Journal
of The
Franklin Institute

Journal of the Franklin Institute 336 (1999) 387–422

Theoretical performance of genetic pattern classifier

S. Bandyopadhyay, C.A. Murthy, Sankar K. Pal*

Machine Intelligence Unit, Indian Statistical Institute, 203 B.T. Road, Calcutta 700 035, India

Accepted 20 May 1998

Abstract

An investigation is carried out to formulate some theoretical results regarding the behavior of a genetic-algorithm-based pattern classification methodology, for an infinitely large number of training data points n, in an N-dimensional space $\Re^N$. It is proved that for $n \to \infty$, and for a sufficiently large number of iterations, the performance of this classifier (when hyperplanes are considered to generate class boundaries) approaches that of the Bayes classifier, which is the optimal classifier when the class distributions and the a priori probabilities are known. It is shown that the optimum number of hyperplanes generated by the proposed classifier is equal to that required to model the Bayes decision boundary when there exists only one partition of the feature space that provides the Bayes error probability. Extensive experimental results on overlapping data sets following triangular and normal distributions with both linear and non-linear class boundaries are provided that conform to these claims. The claims also hold good when circular surfaces are considered as constituting elements/segments of boundaries. It is also shown experimentally that the variation of recognition score with a priori class probability for both the classifiers is similar.

1. Introduction

Genetic algorithms (GAs) [1, 2] are randomized, robust and efficient search algorithms that are modeled on the principles of natural genetics and evolution. These algorithms are known to provide near-optimal solutions in large and highly complex, multimodal search spaces where they utilize domain-specific knowledge, in the form

*Corresponding author. Tel.: 009133 577 8085 3100; Fax: 009133 577 6680 6925; E-mail: sankar@isical.ac.in

of objective function, to perform a directed random search. They are gradually finding applications in a wide variety of fields like image processing and pattern recognition [3], job-shop scheduling [4], VLSI design, machine learning, classifier systems, etc. [5, 6].

Recently, an application of GA has been reported in the area of supervised pattern classification in $\Re^2$ [3]. Here GA is used for the appropriate placement of H lines (fixed a priori) in the feature space such that one or more regions are associated with an unique class. Points (of the training data set) of other classes lying in this region are considered to be misclassified. The H lines are placed in such a way that total misclassification of the training data points is minimized. The search space for the lines is restricted to the rectangle enclosing the training points. The parameters of these H lines are encoded in a chromosome (or string). The fitness of a string is defined as the number of training data points properly classified by the string. Conventional operators like *selection*, *crossover* and *mutation* are applied over a fixed number of generations or till a termination criterion is achieved. Let us subsequently refer to this GA-based classifier as *GA-classifier*.

Since an a priori knowledge of the value of H is difficult to acquire, it is frequently overestimated. Consequently, the resultant decision boundary may contain some redundant hyperplanes, removal of which will not affect the classification capability of the *GA-classifier*. A scheme for automatic deletion of such redundant hyperplanes is described in [7] along with an extension of the classifier in N-dimensional space $\Re^N$. Note that a characteristic feature of the *GA-classifier* is that it utilizes the decision boundary for performing classification. This is in contrary to the conventional pattern classification techniques where the decision boundaries are obtained as a consequence of the decision-making process.

There have been many attempts [8] for learning rules of classification. Some of them use decision trees [9–11]. Recently, a procedure has been suggested by integrating genetic algorithms with similarity-based learning for acquiring rules for classification [12]. Whereas these methods have used the rule-based approach to classify patterns, the proposed *GA-classifier* uses the decision boundaries directly for classification. There is another similar attempt [13] for classification which involves placement of a number of ellipsoids for generating class boundaries. But no theoretical analysis of the performance was provided.

In this article, we provide a theoretical investigation of the performance of the aforesaid *GA-classifier*. This mainly includes establishing the relation between Bayes classifier and the *GA-classifier*. (It may be mentioned here that *Bayes classifier* [14] is one of the most widely used statistical pattern classifiers which provides optimal performance from the standpoint of error probabilities in a statistical framework. It is known to be the best classifier when the class distributions and the a priori probabilities are known. Consequently, the desirable property of any classifier is that it should approximate or approach the Bayes classifier under limiting conditions. Such an investigation was performed in [15] to show that the MLP, when trained as a classifier using back-propagation, approximates the Bayes optimal discriminant function.)

It has been proved in this article that as the number of training data points (n) and the number of iterations of GA go to infinity, the error rate of the *GA-classifier* (defined as the ratio of the number of misclassified points to the number of training data points) will be less than or equal to the error probability of Bayes classifier. It has also been shown theoretically that under limiting conditions, the number of hyperplanes found by the *GA-classifier* to constitute the decision boundary will be equal to the optimum number of hyperplanes (i.e., which provides the Bayes decision boundary) if exactly one partition provides the Bayes error probability. Otherwise, the number of hyperplanes found by the *GA-classifier* will be greater than or equal to the optimum value.

The theoretical findings have also been experimentally verified on a number of training data sets following triangular and normal distribution having both linear and non-linear boundaries. Performance on independent test data sets has also been studied. Experiments have been conducted using different values of H. Instead of using hyperplanes, circular surfaces have also been considered as constituting the decision boundary. The generalization capability of the classifier has been studied as a function of the class a priori probabilities (for two class problems). The empirical findings show that as the training data size (n) increases, the performance of the *GA-classifier* approaches that of Bayes classifier for all the data sets.

The article is organized as follows: Section 2 describes the basic features and principles of genetic algorithms. Section 3 gives a brief outline of the *GA-classifier*. In Section 4 we present a theoretical investigation to find a relationship between the *GA-classifier* and Bayes classifier. A critical discussion of the proof is also given in the same section. This is followed by a description of the process of redundancy elimination along with the associated study of the relationship between the optimal number of hyperplanes required to model the Bayes boundary and those provided by the *GA-classifier* in Section 5. Section 6 contains the experimental results and their analysis. Finally, the discussion and conclusions are presented in Section 7.

2. Genetic algorithms: basic principles and features

The power of GAs lies in their ability to encode complex information and parameters of the search space in simple structures called *chromosomes* or *strings*, which are usually of a fixed length. Each chromosome encodes the parameters of a potential solution. An *objective function* is associated with each string which provides a mapping from the chromosomal space to the solution space. Based on this objective function, a fitness function is also associated with each string that provides a measure of the degree of goodness of the potential solution encoded in it.

GA starts from a collection of chromosomes (called *population*), which is initially created randomly. Various biologically inspired operators like *selection*, *crossover* and *mutation*, based on the Darwinian Principles of Survival of the Fittest and evolution are applied on these strings over a number of generations to yield the solution of the

problem. The basic steps of GAs are as follows:

```
Begin
        t = 0
        initialize population P(t)
        compute fitness P(t)
        repeat
                t = t + 1
                select P(t) from P(t − 1)
                crossover P(t)
                mutate P(t)
                compute fitness P(t)
        until termination criterion is ac  ieved
End
```

Selection is the process of allocating a number (zero or more) of copies to a string, depending on its fitness value, that go into the mating pool for further genetic operations. *Proportional selection strategy* is one commonly used selection mechanism where the number of copies that a string receives is proportional to its fitness value in the population.

Crossover is the process of combining the information of two parent chromosomes in order to produce potentially better solutions or offspring. In *single-point crossover*, one commonly used crossover technique, a crossover point is chosen randomly, and the portions of the parent strings lying to the right of the crossover point are swapped. In mutation, the other genetic operator, a position in a string is selected randomly, and the value in this position is changed. These two basic genetic operators, crossover and mutation, are performed stochastically.

Usually, the process of fitness computation, selection, crossover and mutation are performed for a number of iterations or generations, till a user-specified termination criterion is attained. In the *elitist* model of GA, assumed in this article, the best string seen upto the current generation is preserved in some location within or outside the population. A detailed discussion on GAs can be found in [1].

3. The GA-based classifier: a brief outline

In the realm of pattern classification in N dimensions, a fixed number (H) of hyperplanes is considered to constitute the decision boundary. Note that since each hyperplane provides two regions, H hyperplanes provide a maximum of 2^H regions. Hence for a k class problem, H needs to be greater than or equal to $\log_2 k$. Each chromosome encodes the parameters of these H hyperplanes. The fitness of a chromosome is characterized by the number of points correctly classified by the H hyperplanes encoded in it. The search space for the hyperplanes (which may be considered as candidates for the formation of the decision boundary) is restricted to the hyperrectangle formed around the training pattern points. The following discussion describes the different issues related to the application of GAs to pattern classification in $\Re^N$.

3.1. String representation

From elementary geometry, the equation of a hyperplane in N-dimensional space $(X_1 - X_2 - \cdots - X_N)$ 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.
$(x_1, x_2, \ldots, x_N)$: 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 $(X_1, -X_2 - \cdots - X_{N-1})$ space makes with the X_{N-1}-axis.
$\vdots$
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 $\langle \alpha_1, \alpha_2, \ldots, \alpha_{N-1}, d \rangle$ specifies a hyperplane in N-dimensional space. Each angle $\alpha_j, j = 1, 2, \ldots, N-1$, 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

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

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 $v_1 * \delta * 2\pi$.

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

$$(x_1^{ch_1}, x_2^{ch_2}, \ldots, x_N^{ch_N}),$$

where each ch_i, $i = 1, 2, \ldots, N$, 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

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

A hyperplane is designated as the *base hyperplane* with respect to a given orientation (i.e., for some $\alpha_1, \alpha_2, \ldots, \alpha_{N-1}$) if

(i) it has the same orientation,
(ii) it passes through one of the vertices of the enclosing rectangle,
(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 is the number of hyperplanes. These are initially generated randomly for a population of size *Pop*.

Note that we have used this recursive form of representation over the classical one, viz. $l_1x_1 + l_2x_2 + \cdots + l_Nx_N = d$, where $l_1, l_2, \ldots, l_N$ are known as the direction cosines. The latter representation involves a constraint equation, $l_1^2 + l_2^2 + \cdots l_N^2 = 1$. This, in turn, leads to the complicated issue of getting invalid or unacceptable solutions when the constraint equation is violated. However, the representation that we have chosen avoids this problem by being unconstrained in nature.

3.2. Fitness computation and genetic operations

A chromosome encodes the parameters of H hyperplanes as described earlier. Using these parameters, the region in which each training pattern point lies is determined from Eq. (1). A region is said to provide the demarcation for class i, if among the points that lie in this region, 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.

Roulette wheel selection [1] is adopted to implement the *proportional selection strategy*, where each string is allotted a slot of the roulette wheel subtending an angle, proportional to its fitness, at the center of the wheel. A random number is generated in the range 0 to 2π. A copy of a string goes into the mating pool if the random number falls in the slot corresponding to the string. Thus, the number of copies that each string gets is proportional to its fitness in the population. Elitism, where the best string seen upto the current generation is preserved in some location, is incorporated by replacing the worst string of the current generation with the best string seen up to the last generation.

Single-point crossover among two chromosomes is applied with a fixed crossover probability value of *cr_prob*. The mutation operation is performed on a bit by bit basis for a varying mutation probability value. In the initial stages of the algorithm it has a high value, which is first gradually decreased to a prespecified 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. This 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 optima. This problem is overcome by increasing the mutation probability to a high value, thereby making the search more random once again. The algorithm is terminated if the population contains at least one string with no misclassified points and there is no significant improvement in the average fitness of the population over subsequent generations. Otherwise the algorithm is executed for a fixed number of generations.

