side of its own cluster. This is illustrated in Figure 4.7.

As cluster 2 became larger, $\mathbf{p}_1$ was “drawn” towards it and took large steps in that direction. This explains the steep climb. However, after a certain point, the diameter of cluster 2 extended into the middle region between the two clusters, towards cluster 1, thus, $\mathbf{p}_1$ moved back again towards the left side of the graph, causing the decline in e. As D_2 went through and past the middle towards cluster 1, $\mathbf{p}_1$ followed it progressively

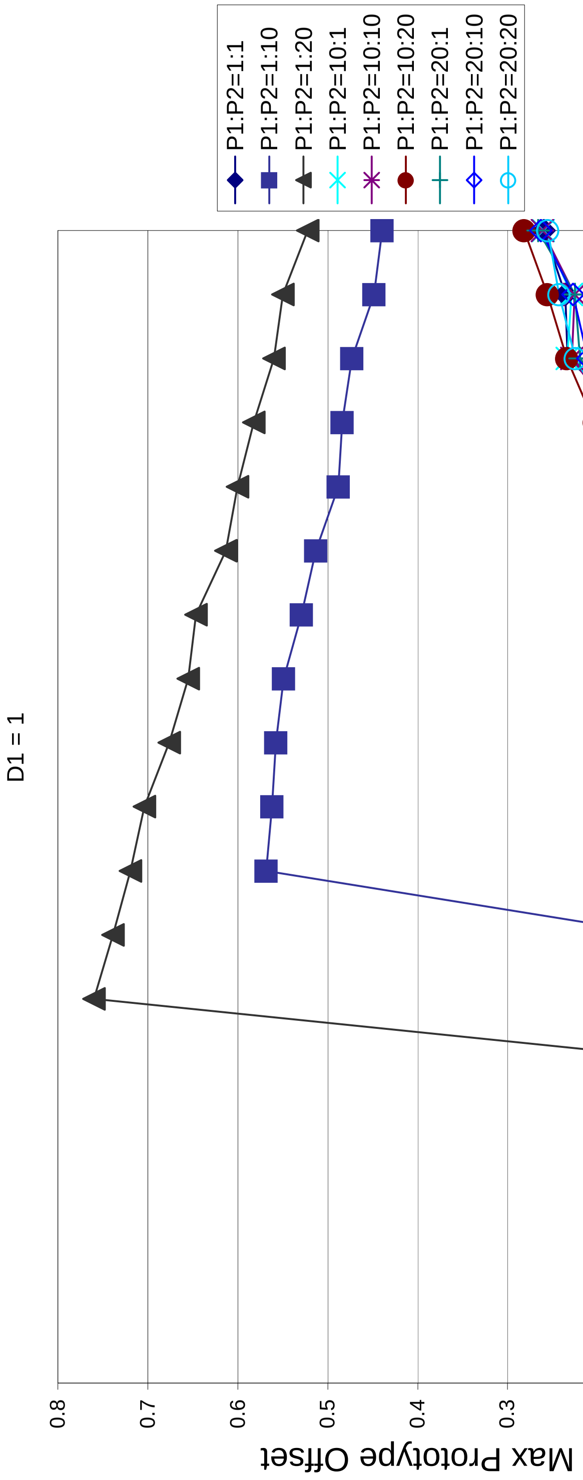
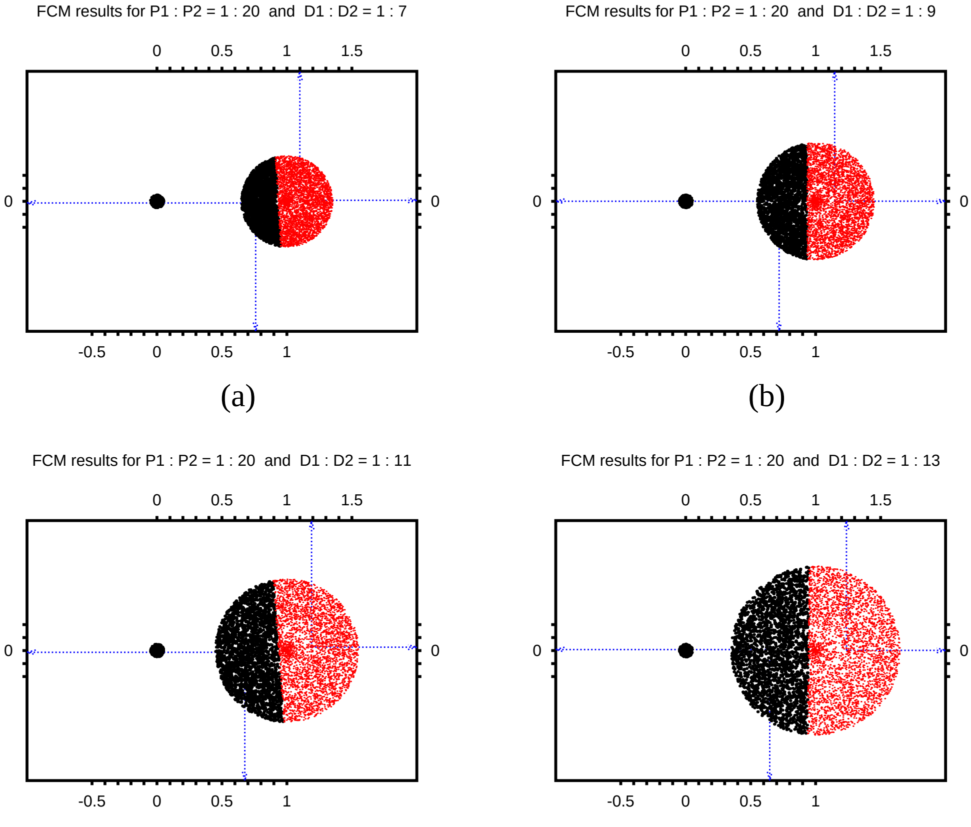


Figure 4.6: Plot of e against $D2$. $D1 = 1$. All nine population configurations are shown. Each curve has a constant population ratio.



(a) (b)

(c) (d)

Figure 4.7: An illustration of the positions of the found prototypes as $D2$ increases from (a) 7, (b) 9, (c) 11, to (d) 13. $D1$ is fixed at 1. This is for $P1 : P2 = 1 : 20$.

back towards the left hand side, albeit with a large margin of error.

The upward-trend curve of the other population configurations (*i.e.*, those excluding $1 : 10$ and $1 : 20$) can be explained that in these configurations, cluster 2 never became as large or "dominant" as the two other cases in terms of population. Thus, there was less requirement for $\mathbf{p}_1$ to move into cluster 2's territory. However, as $D2/D1$ got bigger, the error worsened proportionally.

We examined results for a somewhat larger diameter for cluster 1, at $D1 = 5$. In Figure 4.8, we plotted e against $D2$ while fixing $D1$ for all nine population configurations. The results are quite similar to those of Figure 4.6. The two "sudden rise"

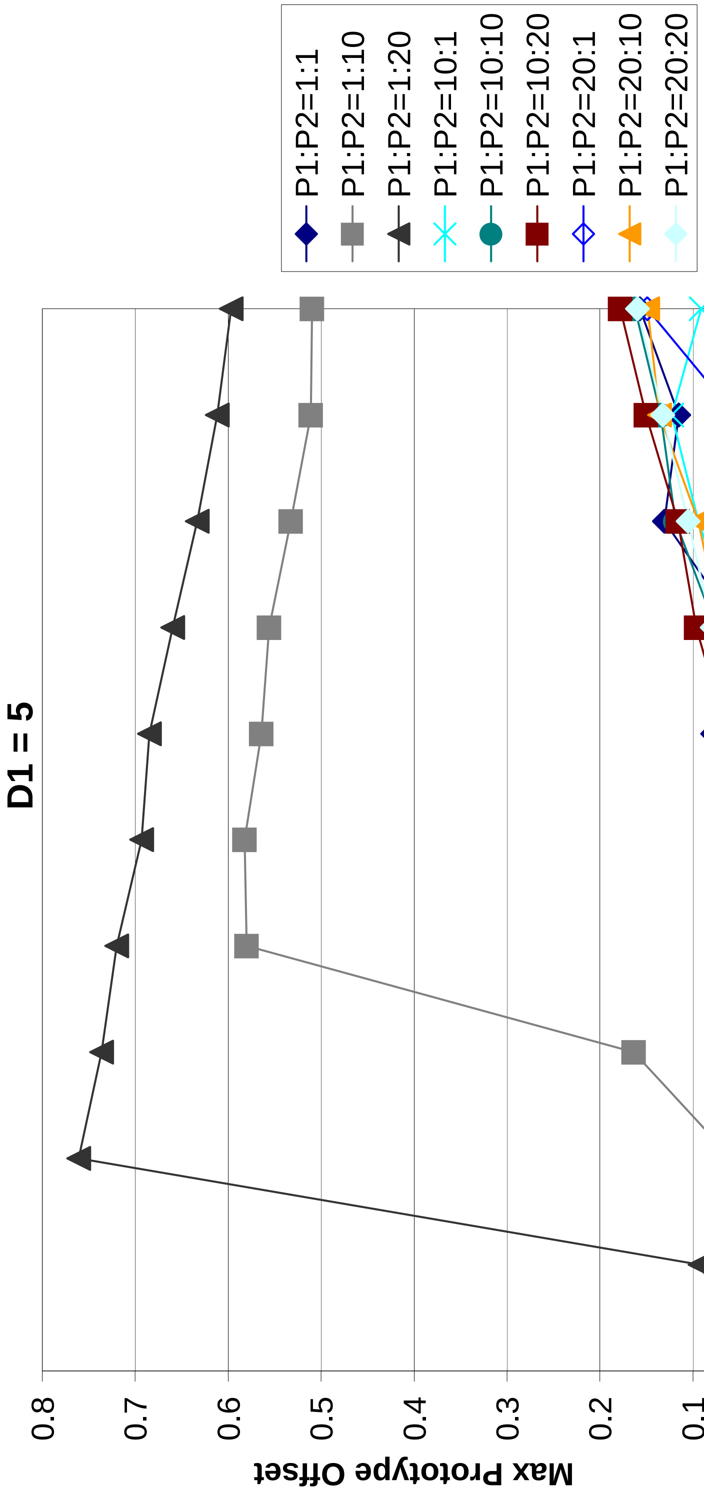


Figure 4.8: Plot of e against $D2$. $D1 = 5$. Each curve has a constant population ratio.

curves of $P1 : P2 = 1 : 10$ and $P1 : P2 = 1 : 20$ are there as well as the "upward trend" curves for the rest of the population configurations. This time, because cluster 1 is fixed at a bigger diameter than in Figure 4.8, the degree of error is generally lower than that of figure (0.18 compared to 0.28, for example).

We found FCM's results on the $P1 : P2 = 1 : 20$ data sets very interesting, so we analysed them all in a separate plot. In Figure 4.9 we plotted the results for all 100 e's resulting from FCM's clustering of the data sets. The almost horizontal curves of the plot illustrate something important: that the effect of the variation of $D1$ is negligible, it is $D2$ that decides the degree of the error. Furthermore, as was observed before, the worst clustering results are those of when $D2$ reaches to within the middle region between the two clusters $(D2 = 7 \dots 13)$, instead of when $D2$ is close to 19.

Next, we studied configurations where the population ratio is in favour of cluster 1 (*i.e.*, cluster 1 is more populous than 2), like $P1 : P2 = 20 : 1$. The results are plotted in Figure 4.10. Here, we note that before $D1 = 7$, e is proportional to $D2$, such that the worst error is at $D2 = 19$. Then, at $D1 = 7$, a sudden jump in e is observed (see Figure 4.11). As of that point, the effect of the variation of $D2$ is negligible, as the plots coincide. This can be explained in a way similar to the $P1 : P2 = 1 : 20$ configuration above. Cluster 1, the more populous and larger cluster dominated the FCM solution. It drew $\mathbf{p}_2$ towards it. However, as cluster 1 expanded ($D1$ increased), $\mathbf{p}_2$ has moved back towards the left thus reducing e progressively.

4.2.2 Discussion of FCM's Results

From the results above, we can deduce preliminarily that as long as the separation between clusters is high, FCM will not have a problem in identifying the output of the pattern generator. Once one of the clusters extends into the middle region between the two centre-points, FCM will produce very bad results. The question of the ratio of the populations of the clusters plays a role in these diameter configurations and makes

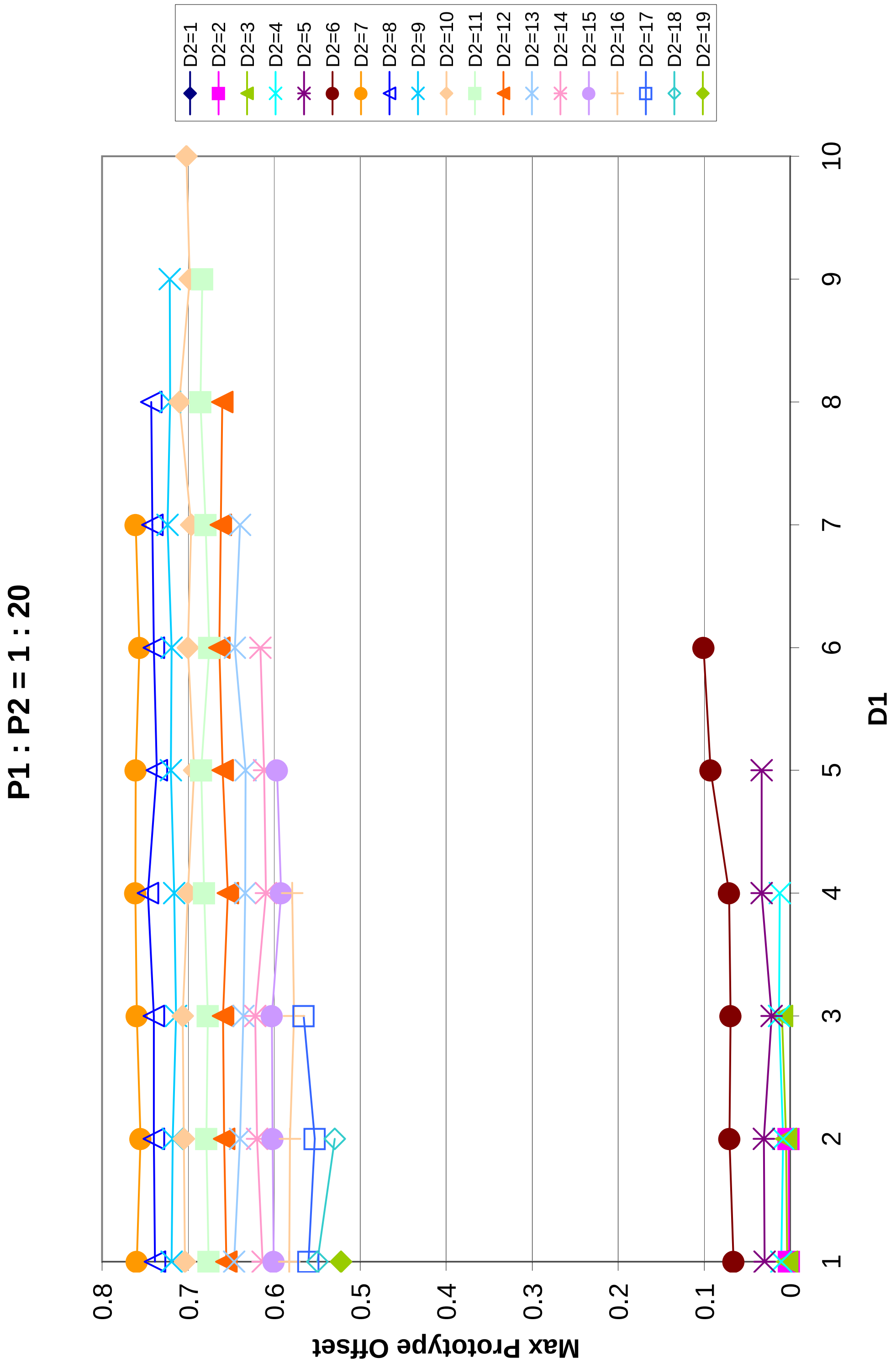


Figure 4.9: Plot of e against $D1$ for $P1 : P2 = 1 : 20$. Each curve has a constant $D2$.

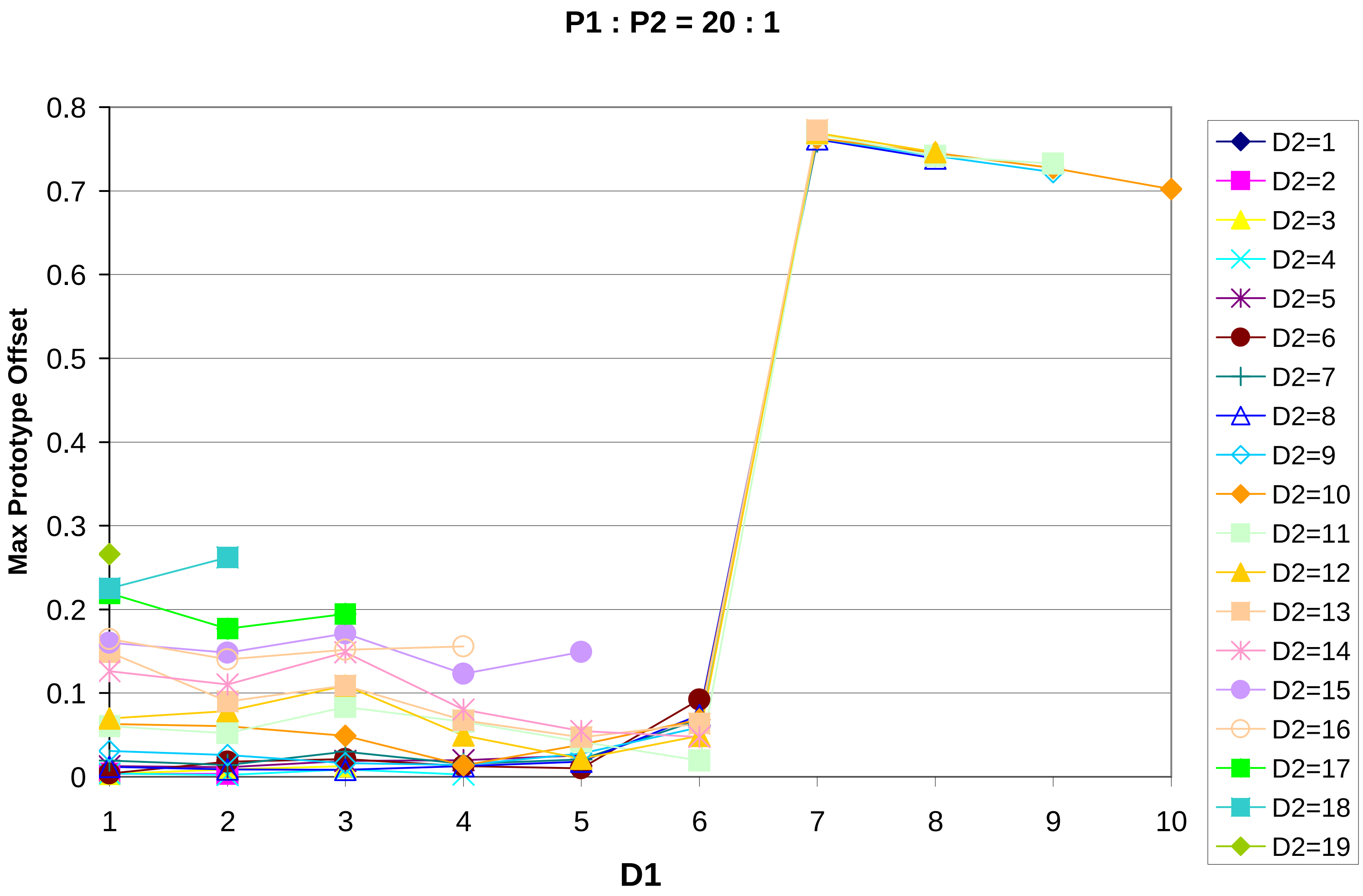
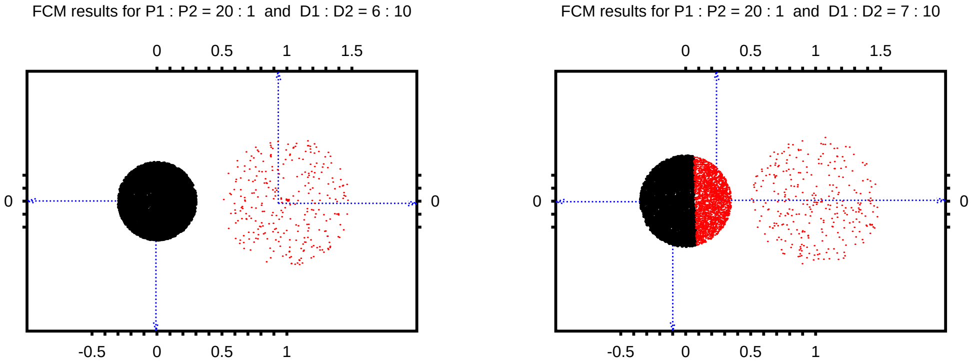


Figure 4.10: Plot of e against $D1$ for $P1 : P2 = 20 : 1$. Each curve has a constant $D2$.



(a) (b)

Figure 4.11: An illustration of the positions of the found prototypes as $D1$ increases from (a) 6 to (b) 7. $D2$ is fixed at 10. This is for $P1 : P2 = 20 : 1$.

the error severer. FCM, effectively, lets clusters with larger populations and larger diameters dominate its solution.

To explain this, let us consider the OF of FCM:

$$\text{Minimise } J_{FCM}(\mathcal{P}, \mathcal{U}; \mathcal{X}, c, m) = \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ik})^m d_{ik}^2(\mathbf{x}_k, \mathbf{p}_i)$$

subject to the constraint

$$\sum_{i=1}^{c} u_{ik} = 1 \qquad\qquad \forall k | k \in \{1, \ldots, N\}.$$

This is a separating-distance-based function that accumulates the weighted distances between the prototypes and the data points. A large cluster (in terms of diameter) will contribute more to the OF than a small one because its distances are higher. Thus, the relative diameters of clusters play a role in determining each cluster's contribution to the OF. In general, a large cluster contributes more than a small one.

The constraint forces a point's membership with a prototype to take into account the point's distances from the other prototypes:

$$u_{ik}^{*} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d_{ik}^{2}}{d_{jk}^{2}}\right)^{1/(m-1)}} \qquad k \in \{1,\ldots,N\}, i \in \{1,\ldots,c\}.$$

Therefore, points that lie very close to a prototype take memberships of almost zero with the other prototypes. However, points lying in the middle between two prototypes will take membership degrees that are close to 0.5. In this way they add to both prototypes's OF contributions. If one prototype can be moved to a position that will "neutralise" these midway points without incurring much penalty from its former neighbourhood, it will be moved. This is because the new location would be close to the optimal solution of the OF.

We also see that the OF is a summation over N. If there is a disparity in the relative diameters of the clusters such that their relative contributions are not equal, the populations play a determining factor. For if the smaller cluster has more points, the small contributions can add up to balance the large cluster's contribution. On the other hand, if a large cluster is more populous, its contribution will dominate the OF. In such a case the accuracy of FCM is further compromised.

In the next Section we will attempt to visualise the shape of the OF of FCM. This will help us to explain the sensitivity of FCM to the middle region between the two prototypes. As we observed in the previous Section, when one of the clusters approaches the diameter level of 7, FCM's accuracy deteriorates significantly.

4.2.3 Shape of FCM Objective Function

We now wish to visualise the shape of the OF of FCM. As before, we assume that there are two cluster prototypes in a two-dimensional feature space. The left-hand prototype is placed at the origin of the coordinate system, $(0, 0)$. The right-hand prototype is placed at coordinate $(1, 0)$. Assuming now that a given data point is placed anywhere

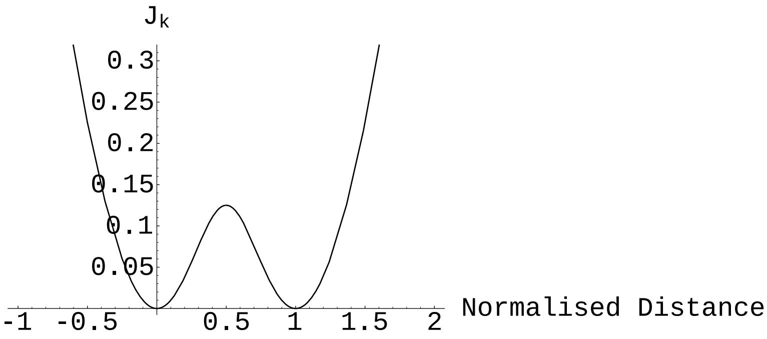


Figure 4.12: FCM ($m = 2$): Plot of point $\mathbf{x}_k$'s OF contribution, J_k, against $\mathbf{x}_k$'s position. $\mathbf{x}_k$ is constrained to move along the x-axis only. The prototypes are located at $(0, 0)$ and $(1, 0)$.

in this 2D feature space, and given an OF, we can calculate the contribution of this point to the OF.

