7-7-1994

Gradient based fuzzy c-means algorithm

Issam J. Dagher
Florida International University

DOI: 10.25148/etd.FI14061586

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

Gradient Based Fuzzy c-means Algorithm

A thesis submitted in partial satisfaction of the

requirements for the degree of

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

by

Issam J. Dagher

1994

To: Dean Gordon R. Hopkins
College of Engineering and Design

This thesis, written by Issam J. Dagher, and entitled GRADIENT BASED FUZZY C_MEANS ALGORITHM, having been approved in respect to style and intellectual content, is referred to you for judgement.

We have read this thesis and recommend that it be approved.

Malek Adjouadi

Malcolm Heimer

Dong C. Park, Major Professor

Date of Defense: July 7, 1994.

The thesis of Issam J. Dagher is approved.

Dean Gordon R. Hopkins
College of Engineering and Design

Dr. Richard L. Campbell
Dean of Graduate Studies

Florida International University, 1994

ACKNOWLEDGEMENTS

I would like to thank the members of my committee for all their support and constructive criticism in the development of my thesis. I also wish to give thanks to the members of the Intelligent Computing Research Laboratory for their timely comments and overall help.

I also would like to give a special thanks to my major professor, Dr. Park, for his extended support in my research and thesis development, to my parents for their good example.

The research described in this thesis was supported in part by the National Science Foundation grant CDA-9313624 with the Center for Advanced Technology and Education.

ABSTRACT OF THE THESIS

GRADIENT BASED FUZZY C-MEANS ALGORITHM

by

Issam J. Dagher

Florida International University, 1994

Professor Dong C. Park, Major Professor

A clustering algorithm based on the Fuzzy c-means algorithm (FCM) and the gradient descent method is presented. In the FCM, the minimization process of the objective function is proceeded by solving two equations alternatively in an iterative fashion. Each iteration requires the use of all the data at once. In our proposed approach one datum is presented at a time to the network and the minimization is proceeded using the gradient descent method. Compared to FCM,the experimental results show that our algorithm is very competitive in terms of speed and stability of convergence for large number of data.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The objective of clustering algorithms is the grouping of similar objects and separating of dissimilar ones [1]. Kohonen [2] introduced a network where the learning is based on the Winner-Takes-All rule. It is based on the premise that one of the output neurons has the maximum response due to an input. Only one winner is declared (membership of 1) and all others are loosers (memberships of 0).

In 1973, Dunn [3] introduced a generalization of the Winner-Takes-All rule. He combined Zadeh's set concept with the criterion approach to clustering. The membership grade is spread over $\{0, \cdots, 1\}$ which will be able to "signal the presence or absence of Well-Separated Clusters". He derived the necessary conditions for minimizing an objective function. He called this method *fuzzy ISODATA*. A generalization of the *fuzzy ISODATA* is done by Bezdek [4]. Bezdek defined a family of objective functions $\{J_m,\ 1 < m < \infty\}$, and established a convergence theorem for that family. Windham presented a cluster validity for the FCM algorithm[5]. He obtained a measure by computing the ratio of the smallest membership to the largest one and transforming this ratio into a probability function. In [6], an unsupervised fuzzy clustering algorithm had been presented. The algorithm located the regions where the data have low degree of memberships, and tried to augment the number of clusters according to these regions.

In this thesis, we present a new algorithm called Gradient Based FCM (GBFCM) which combines the characteristics of Kohonen network (presenting one datum at a time and applying the gradient descent method) and the FCM algorithm (continuous values of the membership grades in the range $\{0, \cdots, 1\}$). In Chapter 2, different clustering techniques are presented. In Chapter 3, the minimization procedure (gradient descent method) is presented. The decision functions for hard and fuzzy clustering are presented respectively in Chapters 4 and 5. A derivation of the GBFCM algorithm is presented in Chapter 6 establishing a comparison between FCM and GBFCM and showing the pseudocode of the algorithm. Chapters 7,8,and 9 contain the results of applying GBFCM on image segmentation,regression analysis, and reduction of features of the Fourier transform of the EMG signal. We conclude in Chapter 10 with a short discussion about the algorithm and how it could be improved.

Chapter 2

Clustering Techniques

Clustering is understood to be the grouping of similar objects and seperating of dissimilar ones. It is the problem of identifying the number of classes according to a certain criterion, and assigning the membership of the patterns in these classes. No information is available as far as the desired responses . This kind of learning is called *unsupervised learning* (figure 2.1)as opposed to the supervised one where the teacher's signal d is available at each iteration (figure 2.2). Because the desired response is not known, explicit error (difference between the output and the desired response) cannot be used to improve the network behavior. The pattern set $\{\vec{x_1}, \vec{x_2}, \ldots, \vec{x_N}\}$ is submitted as input to the network and a decision function is required to identify possible clusters .

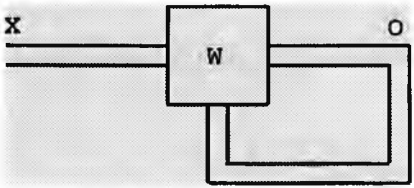

Figure 2.1: Unsupervised learning.

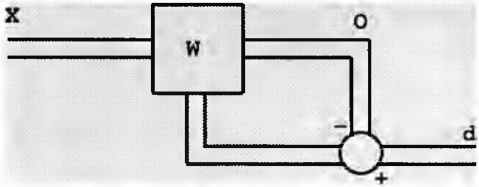

Figure 2.2: Supervised learning.

Depending on the choice of this decision function, two approaches can be used:

- Crisp clustering.
- Fuzzy clustering.

2.1 Crisp clustering

If X is a finite set, a family $\{A_i; 1 \leq i \leq c\}$ is a crisp c-clusters of X if:

$$\cup A_i = X \tag{2.1}$$

$$A_i \cap A_j = \emptyset \text{ for } i \neq j \tag{2.2}$$

The membership functions are:

$$\mu_i(x_k) = \begin{cases} 1, & \text{if } x_k \in A_i \\ 0, & \text{if } x_k \notin A_i \end{cases} \tag{2.3}$$

$$\sum_{i=1}^{c} \mu_i(x_k) = 1 \text{ for } 1 \leq i \leq c \text{ and } 1 \leq k \leq n \tag{2.4}$$

$$0 < \sum_{k=1}^{n} \mu_i(x_k) < n \tag{2.5}$$

where n is the number of data, and c is the number of clusters.

For example :

$X = \{x1, x2, x3\} \quad n = 3$

for 2 clusters $c = 2$:

Example 1:

ex1	x1	x2	x3
A1	1	1	0
A2	0	0	1

Table 2.1: Hard membership grades.

The table shows that $x1$ and $x2$ are in the class (cluster) $A1$ and $x3$ is in $A2$.

2.2 Fuzzy clustering

A fuzzy clustering is defined by the following properties of the membership function:

$$\mu_i(x_k) \in \{0, \cdots, 1\} \tag{2.6}$$

$$\sum_{i=1}^{c} \mu_i(x_k) = 1 \tag{2.7}$$

$$0 < \sum_{k=1}^{n} \mu_i(x_k) < n \tag{2.8}$$

Example 2:

ex3	x1	x2	x3
A1	.91	.58	.13
A2	.09	.42	.87

Table 2.2: Fuzzy membership grades.

From the table, the following conclusions can be drawn:

- $x1$ is most likely in $A1$.

- $x3$ is most likely in $A2$.

- $x2$ combines both properties of $A1$ and $A2$ even though it does not belong completely to one of the groups.

Chapter 3

Decision Functions - Optimization

The basic mathematical optimization problem is to minimize a scalar quantity E which is a function of n system parameters $x_1, x_2, \cdots, x_n$ These variables must be adjusted to obtain the minimum required. The problem can be defined as:

$$\text{minimize} \;\; E = f(x_1, x_2, \cdots, x_n) \tag{3.1}$$

The function f is referred to as the *objective function* whose value is the quantity which is to be minimized.

Gradient methods for optimization are based on the Taylor Series expansion given by:

$$f(\vec{x} + \vec{\Delta x}) = f(\vec{x}) + \vec{g}^T \vec{\Delta x} + \frac{1}{2}\vec{\Delta x}^T H \vec{\Delta x} + \cdots \tag{3.2}$$

where $\vec{\Delta x}$ is the change in the parameter vector $\vec{x}$.

$$\vec{x}^T = [\Delta x_1 \Delta x_2 \cdots \Delta x_n] \tag{3.3}$$

is the transpose of x. And

$$\vec{g}^T = \nabla f = [\frac{\partial f}{\partial x_1} \frac{\partial f}{\partial x_2} \cdots \frac{\partial f}{\partial x_n}] \tag{3.4}$$

is the ***gradient vector*** (first derivatives) of the objective function f. And

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} \cdots \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} \cdots \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} \cdots \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix} \tag{3.5}$$

is the *Hessian matrix* (second derivatives) of the objective function f.

Gradient methods neglect third and higher order derivatives. The objective function is then approximated by:

$$f(\vec{x} + \vec{\Delta x}) = f(\vec{x}) + \vec{g}^T \vec{\Delta x} + \frac{1}{2} \vec{\Delta x}^T H \vec{\Delta x} \tag{3.6}$$

The last two terms on the right hand side are a scalar correction to the function value at $\vec{x}$. This correction is denoted by ΔE so that

$$f(\vec{x} + \vec{\Delta x}) \approx E + \Delta E \tag{3.7}$$

3.1 Steepest descent method

The steepest descent method uses the Jacobian gradient $\vec{g}$ to determine a suitable direction of movement. It is the fundamental first order method. The first approximation to the optimum defines a point at which the function is evaluated to yield E and a suitable change ΔE is found by evaluation of the Jacobian $\vec{g}$. The effect on E of a small change $\vec{\Delta x}$ in $\vec{x}$ is given to the first order approximation by

$$\Delta E = \vec{g}^T \vec{\Delta x} \tag{3.8}$$

This equation can be thought of as involving a dot product of two vectors $\vec{g}^T$ and $\vec{\Delta x}$. This product is equal to:

$$\vec{g}^T \vec{\Delta x} = \|\vec{g}\| \|\vec{\Delta x}\| \cos\theta \tag{3.9}$$

for fixed magnitudes $\|\vec{g}\|$ and $\|\vec{\Delta x}\|$, ΔE depends on $\cos\theta$, taking a maximum positive value if $\theta = 0$ and a maximum negative value if $\theta = \pi$. The maximum reduction in E therefore occurs if $\theta = \pi$, from which it follows that the minimizing change $\vec{\Delta x}$ in $\vec{x}$ should be in the direction of the negative gradient $-\vec{g}$.

3.2 Steepest descent algorithm

The algorithm for implementing the steepest descent method is the following:

1. Set $k = 0$, and ϵ as a small error value.

2. Input data $\vec{x_k}$.

3. Evaluate $E_k = f(\vec{x_k})$.

   If $\|E_k - E_{k-1}\| \leq \epsilon$, output $\vec{x}_{min} = \vec{x_k}$ and $E_{min} = E_k$. Stop the process .

4. Evaluate the gradient $\vec{g_k}$ at point $\vec{x_k}$.

5. Generate a new point $\vec{x}_{k+1} = \vec{x_k} - \alpha_k \vec{g_k}$, where α_k is a constant .

6. Set $k = k + 1$. Goto 2.

The rate of change of the objective function at each iteration along the direction of the negative gradient is determined by α_k which could be chosen to be a constant between 0 and 1 at all the iterations. This method could be improved by choosing at each iteration the optimal value of α_k. This is done by the following equation:

$$\frac{\partial f(\vec{x_k} - \alpha_k \vec{g_k})}{\partial \alpha_k} = 0 \tag{3.10}$$

which will give what is called the *optimal steepest descent method.* An example of an objective function and its projection are in figures 3.1 and 3.2. The progress of the Gradient Descent algorithm is shown in figure 3.3.