The variation of the performance of the above-mentioned *GA-classifier* with the size of the training data (n) and number of hyperplanes (H) was presented in [7]. The results showed that the generalization performance improved with n. Improvement of performance was also observed as H was increased upto a certain value. Beyond this, the generalization score decreased, although the training score increased. This demonstrated that an unnecessarily large number of hyperplanes led to overfitting of the training data set, which resulted in good performance during training, but was not beneficial for the test case. An extensive experimental comparison of the performance of the *GA-classifier* with those of Bayes maximum likelihood classifier (where the a priori probabilities and the covariance matrices were computed from the training data set assuming Gaussian distribution of patterns), k-NN classifier [14], and multilayered perceptron [16] (MLP) for a variety of artificial and real life, overlapping and non-overlapping data sets having non linear class boundaries was also made in [7]. Note that both k-NN classifier and MLP (with hard limiting non-linearities) provide piecewise linear boundaries. Interestingly, it was found in [7] that the recognition score of the *GA-classifier* was comparable to (sometimes better than) those of the Bayes maximum likelihood classifier for data sets having overlapping classes (where Bayes maximum likelihood classifier is known to perform well), and k-NN classifier and MLP for data sets having non-linear, non-overlapping regions (where k-NN and MLP are known to perform well).

4. Relationship with Bayes error probability

In this section we study the theoretical relationship between the *GA-classifier* and Bayes classifier. The mathematical notations and preliminary definitions are described first. This is followed by the claim that for $n \to \infty$ the performance of the *GA-classifier* will no way be worse than that of the Bayes classifier. Finally, some critical comments about the proof are mentioned.

Let there be k classes $C_1, C_2, \ldots, C_k$ with a priori probabilities $P_1, P_2, \ldots, P_k$ and class conditional densities $p_1(x), p_2(x), \ldots, p_k(x)$. Let the mixture density be

$$p(x) = \sum_{i=1}^{k} P_i p_i(x). \tag{2}$$

(According to the Bayes rule, a point is classified to class i iff

$$P_i p_i(x) \geqslant P_j p_j(x), \quad \forall j = 1, \ldots, k \quad \text{and} \quad j \neq i).$$

Let $X_1, X_2, \ldots, X_n, \ldots$ be independent and identically distributed (i.i.d.) N-dimensional random vectors with density $p(x)$. This indicates that there is a probability

space $(\Omega, \mathscr{F}, Q)$, where $\mathscr{F}$ is a σ field of subsets of Ω, Q is probability measure on $\mathscr{F}$, and

$$X_i : (\Omega, \mathscr{F}, Q) \to (\Re^N, B(\Re^N), P), \ \forall i = 1, 2, \ldots$$

such that

$$P(A) = Q(X_i^{-1}(A)) = \int_A p(x)\,\mathrm{d}x$$

$\forall A \in B(\Re^N)$ and $\forall i = 1, 2, \ldots$.
Here $B(\Re^N)$ is the Borel σ field of $\Re^N$.
Let

$$\mathscr{E} = \left\{ E : E = (S_1, S_2, \ldots, S_k),\ S_i \subseteq \Re^N,\ S_i \neq \emptyset \right.$$
$$\left. \forall i = 1, \ldots, k,\ \bigcup_{i=1}^{k} S_i = \Re^N,\ S_i \cap S_j = \emptyset,\ \forall i \neq j \right\}.$$

$\mathscr{E}$ provides the set of all partitions of $\Re^N$ into k sets as well as their permutations, i.e., if $E_1 = (S_1, S_2, S_3, \ldots, S_k) \in \mathscr{E}$, $E_2 = (S_2, S_1, S_3, \ldots, S_k) \in \mathscr{E}$, then $E_1 \neq E_2$. Note that $E = (S_{i_1}, S_{i_2}, \ldots, S_{i_k})$ implies that each S_{i_j}, $1 \leqslant j \leqslant k$, is the region corresponding to class C_j.

Let $E_0 = (S_{01}, S_{02}, \ldots, S_{0k}) \in \mathscr{E}$ be such that each S_{0i} is the region corresponding to the class C_i in $\Re^N$ and these are obtained by using Bayes decision rule. Then

$$a = \sum_{i=1}^{k} P_i \int_{S_{0i}^c} p_i(x)\,\mathrm{d}x \leqslant \sum_{i=1}^{k} P_i \int_{S_{1i}^c} p_i(x)\,\mathrm{d}x \tag{3}$$

$\forall E_1 = (S_{11}, S_{12}, \ldots, S_{1k}) \in \mathscr{E}$. Here a is the error probability obtained using the Bayes decision rule.

It is known from the literature that such an E_0 exists and it belongs to $\mathscr{E}$ because Bayes decision rule provides an optimal partition of $\Re^N$ and for every such $E_1 = (S_{11}, S_{12}, \ldots, S_{1k}) \in \mathscr{E}$, $\sum_{i=1}^{k} P_i \int_{S_{1i}^c} p_i(x)\,\mathrm{d}x$ provides the error probability for $E_1 \in \mathscr{E}$. Note that E_0 need not be unique.

Assumptions. Let H_o be a positive integer and let there exist H_o hyperplanes in $\Re^N$ which can provide the regions $S_{01}, S_{02}, \ldots, S_{0k}$. Let H_o be known a priori. Let the algorithm for generation of class boundaries using H_o hyperplanes be allowed to be executed for a sufficiently large number of iterations. It is known in the literature [17] that as the number of iterations goes towards infinity, an Elitist model of GA will certainly provide the optimal string.

Let $\mathscr{A} = \{A : A \text{ is a set consisting of } H_o \text{ hyperplanes in } \Re^N\}$. Let $A_0 \in \mathscr{A}$ be such that it provides the regions $S_{01}, S_{02}, \ldots, S_{0k}$ in $\Re^N$, i.e., A_0 provides the regions which are also obtained using the Bayes decision rule. Note that each $A \in \mathscr{A}$ generates several

elements of $\mathscr{E}$ which result from different permutations of the list of k regions. Let $\mathscr{E}_A \subseteq \mathscr{E}$ denote all possible $E = (S_1, S_2, \ldots, S_k) \in \mathscr{E}$ that can be generated from A. Let

$$G = \bigcup_{A \in \mathscr{A}} \mathscr{E}_A \text{; let}$$

$$Z_{iE}(\omega) = \begin{cases} 1 & \text{if } X_i(\omega) \text{ is misclassified when } E \text{ is used as a decision} \\ & \text{rule where } E \in G, \ \forall \omega \in \Omega, \\ 0 & \text{otherwise;} \end{cases}$$

Let

$$f_{nE}(\omega) = \tfrac{1}{n} \sum_{i=1}^{n} Z_{iE}(\omega), \text{ when } E \in G \text{ is used as a decision rule;}$$

and let

$$f_n(\omega) = \operatorname{Inf}\{ f_{nE}(\omega) : E \in G \}.$$

It is to be noted that the pattern classification algorithm mentioned in Section 3 uses $n * f_{nE}(\omega)$, the total number of misclassified points, as the objective function which it attempts to minimize. ($f_{nE}(\omega)$ is the error rate of the *GA-classifier* obtained by dividing the number of misclassified points with the total number of training data points.) This is equivalent to searching for a suitable $E \in G$ such that the term $f_{nE}(\omega)$ is minimized, i.e., for which $f_{nE}(\omega) = f_n(\omega)$. As already mentioned, it is known that for infinitely many iterations the Elitist model of GAs will certainly be able to obtain such an E.

Theorem. *For sufficiently large n, $f_n(\omega) \not> a$ (i.e., for sufficiently large n, $f_n(\omega)$ cannot be greater than* a) *almost everywhere.*

Proof. Let

$$Y_i(\omega) = \begin{cases} 1 \text{ if } X_i(\omega) \text{ is misclassified according to Bayes rule } \forall \omega \in \Omega, \\ 0 \text{ otherwise.} \end{cases}$$

Note that $Y_1, Y_2, \ldots, Y_n, \ldots$ are i.i.d. random variables. Now

$$\begin{aligned} Prob(Y_i = 1) &= \sum_{j=1}^{k} Prob(Y_i = 1 / X_i \text{ is in } C_j)\, P(X_i \text{ is in } C_j) \\ &= \sum_{j=1}^{k} P_j\, Prob(\omega : X_i(\omega) \in S_{0j}^{c} \text{ given that } \omega \in C_j) \\ &= \sum_{j=1}^{k} P_j \int_{S_{0j}^{c}} p_j(x)\, \mathrm{d}x = a. \end{aligned}$$

Hence the expectation of Y_i, $E(Y_i)$ is given by

$$E(Y_i) = a, \ \forall i.$$

Then by using Strong Law of Large Numbers [18], $(1/n)\sum_{i=1}^{n} Y_i \to a$ almost everywhere, i.e., $P(\omega\colon (1/n)\sum_{i=1}^{n} Y_i(\omega) \nrightarrow a) = 0$. Let $B = \{\omega : (1/n)\sum_{i=1}^{n} Y_i(\omega) \to a\} \subseteq \Omega$. Then $Q(B) = 1$.

Note that $f_n(\omega) \leqslant (1/n)\sum_{i=1}^{n} Y_i(\omega)$, $\forall n$ and $\forall \omega$, since the set of regions $(S_{01}, S_{02}, \ldots, S_{0k})$ obtained by the Bayes decision rule is also provided by some $A \in \mathscr{A}$ and consequently it will be included in G. Note that $0 \leqslant f_n(\omega) \leqslant 1$, $\forall n$ and $\forall \omega$. Let $\omega \in B$. For every $\omega \in B$, $U(\omega) = \{f_n(\omega),\ n = 1, 2, \ldots\}$ is a bounded, infinite set. Then by Bolzano–Weierstrass theorem [19], there exists an accumulation point of $U(\omega)$. Let $y = \mathrm{Sup}\{y_0\colon y_0 \text{ is an accumulation point of } U(\omega)\}$. From elementary mathematical analysis we can conclude that $y \leqslant a$, since $(1/n)\sum_{i=1}^{n} Y_i(\omega) \to a$ almost everywhere and $f_n(\omega) \leqslant (1/n)\sum_{i=1}^{n} Y_i(\omega)$. Thus it is proved that for sufficiently large n, $f_n(\omega)$ cannot be greater than a for $\omega \in B$.

Remarks. (1) The proof is not typical of the GA-based classifier. It will hold for any other classifier where the criterion is to reduce the number of misclassified points. For example, if simulated annealing is used instead of genetic algorithm, then too the results will hold under limiting conditions.

(2) Instead of hyperplanes, any other higher-order surface could have been used for obtaining the decision boundaries, provided the number of higher-order surfaces is finite and known, and the Bayes boundary is indeed provided by such surfaces. It would lead to only minor modifications to the proof presented earlier, with the basic inference still holding good.

(3) Although theoretically all possible H_o hyperplanes are considered while searching for the decision boundary, this is not feasible practically. The search space has to be discretized into small intervals which excludes some hyperplanes from consideration.

(4) The proof established that the number of points misclassified by the *GA-classifier* will always be less than or equal to the number of points misclassified by Bayes decision rule for sufficiently large number of training data points and iterations. However, the fact that $f_n(\omega) < a$ is true for only a finite number of training data points. This is due to the reason that a small number of identical training points can be generated by different statistical distributions. Consequently, each distribution will result in different error probabilities of the Bayes classifier. The *GA-classifier*, on the other hand, will always find the decision surface yielding the smallest number of misclassified points, irrespective of their statistical properties.

As the number of points increases, the number of possible distributions that can produce them decreases. In the limiting case when $n \to \infty$, only one distribution will produce all the training points [20]. The proof of this statement is as follows: Let X be an N-dimensional random vector. Let the distribution function of X be designated by $F(a_1, a_2, \ldots, a_N)$ which is equal to $P[X \in (-\infty, a_1] \times (-\infty, a_2] \times \cdots \times (-\infty, a_N]]$. Here $P(\cdot)$ is the probability of the event $(\cdot)$. Let the distribution function of X, estimated by n observations of X viz. $X_1, X_2, \ldots, X_n$, be $F_n(a_1, a_2, \ldots, a_N)$. Then from the strong law of large numbers

$$F_n(a_1, a_2, \ldots, a_N) \to F(a_1, a_2, \ldots, a_N)$$

as $n \to \infty$ almost everywhere.