Let us assume that we denote the OF contribution value for a data point, $\mathbf{x}_k$, by J_k. First, let us constrain $\mathbf{x}_k$ to be located along the x-axis. In Figure 4.12, we plot x_k's contribution to the FCM OF versus its location along the x-axis. We left the mathematical derivations of the equation for the curve to Appendix B. From Figure 4.12 we observe each prototype has appropriated symmetrically a region of low cost around it. In the middle between the two prototypes, there is a local peak. A point placed at exactly half-way between both prototypes costs the most amongst points lying between the prototypes. Furthermore, as a point heads away from the prototypes, its cost rises steeply.

Now we allow the location of $\mathbf{x}_k$ to move freely in the 2D space. Thus, we can plot contour lines around the prototypes; points lying on these contour lines contribute identical values towards the OF. Such a contour plot is illustrated in Figure 4.13. We observe again that FCM creates symmetrical contours around the prototypes. As a generalisation of Figure 4.12 in 2D, we observe that the rate at which contributions change in the “valleys” around each prototype is less than further afield. Once again,

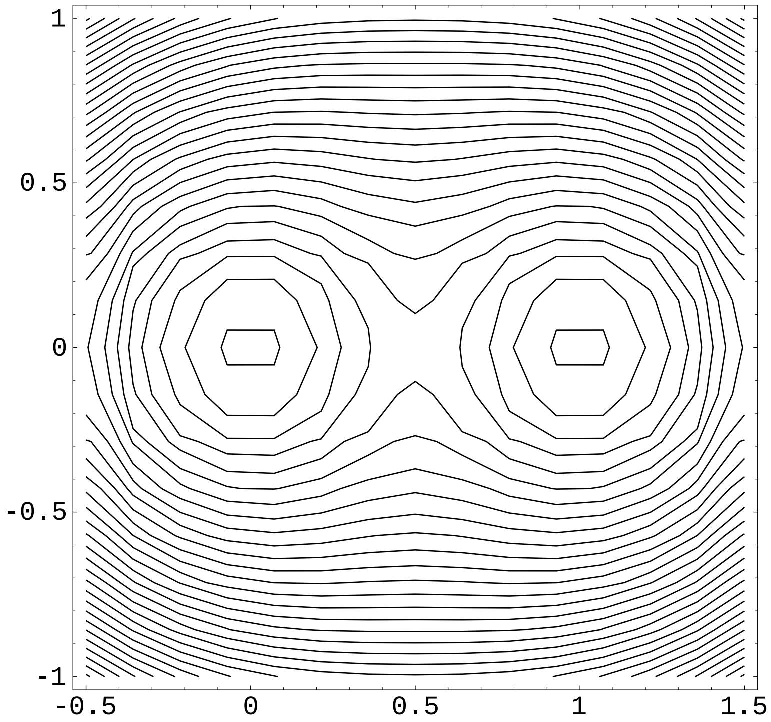


Figure 4.13: FCM ($m = 2$): Contour plot of J_k, representing $\mathbf{x}_k$'s OF contribution; J_k's value depends on $\mathbf{x}_k$'s location in the 2D space.

we left the mathematical derivations to Appendix B. Based on the contour plot we can see the shape of ideal clusters for FCM, and we can guage how well it will perform given any particular constellation of points.

A point of note is that if we were to integrate the area under the curve between $(0, 0)$ and $(1, 0)$ in Figure 4.12, what would that represent? It would represent the total contribution of a continuous line of data points along the x-axis between the two prototypes. Let us now work out the bounds of a region centered around the mid-point which would cover only half of the computed area under the curve. The significance of this computation would be to find the region, along the line between the prototypes, that contributes as much as the remaining parts of the line. The bounds, as worked out in Appendix B, are the points $(0.38, 0)$ and $(0.62, 0)$. In our benchmark data suite, these approximate to values for either D_1 or D_2 of between 7 and 8. This is confirmed by our results as previously presented. This calculation also shows that there is a relatively narrow region of width 0.24 centred around the half-way point which “costs” FCM twice as much as either region to the side of it.

4.3 Population-Diameter Independent Algorithm

When we first investigated FCM, we focused on its inability to cluster accurately when the data set contains one cluster which is highly populated in comparison to the other (in a two-cluster case). Thus, we thought of dividing each cluster’s contribution to the objective function by its population. This way, the new ratio of one cluster’s contribution to another would not be as disproportionate as the old one. In other words, the lightly-populated cluster’s contribution would be increased, and that of the highly-populated one decreased.

However, upon further study, as evidenced above, we concluded that as well as the populations problem, there is also another problem. This occurs when there is a sharp difference in the spans of the clusters (represented by diameters in our experiments) *and* the larger cluster’s span reaches into the middle region between the two. The diameters problem can either be compounded or alleviated by the populations problem depending on the populations-ratio and which cluster it favours. Thus, we concluded that the effects of population and diameter are correlated and it would not be easy to compensate for their effects separately. We found it is more precise to talk about the “relative contributions” of clusters

Obviously, FCM’s objective function does not account for these effects. This is why we introduced the Population-Diameter Independent (PDI) Algorithm. The main idea behind our new algorithm is to normalise the cluster contributions found in the FCM objective function. Thus, in PDI’s objective function, we divide each cluster’s (FCM) contribution by a number that should represent the strength of the contribution. The result of the division would give the cluster’s new (PDI) contribution.

If we were to set no constraints on these “normalisers” they would take infinite values because the OF is being minimised. Therefore, we constrained the sum of the normalisers to 1. This means if one normaliser increases in value, at least one other normaliser must decrease in value. A minimal solution would assign lower-valued normalisers to clusters with small contributions and, correspondingly, higher-valued normalisers to clusters with big contributions. If clusters contribute roughly equally to the OF then the normalisers should take the value $1/c$, where c is the number of clusters.

We named the normaliser variables ρ. Thus, ρ_i is the normaliser for cluster i. In order to allow the user to vary the influence of the ρ's, we raised them to the exponent $r, r \geq 0$. We now formally state our formulation of the optimisation problem.

4.3.1 The New Objective Function

$$\text{Minimise } J_{PDI}(\mathcal{P}, \mathcal{U}, \underline{\rho}; \mathcal{X}, c, m, r) = \sum_{i=1}^{c} \frac{1}{\rho_i^r} \sum_{k=1}^{N} (u_{ik})^m d_{ik}^2(\mathbf{x}_k, \mathbf{p}_i) \qquad (4.1)$$

subject to the constraints:

$$\sum_{i=1}^{c} u_{ik} = 1 \qquad (4.2)$$

and

$$\sum_{i=1}^{c} \rho_i = 1 \qquad (4.3)$$

From the above formulation we can derive an algorithm to achieve a minimal solution. This is effected by means of using the Lagrange multiplier method, setting the differentials to zero, obtaining the update equations for each variable, and then using the Picard successive substitution strategy, as was used with FCM. We leave the derivation of the update equations to Appendix C and now only state them.

4.3.2 Conditions for Optimality

Let the minimisers of $J_{PDI}(\mathcal{P},\mathcal{U},\underline{\rho})$ be called $(\mathcal{P}^*,\mathcal{U}^*,\underline{\rho}^*)$. The necessary conditions for $(\mathcal{P}^*,\mathcal{U}^*,\underline{\rho}^*)$ are :-

$$u_{ik}^* = \frac{(\rho_i^r/d_{ik}^2)^{1/m-1}}{\sum_{i=1}^{c}(\rho_i^r/d_{ik}^2)^{1/m-1}}, \tag{4.4}$$

and

$$\mathbf{p}_i^* = \frac{\sum_{k=1}^{N} u_{ik}^m \mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^m}, \tag{4.5}$$

and

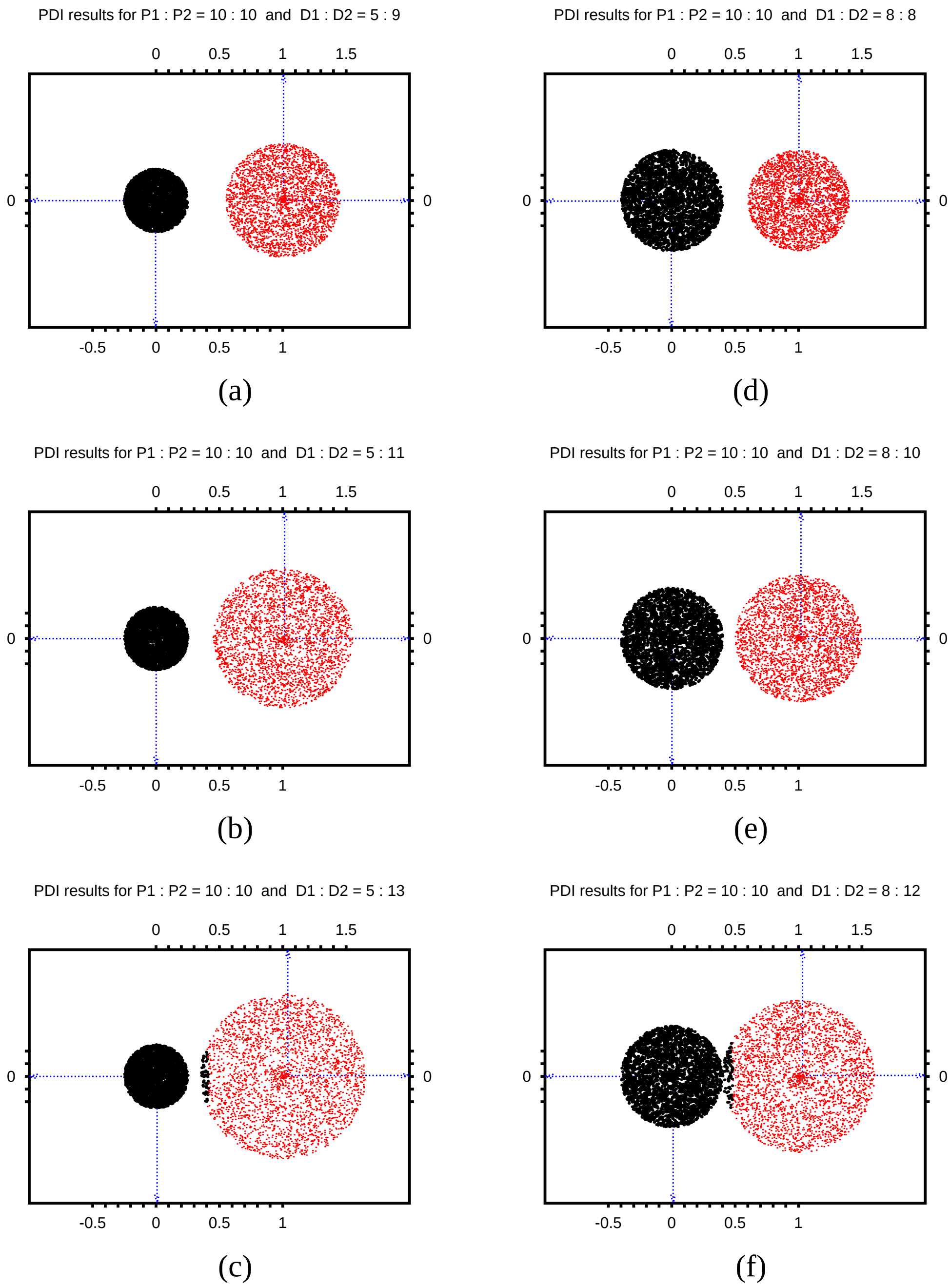
$$\rho_i^* = \frac{[\sum_{k=1}^{N}(u_{ik})^m d_{ik}^2]^{\frac{1}{r+1}}}{\sum_{i=1}^{c}[\sum_{k=1}^{N}(u_{ik})^m d_{ik}^2]^{\frac{1}{r+1}}}. \tag{4.6}$$

We note that the optimality condition for ρ_i has intuitive meaning; it is a ratio of cluster i's contribution to the sum of all the clusters' contributions. The equations also confirm that, as with the OF, setting $r = 0$ collapses PDI to FCM.

4.3.3 PDI's Improvement on FCM

We now present a summary of PDI's performance on the benchmark suite. As with FCM, we used the max rule to the de-fuzzify the clustering results. We also used the same initialisation as we did with FCM. Based on our experience (described in the next Section), we empirically set $r = 1.0$. Similarly to FCM's plots, PDI's plots display both classification results as well as location of found prototypes.

We start with Figure 4.14, the plots can be compared directly to those of FCM shown in Figure 4.4. Through visual assessment, we can observe a great overall improvement in clustering accuracy. The data sets of Figures 4.14(a),(b),(d), and (e) were clustered perfectly. Figures 4.14(c) and 4.14(e) were not, however, compared to FCM, PDI's performance is a great improvement.

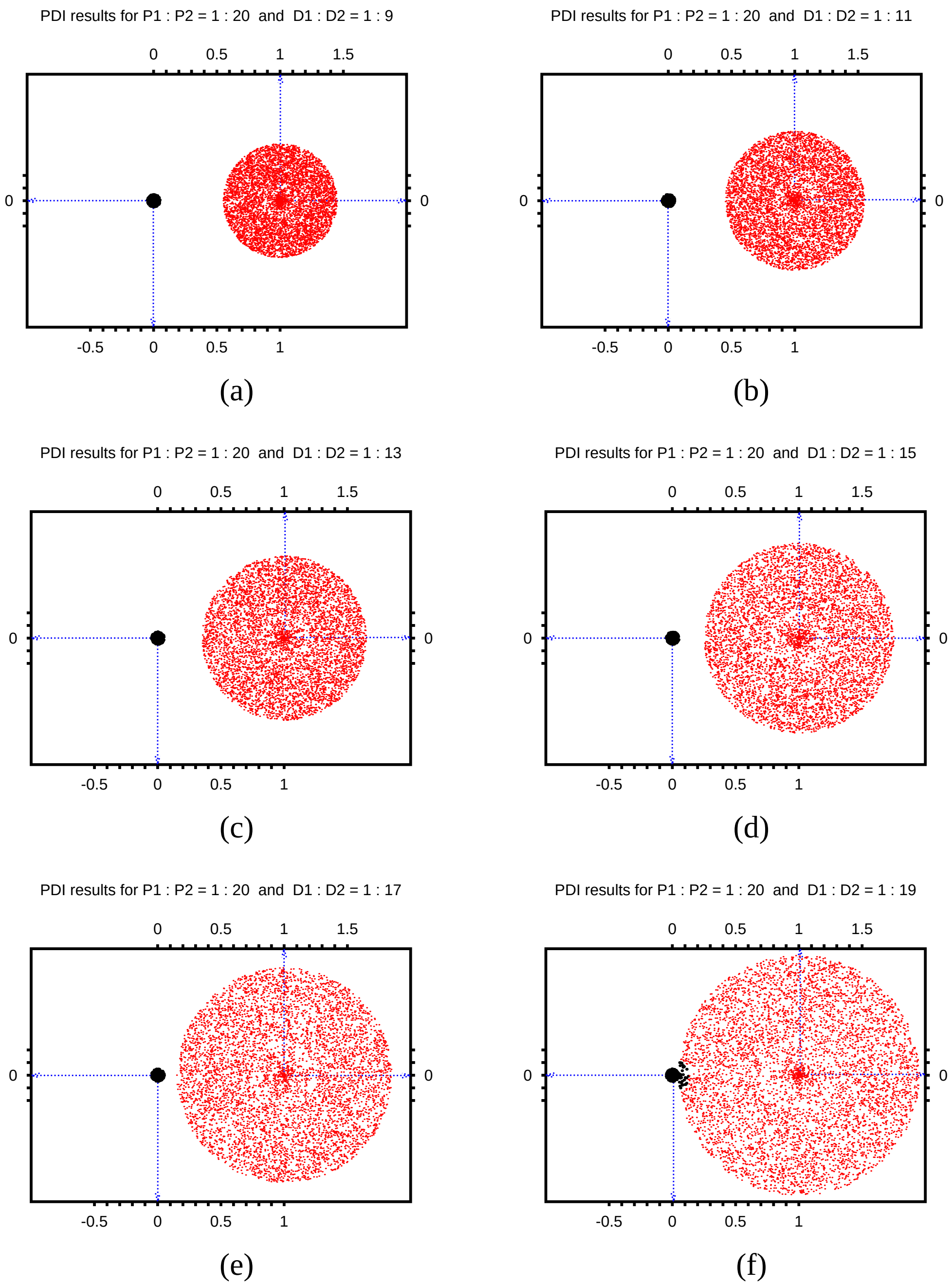


(a) (d)

(b) (e)

(c) (f)

Figure 4.14: PDI clustering of synthetic dot patterns with two colours representing the two found clusters. Prototypes are marked out by the dotted blue lines. Compare with the results in Figure 4.4.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.15: PDI results as $D2$ increases from (a) 9, (b) 11, (c) 13, (d) 15, (e) 17, to (f) 19. $D1$ is fixed at 1. This is for $P1 : P2 = 1 : 20$.

Figure 4.15 shows more PDI results. Here, fixing $P1 : P2 = 1 : 20$ and $D1 = 1$, we observe how PDI copes extremely well with increasing $D2$ incrementally. It is only in the difficult case of $D2 = 19$ that PDI's accuracy is compromised.

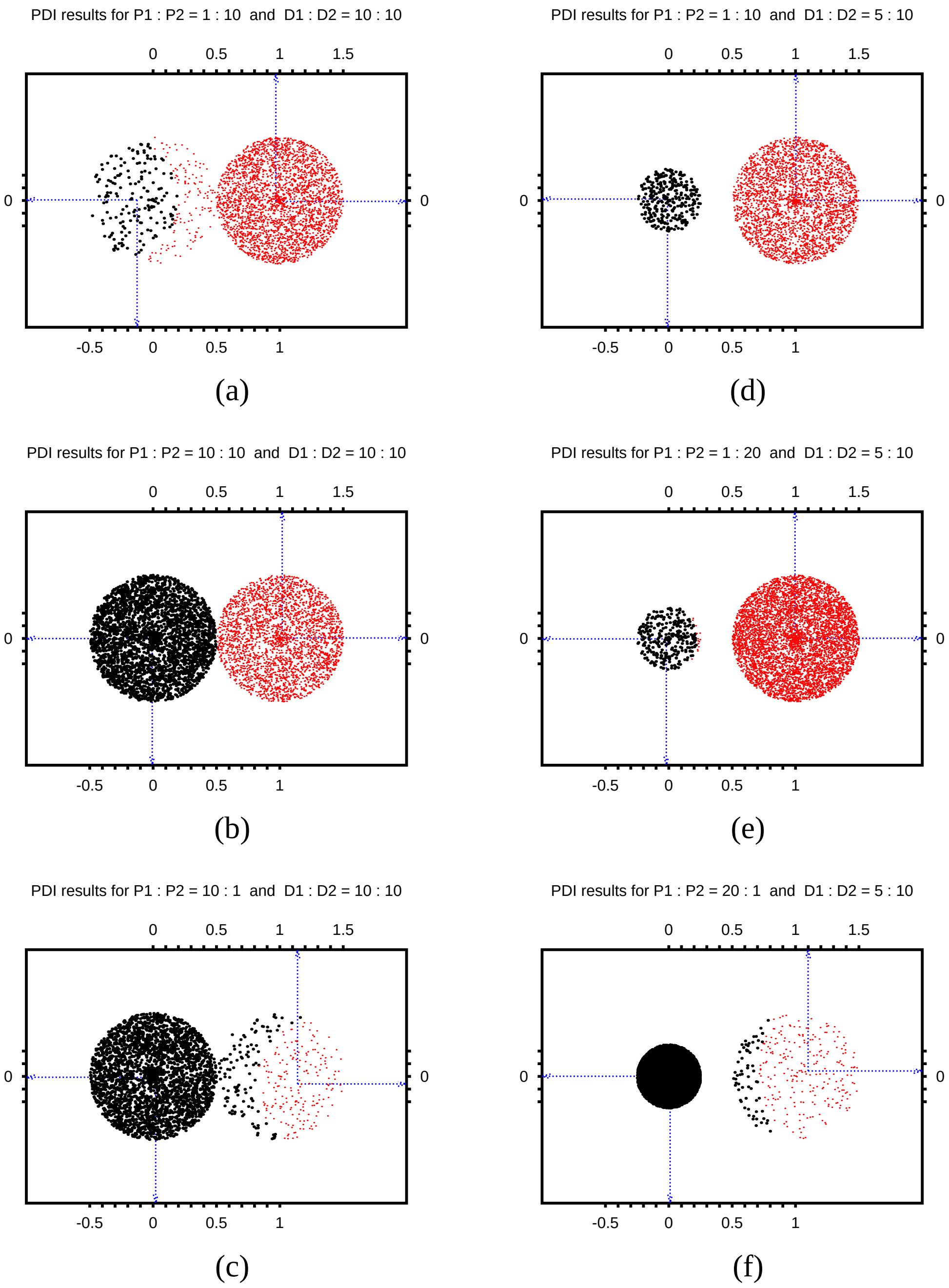
Figure 4.16 continues with more PDI results, the plots can be compared to those of FCM shown in Figure 4.5. In column (a), (b), and (c) of Figure 4.16, the case of two equal-sized, touching clusters ($D1 : D2 = 10 : 10$) is tested with changing population ratios. Here we observe an interesting behaviour of PDI: it finds a cluster within a cluster. This behaviour is also observed in Figure 4.16(f) where population ratios are varied while the diameters remain fixed at $D1 : D2 = 5 : 10$. This anomaly of finding a cluster within a cluster is due to the light density of one of the clusters as compared to the other. Because of the light density, the contribution is weak and thus the corresponding cluster-normaliser takes a low value. This in turn marks a smaller region of influence for the cluster prototype. We explain this in more detail in Section 4.4.

We now plot the improvement of PDI over FCM in a summarised manner, as corresponds Figures 4.6—4.10. In these plots, we use $e_{FCM} - e_{PDI}$ as our measure of PDI's improvement on FCM.

We start with all data sets with a $D1 = 1$ configuration and plot the improvement in Figure 4.17. The plot resembles almost exactly that of e_{FCM} in Figure 4.6. Thus, it confirms that PDI effectively equalises disproportionate objective-function contributions for configurations of $D1 = 1$.

We now compare Figure 4.18 to Figure 4.8 where $D1 = 5$. We observe effective correction of FCM - except for configurations $P1 : P2 = 10 : 1$ and $P1 : P2 = 20 : 1$. Here due to the behaviour mentioned above, namely, identifying a cluster within a cluster, FCM actually performs better than PDI. Nevertheless, FCM's margin of improvement is not a big one - not exceeding 0.15.

Comparing Figure 4.19 to Figure 4.9, where we fix the population ratio at $P1 :$



(a) (d)

(b) (e)

(c) (f)

Figure 4.16: PDI clustering: D1:D2 fixed at 10:10 for column (a), (b), and (c), and 5:10 for column (d), (e), and (f). The P1:P2 ratio is varied. Compare with the results in Figure 4.5.

