Fuzzy Partitioning Using Real Coded Variable Length Genetic Algorithm for Pixel Classification

Ujjwal Maulik, Member, IEEE, *

Sanghamitra Bandyopadhyay, Member IEEE †

Abstract

The problem of classifying an image into different homogeneous regions is viewed as the task of clustering the pixels in the intensity space. Real-coded variable string length genetic fuzzy clustering with automatic evolution of clusters is used here for this purpose. The cluster centers are encoded in the chromosomes, and the Xie-Beni index is used as a measure of the validity of the corresponding partition. The effectiveness of the proposed technique is demonstrated for classifying different landcover regions in remote sensing imagery. Results are compared with those obtained using the well known fuzzy C-means algorithm.

Keywords: cluster validity, fuzzy clustering, pattern recognition, remote sensing imagery,

*Department of Computer Science, Kalyani Government Engineering College, Kalyani 741 235, INDIA, Email : ujjwal_maulik@kucse.wb.nic.in, ujjwal_maulik@yahoo.com

†Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, INDIA, Email: sanghami@isical.ac.in

variable string length genetic algorithm, Xie-Beni index.

1 Introduction

An important task in remote sensing applications is the classification of pixels in the images into homogeneous regions, each of which corresponds to some particular landcover type. This problem has often been modeled as a clustering problem [**?**, **?**]. However, in most of the cases, an assumption about the number of classes present in the image is made. In this article, we model the problem of unsupervised classification, or segmentation, of satellite image data as one of fuzzy clustering in the intensity domain of the different bands of the image, without making any *a priori* assumption about the number of clusters present in the data set.

The aim of any clustering technique is to evolve a partition matrix $U(X)$ representing a possible grouping of the given data set $X = \{x_1, x_2, \ldots, x_n\}$, into a number, say c, of clusters such that patterns in the same group are similar in some sense and patterns in different groups are dissimilar in the same sense. The partition matrix $U(X)$ of size $c \times n$ may be represented as $U = [u_{ik}]$, $1 \leq i \leq c; 1 \leq k \leq n$, where u_{ik} is the membership of pattern x_k to cluster C_i $(i = 1, \ldots, c)$.

Note that, in general, a pixel corresponds to an area of the land space, which may not necessarily belong to a single type of landcover. This in turn indicates that the pixels in a satellite image can be associated with a large amount of imprecision and uncertainty. Therefore, application of the principles of fuzzy set theory appears to be nat-

ural and appropriate in such domains. In fuzzy partitioning of the data, the following conditions hold on the partition matrix U (representing non-degenerate clustering): $0 < \sum_{k=1}^{n} u_{ik} < n, \quad \sum_{i=1}^{c} u_{ik} = 1, \quad \text{and} \quad \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik} = n.$

Fuzzy C-Means (FCM) [**?**] is a widely used technique that uses the principles of fuzzy sets to evolve a partion matrix $U(X)$ while minimizing the measure $\sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m D^2(v_i, x_k)$ where $D(v_i, x_k)$ represents the distance from point x_k $(k = 1, \ldots, n)$ to the center of ith cluster, v_i $(i = 1, \ldots, c)$, and m is the weighting coefficient. However, FCM has two major limitations: it requires the *a priori* specification of the number of clusters (c), and it often gets stuck at suboptimal solutions based on the initial configuration of the system. In this article, we attempt to overcome the above mentioned limitations of FCM by using the search capability of genetic algorithms to automatically evolve the fuzzy partitions of a data such that some measure of goodness of the partitions is optimized.

Genetic Algorithms (GAs) [**?**, **?**] are randomized search and optimization techniques guided by the principles of evolution and natural genetics. Genetic and other evolutionary algorithms have been earlier used for pattern classification [**?**, **?**], including clustering of data [**?**]-[**?**]. However, usually the number of clusters is assumed to be fixed *a priori* and/or the clusters are assumed to be crisp in nature. In contrast, here we attempt to automatically evolve the appropriate number of clusters as well as the fuzzy partitioning of the data. For this purpose, variable string length GAs, where different chromosomes in the same population may encode different number of clusters (and hence have different lengths), is used. The Xie-Beni index (XB-index) of the partitioning encoded in a chromosome is used to measure its fitness value. In order to tackle the concept of variable string lengths, the crossover and

the mutation operators are redefined accordingly.

