# Pixel Classification Using Variable String Genetic Algorithms with Chromosome Differentiation

Sanghamitra Bandyopadhyay, *Member, IEEE* and Sankar K. Pal, *Fellow, IEEE*

***Abstract*—The concept of chromosome differentiation, commonly witnessed in nature as male and female sexes, is incorporated in genetic algorithms with variable length strings for designing a nonparametric classification methodology. Its significance in partitioning different landcover regions from satellite images, having complex/overlapping class boundaries, is demonstrated. The classifier is able to evolve automatically the appropriate number of hyperplanes efficiently for modeling any kind of class boundaries optimally. Merits of the system over the related ones are established through the use of several quantitative measure.**

***Index Terms*—Genetic algorithms, hyperplane fitting, pattern recognition, quantitative indices, remote sensing images.**

## I. Introduction

CLASSIFICATION of pixels for partitioning different landcover regions is an important problem in the realm of satellite imagery. Satellite images usually have a large number of classes with overlapping and nonlinear class boundaries. Fig. 1 shows, as a typical example, the complexity in scatter plot of 932 points belonging to seven classes taken from the Systeme Probatoire d'Observation de la Terre (SPOT) image of a part of the city of Calcutta. Therefore, for appropriate modeling of such nonlinear and overlapping class boundaries, the utility of an efficient search technique 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.

Genetic algorithms (GAs) [1] are randomized search and optimization 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. The utility of GAs in solving problems that are large, multimodal and highly complex has been demonstrated in several areas [2]. Since satellite images usually have highly nonlinear and overlapping class boundaries, application of GAs for searching for the appropriate ones, particularly under nonparametric conditions (i.e., without assuming class distributions and *a priori* probabilities), seems appropriate and natural.

In the present article, such an attempt is made by demonstrating the effectiveness of a GA-based classifier, called the variable string length GA with chromosome differentiation (VGACD)-classifier, in partitioning different landcover regions. In designing the VGACD-classifier, the concepts of variable length strings in GAs (VGAs) [3] and chromosome differentiation into two classes, male (M) and female (F), have been integrated for approximating the class boundaries of a given training data set nonparametrically by an optimum number of hyperplanes such that the number of misclassified points is minimized. Unlike the conventional GAs, in VGACD, the length of a string is not fixed. Moreover, two classes of chromosomes exist in the population. The crossover, implementing a kind of restricted mating, and mutation operators are accordingly defined. The fitness function rewards a string with smaller number of misclassified samples, as well as smaller number of hyperplanes. A comparison, in terms of several quantitative measures [4] and visual quality of the classified images, of the VGACD-classifier with VGA-classifier, i.e., the one incorporating variable string lengths but without chromosome differentiation, and those based on the k-NN rule and the Bayes maximum likelihood (ML) ratio is provided for SPOT image of a part of the city of Calcutta.

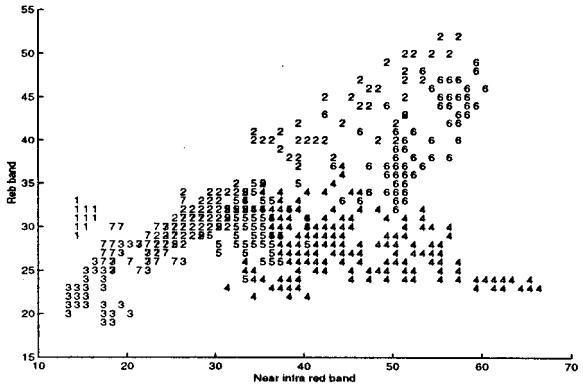


Fig. 1. Scatter plot for a training set of SPOT image of Calcutta containing seven classes (1,$\cdots$,7).