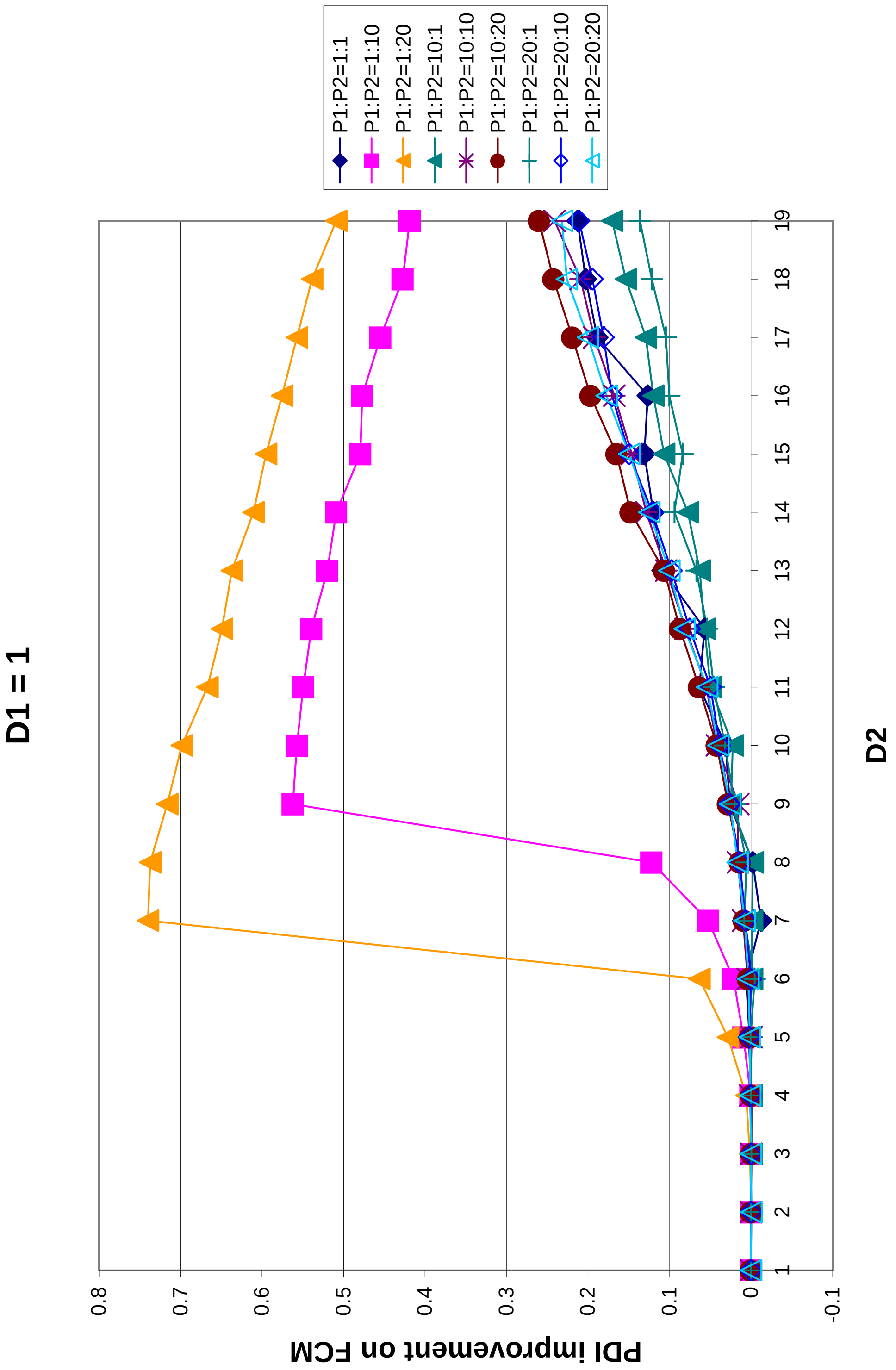


Figure 4.17: Plot of $e_{FCM} - e_{PDI}$ against $D2$. $D1 = 1$. All nine population configurations are shown. Each curve has a constant population ratio.

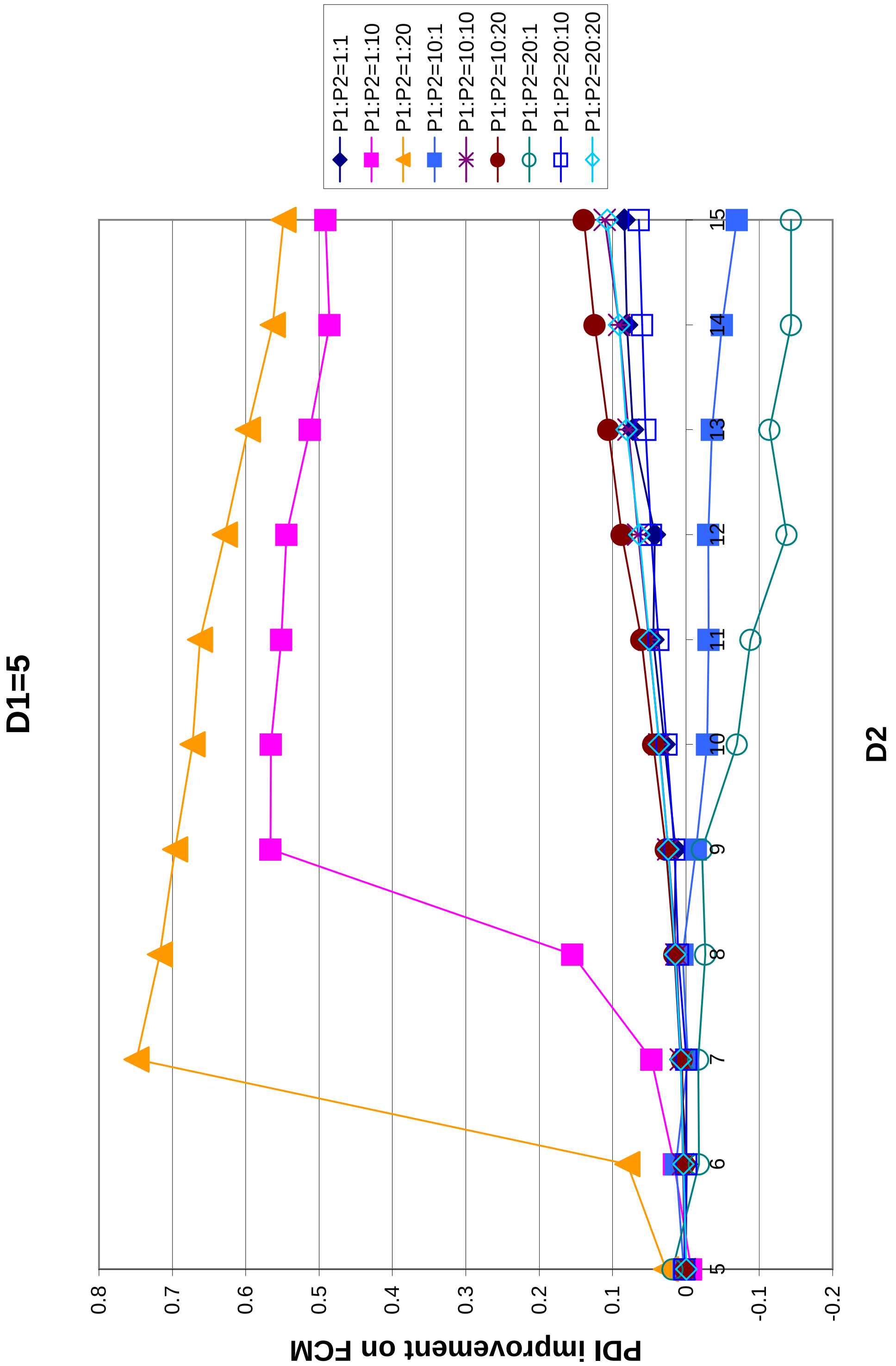


Figure 4.18: Plot $e_{FCM} - e_{PDI}$ against $D2$. $D1 = 5$. Each curve has a constant population ratio.

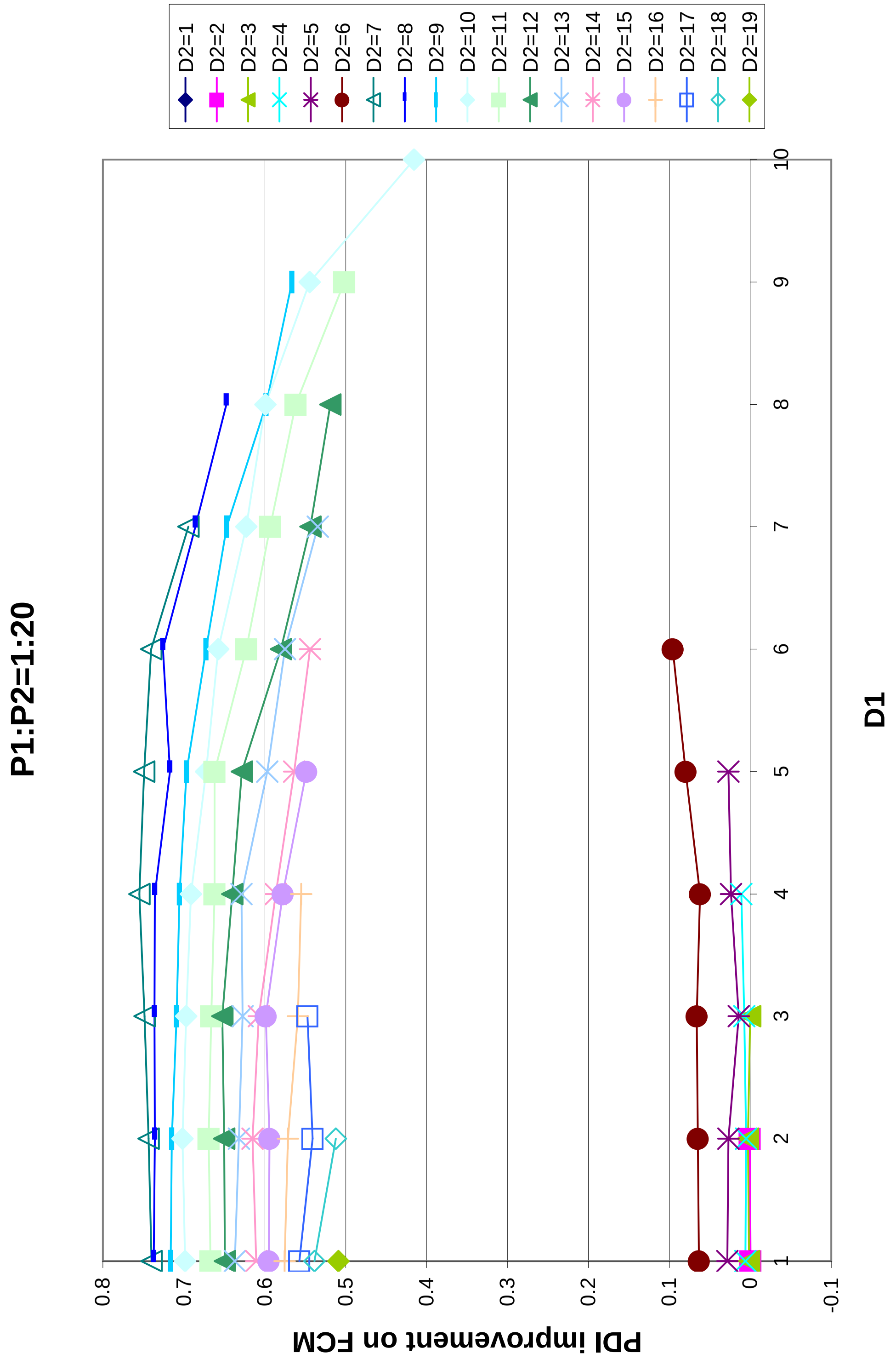


Figure 4.19: Plot of $e_{FCM} - e_{PDI}$ against $D1$ for $P1 : P2 = 1 : 20$. Each curve has a constant $D2$.

$P2 = 1 : 20$, we observe that whereas PDI effectively corrects FCM for values of $D1$ less than 6, its performance declines afterwards. However, PDI still retains a margin of improvement over FCM for values of $D1 > 6$. The decline in performance is due to the fact that at $D1 > 6$ the LHS cluster becomes of such light contribution that correct placement of its prototypes would necessitate a small value for the corresponding normaliser, thus the prototype moves towards the left and PDI identifies only a subsection of the cluster.

Finally, comparing Figure 4.20 to Figure 4.10, where we fix the population ratio at $P1 : P2 = 20 : 1$, we observe that the plot follows the same trend as FCM's except that it ventures below zero for values of $3 \leq D1 \leq 6$. This is the same behaviour as mentioned above. Once again, we note that the margin of error is not great and that for most cases PDI effectively corrects for FCM shortcomings.

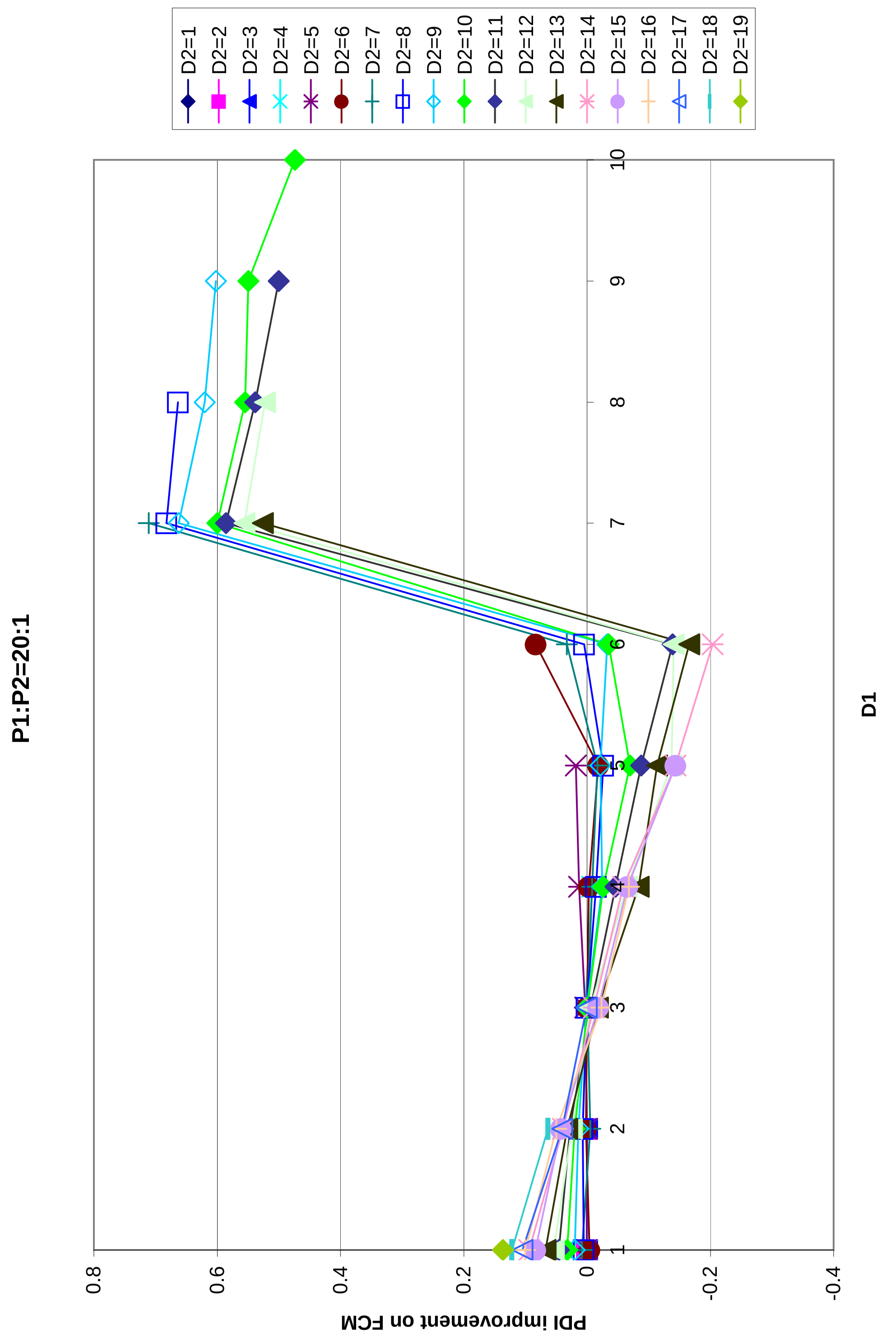


Figure 4.20: Plot of $e_{FCM} - e_{PDI}$ against $D1$ for $P1 : P2 = 20 : 1$. Each curve has a constant $D2$.

4.4 Observations on PDI

The avenues of enquiry that PDI opens are quite numerous. In this Section we observe the shape of the OF of PDI and compare it to that of FCM. We also touch on our experience with the r exponent, and with PDI's resilience to different initialisations.

4.4.1 Shape of Objective Function

In Figure 4.21, we show PDI's point-contribution contour plot which corresponds to that of FCM in Figure 4.13. Recall that the contours around each prototype indicate progressively more expensive point-locations. We observe that setting $\rho_0 = 0.2$ has caused contraction around that cluster's prototype, and a corresponding expansion around the other prototype, compared to the symmetrical contours of FCM.

If we move along only the x-axis and plot the J_k curve, Figures 4.22—4.25 show the variations caused by different values of ρ and r. These can be compared to that of FCM in Figure 4.12.

In Figure 4.22, $r = 1$ and $\rho_0 = \rho_1 = 0.5$. The shape of the curve is exactly the same as for FCM: two symmetrical valleys around each prototype. The OF magnitudes are not, however, directly comparable.

In Figure 4.23, maintaining $r = 1.0$, we emphasise the LHS prototype by setting $\rho_0 = 0.1$. We observe that this causes a thinner valley around the LHS prototype as compared to that of the RHS prototype.

In Figure 4.24, we maintain $\rho_0 = 0.1$, but increase the r exponent to $r = 1.5$. We observe this causes a sharper, thinner valley around the LHS prototype and increases the scope of the RHS prototype. Thus, r can be increased when searching for tiny clusters.

In Figure4.25, where $r = 0$ and $\rho_0 = 0.1$, we observe this causes an exact same

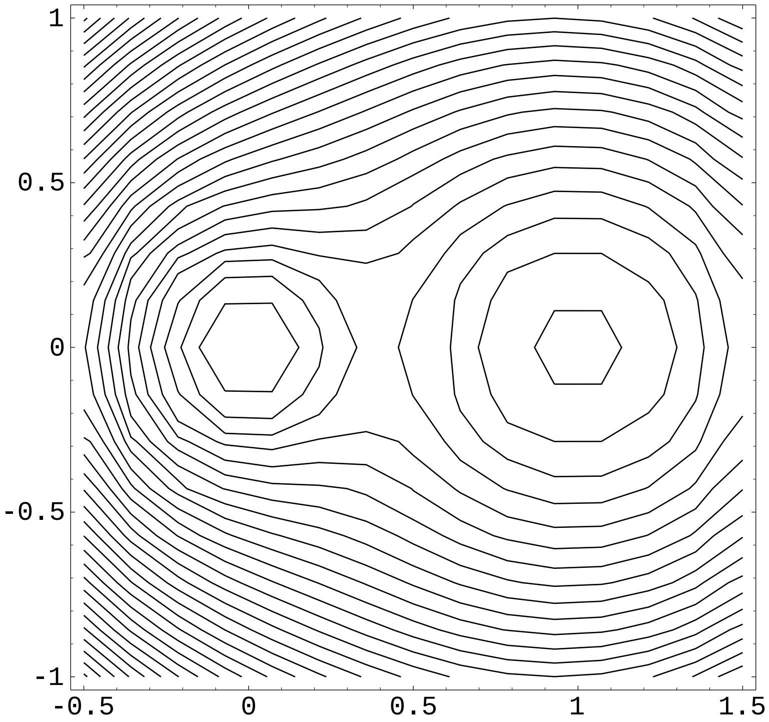


Figure 4.21: PDI ($m = 2, r = 1, \rho_1 = 0.2$): Contour plot of J_k, representing $\mathbf{x}_k$'s OF contribution; J_k's value depends on $\mathbf{x}_k$'s location in the 2D space. Compare with Figure 4.13.

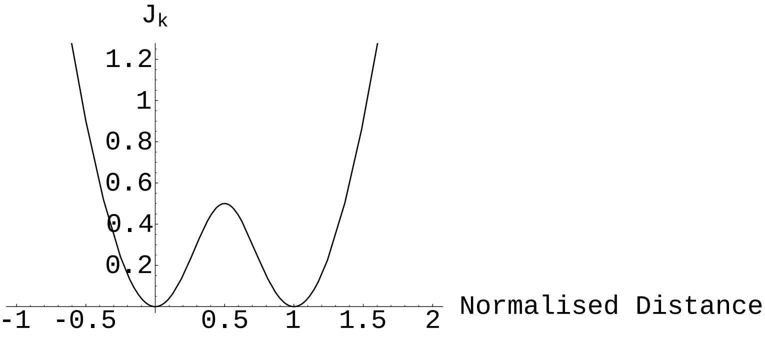


Figure 4.22: PDI ($r = 1$, $\rho_0 = 0.5$): Plot of a point's OF contribution against its position with respect to two prototypes given that both prototypes have equal ρ's of 0.5 each.

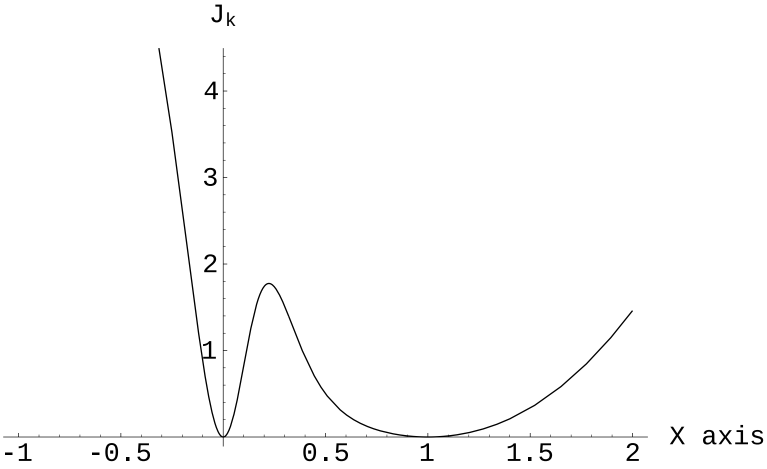


Figure 4.23: PDI ($r = 1$, $\rho_0 = 0.1$): The plot forms a thin valley around the LHS prototype, thereby giving a wider "scope" to the RHS prototype.

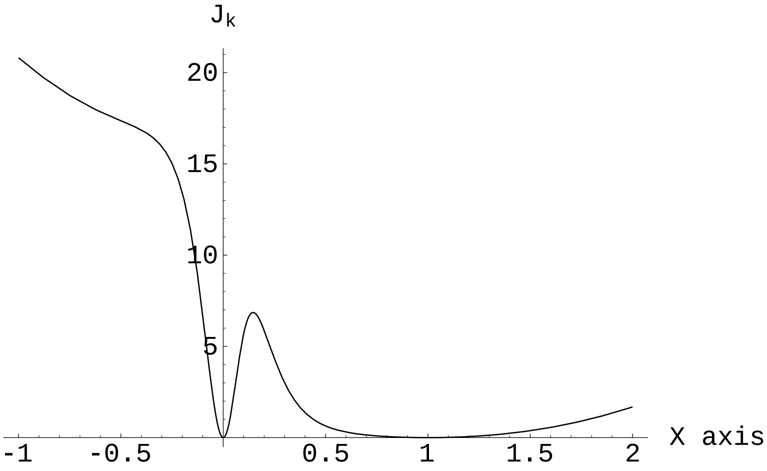


Figure 4.24: PDI ($r = 1.5$, $\rho_0 = 0.1$): Raising r's value causes even stronger emphasis around the LHS prototype, and a much wider scope around the RHS prototype.

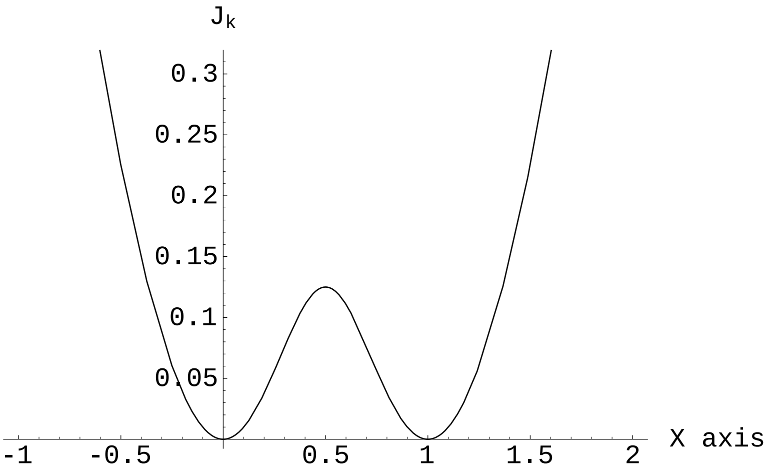


Figure 4.25: PDI ($r = 0$, $\rho_0 = 0.1$): Despite the low value of ρ_0 PDI's OF collapses to FCM's symmetrically-shaped one because r was set to 0. This plot is equal in magnitude to that of Figure 4.12.

curve to that of FCM's since $r = 0$ collapses PDI to FCM.

