J. Indian Inst. Sci., Jan.–Feb. 1997, 77, 85–93.

Feature selection to improve classification accuracy using a genetic algorithm

M. PRAKASH AND M. NARASIMHA MURTY
Department of Computer Science and Automation, Indian Institute of Science, Bangalore - 560 012, India
email: mnm@csa.iisc.ernet.in

Received on August 9, 1995; Revised on July 15, 1996.

Abstract

In this paper, we have investigated the unconstrained optimization version of the feature selection problem. We have searched the space of subsets of features with a genetic algorithm. The nearest-neighbour classifier accuracy is used as the search criterion. Our results on several data sets indicate that considerable improvement in classification accuracy can be obtained. Also, the genetic algorithm is both fast and robust on this problem to yield good solutions. Our limited experimental results to verify bias contradict, for some data sets, the guideline in pattern recognition of having the ratio of sample size to dimensionality of atleast five.

Keywords: Feature selection, genetic algorithm, finite sample size, nearest neighbor classifier.

1. Introduction

In this paper we propose the use of genetic algorithm (GA) for selecting a subset of features from an initially large set of features to improve the classification accuracy. By eliminating irrelevant or redundant features we hope to decrease the error rate by exploiting the peaking phenomenon arising out of the curse of dimensionality[1,2]. There are two versions to the problem of feature subset selection in the design of pattern classifiers, each version addressing a specific objective and leading to a distinct type of optimization. In one version, the objective is to find the smallest subset of features for which the error rate (or perhaps some other measure of performance) is below a given threshold. This version leads to a constrained combinatorial optimization problem in which the error rate serves as a constraint and the number of features as the primary search criterion. In the second version of the problem, the objective is to find a subset of features that yields the lowest error rate of the classifier. This leads to an unconstrained optimization problem in which the error rate is the search criterion. The latter approach is adopted in this paper.

Feature selection has been studied extensively earlier, but most of the studies are limited. They either made a restricted assumption that the criterion function be monotonic, and/or assumed that the underlying structure of data is known, and/or used search techniques that are not good at considering the combinational aspects of features, especially in large dimensional problems.

The *monotonicity property* of the performance criterion states that the performance obtained by a set of features will never be worse than the performance obtained by any subset of it. Since an exhaustive search is not possible even on a problem of reasonable dimensionality, many search techniques use this property to contain the search. Discriminant functions and distance measures such as the Bhattacharyya distance and divergence are examples of criterion functions satisfying the monotonicity property, and hence this property was treated by most of the researches as nonrestrictive. While this assumption is true for the Baye's rule, any practical discrimination method working on a limited number of samples may find it to be restrictive[1,3]. It is in these problems that the unconstrained optimization version seeks a subset of features to improve the performance. Hence, it implicitly rejects the monotonicity property.

The search techniques used for feature subset selection problems can be broadly classified as methods based on: sequential selection[2,4], dynamic programming[5,6], branch-and-bound[7,8] and others[9]. Sequential selection techniques include forward selection, backward elimination and their generalized version (p, q) search. In the Sequential forward selection technique, to begin with, a feature which maximizes a certain performance criterion is chosen. Then, among the rest of the features, another one which maximizes the performance criterion is chosen next. This process is repeated until enough number of features have been chosen that satisfy the constraints imposed. Similarly, backward elimination starts with all features which are eliminated one by one. Clearly, these approaches are greedy, and are found not to be good at considering the combinational aspects of the features[10,2,11]. This problem has only been alleviated to some extent by the generalized (p, q) forward (backward) search which chooses (eliminates) the best (worst) p features at every step, and rejects (adds) q from the features chosen (eliminated) by then.

Dynamic programming approach was used to resolve this problem in that, on the one hand, an exhaustive search can be avoided by applying the Bellman's principle of optimality which states that an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. On the other hand, an application of the foregoing principle together with appropriately formulated recursive functional equations ensures that the final best subset chosen may not necessarily include all of the best single features selected in the previous stages. In addition to this improvement based on the monotonicity property, further improvement in search was obtained by the branch-and-bound technique which explores only the feasible regions, pruning the unfeasible regions based on constraints. However, even searching the feasible regions becomes impractical for feature dimensionality exceeding a moderate value like 20. Recently, genetic algorithm, a population-based search technique, was used to obtain about a two-order improvement over the traditional techniques making it practical to use on large-dimensional problems[3]. However, this approach was for the constrained version.

In this paper, we have investigated the unconstrained optimization version of the problem, which implicitly rejects the monotonicity property, under no assumption of the underlying structure, and with an emphasis on large dimensional problems. This necessitated a good choice for the search technique to be used along with the criterion function. We have searched the space of subsets of features with a genetic algorithm, a search technique that worked best on the constrained optimization version of the problem. The nearest-neighbor classifier (NNC) accuracy is used as the search criterion. A work somewhat akin to ours was done by Kudo and Shimbo[12] in which the error rate was used only as a secondary criterion while the smallest subset was used as the primary search criterion. Also, they used a sequential algorithm.