## II. Description of the VGACD-Classifier

### A. Principle of Hyperplane Fitting

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

$$\beta_1 x_N + \beta_2 x_{N-1} + \cdots + \beta_N x_1 = d \tag{1}$$

where $\beta_i = \cos\alpha_{N-i} \sin\alpha_{N-(i-1)} \cdots \sin\alpha_{N-1}$. Here, $\alpha_j$ is the angle that the projection of the unit normal in the $(X_1 - X_2 - \cdots - X_{j+1})$ space makes with the $X_{j+1}$ axis. Since $\alpha_0 =$

Manuscript received January 11, 2000; revised May 3, 2000.

The authors are with Machine Intelligence Unit, Indian Statistical Institute, Calcutta, India (e-mail: sanghami@www.isical.ac.in; sankar@www.isical.ac.in).

Publisher Item Identifier S 0196-2892(01)01172-X.

0, so the $N$-tuple $\langle \alpha_1, \alpha_2, \cdots \alpha_{N-1}, d \rangle$ specifies a hyperplane uniquely in $N$-dimensional space. An appropriate binary encoding is adopted for these $N$ parameters corresponding to a hyperplane. For details, the reader may refer to [5].

In variable string length GAs, a chromosome encodes the $N$ parameters of several hyperplanes, whose number may vary in the range [1,$H_{\max}$], where $H_{\max}$ is chosen to be suitably high. Note that the choice of $H_{\max}$ will affect the convergence time of the algorithm, not its classification performance. This is in contrast to the earlier fixed string length version of the genetic classifier [5], where the number of hyperplanes $H$ (not just an estimate of the upper bound) had to be specified *a priori*. The performance of the classifier depended heavily on the value of $H$ chosen, since an overestimate led to close or overfitting of the training data, with resultant loss of generalization capability.

### *B. Incorporation of Chromosome Differentiation in Variable String Length GAs*

In conventional GAs, since no restriction is placed upon the selection of mating pair for crossover operation, often chromosomes with similar characteristics are mated. Therefore, no significant new information is gained out of this process, and the result is wastage of computational resources. In VGACD, we try to alleviate this problem by distinguishing the chromosomes into two categories, M and F (determined by two additional bits called class bits), and therefore two populations. These two populations are initially generated in such a way that they are maximally apart. Crossover is restricted only between individuals from these two populations. Since, as a result of this process, we allow crossover only between dissimilar individuals, a higher level of diversity is likely to be introduced and subsequently maintained in the system. This will in turn result in faster information interchange between the chromosomes, and therefore, faster convergence of the algorithm. Interestingly, an analogy of this concept of chromosome differentiation exists in natural genetic systems, in the widely witnessed phenomenon of male and female sexes.

As mentioned above, two additional bits called class bits are used to distinguish the chromosomes into two classes, M and F. If the class bits contain either 01 or 10, the corresponding chromosome is called an M chromosome, and if it contains 00, the corresponding chromosome is called an F chromosome. These bits are not allowed to assume the value 11 (this is in analogy with the X and Y chromosomes in natural genetic systems, where XY/YX indicates male, while XX indicates female). The remaining bits are called data bits, which may be either 1, 0, or # (do not care). The data bits encode the parameters of $H_i$ hyperplanes, where $1 \leq H_i \leq H_{\max}$. The structure of a chromosome in VGACD is shown in Fig. 2.

*Population Initialization:* Two separate populations, one containing the M chromosomes (M population) and the other containing the F chromosomes (F population), are maintained over the generations. The sizes of these two populations, $p_m$ and $p_f$, respectively, may vary. Let $p_m + p_f = p$, where $p$ is fixed (equivalent to the population size of conventional GA). Initially, we considered $p_m = p_f = p/2$. The M population is first generated in such a way that the first two chromosomes

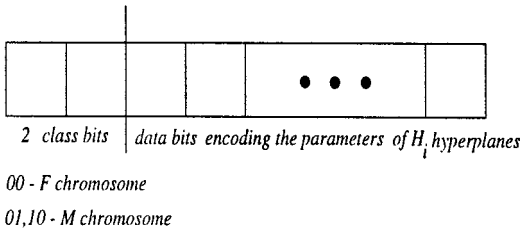


Fig. 2. Structure of a chromosome in GACD.