4.4.2 Varying the r-Exponent

The exponent of the normalisers ρ plays an important role in how PDI performs. The higher its value, the sharper the emphasis of the normalisers. The lower its value the more PDI resembles FCM. In Figure 4.26 we demonstrate the results of applying PDI at various values of r to a data set similar to those in our suite.

At $r = 0$, the results are identical to FCM. At $r = 0.5$, the boundary between both classes becomes slightly curved, indicating that the normalisers have begun to have some effect. Beginning at $r = 2.4$, we see that PDI classified a subset of the small cluster as a cluster of its own. At $r = 3.0$ only one point in the small cluster is identified! The small-cluster prototype is placed at the ideal location. This result indicates that PDI "spotted" the small cluster. However, this result is very sensitive to the initialisation. Our experience is that if the initialisation is far away from the ideal

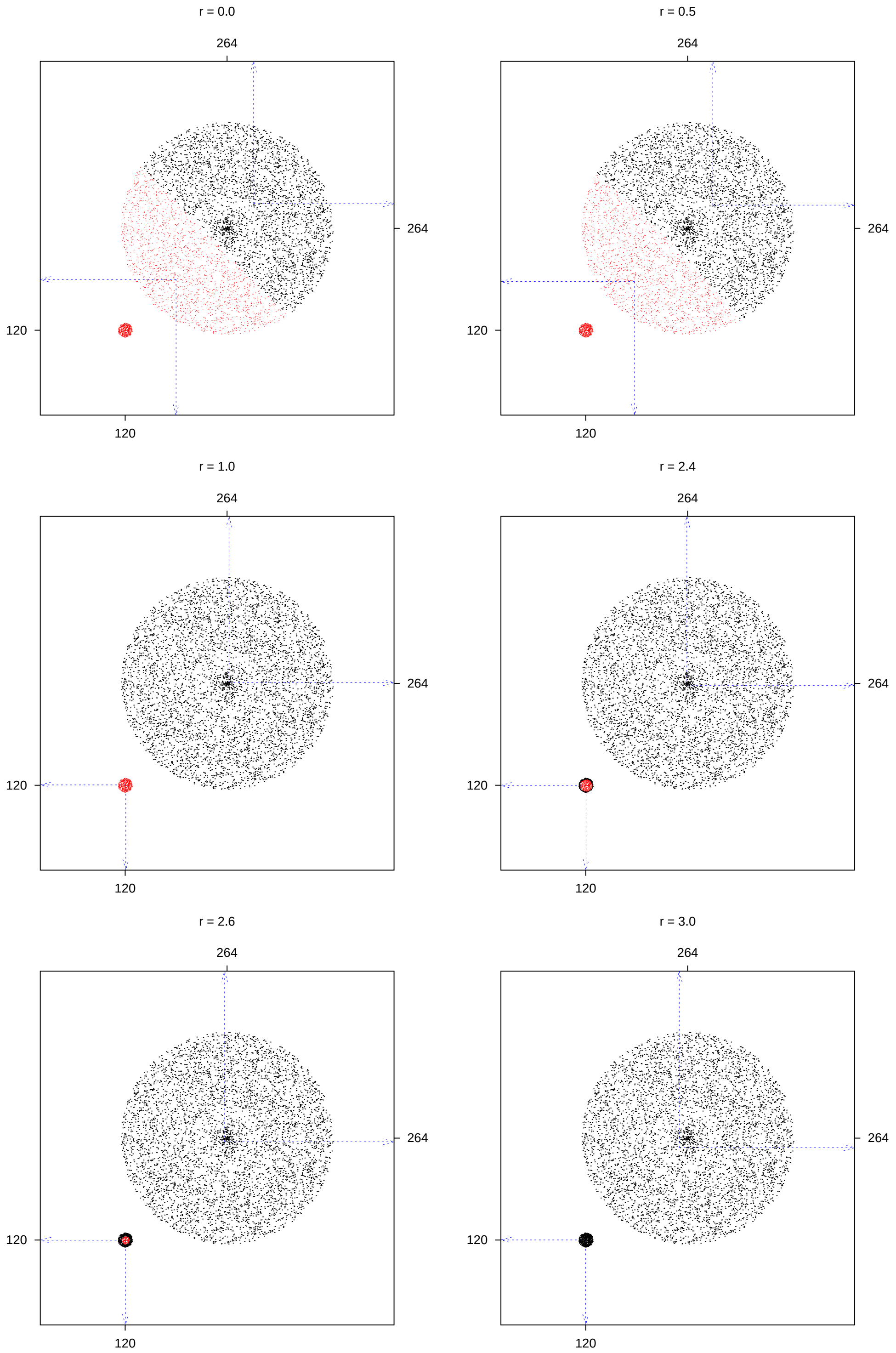


Figure 4.26: Results on varying r in PDI. r's value is labelled at the top of each graph. $r = 0$ renders PDI to be FCM. An interesting behaviour happens at $r = 2.4$.

r	ρ_1	ρ_2
0.0	0.468998	0.531002
0.5	0.444052	0.555948
1.0	0.032148	0.967852
1.2	0.031990	0.968010
1.4	0.037311	0.962689
1.8	0.050938	0.949062
2.0	0.057516	0.942484
2.2	0.063063	0.936937
2.4	0.066393	0.933607
2.6	0.061108	0.938892
2.8	0.000000	1.000000
3.0	0.000000	1.000000
4.0	0.000000	1.000000

Table 4.2: The effect of varying r on ρ_1 and ρ_2 in the data set of Figure 4.26.

locations, different solutions will be found.

In Table 4.2, the different values of r we used are tabulated against the corresponding values for the normalisers ρ_1 and ρ_2. ρ_1 represents the small cluster. At $r = 0$, the normalisers are approximately balanced. At $r = 1.0$, a steep descent in the value of ρ_1 is clearly observed and the solution found is the correct one. The ratio of ρ_2/ρ_1 here is about 30. At $r = 2.4$, the "aperture" of the small cluster begins to narrow and by $r = 2.8$ it has become only wide enough for a very small number of points. The points are located around the ideal location for the prototype. The solution is therefore technically correct! However, as mentioned above, this solution is sensitive to initialisation.

We further observe that at values of r, $r \geq 2.8$, the results became of doubtful use. It is clear some form of divergence has occurred. In algorithmic implementations of PDI such behaviour can be prevented by checking if one of the normalisers is heading towards an infinitesimally small value.

On inspecting Table 4.2, we can speculate that at $r = 1.2$ the best "tuning" of

PDI's performance was achieved. This we can justify on the basis that the normaliser ρ_1 is at a local minimum with respect to its other values as r is varied. So, it would be interesting to conduct a study on tuning the value of r based on the variation of the normalisers and finding out if the tuned value correlates with better clustering accuracy.

4.4.3 Resilience to Initialisation

We investigated PDI's sensitivity to initialisation. Though we do not include the results here, our conclusion is that for values of r, $0.5 \leq r \leq 1.5$, PDI's solution (as found by our iterative implementation) is usually a stable one and is quiet resilient to the different initialisations. Higher values of r cause turbulence in the shape of the objective function. Iterative implementations like ours get entrapped in locally-optimal solutions.

4.5 Summary and Conclusions

In the early parts of this Chapter, we established a shortcoming of FCM: its clustering accuracy drops sharply in situations where there are small clusters lying close to large ones. We rectified this shortcoming by introducing cluster-strength variables, one per each cluster, to normalise cluster contributions. In this way, solutions that identify the clustering structure correctly become optimal - in the eyes of the PDI OF.

The OF of FCM weights each point-to-prototype distance with a membership degree. This way, points close to a prototype get high degrees of membership because they contribute little to the OF's value. The OF of PDI goes further by weighting each cluster's (FCM) contribution to the OF with normalisers. This way, clusters that contribute more acquire large normalisers to minimise their impact, and small normaliser values must be allocated to the other clusters, thus allowing them to be represented.

The rationale for the weighting mechanism in FCM is to place one prototype in the middle of each group of points in the data set. The rationale for PDI's additional weighting mechanism is to allow small clusters to be represented. For FCM, prototype locations determine membership values and, therefore, the value of the OF. For PDI, prototype locations are matched with normaliser values and together they determine the membership values and, therefore, the value of the OF. Thus, normalisers grant a scope to each prototype that matches the prototype's relative contribution.

To fully assess this new algorithm, we reported in full its results on a variety of data sets. We also proved that PDI remedies FCM's shortcoming. Our new OF has on the other hand shown a shortcoming of its own. This shortcoming is that it may over-emphasise small, compact clusters. It is also very sensitive to the value of r.

Our approach in this Chapter has been a fundamental one. We set up an idealised framework and accordingly designed data sets to test specific hypotheses. We believe new clustering algorithms, particularly ones derived from or similar to FCM, should be tested on the specific behavioural properties we raised in this Chapter using our data sets.

In the next Chapter, we present our experience with the use of clustering for image analysis.

CHAPTER 5

Clustering of Medical Images

Fuzzy clustering provides a good framework for medical image analysis. The main advantages of this framework are that it provides a way to represent and manipulate the fuzzy data contained in medical images, and that it provides flexibility in presenting extracted knowledge to clinicians and radiologists.

This Chapter discusses the issues involved in the analysis of images in general, but with particular attention to medical images, using fuzzy clustering methods. Since segmentation is often considered the main step in the image analysis process, we will mainly be discussing the segmentation of medical images using clustering.

We first give a brief background on medical imaging and the main medical imaging modalities involved. In Section 5.2, a segmentation framework based on clustering will be outlined; the decision points within this framework: feature extraction, method, and post-processing, will be discussed. Continuing on our work in the previous Chapter, in Section 5.3, we describe a synthetic 2D model of cardiac images on which we compared the performances of FCM and PDI.

5.1 Medical Image Analysis

Medical imaging has developed exponentially in the past few years in terms of technological advance and wide-spread use. High-resolution, three-dimensional anatomical information can now be obtained in a routine manner with magnetic resonance imaging (MRI) and X-ray computer-aided tomography (CT). These two modalities provide complementary information; CT shows detail of bony structures and some contrast between hard and soft tissues while MRI shows detail of soft tissue structures, with almost no detail of bony structures. CT imaging, like all X-ray techniques, exposes the patient to a dose of X-rays, thus, incurring some health risks. MRI does not expose the patient to radiation, but uses the magnetic properties of the patient's tissues to provide contrast in the image, and as far as we know at present it is completely harmless.

In our research, we focused on cardiac MR images. In common with much medical image analysis work, our images may be used to gain anatomical knowledge of the patient being studied so that diagnostic decisions may be taken. To aid in this, quantitative measures may be calculated or a qualitative analysis may be reported. Thus, segmentation of this type of images is a necessary step.

5.2 Segmentation as a Process Involving Clustering

There is strong similarity between "clustering a data set" and "segmenting an image". Both these processes share the goal of finding "true" classification of the input. "True" here depends very much on the application at hand. In general, however, there is a stronger requirement for accuracy placed on the segmentation process. This is mainly because while the data processed by clustering methods may not represent a physical reality, medical images represent physical anatomy.

The general clustering process, because of its exploratory nature, has license to interpret and may be imprecise. Its main strength is that it is unsupervised, *i.e.*, it does

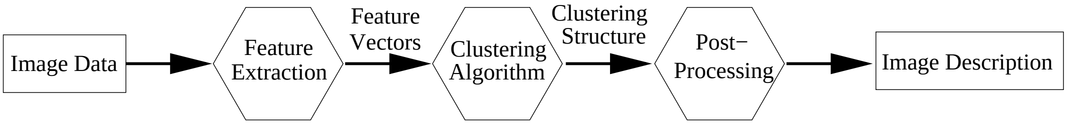


Figure 5.1: The process of clustering image data for the purposes of segmentation.

not require any training data, and is automatic (requires minimal user interaction). Segmentation methods on the other hand are not generally required to interpret, but instead have to be accurate. While many segmentation methods require training data and are only semiautomatic, automatic methods are welcome since they require no training effort, or human resources.

Segmenting images using clustering defines three decision points for the process, as shown in Figure 5.1. The first decision point that arises is: how will we present the image data to the clustering algorithm? This we have named feature extraction and we address below. The next decision point is: what algorithm do we choose to run on the data, and of course, how do we set it up? In response to this, we have already discussed a variety of algorithms in Chapters 2 and 3 and so we will not discuss this further in this Chapter. Embedded in any algorithm chosen, will be the question of choice of distance metric by which to measure the similarity between two constituent points in the extracted data set. The last decision point is: how do we use the output of the clustering method? In some cases, all that may be needed is a suitable colouring scheme or similar human-computer-interaction device so that clinicians (experts) can use the results easily. In Section 5.2.2, we discuss some of the methods to post-process the output of fuzzy clustering methods.

Arguably, workers in the field of image analysis have dealt with the above three questions with increasing sophistication over the past two decades. About twenty years ago, most researchers made straightforward choices when clustering image data [Schachter *et al.*, 1979; Mui *et al.*, 1977]. Recent works have delved deeper into the workings of a clustering algorithm, in some cases modifying it specifically for the application. For example, in [Kottke, 1992] a feature-weighting mechanism that utilises variance estimates is incorporated into the clustering process. In [Tolias & Panas, 1998b; Tolias & Panas, 1998a] an iterative scheme that adapts to the local image characteristics by imposing spatial constraints on the fuzzy partition matrix is used during the clustering process. In [Pham & Prince, 1999] multiplicative intensity inhomogeneities are compensated for by allowing the prototypes for each cluster to vary across the image.

Also, new metrics specifically designed for image data have been proposed. For example, in [Udupa & Samarasekera, 1996] the notion of "fuzzy connectedness" is introduced as a natural, but computationally complex, measure of distance best-suited to images. Also, in [Gath *et al.*, 1997] a data-induced measure of distance was introduced for the purpose of extracting non-convex patterns in images.

The pragmatic idea of carrying out the three steps of Figure 5.1 and then repeating them in order to produce better results has also been considered in the literature. For example, in [Bensaid *et al.*, 1996] an automatic evaluation of the segmentation result is formulated so that based on this evaluation, the process is repeated with a new set of parameters beginning at the second step.

5.2.1 Feature Extraction

We now address three ways in which image data may be presented to a clustering algorithm. These are: using only the voxel intensities, using the voxel intensities and spatial coordinates, and extracting locality measures from the image data. We have called this step feature extraction because, in the data analysis framework, clustering methods work on "feature vectors".

In general, image data arrive in the form of one or more 2-D, 3-D, or even 4-D (including time) data lattices containing the image measurements, or intensities. Every

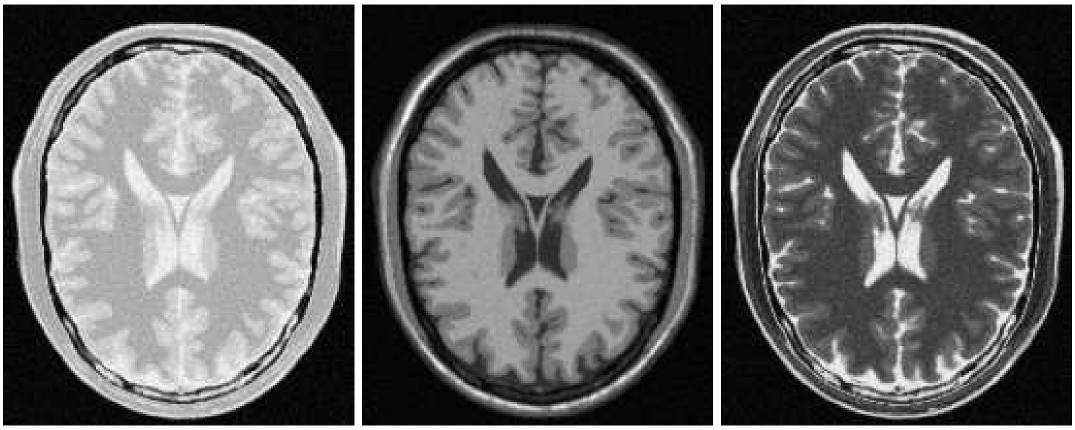


Figure 5.2: An example of three different images obtained by measuring three different properties in MR brain imaging. These are, from left to right: PD, T1, and T2 respectively.

cell in the image lattice is called a voxel (or pixel if the image is 2D). In the cases where there is more than one lattice, each image provides a specific type of measurement. For example, in MR brain imaging there are usually three images acquired at different times: T1 and T2 weighted, and proton density PD. This is illustrated in Figure 5.2.

To illustrate how data are organised, assume two equally-sized 3D image lattices M_1 and M_2. The voxels in each of these lattices are accessed via the spatial cartesian coordinates (x, y, z). So, if at voxel coordinates (x_k, y_k, z_k), the intensity as measured on M_1 is m_{1k}, then $m_{1k} = M_1[x_k][y_k][z_k]$.

Voxel Intensities

The simplest way to extract image data into a clustering algorithm is to define the feature-set as the available image measurements. Every spatial location in every image lattice provides a feature element. These feature vectors are then constructed to serve as $\mathcal{X}$, the input data set. For example, we construct data set $\mathcal{X}$ consisting of two

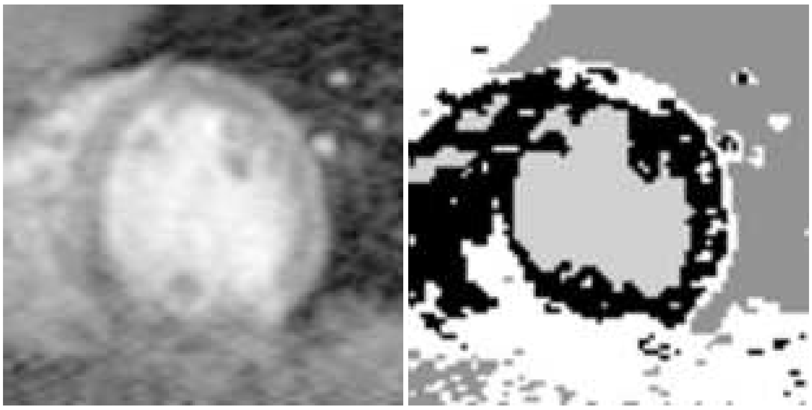


Figure 5.3: An original tissue density MR image is shown on the left, while its PDI-clustered segmented version is shown on the right. ($c = 4$) Only the intensity data was used. The max rule was used for defuzzification.

features that correspond to M_1 and M_2 as follows:

$$\mathbf{x}_k = (m_{1k},\ m_{2k}) \quad \forall k \in \{1, \ldots, N\},$$

where N is the size of either of the image lattices.

The simplicity of this approach and its sometimes quite accurate results are its main strengths. Its most common application is when there are several feature images of the same scene as in MR brain images or CT images [Clark *et al.*, 1994; Clark *et al.*, 1998; Mansfield *et al.*, 1998]. In such cases, the feature set consists of a given voxel's intensity in each image.

Figure 5.3 shows a cardiac MR image of the type we use in our research. Using pixel intensity as the only feature of the data set, a segmentation of the image into four regions using PDI (randomly initialised) is shown. The histogram of the image is shown in Figure 5.4. The placements of the prototypes by PDI is also shown.

By using this feature extraction technique, voxel neighbourhood information is dispensed with and not represented in the feature-set. Two different (spatially-distinct) objects which share the same approximate intensity levels will be clustered into one

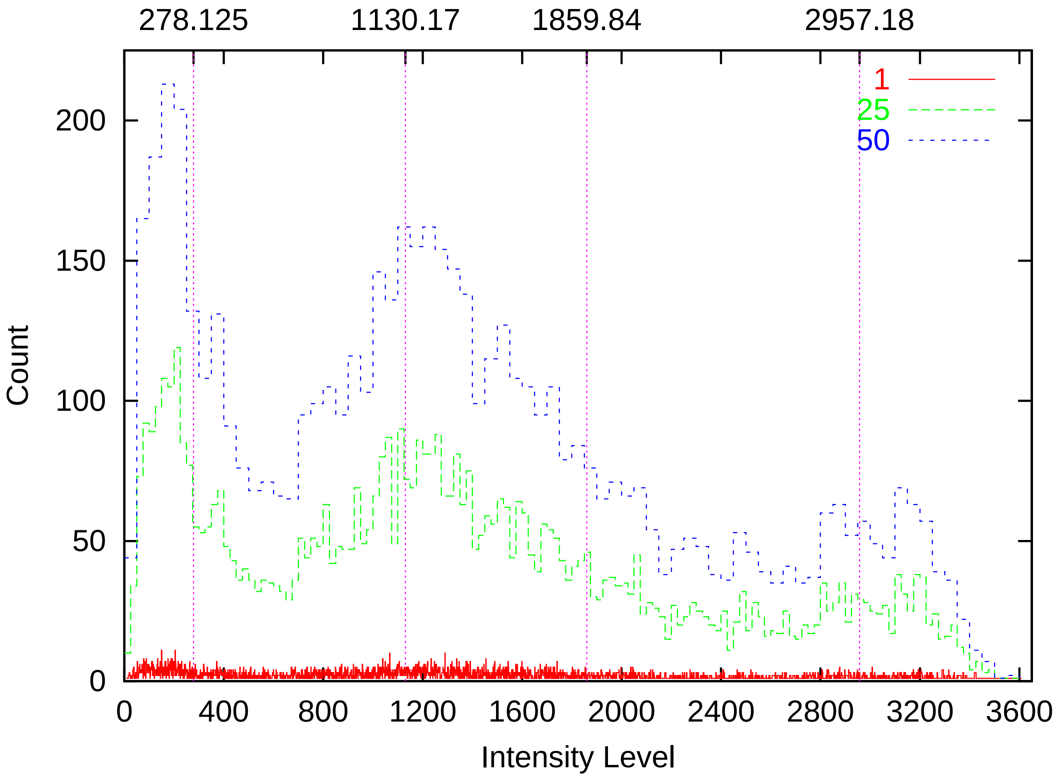


Figure 5.4: The histogram of the MR image of Figure 5.3 for different bin sizes. The vertical lines mark the locations of the found prototypes by PDI.

level. A simple way of addressing this problem is to append to each feature vector in $\mathcal{X}$ (which corresponds to a voxel location) additional features containing that voxel's spatial coordinates.

Spatial Coordinates and Intensities

Clustering voxel intensities only, as described above, does not utilise the proximity relationships that exist between neighbouring voxels. The direct way of taking this into account is to add features for the spatial coordinates of the voxel.

For example, we construct data set $\mathcal{X}$ consisting of five features that correspond to M_1, M_2, and three spatial cartestian coordinates as follows:

$$\mathbf{x}_k = (m_{1k},\ m_{2k},\ x_k, y_k, z_k) \quad \forall k \in \{1, \ldots, N\},$$

where N is the size of either of the image lattices. Note that we may use a different coordinate system, like polar or cylindrical, instead of the cartesian one.

The values of the coordinates can be plotted as an image in their own right. Thus, using the same framework as above, we have the original image lattices plus one or more lattices containing coordinate information. By visualising things in this manner we can see that the data set will contain a lot of regularity. Assuming a 2D image, then we have an $x - y$ coordinate system with a single intensity feature, the data set would be regularised on the grid of $x - y$ coordinates and would look like a 3D rugged terrain. This has influenced the design of special clustering algorithms that have no general utility beyond this type of data, *e.g.*, mountain clustering [Velthuizen *et al.*, 1997].

