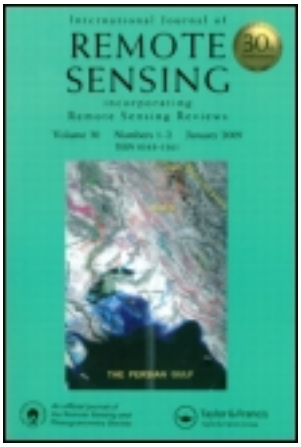

International Journal of Remote Sensing
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tres20

Genetic classifiers for remotely sensed images: Comparison with standard methods

S. K. Pal , S. Bandyopadhyay & C. A. Murthy
Published online: 25 Nov 2010.

To cite this article: S. K. Pal , S. Bandyopadhyay & C. A. Murthy (2001) Genetic classifiers for remotely sensed images: Comparison with standard methods, International Journal of Remote Sensing, 22:13, 2545-2569

To link to this article: http://dx.doi.org/10.1080/01431160120325

INT. J. REMOTE SENSING, 2001, VOL. 22, NO. 13, 2545–2569

Genetic classifiers for remotely sensed images: comparison with standard methods

S. K. PAL, S. BANDYOPADHYAY and C. A. MURTHY

Machine Intelligence Unit, Indian Statistical Institute, 203, B.T. Road, Calcutta 700 035, India; e-mail: sankar,sanghami,murthy@isical.ac.in

(*Received 23 February 1999; in final form 4 February 2000*)

Abstract. In this article the effectiveness of some recently developed genetic algorithm-based pattern classifiers was investigated in the domain of satellite imagery which usually have complex and overlapping class boundaries. Landsat data, SPOT image and IRS image are considered as input. The superiority of these classifiers over *k*-NN rule, Bayes' maximum likelihood classifier and multilayer perceptron (MLP) for partitioning different landcover types is established. Results based on producer's accuracy (percentage recognition score), user's accuracy and kappa values are provided. Incorporation of the concept of variable length chromosomes and chromosome discrimination led to superior performance in terms of automatic evolution of the number of hyperplanes for modelling the class boundaries, and the convergence time. This non-parametric classifier requires very little *a priori* information, unlike *k*-NN rule and MLP (where the performance depends heavily on the value of *k* and the architecture, respectively), and Bayes' maximum likelihood classifier (where assumptions regarding the class distribution functions need to be made).

1. Introduction

Genetic algorithms (GAs) (Goldberg 1989, Davis 1991) are randomised search and optimisation techniques guided by the principles of evolution and natural genetics. They are efficient, adaptive and robust search processes, producing near-optimal solutions and have a large amount of implicit parallelism. GAs deal with individuals called *chromosomes* (usually binary strings), which encode the parameters of the problem space and represent potential solutions. An *objective function* of a string provides a mapping from the chromosomal space to the solution space. A *fitness function* is also associated with each string which indicates the degree of 'goodness' of the solution represented by it. A set of chromosomes constitutes a *population* which is initially created randomly. Biologically inspired operators like *selection*, *crossover* and *mutation* are applied on the population over a number of generations until a termination criterion is achieved. The best string obtained at this point (or obtained so far) represents the solution of the problem.

In pattern recognition there are many tasks involved in the process of analysing/identifying a pattern which need appropriate parameter selection and efficient search in complex spaces in order to obtain optimum solutions. Therefore, the application of GAs for solving certain problems of pattern recognition (which

International Journal of Remote Sensing
ISSN 0143-1161 print/ISSN 1366-5901 online
http://www.tandf.co.uk/journals

need optimisation of computation requirements, and robust, fast and close approximate solution) appears to be appropriate and natural (Gelsema 1995, Pal and Wang 1996). Such an attempt for pattern classification in $\mathbb{R}^N$ (N-dimensional Euclidean space) has recently been made in Pal *et al.* (1998) to develop a *GA-classifier*, where the class boundaries are approximated by a number of hyperplanes. The characteristics of GAs are exploited for search and placement of a fixed number, H, of hyperplanes in the feature space, such that the number of misclassified points is minimised. In order to determine the optimum value of H automatically, the concept of variable length strings in GAs (VGAs) has been adopted. This enables the system to automatically evolve the appropriate number of hyperplanes as a parameter of the problem (Bandyopadhyay *et al.* 1998a) for modelling any kind of class boundaries non-parametrically. Unlike the conventional GAs, here the length of a string is not fixed. Crossover and mutation operators are accordingly defined. The fitness function rewards a string with smaller numbers of misclassified samples as well as smaller numbers of hyperplanes. The classifier based on VGAs is referred to as the *VGA-classifier*. (It has been theoretically shown in Bandyopadhyay *et al.* (1998a) that for an infinitely large number of interations, the number of hyperplanes provided by the *VGA-classifier* will be the minimum). Its performance is comparable to, sometimes better than, those of the Bayes' maximum likelihood classifier, k-nearest neighbour (k-NN) rule and multilayer perceptron (MLP, see figure 1) in handling various intractable pattern classes.

In this article an extensive study of the said genetic classifiers for classification of pixels for partitioning different landcover regions in satellite images is made. Note that satellite images usually have a large number of classes with overlapping and nonlinear class boundaries. Figure 2 shows, as a typical example, the complexity in scatter plot of 932 points belonging to seven classes which are taken from the SPOT image of a part of the city of Calcutta. Therefore, for appropriate modelling of such nonlinear and overlapping class boundaries, the utility of an efficient search technique like GAs is evident. Moreover, it is desirable that the search technique does not need to assume any particular distribution of the data set and/or class *a priori* probabilities.

The present analysis has two parts. In the first part, Landsat data is considered.

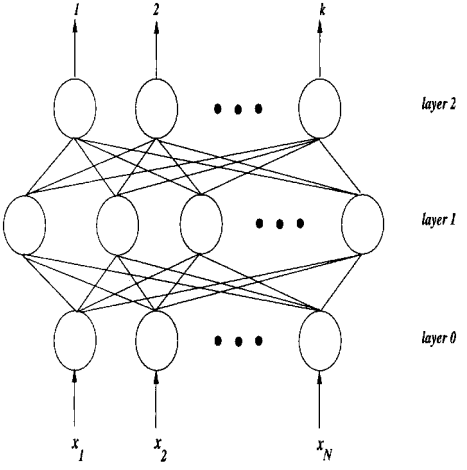

Figure 1. Multilayer perceptron.

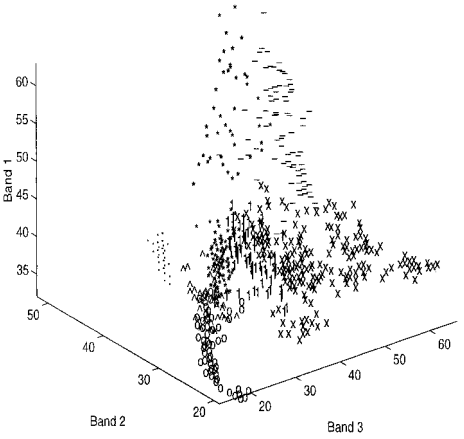

Figure 2. Scatter plot for the training dataset of SPOT image of Calcutta. The seven symbols represent seven different classes.

This is characterised by numerical feature vectors (two principal components constituting the feature space) for the classification of Manda Granite, Romapahari Granite, Vegetation, Black Phillite and Alluvium. In the second part of the investigation, we consider the problem of pixel classification from SPOT image of a part of the city of Calcutta and IRS image of a part of the city of Bombay, for segmenting different image regions, e.g. Water body, Concrete, Vegetation, Habitation. In part of the investigation, the effect of chromosome differentiation is also studied. An extensive comparison of the *GA-classifier* and *VGA-classifier* with those based on *k*-NN rule, Bayes' maximum likelihood ratio and MLP for these datasets is provided.

The article is organised as follows. Section 2 provides a brief outline of the *GA-classifier* and the *VGA-classifier*. Sections 3 and 4 provide the results of classification of Landsat data, and the IRS and SPOT imagery respectively using these genetic classifiers. Finally, the discussion and conclusions are presented in §5.

2. Description of the genetic classifiers

2.1. *GA-classifier (Pal* et al. *1998)*

As already mentioned, the *GA-classifier* attempts to place H hyperplanes in the feature space appropriately such that the number of misclassified training points is minimised. From elementary geometry, the equation of a hyperplane in N dimensional space $(X_1 - X_2 - ... - X_N)$ is given by

$$x_N \cos \alpha_{N-1} + \beta_{N-1} \sin \alpha_{N-1} = d \tag{1}$$

where
$$\beta_{N-1} = x_{N-1} \cos \alpha_{N-2} + \beta_{N-2} \sin \alpha_{N-2}$$
$$\beta_{N-2} = x_{N-2} \cos \alpha_{N-3} + \beta_{N-3} \sin \alpha_{N-3}$$
$$\vdots$$
$$\beta_1 = x_1 \cos \alpha_0 + \beta_0 \sin \alpha_0$$

The various parameters are as follows:
X_i: the ith feature of the training points.
$(x_1, x_2, ..., x_N)$: a point on the hyperplane.
α_{N-1}: the angle that the unit normal to the hyperplane makes with the X_N axis.