The Bayes classifier designed over this distribution will provide the optimal decision boundary. The *GA-classifier*, in that case, will also yield a decision boundary which is same as the Bayes decision boundary provided the number of surfaces is known a priori. This fact has been borne out by the experimental results given in Section 6 that for an increased number of training points, the decision surface provided by the *GA-classifier* indeed approaches the Bayes decision boundary.

(5) From the theorem, it is obvious that the boundary provided by the *GA-classifier* will approach the Bayes boundary as the number of training data points goes to infinity. Otherwise, the number of points misclassified during training of the *GA-classifier* would be more than that of the Bayes classifier (since Bayes classifier is known to be the optimal one); but this is impossible from the theorem. Results are presented in Section 6.2 to verify the validity of this statement.

(6) The theorem states that the error rate of the *GA-classifier* during training is less than or equal to the Bayes error probability for sufficiently large number of training data points and iterations. However, the performance of a classifier is better evaluated if both the training and the test cases are taken into consideration. Such results are presented in Section 6.2.

The results proved analytically in this section are also verified experimentally in Section 6 under different situations. Note that in the proof described above it is assumed that H_o is known a priori. However, in practice, as mentioned earlier, H_o may not be known. In that case, we try to overestimate H_o which may lead to the presence of redundant hyperplanes in the resultant decision boundary. The next section describes a method of eliminating the redundant hyperplanes thereby generating the optimum number of hyperplanes H_{GA} of the *GA-classifier*. A relationship of H_{GA} and the optimum hyperplanes H_o yielding the Bayes boundary is also established in the next section.

5. Relationship between H_o and H_{GA}

Here, we first present a technique for getting H_{GA} hyperplanes from the initial over estimation of H. Subsequently, we establish that H_{GA} is equal to H_o when there exists exactly one partition of the feature space that provides the Bayes error probability a. In case more than one partition can provide the Bayes error probability, then all we can say is that H_{GA} will be greater than or equal to H_o.

5.1. Obtaining H_{GA} from H

A hyperplane is considered to be redundant if its removal has no effect on the recognition score of the classifier for the given training data set. In order to arrive at the optimal number of hyperplanes, one of the ways is to consider first of all, all possible combinations of the H hyperplanes. For each such combination, the first hyperplane is removed and it is checked whether the remaining hyperplanes can successfully classify all the patterns. If so, then this hyperplane is deleted, and the test is repeated for the next hyperplane in the combination.

Obviously, testing all possible combinations results in an algorithm with exponential complexity in H. To avoid this, a branch and bound technique can be adopted where the search within a combination is discontinued (before considering all the H hyperplanes) if the number of hyperplanes found to be non-redundant so far, is greater than or equal to the number of hyperplanes declared to be non-redundant by some earlier combination. The complexity may be reduced further by terminating the algorithm if the combination being tested provides a set of $\lceil \log_2 k \rceil$ non-redundant hyperplanes (see Section 3), this being the minimum number of hyperplanes that is required for generating the k regions. This method guarantees removal of all the redundant hyperplanes from the decision boundary, thereby yielding exactly H_{GA} hyperplanes.

5.2. How H_{GA} is related to H_o

Let H_{GA} be the number of hyperplanes (after elimination of redundancy) found by the *GA-classifier* to provide an error rate $= f_n(\omega)$, which is less than or equal to a (Bayes error probability) when $n \to \infty$. If H_o is the optimal number of hyperplanes, then obviously H_{GA} cannot be less than H_o. It is now our task to ascertain whether $H_{GA} > H_o$ or $H_{GA} = H_o$. For this we must first consider the following situations:

(a) The number of partitions which provides the Bayes error probability is exactly one. Since this partition, formed from H_o hyperplanes, provides the Bayes error probability, which is known to be optimal, for $n \to \infty$, the regions provided by the H_{GA} hyperplanes must be exactly same as the regions provided by the H_o hyperplanes. Thus H_{GA} must be the same as H_o for large values of n.
(b) On the contrary, the number of partitions that provide the Bayes error probability may be greater than one. For example, this will be the case if at least one of the k classes is totally disconnected from the other classes. Another example is provided in Appendix A. In these cases, the regions provided by the H_{GA} hyperplanes may not be identical to the ones provided by the H_o hyperplanes. Consequently, H_{GA} can be greater than H_o for such situations although the classifier still provides an average error $= f_n(\omega)$.

5.3. Some points related to n and H

In practice, we always deal with finite data sets (or finite n). Obviously, in that case, additional hyperplanes, beyond H_o, may be placed appropriately in order to further reduce the number of misclassified points, at the cost of possibly reduced generalizability. These hyperplanes will not be eliminated by the redundancy removal process. However, as n increases, the effect of introducing additional hyperplanes will decrease and also the performance of the classifier in terms of the test data will gradually improve. In the limiting case, for $n \to \infty$, only the optimum number of hyperplanes with a specific arrangement will provide the requisite decision boundary. Any additional hyperplane will obviously be detected as redundant. At the same time, the generalization of the classifier will be optimum.

6. Experimental results

Extensive empirical results are provided in this section, which are seen to conform to the theoretical findings in Section 4. It is also found experimentally that the decision boundary obtained by *GA-classifier* approaches that of Bayes classifier as n increases. Data sets following triangular and normal distributions are considered having both linear and non-linear class boundaries. All the data sets have considerable amount of overlap. In a part of the experiment, instead of hyperplanes, circular segments (in two dimensions) are considered for constituting elements of the decision boundaries. Its objective is to demonstrate whether the theoretical claims made in Section 4 for hyperplanes hold good for higher-order surfaces or not. The effect of class a priori probability on the recognition score has also been experimentally investigated.

Different situations considered for conducting the experiments are as follows:

(i) The decision boundary is provided by H_o hyperplanes, and H_o is known a priori.
(ii) The decision boundary is provided by H_o higher-order surfaces, and H_o is known a priori.
(iii) It is known that the decision boundary can be approximated by H_o hyperplanes but the value of H_o is not known.
(iv) It is known that the decision boundary can be approximated by H_o higher-order surfaces but the value of H_o is not known.
(v) It is known that no value of H_o hyperplanes can approximate the decision boundary.
(vi) It is known that no value of H_o higher-order surfaces can approximate the decision boundary.
(vii) Nothing is known about the given data set. In that case, we may try to approximate the boundary by a fixed number of hyperplanes or any other higher-order surfaces.

This section is divided into three parts. The description of the data sets is given in the first part. The decision boundaries and recognition scores (during training and testing) obtained by *GA-classifier* (using linear as well as circular surfaces) are then compared with those of Bayes classifier for different sizes of the training data in the second part. Finally, the third part demonstrates the variation of the generalization capability of the classifier as a function of the class a priori probability, for two class problems.

6.1. Data sets

Four types of data sets are used which are described here.

Data set 1: A two-dimensional ($X - Y$) data set is generated using a triangular distribution of the form shown in Fig. 1 for the two classes, 1 and 2. The range for class 1 is $[0, 2] \times [0, 2]$ and that for class 2 is $[1, 3] \times [0, 2]$ with the corresponding peaks at $(1, 1)$ and $(2, 1)$. Figure 1 shows the distribution along the X-axis since only this axis has discriminatory capability. The distribution along the X-axis may be formally

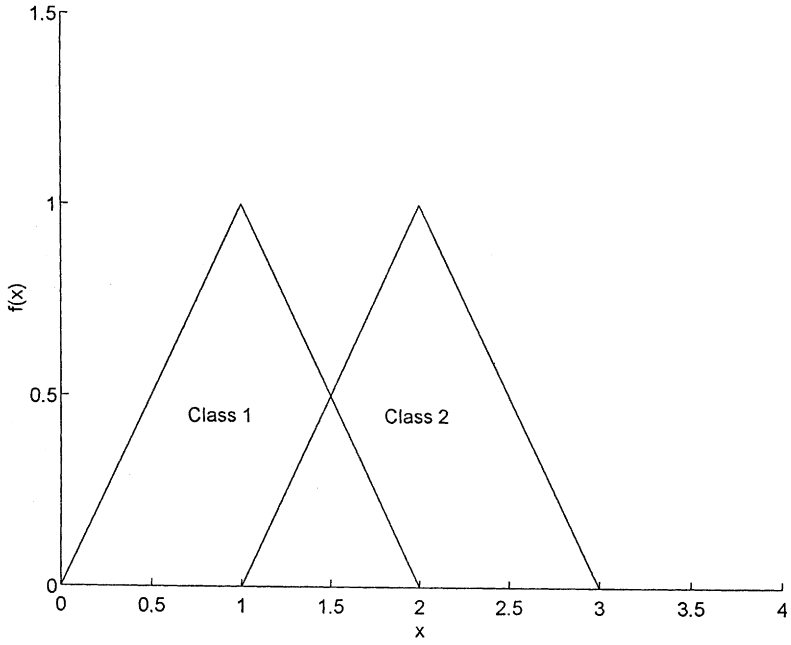

Fig. 1. Triangular distribution along the x-axis for Data set 1 having two classes.

quantified as

$$f_1(x) = \begin{cases} 0 & \text{for } x \leqslant 0, \\ x & \text{for } 0 < x \leqslant 1, \\ 2 - x & \text{for } 1 < x \leqslant 2, \\ 0 & \text{for } x > 2. \end{cases}$$

for class 1. Similarly for class 2

$$f_2(x) = \begin{cases} 0 & \text{for } x \leqslant 1, \\ x - 1 & \text{for } 1 < x \leqslant 2, \\ 3 - x & \text{for } 2 < x \leqslant 3, \\ 0 & \text{for } x > 3. \end{cases}$$

The distribution along the Y-axis for both the classes is

$$f(y) = \begin{cases} 0 & \text{for } y \leqslant 0, \\ y & \text{for } 0 < y \leqslant 1, \\ 2 - y & \text{for } 1 < y \leqslant 2, \\ 0 & \text{for } y > 2. \end{cases}$$

If P_1 is the a priori probability of class 1 then using elementary mathematics, we can show that Bayes classifier will classify a point to class 1 if its X coordinate is less than $1 + P_1$. This indicates that the Bayes decision boundary is given by

$$x = 1 + P_1. \tag{4}$$

Data set 2: Figure 2 shows a normally distributed data set consisting of two classes. The mean (μ_1, μ_2) and covariance values (Σ_1, Σ_2) for the two classes are $\mu_1 = (0.0, 0.0)$, $\mu_2 = (1.0, 0.0)$ and

$$\Sigma_1 = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}, \qquad \Sigma_2 = \begin{pmatrix} 4.0 & 0.5 \\ 0.5 & 4.0 \end{pmatrix},$$

respectively.

The classes are assumed to have equal a priori probability (= 0.5). Mathematical analysis shows that the Bayes decision boundary for such a distribution of points will be of the following form:

$$a_1 x_1^2 + a_2 x_2^2 + 2a_3 x_1 x_2 + 2b_1 x_1 + 2b_2 x_2 + c = 0.$$

The Bayes boundary for such a class distribution is also shown in Fig. 2.

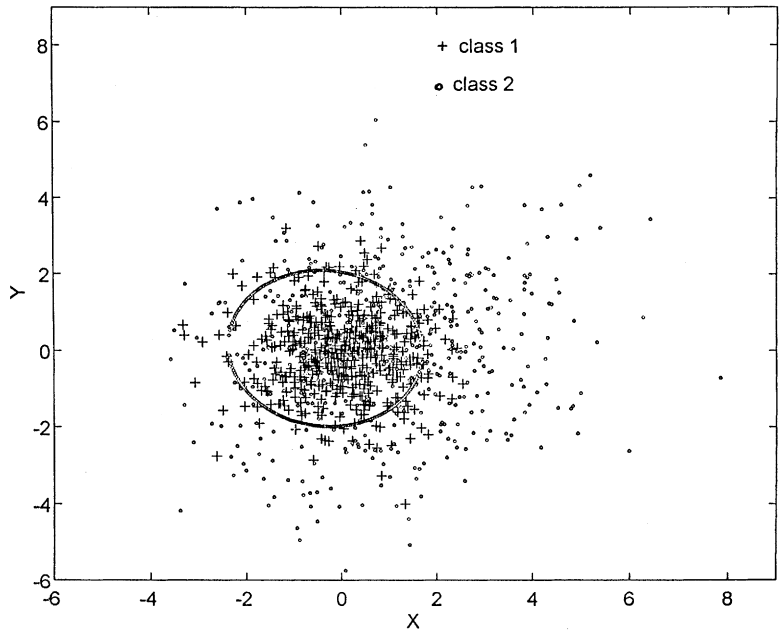

Fig. 2. Data set 2 for $n = 1000$ along with the Bayes decision boundary for two classes.

