Feature Subset Search using Genetic Algorithms

F. J. Ferri*†, V. Kadirkamanathan‡ and J. Kittler
Dpt. of Electronic & Electrical Engineering
University of Surrey
Guildford, Surrey GU2 5XH

Phone: (0483) 300800 (ext. 3026), Fax: (0483) 34139
email: F.Ferri@ee.surrey.ac.uk

Abstract

The purpose of this paper is the study and application of genetic algorithms to a classical problem in Pattern Recognition consisting of selecting the best subset of features from a complete set of measurements with respect to a certain criterion function which maximises the classification performance using such feature subset.
Though this is not the first attempt, the particular formulation of the Feature Selection problem adopted here has not been considered before using Genetic Algorithms. The results obtained show that Genetic Algorithms are powerful quasi-optimal methods for this task and are specially suited for problems involving large dimensionalities for which most of the classical methods require prohibitive amounts of computation.

1 Introduction

Feature Selection (FS) is known to be a critical step in the design of Pattern Recognition systems for several reasons. In many applications (e.g. medical diagnosis) there is a practical need to reduce the number of (possibly costly) measurements without significantly degrading the performance of the system. Moreover, in applications involving a very large number of features the performance of the classifier often degrades if the number of features used increases beyond a certain value. This is known as the *peaking phenomenon* and the dimensionality reduction effectively counteracts this problem.

FS methods aim at selecting a small or prespecified number of features leading to the best possible performance of the entire system. In order to assess the efectiveness of features, a criterion function is usually defined which reflects the degree of dissimilarity or probabilistic distance between classes. Ideally this criterion function should be closely related to the probability of misclassification of the corresponding classifier.

However, instead of using such probabilistic distances, it is possible to directly estimate the probability of error. Depending on how this estimate is obtained, the corresponding criterion function will be closer or not to the "true" probability of misclassification, but in general this kind of criterion is more realistic and convenient for most real problems of moderate size. In contrast, this kind of approach is not suitable for most large-scale problems due to the computational burden involved in the estimation process which depends on both dimensionality and number of samples available.

This work was supported by a SERC grant GR/E 97549
*Contact Person
†Supported by a FPI grant from MEC, Spain. Ref. PF92 73546684
‡Currently at Dpt. of Automatic Control & System Eng., University of Sheffield, Sheffield S1 4DU

When using probabilistic distances in FS it is very important that they exhibit the so-called *monotonicity property* because it is the prerequisite of several efficient search algorithms. Let X be the complete set of D available measurements. We denote by χ_d a subset of X containing d features, $\xi_1, \ldots, \xi_d$. A criterion function is said to be *monotonic* if:

$$\forall\ \chi_i, \chi_j\ :\ \ \chi_i \subset \chi_j \Longrightarrow J(\chi_i) \leq J(\chi_j)$$

Most of the probabilistic distance measures commonly used in practice (e.g., the Mahalanobis and Bhattacharyya distances, etc) are known to fulfil this property [5], while other approaches which use as a criterion function direct estimates of the misclassification error of some kind of classifier, this property is supposed to hold at least approximately [2] under certain conditions.

Provided the criterion function has been appropriately chosen, the problem of FS reduces to a search in the feature space for a *feasible* solution which maximises(minimises) a certain figure of merit. This optimisation process can be formulated in several slightly different ways. For example, one can minimise the number of features considering as feasible any subset whose criterion value is above a certain threshold [2]. In the present work, we address the problem of selecting the best d-subset of features out of the set of D measurements with respect to the criterion function. Thus, we want to maximise the criterion function considering all the d-subsets of features as feasible solutions [5].

One obvious disadvantage of the first formulation of the FS problem is the difficulty in the definition of an appropriate treshold specially when using non-monotonic criteria in which the peaking phenomenon may be present. In this case, it is possible to miss the overall optimal solution and obtain a smaller suboptimal (but feasible) subset of features.

Since the introduction of the optimal (but time-consuming) *Branch and Bound* (B&B) algorithm for FS, a considerable effort has been directed to developing other (suboptimal) search methods to be applied when the total number of features is large. Many different sequential search methods have been proposed for such situations [1]. Some of them have been shown to approach the performance of the B&B method for moderate-size problems with much less computational effort [6].

This work addresses the problem of applying Genetic Algorithms (GA) to the problem of feature subset search. GAs are known to be appropriate search methods for such hard problems [4, 3]. They exploit implicit parallelism along with the tendency of "genetic" operators to create and propagate promising candidates so as to obtain good feasible solutions for such optimization problems. Moreover, the GAs incorporate randomness to the search that seems to improve the results of most of the classical (deterministic) feature search algorithms.

