# Comparative Study of Techniques for Large-Scale Feature Selection*

F.J. Ferri[a], P. Pudil[b], M. Hatef[c] and J. Kittler[c]

[a] Dept. Informàtica i Electrònica. Universitat de València
46100 Burjassot (València) Spain

[b] Inst. of Information Theory and Automation. Academy of Sciencies of the Czech Republic. 18208, Prague. Czech Republic

[c] Dept. of Electronic and Electrical Engineering. University of Surrey
Guildford GU2 5XH, United Kingdom

The combinatorial search problem arising in feature selection in high dimensional spaces is considered. Recently developed techniques based on the classical sequential methods and the $(l, r)$ search called Floating search algorithms are compared against the Genetic approach to feature subset search. Both approaches have been designed with the view to give a good compromise between efficiency and effectiveness for large problems. The purpose of this paper is to investigate the applicability of these techniques to high dimensional problems of Feature Selection. The aim is to establish whether the properties inferred for these techniques from medium scale experiments involving up to a few tens of dimensions extend to dimensionalities of one order of magnitude higher. Further, relative merits of these techniques vis-a-vis such high dimensional problems are explored and the possibility of exploiting the best aspects of these methods to create a composite feature selection procedure with superior properties is considered.

## 1. INTRODUCTION

The problem in Feature Selection (FS) can be easily stated as the search for a sufficiently reduced subset of, say, $d$ features out of the total number of available ones, $D$, without significantly degrading (or even improving in some cases) the performance of the resulting classifier when using either set of features. This search problem is driven by a certain measure of performance or *criterion function* which is used to assess the validity of each feature subset. This criterion has to be related to the final performance measure of the resulting classifier, i.e. its recognition rate.

Recently the battery of tools for feature selection [2] has been augmented by a few im-

*This work was suported by a SERC grant GR/E 97549. The first author was also supported by a FPI grant from the Spanish MEC, PF92 73546684

portant techniques. First of all, the family of sequential feature set search procedures [5] has been extended by the Floating Sequential Search Methods [7] in which the number of forward and backtracking steps is dynamically controlled instead of being fixed before hand as in the *plus $l$ – take away $r$* algorithm (or $(l, r)$-search [10]). Other important contributions include the approaches based on Genetic Algorithms (GA) [9,3] which appear to offer an attractive alternative to heuristic tree search methods. However, none of these techniques has been thoroughly tested on large scale feature selection problems, involving hundreds rather than tens of features.

The purpose of this paper is to investigate the applicability of the recently proposed techniques to high dimensional problems of FS. The aim is to establish whether the properties inferred for these techniques from medium scale experiments involving up to a few tens of dimensions extend to dimensionalities of one order of magnitude higher. Further, relative merits of these techniques vis-a-vis such high dimensional problems will be explored and the possibility of exploiting the best aspects of these methods to create a composite feature selection procedure with superior properties will be considered.

In the next section the different approaches to FS considered in this work are outlined and its main similarities and differences are put forward. Section 3 discusses some implementation details about the particular type of FS problems considered. All the experiments carried out to validate the conclusions presented in this paper are explained in Section 4 and the main implications of the results obtained are discussed in Section 5. Section 6 sumarises the main conclusions about the presented work.

```
plus $l - $ take away $r$ Algorithm
Input:            $Y = \{y_j \mid j = 1,\dots,D\}$ //available measurements//
Output:           $X_k = \{x_j \mid j = 1,\dots,k,\ x_j \in Y\},\ k = 0,1,\dots,D$
Initialisation:   if $l > r$   then $k := 0$; $X_0 := \emptyset$; go to Step 1
                               else $k := D$; $X_D := Y$; go to Step 2
Termination:      Stop when $k$ equals the number of features required

Step 1 (Inclusion)
    repeat $l$ times
        $x^+ := \arg\max_{x \in Y - X_k} J(X_k + x)$      {the most significant feature[2] with respect to $X_k$
        $X_{k+1} := X_k + x^+$; $k := k + 1$
    go to Step 2
Step 2 (Exclusion)
    repeat $r$ times
        $x^- := \arg\max_{x \in X_k} J(X_k - x)$      {the least significant feature in $X_k$
        $X_{k-1} := X_k - x^-$; $k := k - 1$
    go to Step 1
```

Figure 1. *plus $l$ – take away $r$* algorithm.