Indian remote sensing (IRS) satellite images of partS of the cities of Calcutta and Mumbai have been used for demonstrating the effectiveness of the developed genetic fuzzy clustering technique in automatically segmenting the images into an unknown number of regions. From the ground truth available for the images, the effectiveness of the method in automatically identifying the different landcover types present in the images has been verified. The superiority of the proposed technique, as compared to the well known FCM algorithm, is demonstrated both quantitatively and qualitatively.

2 The Fuzzy Clustering Methodology

In this section, we describe the use of variable string length genetic algorithms (VGAs) for automatically evolving the near-optimal $c \times n$ non-degenerate fuzzy partition matrix U^*. The set $\mathcal{U}$ of all possible non-degenerate partition matrices is represented as $\mathcal{U} = \{U \in I\!R^{c \times n} | \sum_{i=1}^{c} u_{ik} = 1, \;\; 0 < \sum_{k=1}^{n} u_{ik} < n, \text{ and } u_{ik} \in [0,1]\}$. For the purpose of this article, we consider the best partition to be the one that corresponds to the minimum value of the Xie-Beni index ($XB(U,V,X)$), where U, V and X are the partition matrix, set of cluster centers and the input data set respectively. In other words, the best partition matrix U^* can be represented as

$$U^* \in \mathcal{U} \text{ and } XB(U^*,V^*,X) = \min_{U_i \in \mathcal{U}} XB(U_i,V_i,X), \tag{1}$$

where V^* represents the set of cluster centers corresponding to U^*. Here both the number of clusters as well as the appropriate fuzzy clustering of the data is evolved simultaneously

using the search capability of genetic algorithms.

In GAs, the parameters of the search space are encoded in the form of strings (called *chromosomes*). A collection of such strings is called a *population*. Initially a random population is created, which represents different points in the search space. An *objective/fitness* function is associated with each string that represents the degree of *goodness* of the solution encoded in the string. Based on the principle of survival of the fittest, a few of the strings are selected and each is assigned a number of copies that go into the mating pool. Biologically inspired operators like *crossover* and *mutation* are applied on these strings to yield a new population. The process of selection, crossover and mutation continues for a fixed number of generations or till a termination condition is satisfied.

For the purpose of clustering, each chromosome encodes a possible partitioning of the data, the goodness of the which is computed as a function of an appropriate cluster validity index. This index must be optimized in order to obtain the best partitions. Since the number of clusters is considered to be variable, the string lengths of different chromosomes in the same population are allowed to vary. As a consequence, the crossover and mutation operators are suitably modified in order to tackle the concept of variable length chromosomes. The technique is described below in detail.

String Representation and Population Initialization: In VGA based fuzzy clustering, the chromosomes are made up of real numbers which represent the coordinates of the centers of the partitions. If chromosome i encodes the centers of M_i clusters in N dimensional space then its length l_i is taken to be $N * M_i$. For example, in three dimensional space, the chromosome $<$ 12.3 1.4 5.6 22.1 0.01 10.2 0.0 5.3 15.3 13.2 10.2 7.5 $>$ encodes 4 cluster

centers, (12.3, 1.4, 5.6), (22.1, 0.01, 10.2), (0.0, 5.3, 15.3) and (13.2, 10.2, 7.5). Each center is considered to be indivisible. Each string i in the population initially encodes the centers of a number, M_i, of clusters, such that $M_i = (rand() \mathrm{mod} M^*) + 2$. Here, $rand()$ is a function returning an integer, and M^* is a soft estimate of the upper bound of the number of clusters. The number of clusters will therefore range from two to $M^* + 1$. Note that M^* is used only for the generation of the initial population. The actual number of clusters in the data set is not related to M^*, and may be any number greater than, equal to or less than M^*. The M_i centers encoded in a chromosome are randomly selected distinct points from the data set.