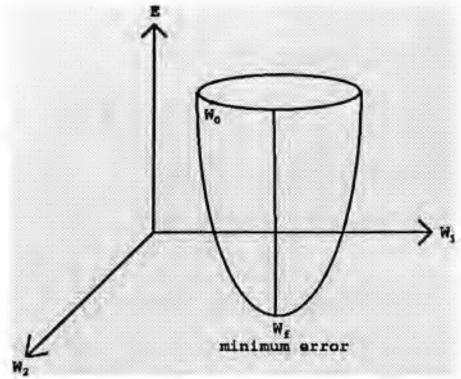

Figure 3.1: Example of an objective function.

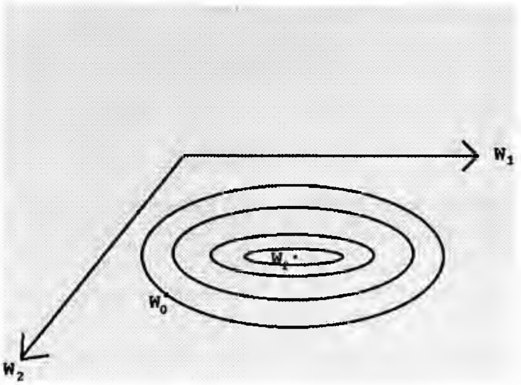

Figure 3.2: Projection of the objective function.

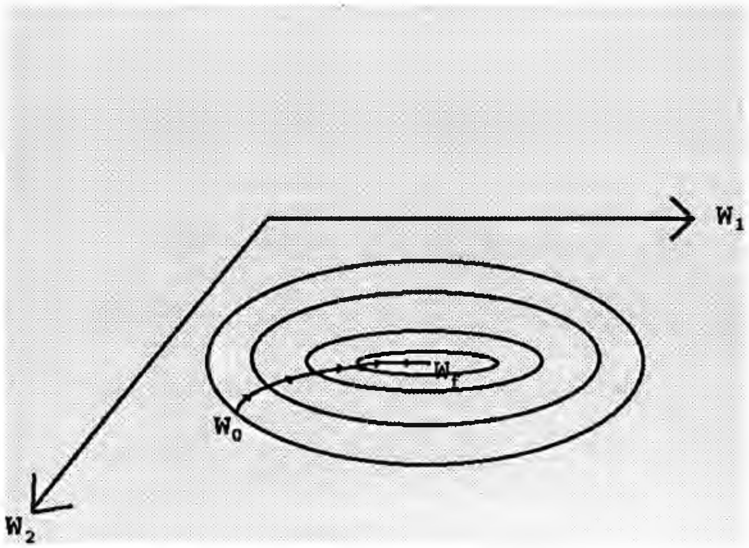

Figure 3.3: Proceedings of the Gradient Descent method.

Chapter 4

Decision Function For Hard Clustering - Kohonen Network

The most common decision function is the Euclidean distance defined as :

$$\|\vec{x} - \vec{w}\| = \sqrt{(\vec{x} - \vec{w})^t(\vec{x} - \vec{w})} \tag{4.1}$$

This rule of similarity states that the smaller the distance, the closer are the patterns. The objective function is defined as:

$$E(\vec{w}) = \|\vec{x} - \vec{w}\|^2 \tag{4.2}$$

The problem is reduced to:

$$\text{minimize} \quad E(\vec{w}) \tag{4.3}$$

This objective function could be minimized by calculating the gradient of E:

$$\frac{\partial E}{\partial \vec{w}} = -2(\vec{x} - \vec{w}) \tag{4.4}$$

And then changing $\vec{w}$ by:

$$\vec{w}(k) = \vec{w}(k-1) - \alpha_k \frac{\partial E}{\partial \vec{w}} \tag{4.5}$$

which is equal to:

$$\vec{w}(k) = \vec{w}(k-1) + 2\alpha_k(\vec{x} - \vec{w}) \tag{4.6}$$

4.1 Kohonen Network

Kohonen's network classifies input vectors into one of the specified number of c categories according to the clusters detected in the training set. The training is performed in an unsupervised mode and the network undergoes the self-organization process. During the training, dissimilar vectors are rejected, and only one ,the most similar, is accepted for weight adaptation. In figure 4.1, the network consists of 2 inputs and 3 outputs. In general, the network consists of:

- The number of inputs which represent the dimension of the input vector $\vec{x}$. In our example, $\vec{x}$ is 2 dimensional vector $[x_1 \;\; x_2]$.
- The number of outputs which represent the number of clusters. In our example, 3 clusters are needed.
- And the weight vector $\vec{w}$ which will be updated at every iteration.

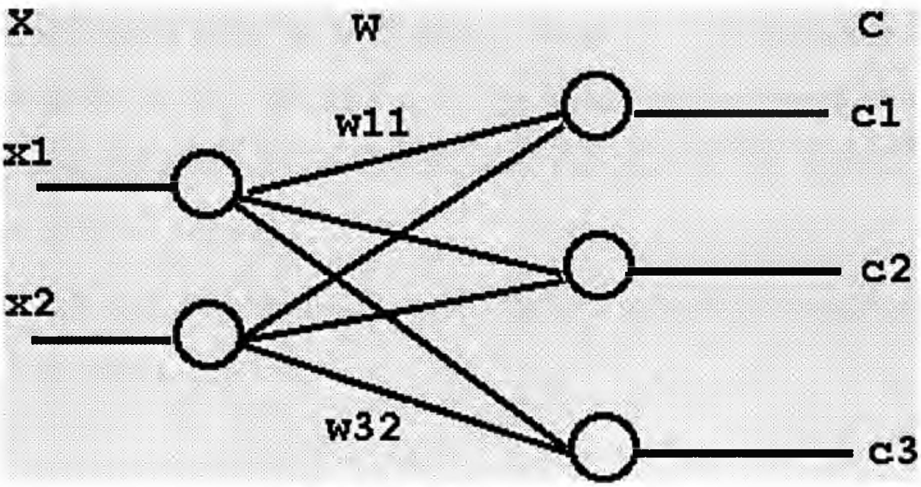

Figure 4.1: Kohonen network.

The updating of the weight vector is proceeded by using the *Winner-Takes-All* method.

4.2 Winner-Takes-All

This learning is based on the premise that one of the neurons in the output layer of figure 4.1 has the maximum response due to input $\vec{x}$. This maximum response is determined by the decision function defined earlier (such as the Euclidian distance). This neuron is declared the *winner*. In other term, the winner is the output of the network which has minimum distance to the input $\vec{x}$. The learning is proceeded as follows:

- Calculate the distances between the different weight vectors and the input data vecor:

$$\|\vec{x} - \vec{w}\| = \sqrt{\sum_{i=1}^{D} (x_i - w_i)^2} \tag{4.7}$$

- The winning neuron will determine which cluster the input data is in.

- Once the winning neuron is determined,all the weights which are connected to that neuron will be updated using the following formula:

$$\vec{w}(k-1) = \vec{w}(k) + \alpha_k(\vec{x} - \vec{w}(k)) \tag{4.8}$$

Chapter 5

Decision Function For Fuzzy Clustering - Fuzzy C-Means Algorithm

Kohonen network assumes that an object can belong to one and only one class. In practice, the separation of clusters is a fuzzy notion and hence the concept of fuzzy subsets offers special advantages over conventional clustering by allowing algorithms to assign each object a partial or distributed membership to each of the c clusters.

5.1 Hard vs.Fuzzy partitions

The following example designed by Bezdek illustrates the difference between hard and fuzzy partitions best:

A typical 2 partitions of $S = \{\text{Peach,Plum,Nectarine}\}$ can be given as follows :

Hard	z1	z2	z3
Peach	1	0	0
Plum	0	1	1

Table 5.1: Hard partitions.

where $z1$ =peach,$z2$ =plum,and $z3$ =nectarine.

The table shows that for the nectarine, the hard clustering assigns full membership to one of the two crisp subsets partitioning this data. In this case, the nectarine is being considered as a plum (membership of 1). Fuzzy clustering enables algorithms to avoid such mistakes. In the following table, the final column allocates most (0.6) of the membership to the plum class. But also assigns a lesser membership (0.4) to the peach class.

Fuzzy	z1	z2	z3
Peach	.9	.2	.4
Plum	.1	.8	.6

Table 5.2: Fuzzy partitions.

Columns such as the the one for the nectarine serve a useful purpose. Lack of strong membership in a single class is a signal " to take a second look." Hard partitions of data cannot suggest this.

In the following figure, 3 clusters were assumed. The membership grade μ

- can only have 3 values $e1$,$e2$,or $e3$ for hard clustering.

- can span the surface E for fuzzy clustering .

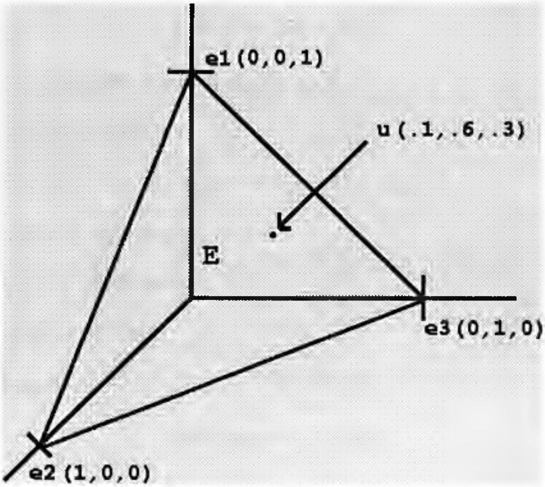

Figure 5.1: Membership grades for 3 clusters.

5.2 Fuzzy c-means partitions

The objective function,which measures the desirability of clustering candidates, is defined as:

$$J_m(\vec{\mu},\vec{v}) = \sum_{k=1}^{n}\sum_{i=1}^{c}\{(\mu_i(\vec{x}_k))^m(d_i(\vec{x}_k))^2\} \tag{5.1}$$

where

$$d_i(\vec{x}_k)^2 = \|\vec{x}_k - \vec{v}_i\|^2 \tag{5.2}$$

the distance from the input data $\vec{x}_k$ to $\vec{v}_i$.

$\vec{v}_i$ is the center of cluster i.

$\mu_i(\vec{x}_k)$ is the membership value of $\vec{x}_k$ in the cluster i.

m is a weighting exponent. $m \geq 1$.

c is the number of clusters.

n is the number of data.

The problem is :

$$\text{Minimize } J_m(\vec{\mu},\vec{v}). \tag{5.3}$$

5.2.1 Minimization of $J_m(\vec{\mu},\vec{v})$

The minimization of $J_m(\vec{\mu},\vec{v})$ is done by :

$$\frac{\partial J_m(\vec{\mu},\vec{v})}{\partial\vec{\mu}} = 0 \tag{5.4}$$

and

$$\frac{\partial J_m(\vec{\mu},\vec{v})}{\partial\vec{v}} = 0 \tag{5.5}$$

This will give:

$$\mu_i(\vec{x}_k) = \frac{1}{\sum_{j=1}^{c} \frac{d_i(\vec{x}_k)}{d_j(\vec{x}_k)}^{\frac{2}{m-1}}} \tag{5.6}$$

and

$$\vec{v}_i = \frac{\sum_{k=1}^{n} \mu_i(\vec{x}_k)^m \vec{x}_k}{\sum_{k=1}^{n} \mu_i(\vec{x}_k)^m} \tag{5.7}$$

5.2.2 Fuzzy c-means algorithm

The algorithm for implementing this method is as follows :

1. Fix c, $2 \leq c < n$.

   Fix m, $1 \leq m < \infty$.