Data set 3: To consider a multi-class problem, a nine class triangular distribution of data points is considered. All the classes are assumed to have equal a priori probabilities (=1/9). The X–Y ranges for the nine classes are as follows:

- Class 1: $[-3.3, -0.7] \times [0.7, 3.3]$.
- Class 2: $[-3.3, 1.3] \times [0.7, 3.3]$.
- Class 3: $[0.7, 3.3] \times [0.7, 3.3]$.
- Class 4: $[-3.3, -0.7] \times [-1.3, 1.3]$.
- Class 5: $[-1.3, 1.3] \times [-1.3, 1.3]$.
- Class 6: $[0.7, 3.3] \times [-1.3, 1.3]$.
- Class 7: $[-3.3, -0.7] \times [-3.3, -0.7]$.
- Class 8: $[-1.3, 1.3] \times [-3.3, -0.7]$.
- Class 9: $[0.7, 3.3] \times [-3.3, -0.7]$.

Thus, the domain for the triangular distribution for each class and for each axis is 2.6. Consequently, the height will be (1/1.3) (since $\frac{1}{2} \times 2.6 \times height = 1$). The resulting Bayes boundary along with the data set is shown in Fig. 3.

Data set 4: This two class two-dimensional data set is used specifically for the purpose of investigating the generalization capability of the *GA-classifier* as a function of the a priori probability of class 1. The data set is normally distributed with the

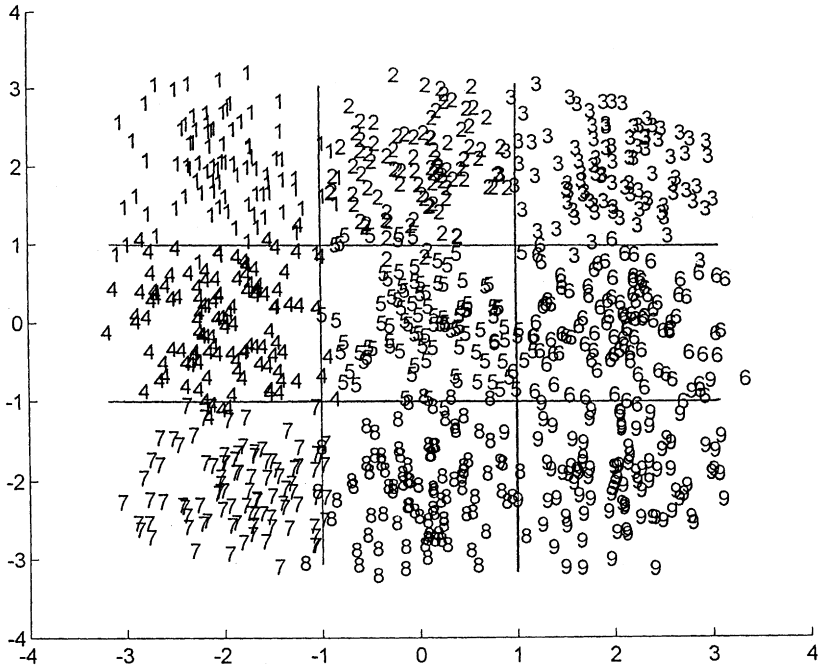

Fig. 3. Data set 3 for $n = 900$ along with the Bayes decision boundary for nine classes.

following parameters: $\mu_1 = (0.0, 0.0)$, $\mu_2 = (1.0, 0.0)$ and

$$\Sigma_1 = \Sigma_2 = \Sigma = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}.$$

Since $\Sigma_1 = \Sigma_2$, the Bayes boundary will be linear [21] of the following form:

$$(\mu_2 - \mu_1)^{\mathrm{T}} \Sigma^{-1} X + \tfrac{1}{2}(\mu_1^{\mathrm{T}} \Sigma^{-1} \mu_1 - \mu_2^{\mathrm{T}} \Sigma^{-1} \mu_2).$$

Note: (1) For each data set and for a particular value of n, two different training data sets are generated using different seed values. Training is performed with each data set for five different initial populations. This means that a total of 10 runs are performed for a given n. The results of the *GA-classifier* presented here are average values over these 10 runs.

(2) The test data set comprises 1000 points. Testing of the *GA-classifier* is performed for each of the 10 training runs. Results shown are the average values over these 10 runs.

(3) Roulette wheel selection strategy with elitism is used in this investigation. Due to memory constraints, population size of 20 is chosen. Consequently, a high crossover probability (*cr_prob*) of 0.8 is fixed for single point crossover scheme. The mutation probability value (*mut_prob*) varies over the range of [0.01, 0.333], initially having a high value, then gradually decreasing and finally increasing again in the later stages of the algorithm. 100 iterations of the GA are performed for each *mut_prob*. A total of 1500 iterations is performed. The mutation probability range is uniformly divided into 8 distinct values. Starting from 0.333, at first the value is gradually decreased after every 100 iterations, till the minimum value is achieved. Subsequently, it is again increased in the reverse order after every 100 iterations till it attains the value 0.333. Thus a maximum of 1500 iterations are executed.

Note that it is proved in [17] that an elitist model of GA will always provide the optimal string as the number of generations goes to infinity, provided the probability of going from any population to the one containing the optimal string is greater than zero. This proof is irrespective of the population sizes and different probability values. However, appropriate selection of GA parameters is necessary in view of the fact that it has to be terminated after a finite number of iterations.

6.2. Learning the class boundaries and performance on test data

6.2.1. Utilizing linear surfaces

An extensive comparison of the performance of the *GA-classifier* with that of Bayes classifier is performed for Data sets 1–3. The following cases are considered for Bayes classifier:

Case 1: The distributions as well as the class a priori probabilities (P_1 and P_2) are known.

Case 2: The distribution is known, with $P_1 = P_2 = 0.5$.

Case 3: The distribution is known, with $P_1 = n_1/n$ and $P_2 = n_2/n$. Here n_1 and n_2 are the number of points belonging to classes 1 and 2, respectively, $n = n_1 + n_2$.

Case 4: Normal distribution with unequal covariance matrices, while P_1 and P_2 are known.

Case 5: Normal distribution with unequal covariance matrices and $P_1 = P_2 = 0.5$.

Case 6: Normal distribution with unequal covariance matrices, and $P_1 = n_1/n$ and $P_2 = n_2/n$, where n_1 and n_2 are defined above.

(a) *Data set 1*: The different values of n considered are 100, 500, 1000, 1500, 2000, 3000, and 4000 and $P_1 = 0.4$. The Bayes boundary for this data set is a straight line ($x = 1.4$). The training recognition scores of Bayes classifier and *GA-classifier* are shown for $H = 1$ and 3 (Table 1). As expected, for each value of n, the recognition score during training of the *GA-classifier* with $H = 3$ was found to be at least as good as that with $H = 1$. Table 2 shows the recognition scores for the test data. Here, it is found that the Bayes classifier yields a better performance than the *GA-classifier*. Also, *GA-classifier* with $H = 3$ provides a consistently lower score than with $H = 1$. This is expected since larger H leads to overfitting of the training data, thereby yielding better training performance (Table 1), while the generalization capability of the classifier degrades (Table 2).

It is seen from Table 1 that for all values of n, the recognition scores for *GA-classifier* during training are better than those of Bayes (all cases). However, this difference in

Table 1
Comparative classwise and overall training recognition scores (%) for Data set 1

n	Class	*GA-classifier*		Bayes classifier					
		$H = 1$	$H = 3$	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
100	1	88.77	89.26	81.08	85.94	81.08	82.60	88.77	83.71
	2	93.24	95.00	91.21	87.88	90.30	89.60	87.18	89.60
	Overall	91.50	92.70	87.00	87.5	87.00	87.00	88.00	87.50
500	1	76.48	78.24	80.29	87.75	81.41	83.43	87.51	82.87
	2	95.69	95.59	89.39	84.73	90.15	87.87	85.38	87.85
	Overall	88.00	88.65	85.70	85.90	86.20	86.10	86.20	86.00
1000	1	86.91	86.17	82.68	87.64	83.03	84.32	87.40	84.80
	2	91.01	92.00	92.93	87.71	92.75	91.84	87.81	91.58
	Overall	89.35	89.67	88.80	87.70	88.85	88.80	87.65	88.50
1500	1	87.17	87.70	82.26	88.18	82.70	84.38	88.33	84.59
	2	90.73	90.45	92.07	88.17	92.10	91.11	87.94	90.54
	Overall	89.08	89.35	88.10	88.16	88.33	88.10	88.10	88.16
2000	1	80.77	81.00	83.13	88.71	83.25	84.24	87.97	84.24
	2	94.81	94.93	91.71	86.77	91.54	91.37	87.52	91.29
	Overall	89.15	89.36	88.25	87.55	88.20	88.50	87.70	88.45
3000	1	75.96	78.72	80.82	85.59	80.57	82.41	85.34	82.24
	2	95.68	94.02	91.86	87.65	91.86	90.75	87.76	90.81
	Overall	87.83	87.90	87.46	86.83	87.36	87.43	86.80	87.40
4000	1	82.54	82.50	87.73	87.39	81.89	84.35	87.39	83.83
	2	91.81	92.08	91.52	86.88	91.93	90.18	86.67	90.59
	Overall	88.22	88.25	88.12	87.07	88.05	87.92	86.95	87.97

performance gradually decreases for larger values of n. This is demonstrated in Fig. 4, which plots α as a function of n, where α = (overall training recognition score of the *GA-classifier* − overall recognition score of the Bayes classifier (Case 1)). This indicates that as n increases, the decision boundaries provided by the *GA-classifier* gradually approach the Bayes boundary (see Remark in Section 4). This observation is demonstrated in Figs. 5, 6, 7 and 8 for $n = 100$, 1000, 2000 and 4000, respectively.

Table 2
Comparative overall test recognition scores (%) for Data set 1

n	*GA-classifier*		Bayes classifier
	$H = 1$	$H = 3$	
100	85.30	85.00	87.90
500	85.90	85.75	87.90
1000	86.20	85.87	87.90
1500	86.21	86.00	87.90
2000	86.55	86.41	87.90
3000	87.10	87.10	87.90
4000	87.32	87.20	87.90

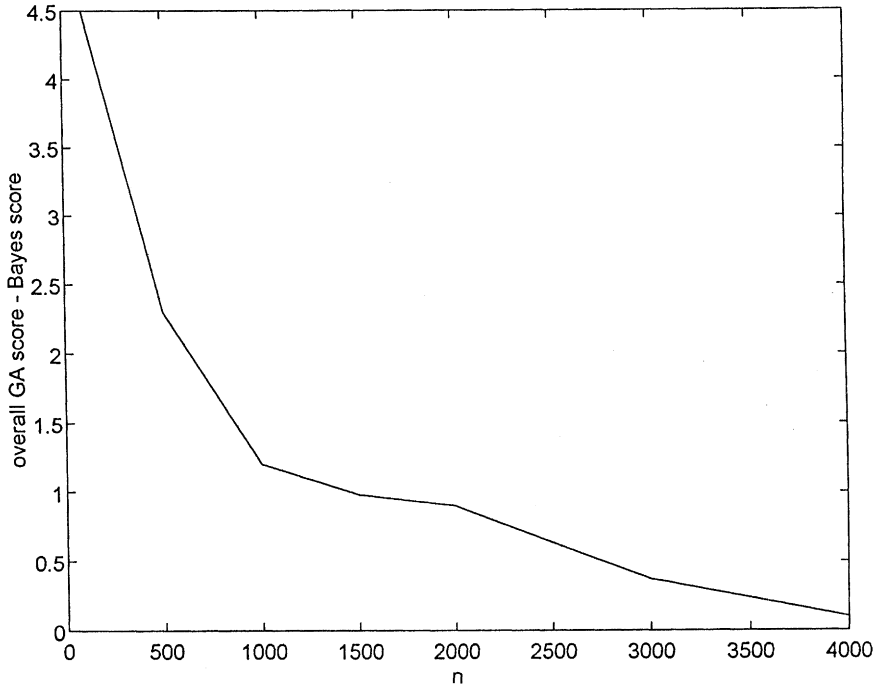

Fig. 4. Variation of α (= overall GA score—Bayes score) with n for Data set 1.