[2] Note that we mean by this that $x^+$ satisfies $J(X_k + x^+) = \max_{x \in Y - X_k} J(X_k + x)$

## 2. FEATURE SUBSET SEARCH ALGORITHMS

### 2.1. Sequential Search Algorithms

The well-known Sequential Forward Selection (SFS) and its backward counterpart (SBS) are suboptimal methods that obtain a chain of nested subsets of features in a straightforward manner, i.e. by adding (substracting) the locally best (worst) feature in the set. This nesting effect constitutes one of their main drawbacks. The algorithms cannot correct previous additions (deletions) of features. This methods are particular cases of the more general *plus $l$ – take away $r$* method [10].

The *plus $l$ – take away $r$* method also called $(l, r)$ method, consists of applying SFS during $l$ steps followed by $r$ steps of SBS with the cycle of forward and backward selection repeated until the required number of features is reached. Even though with this procedures the problem of nested features can be partially overcome, other important problem arises. There is no way of predicting the best values of $l$ and $r$ to obtain good enough solutions with a moderate amount of computation.

The *plus $l$ – take away $r$* method and consequently the SFS and SBS methods (respectively (1,0) and (0,1) methods) can be conveniently expressed as shown in Figure 1.

```
SFFS Algorithm
Input:            $Y = \{y_j \mid j = 1,\ldots,D\}$ //available measurements//
Output:           $X_k = \{x_j \mid j = 1,\ldots,k,\ x_j \in Y\},\ k = 0,1,\ldots,D$
Initialisation:   $X_0 := \emptyset;\quad k := 0$
      (in practice one can begin with $k = 2$ by applying SFS twice)
Termination:      Stop when $k$ equals the number of features required

Step 1 (Inclusion)
      $x^+ := \arg\max_{x \in Y - X_k} J(X_k + x)$      {the most significant feature with respect to $X_k$
      $X_{k+1} := X_k + x^+;\ k := k + 1$
Step 2 (Conditional Exclusion)
      $x^- := \arg\max_{x \in X_k} J(X_k - x)$      {the least significant feature in $X_k$
      if $J(X_k - \{x^-\}) > J(X_{k-1})$ then
              $X_{k-1} := X_k - x^-;\ k := k - 1$
              go to Step 2
      else
              go to Step 1
```

Figure 2. Sequential Floating Forward Algorithm.

The aim behind the above methods can be more efficiently implemented by considering conditional inclusion and exclusion of features. The Sequential Floating Forward Selection (SFFS) procedure consists of applying after each forward step a number of backward steps as long as the corresponding subsets are better than the previously evaluated ones at that