α_{N-2}: the angle that the projection of the normal in the $(X_1 - X_2 - ... - X_{N-1})$ space makes with the X_{N-1} axis.

⋮

α_1: the angle that the projection of the normal in the $(X_1 - X_2)$ plane makes with the X_2 axis.

α_0: the angle that the projection of the normal in the (X_1) plane makes with the X_1 axis $= 0$. Hence, $\beta_0 \sin \alpha_0 = 0$.

d: the perpendicular distance of the hyperplane from the origin.

Thus the N tuple $\langle \alpha_1, \alpha_2, ..., \alpha_{N-1}, d \rangle$ specifies a hyperplane in N-dimensional space.

Each angle α_j, $j = 1, 2, ..., N-1$ is allowed to vary in the range of 0 to 2π. If b_1 bits are used to represent an angle, then the possible values of α_j are

$$0,\ \delta 2\pi,\ 2\delta 2\pi,\ 3\delta 2\pi, ..., (2^{b_1} - 1)\delta 2\pi$$

where $\delta = 1/2^{b_1}$. Consequently, if the b_1 bits contain a binary string having the decimal value v_1, then the angle is given by $v_1 \times \delta \times 2\pi$.

Once the angles are fixed, the orientation of the hyperplane becomes fixed. Now only d must be specified in order to specify the hyperplane. For this purpose the hyper-rectangle enclosing the training points is considered. Let $(x_i^{\min}, x_i^{\max})$ be the minimum and maximum values of feature X_i as obtained from the training points. Then the vertices of the enclosing hyper-rectangle are given by

$$(x_1^{\mathrm{ch}_1}, x_2^{\mathrm{ch}_2}, ..., x_N^{\mathrm{ch}_N})$$

where each ch_i, $i = 1, 2, ..., N$ can be either max or min. (Note that there will be 2^N vertices). Let *diag* be the length of the diagonal of this hyper-rectangle given by

$$diag = \sqrt{(x_1^{\max} - x_1^{\min})^2 + (x_2^{\max} - x_2^{\min})^2 + ... + (x_N^{\max} - x_N^{\min})^2}$$

A hyperplane is designated as the *base hyperplane* with respect to a given orientation (i.e. for some $\alpha_1, \alpha_2, ..., \alpha_{N-1}$) if (i) it has the same orientation; (ii) it passes through one of the vertices of the enclosing rectangle; and (iii) its perpendicular distance from the origin is minimum (among the hyperplanes passing through the other vertices). Let this distance be $d_{\min}$.

If b_2 bits are used to represent d, then a value of v_2 in these bits represents a hyperplane with the given orientation and for which d is given by $d_{\min} + diag/2^{b_2} v_2$.

Thus each chromosome is of a fixed length of $l = H((N-1)b_1 + b_2)$, where H denotes the number of hyperplanes. These are initially generated randomly for a population of size *Pop*.

Using the parameters of the hyperplanes encoded in a chromosome, the region in which each training pattern point lies is determined based on equation (1). A region is said to provide the demarcation for class i if among the points that lie in this region the majority belong to class i. Other points that lie in this region are considered to be misclassified. The misclassifications associated with all the regions (for these H hyperplanes) are summed up to provide the total misclassification, *miss*, for the string. Its fitness is defined as $(n - miss)$, where n is the size of the training data.

After computing the fitness, the genetic operators of selection, crossover and mutation are applied (Goldberg 1989) to generate a new population of chromosomes. Elitism is incorporated in the process for preserving the best candidate found so far. Fitness computation followed by genetic operations are executed for a fixed

number of generations, at the end of which the best chromosome provides the set of hyperplanes constituting the final decision boundary.

2.2. *Determination of optimal* H*: VGA-classifier (Bandyopadhyay* et al. *1998a)*

Since it is very difficult to estimate a proper value of H, the *GA-classifier* often suffered from the problem of over fitting of the dataset, resulting from a conservative estimate of H. This also led to the presence of redundant hyperplanes in the final decision boundary. In order to overcome this limitation, the concept of variable string lengths in GAs (Goldberg *et al.* 1989), encoding the parameters of a variable number of hyperplanes, was incorporated in the *GA-classifier*, thereby providing the *VGA-classifier*.

In the *VGA-classifier*, the chromosomes are represented by strings of 1, 0 and # (don't care), encoding the parameters of variable number of hyperplanes. Let $H_{\max}$ represent the maximum number of hyperplanes that may be required to model the decision boundary of a given dataset. It is specified *a priori*.

2.2.1. *Fitness computation*

For each string i encoding H_i hyperplanes, the number of misclassified points $miss_i$ is found as in the case for *GA-classifier*. If n is the size of the training data, then the fitness of the ith string, fit_i, is defined as

$$fit_i = (n - miss_i) - \alpha H_i$$

where $\alpha = 1/H_{\max}$ and H_i is the number of hyperplanes encoded in the string. A string with zero hyperplane is defined to have zero fitness. Maximisation of the fitness function ensures the minimisation of, primarily, the number of misclassified points and then the number of hyperplanes.

2.2.2. *Genetic operators*

Since the strings have variable length, the operators crossover and mutation were newly defined as follows.

Crossover: Two strings, i and j, having lengths l_i and l_j, respectively, are selected from the mating pool. Let $l_i \leqslant l_j$. Then string i is padded with #s so as to make the two lengths equal. Conventional crossover like single-point crossover, two-point crossover (Goldberg 1989) is now performed over these two strings with probability μ_c. The following two cases may now arise:

1. All the hyperplanes in the offspring are complete. (A hyperplane in a string is called *complete* if all the bits corresponding to it are either defined (i.e. 0s and 1s) or #s. Otherwise it is incomplete.)
2. Some hyperplanes are incomplete.

In the second case let u = number of defined bits (either 0 or 1) and t = total number of bits per hyperplane. Then, for each incomplete hyperplane, all the #s are set to defined bits (either 0 or 1 randomly) with probability u/t. Otherwise, all the defined bits are set to # with a probability $(1-u/t)$. Thus each hyperplane in the string becomes complete. Subsequently, the string is rearranged so that all the #s are pushed to the end.

Mutation: In order to introduce greater flexibility in the method, the mutation operator is defined in such a way that it can both increase and decrease the string length. For this, the strings are padded with #s such that the resultant length becomes equal to $l_{\max}$. Now for each defined bit position, it is determined whether conventional

mutation (Goldberg 1989) can be applied or not with probability μ_m. Otherwise, the position is set to # with probability μ_{m_1}. Each undefined position is set to a defined bit (randomly chosen) according to another mutation probability μ_{m_2}.

Note that mutation may result in some incomplete hyperplanes, and these are handled in a manner, as done for crossover operation. Also, mutation may yield strings having all #s indicating that no hyperplanes are encoded in it. Consequently, this string will have fitness = 0 and will be automatically eliminated during selection. The details are available in Bandyopadhyay *et al.* (1998a).

3. Classification of Landsat data

3.1. *Dataset*

The dataset considered here is obtained by the multispectral scanner (MSS) used in Landsat-V for recording remotely sensed images. The intensity of a pixel is resolved on the electromagnetic spectrum into four bands, which are taken to be four features. The four bands are: green band, wavelength 0.5–0.6 μm; red band, wavelength 0.6–0.7 μm; near-infrared band, wavelength 0.7–0.8 μm; and infrared band, wavelength 0.8–1.1 μm. The area of the Earth's surface covered by each pixel is 79 m $\times$ 79 m.

Since the features are highly correlated, principal component analysis was done (Pal 1990, 1993) to reduce the four features to two principal features. Such a 2-dimensional feature space representing the satellite imagery data of various rocks, vegetation and soil is shown in figure 3. It has 795 samples with five classes, namely Manda Granite, Romapahari Granite, Vegetation, Black Phillite and Alluvium (Pal 1993). These are referred to as classes 1, 2, 3, 4 and 5, respectively. The details of the extraction procedure are available elsewhere (Pal 1990, 1993).

3.2. *Implementation parameters*

For the *GA-classifier*, population size is kept equal to 20. The values of b_1 (bits for representing an angle) and b_2 (bits for representing perpendicular distance) are 8 and 16, respectively. Ten percent of the dataset is used for training while the remaining 90% is used for testing.

Roulette wheel selection is adopted to implement the *proportional selection strategy*. *Single point* crossover is applied with a fixed crossover probability (μ_c) value of 0.8. The mutation operation is performed on a bit by bit basis for a varying mutation probability value (μ_m) in the range [0.015,0.333]. The range is divided into eight equispaced values. In the initial stages of the algorithm, μ_m is set to a high value, which is first decreased in steps to the minimum value and then increased again in the later stages of the algorithm. This ensures that in the initial stage, when the algorithm has very little knowledge about the search domain, it performs a random search through the feature space. The randomness is gradually decreased with the passing of generations so that now the algorithm performs a detailed search in the vicinity of promising solutions obtained so far. In spite of this, the algorithm may still get stuck at a local optimum. This problem is overcome by increasing the mutation probability to a high value, thereby making the search more random once again. The variation of μ_m with generations is shown in figure 4. One hundred iterations are executed with each value of μ_m for a total of 1500 iterations. The algorithm is terminated if the population contains at least one string with no misclassified points. Otherwise, the algorithm is executed for 1500 generations.