Fitness Computation: The fitness of a chromosome indicates the degree of goodness of the solution it represents. In this article we use the Xie-Beni (XB) cluster validity index [**?**] for this purpose. The XB index is defined as a function of the ratio of the total variation σ to the minimum separation *sep* of the clusters. Here σ and *sep* can be written as

$$\sigma(U, V; X) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^2 D^2(v_i, x_k), \tag{2}$$

and

$$sep(V) = \min_{i \neq j} \{||v_i - v_j||^2\}, \tag{3}$$

where $||.||$ is the Euclidean norm, and $D(v_i, x_k)$, as mentioned earlier, is the distance between the pattern x_k and the cluster center v_i. The XB index is then written as

$$XB(U, V; X) = \frac{\sigma(U, V; X)}{n \ sep(V)} = \frac{\sum_{i=1}^{c} (\sum_{k=1}^{n} u_{ik}^2 D^2(v_i, x_k))}{n(\min_{i \neq j} \{||v_i - v_j||^2\})}. \tag{4}$$

Note that when the partitioning is compact and good, value of σ should be low while *sep* should be high, thereby yielding lower values of the Xie-Beni (XB) index. The objective is therefore to minimize the XB index for achieving proper clustering.

Given a chromosome, the centers encoded in it are first extracted. Let the chromosome encode c centers, and let these be denoted as $v_1, v_2, \ldots, v_c$. The membership values u_{ik}, $i = 1, 2, \ldots, c$ and $k = 1, 2, \ldots, n$ are computed as follows [**?**]:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{D(v_i, x_k)}{D(v_j, x_k)}\right)^{\frac{2}{m-1}}}, \quad \text{for} \quad 1 \leq i \leq c; \quad 1 \leq k \leq n, \tag{5}$$

where $D(v_i, x_k)$ and $D(v_j, x_k)$ are as described earlier. m is the weighting coefficient. (Note that while computing u_{ik} using Eqn. 5, if $D(v_j, x_k)$ is equal to zero for some j, then u_{ik} is set to zero for all $i = 1, \ldots, c$, $i \neq j$, while u_{jk} is set equal to one.) The corresponding XB index is computed as in Eqn. 4. The fitness function for a chromosome is then defined as $\frac{1}{XB}$. Note that maximization of the fitness function will ensure minimization of the XB index. Subsequently, the centers encoded in a chromosome are updated using the following equation [**?**]

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m}, \quad 1 \leq i \leq c. \tag{6}$$

Selection: Conventional proportional selection is applied on the population of strings. Here, a string receives a number of copies that is proportional to its fitness in the population. We have used the roulette wheel strategy for implementing the proportional selection scheme.

Crossover: For the purpose of crossover, the cluster centers are considered to be indivisible, i.e., the crossover points can only lie in between two clusters centers. The crossover operator, applied stochastically with probability μ_c, must ensure that information exchange takes place in such a way that both the offspring encode the centers of at least two clusters. For this, the operator is defined as follows : Let parent chromosomes P_1 and P_2 encode M_1 and M_2 cluster centers respectively. τ_1, the crossover point in P_1, is generated as $\tau_1 = rand() \mathrm{mod} M_1$. Let

τ_2 be the crossover point in P_2, and it may vary in between $[LB(\tau_2), UB(\tau_2)]$, where $LB()$ and $UB()$ indicate the lower and upper bounds of the range of τ_2 respectively. $LB(\tau_2)$ and $UB(\tau_2)$ are given by

$$LB(\tau_2) = \min[2, \max[0, 2 - (M_1 - \tau_1)]], \tag{7}$$

$$\text{and} \quad UB(\tau_2) = [M_2 - \max[0, 2 - \tau_1)]]. \tag{8}$$

Therefore τ_2 is given by

$$\begin{aligned} \tau_2 &= LB(\tau_2) + rand()\text{mod}(UB(\tau_2) - LB(\tau_2)) && \text{if } UB(\tau_2) >= LB(\tau_2), \\ &= 0 && \text{otherwise.} \end{aligned}$$

It can be verified by some simple calculations that if the crossover points τ_1 and τ_2 are chosen according to the above rules, then none of the offspring generated would have less than two clusters.