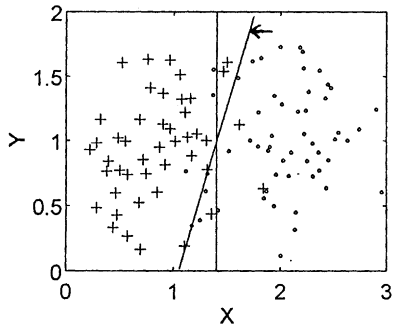

Fig. 5. Data set 1 for $n = 100$ and the boundary provided by *GA-classifier* for $H = 1$ (marked with an arrow) along with Bayes decision boundary. Class 1 is represented by '+' and class 2 by '●'.

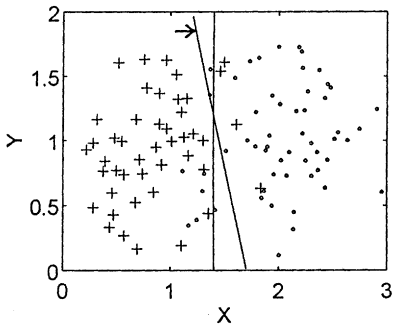

Fig. 6. Data set 1 for $n = 1000$ and the boundary provided by *GA-classifier* for $H = 1$ (marked with an arrow) along with Bayes decision boundary. Class 1 is represented by '+' and class 2 by '●'. (Only 100 data points are plotted for clarity.)

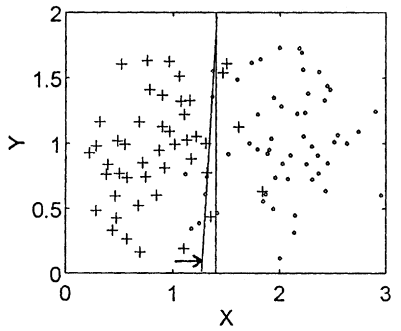

Fig. 7. Data set 1 for $n = 2000$ and the boundary provided by *GA-classifier* for $H = 1$ (marked with an arrow) along with Bayes decision boundary. Class 1 is represented by '+' and class 2 by '●'. (Only 100 data points are plotted for clarity.)

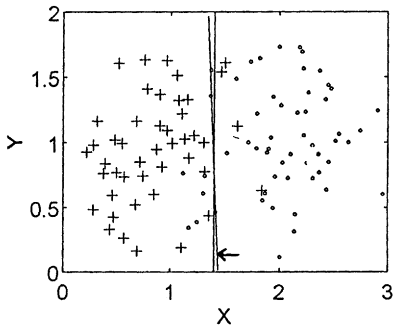

Fig. 8. Data set 1 for $n = 4000$ and the boundary provided by *GA-classifier* for $H = 1$ (marked with an arrow) along with Bayes decision boundary. Class 1 is represented by '+' and class 2 by '●'. (Only 100 data points are plotted for clarity.)

A point to be mentioned here is that although the result of Bayes classifier (Case 1) is the ideal one considering the actual distribution and a priori probabilities, interestingly some of the other cases (e.g., Case 5 for $n = 100$, Case 3 for $n = 500$) are seen to produce better scores. A reason for this discrepancy may be because of the effect of stochastic errors in sampling of the data points. It is also found that results for cases 1 and 4 are similar (having same values for $n = 100$, 1000 and 1500, and close values for other values of n). The reason for this is that normal distribution with proper variance has been found to be able to approximate the triangular distribution closely.

(b) *Data set 2*: Four values of n considered here for the experiments are $n = 500$, 1000, 2000 and 5000. Cases 1, 3, 4 and 6 are investigated since only these are relevant in this context. Table 3 presents the results corresponding to them during training. Figure 2 shows the data set for $n = 1000$ and the corresponding Bayes boundary. (It is found that the Bayes boundary totally surrounds class 1.) Due to practical limitations, we have considered $H = 4$ and 6 only for executing the *GA-classifier*, although from the complexity of the decision surface it appears that a still larger number of lines may be required for proper approximation.

For smaller values of n (500 and 1000), it is found, as in the case of Data Set 1, that the *GA-classifier* yields a better score than the Bayes classifier for the training data set (Table 3). Figure 9 shows the decision boundary obtained for $n = 1000$ and $H = 4$, along with the Bayes boundary. Although all the four lines of GA are found to be necessary, and they surround class 1, the boundary formed by them could not approximate the Bayes boundary well. In spite of this fact, its recognition score during training is relatively larger than that of the Bayes classifier. Increasing H to a value 6 improves both the approximation of the boundary and the recognition score during training (Table 3).

For $n = 2000$, one out of the four lines is found to be redundant by the *GA-classifier* (see Fig. 10) and they fail to surround class 1 for the same number (1500) of iterations. The training score is accordingly found to be lower than that of the Bayes classifier. For $H = 6$ (Fig. 11), the recognition score during training exceeds the one obtained by

Table 3
Comparative classwise and overall training recognition scores (%) for Data set 2

n	Class	*GA-classifier*		Bayes classifier			
		$H = 4$	$H = 6$	Case 1	Case 3	Case 4	Case 6
500	1	89.23	89.23	85.38	86.54	85.77	87.31
	2	62.92	65.83	65.83	64.17	64.17	63.33
	Overall	76.60	78.00	76.00	75.80	75.40	75.80
1000	1	83.92	84.90	83.33	84.12	83.33	84.71
	2	67.96	67.44	66.53	65.92	65.71	65.10
	Overall	76.10	76.35	75.10	75.20	74.70	75.10
2000	1	87.98	87.58	85.27	83.73	84.77	85.22
	2	61.78	63.37	64.77	65.43	65.17	65.18
	Overall	74.85	75.45	75.00	74.87	74.95	74.98
5000	1	85.18	83.04	85.42	85.50	85.74	85.82
	2	65.04	68.39	66.33	66.01	66.09	65.81
	Overall	75.18	75.76	75.94	75.82	75.98	75.88

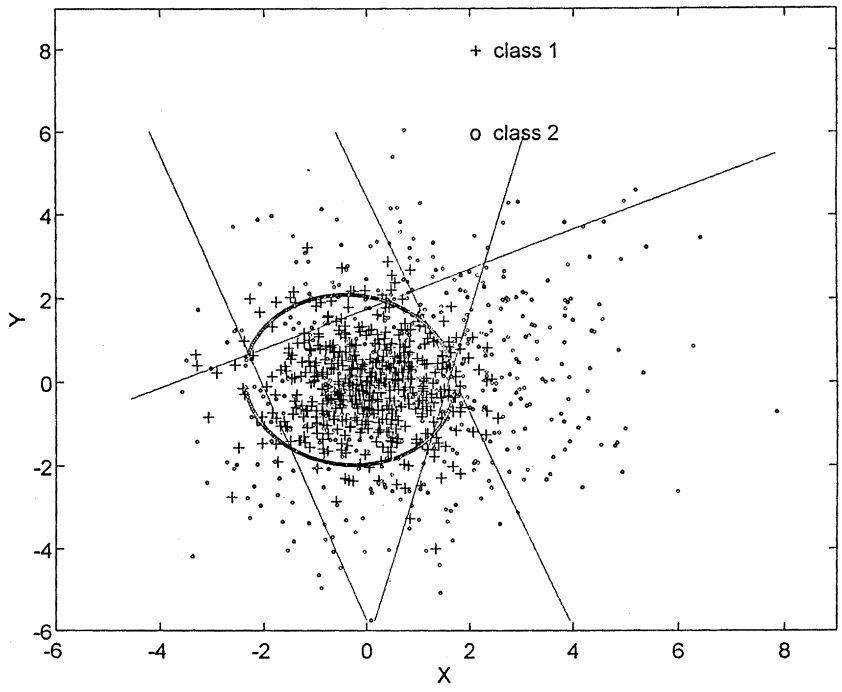

Fig. 9. Data set 2 for $n = 1000$ and the boundary provided by *GA-classifier* for $H = 4$ along with Bayes decision boundary (circular one).

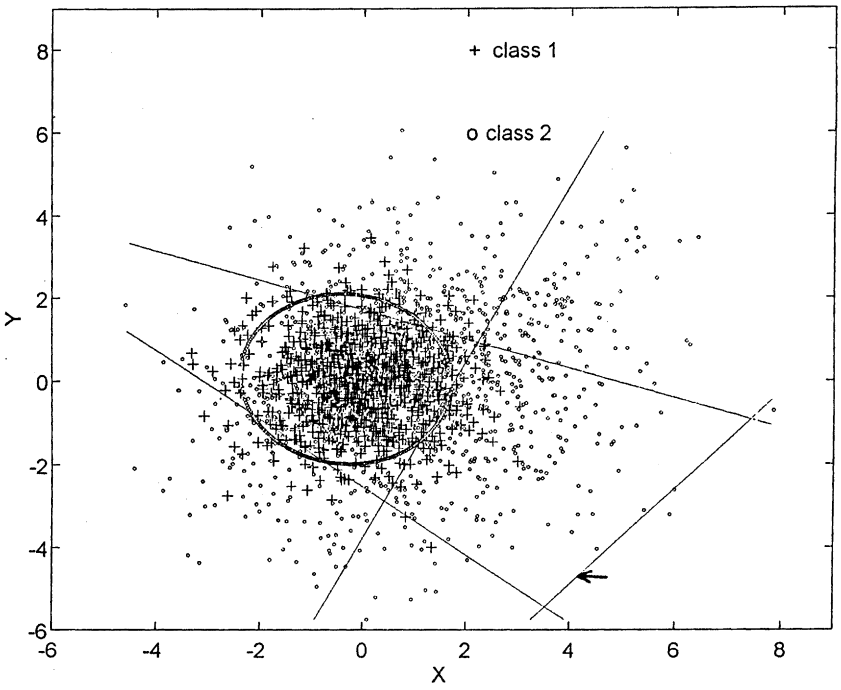

Fig. 10. Data set 2 for $n = 2000$ and the boundary provided by *GA-classifier* for $H = 4$ along with Bayes decision boundary (circular one). The redundant line is marked with an arrow.

Bayes, and the approximation to the Bayes boundary improves (although only 5 lines are utilized effectively).

For a further increase in n to 5000, the training recognition scores (Table 3) for both $H = 4$ and 6 are seen to be worse than Bayes scores for the same number of iterations. However, the approximation to Bayes boundary is seen to be much improved here as compared to $n = 1000$ and 2000. This is evident from Figs. 12 and 13, where $H = 6$, as expected, provides better performance that $H = 4$. From Figs. 10 and 13 it is interesting to note that one line is found to be redundant and at the same time the score is worse than that of Bayes. This may be attributed to the premature termination of GA.

Table 4 shows the recognition scores for Data set 2 during testing. Again, it is found that the *GA-classifier* provides poorer scores than Bayes classifier. Also, *GA-classifier* with $H = 6$ provides better performance than with $H = 4$. This is expected since $H = 6$ can better approximate the complex decision surface.