For the *VGA-classifier*, initial population is created in such a way that the first

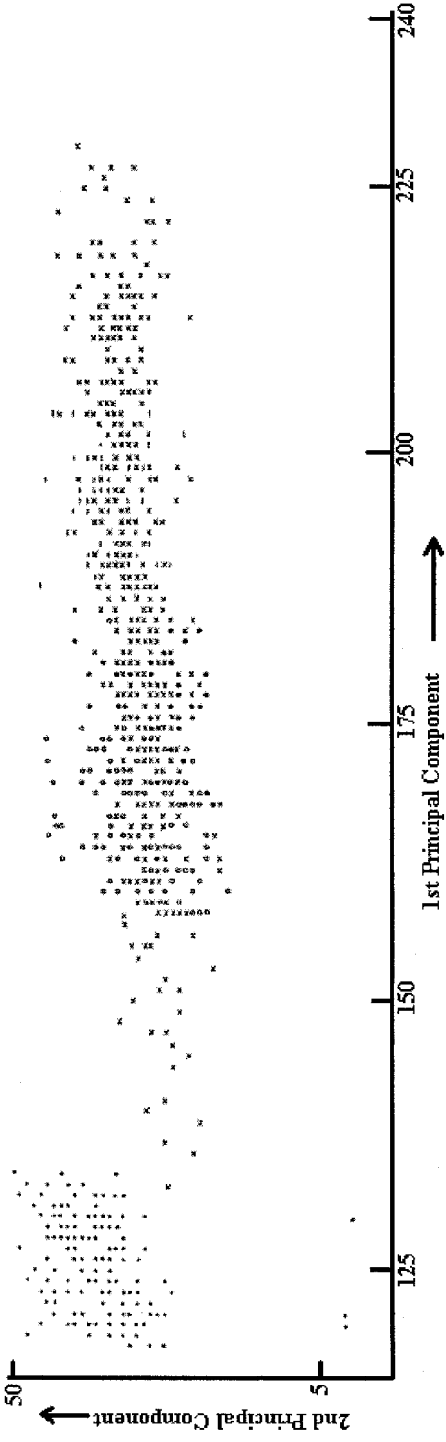

Figure 3. Landsat data.

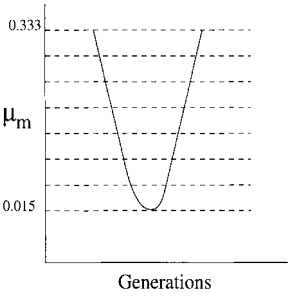

Figure 4. Variation of mutation probability value with the number of generations.

and the second strings encode the parameters $H_{\max}$ and 1 hyperplanes, respectively, to ensure sufficient diversity in the population. For the remaining strings, the number of hyperplanes is generated randomly in the range $[1, H_{\max}]$, and the corresponding bits are initialised randomly to 1s and 0s. Note that the search space for *VGA-classifier* is larger (since it considers any value of H in the range $[1, H_{\max}]$) compared to the *GA-classifier* (which considers only a preassigned value of H). The former is therefore given more time to come up with an appropriate solution by increasing the number of generations for each value of μ_m from 100 to 200; thereby allowing it to execute for a maximum of 3000 generations. The other implementation parameters are kept the same as for *GA-classifier*.

3.3. *Results*

In order to compare the performance of the genetic classifiers with some other well known ones, we have considered the k-NN classifier, Bayes' maximum likelihood classifier and MLP. For the convenience of the readers, a brief description of these classifiers is provided in the Appendix. For k-NN classifier, the different values of k considered are 1, 3, 5 and $\sqrt{n}$ where n is the size of the training data. (It is known that as the number of training patterns n goes to infinity, if the values of k and k/n can be made to approach infinity and 0, respectively, then the k-NN classifier approaches the optimal Bayes' classifier (Cover and Hart 1967, Fukunaga 1972). One such value of k for which the limiting conditions are satisfied is $\sqrt{n}$.) Bayes' maximum likelihood classifier is executed assuming normal distribution of the dataset with unequal dispersion matrices and class *a priori* probabilities equal to n_i/n where n_i is the number of points in class i. For MLP, the learning rate and momentum factor are fixed at 0.8 and 0.2, respectively. Maximum of 3000 iterations are executed. Two typical architectures, namely *Arch 1* = 2:5:5 (*input* : *hidden* : *output*) and *Arch 2* = 2:10:10:5 (*input* : $hidden_1$: $hidden_2$: *output*) are considered. Online weight updation is performed.

Table 1 shows a comparative performance in terms of percentage recognition scores of the different classifiers for Landsat data. (A comparison in terms of additional measures is also provided in §3.5.) The entries in table 1 for the GA-based classifiers and MLP are the values averaged over five different runs of the algorithms starting from five different initial configurations. As expected, class 3 (Vegetation) is amenable to a consistently good recognition score since this class has almost no overlap with the other classes (those of various types of rocks) and lies at one extreme end of the feature space. Recognition of class 2 (Romapahari granite) is seen to be poor for most of the classifers since this is totally overlapped with other classes, especially, with classes 4 (Black Phillite) and 5 (Alluvium). These findings are also

Table 1. Comparative recognition scores (%) during testing for Landsat data for $perc = 10$.

		Class					
		1	2	3	4	5	Overall
GA-classifier	$H = 4$	97.62	40.74	100.00	98.45	45.26	82.98
	$H = 6$	76.98	45.37	98.45	94.32	81.05	83.26
VGA-classifier	$H_{max} = 10$	86.50	44.44	98.45	89.69	78.94	83.35
Bayes'		82.54	75.92	96.91	60.82	71.58	78.10
k-NN classifier	$k = 1$	87.30	44.44	100.00	72.68	61.05	76.84
	$k = 3$	87.68	36.96	100.00	84.05	72.68	79.89
	$k = 5$	89.68	20.07	100.00	92.26	78.94	81.86
	$k = \sqrt{n}$	88.09	31.48	100.00	90.72	85.26	83.12
MLP	Arch 1	65.07	70.37	57.21	90.20	33.68	66.38
	Arch 2	42.85	72.22	76.20	79.89	36.31	61.50

corroborated in table 2, which shows the confusion matrix over the test dataset for one particular run of the *VGA-classifier*. Column UC denotes the points that the *VGA-classifier* could not classify into any one of the classes (which may happen if the corresponding points lie in a region from where no training points came). The confusion matrix shows that a large number of points from class 2 are misclassified as points of class 4 and then 5. Class 5 also got confused with class 2 for a significant number of points. Interestingly, most of the points belonging to Black Phillite (class 4) are recognised correctly.

As seen from table 1, the *VGA-classifier* provides the best recognition score. Interestingly, the different versions of the genetic classifiers yield better recognition scores as compared to those of Bayes' maximum likelihood ratio and MLP-based classifiers. The performance of the k-NN classifier for $k = \sqrt{n}$ is found to be comparable to or marginally poorer than those of the GA-based classifiers. The results of the k-NN classifier are found to depend on the proper choice of k, being best for $k = \sqrt{n}$. Similarly, the performance of MLP is found to depend heavily on the architecture chosen. Its poor scores for this dataset may be due to an inappropriate choice of the architecture. Moreover, some pruning techniques and sophisticated learning algorithms may be applied for improving the performance of MLP. Bayes' maximum likelihood classifier is also found to yield poor recognition score. It appears that the assumption of normal distribution is not proper for this dataset. The *VGA-classifier*, on the other hand, is found to be able to automatically reduce the number

Table 2. Confusion matrix for Landsat data obtained during testing of *VGA-classifier* (one run) for $H_{max} = 10$. (UC denotes points that are unclassified.)

		Recognised as					
		1	2	3	4	5	UC
Actual	1	118	0	3	0	5	0
	2	0	52	2	38	16	0
	3	0	0	194	0	0	0
	4	0	0	3	185	0	6
	5	13	39	3	0	40	0

of surfaces to 5 (thereby eliminating the need to fix a proper value of H as in *GA-classifier*) while providing the best recognition score for this dataset. Moreover, it does not assume or use any underlying property of the data. This indicates that the *VGA-classifier*, which is nonparametric in nature, is able to appropriately model any type of class boundary automatically and also provide good generalisation capability.

3.4. *Incorporation of chromosome differentiation*

In a part of the experiment, we investigated the effect of chromosome discrimination, thereby providing a methodology called GACD (Bandyopadhyay *et al.* 1998b), on the performance of the said *GA-classifier*. In GACD, the chromosomes are divided into two categories, namely M and F based on the value contained in the two class bits prepended to the chromosome. The initial population is created in such a way that the M population is the farthest, in terms of Hamming distances, from the F population. Crossover is allowed only between two chromosomes belonging to the aforesaid two classes. The class bits of the offspring are determined by mainly the M parent. These bits are excluded while performing the mutation operation. A schema analysis (Bandyopadhyay *et al.* 1998b) shows that in many situations the lower bound of above average schemata sampled by GACD is greater than that of conventional GAs.