encode the parameters of 1 and $H_{\max}$ hyperplanes, respectively. The remaining chromosomes encode the parameters of $H_i$ hyperplanes where $1 \leq H_i \leq H_{\max}$. For these chromosomes, one of the two class bits, chosen randomly, is initialized to 0 and the other to 1. The data bits of the F chromosomes are initially generated in such a way that the hamming distance between the M and F populations (in terms of the data bits) is maximum. The hamming distance between two chromosomes $c_1$ and $c_2$, denoted by $h(c_1, c_2)$, is defined as the number of bit positions in which the two chromosomes differ. Hamming distance between two populations $P_1$ and $P_2$, denoted by $h(P_1, P_2)$, is defined as follows:

$$h(P_1, P_2) = \sum_i \sum_j h(c_i, c_j), \qquad \forall c_i \in P_1, \quad \forall c_j \in P_2.$$

*Fitness Computation:* Let $H_{\max}$ represent the maximum prespecified number of hyperplanes that may be required to model the decision boundary of a given data set and $H_i$ the number of hyperplanes encoded in chromosome $i$. Using the parameters of the hyperplanes encoded in a chromosome, the region in which each training pattern point lies is determined based on (1). A region is said to provide the demarcation for class $j$, if among the points that lie in this region, a majority belong to class $j$. Other points that lie in this region are considered to be misclassified. All ties are resolved arbitrarily. The misclassifications associated with all the regions (for these $H_i$ hyperplanes) are summed up to provide the total misclassification $miss_i$ for the string. If $n$ is the size of the training data, then the fitness of the $i$th string, $fit_i$, is defined as $fit_i = (n - miss_i) - \alpha H_i$, where $\alpha = 1/H_{\max}$. A string with zero hyperplane is defined to have zero fitness. Maximization of the fitness function ensures the minimization of, primarily, the number of misclassified points and then the number of hyperplanes.

*Genetic Operators:* Since the strings have variable length, the operators crossover and mutation are newly defined as follows.

Crossover: Two strings $i$ and $j$ having lengths $l_i$ and $l_j$ are selected from the M and F populations, respectively. Let $l_i \leq l_j$. Then string $i$ is padded with #s so as to make the two lengths equal. Conventional crossover like single point crossover, two point crossover [1] is now performed over these two strings with probability $\mu_c$. 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 either all the bits corresponding to it are defined (i.e., 0s and 1s) or all are

#s. Otherwise, i.e., if the bits corresponding to a hyperplanes are a combination of 1s, 0s, and #s, 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 probability $(1-(u/t))$. Thus, each hyperplane in the offspring becomes complete. Subsequently, the string is rearranged so that all the #s are pushed to the end. Each parent contributes one class bit to the offspring. Since the F parent can only contribute a 0 (its class bits being 00), the class of the child is primarily determined by the M parent, which can contribute a 1 (yielding an M child) or a 0 (yielding an F child) depending upon the bit position (among the two class bits) of the M parent chosen. This process is performed for both the offspring, whereby either two M or two F or one M and one F offspring will be generated. These are put in the respective populations.

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. Only the data bits, and not the class bits, are considered for mutation. The strings are first padded with #s such that the resultant length becomes equal to $l_{\max}$, where $L_{\max}$ (proportional to $H_{\max}$) is the maximum possible number of data bits in a string. Now for each defined bit position, it is determined whether conventional mutation [1] can be applied or not with probability $\mu_m$. Otherwise, the position is set to # with probability $\mu_{m_1}$. Each undefined position is set to a defined bit (randomly chosen) according to another mutation probability $\mu_{m_2}$.

Note that mutation may again 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 [3].

## III. Pixel Classification of *SPOT* Image

The $512 \times 512$ image considered in this experiment has three bands *viz.*, green band of wavelength 0.50–0.59 $\mu$m, red band of wavelength 0.61–0.68 $\mu$m, and near infrared band of wavelength 0.79–0.89 $\mu$m. The design set comprises 932 points belonging to seven classes that are extracted from the above image. 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). In the first part of the experiment, a percentage of these 932 points (30%, 50%, and 80%) are used for training, while the remaining are used for the purpose of testing. In the second part, the trained classifier (using 80% of the data set for training) is utilized for classifying the pixels in the $512 \times 512$ image.