The organization of the paper is as follows. In the next section we describe the nearest neighbor classifier briefly. In Section 3 we describe the genetic approach to feature subset selection problem. Section 4 presents the experiments conducted, results obtained, and a discussion of them. Finally, in the last section, we summarize and conclude.

2. Nearest Neighbor Classifier

When an unlabelled pattern in an m-dimensional feature space X and represented by the m-dimensional feature vector $Y = [y_1, y_2, y_m]$ is to be classified, the nearest neighbor of Y is found among all the training samples of the c classes,T. Y is assigned to the same class as this nearest neighbor. The distance between Y and a training sample is measured using the squared Eucledian distance.

Let $t_k^i = \left[t_{k_1}^i, t_{k_2}^i, \ldots, t_{k_m}^i\right]$ and N_{ti} denote the kth m-dimensional training feature vector of the ith] class and the number of available training samples of class i, respectively. Similarly, let $s_k^i = \left[s_{k_1}^i, s_{k_2}^i, \ldots, s_{k_m}^i\right]$ and N_{si} denote the kth m-dimensional test feature vector of the ith class and the number of available test samples of class i respectively. Let N_s and N_t be the total number of test and training samples, respectively. Then, the classifier accuracy obtained on all test samples, S, using the nearest neighbor classifier(NNC), NNCA, is given by

$$NNCA(X,T,S) = \sum_{i=1}^{c} \sum_{j=1}^{N_{si}} f\left(s_j^i\right) * 100 / N_s \tag{1}$$

$$f\left(s_j^i\right) = \begin{cases} 1 & \text{if } i = l* \\ 0 & \text{otherwise} \end{cases}$$

$$d\left(s_j^i, t_k^{l*}\right) = \mathrm{Min}_{l,k} d\left(s_j^i, t_k^l\right), l = 1,2,\ldots,c, k = 1,2,\ldots,N_{t_i}.$$

$$d\left(s_j^i, t_k^l\right) = \sum_{u=1}^{m} \left(s_{j_u}^i - t_{k_u}^l\right)^2, m = |X|$$

3. GA approach to search for an optimal subset of features

Genetic algorithms belong to a class of stochastic algorithms, based on the mechanisms of natural selection and natural genetics[13] possessing an inherent capability to perform parallel search in complex search spaces. GAs are shown to be competent in obtaining optimal or near-optimal solutions to many optimization problems arising in diverse areas including pattern recognition. For a general introduction to genetic algorithms, the reader is referred to Goldberg[13].

The problem is formulated as determining an optimal subset of a set of n given features resulting in the best discriminating capability. Hence, it is an optimization problem in the space of all subsets of features whose criterion function is the classification accuracy. Given a particular set of features, a training set and a test set, the NNC computes the classifier accuracy according to eqn 1. Here, we pose the selection of optimal subset of features, Z, as an optimization problem. T and S are the training and the test data sets, respectively. Given a subset of features, X, T_X and S_X represent the training and the test data sets, and are obtained by projecting T and S onto the subspace spanned by the features of X.

$$\text{maximize } NNCA(X, T_X, S_X) \text{ as given by eqn 1}$$
$$\forall X \in \mathcal{P}(Z)$$

The GA requires a solution to be represented in the form of a string or a chromosome. So, we require a coding scheme for transforming the subset of features chosen for the NNC into a string form, and *vice versa*. Each chromosome is represented by a binary string of length n bits. The ith bit corresponds to the ith feature. If the bit has a value of 1, then the corresponding feature is included in the subset. Otherwise, it is excluded from it. Note that this simple encoding is a bijection.

The GA maintains a population of chromosomes which represent different NNC configurations, and the search is carried out using genetic operators. For each solution string, S_1, in the population, we need to compute its fitness value. We take the NNCA as given by eqn 1 as its fitness value. Since the encoding is a bijection, we know what features are chosen for a given chromosome, and hence can compute the classifier accuracy.

Now, we describe the genetic operators used.

Fitness function: Given a chromosome q, the fitness function, f, returns its fitness value as: $f(q) = NNCA(X, T_X, S_X)$, where X is a subset of features having a 1 in their corresponding bit positions in q. The fitness function used in constrained optimization approach by Siedlecki and Sklansky[3] is much more complicated.