The parameters for *GACD-classifier* are kept the same as those of *GA-classifier*. Figure 5 shows the variation of the best recognition score during training of the two

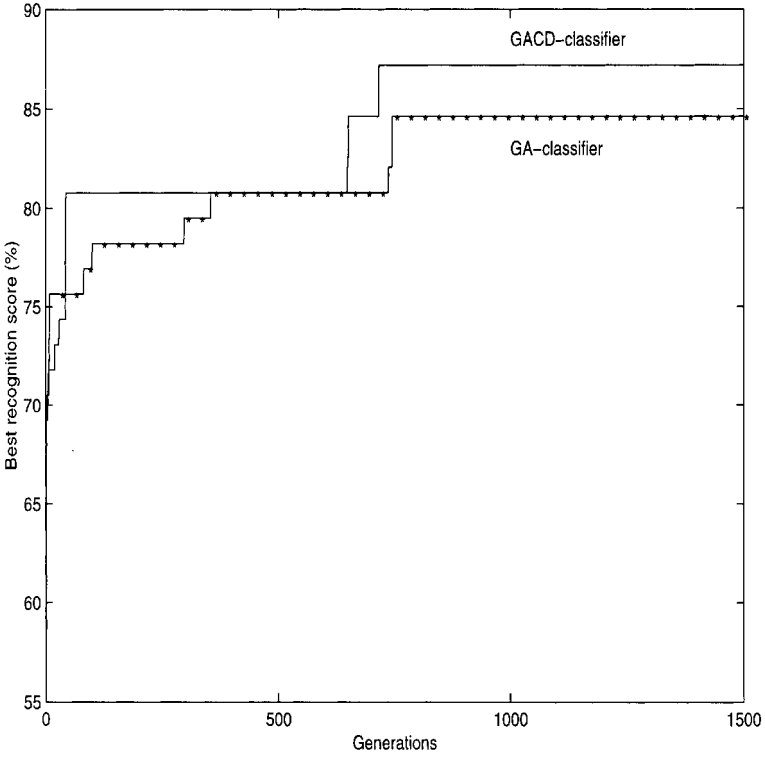

Figure 5. Variation of best recognition scores of *GACD-classifier* and *GA-classifier* with the number of generations for Landsat data with $H=4$.

classifiers for $H=4$. It is evident that except in the very early stages, the *GACD-classifier* is able to provide better recognition scores than the *GA-classifier* in a given number of iterations. This indicates a faster rate of convergence of the former, which conforms to the earlier findings in (Bandyopadhyay *et al.* 1998b). Table 3 shows the recognition scores during training of the two classifiers, and the test score of the *GACD-classifier* (the test scores for the *GA-classifier* are already mentioned in table 1). It is found that for $H=4$ the *GACD-classifier* outperforms the *GA-classifier* while for $H=6$, the opposite is true. However for both values of H, the training performance of *GACD-classifier* is better than that of the *GA-classifier*. This indicates that better recognition scores during training (which may take place when a large value of H quickly and closely fits the training data) may not necessarily result in better generalisation capability.

3.5. *Comparison in terms of user's accuracy and kappa measure*

In the previous section we have provided the comparative results in terms of percent of correct classification obtained from the confusion matrix. The percentage correct classification is defined as

$$\frac{n_{i_c}}{n_i} \times 100 \tag{2}$$

where n_i is equal to the number of points in class i, of which n_{i_c} points have been correctly classified (present in the diagonal element (i,i) of the confusion matrix). Note that n_{i_c}/n_i is also called the Producer's accuracy (Schriever and Congalton 1995).

In this section, we provide two more statistical measures, namely user's accuracy and kappa (Congalton *et al.* 1983, Rosenfield and Fitzpatrick-Lins 1986). If n'_i points (of all the n points) are found to be classified into class i, then the user's accuracy (U) is defined as

$$U = n_{i_c}/n'_i \tag{3}$$

where n_{i_c} is as defined earlier. Note that the user's accuracy gives a measure of the confidence that a classifier attributes to a region as belonging to a class. In other words, it denotes the level of purity associated with a region.

The coefficient of agreement called 'kappa' measures the relationship of beyond chance agreement to expected disagreement. It uses all the cells in the confusion matrix, not just the diagonal elements. The estimate of kappa (κ) is the proportion

Table 3. Recognition scores (%) during training and testing of *GACD-classifier* for Landsat data.

	Recognition scores (%)		
	During training of		During testing of
H	*GA-classifier*	*GACD-classifier*	*GACD-classifier*
4	84.61	87.17	84.52
6	88.46	89.74	78.80

of agreement after chance agreement is removed from consideration. The estimate of kappa for class i (κ_i) is defined as

$$\kappa_i = \frac{n \times n_{i_c} - n_i \times n'_i}{n \times n'_i - n_i \times n'_i} \tag{4}$$

The numerator and denominator of the overall kappa are obtained by summing the respective numerators and denominators of κ_i separately over all classes i.

Table 4 shows the above measures for different classifiers. Here we considered $H=6$ for *GA-classifier*, $H=4$ for *GACD-classifier*, $k=\sqrt{n}$ for k-NN rule, and Arch 2 for MLP which were found to provide best scores (in the respective categories) in table 1. The other parameters for the classifiers were kept the same as in §3.3 and 3.2. As earlier, the *GACD-classifier* for $H=4$ provides the largest kappa value. This is followed by those for *GA-classifier* ($H=6$), *VGA-classifier*, k-NN rule, Bayes' and MLP.

It is found from table 4 that the user's accuracy value is usually high for class 3, indicating that the region identified as belonging to this class has high purity. This is expected since it lies at one extreme end of the feature space (figure 3). On the contrary, the user's accuracy is, in general, low for class 5 indicating that the corresponding region is relatively impure. This is also expected since, as seen from figure 3, this class is totally dispersed among the other classes.

4. Pixel classification of SPOT and IRS images

In this section, some results on classifying the pixels in IRS and SPOT images of a part of the city of Calcutta and Bombay using genetic classifiers are provided. The performance is compared to those of Bayes' maximum likelihood classifier and k-NN classifiers for $k=1$, 3 and $\sqrt{n}$.

4.1. *Datasets*

4.1.1. *SPOT image of a part of Calcutta*

The French satellites SPOT (Systems Probataire d'Observation de la Terre) (Richards 1993), launched in 1986 and 1990, carry two imaging devices that consist of a linear array of charge-coupled device (CCD) detectors. Two imaging modes are possible, the multispectral and panchromatic modes. The image considered in this

Table 4. Users accuracy (U) and kappa values (κ) in % corresponding to the different classifiers for Landsat data when $perc=10$.

Class	Measure	GA classifier	VGA classifier	GACD classifier	Bayes' classifier	k-NN rule	MLP
1	U	95.04	94.30	93.54	96.33	87.76	100.0
	κ	93.98	93.09	92.17	95.54	85.16	100.0
2	U	100.0	56.04	97.14	42.56	54.16	37.62
	κ	100.0	48.24	96.63	32.38	46.03	26.56
3	U	94.63	94.63	94.17	99.46	94.63	94.17
	κ	92.64	92.64	92.01	99.27	92.64	92.01
4	U	83.32	84.14	81.44	82.06	86.05	85.71
	κ	75.77	78.25	78.67	75.41	80.88	80.41
5	U	59.85	63.38	66.14	67.71	62.31	13.63
	κ	53.72	57.78	60.97	62.78	56.56	0.44
Overall	κ	80.20	78.55	82.21	72.38	77.20	66.89

experiment has three bands in the multispectral mode (figure 6 shows the image in the third band). These bands are: band 1—green band, wavelength 0.50–0.59 μm; band 2—red band, wavelength 0.61–0.68 μm; and band 3—near-infrared band, wavelength 0.79–0.89 μm.

The training dataset comprises 932 points belonging to seven classes, with three features corresponding to the above-mentioned three bands. The seven classes are Turbid water (TW), Pond water (PW), Concrete (Concr), Vegetation (Veg), Habitation (Hab), Open space (OS) and Roads (including bridges) (B/R).

Some important landcovers are seen to be present in the image. Most of these can be identified, from a knowledge about the area, more easily in band 3 of the input image (figure 6). These are the following:

The prominent black stretch across the figure is the river Hooghly. Portions of a bridge (referred to as the second bridge), which was under construction when the picture was taken, protrude into the Hooghly near its bend around the centre of the image. There are two distinct black, elongated patches below the river, on the left side of the image. These are water bodies, the one to the left being Garden Reach lake and the one to the right being Khidirpore dockyard. Just to the right of these water bodies, there is a very thin line, starting from the right bank of the river, and going to the bottom edge of the picture. This is a canal called the Talis nala. Above the Talis nala, on the right hand side of the picture, there is triangular patch, the race course. On the top, right-hand side of the image, there is a thin line, stretching

Figure 6. SPOT image of Calcutta in the near-infrared band (band 3). The image is histogram stretched.

from the top edge, and ending on the middle, left edge. This is the Beleghata canal with a road by its side. There are several roads on the right side of the image, near the middle and top portions. These are not very obvious from the images. A bridge cuts the river near the top of the image. This is called the Rabindra Setu.