Mutation: Each gene position of a chromosome is subjected to mutation with a fixed probability μ_m, resulting in the overall perturbation of the chromosome. A number δ in the range [0, 1] is generated with uniform distribution. If the value at a gene position is v, after mutation it becomes $(1 \pm 2 * \delta) * v$, when $v \neq 0$, and $\pm 2 * \delta$, when $v = 0$. The '+' or '-' sign occurs with equal probability. Note that because of mutation more than one cluster center may be perturbed in a chromosome.

Termination: In this paper, we have executed the algorithm for a fixed number of generations. Moreover, the elitist model of GAs has been used, where the best string seen so far is stored in a location within the population. The best string of the last generation provides the solution to the clustering problem.

3 Results

This section provides a description of the data set and the experimental results obtained on application of the above mentioned genetic fuzzy clustering technique for segmenting two remote sensing satellite images of parts of the cities of Calcutta and Mumbai. The parameters of the algorithm are as follows: population size is equal to 20, crossover and mutation probabilities are kept to be 0.8 and 0.01 respectively, $M^* = 20$ and the weighting coefficient $m = 2.0$. The algorithm is executed for a maximum of 100 iterations.

IRS Image of Calcutta

The data used here was acquired from Indian Remote Sensing Satellite (IRS-1A) [**?**] using the *LISS-II* sensor that has a resolution of 36.25m×36.25m. The image is contained in four spectral bands namely, blue band of wavelength 0.45 - 0.52 μm, green band of wavelength 0.52 - 0.59 μm, red band of wavelength 0.62 - 0.68 μm, and near infra red band of wavelength 0.77 - 0.86 μm.

Fig. 1 shows the Calcutta image in the near infra red band. Some characteristic regions in the image are the river *Hooghly* cutting across the middle of the image, several fisheries observed towards the lower-right portion, a township, *SaltLake*, to the upper-left hand side of the fisheries. This township is bounded on the top by a canal. Two parallel lines observed towards the upper right hand side of the image correspond to the airstrips in the *Dumdum* airport. Other than these there are several water bodies, roads etc. in the image.

The genetic fuzzy clustering technique automatically provided four clusters for this data

(Fig. 2). From our ground knowledge, we can infer that these four clusters correspond to the classes turbid water (TW), pond water (PW), concrete (Concr.) and open space (OS). It may be noted that the water class has been differentiated into turbid water (the *Hooghly*) and pond water (fisheries etc.) because of a difference in their spectral properties. Here, the class turbid water contains sea water, river water etc., where the soil content is more than that of pond water. *SaltLake* township has come out partially as classes concrete and open space, which appears to be correct, since this particular region is known to have several open spaces. The canal bounding *SaltLake* from the upper portion has also been correctly classified as PW. The airstrips of *Dumdum* airport has again been classified correctly as belonging to the class concrete. Presence of some small areas of PW beside the airstrips is also correct since these correspond to the several ponds that dot the region. The predominance of concrete on both sides of the river, particularly towards the bottom of the image is also correct. This region corresponds to the central part of the city of Calcutta.

In order to demonstrate the performance of the genetic clustering scheme quantitatively, we provide the variation of the best and average values of the XB-index with the number of generations in Fig. 3. As can be seen from the figure, the best value of the XB index is attained at around generation 25, after which the value does not change anymore. (It may be noted that because of elitism, the best value of XB-index can only decrease or remain the same with increase in the number of generations.) The average value of the XB index, on the other hand, shows frequent variation, although the general trend is towards reducing this as well. Note that with the introduction of variable string lengths, the diversity of the population remains high even after a significant number of generations which is reflected

correctly in the continuous variation of the average XB value.

Fig. 4 shows the variation of the XB-index with the number of clusters when the well-known FCM algorithm is used as the underlying clustering technique. If c^* is the number of clusters provided by the proposed clustering scheme, then the number of clusters in FCM is varied from $c^* - 3$ to $c^* + 3$. In this case, since $c^* = 4$, the corresponding range is taken to be 2 to 7 (since one cluster is not practically meaningful). As can be seen from the figure, the minimum value of the XB index is again obtained for four clusters with the FCM algorithm. However, the corresponding value as obtained with the genetic clustering scheme (denoted by '*' in the figure) is found to be still smaller. This not only shows that the genetic scheme found the correct number of clusters, but also demonstrates its superiority over the FCM technique, which often gets stuck at sub-optimal solutions. For the purpose of demonstration we provide Fig. 5 which plots the values of the XB-index for ten different runs of the FCM algorithm. As can be seen, out of the ten runs, the best value of the index is obtained in only two runs. Even in these cases, the value is worse than that obtained using the proposed technique as is evident from Fig. 4.