   Fix ϵ as a small positive error number.

2. Initialize the cluster centers $\vec{v}_i(t)$.

3. Input data $X = \{x_1, \cdots, x_n\}$.

4. Update $\vec{v}_i(t+1)$ then calculate $\mu_i(\vec{x}_k)$.

5. calculate the error $e = \|\vec{v}_i(t+1) - \vec{v}_i(t)\|$

   if $e \leq \epsilon$ output μ_i and $\vec{v}_i$, stop.

   else goto step 3.

Chapter 6

Gradient Based Fuzzy C-Means Algorithm

One attempt to improve the FCM algorithm in this thesis is made by minimizing the objective function using one datum at a time instead of the whole data at once.

6.1 Derivation of the algorithm

Given one datum $\vec{x}_i$ and c clusters with centers at $\vec{v}_j$ $(j = 1, 2, \cdots, c)$, the objective function to be minimized is :

$$J_i = \mu_{1i}^2 ||\vec{v}_1 - \vec{x}_i|| + \mu_{2i}^2 ||\vec{v}_2 - \vec{x}_i|| + \cdots + \mu_{ci}^2 ||\vec{v}_c - \vec{x}_i|| \qquad (6.1)$$

with the constraint:

$$\mu_{1i} + \mu_{2i} + .. + \mu_{ci} = 1 \qquad (6.2)$$

The basic procedure of the gradient descent method is that starting from an initial center vector $\vec{v}$, the gradient $\nabla \vec{J}_i$ of the current objective function is computed [10]. The next value of v is obtained by moving in the direction of the negative gradient along the multidimensional error surface such that:

$$\vec{v}_{k+1} = \vec{v}_k - \eta \frac{\partial J_i}{\partial \vec{v}_k} \tag{6.3}$$

where :

$$\frac{\partial J_i}{\partial \vec{v}_k} = 2\mu_{ki}^2(\vec{v}_k - \vec{x}_i) \tag{6.4}$$

Equivalently,

$$\vec{v}_{k+1} = \vec{v}_k - \eta\mu_{ki}^2(\vec{v}_k - \vec{x}_i) \tag{6.5}$$

where η is a small learning constant.

For the membership grades, we set :

$$\frac{\partial J_i}{\partial \vec{\mu}} = 0 \tag{6.6}$$

and obtain :

$$\mu_{ki} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_i(\vec{x}_k)}{d_j(\vec{x}_k)}\right)^2} \tag{6.7}$$

6.2 Comparison between FCM and GBFCM

FCM and GBFCM both have an objective function which tries to minimize the distance between each center and the data with a membership grade reflecting the degree of their similarities with respect to other centers. On the other hand, they differ in the way they try to minimize it:

- In the FCM algorithm, all the data are present in the objective function, and the gradients are set to zero in order to obtain the equations necessary for minimization[2-5] .

- In the GBFCM, only one datum is present at a time. And only the gradients of the objective function with respect to the membership grades are set to zero. The gradients with respect to the centers are not set to zero. their negative values are used to minimize the objective function.

6.3 GBFCM algorithm

The algorithm for implementing this method is as follows :

```
Procedure main()
  Read $c,\epsilon,m$
    While ($error > \epsilon$ )
      $e \leftarrow 0$
      While (input file is not empty )
          Read one datum $\vec{x}_i$
          $\vec{v}_{k+1} \leftarrow \vec{v}_k - \eta\mu_{ki}^2(\vec{v}_k - \vec{x}_i)$
          $\mu_{ki} \leftarrow \frac{1}{\sum_{j=1}^{c}(\frac{d_i(\vec{x}_k)}{d_j(\vec{x}_k)})^2}$
          $e \leftarrow e + \vec{v}_i(k+1) - \vec{v}_i(k)$
      end while
      $error \leftarrow e$
    end while
  Output $\mu_i$ and $\vec{v}_i$.
end main()
```

6.4 Improving the speed of convergence

To improve the speed of convergence of the algorithm, the following two steps are added:

6.4.1 Moving the centers to their transitional directions

After every iteration, each initial center tends to move towards the actual center of some group of data.This center could be calculated using the classical center of weights formula:

$$\vec{v}_i = \frac{1}{n} \sum_{i=1}^{n} \vec{x}_i \tag{6.8}$$

where n is the number of data $\vec{x}_i$ which are closer to the center v_i. These data could be determined using the membership functions. The bigger the membership, the closer the datum is.

6.4.2 Decreasing the learning step

After one epoch calculate the difference of errors $dif = e(k+1) - e(k)$ if that difference is very small decrease the step η.

These two steps guaranteed a speed of convergence comparable to both Kohonen network and FCM algorithm. And for a large number of input data, GBFCM converges faster.

6.5 Convergence of GBFCM vs. FCM

To show the convergence of GBFCM vs. FCM, input data were created using the gaussian distribution. And the behavior of the error with respect to each iteration for both algorithm were recorded. The following graphs were obtained:

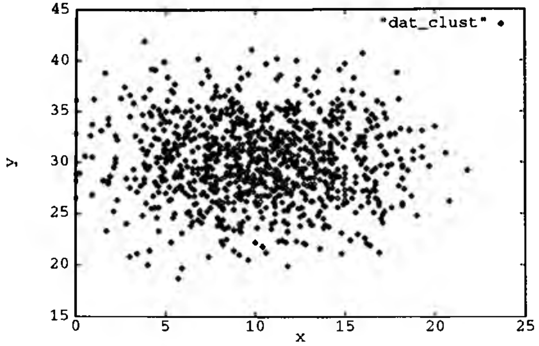

Figure 6.1: Input data1.

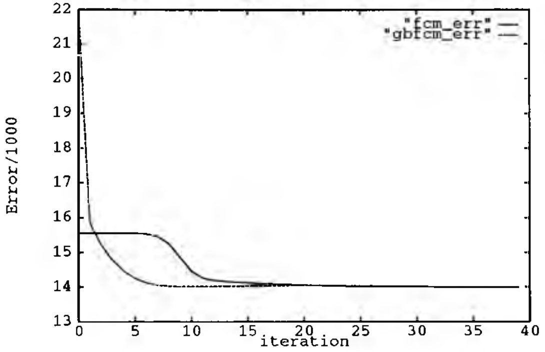

Figure 6.2: Error vs. iteration for 2 clusters for data1.

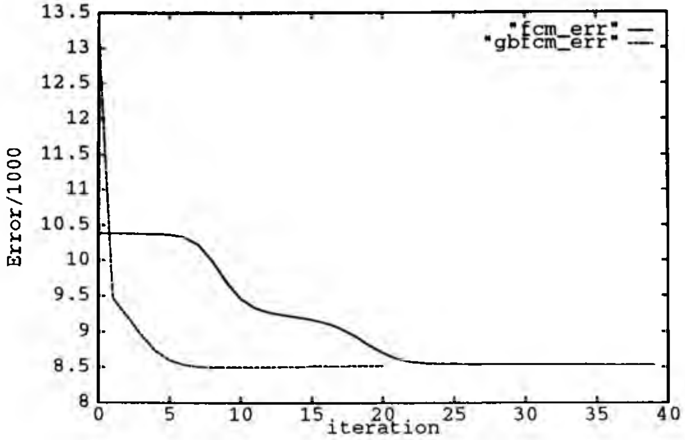

Figure 6.3: Error vs. iteration for 3 clusters for data1.

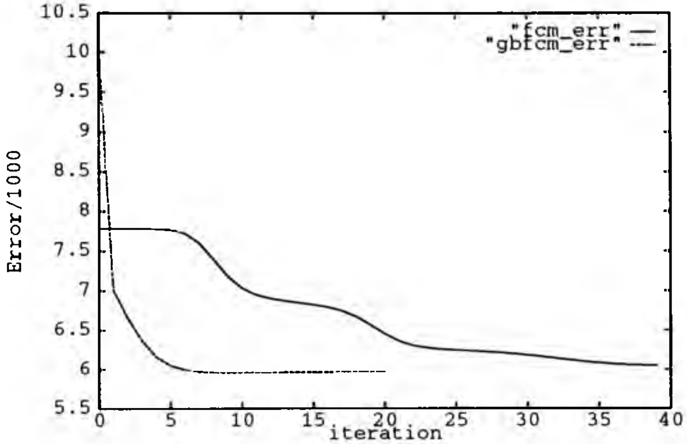

Figure 6.4: Error vs. iteration for 4 clusters for data1.

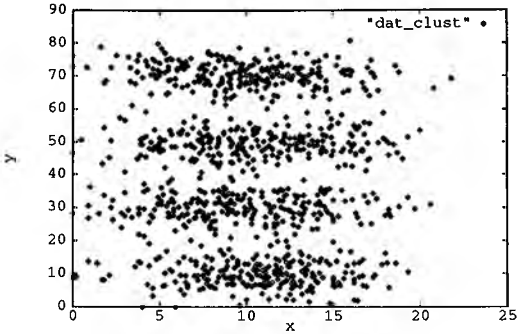

Figure 6.5: Input data2

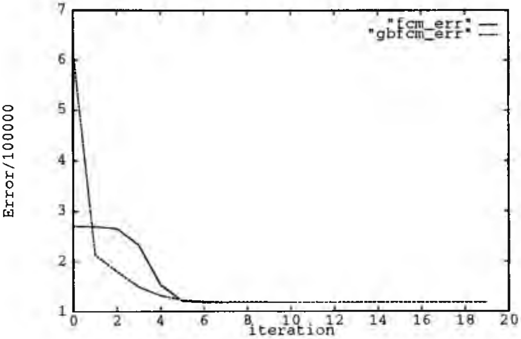

Figure 6.6: Error vs. iteration for 2 clusters for data2.

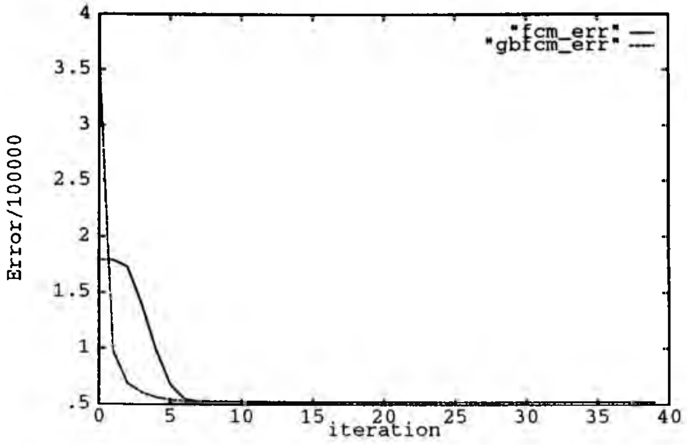

Figure 6.7: Error vs. iteration for 3 clusters for data2.

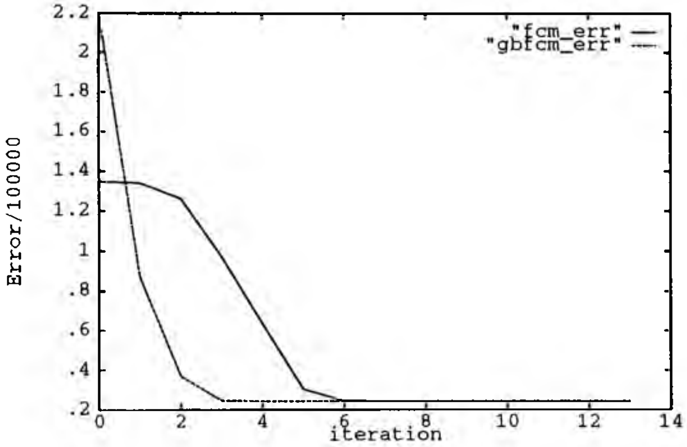

Figure 6.8: Error vs. iteration for 4 clusters for data2.