4.1.2. *IRS image of Bombay*

This image was obtained from Indian Remote Sensing Satellite (IRS-1A). This is a circular sun-synchronous satellite, rotating around the Earth at the rate of 14 orbits per day, at an altitude of 904 km and a repetition cycle of 22 days (NRSA 1986). This satellite is equipped with two different sensors LISS-I and LISS-II. LISS-I has a resolution of 72.5 m × 72.5 m while LISS-II has a resolution of 36.25 m × 36.25 m. Data used for this work was obtained from LISS-II sensor which has a focal length of 324.4 m and radiometric resolution of 128. The whole spectrum range has been decomposed into four spectral bands namely: blue band, wavelength 0.45–0.52 μm; green band, wavelength 0.52–0.59 μm; red band, wavelength 0.62–0.68 μm; and near-infrared band, wavelength 0.77–0.86 μm.

We have considered here the green, red and near-infrared bands only, subsequently referred to as band 1, band 2 and band 3, respectively, since these bands were found to be more sensitive than the blue band to discriminate various land cover types. Figure 7 shows the IRS image of a part of Bombay in the third band. The elongated city area is surrounded by the Arabian sea. There is a concrete structure (on the right side top corner) connecting Bombay to New Bombay. On the

Figure 7. IRS image of Bombay in the near-infrared band (band 3). The image is histogram stretched.

southern part of the city, there are several islands, including the well known Elephanta islands. The dockyard is situated on the south eastern part of Bombay, which can be seen as a set of three finger like structures. On the upper part of the image, towards left, there is a distinct criss-crossed structure. This is the Santa Cruz airport.

The training dataset comprises 198 points belonging to six classes, with three features corresponding to the above-mentioned three bands. The six classes are labelled Turbid water 1 (TW1), Turbid water 2 (TW2), Concrete (Concr), Habitation (Hab), Vegetation (Veg) and Open space (OS). Note that the sea water is decomposed into two classes TW1 and TW2 for better classification since they have somewhat different reflectance properties (as can be seen in figure 8). However, in the output decision while assigning class labels, these two categories are merged into one class, namely Turbid water (TW), i.e. any pixel being classified either as TW1 or TW2 is labelled as TW.

4.2. *Issue of large value of* H

In view of the complexity of the datasets, high values of H like 15 and 20 for the GA-based classifiers were considered. Since the maximum number of regions provided by H hyperplanes is equal to 2^H, the aforesaid high values of H make the number of regions ($=2^H$) also very large. This leads to a practical limitation of the method. However, an important point that needs to be taken into consideration is that the possible number of regions can never be larger than the number of points n in the training dataset. Also, $n \ll 2^H$ for large H. Thus we need to consider at most n regions while tackling this problem. In fact, the number of regions for this problem was found to be considerably less than n as well.

4.3. *Implementation parameters*

The performances of the GA-based classifiers are compared to those of the Bayes' maximum likelihood classifier and k-NN rule (for three values of k) for these images. The parameters are kept the same as described in §3.2. Different values of H considered for the *GA-classifier* are 10, 15 and 20. The value of $H_{\max}$ for *VGA-classifier* is taken to be 15. The values of k, for k-NN classifier, are 1, 3 and $\sqrt{n}$.

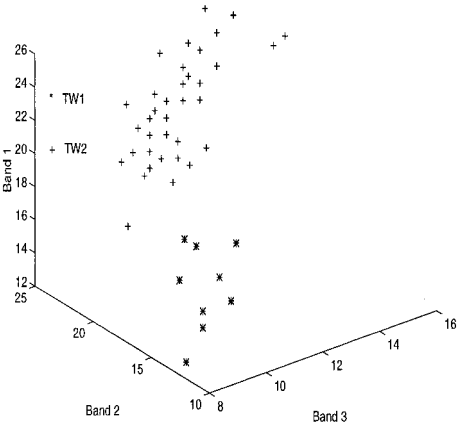

Figure 8. Scatter plot for classes TW1 and TW2 of the training dataset for IRS image of Bombay.

4.4. *Results for SPOT image of Calcutta*

The comparative results corresponding to the SPOT image of Calcutta are demonstrated by zooming only the race course (a triangular-shaped structure), which clearly helps to discriminate between the output classified images obtained from the different classifiers. Figures 9(*a*)–(*c*) present the results of the *GA-classifier* for $H=$ 10, 15 and 20. Figure 9(*d*) shows the result of the *VGA-classifier* corresponding to $H_{\max}=15$. Figures 9(*e*)–(*g*) present the same based on k-NN rule corresponding to $k=1$, 3 and $\sqrt{n}$, while figure 9(*h*) provides the results for the Bayes' maximum likelihood classifier. For the convenience of the readers, the full classified images, showing the other regions as well, corresponding to the *VGA-classifier* and Bayes' maximum likelihood classifer are also shown in figures 10 and 11.

k-NN rule with $k=1$ (figure 9(*e*)) performs the poorest among all the classifiers. It fails to retain the shape of the *race course*. It was also found to be unable to classify roads on the right side of the river; being able to identify some scattered road pixels (white dots) instead. The performance of k-NN rule is found to improve with the value of k, being best for $k=\sqrt{n}$, where the extracted features are found to be much more prominent.

Although all the classifiers are able to locate the race course, only the *VGA-classifier* and the *GA-classifier* for $H=10$ and 15 are able to identify a triangular lighter outline within it properly. This is an open space, corresponding to the tracks in the race course. The overall performance of the GA classifier is found to improve from $H=10$ to 15 but is poorer for $H=20$. One reason for the degradation in performance for $H=20$ is that the classifier is able to surround the training data points very closely with the help of these 20 hyperplanes. This leads to overfitting of the dataset during training and hence reduced generalisation capability during testing.

Bayes' maximum likelihood classifier was found to overestimate some road classes. For example, the thick road structures obtained along both the banks of Hooghly (see figure 11) are an overestimation. The *VGA-classifier*, which succeeds in reducing the number of hyperplanes from 15 to 9, is found to confuse between the classes PW and Concr, and B/R (see figure 10). This is because of a large overlap between the classes Concr and B/R on one hand (in fact, the latter class has been extracted from the former) and PW and B/R on the other hand. These are evident from figures 12 and 13, respectively. The fact that a large number of points actually belonging to the class B/R is wrongly classified to either PW or Concr is also evident from table 5 which shows the confusion matrix on the training dataset obtained by the Bayes' maximum likelihood classifier.

4.5. *Results for IRS image of Bombay*

As in §4.4, the comparative results corresponding to the IRS image of Bombay are demonstrated by zooming only the dockyard (the finger-like structures), which clearly helps to discriminate between the output classified images obtained from the different classifiers. Figures 14(*a*)–(*c*) present the dockyard in the output classified images of Bombay using the *GA-classifier* corresponding to $H=10$, 15 and 20. Figure 14(*d*) provides the same for the *VGA-classifier* for $H_{\max}=15$. Figures 14(*e*)–(*g*) demonstrate the dockyard in the output images of the k-NN classifier corresponding to $k=1$, 3 and $\sqrt{n}$. Figure 14(*h*) presents the same for the Bayes' maximum likelihood classifier. For the convenience of the readers, the full classified