Intensity and spatial coordinate data will almost certainly not share the same units and range. Thus, it is important to determine the weighting to give to each feature. This is however a largely empirical exercise. In the image of Figure 5.3, the intensity (tissue

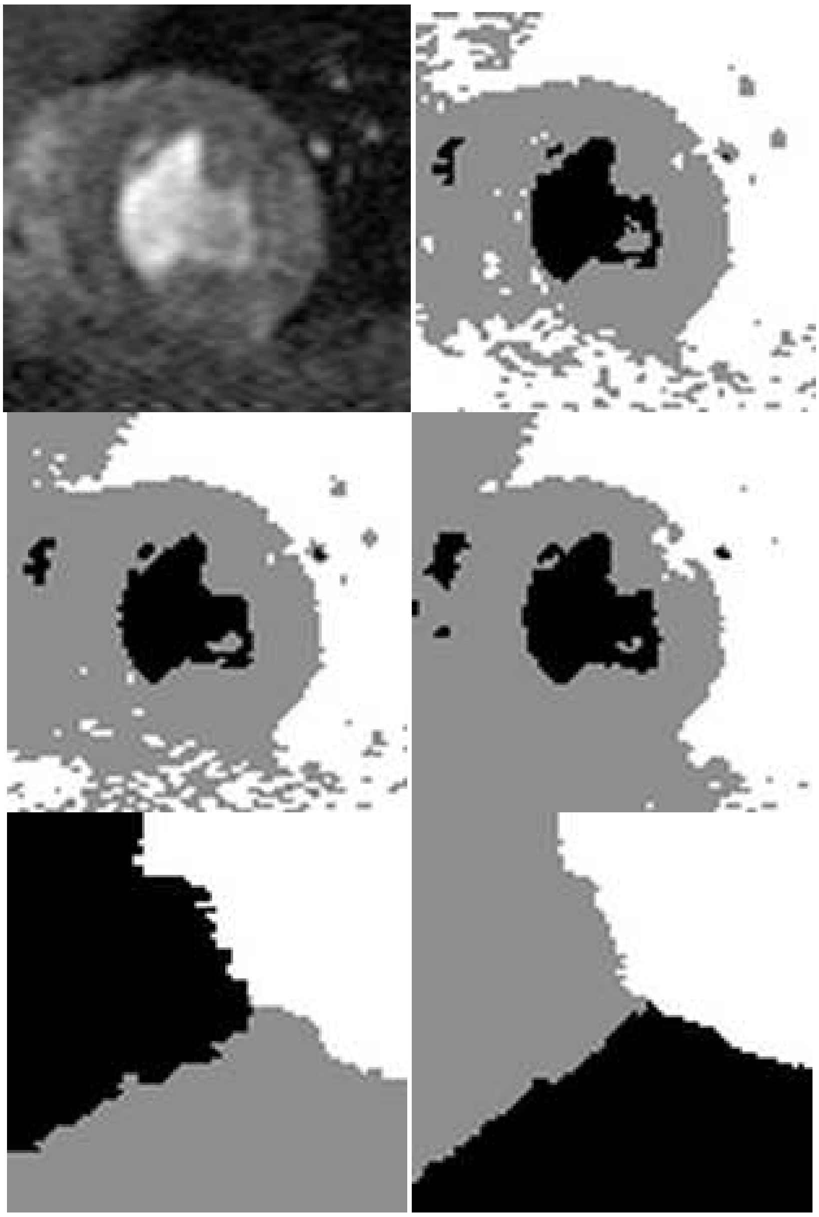


Figure 5.5: An original tissue density MR image is shown on the left, and different FCM-clustered segmented versions are shown on the right. ($c = 3, q = 1.5$) The first segmentation was produced with zero weighting given to the $x - y$ coordinates, then a weighting of 10 was given to x and y, then 20, then 40, and finally a weighting of 60 was used. In the final image the clusters divide the $x - y$ space equally between them.

density) values range from 0 to approximately 3700, while the x and y coordinates range from 0 to 77 only.

One approach to overcome this is to dynamically weight the spatial features and then choose the value of the weight that minimises a suitable clustering validity criterion [Boudraa *et al.*, 1993]. In this case, the usual clustering validity measures may not be suitable to make a judgement which is grounded in physical anatomy. However, they may be useful in guiding the user to choose between different clustering results. But a further problem lies in the fact that the objects in the image may not cluster in shapes recognisable by the algorithm, *e.g.*, spheres or ellipsoids.

Locality Measures

In this feature extraction approach, voxel intensity values are supplemented with other "locality" features. A data point in $\mathcal{X}$ will therefore be composed of the intensity values at the corresponding voxel and other numeric indicators that may be edge- or region-based. These are usually measured over a small window centered around the voxel. The histogram of this window region will have such features as mean, mean square value (average energy), dispersion, skew, and so on. Results from this approach are empirical and vary from one application to another [Tuceryan & Jain, 1993; Marchette *et al.*, 1997].

As we have not conducted much research into this approach we can say that whereas this approach may provide very accurate results, it requires much more experimentation than the above two approaches. There are a lot of studies on novel locality measures, and while these may be effectively applied to images containing textures, most medical imaging modalities produce pictures that may not be aptly described by mixtures of textures.

5.2.2 Post Processing

We now address three ways in which the output of a fuzzy clustering algorithm may be processed for the purposes of obtaining a segmentation. First, the fuzzy membership images provided by the algorithm can be thresholded to obtain crisp, segmented images. Second, the fuzzy membership images can be combined to provide image enhancement or used for segmentation display. Or, a small knowledge-base can used to supplement the fuzzy output of the algorithm.

Crisp Segmentation

From the outset, we should say that obtaining crisp membership values from fuzzy ones involves throwing away information. This is one of the conundrums of fuzzy logic applications. However, the argument of fuzzy logic proponents is: it is better to have more information, which may be pared down at some point, than less information, which may be wrong. Obtaining a fuzzy partition of the image gives us the option of assessing the fuzziness of the solution before applying the “de-fuzzification” step. Furthermore, the fuzzy partition provides more information than a crisp one, in case high-level processing were conducted.

One of the most common ways of obtaining a crisp partition or segmentation is to use the max rule which stipulates that a point be allocated to the cluster with which it has highest membership.

Another common way of obtaining crisp segmentation is by means of identifying the cluster of interest and setting a threshold for its membership values. This is also called obtaining an α-cut of the cluster’s fuzzy set. Determining an optimum value for α, the threshold, remains a largely empirical exercise. (0.75 seems a popular value.)

Both post-processing methods must be addressed carefully especially when a majority of points have nontrivial membership values with more than one cluster. In this

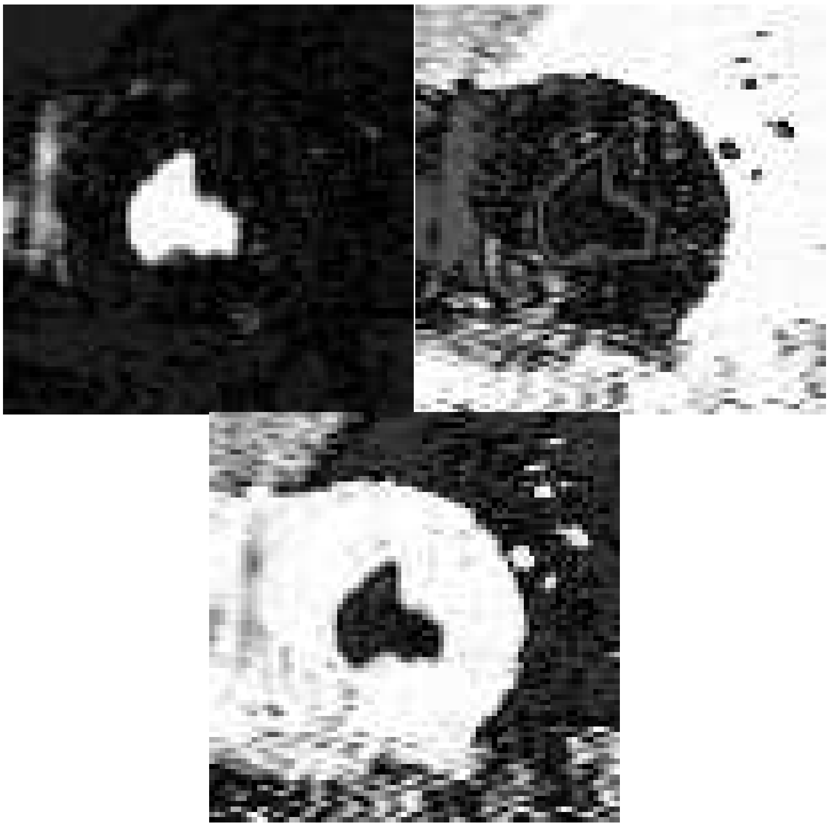


Figure 5.6: Three clusters' membership images. PDI used ($c = 3, m = 2.0$).

case, the solution is very fuzzy and de-fuzzification may lead to tentative (inaccurate) results.

Membership Images and Contrast Enhancement

Provided a cluster of interest is determined, the memberships with that cluster can be plotted as a gray-level image. In such a case, maximum membership, 1, may be shown as white and all other membership values scaled accordingly. Gray-level membership images can provide good enhancement of an object of interest. Like standard contrast enhancement techniques which give a bigger dynamic range to a particular section of the intensity histogram, a fuzzy membership image will emphasise those pixels that

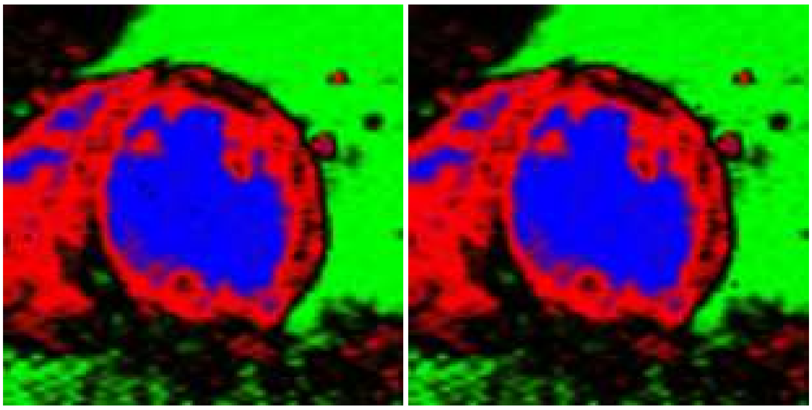


Figure 5.7: The image on the left is a colour-coded segmentation obtained using FCM ($c = 3$), while the image on the right is its PDI counterpart.

most belong to a cluster. This is seen in Figure 5.6.

Also, in cases of a small number of clusters (ideally three or four), the membership values of all clusters can be plotted as a colour image. A colour is selected to represent a cluster and a given membership value is allocated a proportional strength of that colour. The resulting colour image provides at the very least a neat summary of the fuzzy output. This is shown in Figure 5.7 where we show both FCM and PDI's combined membership image using a colour coding. In these images, the pixels are labelled with varying strengths of red, green, blue, depending on their respective cluster memberships. The dark pixels are, therefore, those whose membership values are not strongly in favour of any one cluster.

High-level Rule-based Reasoning

Clustering provides an initial approximation to the real classification of objects in the image. If high accuracy and reliability is required the fuzzy output can then be fed into a high-level reasoning "unit". Often such units are fuzzy rule bases. Depending on the application at hand, the rule base may seek to combine fuzzy regions (clusters) together, determine certain properties of them, or establish spatial relations between them [Rosenfeld, 1984; Krishnapuram & Keller, 1993b; Chi *et al.*, 1996]. Often, this is done with the purpose of designing an automatic classifier; [Clark *et al.*, 1998] provides a good example of this type of work.

5.3 Comparison Between FCM and PDI on Synthetic Medical Images

Having explained how clustering is used in image analysis, in this Section, we provide a comparison between the performance of FCM and PDI on synthetic images that have some similarity to the medical images we used in our research. We first describe our synthetic model, then we present the results of both algorithms.

5.3.1 Synthetic Model

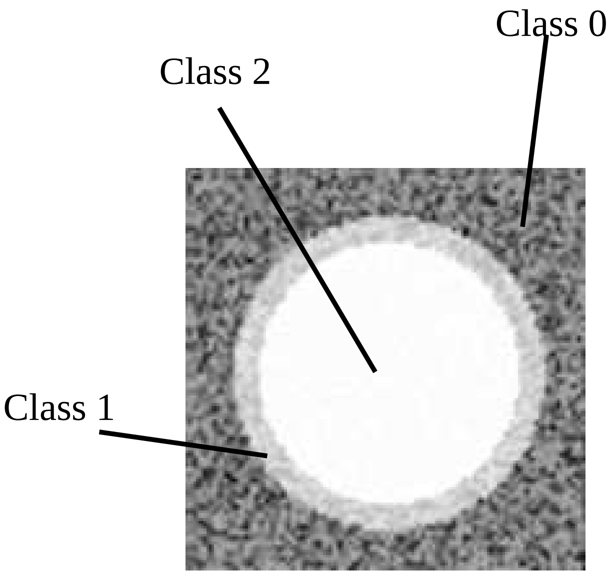


Figure 5.8: A synthetic image with $w = 5$. Class 0 is the background, class 1 is the shell, and class 2 is the inside of the shell.

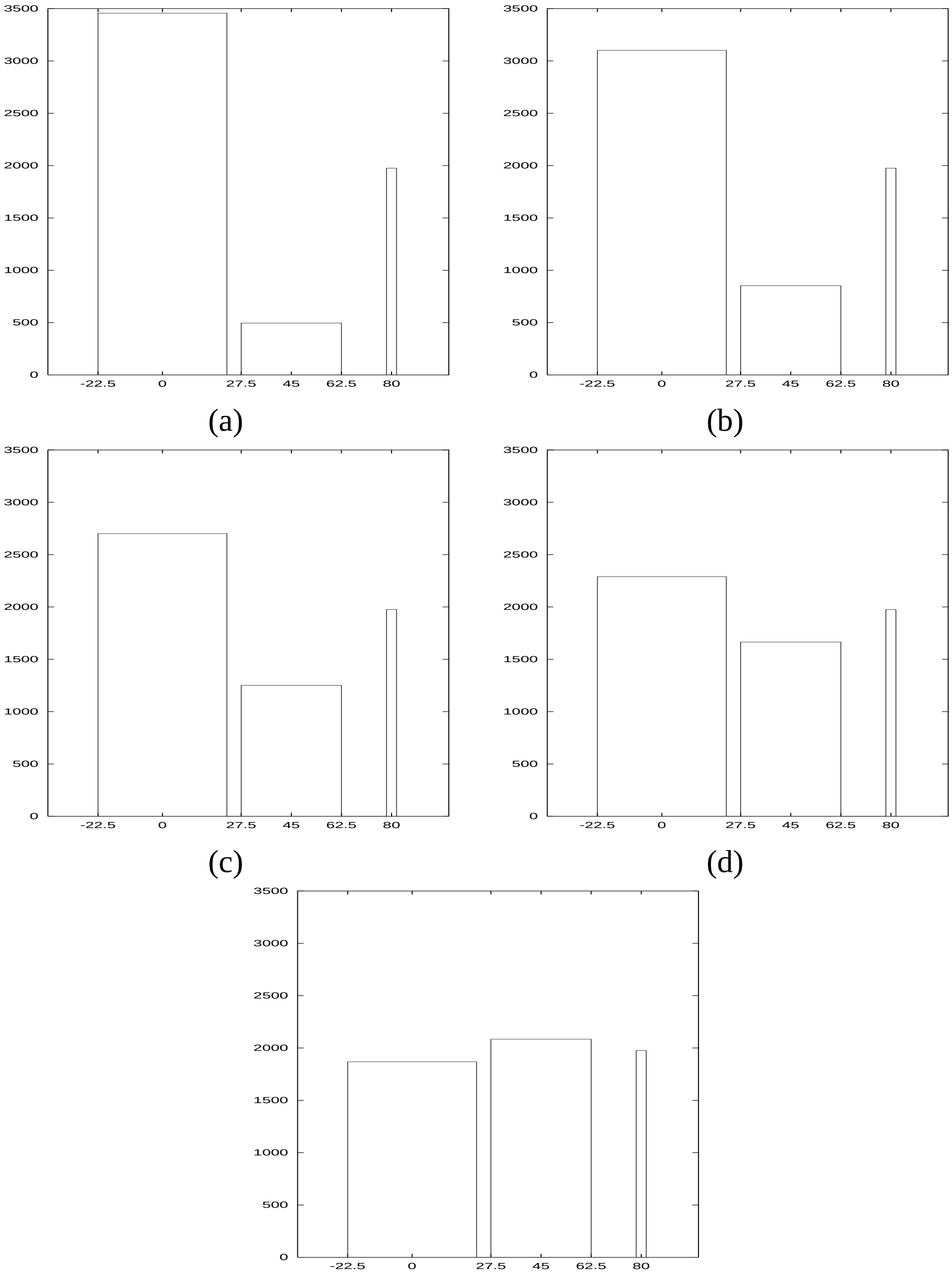
The images were designed to be 77×77 with a structure resembling the one we have in our medical images. This consists of three objects: a background, a circular shell and the inside of a shell. The classes of the objects were chosen so that: class 0 stands for the background, class 1 for the shell, and class 2 for the inside of the shell.

The shell was given a width, w, which we varied in our experiments. Figure 5.8 shows an example of one such image with $w = 5$.

The three classes consisted of pixel intensities described by uniform distributions. The parameters of Class 0 (background) were: a, representing the average intensity level, and α, representing the width of the distribution. Class 1 (shell) was given average intensity b and width β. Class 2 (inside) was given average intensity c and width γ.

Our methodology will now be to vary w and see its effect on the quality of both FCM and PDI's clustering. We measure the quality by counting the number of misclassified pixels.

In all our experiments below, we use $m = 2$, and for PDI $r = 1.5$. These values were selected in accordance with our experiences from the previous Chapter. We chose the values 0, 45, and 80 for a, b, and c respectively, and the values 45, 35, and 4 for α, β, and γ respectively. These values were arbitrary but selected to test the familiar problem of close clusters of different sizes (Classes 0 and 1) but this time there is a third cluster present (Class 2). Class 2 is a relatively compact and well-sepearated cluster in comparison to the other two. This is evident in Figure 5.9 which shows the histogram distributions of the synthetic images corresponding to $w = 3, 5, 7, 9$, and 11 respectively.



(a)

(b)

(c)

(d)

(e)

Figure 5.9: Plots (a), (b), (c), (d), and (e) are the histogram distributions of the synthetic images corresponding to w=3,5,7,9, and 11 respectively. The columns in each plot correspond, from left to right, to classes 0, 1, and 2 respectively (background, shell, and inside of the shell). The height of a column depicts the number of pixels in the class it represents. The width of a column depicts the intensity distribution of the class. The background and shell contain a varying number of pixels according to w and have a wide almost-touching range, but the inside of the shell has a narrow range.

Width	% misclassified pixels			
w	FCM		PDI	
3	24.57	(1547 pixels)	2.44	(145 pixels)
5	19.90	(1180 pixels)	2.46	(146 pixels)
7	6.04	(358 pixels)	0.89	(35 pixels)
9	3.88	(230 pixels)	0	
11	2.41	(143 pixels)	1.62	(64 pixels)

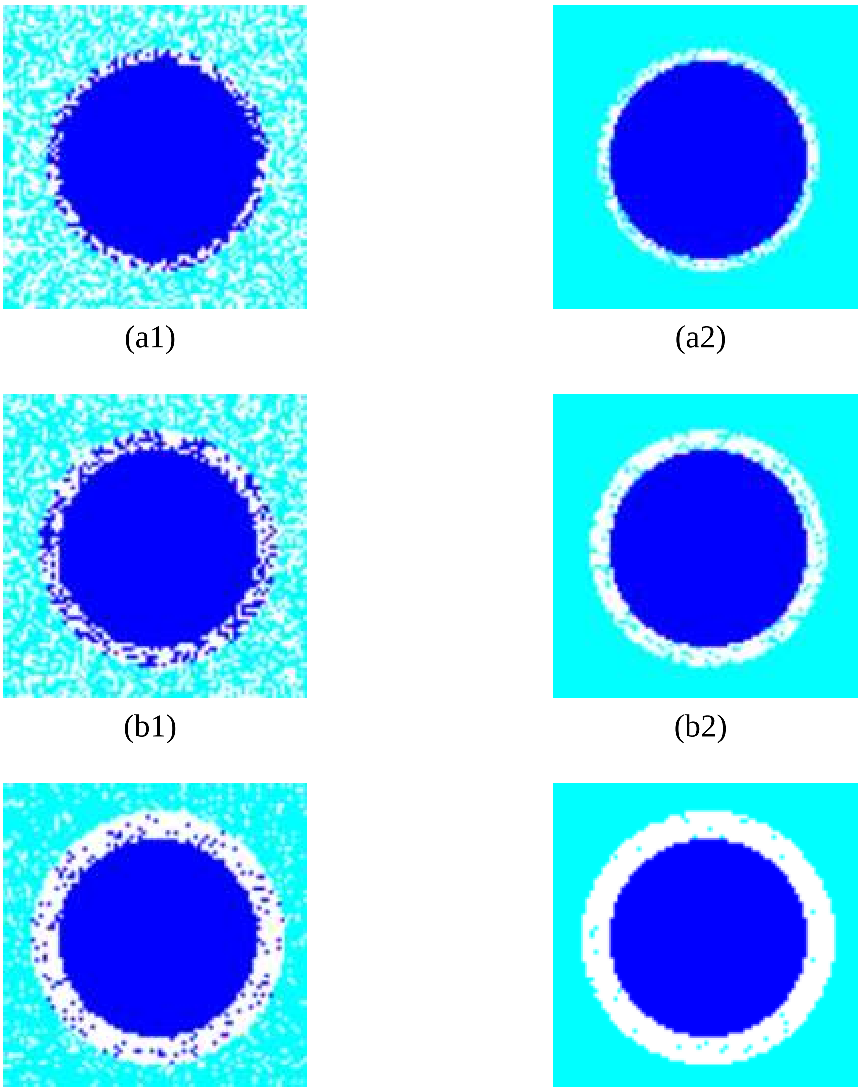
Table 5.1: Comparison of accuracy of FCM vs. PDI in classification of the synthetic images.

5.3.2 Results

PDI's segmentation results were a great improvement over FCM's. This is confirmed by Table 5.1 which is a comparison between FCM and PDI in terms of classification accuracy. The visual segmentation results obtained for both FCM and PDI are shown in Figures 5.10 and 5.11. We observe that FCM performs rather badly at smaller values of w.

For example, at $w = 3$, where class 0 is seven times the population of class 1, FCM splits class 0 into two (bahaviour seen in Chapter 4) and therefore misclassifies large chunks of it. Class 1 thus is divided between class 0 and class 2. This is so even though class 2 is very focussed in terms of intensity range (see Figure 5.9). PDI does not have any problems in identifying class 2 accurately. However, PDI does fail to classify correctly a small number of pixels belonging to class 1 and assigns them instead to class 0. Those mis-classified pixels have an intensity level close to class 0's range.

At $w = 7$, most of the pixels in class 1 are correctly classified by PDI and near-perfect results are attained at $w = 9$. FCM continues to struggle. We note that whereas PDI misclassifies a small section of class 1's pixels at smaller values of w, by $w = 11$ (where class 1 is now more populous than class 0), it extends class 1 to cover some of the noiser points in class 0.

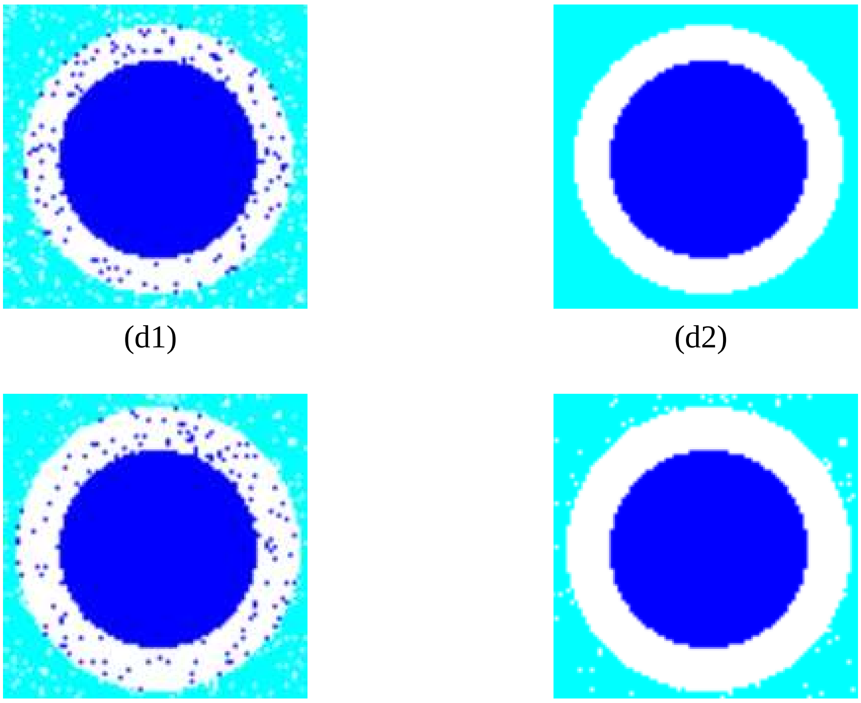


(a1) (a2)

(b1) (b2)

(c1) (c2)

Figure 5.10: The left side column shows FCM results and the right side column shows PDI results. The top row shows results for $w = 3$, next is $w = 5$, and bottom-most is $w = 7$.