(c) *Data set 3*: In order to extend the results for problems where $k > 2$, we utilized a nine class, two dimensional data set. The data set for $n = 1800$ along with the corresponding Bayes boundary is shown in Fig. 3. The experiments on training as well as test data sets have been conducted for $H = 4$ and $n = 450, 900, 1350$ and 1800. Only the training results are presented here. As before, the test results, show a superior

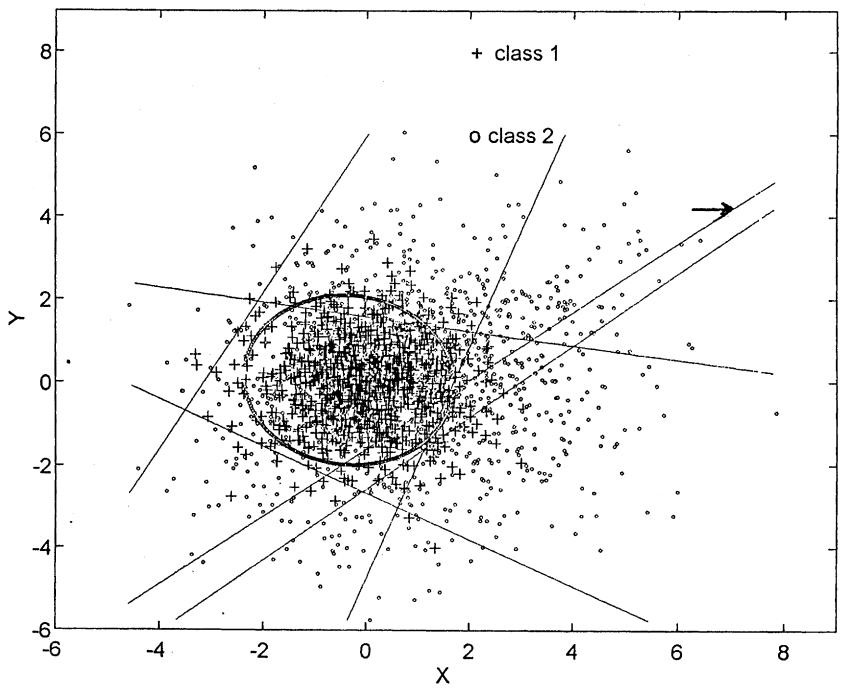

Fig. 11. Data set 2 for $n = 2000$ and the boundary provided by *GA-classifier* for $H = 6$ along with Bayes decision boundary (circular one). The redundant line is marked with an arrow.

performance of the Bayes classifier, and are omitted. Comparison with the Bayes classifier is made for case 1 only. The results for the overall training recognition scores are shown in Table 5. Figure 14 shows the boundary obtained using the *GA-classifier* along with the Bayes decision boundary for $n = 1800$ where it seen is that the four lines more or less approximate the Bayes boundary. It is found from Table 5 that although for each value of n the training scores of the *GA-classifier* and the Bayes classifier are comparable, the latter one appears to provide a slightly better performance. The *GA-classifier* is not able to approximate the Bayes boundary very closely. One of the reasons may again be the premature termination of the GA. Another factor may be the coding itself which does not allow encoding of the actual Bayes lines (this may be due to an insufficiency of the precision defined for the perpendicular distance of the hyperplane).

6.2.2. Utilizing higher-order surfaces

It has been pointed out earlier (Section 4) that instead of approximating the decision boundary by hyperplanes, we can assume any other higher-order surface with similar effect. This subsection describes the method of utilizing a fixed number of circular segments to constitute the decision surface in two-dimensional space.

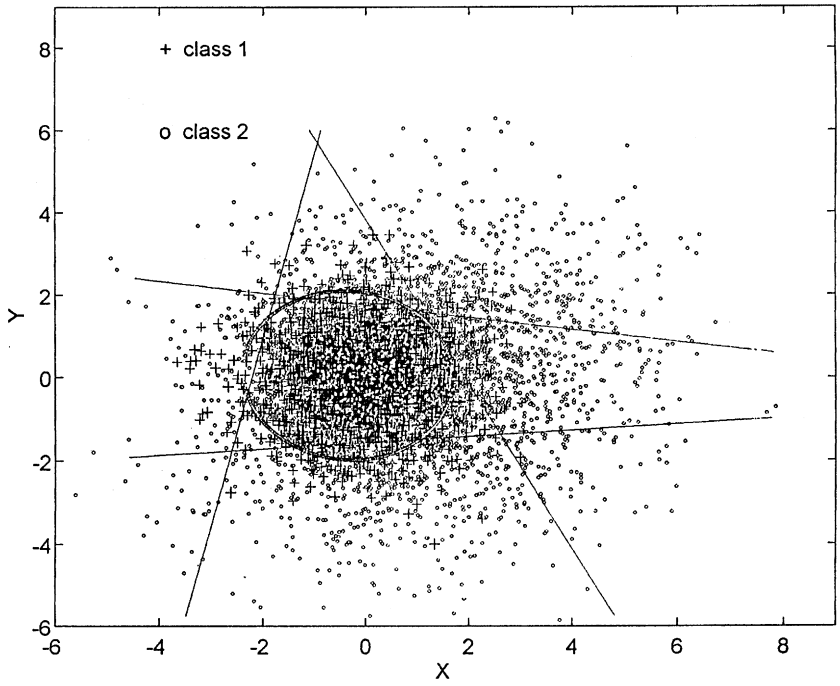

Fig. 12. Data set 2 for $n = 5000$ and the boundary provided by *GA-classifier* for $H = 4$ along with Bayes decision boundary (circular one).

The equation of a circle with center (h, k), and radius r is given by

$$(x - h)^2 + (y - k)^2 = r^2.$$

Thus, the three parameters h, k and r correspond to a unique circle and these are encoded in a string of the *GA-classifier*. Let l and b be the length and breadth, respectively, of the rectangle constructed around the data points in the two-dimensional space. This rectangle will subsequently be referred to as the *inner rectangle*. Surrounding the *inner rectangle*, let us consider a larger rectangle, to be referred to as the *outer rectangle*, of length $(2 \times p + 1)\, l$ and breadth $(2 \times p + 1)\, b$. Then the center of the circle (which will be considered for constituting the decision boundary) is allowed to lie anywhere within the *outer rectangle*. p can be chosen sufficiently large in order to approximate any form of the decision boundary. The center (h, k) of the circle is a randomly chosen point within the outer rectangle. For computing the radius, the nearest distance of the center (h, k) from the vertices of *inner rectangle* is determined. Let this be d_1. Similarly, let the farthest distance be d_2. Now if (h, k) lies within the *inner rectangle*, then the radius can take on values in the range $[0, d_2]$. Otherwise the range is considered as $[d_1, d_2]$. A random value chosen from the range corresponds to the radius of the circle.

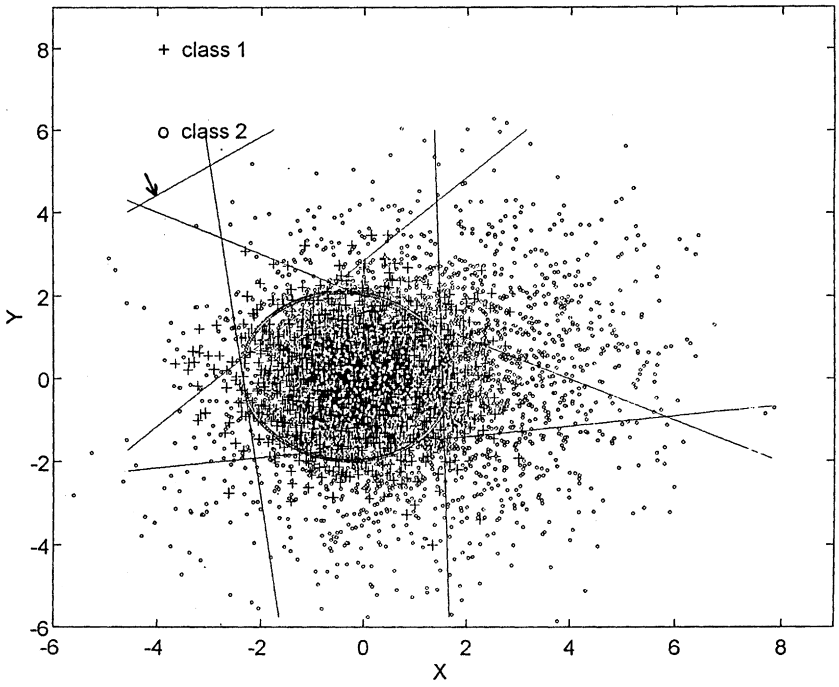

Fig. 13. Data set 2 for $n = 5000$ and the boundary provided by *GA-classifier* for $H = 6$ along with Bayes decision boundary (circular one). The redundant line is marked with an arrow.

Table 4
Comparative overall test recognition scores (%) for Data set 2

n	*GA-classifier*		Bayes classifier
	$H = 4$	$H = 6$	
500	72.75	73.20	74.80
1000	72.92	73.35	74.80
2000	73.05	73.62	74.80
5000	73.34	74.15	74.80

Analogous to using hyperplanes, a fixed number of circles C is considered to constitute the boundaries between classes. The regions corresponding to the different classes are determined from the training data set. Fitness of a string is determined by the number of points correctly classified by the string. For our experiment, we assumed $p = 8$. The other genetic parameters are kept the same as discussed earlier.

The results obtained during both training and testing, when circular segments are used to model the decision boundaries for Data sets 1 and 2, are shown in Table 6.

Table 5
Comparative overall recognition scores (%) for Data set 3

n	Overall recognition score	
	GA-classifier ($H = 4$)	Bayes classifier (Case 1)
450	93.56	93.78
900	93.22	93.11
1350	90.08	92.52
1800	92.25	92.50

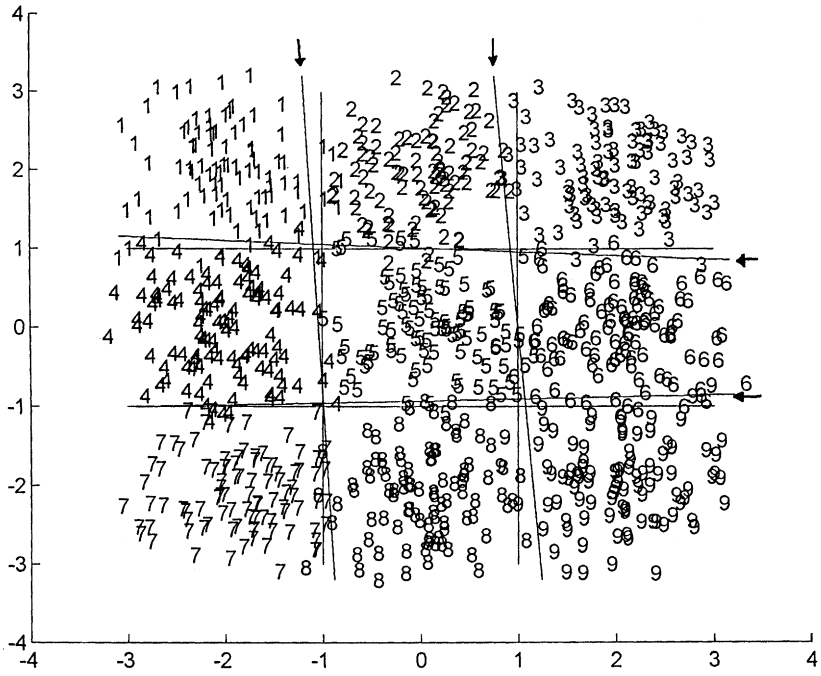

Fig. 14. Data set 3 and the boundary provided by *GA-classifier* for $n = 1800$ and $H = 4$ (marked with arrows) along with Bayes decision boundary. (Only 900 points are shown.)

Interestingly, the approximation of the decision boundary for Data sets 1 and 2 could be made better using linear and circular segments respectively. This is followed for both data sets. Figures 15 and 16 show the GA and Bayes boundaries obtained for Data set 1, $n = 1000$ and $C = 1$, and Data set 2, $n = 5000$ and $C = 6$ (where one segment is found to be redundant as in Fig. 13), respectively.