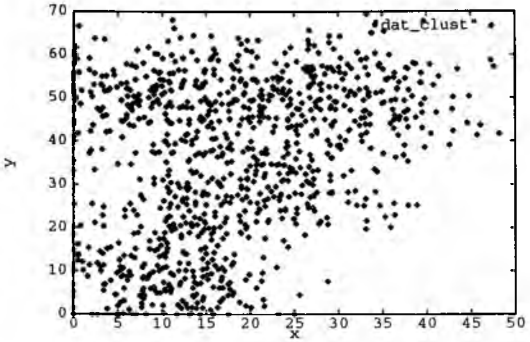

Figure 6.9: Input data3

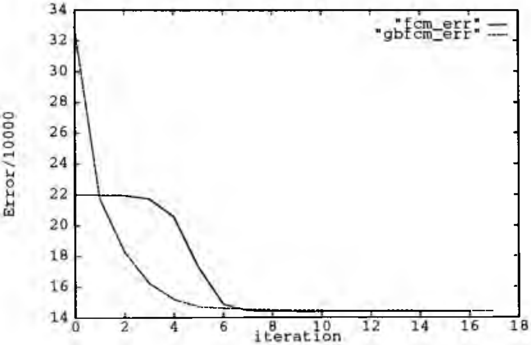

Figure 6.10: Error vs. iteration for 2 clusters for data3.

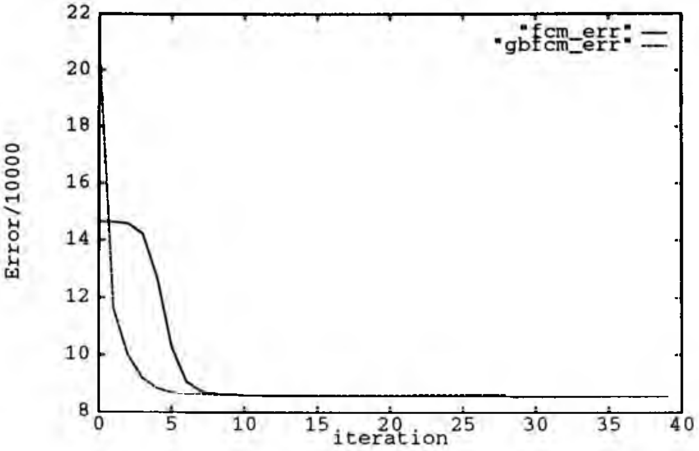

Figure 6.11: Error vs. iteration for 3 clusters for data3.

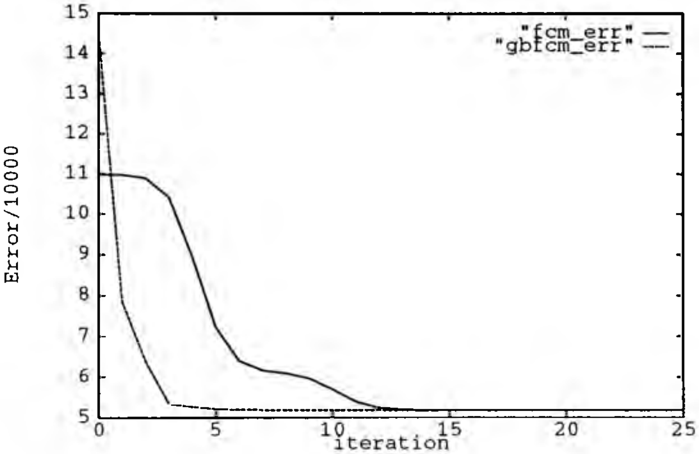

Figure 6.12: Error vs. iteration for 4 clusters for data3.

Chapter 7

Application Of GBFCM To Image Segmentation

Image segmentation is defined as subdivision of an image into its constituent parts or objects. This is achieved generally based on one of two basic properties of gray level values: discontinuity and similarity. The detection of discontinuities approach, is to partition an image based on abrupt changes in gray level. Therefore, detection of isolated points, lines and detection of edges fall into this category. On the other hand detection of similarities approach is based on thresholding, region growing, and region splitting and merging.

7.1 Detection of edges

By definition, an edge is the boundary between two regions with relatively distinct gray-level properties. For edge detection to work as a tool for image segmentation, the regions under investigation are sufficiently homogeneous so that the transition between two regions can be determined on the basis of gray level discontinuities alone. The basic tool used for edge detection is taking the derivative or gradient of the image. The gradient of an image f(x,y) at location (x,y) is the vector given by,

$$\vec{\nabla} f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{7.1}$$

It is important to mention a popular and important operator known as the Laplacian of the Gaussian. The Gaussian is defined as:

$$G(x,y) = \frac{1}{2\pi\sigma^2} exp(-\frac{x^2+y^2}{2\sigma^2}) \tag{7.2}$$

The Laplacain of the Gaussian is defined as:

$$\nabla^2 G(x,y) = \frac{\partial^2 G(x,y)}{\partial x^2} + \frac{\partial^2 G(x,y)}{\partial y^2} \tag{7.3}$$

Which will give:

$$\nabla^2 G(x,y) = \frac{1}{2\pi\sigma^4} \frac{x^2+y^2-2\sigma^2}{\sigma^2} exp(-\frac{x^2+y^2}{2\sigma^2}) \tag{7.4}$$

Applying the Laplacian of the Guassian to an image $f(x,y)$ is given by:

$$f(x,y) * \nabla^2 G(x,y) \tag{7.5}$$

where $*$ is the convolution operator.

This operator will be transformed into regular multiplication in the frequency domain.

$$F(f(x,y) * \nabla^2 G(x,y)) = F(f(x,y)F(\nabla^2 G(x,y)) \qquad (7.6)$$

where F is the Fourier transform.

The final output is the inverse of the Fourier transform obtained:

$$F^{-1}(F(f(x,y)F(\nabla^2 G(x,y))) \qquad (7.7)$$

7.2 Thresholding

Given an image, f(x,y), like a Japanese flag image, the gray-level histogram of this image clearly shows two distinct regions of dominance in gray level. For this image a threshold $T = 100$ will separate these two regions. For instance, any pixel whose gray level value is greater than T, is a background point, otherwise it is a object point. This kind of example is a simple one and in general such separation due to thresholding is very complicated and application dependent. Figures 7.1 and 7.2 show the Japanese flag and its histogram for illustration.

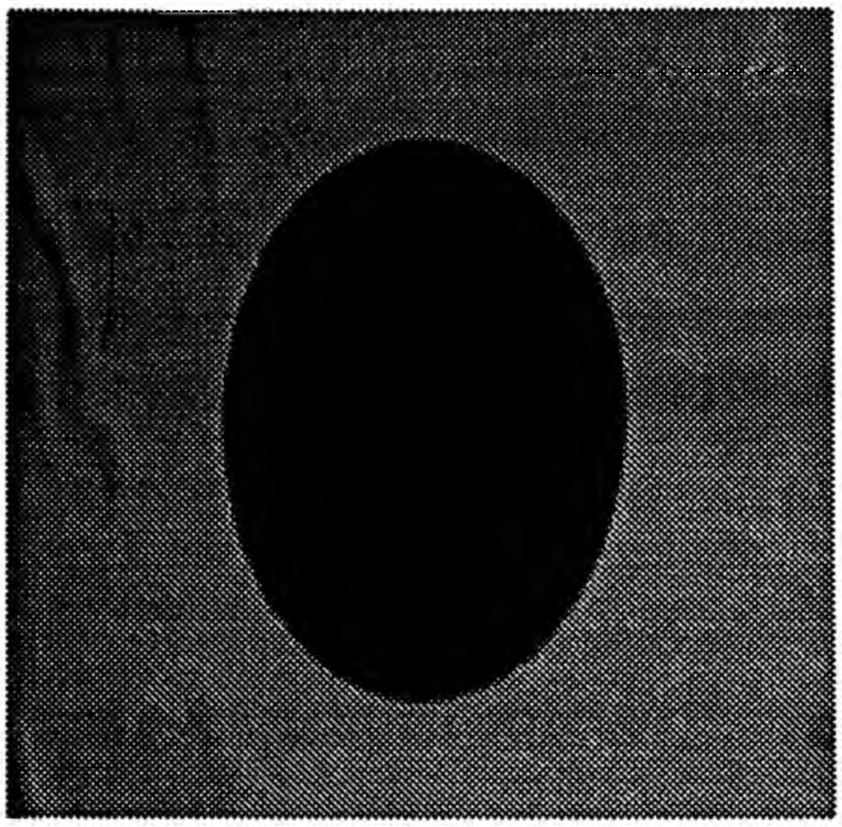

Figure 7.1: Image of the Japanese flag .

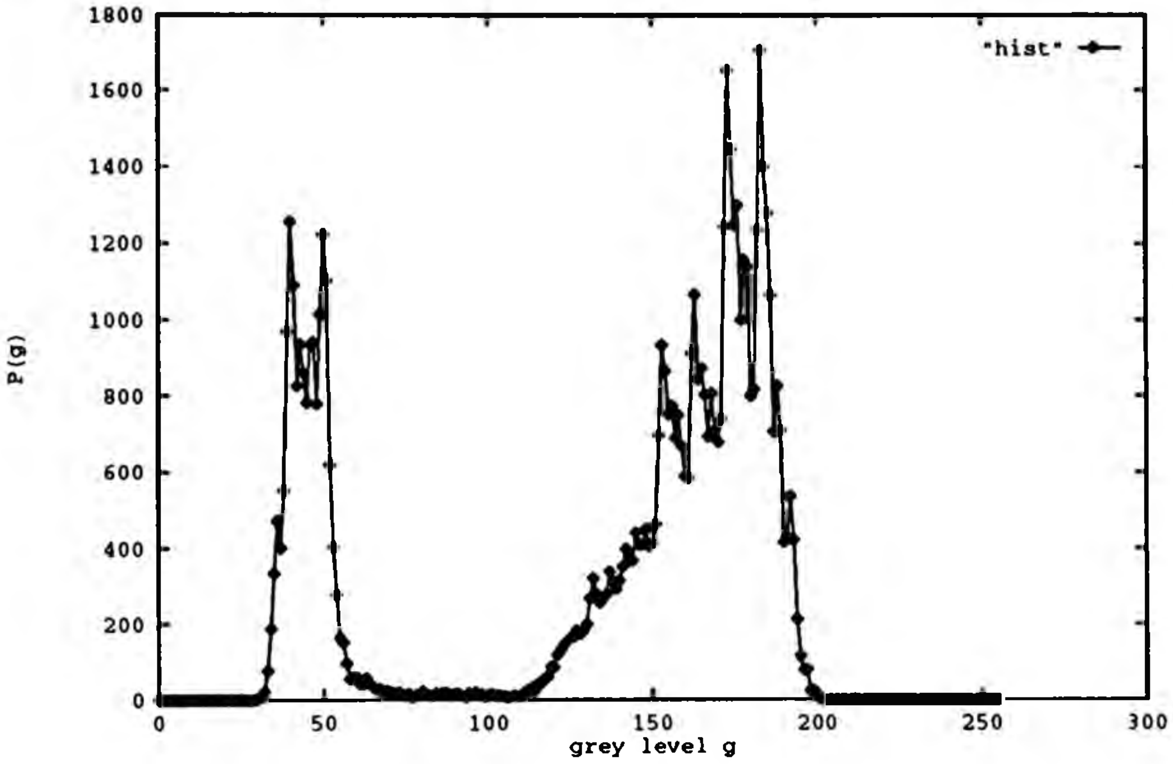

Figure 7.2: Histogram of the Japanese flag.