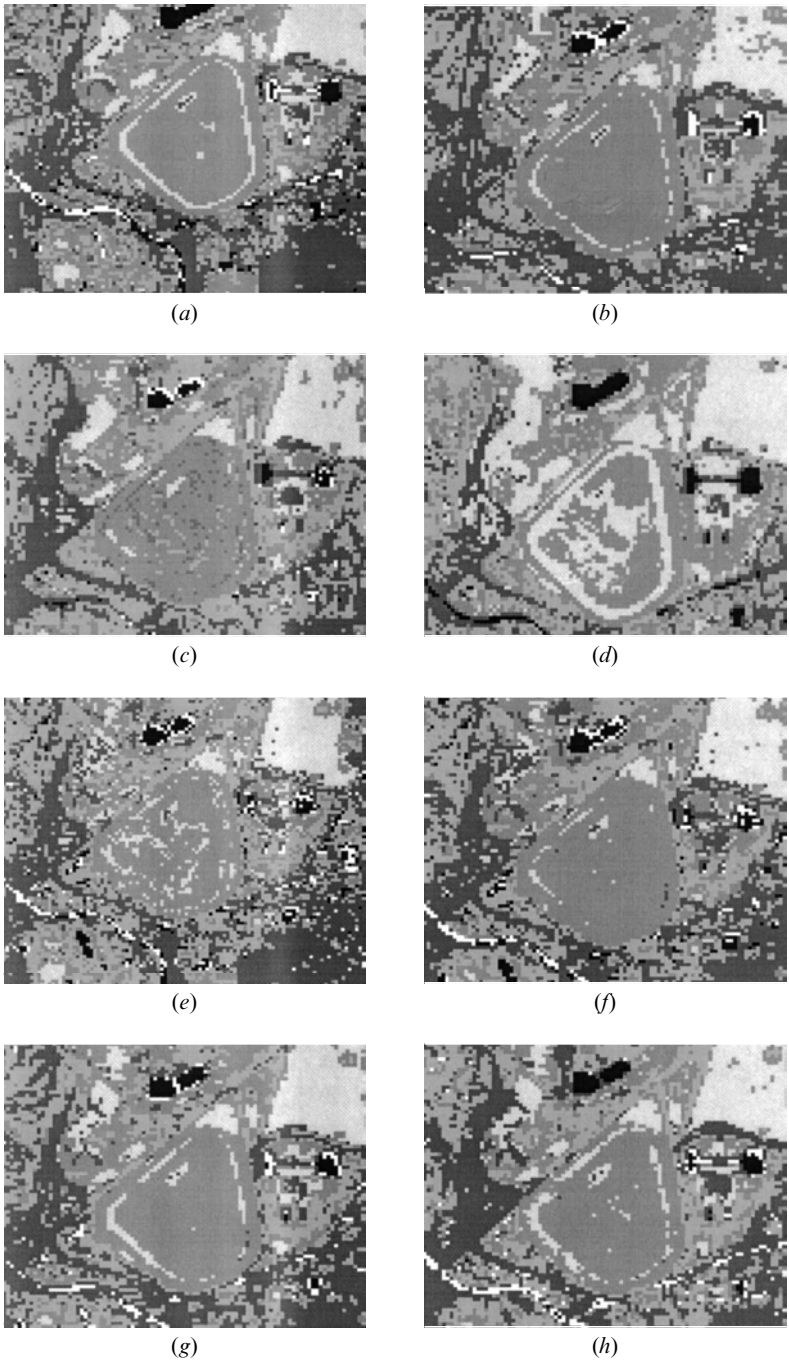

Figure 9. Classified SPOT image of Calcutta (zooming the race course only) using (*a*) GA classifier, $H=10$, (*b*) GA classifier, $H=15$, (*c*) GA classifier, $H=20$, (*d*) VGA classifier, $H_{\max}=15$, final value of $H=9$, (*e*) k-NN rule, $k=1$, (*f*) k-NN rule, $k=3$, (*g*) k-NN rule, $k=\sqrt{n}$, (*h*) Bayes' maximum likelihood classifier.

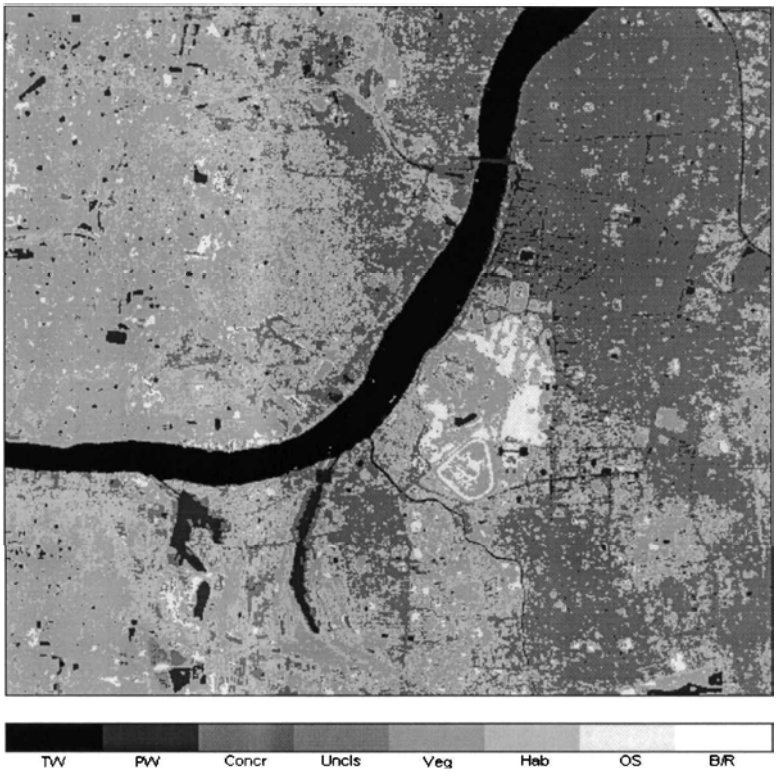

Figure 10. Classified SPOT image of Calcutta using *VGA-classifier*, $H_{\max} = 15$, final value of $H = 9$.

images, showing the other regions as well, corresponding to *VGA-classifier* and Bayes' maximum likelihood classifier are also provided in figures 15 and 16.

From figures 14(*a*)–(*c*) it is found that, as in the case of Calcutta image, the performance of the *GA-classifier* is better for $H = 15$ than with $H = 10$, but poorer for $H = 20$. Overfitting of the dataset and the subsequent reduction in the generalisation capability is once again responsible for the degradation of performance from $H = 15$ to $H = 20$. Output images for both $H = 10$ and 20 were found to have a large number of unclassified points (6327 and 3578, respectively) as compared to $H = 15$ (which has 1242 unclassified points). The *VGA-classifier* (figure 14(*d*)) is found to extract the dockyard nicely while reducing automatically the number of hyperplanes to 11. Only a small number of points (= 266) remained unclassified in the image. The Bayes' maximum likelihood classifier was found to identify an unusually large portion of the image as belonging to the concrete class. This is evident from figure 14(*h*), as also from figure 16, where the shape of the dockyard is found to be lost. Because of this reason, the bridge connecting Bombay to New Bombay came out correctly as a more or less continuous structure for the Bayes' classifier (see figure 16). This was in contrast to all other classifiers where it came out usually as a discontinuous concrete structure (see figure 15 as an illustration). For *k*-NN classifiers, the performance was again found to improve gradually with the value of *k*. While for $k = 1$, the shape of the dockyard is not clear, it becomes better for $k = 3$ and $\sqrt{n}$.

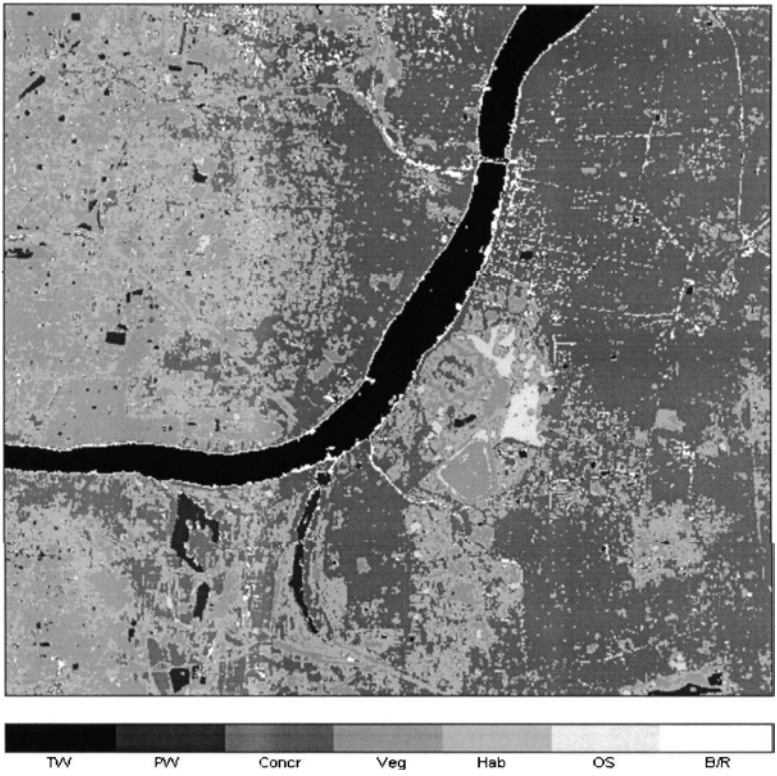

Figure 11. Classified SPOT image of Calcutta using Bayes' maximum likelihood classifier.

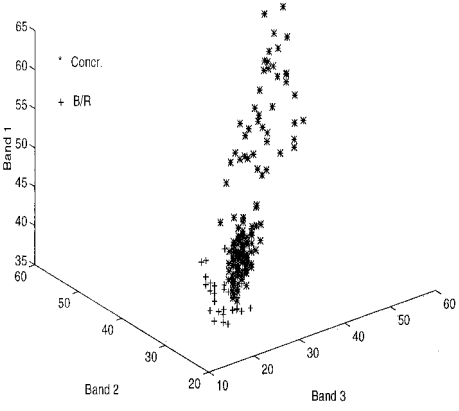

Figure 12. Scatter plot for classes Concr and B/R of the training dataset for SPOT image of Calcutta.

5. Discussion and conclusions

In this article, a comparison is made between some GA-based classifiers, Bayes' maximum likelihood classifier, *k*-NN rule and MLP, for classifying different land-cover types. For Landsat data, the performance is measured quantitatively (e.g. percent recognition scores, user's accuracy, and kappa values), whereas it is the

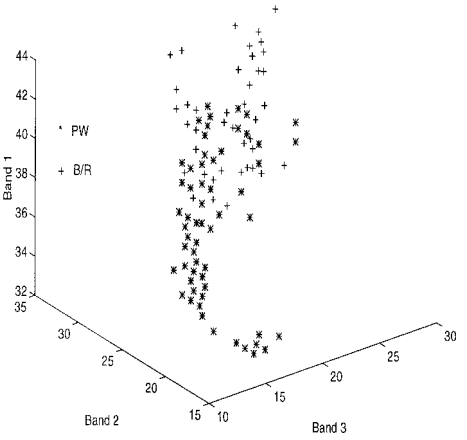

Figure 13. Scatter plot for classes PW and B/R of the training dataset for SPOT image of Calcutta.