**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
```

Figure 3. Genetic Algorithm.

level. The same applies for the Sequential Floating Backward Selection (SBFS) procedure. Thus backtracking in these algorithms is controlled dynamically and, as a consequence, no parameter setting is needed at all. A detailed description of these algorithms is given in [8]. An outline of the SFFS algorithm is shown in Figure 2

## 2.2. The Genetic Algorithm Approach

The GA approach to FS constitutes a different way of looking for features since it allows a randomised search guided by a certain fitness measure. 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*.

The coding used to represent feature subsets consists of strings of $D$ bits, $\alpha_1, \ldots, \alpha_D$, where $\alpha_i = 1$ if the feature $i$ is in the subset and $\alpha_i = 0$ otherwise. This coding allows the use of all the standard genetic operators.

The first GA approach [9] incorporates an appropriate penalty function to force the algorithm to search those feature subsets near to the feasibility boundary (threshold on the criterion function). Even though it was shown that this approach compares favourably with the (2,1)-search for a 30-dimensional problem using error estimates of the 5-Nearest Neighbour rule, no study of other criterion functions and how this approach behaves as the dimensionality increases has been reported.

The other GA approach has been shown to be a kind of randomised backward search [3]. The approach consists of allowing the algorithm to select the best subsets depending on its criterion only and then progressively weighting the solutions to make the algorithm finish with the prespecified number of features. This algorithm was shown to perform reasonably well as compared to sequential methods including Floating search. But a number of problems arise specially for large dimensionalities mainly due the above mentioned backward behaviour.

## 3. IMPLEMENTATION DETAILS

The main problem when applying FS to large dimensional problems comes from the fact that the criterion function must be evaluated in spaces of high dimensionality. Specially when parametric approaches to classification are used, the inversion of (possibly rank deficient) covariance matrices can lead to serious problems. From this point of view the forward methods are better suited for this kind of problems because one can go forward until the matrices cannot be inverted.

As a consequence of this, only the different sequential forward approaches are considered in this study. In particular, in the experiments reported in the next section, only the results corresponding to the SFFS method (the best) and to the SFS method (for reference purposes) are explicitly shown in the figures.

For the same reason, only the first genetic approach is considered for the experimental study considering large problems. In this approach, the smallest subset of features for which the criterion function is above a certain specified *feasibility threshold* is searched for. To allow the GA to handle this constrained optimization problem, the following penalty function is introduced [9].

$$\phi(J) = \frac{e^{\frac{J-t}{m}}}{e-1} \tag{1}$$

where $t$ and $m$ are the feasibility threshold and margin respectively. The *feasibility margin*, $m$, controlls the width of the search by changing the weighting function. From this penalty function the criterion to be minimised by the GA is constructed for each feature subset, $X_i$, as

$$F(X_i) = |X_i| + \phi(J(X_i)) \tag{2}$$

This function can readily be converted into a standard fitness function to be used with the GA using the maximum value of $F$ at each generation. For example, as in [9].

$$f(X_i) = (1+\epsilon)\max_{X_j \in \mathcal{P}} F(X_j) - F(X_i) \tag{3}$$