(d1)

(d2)

(e1)

(e2)

Figure 5.11: The left side column shows FCM results and the right side column shows PDI results. The top row shows results for $w = 9$ and the bottom row is for $w = 11$.

This study shows how the effects of the population and diameter of a cluster affect clustering algorithms' performance. FCM would have coped well with this problem if there was large separation between the intensity ranges of each class. PDI performs much better at this type of problems because of the inequity between cluster sizes and populations.

5.4 Conclusions

This Chapter provided a summary of our experience with clustering images for the purpose of segmentation. We have divided the segmentation-by-clustering process into three decision phases: feature extraction, clustering, and post-processing. Within the clustering phase itself there are also decisions to be made about algorithm and distance metric. We also demonstrated the advantage of PDI over FCM for some synthetic images. Furthermore, we briefly reviewed the image clustering literature.

Since most clustering algorithms suffer from shortcomings that may affect accuracy, it is essential for the user to be aware of the shortcomings of their preferred algorithm. Some segmentations are impossible to produce using clustering, unless the right features are extracted to act as input to the clustering algorithm. Thus, empirical feature extraction plays an important role as will be seen in the next Chapter.

CHAPTER 6

Application to Medical Image Analysis

This Chapter presents the results of our published work on using fuzzy clustering in a cardiac imaging application. The aim was to segment and track the volume of the left ventricle during a complete cardiac cycle. The images used are MR images containing tissue density and velocity data. Since there is no other published work on analysing this type of image using fuzzy clustering, our application is a novel one. Our results may be viewed to be an investigation into the feasibility of this type of research.

The Chapter proceeds as follows. Section 6.1 presents a brief review of the anatomy and physiology of the cardiovascular system. Section 6.2 describes the type of velocity (or flow) images we used in this research. Section 6.3 gives the specifics of our application and Section 6.4 describes our results in full. The research presented here uses PDI for clustering.

6.1 The Cardiovascular System

For a detailed introduction to cardiovascular anatomy and physiology see [Davson & Segal, 1975; Wilson, 1990], and for a more detailed review of the imaging of the cardiovascular system see [Underwood & Firmin, 1991; van der Wall & de Ross, 1991; Pettigrew *et al.*, 1999].

The Heart is a Pump

The cardiovascular system is responsible for blood circulation in the human body. It supplies blood to cells throughout the body. Blood acts as a transport medium, where it transports oxygen from the lungs to the cells and carbon dioxide from the cells back to the lungs. This circulation of the blood is achieved by a pump — the heart — which forces the blood through elastic tubes — the blood vessels.

Blood Vessels

The main function of the blood vessels is to carry the blood throughout the body. If the blood flows away from the heart the blood vessels are called arteries. If the blood flows to the heart the blood vessels are called veins. The largest artery is the aorta which is characterised by a number of bifurcations. A third type of blood vessels called capillaries connect the arteries to veins.

Heart Structure

A schematic diagram of the heart is shown in Figure 6.1. The heart consists of two pairs of chambers: the left and right ventricles and the left right atria. The ventricles act as pumps while the atria act as reservoirs. Blood enters the heart from its long journey around the body through the superior and inferior vena cava into the right atrium. This blood has very little if any oxygen. Then it passes by the tricuspid valve into the right

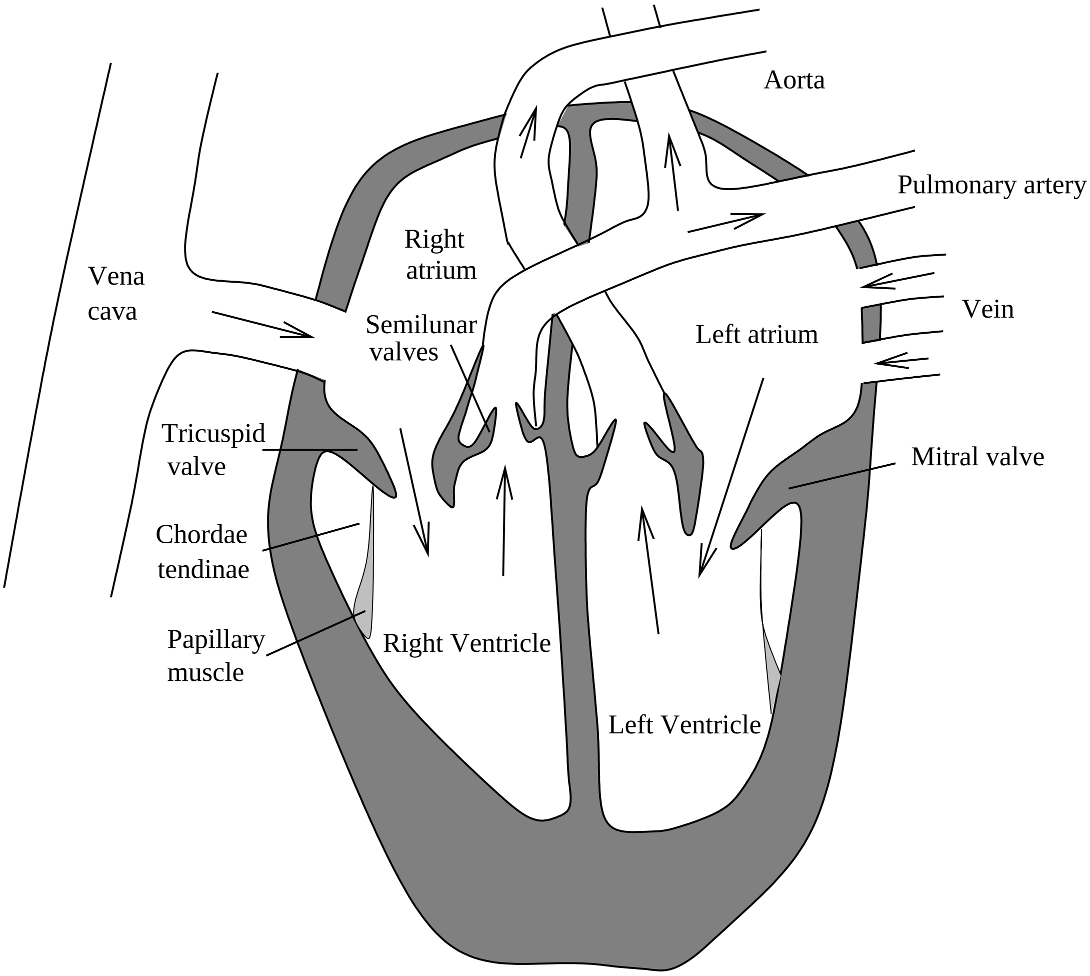


Figure 6.1: A simplified diagram of the heart.

ventricle. After the right ventricle contracts, the blood is forced through the pulmonary semilunar valve and into the pulmonary artery. The pulmonary artery splits into the right and left pulmonary artery where the still oxygen-deficient blood travels through the lungs. The blood becomes enriched with oxygen and travels back toward the heart. The blood enters the heart via the right and left pulmonary vein which come directly from the lungs. The blood then enters the left atrium. The bicuspid valve opens up and the blood falls into the left ventricle. The ventricle contracts and the blood goes rushing passed the aortic semilunar valve and into the aorta which is the largest artery in the body. Now the blood is on its way back to the body.

The Myocardium and Systole and Diastole

The walls of the ventricles are composed of muscular tissue and form what is known as the myocardium. During the cardiac cycle, the myocardium contracts, pumping blood out of the ventricular chambers and through the semilunar valves. The myocardium's inner surface is called endocardium while the outer surface is called epicardium.

In normal conditions the human heart beats between 65 and 75 times per minute. Each heart beat corresponds to an entire cardiac cycle which can be characterised by a contraction phase (systole) and a relaxation phase (diastole) of the atria and ventricles. The systole can be divided into two phases. In the first phase the atrioventricular valves close, the ventricular muscle starts to contract, and the ventricular pressure increases due to the closed artery valves. At this stage the volume does not change and the phase is referred to as iso-volumetric contraction. In normal conditions this phase lasts for 60ms. In the second phase, the artery valves open due to the increased pressure, the ventricular muscles contract and the ejection starts. Normally, the left ventricle ejects only half of its volume of ca. 130ml as stroke volume into the aorta. At the end of this phase a rest volume of ca. 70 ml remains in the ventricle, and the arteries valves close.

Similarly to systole, diastole can also be divided into two phases. During the first phase of the relaxation all valves are closed and the relaxation is iso-volumetric. The ventricular pressure drops rapidly. During the second phase the valves separating atria and ventricles open and the ventricles are filled first rapidly and then more slowly. The ventricular pressure increases slightly. Then the cardiac cycle starts again.

Quantitative Measurements

There are a number of quantitative measurements which can provide valuable clinical information for the assessment of the heart [Mohiaddin & Longmore, 1993]. Mycoradial functionality can be assessed by measuring the ventricular volume, the stroke volume and the rest volume. Based on these quantities it is possible to calculate the ejection fraction of the ventricles which measures the ratio between stroke and rest volume. Other indicators of myocardial functionality are the muscle thickness and mass as well as wall motion and thickening during the cardiac cycle. Arterial functionality can be assessed by measuring the distensibility or elasticity of arteries in terms of *compliance* and is defined as change in volume per change in pressure during the cardiac cycle.

6.2 MR imaging and Velocity Quantification

Magnetic Resonance images picture anatomic detail by measuring tissue density in the plane of imaging. Every pixel in an MR image carries a value that is proportional to the average tissue density registered by the MR scanner at the corresponding approximate location in the plane of imaging.

The magnetic resonance signals are caused by Hydrogen nuclei present in the tissue. The nuclei spin on their axes generating magnetic moments making them become magnetic dipoles. When these nuclei are placed in the magnetic field of the scanner, the axes of spin precess about the direction of the applied magnetic field. The frequency of precession is directly proportional to the strength of the magnetic field each nucleus experiences.

Flow velocity quantification [Rueckert, 1997; Yang, 1998] is based on the observation that as spins move along an imaging magnetic field gradient, they acquire a shift in their angular position relative to those spins that are stationary. This is called a spin phase shift, and it is proportional to the velocity with which a spin moves. This shift in the phase angle of the spins is a parameter contained within the detected MR signal and can be readily measured.

The composite MR signal provides two images. The first one is the conventional image, called the modulus of the magnitude image, in which the image signal intensity is simply related to the magnitude of the MR signal. The second image is the phase image in which the signal intensity is proportional to the shift in spin phase relative to the stationary spins. This phase image, therefore, provides a pixel-by-pixel mapping of spin velocities, given that both the strength of the magnetic field gradient and the time during which the spins are exposed to the gradient are known. Since these features of the sequence can be explicitly determined, it is possible for the user to define a desired amount of spin phase shift per unit velocity and consequently determine flow rates from the phase image.

To display flow in two opposite directions, a gray scale for displaying the spin phases is chosen so that zero phase shift is medium gray. Spins that move into the scanner will typically acquire positive phase shifts of 0 to 180 degrees. These are assigned a proportional intensity from midgray to white. Spins that move in the opposite direction will acquire negative phase shifts of 0 to 180 degrees. These are assigned a proportional intensity from medium gray to black. This is similar to color Doppler echocardiography, in which the flow toward and away from the transducer is displayed with two different colors, red and blue.

6.3 Novel Application to Velocity Images

We now detail the results of our work [Shihab & Burger, 1998a; Shihab & Burger, 1998b] using cardiac velocity MR images. We describe the feature extraction, clustering, and post-processing decisions we made in this specific application. Our application consists of analysing MR image cine sequences acquired at the mid-ventricular plane of the heart. The images are conventional MR tissue density images as well as velocity images. Our objective is to segment and track the Left Ventricle (LV).

The cine sequences of images are aligned with the short-axis of the left ventricle

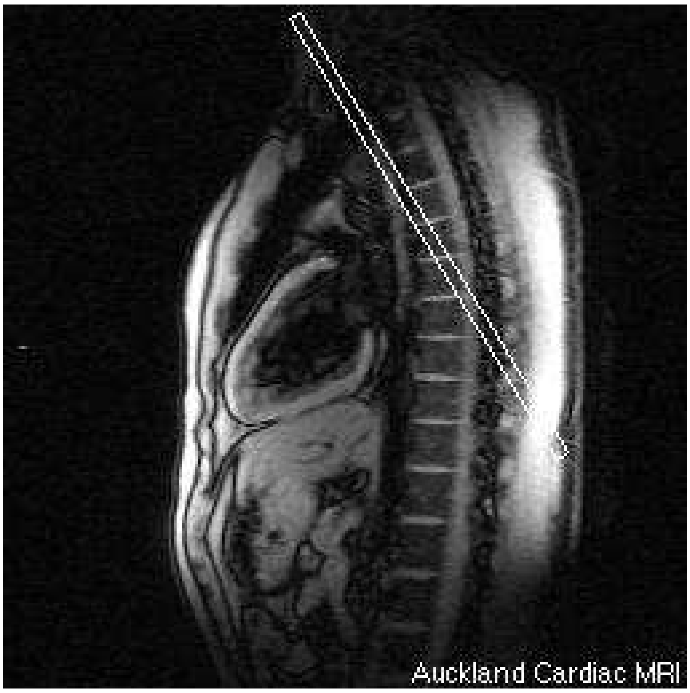


Figure 6.2: A plane of imaging that provides a short-axis view of the heart would be parallel to the plane shown. ©Auckland Cardiac MRI Unit

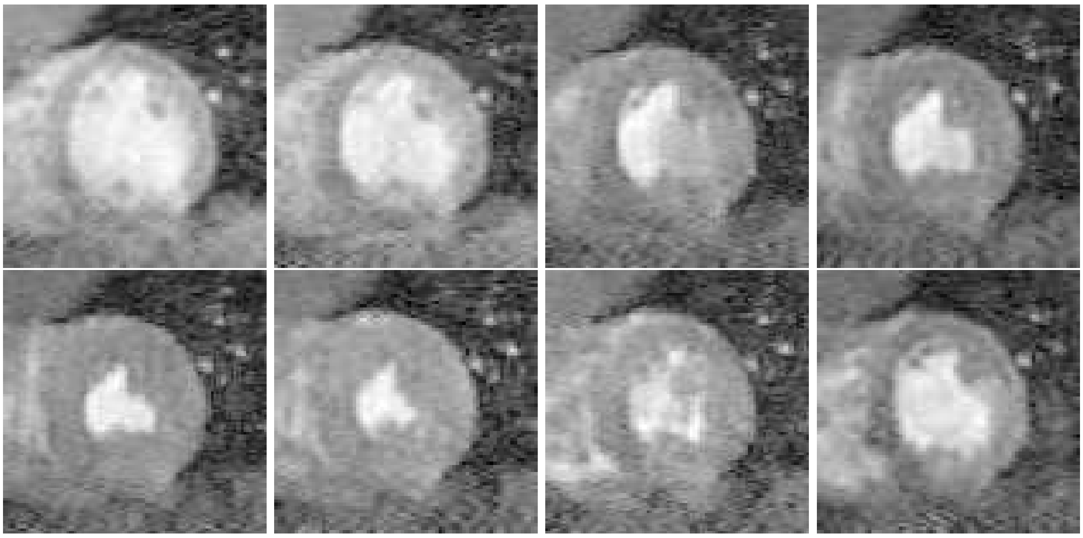
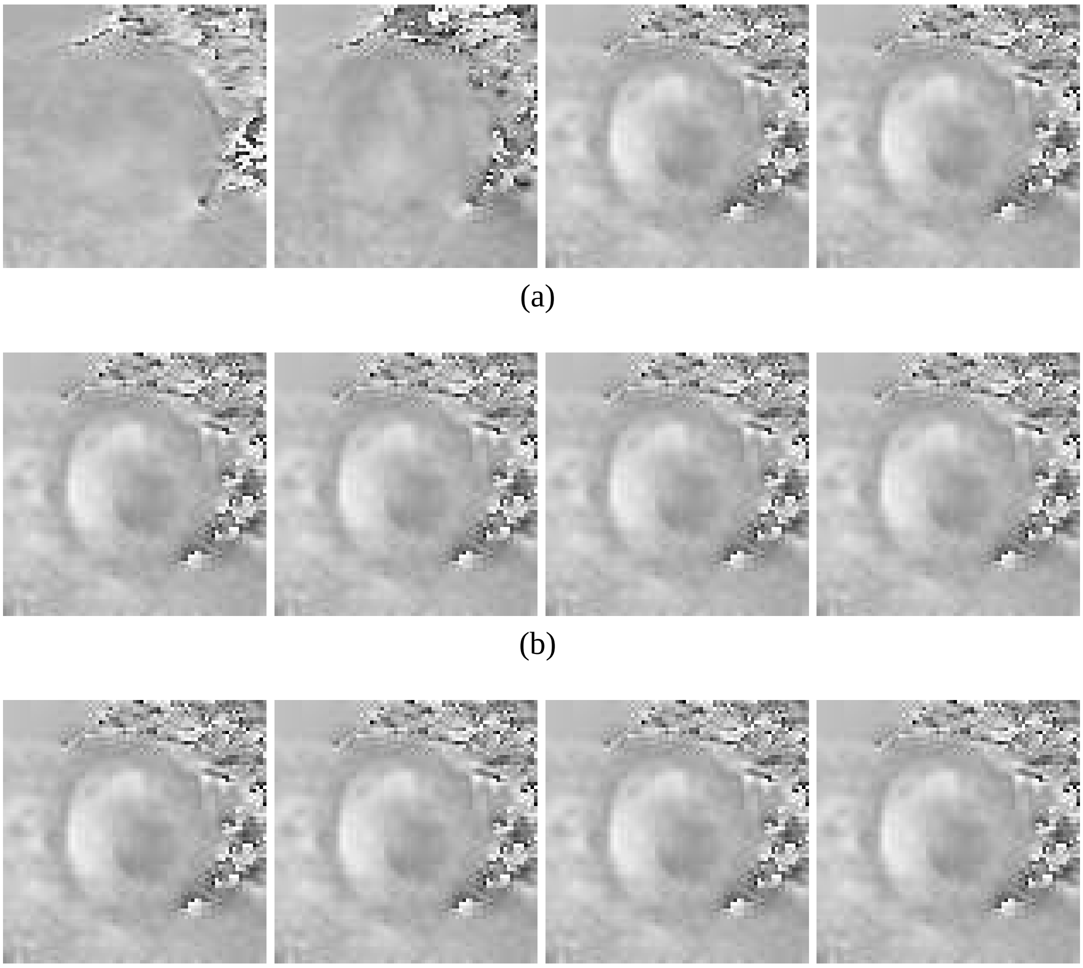


Figure 6.3: Examples of tissue density images: frames 0, 2, 4, 6, 8, 10, 12, and 14 in an image sequence.

(illustrated in Figure 6.2). The velocity data is rendered as 3 images, v_x, v_y and v_z, corresponding to the cartesian components of the velocity vector field V at each pixel. The reference coordinate system has the x-y plane lying on the plane of imaging (aligned with the short-axis of the left ventricle) and the z axis perpendicular to it (aligned with the LV long-axis).

The image sequences contain 16 frames. The sequences start at systole and end at early diastole. The time space between each frame and the next is approximately 40 ms. Figure 6.3 displays example frames from a sequence. Figure 6.4 displays four frames from each of the three velocity components. We remark that each image is generated out of normally 256 heartbeats and therefore each image depicts the average behaviour of the heart during a large number of heartbeats. However, the information provided is useful for observing the global dynamics of the heart and we can still refer in a meaningful manner to a particular time of the cine sequence since it belongs to a definite phase of the cardiac cycle.



(a)

(b)

(c)

Figure 6.4: Examples of the velocity images, frames 0, 4, 8, and, 12 of v_x, v_y, and v_z, from top to bottom respectively.

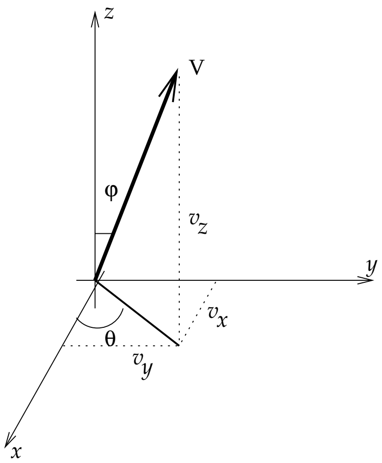


Figure 6.5: θ and ϕ define the direction of the velocity vector at a given point.

6.3.1 Feature Extraction

Each frame in a cine sequence contains several types of data. It contains the tissue density data: I and the velocity data: v_x, v_y, and v_z. Further, we can use the x, y spatial coordinates for each pixel, assuming a cartesian coordinate system or the r and θ coordinates, assuming a polar coordinate system. The cartesian velocity data can also be transformed to spherical or cylindrical data values. Thus, with very little pre-processing, many possible features can be selected for each pixel.

In all our experiments, we used the two cartesian spatial coordinates, x and y, as features. However, we did not enter into the issue of finding suitable weighting for the spatial features. As their range is much smaller than that of either the tissue density or velocity data, they had little effect on the results. However, we left them in since they are useful in the post-processing stage.

We assessed the impact of velocity features by clustering first without them, and then with combinations of them. The features for the first experiment consisted of x, y, and I (tissue density data without V). In the second experiment we added V which is the magnitude of the three velocity components at each pixel: v_x, v_y, and v_z

($V = \sqrt{{v_x}^2 + {v_y}^2 + {v_z}^2}$). In the third experiment, we removed V and replaced it with θ and ϕ. These angles describe the direction of the velocity field at a given pixel, as shown in Figure 6.5.

6.3.2 Method

In all experiments we ran the PDI algorithm. The m fuzziness factor was set at 1.5, and the r the normalisers' exponent was fixed at 1.0. Also, c was set to four, as this gave the most intuitive segmentation of the images. As is known, PDI's output is in the form of cluster prototypes, membership matrix, and normalisers. In the results we present here, we utilised the membership matrix.

For each data set belonging to a frame after the first one, we initialised PDI with the found prototype locations of the previous frame. The first frame's data was randomly initialised. An entire patient sequence would take between 3—4 minutes on a recent Pentium PC model.

6.3.3 Post-Processing

Having clustered a patient's data (in the three ways stated above), we then selected the cluster corresponding to the LV blood pool area. This could be effected in two ways: the first is to estimate which of the found prototypes represents the LV, or to plot a max-rule segmentation of the first frame, from which one can visually determine the LV-cluster. Membership images of the LV-cluster for the two cases of *without-V* and *with-V* are shown for a normal patient in Figures 6.6 and 6.7.

Once we have determined the LV cluster, we can now count the pixels in the LV area. Using the x and y features of the LV cluster's prototype as a seed, we ran a region growing routine on the max-rule segmented images. These provided us with a count of the pixels in the LV area for each of the chosen data sets, for each patient.

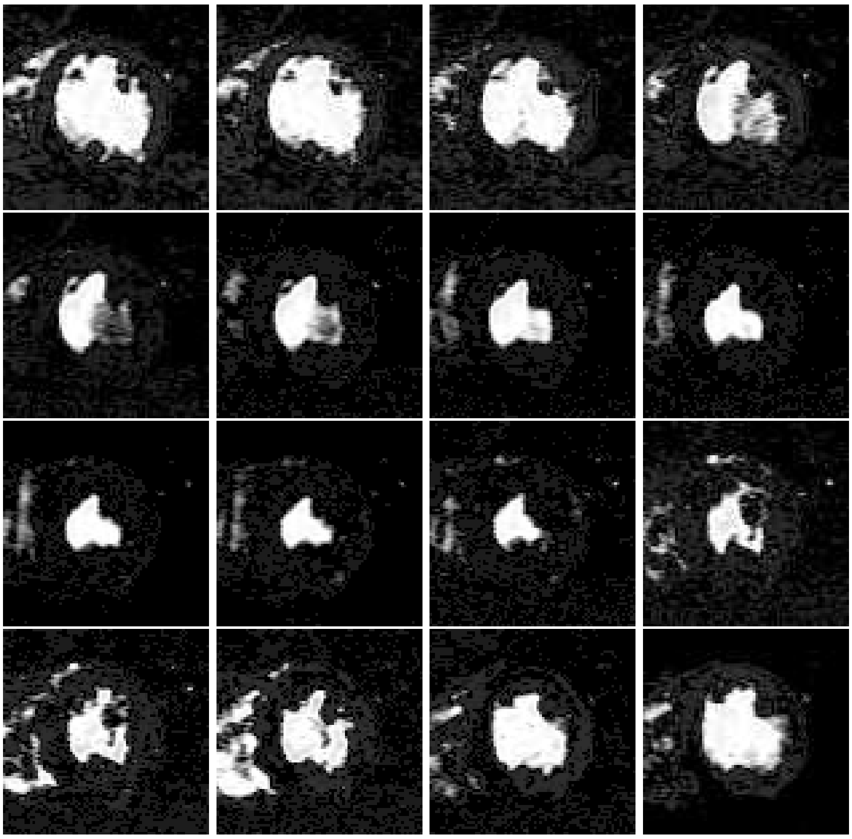


Figure 6.6: First experiment (only tissue density data): membership images of the LV cluster tracked from frames 0 to 15 (left-to-right, top-to-bottom) for a normal patient.