The numbers of bits used to represent an angle and the perpendicular distance are 8 and 16, respectively. Roulette wheel selection is adopted to implement the proportional selection strategy for a population size of 20. Single point crossover is applied with a fixed crossover probability $\mu_c$. $\mu_{m_1}$ and $\mu_{m_2}$ are also kept fixed (note that we have experimented with several combination of these values, the results of which are reported in Table I). We have kept $\mu_{m_1} = \mu_{m_2}$ since we would like the chances of increasing or decreasing the number of hyperplanes in a chromosome to be same. Or, in other words, it would be undesirable to inflict any external bias to the algorithm in either direction. The conventional mutation operation is performed on a bit by bit basis for varying mutation probability values ($\mu_m$) in the range [0.015, 0.333], which was found to provide good performance in previous experiments [3], [5]. The algorithm is terminated if the population contains at least one string with no misclassified points. Otherwise, the algorithm is executed for 3000 generations.

TABLE I
Performance of VGACD-Classifier for some Typical GA Parameter Values. Here, "perc" = 30, and $\mu_m$ is Kept Variable in the Range [0.015, 0.333]

| $\mu_c$ | $\mu_{m_1}$ | $\mu_{m_2}$ | % Recognition Score |
|---|---|---|---|
| 0.6 | 0.2 | 0.2 | 76.98 |
| 0.7 | 0.01 | 0.01 | 82.16 |
| 0.8 | 0.1 | 0.1 | 83.38 |
| 0.8 | 0.2 | 0.2 | 80.48 |
| 0.8 | 0.01 | 0.01 | 81.40 |

TABLE II
Comparative Results in Terms of Recognition Score "recog" (%) and Kappa Values (%). Here, "perc" Indicates Percentage of Data Used for Training

| perc | *VGACD-classifier* | | *VGA-classifier* | | Bayes | | k-NN | |
|---|---|---|---|---|---|---|---|---|
| | recog | kappa | recog | kappa | recog | kappa | recog | kappa |
| 30 | 83.38 | 81.56 | 80.94 | 76.56 | 82.16 | 78.38 | 82.77 | 78.94 |
| 50 | 85.01 | 82.16 | 84.36 | 81.07 | 85.86 | 82.85 | 88.01 | 85.34 |
| 80 | 88.89 | 87.07 | 84.12 | 80.62 | 86.24 | 83.28 | 89.41 | 87.13 |

Table I shows the performance of the VGACD-classifier for some typical values of $\mu_c$, $\mu_{m_1}$, and $\mu_{m_2}$ when the algorithm is initiated from the same state (i.e., with the same chromosomes in the initial population). The mutation probability value is kept variable within the range [0.015, 0.333]. As can be seen from the table, the performance of the VGACD-classifier is best when $\mu_c = 0.8$ and $\mu_{m_1} = \mu_{m_2} = 0.1$. Results in the subsequent tables and figures are therefore shown corresponding to only the said parameter values.

The performance of the genetic classifiers is compared with those of the Bayes ML classifier (capable of handling overlapping classes) assuming normal distribution of the data set for each class with different covariance matrices and *a priori* probabilities for the classes and the k-NN classifier (well known for generating highly nonlinear boundaries through piecewise linear segments) with k= $\sqrt{n}$. 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 [6]. One such value of k for which the limiting conditions are satisfied is $\sqrt{n}$. Also, as demonstrated later in Fig. 4(d)–(f), the performance of

the k-NN rule is found to improve with the value of k, being the best for k= $\sqrt{n}$ for the SPOT data.