Selection operator: We have used a stochastic remainder selection strategy[13]. In this strategy, the probabilities of selection are calculated as $pselect_i = f_i \Big/ \sum_{j=1}^{N} f_j$ where f_i is the fitness of the chromosome i. Then the expected number of individuals for each string e_i is calculated as $e_i = pselect_i * N$. Each string is allocated with a number of samples according to the integer part of the e_i values. The remainder strings are filled in as follows: In stochastic remainder selection with replacement, the fractional parts of the expected number values are used to calculate the weights in a roulette wheel selection procedure. This selection procedure is then used to fill the remaining population slots. In stochastic remainder selection without replacement, the fractional parts of the expected number values are treated as probabilities. One by one, weighted coin tosses (Bernoulli trials) are performed using the fractional parts as success probabilities. For example, a string with an expected number of copies equal to 1.5 would receive a single copy surely, and another copy with a probability of 0.5. This process is continued until the population is full. We have used the stochastic remainder selection with replacement. We have also adopted the elite strategy where the current best chromosome is always carried to the next generation.

Crossover and mutation operators: We have chosen the operators discussed in the previous sectior

4. Experiments and Results

In this section, we describe the data sets used, the experiments conducted, and the results obtained. Finally, we discuss the results.

4.1. *The utilized data sets*

We have chosen eight data sets in this study: fiveclass[14], iris[15], fossil[15], 8OX[16], vowel, waveform, soybean and sonar[17]. Of these, the first four are small dimensional, but have been included for illustrative purposes. Although the vowel data set is of dimensionality 10, it has been chosen because of its complexity. The last three data sets can be considered as large dimensional. The details of dimensionality, number of training, test and total samples, and the ratio of number of design samples to dimensionality are given in Table I. For some data sets there is no explicit separation between training and test sets. Leave-one-out method is used on them.

The first four data sets have been used extensively in pattern recognition literature, and are available in the cited references. The vowel, waveform and sonar data sets have been used by neural network community, and are available on CMU NN benchmark collection. The soybean data set is available on the UCI machine learning benchmark collection.

4.2. *Experiments*

We have conducted an exhaustive search on the first five low-dimensional data sets. Genetic algorithm is used to search for an optimal solution for the large-dimensional prob-

Table I
Data sets used

Sl. No.	*Data set*	*Dimensionality (m)*	*Number of classes (c)*	*Number of samples*			*s/c*m*
				Training	*Test*	*Total(s)*	
1	Five class	3	5	–	–	35	2.3
2	Iris	4	3	–	–	150	12.5
3	Fossil	6	3	–	–	87	4.8
4	8OX	8	3	–	–	45	1.9
5	Vowel	10	11	528	462	990	9.0
6	Waveform	21	3	300	1000	1300	20.6
7	Soybean	35	9	200	242	442	1.4
8	Sonar	60	2	104	104	208	1.7

Table II
Results obtained on best subsets chosen

Sl. No.	*Data set*	*Search technique*	*No. of features*		*NNC accuracy (%)*	
			all	*subset*	*all*	*subset*
1	Five class	Exhaustive	3	3	100.0	100.0
2	Iris	Exhaustive	4	4	96.0	96.0
3	Fossil	Exhaustive	6	5	98.8	100.0
4	8OX	Exhaustive	8	7	93.3	100.0
5	Vowel	Exhaustive	10	7	56.1	60.1
6	Waveform	Genetic	21	12	77.0	80.0
7	Soybean	Genetic	35	22	86.4	95.0
8	Sonar	Genetic	60	33	91.3	100.0

lems. For the first four data sets, NNC accuracy is obtained using the leave-one-out technique. For the last four data sets the accuracy is obtained using separate training and test sets. The best results obtained are shown in Table II.

In our GA, we have varied the crossover probability from 0.95 to 0.45 in steps of 0.1. We have used a fixed mutation probability of 0.05 for all data sets. A population size of 50, 100, and 100 were used for the waveform, soybean and the sonar data sets, respectively. Since GA is a stochastic algorithm, we repeated the above experiment five times for every combination, and the mean results are obtained. The mean results corresponding to a crossover probability of 0.95 are plotted in Fig. 1. Since the other parameter settings also led to a similar convergence, they are not shown. There were multiple subsets of features resulting in the same classification accuracy for some data sets. We have chosen one of them arbitrarily, and reported the results for that choice in Tables II and III.