2 Genetic Algorithms in Feature Selection

GAs are a class of search methods deeply inspired by the natural process of evolution. In each iteration of the algorithm *(generation)*, a fixed number *(population)* of possible solutions *(chromosomes)* is generated by means of applying certain "genetic" operators in a stochastic process guided by a *fitness measure*. The most important and commonly used genetic operators are *recombination*, *crossover* and *mutation*. The result is a probabilistic algorithm which obtains good (nearly optimal) solutions for problems in which classical methods fail or are not applicable. A particular GA is identified by a particular method of *coding* the solutions into strings of some alphabet (usually binary), a particular form of the *genetic operators* adopted, and a particular definition of the *fitness function*.

GAs have already been applied to a different formulation of the feature selection problem [8, 7]. As in this previous attempt, the problem considered herein is not in a form amenable to a direct GA optimization. Even though the space of solutions has a direct and natural translation into strings of bits, the definition of an appropriate fitness function and a meaningful crossover operator is far from obvious.

The coding of the possible solutions adopted for the problem consists of strings of D bits, $\alpha_1, \ldots, \alpha_D$, where $\alpha_i = 0$ if the feature i is excluded from the subset, and $\alpha_i = 1$ if not.

Genetic Algorithm

Input:

$\mathcal{P}$: Randomly initialised population
p_c, p_m: Crossover and Mutation rates
MAXGEN: Maximum number of generations
N: Population size, $|\mathcal{P}|$

Output:

x: Best individual from current $\mathcal{P}$

Method:

EvaluateFitness($\mathcal{P}$)
while (generation<MAXGEN)$\wedge$(NotConvergence) *do*
  $\mathcal{M} \leftarrow$ Recombine($\mathcal{P}$)
  $\mathcal{O} \leftarrow$ Crossover($\mathcal{M}, p_c$)
  $\mathcal{O} \leftarrow$ Mutate($\mathcal{O}, p_m$)
  EvaluateFitness($\mathcal{P}$)
  $\mathcal{P} \leftarrow$ Select($\mathcal{P}, \mathcal{O}$)
  generation $\leftarrow$ generation + 1
endwhile

The genetic operators used for this task are the standard one-point-crossover and mutation. The population in each generation is selected from among the previous populations and its offspring according to the expected value of the fitness function (weighted roulette wheel rule). And the best individual in each iteration is automatically selected for reproduction without mutation [3].

The definition of an appropriate fitness function requires a detailed analysis of the behaviour of the criterion function which is to be minimised. In the present work both probabilistic distance measures and direct estimates of the misclassification rate are considered as objective function. Because of the monotonicity property in the first case and the same approximate tendency in the second, these functions cannot be readily optimised if we allow search in feature spaces of all dimensionalities. In other words the main problem in designing a GA for this task arises from the fact that the final solution is constrained to have a fixed number of features. This implies a problem in using the standard crossover operator since it can make the current offspring to fall into the non-feasible region ($d' \neq d$). But, if on the contrary we constrain all the individuals in the population to be in the feasible region (by using for example penalty functions or repairing algorithms [3]) we can loose the ability of the system to propagate the parts (schemata) of the best (but non-feasible) individuals to the next generation.

One way to implicitly constrain the final solution to be in the feasible region consists of defining a modified fitness function using an appropriate penalty function to weigh the criterion function depending on the difference between the required number of features and the total number of features. This weighting function has to be carefully chosen because if only the solutions in the feasible region are allowed to proliferate, the crossover operator will be highly inefficient and the corresponding GA will become practically a plain random search. On the other hand, if we consider weaker constraints, solutions with higher number of features can proliferate more easily than feasible solutions.