Table II presents the comparative results of the different classifiers for different percentages of training data, and $H_{\max} = 15$ for the genetic classifiers. Note that the choice of $H_{\max}$, as long as it is sufficiently high, is not crucial for the performance of the genetic classifiers. Results are provided in terms of two measures, the percentage recognition scores and kappa values [4]. The classwise recognition scores for the different classifiers when perc = 80% are shown in Table III. Table IV presents, as an example, the confusion matrix obtained for the 80% training data corresponding to the Bayes classifier.

As seen from Table II, the performance of the VGACD-classifier is always better than that of the VGA-classifier, irrespective of the percentage of data used for training. This indicates that incorporation of the concept of chromosome differentiation leads to an improvement in performance of the variable string length GA classifier as well. Overall, the performance of the VGACD-classifier is found to be better than or comparable to that of the k-NN rule for 30% and 80% training data, while for 50% training data, the k-NN rule outperformed all other classifiers.

From the classwise scores shown in Table III, it is found that the VGACD-classifier recognizes the different classes consistently with a high degree of accuracy. On the contrary, the other classifiers can recognize some classes very nicely, while for some other classes their scores are much poorer. An example, the Bayes ML classifier provides 100% accuracy for the classes TW and PW, its scores for classes B/R and Hab. are only 36.36% and 49.55%, respectively.

Fig. 3 demonstrates the variation of the number of points misclassified by the best chromosome with the number of generations for the VGACD-classifier and VGA-classifier (when $perc = 80$). As is seen from Fig. 3, both the classifiers consistently reduce the number of misclassified points. The best value is obtained just after the 1700 and 1900 iterations for VGACD-classifier and VGA-classifier, respectively. Incorporating the concept of chromosome differentiation therefore helps in faster convergence of the algorithm, since any given value of the number of misclassified points is achieved earlier by the VGACD-classifier than the VGA-classifier (if at all).

Fig. 4(a)–(f) provide, as an illustration, the results obtained by the different classifiers (including results for k-NN rule with k = 1 and 3) for partitioning the 512 × 512 image by zooming a characteristic portion of the image containing the race course (a triangular structure). Here, 80% of the design set is used for training. Table V provides the classwise and overall values of an index, called $\beta$ [7], which is the ratio of the total variation and within class variation. The higher this value is, the better the performance of the classifier.

As seen from the figures, although all the classifiers (with the exception of k = 1 for k-NN rule) are able to identify the race course, only the VGACD-classifier and the VGA-classifier are able to identify a triangular lighter outline (which is an open space, corresponding to the tracks) within the race course properly. The performance of k-NN rule is found to gradually improve with the value of k, being the best for k = $\sqrt{n}$. On inspection of the full classified images, it was found that the Bayes ML classifier tends to overestimate the roads in the image. This is also reflected

TABLE III
COMPARATIVE CLASSWISE RECOGNITION SCORES (%) FOR "perc" = 80.

| Class | Recognition Scores | | | |
|---|---|---|---|---|
| | *VGACD-classifier* | *VGA-classifier* | Bayes | k-NN |
| TW | 100.00 | 100.00 | 100.00 | 100.00 |
| PW | 93.93 | 89.55 | 100.00 | 96.97 |
| Concr | 80.83 | 91.42 | 91.42 | 92.53 |
| Veg | 92.29 | 85.31 | 88.45 | 94.21 |
| Hab | 70.73 | 43.75 | 49.55 | 62.50 |
| OS | 94.73 | 87.27 | 89.47 | 94.74 |
| B/R | 72.72 | 21.35 | 36.36 | 45.45 |

TABLE IV
CONFUSION MATRIX FOR TRAINING DATA OF SPOT IMAGE OF CALCUTTA OBTAINED USING BAYES ML CLASSIFIER

| | | Recognized as | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | TW | PW | Concr | Veg | Hab | OS | B/R |
| Actual | TW | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| | PW | 0 | 115 | 11 | 0 | 2 | 8 | 12 |
| | Concr | 0 | 11 | 116 | 0 | 0 | 0 | 4 |
| | Veg | 0 | 0 | 0 | 173 | 24 | 3 | 0 |
| | Hab | 0 | 7 | 0 | 10 | 42 | 0 | 1 |
| | OS | 0 | 5 | 0 | 5 | 0 | 64 | 0 |
| | B/R | 0 | 11 | 8 | 0 | 0 | 0 | 22 |