The particular GA settings used in this study consists of the usual 2-point crossover and mutation along with the rank-based selection scheme with an elitist strategy [4].

The particular value of the feasibility threshold, $t$, will be selected according to the available knowledge about the problem. In principle, a sequential method can be used prior to the application of the GA, and then, the parameter $t$ is fixed according to the result obtained with this method. the search. The results obtained may depend on the parameter $m$, but, as in the original paper, we have found a value of $m$ between $.01J_m$ and $.005J_m$ particularly suitable for a wide range of problems, where $J_m$ is either the maximum obtained value of the criterion or 1 if we use the recognition rate as a criterion function.

## 4. EXPERIMENTS

We consider first a 20-dimensional Diagnostic problem [1] for which both floating and GA approaches seem to perform reasonably well using various criterion functions with the only difference that all GA approaches need a certain "parameter tuning". Moreover, GA approaches cannot benefit from the fast computation of probabilistic distances and consequently they are highly inefficient in this case.

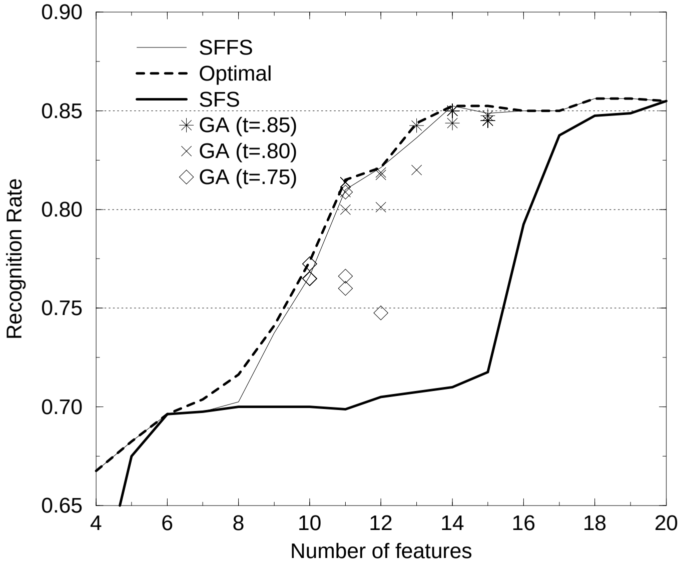

Figure 4. Recognition Rate for different number of features obtained by SFS, SFFS and Exhaustive search. Different runs of the GA are also shown using symbols.

Figure 4 shows the results obtained with the GA with different feasibility threshold values using the parameter setting suggested in [9], compared to the optimal and SFFS ones considering the recognition rate as a criterion function (when the Mahalanobis or Battacharyya distances are used, all the methods considered obtain approximately the same –good– results for this problem). For this medium size problem both approaches obtained similar results. Note, the GA led to the optimal solution in comparable time (about 1500 subset evaluations) even taking into account the need to run it a number of times to achieve good performance. In this experiment, the GA was run 10 times for each value of $t$ and, in more than half of the cases the GA obtained better or the same results than the SFFS ones (the figure can be misleading in this sense because each plotted symbol may represent more than one result).

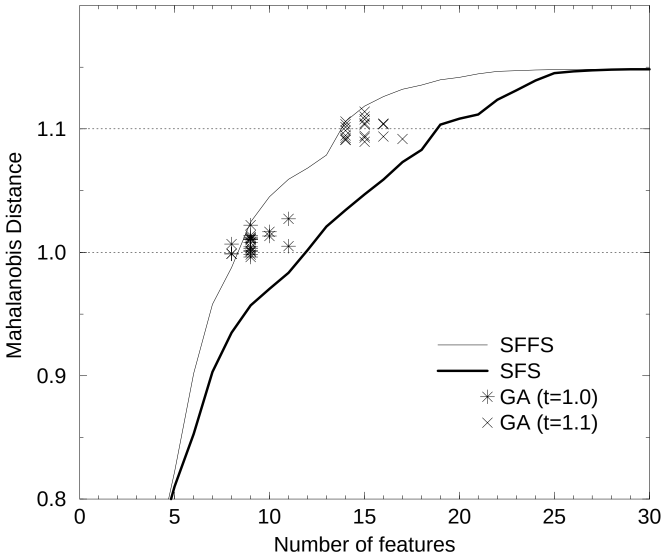

Figure 5. Results of Feature Selection obtained by SFS and SFFS methods for the $D = 30$ experiment. Crosses and asterisks show the results corresponding to different runs of the GA with two different values of the threshold parameter.

We then considered a document recognition problem in which the problem consists of discriminating between correct and defective records of banking documents consisting of 360 optical measurements. Previous experimentation showed that both classes could be conveniently modelled by Gaussian distributions.

To study the behaviour of the different methods as the dimensionality increases, we considered different subproblems acording to the number of features taken into account in the FS process. In particular, values of $D = 30$, 50, 120 and 360 were considered. For this time-consuming experiment only the (generalised) Mahalanobis distance was considered as criterion function.

In order to minimise the effects of random starting points in the GA optimisation process, 20 different runs of the GA were carried out in the $D = 30$ and $D = 50$ experiments,

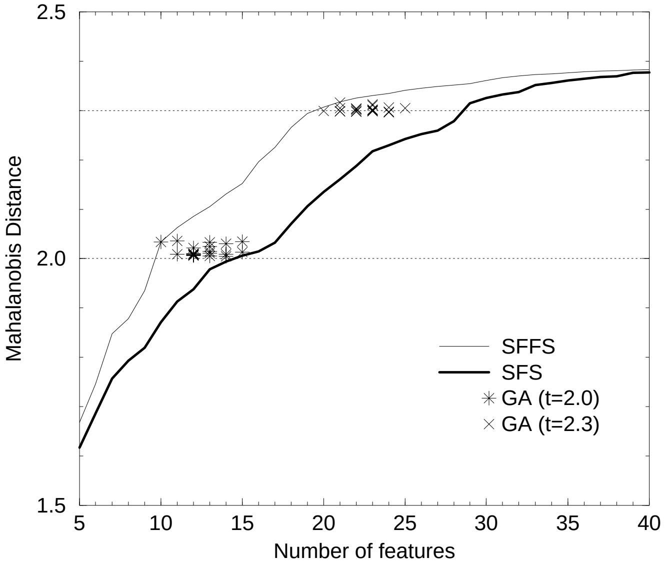

Figure 6. Results of Feature Selection obtained by SFS and SFFS methods for the $D = 50$ experiment. Crosses and asterisks show the results corresponding to different runs of the GA with two different values of the threshold parameter.