The penalty function used in the present approach is used not only to keep the search close to the feasible region but also to modify the behaviour of the criterion function. Without an appropriate penalty function the GA would tend to select a high number of features because of the monotonicity tendency. As a consequence, the following fitness function has been defined.

$$f(\chi_{d'}) = \begin{cases} J(\chi_{d'}) \cdot e^{-\frac{(d'-d)}{1/\sigma}} & d' \neq d \\ J(\chi_{d'}) & d' = d \end{cases}$$

In the present work, the weighting function defined is allowed to change its "width", $1/\sigma$, in each iteration of the algorithm, t, depending on the number of features of the current best solution, $d^\star$, and the required number of features according to the following expression.

$$\sigma(t+1) \;=\; \sigma(t) + \rho \mid d^\star - d \mid$$

The parameter ρ controls the rate of change of the weighting function. In this way, starting with a wider weighting function ($\sigma = 0$), the GA begins selecting subsets with many features in a natural way (due to the monotonicity property of the criterion function) and this behaviour is progresively modified finishing with a sufficiently narrow weighting function to keep the solution in the feasible region while allowing local search in the neighboring non-feasible zones. The value of this parameter should be chosen in a way that during an appropriate number of generations the GA is searching in the whole feature space (e.g. during half the total number of generations) and after that the search is concentrated only in the near-feasible region. This behaviour of the present GA implementation can be seen in Figure 1 where plots of the fitness function and criterion (Mahalanobis distance) both for the best and the average of the population are shown for 200 generations.

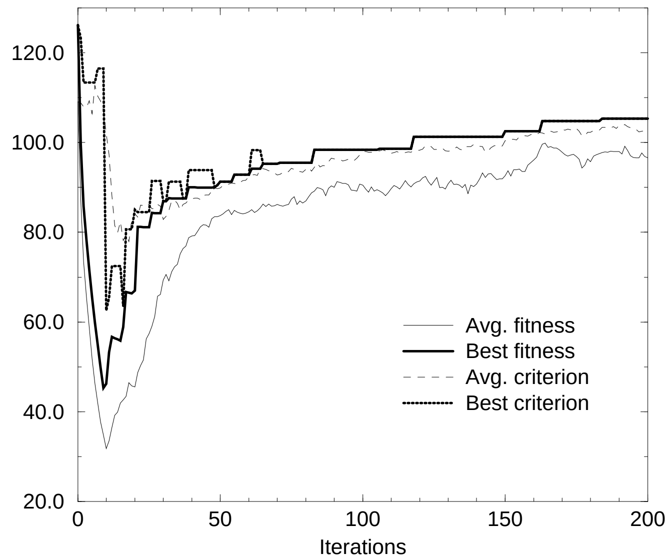

Figure 1: Fitness value and criterion value (Mahalanobis distance) of the best individual and averaged values over 200 generations of the GA selecting 18 features out of 30.

3 Experiments and Results

Several experiments have been conducted to analyse the behaviour of the GA as applied to FS problems considering different dimensionalities and criterion functions. The purpose of the experiments is to assess the validity of the method with low dimensional problems using the two kinds of criteria discussed and then apply the method to a high dimensional and more realistic problem to emphasise the speed-up obtained in most of the cases with respect to the classical applicable methods.

First, a problem with moderate dimensionality consisting of 1418 pattern vectors was considered. These pattern vectors correspond to the utterances "YES" and "NO" spoken over the

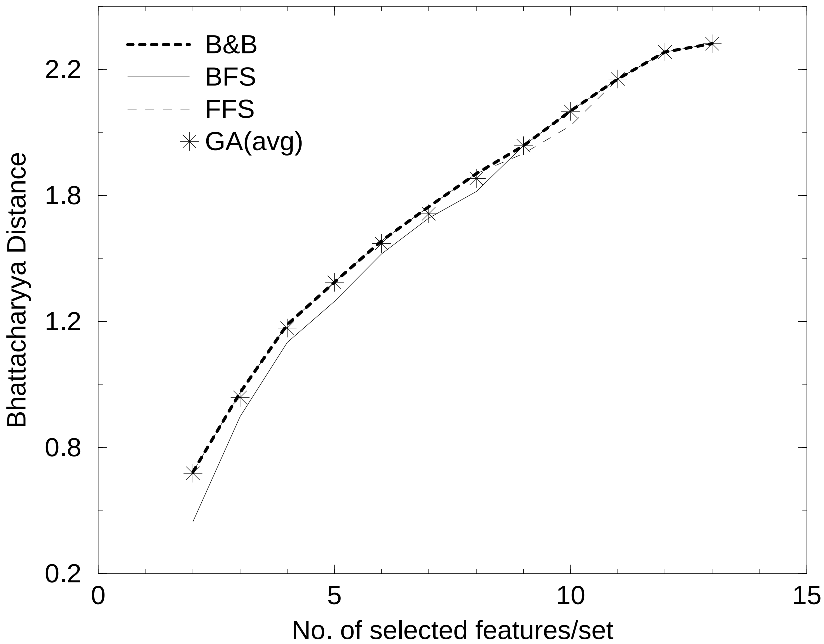

Figure 2: Comparative results of different Feature Selection methods over 15-dimensional speech data.

public switched telephone network. Each 15-dimensional feature vector contains 5 sets of 3 features derived by low order linear prediction analysis. All data was supplied by British Telecom [1].