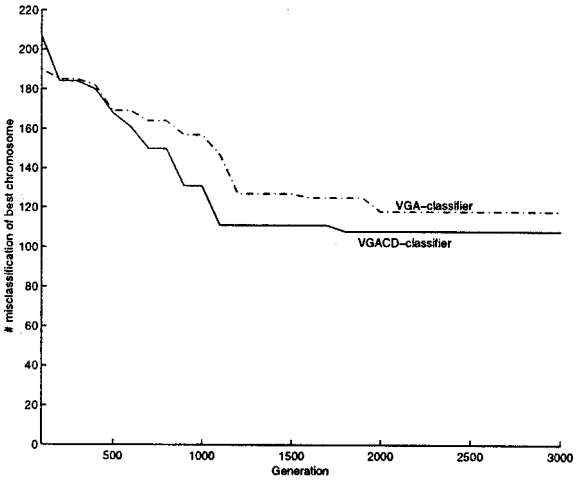


Fig. 3. Variation of the number of points misclassified by the best chromosome with generations for VGACD-classifier and VGA-classifier.

in Table V, where this class is seen to have low beta value and hence low homogeneity corresponding to the Bayes classifier. On the other hand, the VGA-classifier tends to confuse between the classes bridges and roads (B/R) and pond water (PW). It was revealed on investigation that a large amount of overlap exists between the classes Concr and B/R on the one hand (in fact, the latter class has been extracted from the former), and PW and B/R on the other hand. This is also evident from Table IV, where 27% and 20% of the points belonging to the class B/R went to class PW

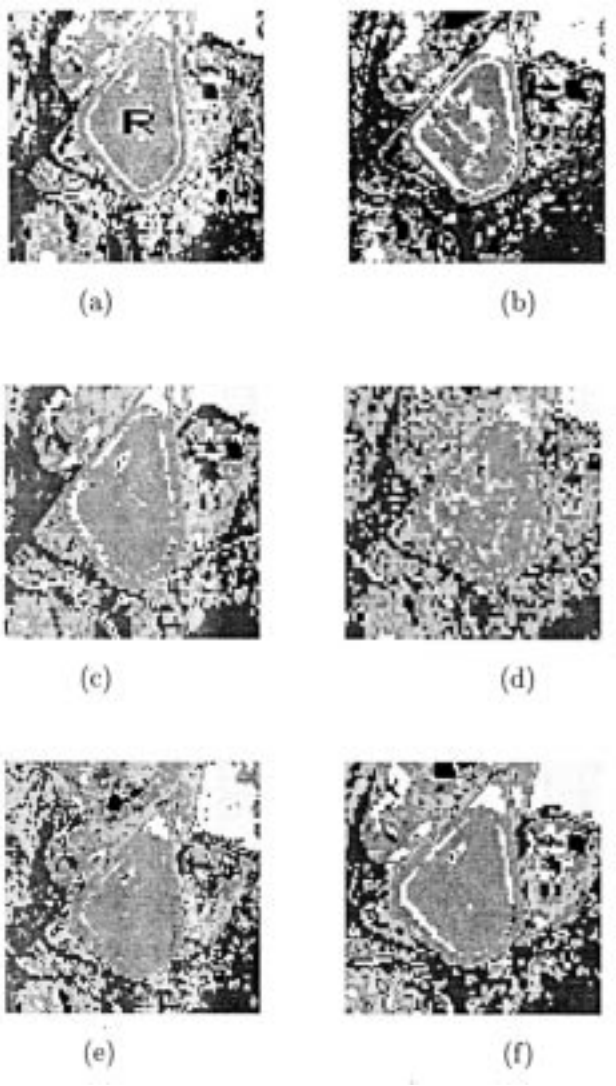


Fig. 4. Classified SPOT image of Calcutta (zooming the race course, represented by "R" on the first figure, only) using (a) VGACD-classifier, $H_{\max} = 15$, final value of $H = 13$, (b) *VGA-classifier*, $H_{\max} = 15$, final value of $H = 10$, (c) Bayes ML classifier, (d) k-NN rule, $k = 1$, (e) k-NN rule, $k = 3$, (f) k-NN rule, k= $\sqrt{n}$.