Table 5. Confusion matrix for training data of SPOT image of Calcutta obtained using Bayes' maximum likelihood classifier.

		Recognised as						
		TW	PW	Concr	Veg	Hab	OS	B/R
Actual	TW	125	0	0	0	0	0	0
	PW	0	149	11	0	0	0	4
	Concr	0	0	152	0	2	8	10
	Veg	0	0	1	223	22	4	0
	Hab	0	0	12	12	51	0	1
	OS	0	0	5	6	0	82	0
	B/R	0	10	15	0	0	0	27

quality of the segmented image output which was considered for analysing the same on the SPOT and IRS images.

For Landsat data, the concept of variable string lengths is found to be useful for evolving a proper value of H. Overall, the genetic classifiers performed better (in terms of recognition scores and kappa values) than the Bayes, k-NN rule ($k=1$, 3 and 5) and MLP; thereby strengthening the earlier findings of Pal *et al.* (1998) and Bandyopadhyay *et al.* (1998a). In the case of k-NN rule with $k=\sqrt{n}$, the performance was found to be comparable to those of the genetic classifiers, except for the case of *GACD-classifier* for $H=4$, whose performance was significantly superior. As expected, the user's accuracy is seen to be high for class 3 (which has least overlap with the other classes), and low for class 5 (being totally overlapped with all classes).

Most of the classifiers are found more or less to identify some characteristic regions in the different images properly. The quality of the segmented image provided by k-NN classifier improves with the value of k. The Bayes' maximum likelihood classifier is able to identify most of the regions in the image correctly, although it overestimates some concrete (in the Bombay image) and road structures (in the SPOT image of Calcutta). The *VGA-classifier* is found to be able to automatically evolve a proper value of H, while providing reasonably good region partitioning. In other words, this non-parametric classifier, unlike k-NN where k needs to be supplied,

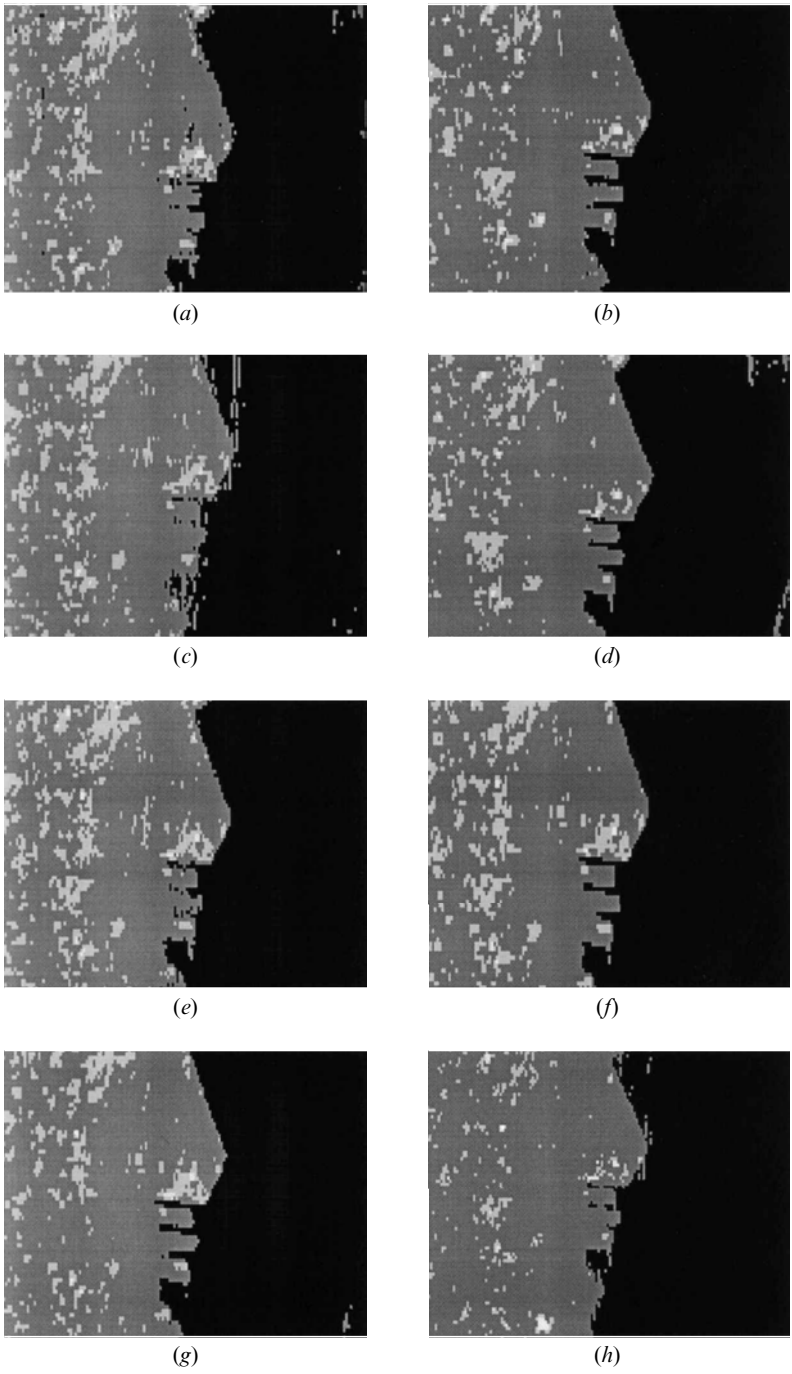

Figure 14. Classified IRS image of Bombay (zooming the dockyard only) using (*a*) GA classifier, $H=10$, (*b*) GA classifier, $H=15$, (*c*) GA classifier, $H=20$, (*d*) VGA classifier, $H_{\max}=15$, final value of $H=9$, (*e*) k-NN rule, $k=1$, (*f*) k-NN rule, $k=3$, (*g*) k-NN rule, $k=\sqrt{n}$, (*h*) Bayes' maximum likelihood classifier.

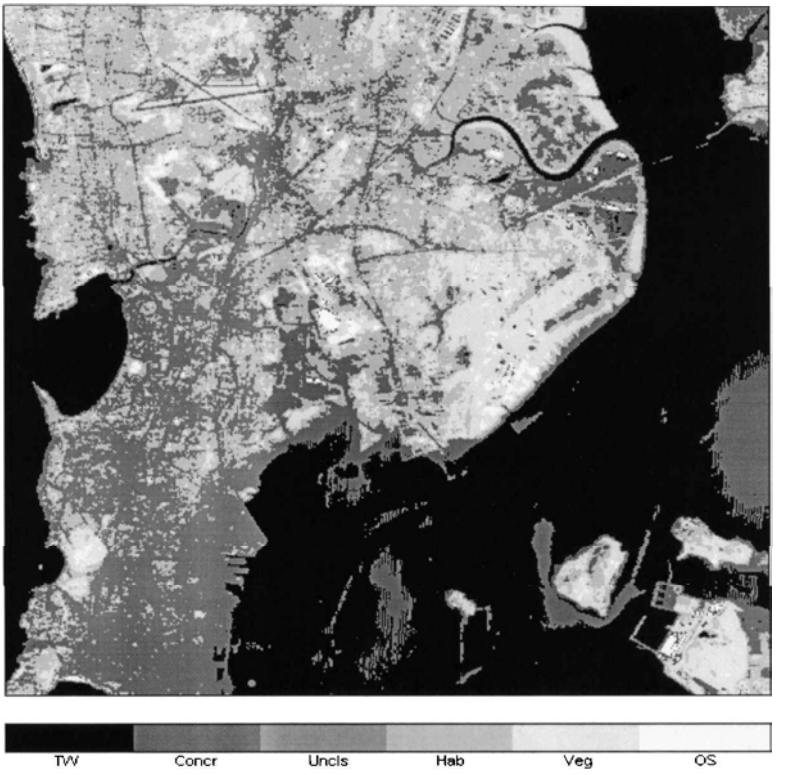

Figure 15. Classified IRS image of Bombay using *VGA-classifier*, $H_{\max} = 15$, final value of $H = 9$.

does not need the number of hyperplanes to be specified to model any kind of class boundaries. Moreover, unlike Bayes' maximum likelihood classifier where some assumption regarding the class distribution functions needs to be made, no underlying distribution of the dataset is assumed for the *VGA-classifier*. However, it may be noted that the *VGA-classifier* is generally found to take a significant amount of time for its proper training. For example, the time taken by the *VGA-classifier* when it was trained for 3000 iterations for the SPOT image of Calcutta on DEC-Alpha machine was 515.76 s. The problem is compounded by the fact that no appropriate criterion for terminating GAs is available in the literature. The time for testing taken by the *k*-NN classifier is also found to be significantly large especially for $k = \sqrt{n}$. For example, for the said image, the testing time for *k*-NN rule for $k = \sqrt{n}$ was 659.90 s. One may note that the time required for testing for the other classifiers is usually low. For example, these are 3.54 s and 2.06 s corresponding to the VGA classifier and Bayes' maximum likelihood classifier, respectively, for the Calcutta image.