For this problem it is possible to assume that both classes have a gaussian distribution. As a consequence we can consider the Bhattacharyya distance as a criterion function. The B&B algorithm and the forward and backward floating search methods (FFS and BFS) are applied to this problem to compare the results obtained with the ones from the GA approach.

Typical settings for GAs have been used, i.e. mutation rate 0.005, crossover rate 0.6 and population size 20. Further experimentation showed that the algorithm has a low sensitivity to changes of these parameters. Using these settings a value of $\rho = 0.001$ has proved to be appropriate to keep the search in the feasible zone after the first 100 generations. In almost all the cases the algorithm gave the final solution before 200 generations. This leads to a computational burden comparable to the B&B one and not far from the other sequential methods. The results corresponding to the average of 5 different runs of the GA are shown in Figure 2. The results obtained with the B&B algorithm and FFS and BFS methods have also been considered and are displayed for comparison purposes.

The same problem using estimates of the Recognition Rate of the corresponding gaussian classifier as a criterion function was also considered. This case is specially interesting because of the non-monotonicity of the criterion function and because of the behaviour of the GA in the presence of the peaking phenomenon. In fact, due to this peak in the performance, the algorithm obtained a solution faster when looking for feature sets near the peak. The results obtained are shown in Figure 3.

After these promising results with a small scale problem, a two-class document classification problem is considered. The data set consists of 2000 30-dimensional pattern vectors corresponding to optical measures extracted from certain discriminative parts of the documents.

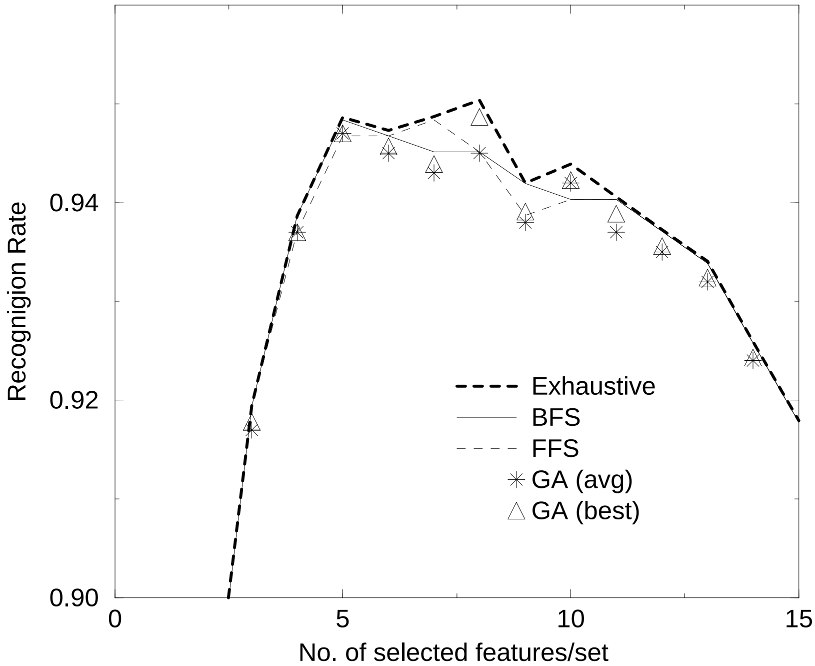

Figure 3: Comparative results of different Feature Selection methods over 15-dimensional speech data.