Fig. 6 shows the Calcutta image partitioned using 4 clusters using the FCM algorithm. As can be seen, the river *Hooghly* as well as the city region has been incorrectly classified as belonging to the same class. Therefore, we have labelled this region as TW+Concr. Again the entire *SaltLake* region, which we know to have both concrete and open space, has gone to only one class. Therefore, we have put the corresponding label as OS1 + Concr. In addition the class corresponding to several other open spaces in the image is labelled as OS2. Therefore we can conclude that although some regions, viz., fisheries, canal bounding

SaltLake, parts of the airstrip etc., have been correctly identified, a significant amount of confusion is evident in the FCM clustering result.

IRS Image of Mumbai

As for the Calcutta image, the IRS image of Mumbai was also obtained using the LISS-II sensor. It is available in four bands, viz., blue, green, red and near infra-red. Fig. 7 shows the *IRS* image of a part of Mumbai in the near infra red band. As can be seen, the elongated city area is surrounded on three sides by the Arabian sea. Towards the bottom right of the image, there are several islands, including the well known *Elephanta islands*. The dockyard is situated on the south eastern part of Mumbai, which can be seen as a set of three finger like structure.

The result of the application of the proposed clustering technique on the Mumbai image is shown in Fig. 8. The method automatically yielded seven clusters. We have labelled the different clusters, concrete (Concr.), open spaces (OS1 and OS2), vegetation (Veg), habitation (Hab) and turbid water (TW1 and TW2), based on the ground information available with us. Here, the class habitation refers to the regions which have concrete structures and buildings, but with relatively lower density than the class Concr. Thus these two classes share common properties. From the result it can be seen that the large water body of Arabian sea has been distinguished into two classes which we named TW1 and TW2. It has been observed earlier [**?**], and is also evident from Fig. 7, that the sea water has two distinct regions with different spectral properties. Hence the clustering result providing two partitions for this region is expected. The islands, dockyard, several road structures have mostly been correctly identified in the image. Within the islands, as expected, there is a predominance

of open space and vegetation. The southern part of the city, which is heavily industrialized, has been classified as primarily belonging to habitation and concrete. Some confusion within these two classes, viz., Hab and Concr, is observed (as reflected in the corresponding label); but this may be expected since, as mentioned earlier, these two classes are somewhat similar. The results obtained, for both the Calcutta and Mumbai images are quite encouraging, since the technique has managed to automatically discriminate the classes without any sort of *a priori* knowledge about the data or the number of clusters.

The variation of the best and average values of the XB-index with the number of generations for the Mumbai image was similar to that observed for the Calcutta image (and is not included here). The best value of the index was obtained quite early, at around generation thirty four. The variation of the XB-index with the number of clusters when the well-known FCM algorithm was used as the underlying clustering technique was also similar to that observed for the Calcutta image (and has been omitted here for brevity). In this case, since $c^* = 7$, the corresponding range for the number of clusters in FCM was taken to be 4 to 10. The minimum value of the XB index was found for 7 clusters, although, as earlier, the corresponding value as obtained with the genetic clustering scheme was still smaller. This once again confirmed the superiority of the proposed technique. Fig. 9 demonstrates the Mumbai image clustered using the FCM technique. As can be seen, the water of the Arabian sea has been partitioned into three regions, rather than two as obtained earlier. The other regions appear to be classified more or less correctly for this data. Analogous to the case for the Calcutta image, here also it was observed that the FCM algorithm gets trapped at local optima often enough, and the best value of the XB-index was worse than

that obtained using the genetic fuzzy clustering scheme.