and only 5 runs in the $D = 120$ and $D = 360$ ones. In all the cases the GA was forced to stop when the total number of trials (criterion functions evaluated) reached the same number of trials as for the SFFS algorithm (approximately, 5000, 15000, 100000 and 500000[3]). Both SFS and SFFS results are displayed in the figures along with the GA results for comparison purposes.

Althoug many different parameter settings were tried for the GA approach (both genetic options as well as criterion threshold and margin), only the results corresponding to the ones explained in Section 3 are shown in the figures.

For the $D = 30$ subproblem, the Figure 5 shows that the results of the GA algorithm are quite similar to the ones from SFFS. Some of the GA runs obtained better results than the SFFS.

Figure 6 shows the corresponding results for the $D = 50$ subproblem. The results obtained in this case for both approaches exhibit the same behaviour as in the previous subproblem. The results obtained by the GA range from the “good” ones from SFFS to the “bad” ones from SFS.

This tendency of the GA to obtain deteriorating results as the dimensionality increases is clearly confirmed in Figure 7 in which the results corresponding to $D = 120$ are shown. In this case, some of the results from the GA are even worse than the ones from SFS.

The results for the original problem with $D = 360$ are even more discouraging from

[3] in the case $D = 360$ the SFFS and SFS algorithms were only run from $d = 1$ to $d = 120$

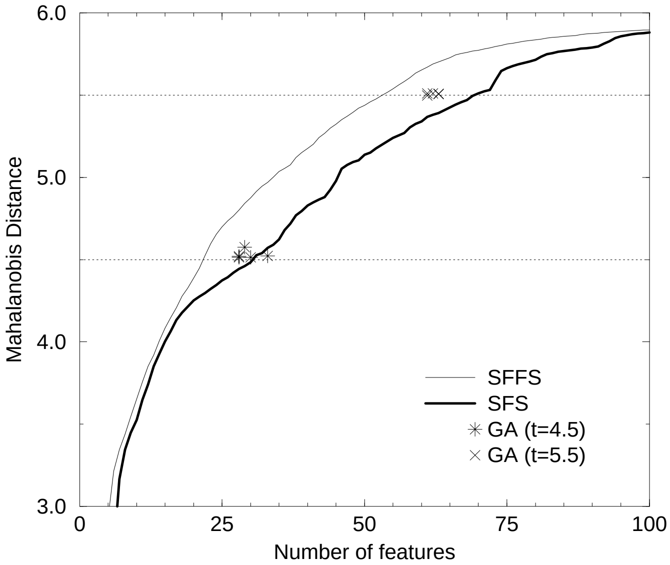

Figure 7. Results of Feature Selection obtained by SFS and SFFS methods for the $D = 120$ experiment. Crosses and asterisks show the results corresponding to different runs of the GA with two different values of the threshold parameter.

the point of view of the GA approach compared to the SFFS algorithm. In fact, the GA results hardly approached the results of the SFS algorithm.

Additional experimentation has been carried out to test both approaches in a more realistic case when the criterion function used is known to be nonmonotonic. To this end, estimates of the recognition rate of the gaussian classifier are used as criterion function and experiments for $D = 50$ and $D = 360$ only where repeated.

The results corresponding to $D = 50$ are shown in Figure 8. The time-consuming case of $D = 360$ was repeated only 5 times for the GA approach and from 1 to 80 features for the SFS and SFFS algorithms.

In the "full" experiment, $D = 360$ using the recognition rate as criterion function, the SFFS algorithm allowed us to obtain feature subsets of up to 80 features using the recognition rate as a criterion function. The best solution consisted of 76 features with 99.9% recognition rate while the best solution obtained by the GA for this problem was a subset of 74 features with a 97.6% recognition rate. For this experiment the GA was forced to stop so that the number of subsets evaluated by both approaches were approximately the same as for the SFFS algorithm (50,000).

## 5. DISCUSSION

From the results obtained for both families of methods it is possible to draw some conclusions. In principle, this work confirms the results in [9] about moderate size (20 –

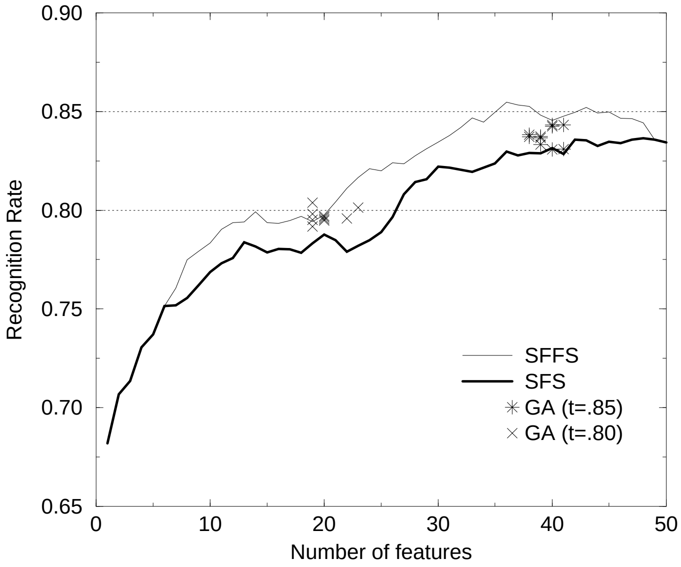

Figure 8. Results of Feature Selection obtained by SFS and SFFS methods for the $D = 50$ experiment using the recognition rate as a criterion function. Crosses and asterisks show the results corresponding to different runs of the GA with two different values of the threshold parameter.

30 features) problems. In this case the only contribution consists of comparing the GA with the SFFS algorithm. In this sense, we can say that both algorithms obtain similar results both in performance and efficiency.