7.3 Region Splitting and Merging

The technique of region splitting and merging can be summarized as follows:

1. Split the image into four equal parts with no intersection.
2. If all the pixels in a region R_i possess approximately the same property then splitting of that region is complete and it is left alone. (The property of the pixels depends on the application, and on the type of segmentation involved).
3. If the pixels in a region R_i, do not possess approximately the same property, it is split again into four other disjoint regions. After this, the regions with similar properties are merged together to form one region.

7.4 Unsupervised image segmentation

The following steps are taken in applying the unsupervised image segmentation:

1. A sliding window of size M1 × N1 is selected and the window is moved through the image at displacements M2 horizontally and N2 vertically.

2. At each window position, certain window parameters are estimated from the data and stored as a vector.

3. Then the number of clusters in the set of parameter vectors are identified.

4. Finally, the image is segmented.

7.4.1 Implementation

For the implementation part, a 256 × 256 image was used, and a 16 × 16 sliding window was selected. In addition, the model parameters for each window were the mean and standard deviation. The formulas used to calculate the mean and standard deviation of all the pixels in a particular window are shown respectively below.

$$m = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{7.8}$$

$$\sigma = \frac{1}{N} \sum_{i=1}^{N} (x_i - m)^2 \tag{7.9}$$

where N is the total number of pixels in the sliding window. and x_i is the gray level value of pixel position i in the window. The data available were the mean and standard deviation of each window. The unsupervised learning is used to to cluster the data. The number of clusters obtained depends on the level of segmentation desired.

Furthermore, after the centers of the desired clusters were obtained, segmentation of the image took place as follows:

1. Each pixel was compared to all the centers obtained using the gaussian density function :

$$G_i(x) = \frac{1}{2\pi\sigma_i^2} exp(-\frac{(x-m_i)^2}{2\sigma_i^2}) \tag{7.10}$$

2. Each pixel was then assigned the value of the center of a particular cluster based on the following formula:

$$x = m_k \tag{7.11}$$

where

$$k = \max(G_i(x)) \qquad 0 < i \leq c \tag{7.12}$$

c is the number of clusters.

7.4.2 Results

Depending on the level of segmentation (number of clusters) chosen, the following images are obtained for 2,3,and 4 clusters.

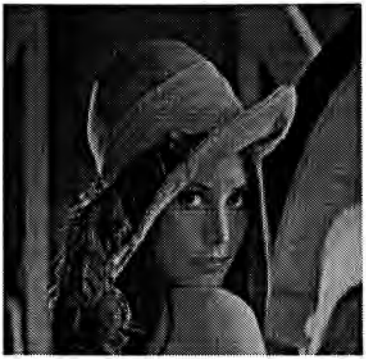

Figure 7.3: Image1 of Lenna .

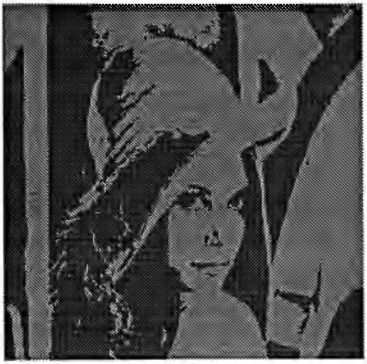

Figure 7.4: 2-level segmentation of image1.

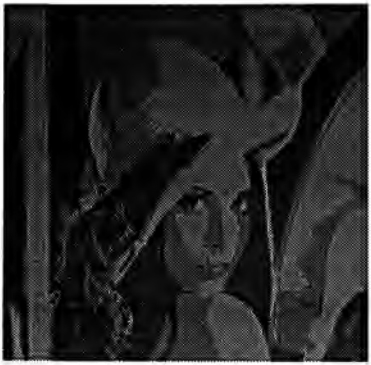

Figure 7.5: 3-level segmentation of image1.

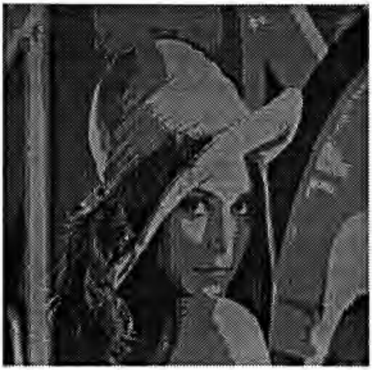

Figure 7.6: 4-level segmentation of image1.

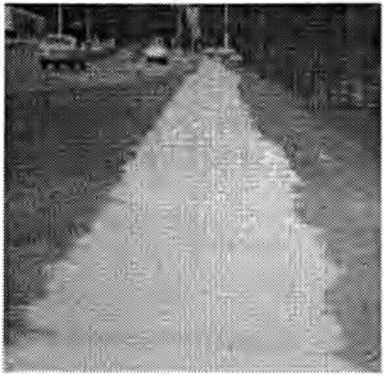

Figure 7.7: Image2 of a walkway.

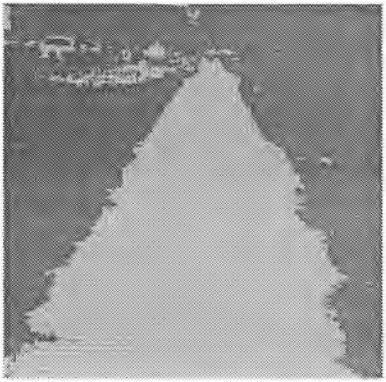

Figure 7.8: 2-level segmentation of image2.

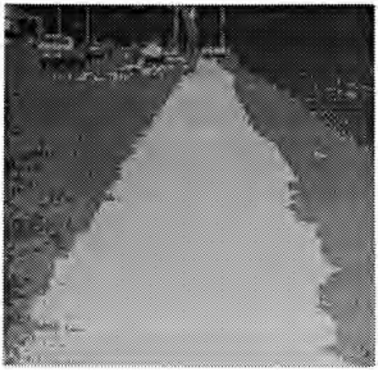

Figure 7.9: 3-level segmentation of image2.

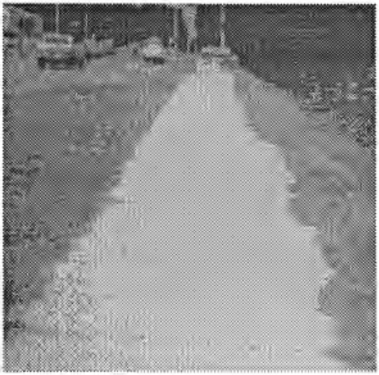

Figure 7.10: 4-level segmentation of image2.

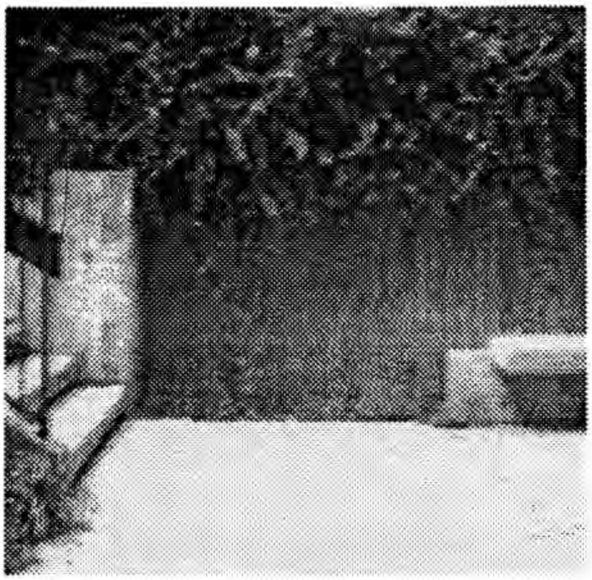

Figure 7.11: Image3 of a scene.

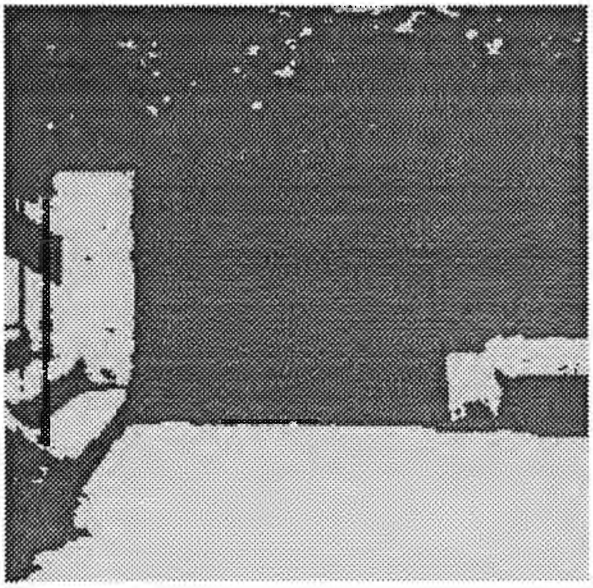

Figure 7.12: 2-level segmentation of image3.

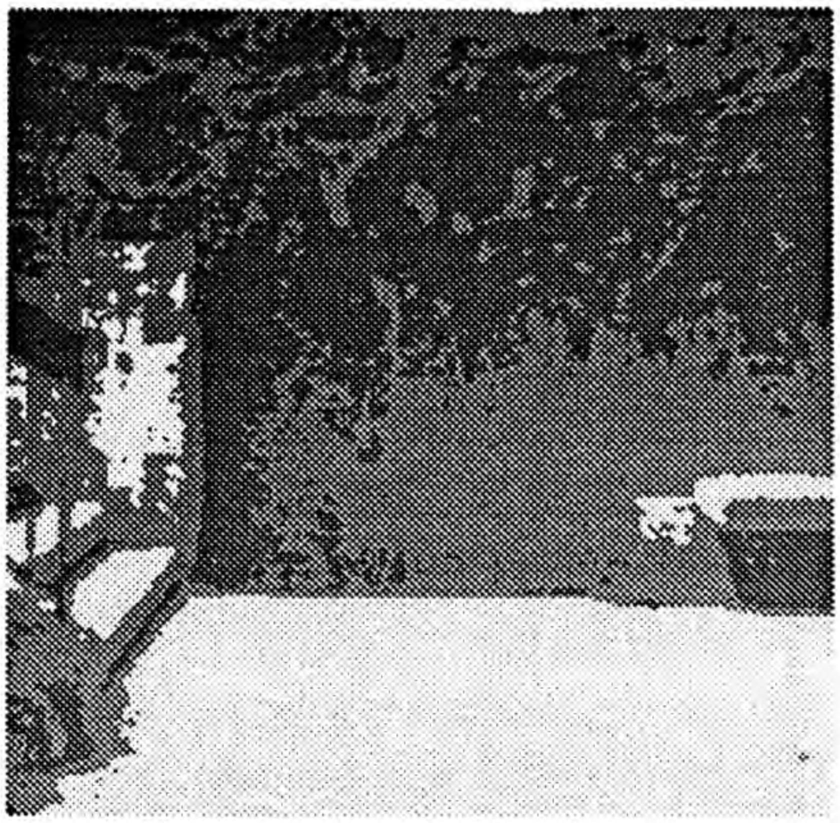

Figure 7.13: 3-level segmentation of image3.