4 Discussion and Conclusions

In this article, classification of satellite images into different landcover regions is modeled as the task of clustering the pixels in the intensity space. Consequently an unsupervised genetic fuzzy clustering technique has been used for classifying the image. Note that the well known FCM algorithm is a standard and popular fuzzy clustering technique when the number of clusters is known *a priori*. However, FCM solves the minimization problem through an iterative process to provide the local minimum solution. We have attempted to tackle both the problems of FCM by developing a strategy that does not require the *a priori* assumption of the number of clusters while attempting to provide near-optimal solutions. For this purpose variable string length GAs, or VGAs, has been used. In VGAs the cluster centers are encoded in the chromosome, and the fuzzy Xie-Beni index is used as a measure of its fitness. Superiority of the proposed technique over the widely used FCM algorithm is established for two IRS images of Calcutta and Mumbai both quantitatively and qualitatively.

It may be mentioned in this context that the size of the data set to be clustered in both the images is 262144. This is a reasonably large data set. The good performance of genetic fuzzy clustering method for such large data sets shows that it may be interesting to use this algorithm in data mining applications also.

There are several directions in which this work may be extended further. Firstly, a detailed

Figure 1: IRS Image of Calcutta in the Near Infra Red Band with Histogram Equalization

comparative analysis can be carried out with other validity indices as well as other similar search techniques in order to justify the use of a particular index and an underlying search tool for a given problem. Next, a detailed time and sensitivity analysis of the developed technique can be performed, and the use of other distance metrics may be investigated in the future. Moreover, several other issues like consideration of spatial information in the pixels and merging of small segments with nearby larger ones may be addressed in future. Finally the algorithm developed here can be suitably modified and tailored so that it is applicable to data mining problems where the size of the data to be clustered is often very large.

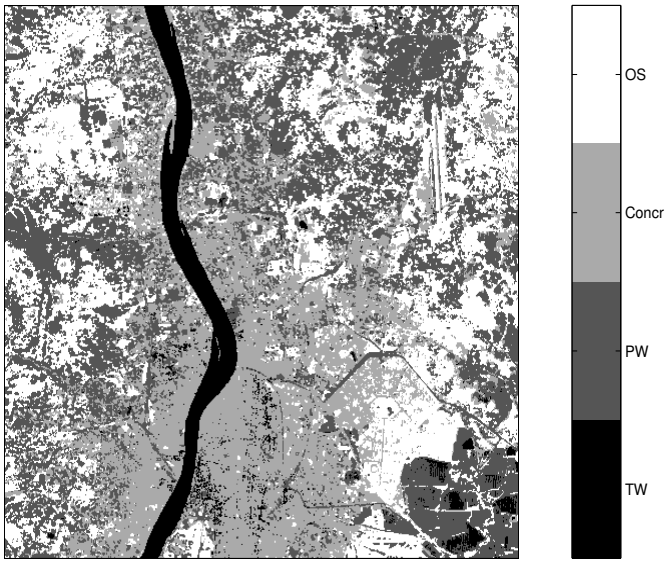

Figure 2: Clustered Image of Calcutta Using Genetic Fuzzy Clustering

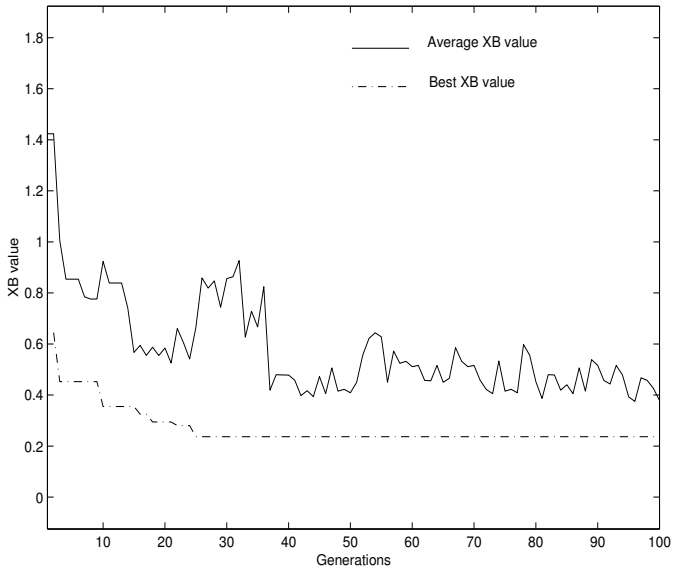

Figure 3: Variation of the XB-index with the Number of Generations for IRS Image of Calcutta