Again, the test scores for Data set 1 with circular segments are found to be poorer compared to those with linear segments. The case is reversed for Data set 2 (except

Table 6
Overall training and test recognition scores (%) for higher-order surfaces

Data set	n	No. of circles C	*GA-classifier* (circular surface)		*GA-classifier* (linear surface $H = C$) from Tables 1 and 3	
			Training	Testing	Training	Testing
Data set 1	1000	1	89.40	86.05	89.35	86.20
	2000	1	89.10	86.20	89.15	86.55
Data set 2	2000	4	74.97	73.10	74.85	73.05
	2000	6	75.55	73.60	75.45	73.62
	5000	6	76.00	74.45	75.76	74.15

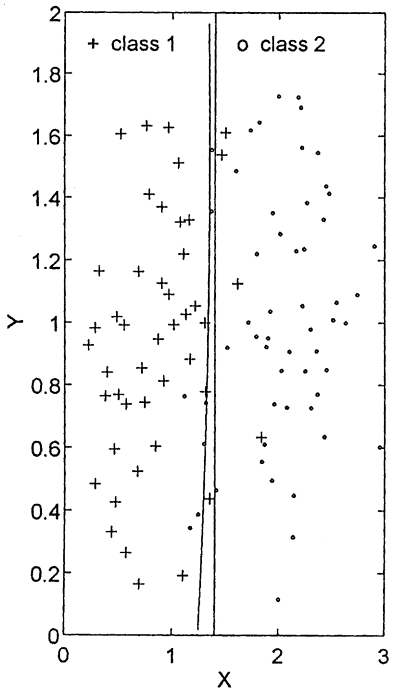

Fig. 15. Data set 1 for $n = 1000$ and the circular boundary (left one) provided by *GA-classifier* when circular segments are considered to constitute the decision boundary for $C = 1$ along with Bayes decision boundary. (Only 100 data points are plotted for clarity.)

with $n = 2000$ and $H = 6$, where the two are comparable). This is expected since the decision boundary is linear for Data set 1, while it is circular for Data set 2. Note that all these results are however inferior to the Bayes scores for the test data (Tables 2 and 4).

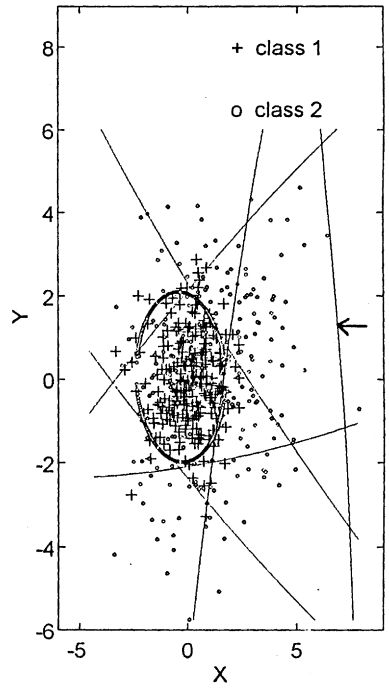

Fig. 16. Data set 2 for $n = 5000$ and the boundary provided by *GA-classifier* when circular segments are considered to constitute the decision boundary for $C = 6$ along with Bayes decision boundary. The redundant segment is marked with an arrow. (Only 500 data points are plotted for clarity.)

Note that for $p = 8$, the circular property of the segments does not appear to be fully exploited (Fig. 16) because of large radius. This was done in order to be able to approximate better any type of decision boundary using circular segments only. Figure 17 shows another result for $n = 2000$, $p = 0$ and $C = 2$ which corresponds to smaller radius, and hence shows better circular characteristics. Here, the recognition scores over the training and the test data are 75.10% and 74.65%, respectively.

6.3. *Variation of recognition scores with* P_1

The variation of the test recognition scores of the *GA-classifier* and Bayes classifier with P_1 is now demonstrated for Data set 1, 2 and 4 in Figs. 18 ($H = 1$ and 3), 19 ($H = 6$), and 20 ($H = 1$), respectively. Here one training data set of size 200 and two test data sets of size 1000 are taken. Training of the *GA-classifier* is performed for five initial conditions. Subsequently, testing is performed for the two test data sets. The results shown are the average values over the ten runs. Note that the variation of the test recognition scores of the *GA-classifier* with P_1 is similar to that of Bayes classifier for all the three data sets. (For the convenience of the readers, the above-mentioned variation for Bayes classifier is discussed theoretically in Appendix B with reference to triangular and normal distribution of data points.)

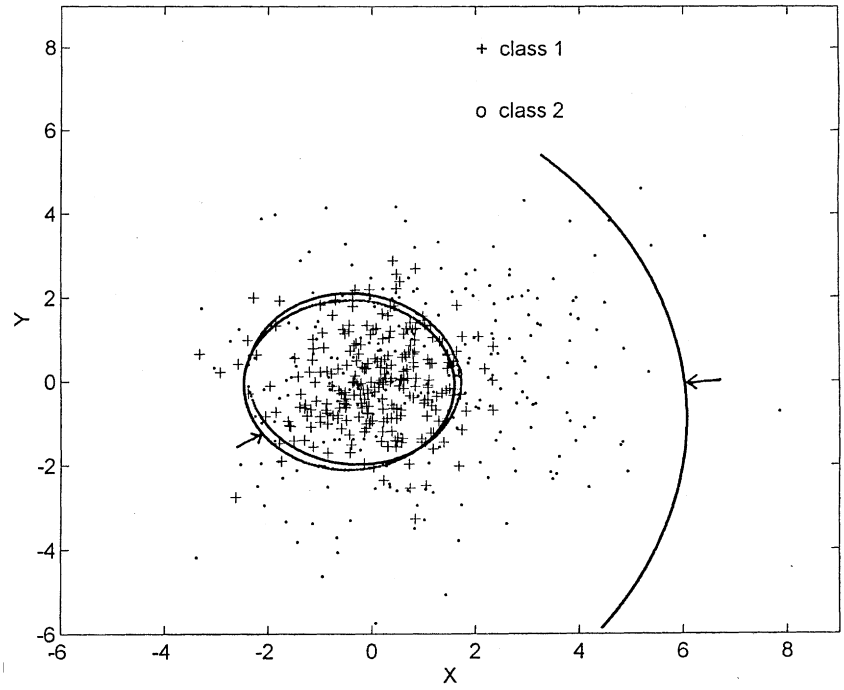

Fig. 17. Data set 2 for $n = 2000$ and the boundary provided by *GA-classifier* (marked with arrows) when circular segments are considered for $C = 2$ and $p = 0$. The Bayes decision boundary is also shown. (Only 500 data points are plotted for clarity.) The arc on the right is obviously redundant.

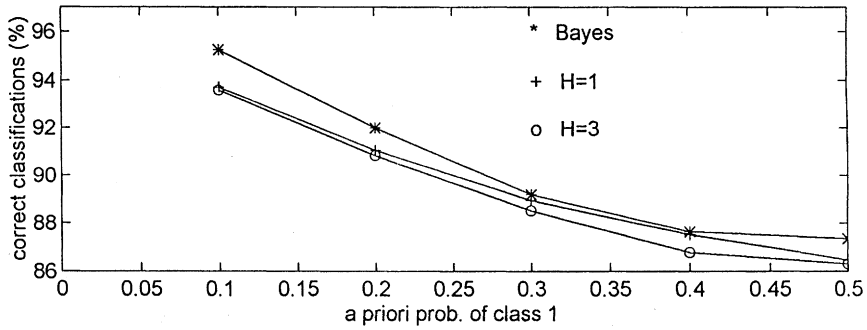

Fig. 18. Variation of recognition score of test data with P_1 for Data set 1 corresponding to Bayes classifier and *GA-classifier* ($H = 1$ and 3): $n = 200$.

It is shown in Appendix B that for triangular distribution the error probability varies symmetrically with P_1 having the maximum value for $P_1 = 0.5$. Similar observations were made in the investigation for Data set 1. Consequently, results are presented for P_1 lying in the range [0, 0.5] only. The test scores for the *GA-classifier*

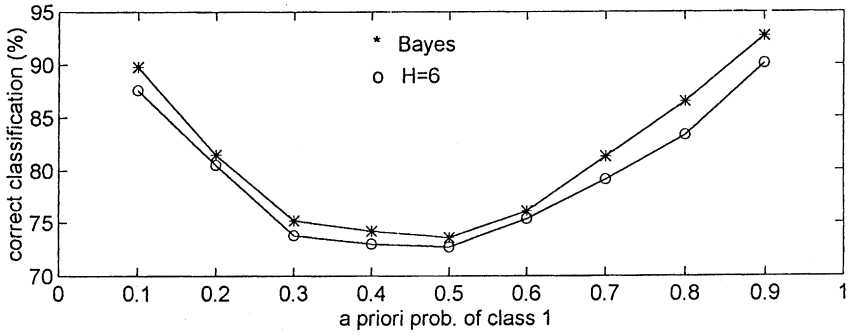

Fig. 19. Variation of recognition score of test data with P_1 for Data set 2 corresponding to Bayes classifier and *GA-classifier* ($H = 6$): $n = 200$.

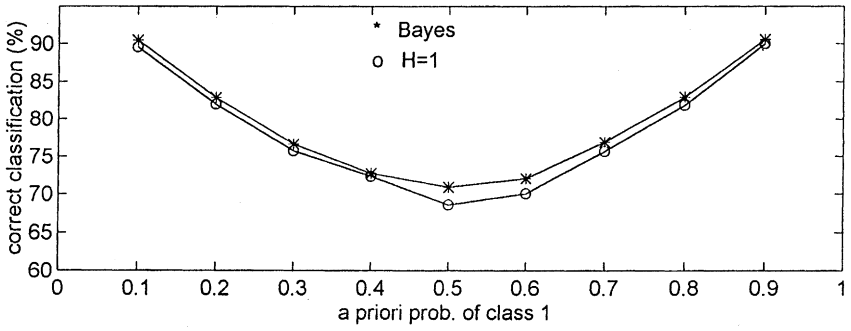

Fig. 20. Variation of recognition score of test data with P_1 for Data set 4 corresponding to Bayes classifier and *GA-classifier* ($H = 1$): $n = 200$.

with $H = 3$ are found to be consistently poorer than those with $H = 1$ for this data set (Fig. 18). It is to be mentioned here that recognition of the training data was better for $H = 3$ than for $H = 1$ (Table 1). This again supports the observations that increasing H beyond the optimum value leads to better training performance but poorer generalization of the *GA-classifier*.

7. Discussion and conclusions

A theoretical investigation is made here to find the relationship between a GA based classifier developed earlier [3] and the Bayes classifier. It is shown that for a sufficiently large number of training data points, and for a sufficiently large number of iterations the error rate during training obtained by the *GA-classifier* (where hyperplanes are considered to constitute the class boundaries) will be less than or equal to the Bayes error probability.

It is also shown that the number of hyperplanes provided by the *GA-classifier* for constituting the decision boundary will be optimum if the minimal partition providing the Bayes error probability is unique. Otherwise it will be greater than or equal to the optimum value (see Section 5). In either case, the classifier will still be optimal in terms of the number of misclassified training data points.

The experimental results on the different distribution of overlapping data with both linear and non-linear boundaries show that the decision regions provided by *GA-classifier* gradually approach the ones provided by Bayes classifier for a large number of training data points. For small values of n, the *GA-classifier* yields a significantly better recognition score during training. This is due to the reason that such a relatively small data set can be generated by many distributions, each providing a different error function. As the number of points increases, the number of distributions that are able to generate the points becomes smaller. In the limiting case, only one distribution will provide the optimal error. The performance of the *GA-classifier*, in such a case, will be same as that of the Bayes classifier.

When the decision boundaries are non-linear, the *GA-classifier* using circular segments performs better than the one using linear surfaces (Table 6, for Data set 2). On the other hand, with the decision boundary being actually linear, the *GA-classifier* with lines performs better. In real life, data sets are often overlapping with non-linear boundaries. Thus using higher-order surfaces for modeling the decision boundary appears to be better in real-life problems.