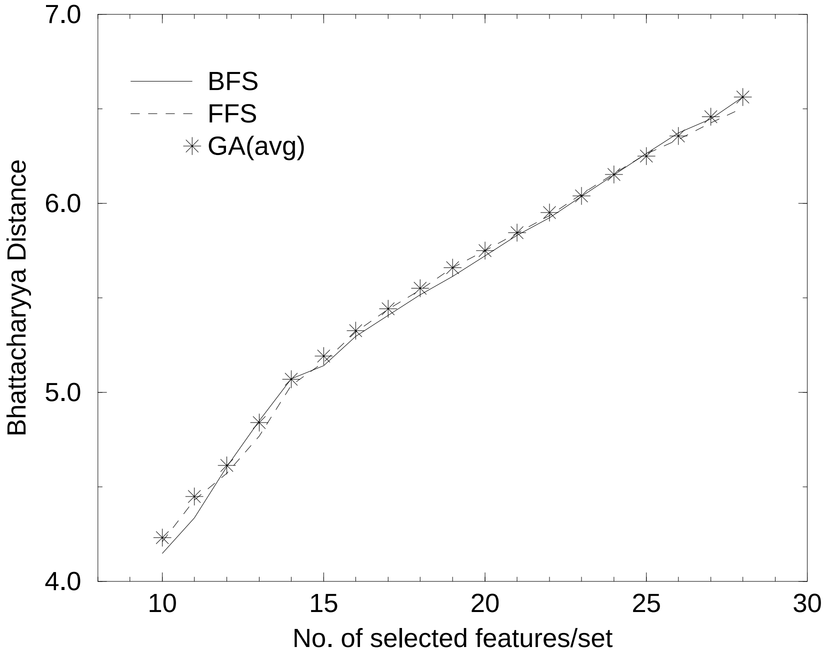

Figure 4: Comparative results of different Feature Selection methods over 30-dimensional optical data.

Even though for this problem the B&B algorithm is still applicable we do not consider it because the prohibitive amount of computation needed. Consequently, only the BFS and FFS methods were considered apart from the GA approach. The same behaviour as in the previous problem was exhibited by the GA when applied to this high dimensional problem. As shown in Figure 4, the results yielded by all the above methods are quite similar. Nevertheless, the GA obtained at least as good or better solutions than the BFS and FFS methods but with a considerable saving in the number of nodes or subsets evaluated which ranked from 25% to 50% with respect to the forward floating algorithm. Compared to the backward floating algorithm the savings were quite similar but this method was faster than the other one for subsets with cardinality close to 30.

4 Conclusions and Further Work

The use of GAs for feature subset search seems to be quite promising for dealing with high dimensional problems. The results reported here confirm the claims in [8] concerning the applicability of GAs to feature subset search problems and demonstrate their utility in the context of the Feature Selection problem formulation considered here which is distinct from the previous work reported in [8, 7].

Despite this, there are still some open questions. First, the extreme complexity of the space makes the GA inefficient when compared to other applications of GAs. Moreover, in our approach, the penalty functions need to be "adaptive" to appropriately guide the search to the (nearly) optimal feasible solutions. This implies the use of extra parameters which in principle have to be tuned for each different problem and/or criterion function.

Second, as a result of the peculiarities of the search space, the local search in the final steps of the GA is unusually "wide" leading to an extra difficulty in defining an appropriate convergence criterion. In consequence there is a need to run the GA a sufficiently large (and fixed) number of generations to assure the convergence of the algorithm to a nearly optimal solution.

Answers to these questions appear to lead to a modification of the GA itself rather than looking for more appropriate penalty functions. This could be done by including some of the heuristics present in the sequential methods of FS into the genetic operators, in particular into recombination and crossover.

References

[1] N. Choakjarernwanit. *Feature Selection in statistical pattern recognition.* PhD thesis, University of Surrey, England, 1992.

[2] I. Foroutan and J. Sklansky. Feature Selection for Automatic Classification on Non-Gaussian Data. *IEEE Transactions on Systems Man and Cybernetics*, SMC-17(2):187–198, March/April 1987.

[3] D. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison-Wesley, Reading, MA, 1989.

[4] J. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI, 1975.

[5] J. Kittler. Feature Set Search Algorithms. In *Pattern Recognition and Signal Processing, C. H. Chen, Ed.*, pages 41–60, The Netherlands: Sijthoff and Noordhoff, 1978.

[6] R. Pecinovský, P. Pudil, and S. Bláha. The Algorithms for Sequential Feature Selection Based on the Measure of Discriminative Power. In *Proc of DIANA – Conf. on Discr. Anal., Cluster Anal. and other methods on Data Class., Liblice*, pages 277–284, Czechoslovakia, 1982.

[7] W. Siedlecki and J. Sklansky. Constrained Genetic Optimization via Dynamic Reward-Penalty Bal ancing and Its Use in Pattern Recognition. In *Proceedings of the International Conference on Genetic Algorithms*, pages 141–150, San Mateo, California, 1989.

[8] W. Siedlecki and J. Sklansky. A Note on Genetic Algorithm for Large-scale Feature Selection. *Pattern Recognition Letters*, 10(5):335–347, November 1989.