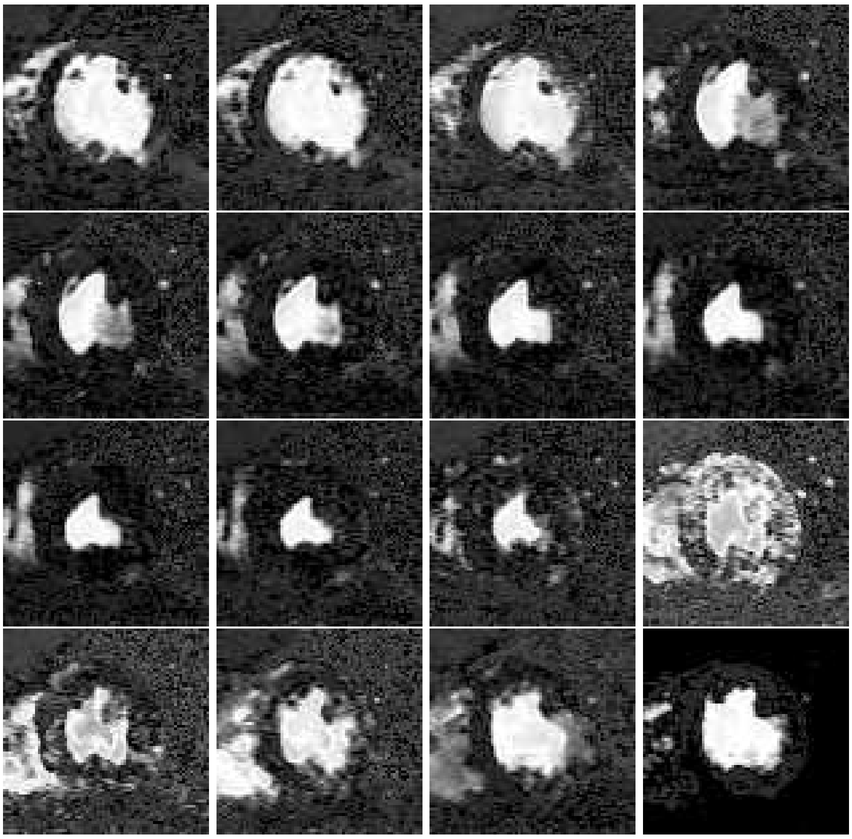


Figure 6.7: Second experiment (tissue density and V data): membership images of the LV cluster tracked from frames 0 to 15 (left-to-right, top-to-bottom) for the same patient as in Figure 6.6.

6.3.4 Results

We remark here that we faced difficulties in our investigations due to the unreliable data values sometimes produced in phase-contrast MRI studies, and due to the length of time required for a single patient study (to collect this data). Thus, we clarify that our intention is to illustrate the application of fuzzy clustering to this type of studies, instead of to present a complete, validated medical investigation.

In Figure 6.8, we compare the calculated areas of the left-ventricle using the three routes we took with a ’ground truth’ established by a clinician. The cine sequence is that of a normal patient.

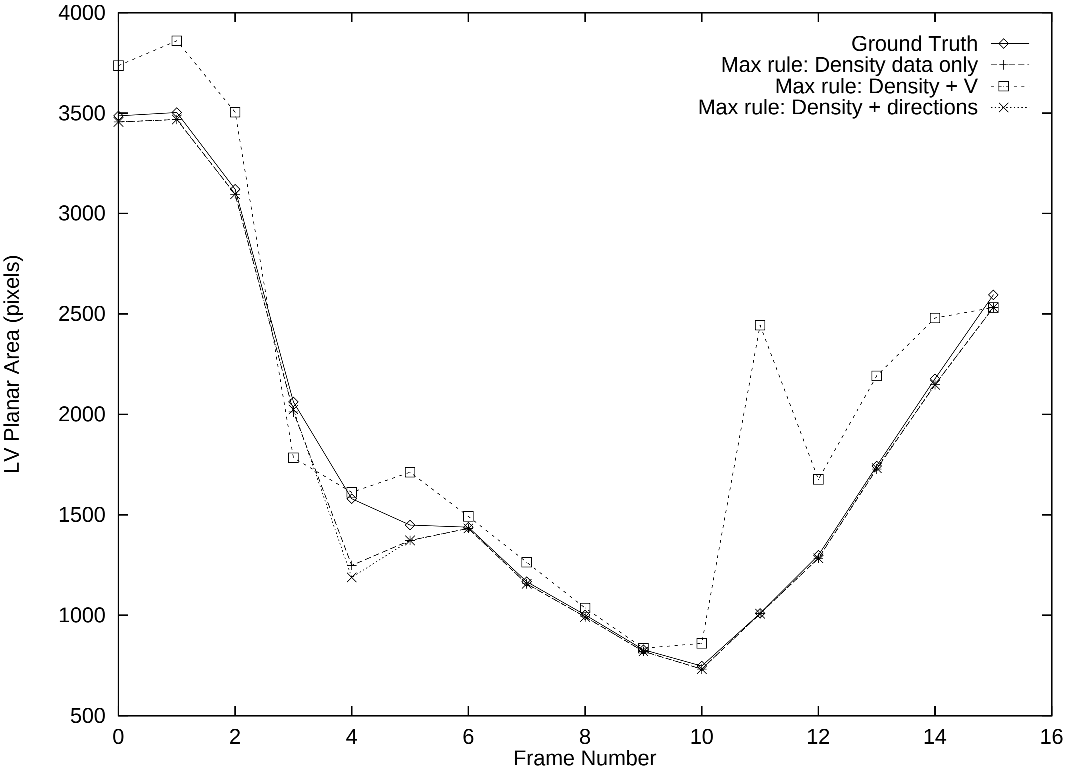


Figure 6.8: Comparison of calculated LV area for the three data sets used.

The general trend of all the curves as compared to the ground truth is correct. However, we observe that using the velocity-magnitude feature causes somewhat erratic estimates of LV area. Furthermore, these estimates were generally greater than the correct values. In general, it was difficult to distinguish between the results of density-only and density-and-velocity-direction features. As can be seen in the plot, the estimates using these two feature sets were consistently less than the correct values.

6.4 Conclusions

In this Chapter, we studied the cardiac system and then investigated the viability of using fuzzy clustering as the principal method for segmentation and tracking of the LV. We proceeded along the same steps outlined in the previous Chapter: feature extraction, clustering, and post-processing. In the feature extraction step, we experimented with novel feature sets that include velocity data made available through phase contrast MR. In the clustering step, we used our novel PDI clustering algorithm. In the post-processing step, we took a conventional route and used the max rule.

We conclude by reviewing our experience. First, our results were generally accurate and can be used for quantifying cardiac measures. Clinicians easily understood the concept of clustering and immediately grasped its application. The strength of the method lies in its general flexibility and accuracy. Decisions like: setting a value for c, fixing values for the clustering parameters, and identifying the cluster of interest, allow flexibility for the user. Once these decisions are gone through for one patient, the processing of the other data sets can be automated.

Second, in studying the effect of using extra velocity-related features, we found that they enhanced accuracy for only one frame out of the 16, as compared to a conventional feature set containing tissue density data. We also found that velocity-directional features provided more accurate results than velocity-magnitude features.

An interesting problem which should be a fruitful line of future research is to track the myocardium in the same image sets (containing velocity data). This would probably necessitate using polar coordinates instead of cartesian ones and weighting the spatial coordinates suitably. Including velocity features would probably increase the extent of accurate segmentation because of the relative lack of motion of the myocardium.

CHAPTER 7

Conclusions and Further Work

7.1 Summary of Main Results

This dissertation investigated the FCM algorithm and devised a new algorithm, PDI, to address a behavioural shortcoming of FCM. The shortcoming is that FCM does not classify accurately a data set containing small clusters lying close to big clusters. We found the reason for this to be that the objective function which is at the heart of FCM becomes inadequate in situations like those stated above. It does not have the flexibility of narrowing or widening the scope of a cluster prototype. By *scope of a cluster prototype* we mean an area around the prototype in which points would add little cost to the objective function. If the objective function allows a given prototype to possess a relatively wider scope than other prototypes, points that lie far from the given prototype, but within its scope, would not be costly. FCM's objective function gives an equal amount of scope to each prototype and this causes the correct solution to be costly when clusters are of unequal sizes, the situation is made worse if the clusters are of unequal populations as well.

To overcome this shortcoming, we devised a modification of FCM. The new PDI objective function attempts to equalise cluster contributions and by doing so, it allows the smaller clusters to be found. For each prototype, PDI redefines its “cluster contribution” to be the same as FCM’s but divides it by a variable, the cluster normaliser. This normalisation creates a non-equal distribution of scopes for each prototype. Thus, small clusters are granted small scopes, because they take small normaliser values, and they therefore become less costly and have a higher chance of being found.

We demonstrated FCM’s shortcoming through systematic study. We formulated a framework and generated dot patterns to specifically test this shortcoming. We also showed, using the same data, that PDI improves quite a lot on FCM’s performance. Furthermore, we investigated some aspects of PDI’s behaviour.

This dissertation also critically investigated the process of analysing image data by using fuzzy clustering. We highlighted three decisions points in this process: feature extraction, algorithm and parameters, and post-processing method. We described examples of each of these decision points. Furthermore, we compared FCM’s and PDI’s clustering of medical MR images, and designed synthetic data to test this.

Finally, the thesis presented the results of a novel application of fuzzy clustering in medical image analysis. We used velocity data obtained by using a phase-sensitive MR technique, as well as the usual tissue density data, to track the left ventricle in image cine sequences. We found the availability of velocity directional data increases the accuracy of the overall clustering.

7.2 Further Research

1. **Further Analysis of PDI**

   (a) PDI requires some investigation from an optimisation perspective. This may be achieved using some of the global optimisation software libraries. An assessment may then be made as to how prone the model is to local solutions. The model itself may require improvements, as it is sensitive to initialisation and prone to divergence. If the iterative implementation is used, divergence may be studied by means of tracking the values of the normalisers, ρ_i, and re-starting the algorithm when a value becomes too low (indicating divergence).

(b) We have not carried out a computational complexity analysis of PDI as compared to FCM, in terms of an iterative implementation scheme. Such an analysis may be useful in finding ways to optimise PDI's computational efficiency. This would naturally enhance the feasibility of using PDI for the analysis of very large data sets.

(c) In Chapter 4, we structured our data sets in two-dimensional feature space. While all indications are that PDI will continue to perform more accurately than FCM for higher dimensional spaces, it would be useful to quantify the limit at which PDI no longer provides a substantial advantage over FCM.

(d) In our experiments in Chapter 4, we only tested PDI's performance on two-cluster data sets. In Chapter 5, we compared PDI and FCM's performance on images containing three clusters. While both experiments showed that PDI improves over FCM's accuracy, will PDI's performance decrease with increasing numbers of clusters?

(e) It might be useful to extend PDI in some of the ways FCM was extended. So, for example, how would a PDI-G-K algorithm (see Section 3.3.1) differ from the plain G-K algorithm? Likewise, we can create a possibilistic (see Section 3.4.1) version of PDI and compare its performance to the original.

2. **Cardiac Medical Image Analysis**

   The points we propose below are independent of clustering algorithm used, except when mentioned.

   (a) In Chapter 6, we only clustered the image data available in one cross-sectional slice. Even though there is no spatial continuity in multi-slice volume data, investigating clustering the entire volume data in 3D spatial space may provide a good challenge. The question of weighting the spatial features appropriately will come up. Excluding spatial coordinates from the data set may well turn out to be an effective approach to initialise a more precise clustering process operating on each slice.

(b) Similarly to above, investigating clustering the volume data with time periodic information summarised in a phase angle feature per voxel (an extension of the approach in [Boudraa *et al.*, 1993]) may yield an improvement to the accuracy of results obtained via point (a) above.

(c) In Chapter 5, we mentioned in passing using clustering for image contrast enhancement. This can be facilitated by the membership images. Once a cluster of interest has been identified, it would be useful to evaluate how a clustering-enhanced membership image compares with traditional contrast enhancement techniques.

7.3 Final Conclusions

The goal of clustering methods: detecting an inherent clustering in the data set and then accurately describing it is a complex exploratory process. In two dimensional feature space, it seems that no method or strategy is as versatile as the human. In practical applications, therefore, misleading interpretations of cluster structure will have to be detected and corrected by human expertise.

Humans, however, need clustering methods to automate repetitive clustering tasks and to deal with the huge volumes of data that exist today. It is necessary that for data sets that possess cluster structures for which there is little doubt about their correct interpretation, a clustering method be found to perform accurately on them. It was in this vein, that we proposed PDI as a better successor to FCM. PDI, like other proposed successors to FCM, opens many questions about its wide applicability and accuracy.

Widening our view to beyond our PhD research, we offer the following conclusions on the subjects of clustering and image analysis :-

1. It is necessary that more research be conducted on the topic of clustering tendency — a topic that is little-studied at present. Tests for clustering tendency would precede actual clustering and would report on whether it would be worthwhile to use a clustering algorithm. This would probably involve comparing the information content of the data to that of randomly distributed data.

   The usual logic which consists of applying a clustering algorithm *first* and then assessing the clustering tendency from the algorithm's results assumes perfect accuracy of the clustering algorithm — which is not guaranteed. Furthermore, this two-step computational effort ought to be replaced with a simpler one-off test. The approaches of [Dunn, 1973; Windham, 1982] are interesting and should be followed on.

2. Graph-theoretic methods have not been combined with objective function methods. It would seem that this a fruitful research area as objective function methods rely on distance metrics that do not "see" connectivity or the lack of it, while that is graph-theoretic methods' strongest point.

References

Ahuja, N, & Tuceryan, M. 1989. Extraction of early perceptual structure in dot patterns: Integrating region, boundary and component gestalt. *Computer Vision, Graphics, and Image prcessing*, **48**(3), 304–356.

Al-Sultan, Khaled S, & Fedjki, Chawki A. 1997. A Tabu Search-Based Algorithm for the Fuzzy Clustering Problem. *Pattern Recognition*, **30**(12), 2023–2030.

AlSultan, K S, & Khan, M M. 1996. Computational experience on four algorithms for the hard clustering problem. *Pattern Recognition Letters*, **17**(3), 295–308.

AlSultan, K S, & Selim, S Z. 1993. A Global Algorithm for the Fuzzy Clustering Problem. *Pattern Recognition*, **26**(9), 1357–1361.

Backer, E. 1995. *Computer-Assisted Reasoning in Cluster Analysis*. Prentice Hall.

Bajcsy, P, & Ahuja, N. 1998. Location- and density-based hierarchical clustering using similarity analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(9), 1011–1015.

Banfield, Jeffrey D, & Raftery, Adrian E. 1993. Model-Based Gaussian and Non-Gaussian Clustering. *Biometrics*, **49**(September), 803–821.

Barni, M, Cappellini, V, & Mecocci, A. 1996. A possibilistic approach to clustering - Comments. *IEEE Transactions on Fuzzy Systems*, **4**(3), 393–396.

Bensaid, Amine M, Hall, Lawrence O, Bezdek, James C, Clarke, Laurence P, Silbiger, Martin L, Arrington, John A, & Murtagh, Reed F. 1996. Validity-Guided (Re)Clustering with Applications to Image Segmentation. *IEEE Transactions on Fuzzy Systems*, **4**(2), 112–123.

Bertsekas, Dimitri P. 1996. *Nonlinear Programming*. Athena Scientific.

Bezdek, J C, Hall, L O, & Clarke, L P. 1993. Review of MR Image Segmentation Techniques Using Pattern Recognition. *Medical Physics*, **20**(4), 1033–1048.

Bezdek, J C, Hall, LO, Clark, MC, Goldgof, Dmitri B, & Clarke, LP. 1997. Medical Image Analysis with Fuzzy Models. *Statistical Methods in Medical Research*, **6**, 191–214.

Bezdek, James. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press.

Bezdek, James C. 1980. A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms. *IEEE Transactions Pattern Analysis and Machine Intelligence*, **2**(1).

Bezdek, James C. 1992 (November). Integration and Generalization of LVQ and c-means clustering. *Pages 280–299 of: Intelligent Robots and Computer Vision*, vol. XI. SPIE, Boston, Massachusetts.

Bezdek, James C, & Pal, Sankar K (eds). 1992. *Fuzzy Models for Pattern Recognition*. IEEE Press.

Bishop, Christopher M. 1995. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.

Bobrowski, Leon, & Bezdek, James C. 1991. c-means Clustering with the l_1 and l_∞ Norms. *IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics*, **21**(3), 545–554.

Boudraa, A, Mallet, J-J, Besson, J-E, Bouyoucef, S, & Champier, J. 1993. Left Ventricle Automated Detection Method in Gated Isotopic Ventriculography Using Fuzzy Clustering. *IEEE Transactions on Medical Imaging*, **12**(3).

Brito, M R, Chavez, E L, Quiroz, A J, & Yukich, J E. 1997. Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statistics and Probability Letters*, **35**(1), 33–42.

Cannon, R L, Dave, J, & Bezdek, J C. 1986. Efficient implementation of the fuzzy c-means clustering algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**, 248–55.

Chaudhuri, D, & Chaudhuri, B B. 1997. A Novel Multiseed Nonhierarchical data clustering technique. *IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics*, **27**(5), 871–877.

Chen, Mu-Song, & Wang, Shinn-Wen. 1999. Fuzzy Clustering Analysis for Optimizing Fuzzy Membership Functions. *Fuzzy Sets and Systems*, **103**, 239–254.

Cheng, TW, Goldgof, DB, & Hall, LO. 1998. Fast Fuzzy Clustering. *Fuzzy Sets and Systems*, **93**(1), 49–56.

Chi, Zheru, Yan, Hong, & Pham, Tuan. 1996. *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*. World Scientific.

Chintalapudi, Krishna K, & Kam, Moshe. 1998 (October). The Credibilistic Fuzzy C Means clustering algorithm. *Pages 2034–2039 of: IEEE International Conference on Systems, Man, and Cybernetics*.

Clark, Mathew C, Hall, Lawrence O, Goldgof, Dimitry B, P, Clarke L, P, Velthuizen R, & Silbiger, Martin S. 1994. MRI Segmentation Using Fuzzy Clustering Techniques. *IEEE Engineering in Medicine and Biology Magazine*, **13**(5), 730–742.

Clark, MC, Hall, LO, Goldgof, DB, Velthuizen, R, Murtagh, FR, & Silbiger, MS. 1998. Automatic tumor segmentation using knowledge-based techniques. *IEEE Transactions on Medical Imaging*, **17**(2), 187–201.

Cox, Earl. 1998. *The Fuzzy Systems Handbook*. 2nd edn. Academic Press/Morgan Kaufmann.

Davé, Rajesh N. 1992. Boundary Detection through Fuzzy Clustering. *Pages 127–134 of: IEEE Conference on Fuzzy Systems*. IEEE Press.

Davé, Rajesh N, & Krishnapuram, Raghu. 1997. Robust clustering methods: A unified view. *IEEE Transactions on Fuzzy Systems*, **5**(2), 270–293.

Davson, H., & Segal, M. B. 1975. *Introduction to Physiology*. Vol. 1. Academic Press.

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

Dunn, J C. 1973. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *J. Cybernetics*, **3**(3), 32–57.

ElSonbaty, Y, & Ismail, M A. 1998. On-line hierarchical clustering. *Pattern Recognition Letters*, **19**(14), 1285–1291.

Everitt, Brian. 1974. *Cluster Analysis*. Heinemann Educational Books Ltd.

Everitt, Brian. 1978. *Graphical Techniques for Multivariate Data*. North Holland Publ.

Fayyad, U M, Piatetsky-Shapiro, G, & Smyth, P (eds). 1996a. *Advances in knowledge discovery and data mining*. AAAI Press/MIT Press.

Fayyad, Usama, Haussler, David, & Stolorz, Paul. 1996b (August). KDD for science data analysis: issues and examples. *In: Proceedings of the second international conference on knowledge discovery and data mining KDD-96*.

Fisher, Doug. 1996. Iterative Optimization and Simplification of Hierarchical Clusterings. *Journal of Artificial Intelligence Research*, **4**, 179–208.

Forgy, E W. 1965. Cluster analysis of multivariate data: efficiency vs. interpretability of classifications. *Biometrics*, **21**, 768–769. (Abstract).

Fraley, C, & Raftery, A E. 1998. How many clusters? Which clustering method? Answers via model-based cluster analysis. *Computer Journal*, **41**(8), 578–588.

Frigui, Hichem, & Krishnapuram, Raghu. 1996. A Comparison of Fuzzy Shell-Clustering Methods for the Detection of Ellipses. *IEEE Transactions on Fuzzy Systems*, **4**(2).

Frigui, Hichem, & Krishnapuram, Raghu. 1997. Clustering by Competitive Agglomeration. *Pattern Recognition*, **30**(7), 1109–1119.

Frigui, Hichem, & Krishnapuram, Raghu. 1999. A robust competitive clustering algorithm with applications in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**(5), 450–465.

Gath, I, & Geva, A. 1989. Unsupervised Optimal Fuzzy Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 773–781.

Gath, I, Iskoz, A S, & Cutsem, B Van. 1997. Data induced metric and fuzzy clustering of non-convex patterns of arbitrary shape. *Pattern Recognition Letters*, **18**, 541–553.

Geva, Amir B. 1999. Hierarchical-Fuzzy Clustering of Temporal-Patterns and its Application for Time-Series Prediction. *Pattern Recognition Letters*, **20**, 1519–1532.

Greenberg, Michael. 1998. *Advanced Engineering Mathematics*. 2nd edn. Prentice Hall.

Gustafson, Donald E, & Kessel, William C. 1979 (Jan. 10-12). Fuzzy Clustering with a Fuzzy Covariance Matrix. *Pages 761–766 of: Proc. IEEE CDC*.

Hall, Lawrence O, Ozyurt, Ibrahim Burak, & Bezdek, James C. 1999. Clustering with a Genetically Optimized Approach. *IEEE Transactions on Evolutionary Computation*, **3**(2), 103–112.

Hathaway, Richard J, & Bezdek, James C. 1993. Switching Regression Models and Fuzzy Clustering. *IEEE Transactions on Fuzzy Systems*, **1**(3), 195–203.

Hathaway, Richard J, & Bezdek, James C. 1995. Optimization of Clustering Criteria by Reformulation. *IEEE Transactions on Fuzzy Systems*, **3**(2).

Hathaway, Richard J, Bezdek, James C, & Pedrycz, Witold. 1996. A Parametric Model for Fusing Heterogeneous Fuzzy Data. *IEEE Transactions on Fuzzy Systems*, **4**(3), 270–281.

Hoppner, Frank, Klawonn, Frank, Kruse, Rudolf, & Runkler, Thomas. 1999. *Fuzzy Cluster Analysis*. Wiley.

Huang, ZX. 1998. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, **2**(3), 283–304.

Jain, Anil K. 1986. Cluster Analysis. *Chap. 2 of:* Young, Tzay Y, & Fu, King-Sun (eds), *Handbook of Pattern Recognition and Image Processing*, vol. 1. Academic Press.

Jardine, Nicholas, & Sibson, Robin. 1971. *Mathematical Taxonomy*. Wiley.

Karayiannis, N B, Bezdek, J C, Pal, N R, Hathaway, R J, & Pai, P-I. 1996. Repairs to GLVQ: A New Family of Competitive Learning Schemes. *IEEE Transactions on Neural Networks*, **7**(5), 1062–1071.

Klir, George J., Clair, Ute St., & Yuan, Bo. 1997. *Fuzzy Set Theory: Foundations and Applications*. Prentice Hall.

Kosko, Bart. 1993. *Fuzzy Thinking*. HarperCollinsPublishers.

Kothari, Ravi, & Pitts, Dax. 1999. On finding the number of clusters. *Pattern Recognition Letters*, **20**, 405–416.

Kottke, Dane P. 1992. Spatially Coherent Clustering Applied to Cardiac Angiograms. *Pages 11–12 of: Proceedings Of The Eighteenth Ieee Annual Northeast Bioengineering Conference*.