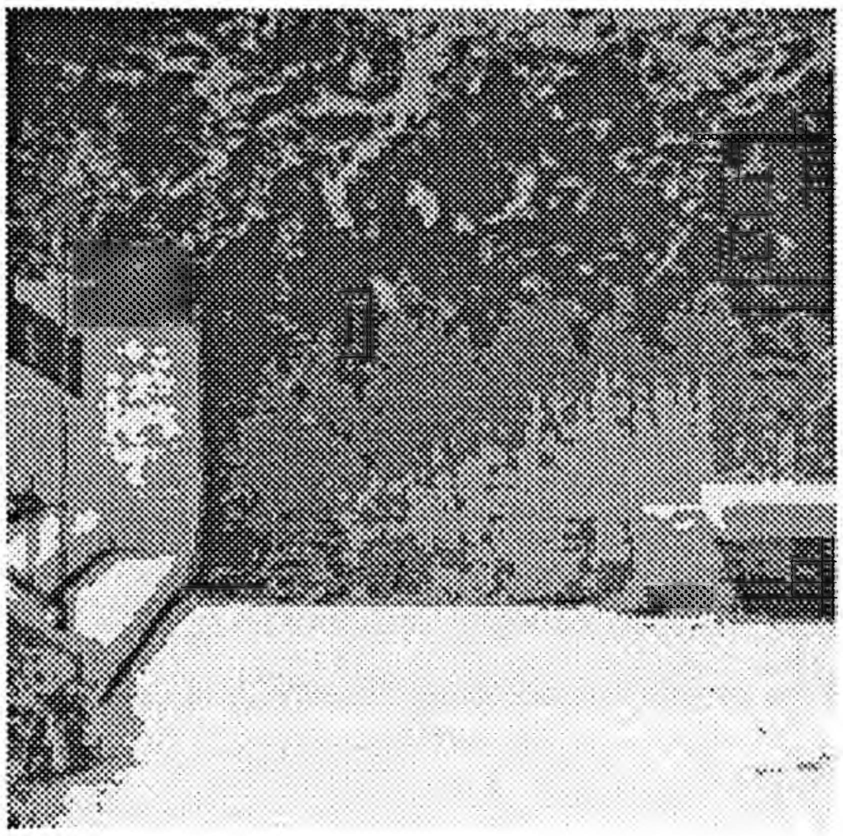

Figure 7.14: 4-level segmentation of image3.

7.5 Stopping GBFCM and FCM at different iterations

A 4-level segmentation of the Lenna image has been implemented. FCM and GBFCM were compared by stopping the algorithms at 2,4,6,and 8 iterations .

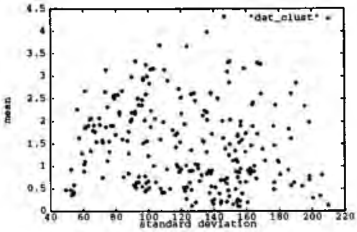

Figure 7.15: Mean vs. standard deviation for the Lenna image.

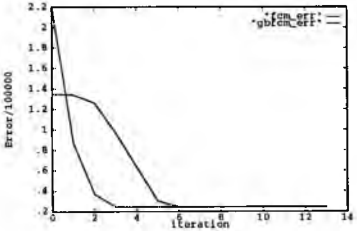

Figure 7.16: Error vs. iteration for 4 clusters.

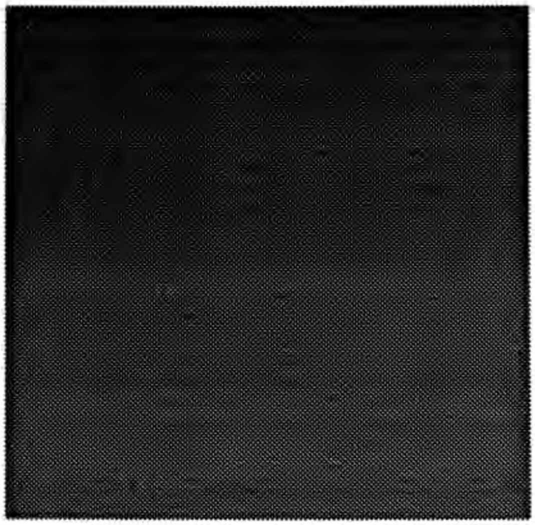

Figure 7.17: FCM 4 clusters - 2 iterations.

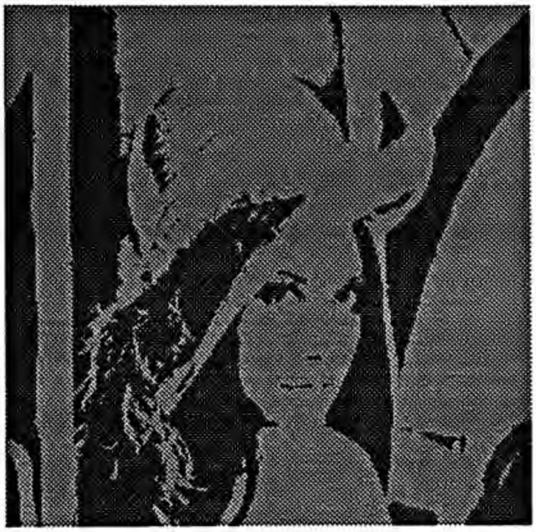

Figure 7.18: GBFCM 4 clusters - 2 iterations.

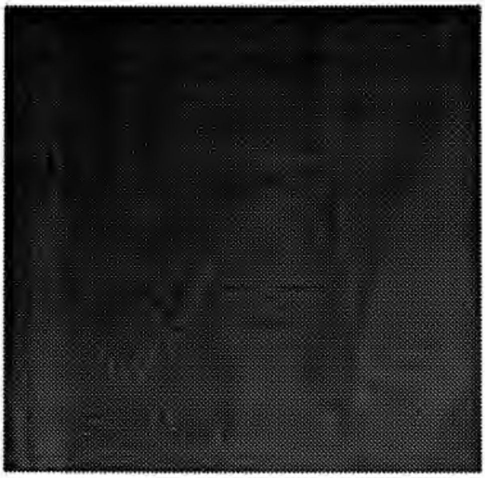

Figure 7.19: FCM 4 clusters - 4 iterations.

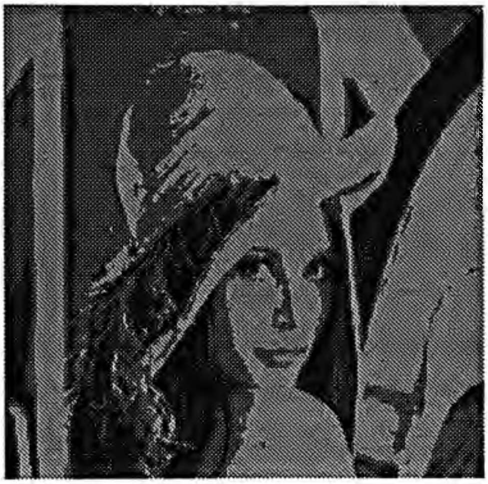

Figure 7.20: GBFCM 4 clusters - 4 iterations.

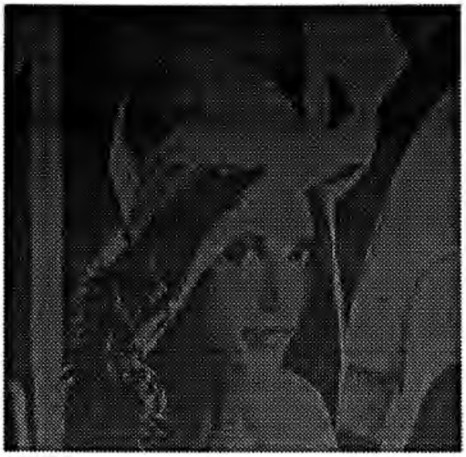

Figure 7.21: FCM 4 clusters - 6 iterations.

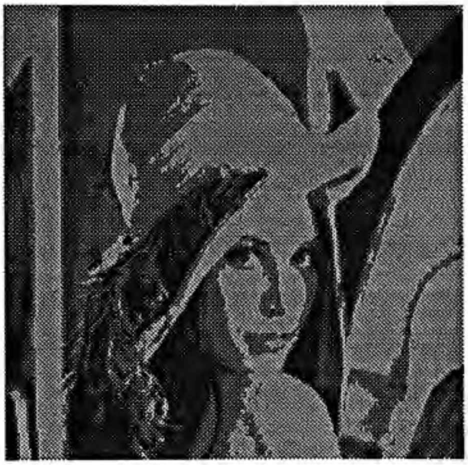

Figure 7.22: GBFCM 4 clusters - 6 iterations.

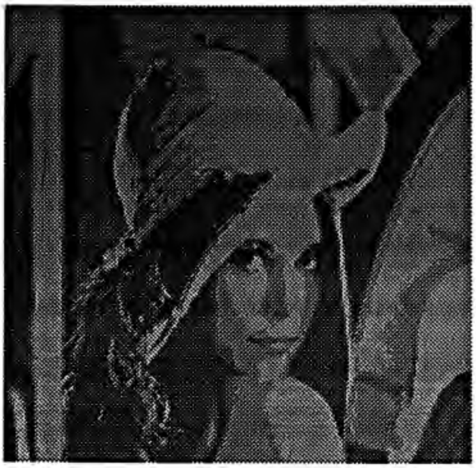

Figure 7.23: FCM 4 clusters - 8 iterations.

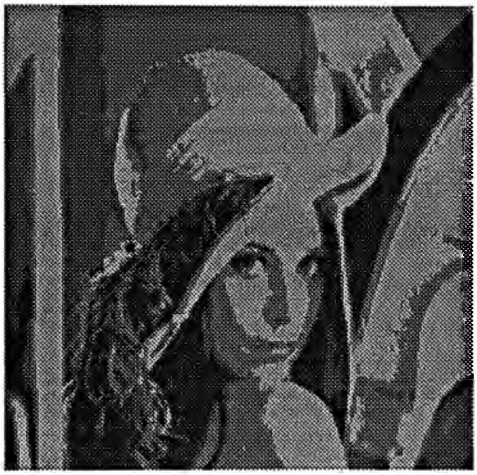

Figure 7.24: GBFCM 4 clusters - 8 iterations.

7.6 Discussion

Stopping the algorithms at different iterations,as shown in the above figures, is a strong tool which can be used to show the progress of an algorithm. Starting at 2 iterations,FCM algorithm couldnot yet distinguish between 2 regions (clusters) of the input data (figure 7.15). Only one region could be seen (figure 7.17). GBFCM algorithm,on the other hand, clustered the data into at least 2 regions (figure 7.18). Going to 4 iterations figure 4 shows that the FCM algorithm started to distinguish between 2 levels (at least 3 levels for the GBFCM algorithm). Increasing the number of iterations has the effect of clustering the data into the 4 regions desired (4-level segmentation) as shown in figures 7.23 and 7.24.

Chapter 8

Application Of GBFCM To Regression Analysis

Regression analysis is a statistical technique for investigating and modeling the relationship between variables. It is used to develop equations which summarize or describe the behavior of a set of data. To develop these equations, the following 2 steps should be taken:

1. Building a regression model.
2. Estimating the parameters of the model .

The data are used to estimate the unknown parameters of the system.

8.1 Least squares estimation of the parameters

A regression model could be given by the following relationship:

$$y = f(x, \beta) + \epsilon \tag{8.1}$$

β is a vector of parameters to be determined.

ϵ random vector with mean vector 0 and covariance matrix ξ

Regression analysis is defined as the search for the best function f which will fit a set of observed data. $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ The method of least squares is used to estimate the vector β. This vector will be estimated so that the sum of the squares of the differences between the observations y_i and the response of the model y is a minimum.Thus the least squares criterion is

$$L(\beta) = \sum_{i=1}^{n} (y_i - f(x_i, \beta) - \epsilon_i)^2 \tag{8.2}$$

The least squares estimators of β must satisfy:

$$\frac{\partial L}{\partial \beta} = 0 \tag{8.3}$$

which will give the value of the vector β.

8.2 Unsupervised regression analysis

Unsupervised regression analysis is a generalization of the regression discussed earlier in the following 2 ways:

1. More than a single model are assumed.

2. And at the same time $S = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ is not labeled.

Thus, instead of assuming that a single model can can account for all n pairs in S, the data can be drawn from c models.

$$y = f_i(x, \beta_i) + \epsilon_i \quad 1 \leq i \leq c \tag{8.4}$$

and for a given (x_k, y_k) it is not known which model can be applied.