TABLE V
CLASSWISE AND OVERALL $\beta$ VALUES FOR DIFFERENT CLASSIFIERS

| Class | *VGACD-classifier* | *VGA-classifier* | Bayes | k-NN |
|---|---|---|---|---|
| TW | 96.01 | 47.68 | 113.06 | 73.65 |
| PW | 4.66 | 3.15 | 16.38 | 14.01 |
| Concr | 1.32 | 1.15 | 1.22 | 1.25 |
| Veg | 2.88 | 3.32 | 2.07 | 2.69 |
| Hab | 1.50 | 1.59 | 1.50 | 1.31 |
| OS | 10.33 | 8.03 | 8.82 | 11.84 |
| B/R | 11.60 | 22.62 | 5.19 | 8.99 |
| Overall | 3.1932 | 2.8738 | 2.5847 | 2.9061 |

and Concr., respectively. These problems were not evident for the case of the VGACD-classifier. Also note that the overall $\beta$ value is largest for this classifier and is worst for the Bayes ML classifier (from Table V), thereby indicating the significant superiority of the former. The individual $\lambda$B values for all the classes consistently show that the regions with the highest and lowest homogeneity are the classes TW and Concr., respectively.

## IV. CONCLUSIONS

The concepts of variable string length and chromosome differentiation in GAs have been integrated for the development of a nonparametric classifier which can approximate well any kind of highly nonlinear boundaries (e.g., in remote sensing images) by evolving automatically an optimum number of hyperplanes. Unlike the k-NN rule, where k needs to be supplied, the genetic classifiers with variable length strings do not require the number of hyperplanes to be specified to model various landcover boundaries, while providing good region partitioning. Moreover, unlike the Bayes ML classifier, no assumption on class distributions is needed here.

An interesting analogy of the concept of chromosome differentiation can be found in the sexual differentiation found in nature. The class bits in VGACD are chosen in tune with the way the X and Y chromosomes help to distinguish the two sexes. Because the initial M and F populations are generated so that they are at a maximum hamming distance from each other, and crossover is restricted only between individuals of these two classes, VGACD appears to be able to strike a greater balance between exploration and exploitation of the search space. This is in contrast to its asexual version. It is because of this fact that the former is consistently seen to outperform the latter.

With regard to timing requirements, it may be noted that the genetic classifiers take significantly large amount of time during training. However, the time taken during testing is very small. On the contrary, the k-NN rule (with k = $\sqrt{n}$) takes significant amount of time for testing, while for the Bayes classifier both the training and testing times are quite small. As an illustration, the VGA-classifier took 515.76 s during training on a DEC-Alpha machine (when 3000 iterations were executed). Note that the problem is compounded by the fact that no appropriate criterion for terminating GAs is available in the literature. The k-NN rule took 659.90 s when it was tested on the full SPOT image of Calcutta, whereas for the VGA-classifier and the Bayes ML classifier these values were 3.54 s and 2.06 s, respectively.

## REFERENCES

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[2] S. K. Pal and P. P. Wang, Eds., *Genetic Algorithms for Pattern Recognition*. Boca Raton, FL: CRC, 1996.
[3] S. Bandyopadhyay, C. A. Murthy, and S. K. Pal, "Pattern classification using genetic algorithms: Determination of H," *Pattern Recognit. Lett.*, vol. 19, no. 13, pp. 1171–1181, 1998.
[4] G. H. Rosenfield and K. Fitzpatrick-Lins, "Coefficient of agreement as a measure of thematic classification accuracy," *Photogramm. Eng. Remote Sensing*, vol. 52, pp. 223–227, 1986.
[5] S. K. Pal, S. Bandyopadhyay, and C. A. Murthy, "Genetic algorithms for generation of class boundaries," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 816–828, Dec. 1998.
[6] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1972.
[7] S. K. Pal, A. Ghosh, and B. Uma Shankar, "Segmentation with remotely sensed images with fuzzy thresholding, and quantitative evaluation," *Int. J. Remote Sensing*, vol. 21, pp. 2269–2300.