Implementation of the GA-based classifiers for classifying the images necessitated a large value of H. This led to the problem of having to consider 2^H, a very large value, for the number of regions. This problem has been solved by noting that in practice, the number of regions can never exceed the number of training points n.

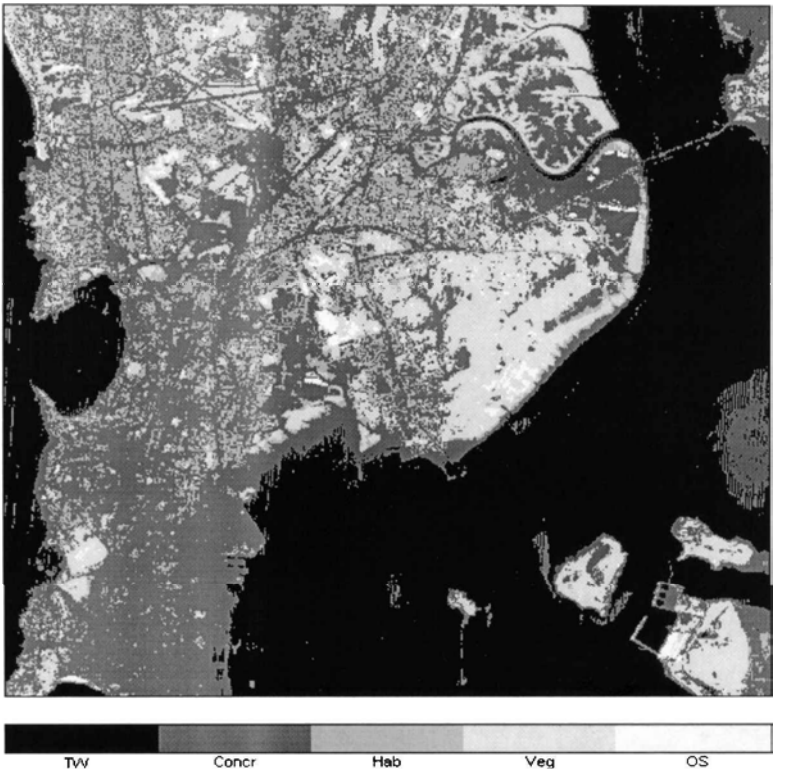

Figure 16. Classified IRS image of Bombay using Bayes' maximum likelihood classifier.

Appendix

k-*NN rule (Tou and Gonzalez 1974)*

The *k*-NN rule assigns a pattern $\mathbf{x}$ of unknown classification to class C_i if the majority of the k nearest neighbours of $\mathbf{x}$ belongs to class C_i. $\mathbf{x}_j$ is called the nearest neighbour of $\mathbf{x}$ if

$$D(\mathbf{x}_j, \mathbf{x}) = \min_l [D(\mathbf{x}_l, \mathbf{x})], \qquad l = 1, 2, ..., n \tag{A1}$$

where n denotes the total number of points, and D is any distance measure defined over the pattern space. The details on the *k*-NN rule along with the probability of error is available elsewhere (Fukunaga 1972).

Bayes' maximum likelihood classifier (Tou and Gonzalez 1974)

Let P_i denote the *a priori* probability and $p_i(\mathbf{x})$ denote the class conditional density corresponding to the class C_i. Assuming normal (Gaussian) distribution of patterns, with mean vector μ_i and covariance matrix Σ_i, the Gaussian density $p_i(\mathbf{x})$ may be written as

$$p_i(\mathbf{x}) = \frac{1}{(2\pi)_{N/2} |\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mu_i)' \Sigma_i^{-1} (\mathbf{x} - \mu_i) \right], \qquad i = 1, 2, ..., k \tag{A2}$$

where k denotes the total number of classes. Then, Bayes' maximum likelihood classifier assigns an unknown pattern $\mathbf{x}$ to class C_i if

$$D_i(\mathbf{x}) > = D_j(\mathbf{x}), \qquad j = 1, 2, ..., k, j! = i \tag{A3}$$

where $D_i(\mathbf{x})$ is defined as

$$D_i(\mathbf{x}) = \ln P_i - \frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(\mathbf{x}-\mu_i)'\,\Sigma_i^{-1}(\mathbf{x}-\mu_i), \qquad i = 1, 2, ..., k \tag{A4}$$

Multilayer perceptron (MLP) (Dayhoff 1990)

A multilayer perceptron (MLP) consists of several layers of simple neurons with full connectivity existing between neurons of adjacent layers. The number of nodes in the input and output layers correspond to the number of features and classes, respectively. Figure 1 shows an example of a three-layer MLP which consists of an input layer (layer 0), one hidden layer (layer 1) and an output layer (layer 2).

The neurons in the input layer serve the purpose of fanning out the input values to the neurons of layer 1. Let

$$w_{ji}^{(l)}, \qquad l = 1, 2 \tag{A5}$$

represent the connection weight on the link from the ith neuron in layer $l-1$ to the jth neuron in layer l. Let $\theta_j^{(l)}$ represent the threshold of the jth neuron in layer l. The total input, $x_j^{(l)}$, received by the jth neuron in layer l is given by

$$x_j^{(l)} = \sum_i y_i^{(l-1)} w_{ji}^{(l)} + \theta_j^{(l)} \tag{A6}$$

where $y_i^{(l-1)}$ is the output of the ith neuron in layer $l-1$. For the input layer

$$y_i^{(0)} = x_i \tag{A7}$$

where x_i is the ith component of the input vector. For the other layers

$$y_i^{(l)} = f(x_i^{(l)}) \qquad l = 1, 2 \tag{A8}$$

Several functional forms like threshold logic, hard limiter, sigmoid can be used for $f(\cdot)$.

Backpropagation (BP) is a commonly used learning algorithm, where the least mean square error of the network output is computed, and this is propagated in a top-down manner (i.e. from the output side) in order to update the weights. The error is computed as the difference between the actual and the desired output when a known input pattern is presented to the network. A gradient descent method along the error surface is used in BP. After training, the MLP decides the class of an unknown input to the one for which the corresponding output is the maximum.

References

Bandyopadhyay, S., Murthy, C. A., and Pal, S. K., 1998a, Pattern classification using genetic algorithms: determination of H. *Pattern Recognition Letters*, **19**, 1171–1181.

Bandyopadhyay, S., Pal, S. K., and Maulik, U., 1998b, Incorporating chromosome differentiation in genetic algorithms. *Information Sciences*, **104**, 293–319.

Congalton, R. G., Oderwald, R. G., and Mead, R. A., 1983, Assessing Landsat classification accuracy using discrete multivariate analysis statistical techniques. *Photogrammetric Engineering and Remote Sensing*, **49**, 1671–1678.

Cover, T. M., and Hart, P. E., 1967, Nearest neighbour pattern classification. *IEEE Transactions on Information Theory*, **13**, 21–27.

Davis, L. (Ed.), 1991, *Handbook of Genetic Algorithms* (New York: Van Nostrand Reinhold).

Dayhoff, J. E., 1990, *Neural Network Architecture: An Introduction* (New York: Van Nostrand Reinhold).

Fukunaga, K., 1972, *Introduction to Statistical Pattern Recognition* (New York: Academic Press).

Gelsema, E. S. (Ed.), 1995, *Special Issue on Genetic Algorithms. Pattern Recognition Letters*, **16** (Amsterdam: Elsevier).

Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimisation and Machine Learning* (New York: Addison-Wesley).

Goldberg, D. E., Deb, K., and Korb, B., 1989, Messy genetic algorithms: motivation, analysis, and first results. *Complex Systems*, **3**, 493–530.

NRSA, 1986, IRS data users handbook. Document No. IR/NRSA/NDC/HB-01/86, NRSA, Hyderabad, India.

Pal, A., 1990, On a class of stochastic approximation-type parameter-learning algorithms for pattern recognition. PhD thesis, Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta, India.

Pal, A., 1993, Some applications of GGA for automatic learning of class parameters in the presence of wrong samples. *Information Sciences*, **67**, 189–208.

Pal, S. K., Bandyopadhyay, S., and Murthy, C. A., 1998, Genetic algorithms for generation of class boundaries. *IEEE Transactions on System, Man and Cybernetics*, **28**, 816–828.

Pal, S. K., and Wang, P. P. (Eds), 1996, *Genetic Algorithms for Pattern Recognition* (Boca Raton: CRC Press).

Richards, J. A., 1993, *Remote Sensing Digital Image Analysis: An Introduction* (New York: Springer).

Rosenfield, G. H., and Fitzpatrick-Lins, K., 1986, Coefficient of agreement as a measure of thematic classification accuracy. *Photogrammetric Engineering and Remote Sensing*, **52**, 223–227.

Schriever, J. R., and Congalton, R. G., 1995, Evaluating seasonal variability as an aid to cover-type mapping from Landsat thematic mapper data in the northeast. *Photogrammetric Engineering and Remote Sensing*, **3**, 321–327.

Tou, J. T., and Gonzalez, R. C., 1974, *Pattern Recognition Principles* (Reading: Addison-Wesley).