8.3 Unsupervised learning - Objective function

Because the data set S is unlabeled, the problem is to assign to each pair of data in S labels (memberships) which will identify its degrees of belongings to each model (group) considered. This is called unsupervised learning where the memberships μ are interpreted as the importance or weight attached to the extent to which the model value $f_i(x, \beta_i)$ matches y_k. Crisp memberships would place all of the weight in the approximation of y_k by $f_i(x, \beta_i)$ on one class for each k. But fuzzy partitions can represent situations where a data point fits all the models to varying degrees.

8.3.1 Objective function

For one pair of data (x, y), the error function can be defined as:

$$E_i(\beta_i) = \|y - f_i(x, \beta_i)\|^2 \tag{8.5}$$

which measures the error in $f_i(x, \beta_i)$ as an approximation to y.

Considering all the c models, the objective function is:

$$J(\mu, \beta) = \sum_{i=1}^{c} \{(\mu_i)^2(E_i(\beta_i))\} \tag{8.6}$$

8.3.2 Estimating the parameters of the system

The minimization of $J(\mu, \beta)$ is done by :

$$\frac{\partial J(\mu, \beta)}{\partial \mu} = 0 \tag{8.7}$$

and

$$\beta_{k+1} = \beta_k - \eta \frac{\partial J(\mu, \beta)}{\partial \beta_k} \tag{8.8}$$

Using these 2 formulas, the parameters β and μ could be calculated (η is a small learning constant which decreases as learning proceeds).

8.3.3 Results

The following results are obtained for different inputs and different models.

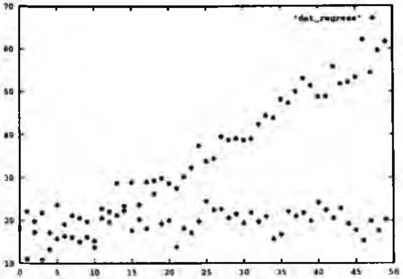

Figure 8.1: Input data1.

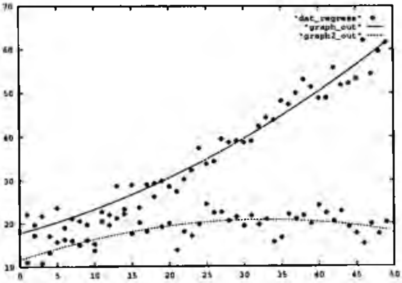

Figure 8.2: 2 clusters - 2 curves for data1.

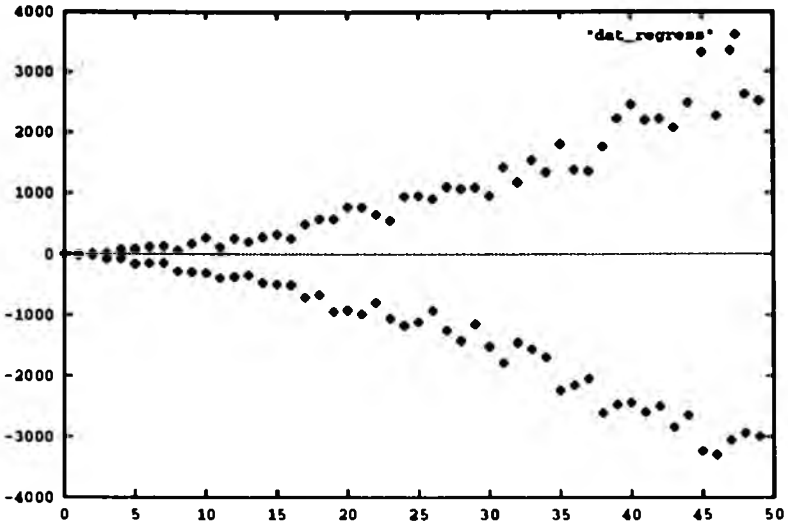

Figure 8.3: Input data2.

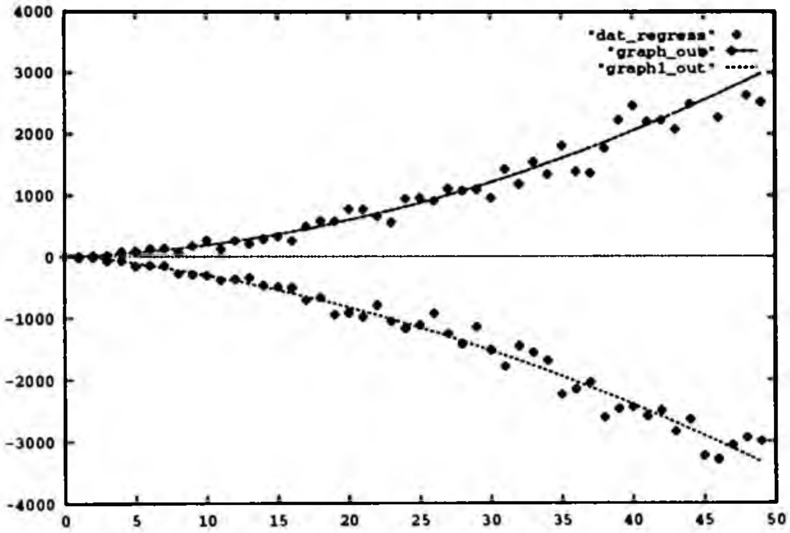

Figure 8.4: 2 clusters - 2 curves for data2.

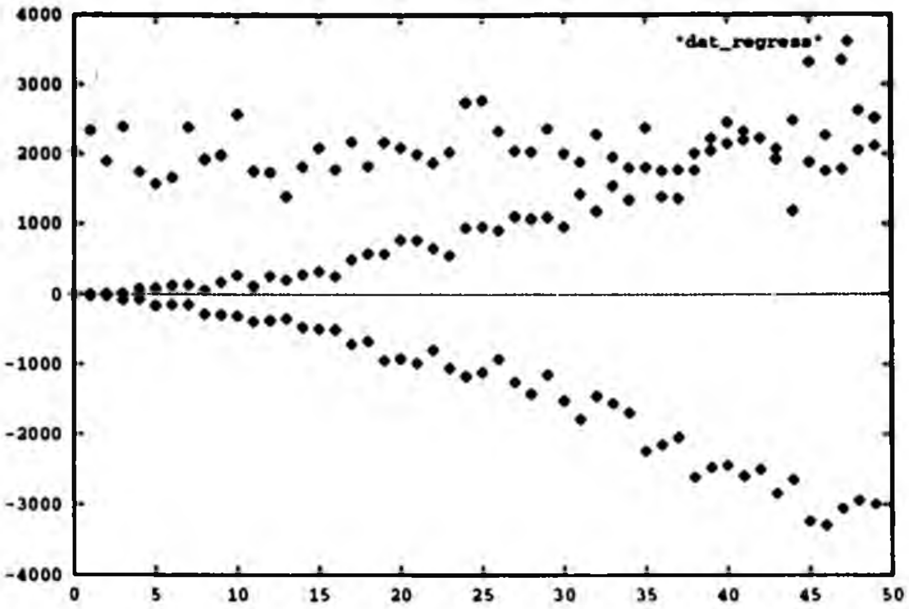

Figure 8.5: Input data3.

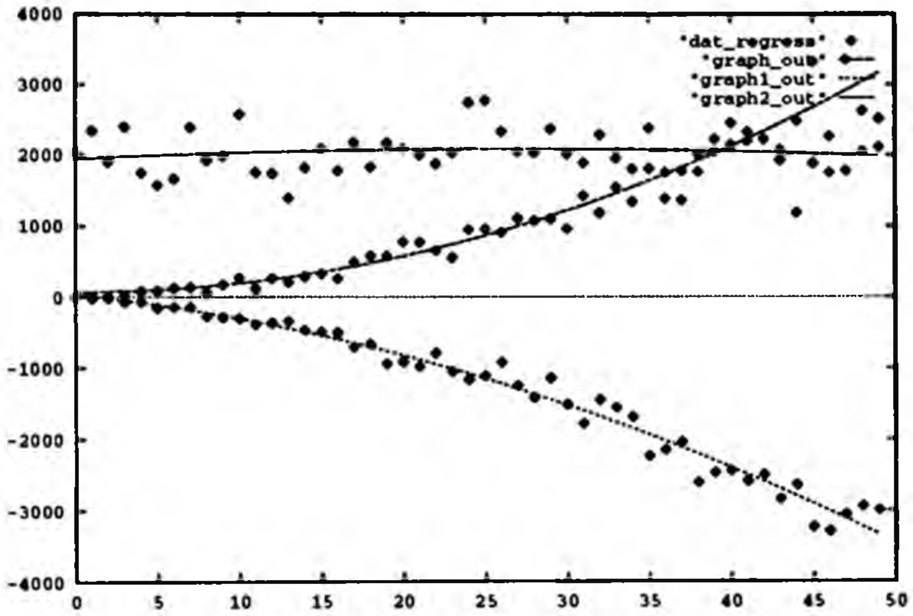

Figure 8.6: 3 clusters - 3 curves for data3.

Chapter 9

Application Of GBFCM To Reduction of features of the Fourier transform of the EMG signal

The source of the bioelectric signal is the single neural or muscular cell. The accumulated effects of all active cells in the vicinity produce an electric field which creates important electric signals used for the diagnosis of neural and muscular systems. Electromyographic (EMG) is the recording of the electric potential generated by the muscle. EMG signals are nonperiodic and possess all the characteristics of a random signal [16]. They can be anlysed using Fourier analysis which extracts features that characterize the important aspects of the signal.

9.1 Signal classification and recognition

The recognition of the signal is done by means of a classification process. The machine has a priori knowledge on the types (classes) of signals under consideration. An unknown signal is then classified into one of the known classes. The mechanism could be represented by:

1. The input to the recognition machine is a set of N measurements (samples of the signal).
2. A set of features is then extracted (Fourier transform).
3. And a classifier operates on the feature vector to perform the classification.

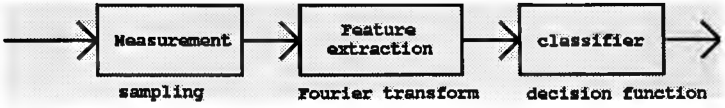

Figure 9.1: Signal classification.

9.2 Neural networks as a classifier

In this method :

- A neural network (NN)is used to classify the data.

- The number of classes desired is 3 :min,mid, and max types.

These represent low,medium,and high level muscle activity respectively. The first set of data was collected with the arm being in a resting position with very little firing of muscles. The second set was collected with arm muscles being semi-tense. The third set was collected with arm muscles being in a tense position. For every set Fourier transform is applied to the data collected .

9.3 Unsupervised learning

A different approach based on the use of the unsupervised learning which is capable of reducing the number of features obtained by the Fourier analysis. Then these reduced features will be used to train a NN with less number of neurons than the one used before.

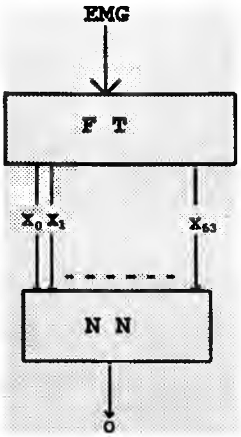

Figure 9.2: Neural network as a classifier.

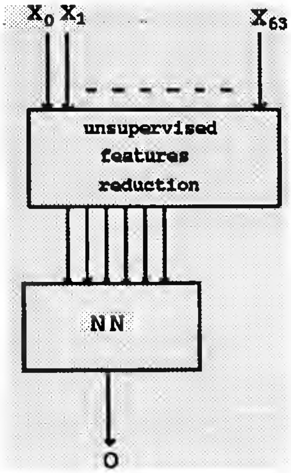

Figure 9.3: Reduction of features using GBFCM algorithm.