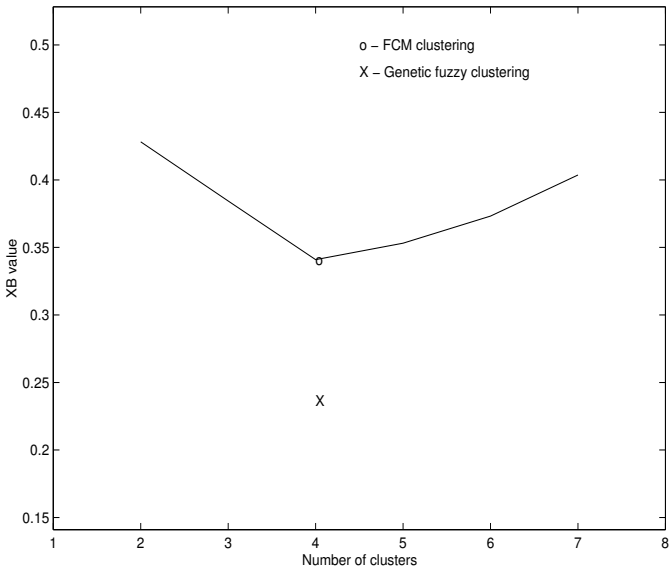

Figure 4: Variation of the XB-index with the Number of Clusters for IRS Image of Calcutta when FCM Clustering is Used

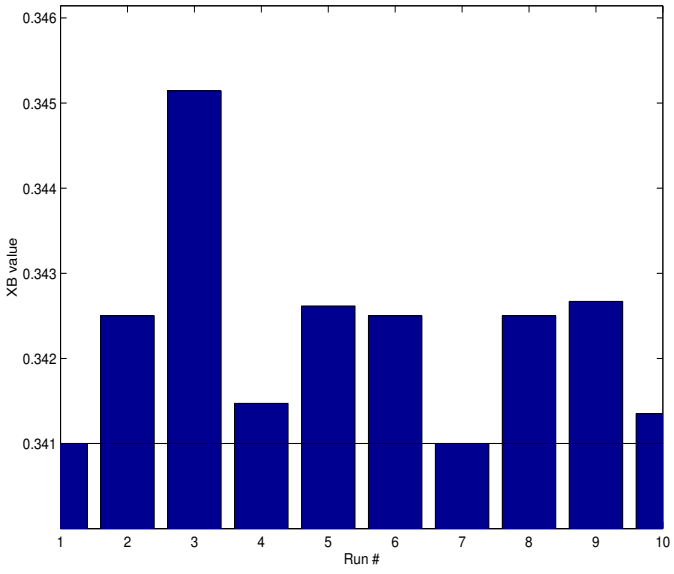

Figure 5: Histogram of the Objective Values for Ten Runs of the FCM Clustering for IRS Image of Calcutta