As far as the generalization capability is concerned, the Bayes classifier generalizes better than the *GA-classifier* for all the data sets, and also for all values of a priori class probability P_1. In fact, it is known in the literature that when the class a priori probabilities and the probability distributions are known, the Bayes classifier provides the optimal performance. No other classifier, in that case, can provide a better performance. However, in most practical cases, these quantities are usually unknown. It is in such situations that the *GA-classifier*, which is non-parametric in nature, can be utilized. It has already been shown in [3, 7] that the performance of the *GA-classifier* is comparable to (sometimes better than) that of some other standard classifiers.

Acknowledgements

This work was carried out when Ms. Sanghamitra Bandyopadhyay held a fellowship awarded by the Department of Atomic Energy, Govt. of India, and Professor S.K. Pal held the Jawaharlal Nehru Fellowship.

Appendix A

Let us consider a three class problem. Let the a priori probabilities for the three classes by P_1, P_2 and P_3 and the class conditional densities be $p_1(x)$, $p_2(x)$ and $p_3(x)$. Let the regions associated with the three classes be Ω_1, Ω_2 and Ω_3 such that $\Omega_i \cap \Omega_j = \emptyset$,

$\forall i \neq j$ and $\Omega_1 \cup \Omega_2 \cup \Omega_3 = \Omega$. Then the error probability e is given by

$$
\begin{aligned}
e &= \sum_{i=1}^{3} \int_{\Omega_i^c} P_i p_i(x)\,\mathrm{d}x \\
&= \int_{\Omega_2 \cup \Omega_3} P_1 p_1(x)\,\mathrm{d}x + \int_{\Omega_1 \cup \Omega_3} P_2 p_2(x)\,\mathrm{d}x + \int_{\Omega_1 \cup \Omega_2} P_3 p_3(x)\,\mathrm{d}x \\
&= \int_{\Omega_1} P_1 p_1(x)\,\mathrm{d}x - \int_{\Omega_1} P_1 p_1(x)\,\mathrm{d}x + \int_{\Omega_2 \cup \Omega_3} P_1 p_1(x)\,\mathrm{d}x + \int_{\Omega_1 \cup \Omega_3} P_2 p_2(x)\,\mathrm{d}x \\
&\quad + \int_{\Omega_1 \cup \Omega_2} P_3 p_3(x)\,\mathrm{d}x \\
&= P_1 - \int_{\Omega_1} P_1 p_1(x)\,\mathrm{d}x + \int_{\Omega_1 \cup \Omega_3} P_2 p_2(x)\,\mathrm{d}x + \int_{\Omega_1 \cup \Omega_2} P_3 p_3(x)\,\mathrm{d}x \\
&= P_1 + P_2 - \int_{\Omega_1} P_1 p_1(x)\,\mathrm{d}x - \int_{\Omega_2} P_2 p_2(x)\,\mathrm{d}x + \int_{\Omega_1 \cup \Omega_2} P_3 p_3(x)\,\mathrm{d}x \\
&= P_1 + P_2 + \int_{\Omega_1} (P_3 p_3(x) - P_1 p_1(x))\,\mathrm{d}x + \int_{\Omega_2} (P_3 p_3(x) - P_2 p_2(x))\,\mathrm{d}x.
\end{aligned}
$$

Similarly

$$
e = P_2 + P_3 + \int_{\Omega_2} (P_1 p_1(x) - P_2 p_2(x))\,\mathrm{d}x + \int_{\Omega_3} (P_1 p_1(x) - P_3 p_3(x))\,\mathrm{d}x
$$

and

$$
e = P_3 + P_1 + \int_{\Omega_3} (P_2 p_2(x) - P_3 p_3(x))\,\mathrm{d}x + \int_{\Omega_1} (P_2 p_2(x) - P_1 p_1(x))\,\mathrm{d}x.
$$

Summing up, we get

$$
\begin{aligned}
3e = 2 + \Bigg[& \int_{\Omega_1} ((P_3 p_3(x) - P_1 p_1(x)) + (P_2 p_2(x) - P_1 p_1(x)))\,\mathrm{d}x \\
& + \int_{\Omega_2} ((P_3 p_3(x) - P_2 p_2(x)) + (P_1 p_1(x) - P_2 p_2(x)))\,\mathrm{d}x \\
& + \int_{\Omega_3} ((P_1 p_1(x) - P_3 p_3(x)) + (P_2 p_2(x) - P_3 p_3(x)))\,\mathrm{d}x \Bigg].
\end{aligned}
$$

Let γ be used to represent the term in square brackets. Therefore $3e = 2 + \gamma$. Bayes classifier classifies a point x to the class which minimizes e, i.e., which in effect minimizes γ. Accordingly in order to classify a point x to one of the three classes the following cases may arise:

1. $\{x\colon P_1 p_1(x) > P_2 p_2(x) > P_3 p_3(x)\}$: classify to class 1.
2. $\{x\colon P_1 p_1(x) > P_2 p_2(x) = P_3 p_3(x)\}$: classify to class 1.

3. $\{x: P_1p_1(x) > P_3p_3(x) > P_2p_2(x)\}$: classify to class 1.
4. $\{x: P_1p_1(x) = P_3p_3(x) > P_2p_2(x)\}$: classify to class 1 or 3.
5. $\{x: P_3p_3(x) > P_1p_1(x) > P_2p_2(x)\}$: classify to class 3.
6. $\{x: P_3p_3(x) > P_1p_1(x) = P_2p_2(x)\}$: classify to class 3.
7. $\{x: P_3p_3(x) > P_2p_2(x) > P_1p_1(x)\}$: classify to class 3.
8. $\{x: P_3p_3(x) = P_2p_2(x) > P_1p_1(x)\}$: classify to class 3 or 2.
9. $\{x: P_2p_2(x) > P_3p_3(x) > P_1p_1(x)\}$: classify to class 2.
10. $\{x: P_2p_2(x) > P_3p_3(x) = P_1p_1(x)\}$: classify to class 2.
11. $\{x: P_2p_2(x) > P_1p_1(x) > P_3p_3(x)\}$: classify to class 2.
12. $\{x: P_2p_2(x) = P_1p_1(x) > P_3p_3(x)\}$: classify to class 2 or 1.
13. $\{x: P_1p_1(x) = P_2p_2(x) = P_3p_3(x)\}$: classify to any of the three classes.

As is obvious from the previous discussion, regions represented by cases 4, 8, 12 and 13 do not have unique classification associated with them. These regions may be included in more than one class provided they are non-empty, while still providing the least error probability.

Appendix B

As mentioned earlier (Eq. (3)), the Bayes error probability a is given by

$$a = \sum_{i=1}^{k} P_i \int_{S_{0i}^c} p_i(x)\,\mathrm{d}x,$$

where S_{0i} is the region for class i. For a two class problem a may be written as

$$a = P_1 \int_{S_{01}^c} p_1(x)\,\mathrm{d}x + P_2 \int_{S_{02}^c} p_2(x)\,\mathrm{d}x$$

Since $P_2 = 1 - P_1$, we get

$$a = P_1 \int_{S_{01}^c} p_1(x)\,\mathrm{d}x + (1 - P_1) \int_{S_{02}^c} p_2(x)\,\mathrm{d}x. \tag{B.1}$$

For the triangular distribution mentioned in Section 6.2 (for generating Data set 1), using Eq. (4) we may write

$$a = P_1 \int_{1+P_1}^{2} (2 - x)\,\mathrm{d}x + (1 - P_1) \int_{1}^{1+P_1} (x - 1)\,\mathrm{d}x.$$

Solving for a we get

$$a = P_1 \frac{(1 - P_1)}{2}.$$

Obviously, this is a symmetric function with minimum values at $P_1 = 0$ or 1, and maximum value at $P_1 = 0.5$. Thus, the recognition score of the Bayes classifier should be minimum for $P_1 = 0.5$, increasing symmetrically on both sides.

For normal distribution, it is very difficult to obtain a closed-form expression for a in terms of P_1. An analysis presented in [21, Section 3.1] indicates that the risk r associated with a particular decision is maximum for some value of $P_1 = P_1^*$, decreasing on both sides of this value when the regions associated with each class change with the class a priori probabilities.

Alternatively, one can also derive bounds on the error probabilities. One such bound for normal distribution is given by [21]

$$a \leqslant \sqrt{P_1 P_2}\, e^{-\mu(1/2)}, \tag{B.2}$$

where $\mu(1/2)$ is called the *Bhattacharyya distance*. Let us define the upper bound of a by a', i.e., $a' = \sqrt{P_1 P_2}\, e^{-\mu(1/2)}$, or

$$a' = \sqrt{P_1(1-P_1)}\, e^{-\mu(1/2)},$$

$$\frac{da'}{dP_1} = \frac{1}{2}\frac{1-2P_1}{\sqrt{P_1(1-P_1)}}\, e^{-\mu(1/2)}.$$

This shows that the error bound is maximum when $P_1 = P_2 = 0.5$.

References

[1] D.E. Goldberg, Genetic Algorithms, in: Search, Optimization and Machine Learning, Addison-Wesley, New York, 1989.

[2] L. Davis (Ed.), Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.

[3] S. Bandyopadhyay, C.A. Murthy, S.K. Pal, Pattern classification using genetic algorithms, Pattern Recog. Lett. 16 (8) (1995) 801–808.

[4] L. Davis, Job shop scheduling with genetic algorithms, in: J.J. Grefenstette (Ed.), Proc. 1st Int. Conf. Genetic Algorithms, Lawrence Earlbaum Associates, Hillsdale, NJ, 1985, pp. 136–140.

[5] J.J. Grefenstette (Ed.), Proc. 1st Int. Conf. Genetic Algorithms, Lawrence Earlbaum Associates, Hillsdale, NJ, 1985.

[6] S. Forrest (Ed.), Proc. 5th Int. Conf. Genetic Algorithms, Univ. of Illinois, Urbana Champaign, July 1993.

[7] S.K. Pal, S. Bandyopadhyay, C.A. Murthy, Genetic algorithms for generation of class boundaries, IEEE Trans. System Man Cybernet 28 (6) (1998) 816–828.

[8] S.K. Pal, P.P. Wang (Eds.), Genetic Algorithms for Pattern Recognition, CRC Press, Boca Raton, Fl, 1996.

[9] L.A. Rendall, A doubly layered genetic penetrance learning system, Proc. 3rd National Conf. Artificial Intelligence, 1983, pp. 343–347.

[10] J.R. Quinlan, Induction of decision trees, Machine Learning 3 (1986) 81–106.

[11] B.G. Buchanan, Can machine learning offer anything to expert systems ?, Machine Learning 4 (3/4) (1989) 251–254.

[12] R. Sikora, M. Shaw, A double layered genetic approach to acquiring rules for classification: integrating genetic algorithms with similarity based learning, ORSA J. Comput. 6 (2) (1994) 174–187.

[13] R. Srikanth, R. George, N. Warsi, D. Prabhu, F. Petry, B. Buckles, A variable-length genetic algorithm for clustering and classification, Pattern Recog. Lett. 16 (1995) 789–800.

[14] J.T. Tou, R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, Reading MA, 1974.

[15] D.W. Ruck, S.K. Rogers, M. Kabrisky, M.E. Oxley, B.W. Suter, The multilayer perceptron as an approximation to a Bayes optimal discriminant function, IEEE Trans. Neural. Networks 1 (1990) 436–438.

[16] J. Hertz, A. Krogh, R.G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, New York, 1991.
[17] D. Bhandari, C.A. Murthy, S.K. Pal, Genetic algorithm with elitist model and its convergence, Int. J. Pattern Recog. Artif. Intell. 10 (1996) 731–747.
[18] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[19] T.M. Apostol, Mathematical Analysis, Narosa Publishing House, New Delhi, 1985.
[20] R.B. Ash, Real Analysis and Probability, Academic Press, New York, 1972.
[21] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, New York, 1972.