The following figures represent 2 patterns of the resting position:

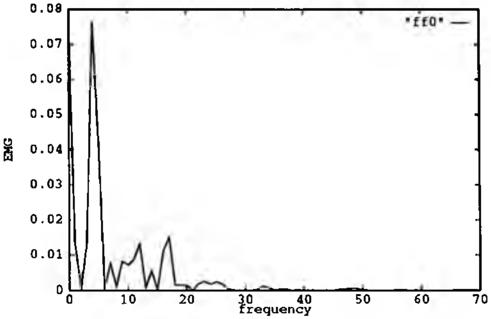

Figure 9.4: Pattern 1 of the resting position.

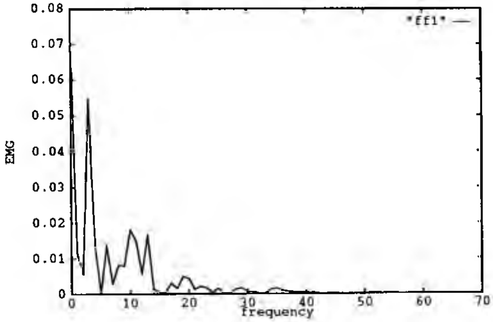

Figure 9.5: Pattern 2 of the resting position.

The following figures represent 2 patterns of the arm in the mid position:

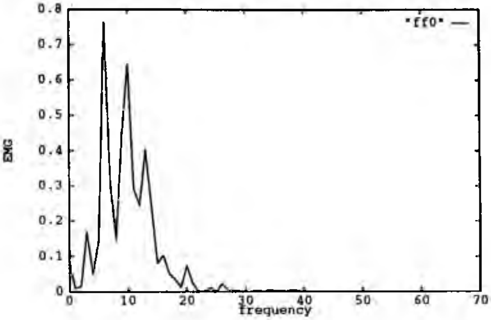

Figure 9.6: Pattern 1 of the mid position.

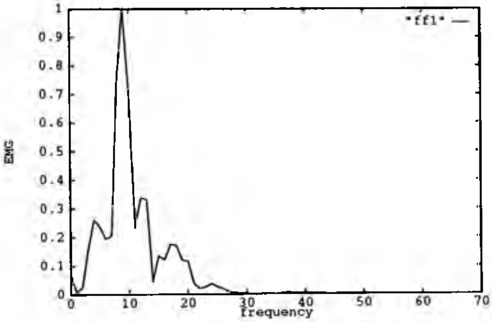

Figure 9.7: Pattern 2 of the mid position.

The following figures represent 2 patterns of the arm in the max position:

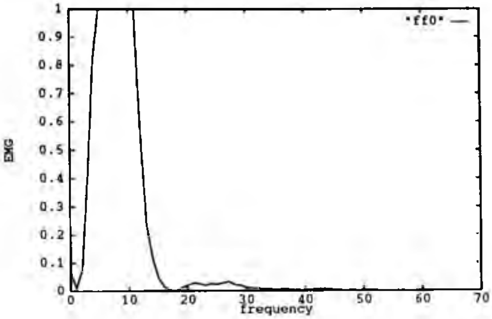

Figure 9.8: Pattern 1 of the max position.

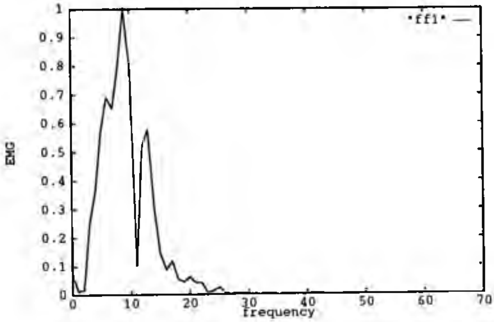

Figure 9.9: Pattern 2 of the max position.

Applying unsupervised learning to these patterns will give:

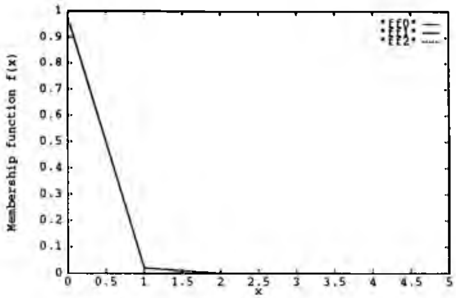

Figure 9.10: 6-point patterns for the min position.

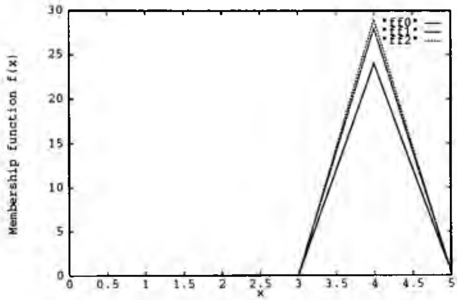

Figure 9.11: 6-point patterns for the mid position.

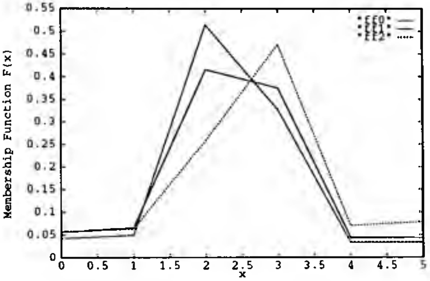

Figure 9.12: 6-point patterns for the max position.

9.4 Results

The following table is obtained after testing the network with untrained data:

Pattern	EMG (rest)	EMG (between1)	EMG (mid)
EMG(rest)	95.92%	4.08%	0%

Table 9.1: Untrained *rest* test data results.

Pattern	EMG(mid)	EMG (between2)	EMG(max)
EMG(mid)	91.81%	8.19%	0%

Table 9.2: Untrained *mid* test data results.

Pattern	EMG (between2)	EMG(max)
EMG(max)	7.7%	92.3%

Table 9.3: Untrained *max* test data results.

Chapter 10

Conclusions

A Gradient-Based FCM (GBFCM) algorithm is presented. This algorithm combines the characteristics of Kohonen network (Requiring one datum at a time in a gradient fashion) and Fuzzy c-means algorithm (fuzzy membership in $\{0, \cdots, 1\}$). Both FCM and GBFCM were applied to the same problem (4-level segmentation). The experiments performed in this thesis such as clustering of random data, image segmentation, regression analysis, and reduction of features of the EMG signal show that GBFCM converges faster giving competitive results when compared to FCM.

10.1 Future Work

GBFCM algorithm proceeds using the gradient descent method which could be improved in terms of speed of convergence and complexity of calculations. This could be done using the :

1. conjugate gradient method which is much faster than the gradient descent method.
2. genetic algorithm which doesn't require the calculation of any derivative.

Both of these methods could be used for future improvements.

Bibliography

[1] J.M.Zurada, *Artificial neural systems*, West 1992

[2] T. Kohonen," Learning Vector Quantization," Helsinki University of Technology, Laboratory of Computer and Information Science, Report TKK-F-A-601, 1986

[3] J.C.Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well separated clusters," *J. Cybern.*, 1973,(3):32-75.

[4] J.C.Bezdek, " A convergence theorem fo the fuzzy ISODATA clustering algorithms,"*IEEE trans. pattern Anal. Mach. int.*, 1980,(2):1-8. 1975,(24):835-838.

[5] Windham M.P.,"Cluster Validity for the Fuzzy c-means clustering algorithm,"*IEEE trans. pattern Anal. Mach. int.*, 1982,(4):357-363.

[6] I.Gath,A.B.Geva, “Unsupervised optimal fuzzy clustering,“ *IEEE trans. pattern Anal. Mach. int.*, 1989,(11):773-781.

[7] Rafael C. Gonzalez and Richard E. Woods,*Digital Image Processing,* New York: Addison-Wesley, 1992.

[8] Young Won Lim and Sang Uk Lee,” On the color image segmentation algorithm based on the thresholding and the Fuzzy c-means techniques,“ *Pattern Recognition*, vol.23,no.9,1990.

[9] J.C.Bezdek, *Pattern recognition with fuzzy objective function algorithms,* New York: Plenum,1981.

[10] B.S.Gottfried,J.weisman, *Introduction to optimization theory* , Prentice Hall 1973

[11] J.C.Bezdek,J.C.Dunn,“ Optimal fuzzy partition:A heuristic for estimating the parameters in a mixture of normal distributions,“ *IEEE Trans. Comput.*, 1975,(24):835-838.

[12] Xuanli l.x.,BEN G., “ A validity measure for fuzzy clustering,“ *IEEE trans. pattern Anal. Mach. int.*, 1991,(13):841-847.

[13] J. Hertz et al., *Introduction to the Theory of Neural Computation*, Reading, MA: Addison-Wesley, 1990

[14] G.J.Klir,T.A.Folger, ***Fuzzy sets, uncertainty, and information*** , Prentice Hall 1988

[15] Dong C. Park and Issam Dagher,"Gradient Based Fuzzy c-means (GBFCM) agorithm," *Proc. of IEEE World Congress on Computational Intelligence '94*,Orlando,June26-July2, 1994.

[16] Adrian Del Boca and Dong C. Park, " Myoelectric Signal Recognition using Artificial Neural Networks in Real Time ,"*Proc. of IEEE World Congress on Computational Intelligence '94*,Orlando,June26-July2, 1994.

[17] V.Tiel, *Convex Analysis*, Wiley 1984

[18] J. Zhang and J. W. Modestino,"A Model Fitting Approach To Cluster Validation," *IEEE Transactions On Pattern Analysis And Machine Intelligence,* Vol. 12, No. 10, October 1990.

[19] C. Bouman and B. Liu.,"Multiple Resolution Segmentation of Textured Images," *IEEE Transactions On Pattern Analysis And Machine Intelligence,* PAMI-13, 1991, 99-113.

[20] Haluk Derin, and Howard Eliot,"Modelling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields,"

IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol.PAMI-9, No. 1, January 1987.

[21] Hong Hai Nguyen and Paul Cohen,"Gibbs Random Fields, Fuzzy Clustering, and the Unsupervised Segmentation Of Textured Images," *Graphical Models And Image Processing,* Vol. 55, No. 1,January, pp. 1-19, 1993.

[22] Chee Sun Won And Haluk Derin."Unsupervised Segmentation of Noisy and Textured Images Using Markov Random Fields,"*Graphical Models And Image Processing,* Vol. 54, No. 4, July, pp. 308-328,1992.

[23] R.J.Hathaway and J.C.Bezdek,"Switching Regression models and Fuzzy Clustering,"*IEEE Transactions on Fuzzy Systems,* Vol. 1,No. 3,August 1993.

[24] R.D.De Veaux,"Mixtures of linear regression,"*Computational Statistics and Data Analysis,* Vol. 8,pp 227-245, 1989.

[25] M.Aitkin and G.T.Wilson,"Mixture models,outliers,and the EM algorithm,"*Technometrics,* Vol. 22,No. 3, 1980.

[26] R.E.Quandt,"A new approach to estimating switching regressions," *J. Amer. Stat. Ass.,* Vol. 67,pp. 306-310, 1972.

[27] R.E.Quandt and J.B.Ramsey,“Estimating mixtures of normal distributions and switching regressions,“ *J. Amer. Stat. Ass.,* Vol. 73,pp. 730-752, 1987.

[28] R.J.Hathaway and J.C.Bezdek,“Numerical convergence and interpretation of the fuzzy c- shells clustering algorithm,“ *IEEE Trans. Neural Networks,* Vol, 3,pp. 787-793,Sept. 1992.

[29] Sanford Weiberg,*Applied Linear Regression,* New York: John Wiley , 1980.

[30] Douglas C. Montgomery And Elizabeth A.Peck,*Introduction to Linear Regression Analysis,* New York:John Wiley , 1982.