The other interesting fact is the behaviour of both approaches when the dimensionality increases. As we can see in the figures, this effect makes the GA results worse and worse on average. It appears from these experiments, that the region of the search space the GA needs to explore increases faster than the region searched by the SFFS algorithm. If this were true it would be in contradiction with the conclusions in [9] where a *linear* increase of time complexity was postulated for this GA approach!

Another interesting behaviour that can be extracted from these experiments is the dependence of the GA results on the problem and, in particular, on the shape of the criterion function. Obviously, the feasibility threshold defines a kind of "hyperplane" in the search space. The particular shape and size of the subspace affects the genetic search for $D$ fixed. This could be the reason for the different results obtained for the same problem with different values of $t$. For example, in Figure 8 the GA outperforms the SFFS method if $t = .80$. But, for $t = .85$ the GA results are quite far from the best obtained by the SFFS method.

## 6. CONCLUDING REMARKS

From the experimental results obtained it appears that the Floating methods yield very good performance even for high dimensional problems. Although the GA approach gives reasonable solutions, some of the problems that arise make them of limited applicability in general. However, the GA approach has a significant advantage in its ability to perform the search in the near-optimal region of the space by virtue of the inherent randomisation mechanism employed in searching. This fact can be exploited to hybridise the GA approach with sequential methods in order to benefit from the best features of both approaches. Some tentative experiments using SFS and SFFS methods to partially initialise the population have been carried out. The results confirm a well-known effect in GA theory. When a "too good" solution is found, the genetic search is disabled because of the so-called premature convergence. As a consequence of this the results obtained were worse than the presented ones. A number of different approaches to maintain the diversity of the search in the GA have already been proposed [4,6]. Apart from this, another approach in which the floating search were embedded into the genetic operators appears to be a more attractive way of hybridising the GA.

## REFERENCES

1. N. Choakjarernwanit. *Feature Selection in statistical pattern recognition.* PhD thesis, University of Surrey, England, 1992.
2. P. A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice-Hall, 1982.
3. F. J. Ferri, V. Kadirkamanathan, and J. Kittler. Feature Subset Search using Genetic Algorithms. In *IEE/IEEE Workshop on Natural Algorithms in Signal Processing*, Essex, 1993.
4. D. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison-Wesley, Reading, MA, 1989.
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. M. L. Mauldin. Maintaining Diversity in Genetic Search. In *Proc. of the Nat. Conf. on Artificial Intelligence*, pages 247–250, Brighton, UK, 1984.
7. P. Pudil, J. Novovičová, and S. Bláha. Statistical Approach to Pattern Recognition: Theory and Practical Solution by Means of PREDITAS System. *Kybernetika*, 27:Supplement, 1–78, 1991.
8. P. Pudil, J. Novovičová, and J. Kittler. Floating Search Methods in Feature Selection. *Pattern Recognition Letters*, (submitted).
9. W. Siedlecki and J. Sklansky. A Note on Genetic Algorithm for Large-scale Feature Selection. *Pattern Recognition Letters*, 10(5):335–347, November 1989.
10. S. D. Stearns. On Selecting Features for Pattern Classifiers. In *Third Int. Conf. on Pattern recognition*, pages 71–75, Coronado, CA, 1976.