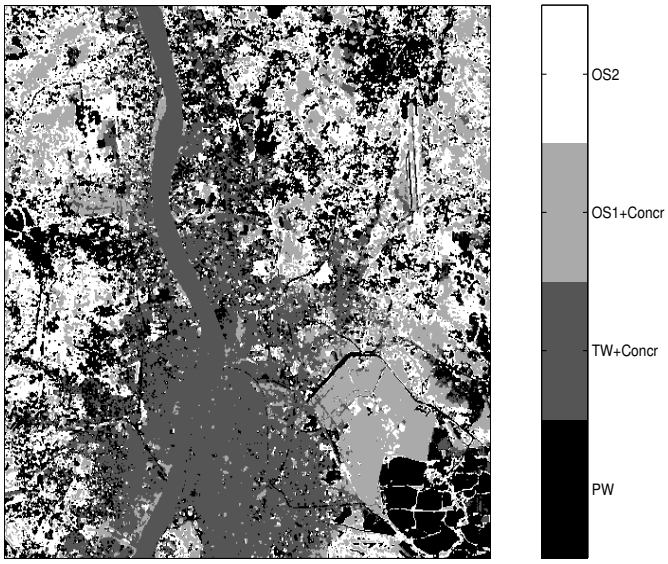

Figure 6: Clustered Image of Calcutta Using FCM Clustering

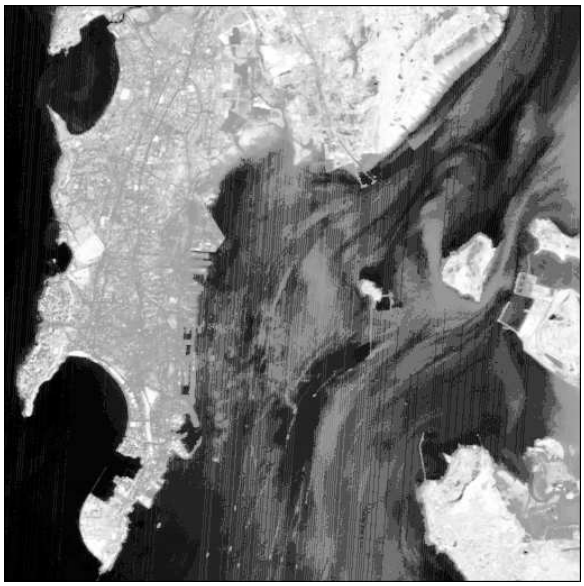

Figure 7: IRS Image of Mumbai in the Near Infra Red Band with Histogram Equalization

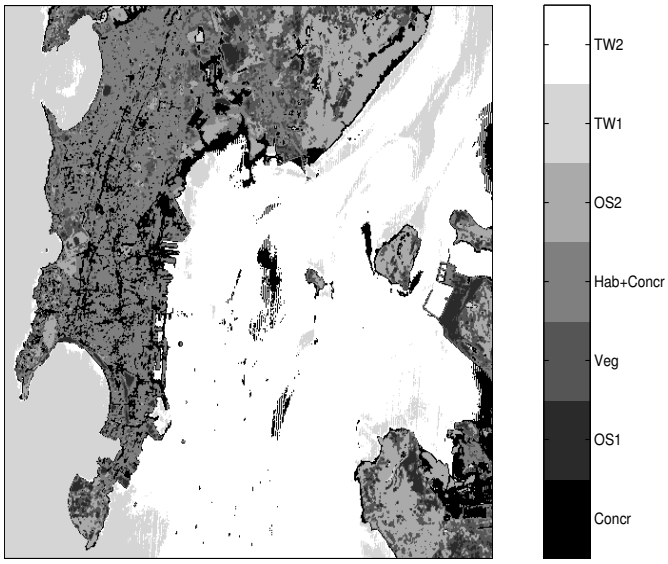

Figure 8: Clustered Image of Mumbai Using Genetic Fuzzy Clustering

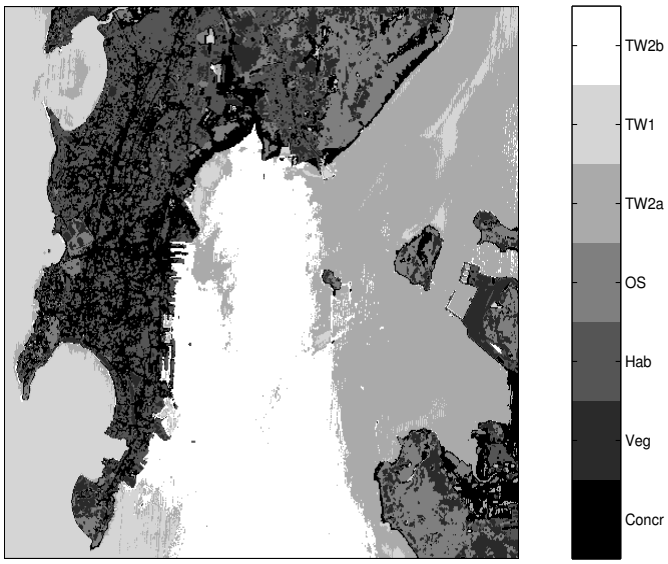

Figure 9: Clustered Image of Mumbai Using FCM Clustering