Krishnapuram, Raghu, & Keller, James M. 1993a. A Possibilistic Approach to Clustering. *IEEE Transactions on Fuzzy Systems*, **1**(2).

Krishnapuram, Raghu, & Keller, James M. 1993b. Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions. *IEEE Transactions on Fuzzy Systems*, **1**(3).

Krishnapuram, Raghu, & Keller, James M. 1996. The possibilistic C-means algorithm: Insights and recommendations. *IEEE Transactions on Fuzzy Systems*, **4**(3), 385–393.

Krishnapuram, Raghu, & Kim, Jongwoo. 1999. A note on the Gustafson-Kessel and Adaptive Fuzzy Clustering Algorithms. *IEEE Transactions on Fuzzy Systems*, **7**(4), 453–461.

Kwon, S H. 1998. Cluster Validity Index for Fuzzy Clustering. *Electronic Letters*, **34**(22).

Li, Zhaoping. 1997 (August). *Visual Segmentation without Classification in a Model of the Primary Visual Cortex*. Tech. rept. 1613. AI Lab, MIT.

Lin, Ja-Chen, & Lin, Wu-Ja. 1996. Real-time and Automatic Two-Class Clustering by Analytical Formulas. *Pattern Recognition*, **29**(11), 1919–1930.

Liu, Xiaohui. 2000. Progress in Intelligent Data Analysis. *Applied Intelligence*, **11**(3).

Mansfield, J R, Sowa, M G, Payette, J R, Abdulrauf, B, Stranc, M F, & Mantsch, H H. 1998. Tissue Viability by Multispectral Near Infrared Imaging: A fuzzy c-Means Clustering Analysis. *IEEE Transactions on Medical Imaging*, **17**(6).

Marchette, D J, Lorey, R A, & Priebe, C E. 1997. An Analysis of Local Feature Extraction in Digital Mammography. *Pattern Recognition*, **30**(9), 1547–54.

McLachlan, Geoffrey J, & Basford, Kaye E. 1988. *Mixture models: inference and applications to clustering*. Marcel Dekker.

Michalski, R. S., & Stepp, R. 1983. Automated Construction of Classifications: Conceptual Clustering versus Numerical Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **5**(4), 396–410.

Millar, Anne Michele, & Hamilton, David C. 1999. Modern Outlier Detection Methods and their Effect on Subsequent Inference. *Journal of Statistical Computation and Simulation*, **64**(2), 125–150.

Mirkin, Boris. 1999. Concept Learning and Feature Selection Based on Square-Error Clustering. *Machine Learning*, **35**(1), 25–39. a mechanism for selecting and evaluating features in the process of generating clusters is proposed.

Mitchell, Tom M. 1997. *Machine Learning*. WCB McGraw-Hill.

Mohan, Rakesh. 1992. Perceptual Organization for Scene Segmentation and Description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**(6).

Mohiaddin, R. H., & Longmore, D. B. 1993. Functional aspects of cardiovascular nuclear magnetic resonance imaging. *Circulation*, **88**(1), 264–281.

Mui, J K, W, Bacus J, & S, Fu K. 1977. A scene segmentation technique for microscopic cell images. *Pages 99–106 of: Proceedings of the Symposium on Computer-Aided Diagnosis of Medical Images*. IEEE, New York, NY, USA.

Niyogi, Partha. 1995. *The Information Complexity of Learning from Examples*. Ph.D. thesis, Department of Electrical Engineering and Computer Science, MIT, USA.

Pacheco, F A L. 1998. Finding the number of natural clusters in groundwater data sets using the concept of equivalence class. *Computers and Geosciences*, **24**(1), 7–15.

Pal, Nikhil R, & Bezdek, James C. 1995. On Cluster Validity for the Fuzzy c-Means Model. *IEEE Transactions on Fuzzy Systems*, **3**(3).

Pal, Nikhil R, & Pal, Sankar K. 1993. A Review on Image Segmentation Techniques. *Pattern Recognition*, **26**(9), 1277–1294.

Pei, JH, Fan, JL, Xie, WX, & Yang, X. 1996. A New Effective Soft Clustering Method — Section Set Fuzzy C-Means (S2FCM) Clustering. *Pages 773–776 of: ICSP '96 - 1996 3rd International Conference On Signal Processing, Proceedings, Vols I And Ii*.

Pettigrew, Roderic I, Oshinski, John N, Chatzimavroudis, George, & Dixon, W Thomas. 1999. MRI Techniques for Cardiovascular Imaging. *Journal of Magnetic Resonance Imaging*, **10**, 590–601.

Pham, Dzung L, & Prince, Jerry L. 1999. An Adaptive Fuzzy C-Means Algorithm for Image Segmentation in the Presence of Intensity Inhomogeneities. *Pattern Recognition Letters*, **20**(1), 57–68.

Philip, K P, Dove, E L, McPherson, D D, Gotteiner, N L, Stanford, W, & Chandran, K B. 1994. The Fuzzy Hough Transform — Feature Extraction in Medical Images. *IEEE Transactions on Medical Imaging*, **13**(2).

Rezaee, M Ramze, Lelieveldt, BPF, & Reiber, JHC. 1998. A new cluster validity index for the fuzzy c-means. *Pattern Recognition Letters*, **19**, 237–246.

Roberts, Stephen J. 1996 (August). Scale-space Unsupervised Cluster Analysis. *Pages 106–110 of: Proceedings of ICPR-96*, vol. 2. IEEE.

Rosenfeld, Azriel. 1979. Fuzzy Digital Topology. *Information Control*, **40**(1), 76–87.

Rosenfeld, Azriel. 1984. The Fuzzy Geometry of Image Subsets. *Pattern Recognition Letters*, **2**(5), 311–317.

Rousseeuw, P J, Trauwaert, E, & Kaufman, L. 1995. Fuzzy Clustering with Hight Contrast. *Journal of Computational and Applied Mathematics*, **64**, 81–90.

Rueckert, Daniel. 1997. *Segmentation and Tracking in Cardiovascular MR Images using Geometrically Deformable Models and Templates*. Ph.D. thesis, Department of Computing, Imperial College, University of London, London, UK.

Rumelhart, D E, McClelland, J L, & Group, PDP Research (eds). 1986. *Parallel distributed processing : explorations in the microstructure of cognition*. PDP Research Group, vol. 1. MIT Press.

Runkler, T A, & Bezdek, J C. 1999. Alternating cluster estimation: A new tool for clustering and function approximation. *IEEE Transactions on Fuzzy Systems*, **7**(4), 377–393.

Ruspini, Enrique H. 1969. A New Approach to Clustering. *Information Control*, **15**(1).

Ruspini, Enrique H. 1970. Numerical Methods for Fuzzy Clustering. *Information Sciences*, **2**, 319–350.

Russel, Stuart, & Norvig, Peter. 1995. *Artificial Intelligence*. Prentice Hall.

Schachter, B J, Davis, L S, & Rosenfeld, A. 1979. Some experiments in image segmentation by clustering of local feature values. *Pattern Recognition*, **11**(1), 19–28.

Selim, Shokri Z, & Kamel, M S. 1992. On the Mathematical Properties of the Fuzzy c-means Algorithm. *Fuzzy Sets and Systems*, **49**, 181–191.

Shapiro, Larry S. 1995. *Affine Analysis of Image Sequences*. Ph.D. thesis, University of Oxford.

Shihab, Ahmed Ismail, & Burger, Peter. 1998a. The Analysis of Cardiac Velocity MR Images Using Fuzzy Clustering. *Pages 176–183 of: Proc. SPIE Medical Imaging 1998 — Physiology and Function from Multidimensional Images*, vol. 3337. San Diego, USA: SPIE.

Shihab, Ahmed Ismail, & Burger, Peter. 1998b (July). Tracking the LV in Velocity MR Images Using Fuzzy Clustering. *In: Proc. Medical Image Understanding and Analysis*.

Smith, Norman Ronald. 1998. *Fast and Automatic Techniques for 3D Visualisation of MRI Data*. Ph.D. thesis, Imperial College, University of London, London, UK.

Stepp, R., & Michalski, R. S. 1986. Conceptual Clustering of Structured Objects: A Goal-Oriented Approach. *AI Journal*.

Tolias, Yannis A., & Panas, Stavros M. 1998a. A Fuzzy Vessel Tracking Algorithm for Retinal Images Based on Fuzzy Clustering. *IEEE Transactions On Medical Imaging*, **17**(2), 263–273.

Tolias, Yannis A., & Panas, Stavros M. 1998b. Image Segmentation by a Fuzzy Clustering Algorithm Using Adaptive Spatially Constrained Membership Functions. *IEEE Transactions on Systems, Man, and Cybernetics — Part A: Systems and Humans*, **28**(3), 359–370.

Tuceryan, M, & Jain, A K. 1993. Texture Analysis. *Pages 235–276 of: Handbook of Pattern Recognition and Computer Vision*. World Scientific.

Tyree, Eric W, & Long, J A. 1999. The use of linked line segments for cluster representation and data reduction. *Pattern Recognition Letters*, **20**(1), 21–29.

Udupa, J K, & Samarasekera, S. 1996. Fuzzy Connectedness and Object Definition - Theory, Algorithms, and Applications in Image Segmentation. *Graphical Models and Image Processing*, **58**(3), 246–261.

Underwood, R., & Firmin, D. 1991. *Magnetic Resonance of the Cardiovascular System*. Blackwell Scientific Publications.

van der Wall, E. E., & de Ross, A. 1991. *Magnetic Resonance Imaging in Coronary Artery Disease*. Kluwer Academic Publishers.

Velthuizen, Robert P, Hall, Lawrence O, Clarke, Laurence P, & Silbiger, Martin L. 1997. An Investigation of Mountain Method Clustering for Large Data Sets. *Pattern Recognition*, **30**(7), 1121–1135.

Wilson, Kathleen J W. 1990. *Anatomy and Physiology in Health and Illness*. 7th edn. ELBS Churchill Livingstone.

Windham, Michael P. 1982. Cluster Validity for the Fuzzy c-Means Clustering Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **4**(4), 357–363.

Xie, X-L, & Beni, G. 1991. Validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**(8), 841–847.

Yang, Guang-Zhong. 1998. Exploring In Vivo Blood Flow Dynamics. *IEEE Engineering in Medicine and Biology*, **17**(3).

Zadeh, Lotfi A. 1965. Fuzzy Sets. *Information Control*, **8**, 338–353.

Zadeh, Lotfi A. 1995. Probability Theory and Fuzzy Logic are Complementary rather than Competitive. *Technometrics*, **37**, 271–276.

Zadeh, Lotfi A. 1996. Fuzzy Logic = Computing with Words. *IEEE Transactions on Fuzzy Systems*, **4**(2), 103–111.

Zadeh, Lotfi A. 1999. From Computing with Numbers to Computing with Words – From Manipulation of Measurements to Manipulation of Perceptions. *IEEE Transactions on Circuits and Systems*, **45**(1), 105–119.

Zadeh, Lotfi A, & Klir, George J. 1996. *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems : Selected Papers by Lotfi A. Zadeh*. World Scientific Pub Co.

Zahn, Charles T. 1971. Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers*, **C-20**(1), 68–86.

APPENDIX A

Details of Benchmark Data Suite Generation

If D_1 were plotted against D_2 as in Figure A.1, the line $D_1 + D_2 = 20$ would describe configurations where the clusters touch. Since it is not our aim to test a clustering algorithm on detection of overlapping clusters, only the 200 configurations under the line should be considered. If we eliminate, by symmetry, equivalent diameter configurations (*i.e.*, $D_1 = 5$ and $D_2 = 10$ is equivalent to $D_1 = 10$ and $D_2 = 5$), then only 100 diameter configurations remain.

In the above, when we cut down the number of possible diameter configurations to 100, we said that a configuration of $D_1 = 5$ and $D_2 = 10$ is equivalent to $D_1 = 10$ and $D_2 = 5$. However, when we take the populations into consideration, this is no longer case. For example, imagine a $1 : 10$ population configuration combined with a $5 : 10$ diameter configuration: $(P_1 = 1, D_1 = 5)$ and $(P_2 = 10, D_2 = 10)$. If we keep the populations as they are but swap the diameters, the resulting configuration, $(P_1 = 1, D_1 = 10)$ and $(P_2 = 10, D_2 = 5)$, is not equivalent to the former configuration. This is illustrated in Figure A.2. Thus, it seems we must keep the second configuration as it describes a different data set, and we can not discard the “equivalent” region of Figure A.1.

However, when we arrive at the configuration consisting of $10 : 1$ population ratio

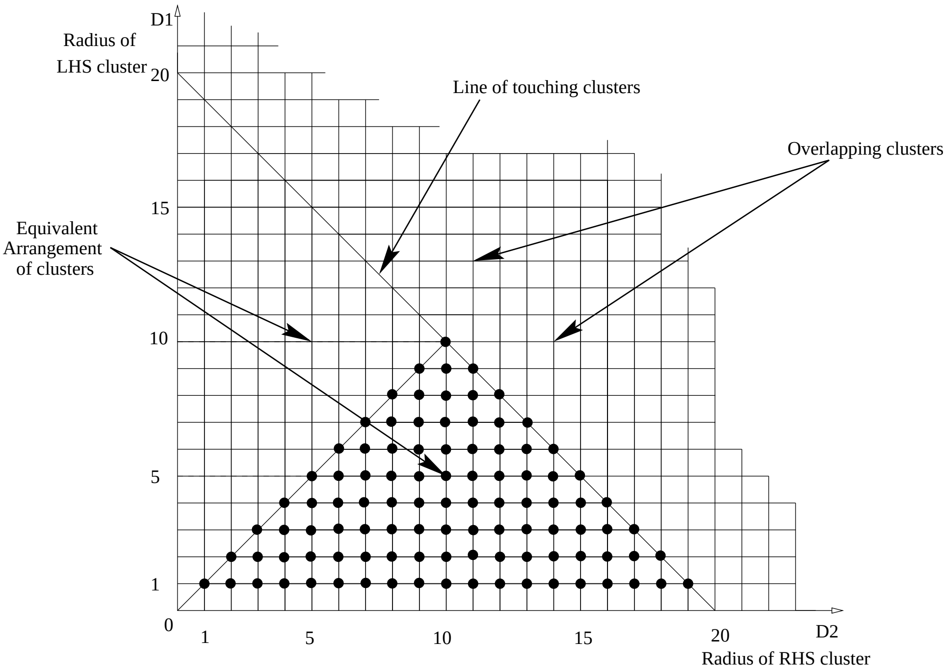


Figure A.1: Plot of possible diameter configurations. Data sets corresponding to the black dots in the triangular region were generated. If we eliminate overlapping and equivalent configurations only 100 data sets remain.

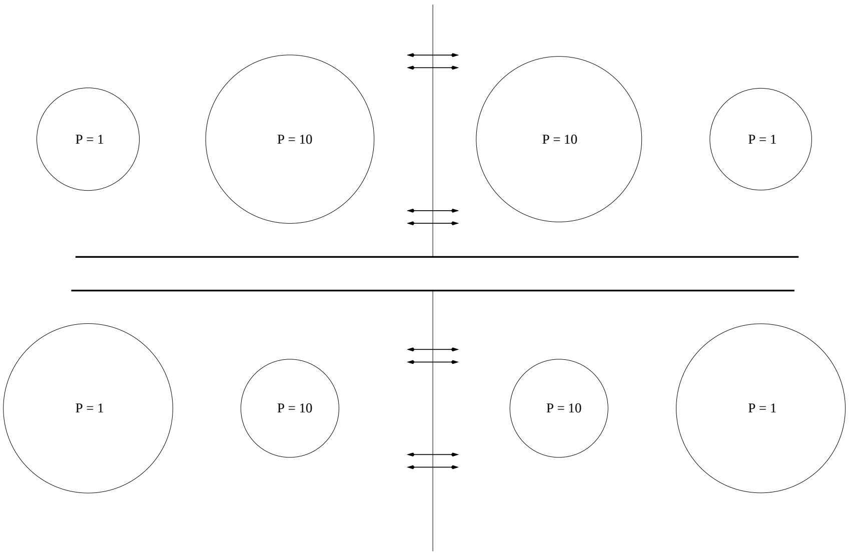


Figure A.2: Each row illustrates equivalent p-d configurations. Only one of each suffices when generating the suite of data sets.

and $5 : 10$ diameter ratio we will discover that it is the same as that of the second configuration above. Therefore, in order to not count the same $p - d$ configurations twice, we can still consider only the 100 diameter configurations of Figure A.1 for each of the nine population configurations.

APPENDIX B

Derivation of 2D 2-Prototype Model

These are the derivations used to plot the shape of FCM and PDI objective functions on Mathematica.

B.1 FCM's derivations

Assume two 1D cluster prototypes located at the origin and $(1, 0)$ respectively. Denote prototype at the origin by a and the other by b.

Assume a point located at location (x) somewhere on the x-axis. Let's calculate its contribution towards the FCM objective function. Assume $m = 2$.

$$J_x = u_{xa}^2 d_{xa}^2 + u_{xb}^2 d_{xb}^2$$

$$u_{xa} = \frac{(1/d_{xa}^2)}{(1/d_{xa}^2) + (1/d_{xb}^2)}$$

$$u_{xb} = \frac{(1/d_{xb}^2)}{(1/d_{xa}^2) + (1/d_{xb}^2)}$$

$$J_x = \frac{(1/d_{xa}^2) + (1/d_{xb}^2)}{[(1/d_{xa}^2) + (1/d_{xb}^2)]^2}$$

Since

$$d_{xa} = x \text{ and } d_{xb} = x - 1$$

$$\Rightarrow J_x = \frac{(1/x^2) + (1/(x-1)^2)}{[(1/x^2) + (1/(x-1)^2)]^2}$$

which simplifies to:

$$J_x = \frac{1}{\frac{1}{(x-1)^2} + \frac{1}{x^2}}$$

For the two dimensional case, where the point is now located anywhere on the plane and is of coordinates (x, y), we derive the point's contribution, J_{xy}:

$$d_{xa}^2 = x^2 + y^2 \text{ and } d_{xb}^2 = (x-1)^2 + y^2$$

$$\Rightarrow J_{xy} = \frac{(\frac{1}{x^2+y^2}) + (\frac{1}{(x-1)^2+y^2})}{[(\frac{1}{x^2+y^2}) + (\frac{1}{(x-1)^2+y^2})]^2}$$

which simplifies to:

$$J_{xy} = \frac{1}{\frac{1}{(x-1)^2+y^2} + \frac{1}{x^2+y^2}}$$

B.2 PDI's derivations

Assume two 2D cluster prototypes located at the origin and $(1, 0)$ respectively. Denote prototype at the origin by a and the other by b.

Assume a point located at coordinates (x, y) somewhere on the $x - y$-plane. Let's calculate its contribution towards the PDI objective function. Assume $m = 2$ and the normalisers of each cluster to have values of ρ_a and $\rho_b = (1 - \rho_a)$.

$$J_x = \frac{u_{xa}^2 d_{xa}^2}{\rho_a^r} + \frac{u_{xb}^2 d_{xb}^2}{(1-\rho_a)^r}$$

$$u_{xa} = \frac{(\rho_a^r / d_{xa}^2)}{(\rho_a^r / d_{xa}^2) + ((1-\rho_a)^r / d_{xb}^2)}$$

$$u_{xb} = \frac{((1-\rho_a)^r/d_{xb}^2)}{(\rho_a^r/d_{xa}^2) + ((1-\rho_a)^r/d_{xb}^2)}$$

Since

$$d_{xa} = x \text{ and } d_{xb} = x - 1$$

$$\Rightarrow J_x = \frac{1}{\frac{(1-\rho_a)^r}{(x-1)^2} + \frac{\rho_a^r}{x^2}}$$

For the two dimensional case, where the point is now located anywhere on the plane and is of coordinates (x, y), we derive the point's contribution, J_{xy}:

$$d_{xa}^2 = x^2 + y^2 \text{ and } d_{xb}^2 = (x-1)^2 + y^2$$

$$\Rightarrow J_{xy} = \frac{1}{\frac{(1-\rho_a)^r}{(x-1)^2+y^2} + \frac{\rho_a^r}{x^2+y^2}}$$

APPENDIX C

Derivation of PDI Optimality Conditions

Finding a solution is effected with the Lagrange multipliers method [Bertsekas, 1996]. Since an exact analytical solution can not be obtained, the first order optimality conditions are found and these are used as update equations in Picard iterations. The algorithm is started with any initial values for $\mathcal{U}, \mathcal{P}$, and $\underline{\rho}$ and then these are iteratively improved until convergence is attained.

We define the Lagranian function as follows:

$$L(\mathcal{U}, \underline{\rho}, \mathcal{P}) = \sum_{i=1}^{c} \frac{1}{\rho_i^r} \sum_{k=1}^{N} u_{ik}^m d_{ik}^2 + \underline{\lambda}(\sum_{i=1}^{c} u_{ik} - 1) + \mu(\sum_{i=1}^{c} \rho_i - 1)$$

Where $\underline{\lambda}$ and μ are Lagrange multipliers for each of the constraints. $\underline{\lambda}$ is a vector of N elements, and μ is a single value.

According to the Lagrange multipliers method, the necessary first order optimality conditions are:

$$\nabla_{\mathcal{U}} L = 0, \tag{C.1}$$

$$\nabla_{\mathcal{P}} L = 0, \tag{C.2}$$

$$\nabla_{\underline{\rho}} L = 0, \tag{C.3}$$

$$\sum_{i=1}^{c} u_{ik} - 1 = 0 \quad \forall k = 1..N, \tag{C.4}$$

and,

$$\sum_{i=1}^{c} \rho_i - 1 = 0 \quad \forall i = 1..c. \tag{C.5}$$

From the optimality condition of equation C.1, we obtain:

$$\frac{m u_{ik}^{m-1} d_{ik}^2}{\rho_i^r} - \lambda_k = 0. \tag{C.6}$$

$$\Rightarrow u_{ik} = [\frac{\lambda_k \rho_i^r}{m d_{ik}^2}]^{1/m-1}. \tag{C.7}$$

Substituting the above equation in C.4, we obtain:

$$\sum_{i=1}^{c} [\frac{\lambda_k \rho_i^r}{m d_{ik}^2}]^{1/m-1} = 1.$$

$$\Rightarrow \lambda_k = \frac{m}{[\sum_{i=1}^{c} (\rho_i^r / d_{ik}^2)^{1/m-1}]^{1/m-1}}. \tag{C.8}$$

Substituting λ_k into C.7, we obtain the update equation for u_{ik}:

$$u_{ik} = \frac{(\rho_i^r / d_{ik}^2)^{1/m-1}}{\sum_{i=1}^{c} (\rho_i^r / d_{ik}^2)^{1/m-1}}. \tag{C.9}$$

From the optimality condition of equation C.2 and noting that d_{ik} is any inner-product induced norm on the difference between x_k and v_i, we obtain:

$$2/\rho_i^r \sum_{k=1}^{N} u_{ik}^m (x_k - \mathbf{p}_i) = 0.$$

$$\Rightarrow \sum_{k=1}^{N} u_{ik}^m x_k = \sum_{k=1}^{N} u_{ik}^m \mathbf{p}_i$$

which leads to the following update equation for p_i:

$$\mathbf{p}_i = \frac{\sum_{k=1}^{N} u_{ik}^m x_k}{\sum_{k=1}^{N} u_{ik}^m}. \tag{C.10}$$

Finally, from the optimality condition of equation C.3, we obtain:

$$\rho_i = (\frac{r\sum_{k=1}^{N} u_{ik}^{m} d_{ik}^{2}}{\mu})^{1/r+1}. \tag{C.11}$$

And from the optimality condition of equation C.5, we obtain:

$$\sum_{i=1}^{c}(\frac{r\sum_{k=1}^{N} u_{ik}^{m} d_{ik}^{2}}{\mu})^{1/r+1} = 1. \tag{C.12}$$

$$\Rightarrow \mu = [\sum_{i=1}^{c}[\sum_{k=1}^{N} u_{ik}^{m} d_{ik}^{2}]^{\frac{1}{r+1}}]^{r+1}. \tag{C.13}$$

Substituting for μ in equation C.11, we obtain the update equation for ρ_i:

$$\rho_i = \frac{(\sum_{k=1}^{N} u_{ik}^{m} d_{ik}^{2})^{1/r+1}}{\sum_{i=1}^{c}(\sum_{k=1}^{N} u_{ik}^{m} d_{ik}^{2})^{1/r+1}}. \tag{C.14}$$