**Sanghamitra Bandyopadhyay** (M'99) received the B.S. degree in physics and computer science from the University of Calcutta, Calcutta, India, in 1988 and 1991, respectively, the M.S. degree in computer science from the Indian Institute of Technology, Kharagpur, India, in 1993, and the Ph.D degree in computer science from the Indian Statistical Institute, Calcutta, India, in 1998.

She was with the Los Alamos National Laboratory, Los Alamos, NM, in 1997 as a Graduate Research Assistant and the University of New South Wales, Sydney, Australia, as a Postdoctoral Fellow. Her research interests include evolutionary computation, pattern recognition, parallel processing and interconnection networks.

Dr. Bandyopadhyay is the first recipient of the Dr. Shanker Dayal Sharma Gold Medal and the Institute Silver Medal for being judged the best all round Postgraduate Performer in 1994. She received the Indian National Science Academy (INSA) and the Indian Science Congress Association (ISCA) Young Scientist Awards in 2000.

**Sankar K. Pal** (M'81–SM'84–F'93) received the M.Tech. and Ph.D. degrees in radio physics and electronics in 1974 and 1979, respectively, from the University of Calcutta, Calcutta, India, and the Ph.D. in electrical engineering along with the DIC degree from Imperial College, University of London, London, U.K.

From 1986 to 1987, he was with the University of California, Berkeley, and the University of Maryland, College Park, as a Fulbright Postdoctoral Visiting Fellow, from 1990 to 1992, he was with NASA Johnson Space Center, Houston, TX, and in 1994, he was a Guest Investigator under the NRC-NASA Senior Research Associateship Program. He was also a Visiting Professor with the Hong Kong Polytechnic University, Hong Kong, in 1999. He is a Distinguished Scientist and Founding Head of the Machine Intelligence Unit, Indian Statistical Institute, Calcutta. His research interests includes pattern recognition, image processing, soft computing, neural nets, genetic algorithms, and fuzzy systems. He is co-author of seven books, including *Fuzzy Mathematical Approach to Pattern Recognition*, (NY: Wiley, 1986) and *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing* (NY: Wiley, 1999). He is an Associate Editor of *Pattern Recognition Letters*, *Neurocomputing*, *Applied Intelligence*, *Information Sciences*, *Fuzzy Sets and Systems*, and *Fundamenta Informaticae*. He is also a Member, Executive Advisory Editorial Board, of *International Journal of Image and Graphics* and *International Journal of Approximate Reasoning*

Prof. Pal has received the 1990 S. S. Bhatnagar Prize (which is the most coveted award for a scientist in India), the 1993 Jawaharlal Nehru Fellowship, the 1993 Vikram Sarabhai Research Award, the 1993 NASA Tech Brief Award, the 1994 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award, the 1995 NASA Patent Application Award, the 1997 IETE Ram Lal Wadhwa Gold Medal, the 1998 Om Bhasin Foundation Award, the 1999 G. D. Birla Award for Scientific Research, and the 14th Khwarizmi International Award, in 2000. From 1997 to 1999, he served as a Distinguished Visitor of IEEE Computer Society (USA) for the Asia-Pacific Region. He is a Fellow of the Third World Academy of Sciences, Italy, and all four National Academies for Science/Engineering in India. He is an Associate Editor, IEEE TRANSACTIONS ON NEURAL NETWORKS (1994–1998). He is a Member, Executive Advisory Editorial Board, of IEEE TRANSACTIONS ON FUZZY SYSTEMS, and is a Guest Editor of many journals, including IEEE COMPUTER.