The optimization of error rate may introduce an unwanted bias in the sense that the selected features may strongly suit the data at hand, and not necessarily the underlying distributions of the feature space. An accepted guideline in pattern recognition is that the ratio of the number of design samples per class to the dimensionality should be larger than 5 for the bias to be small[1]. Since some of the data sets do not satisfy this

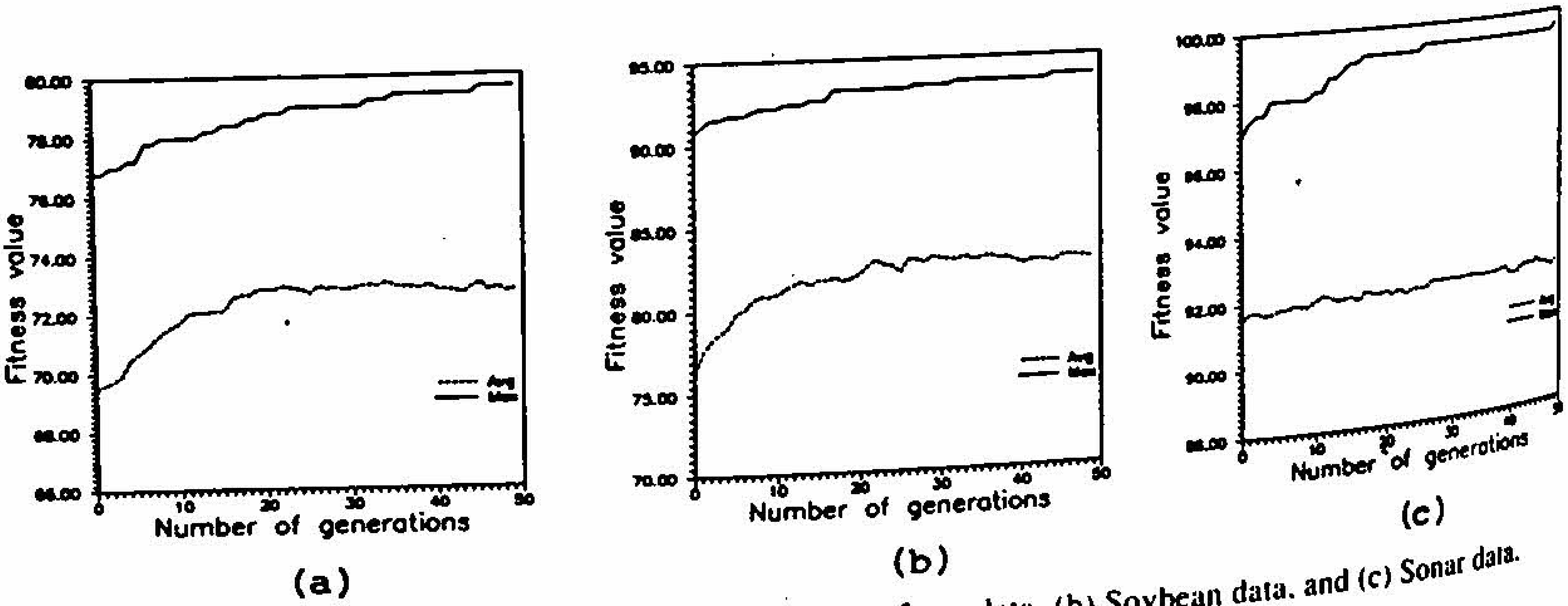

FIG. 1. Convergence of the genetic algorithm for (a) waveform data, (b) Soybean data, and (c) Sonar data.

Table III
Results obtained to verify bias

Data set	*Features*	*Classification accuracy (%)*				
		Leave-one-out			*Trg: Set1*	*Trg: Set2*
		Set 1	*Set 2*	*Total set*	*Test: Set2*	*Test: Set1*
Vowel	all	99.1	99.1	99.1	56.1	55.1
	subset	98.7	99.1	98.1	60.1	53.0
Waveform	all	75.6	75.6	76.4	77.0	73.0
	subset	76.4	75.6	78.2	80.0	77.0
Soybean	all	84.0	86.4	89.1	86.4	85.0
	subset	91.0	90.9	93.2	95.0	89.5
Sonar	all	63.5	89.4	82.7	91.3	77.9
	subset	74.0	92.3	88.9	100.0	75.0

guideline as seen from Table I, a bias may be introduced. For the last four data sets where separate training and test sets are used, we conducted further experiments to gain some insight into the bias. We obtained error rates using the leave-one-out technique on the training, test and total samples separately. Also, we obtained the NNC accuracy by reversing the roles of the training and the test sets. These experiments were done for both the complete and subset of features chosen by the genetic algorithm which used different training and test sets. Since optimization is carried out with respect to separate training and test sets, we feel that these experiments serve the purpose of testing unknown samples. If no bias is introduced, we would expect similar improvement in all the experiments. We could have partitioned the total available samples into three sets: a training and a test set to be used in optimization, and another test set to be used after optimization. However, we have not done this because when the sample size is small, every effort should be made to use all of them in the design set. The results of these experiments are shown in Table III. Here, Data sets 1 and 2 correspond to the original training and test sets used by the genetic algorithm.

4.3. *Discussion of results*

The results in Table II indicate clearly a trend that with increase in the number of features the difference also increases in classification accuracies among the features and the best subset of features. This can be expected because as the number of features increases so are the chances of the presence of redundant or irrelevant features also increase. However, the 8OX data sét appears to be an exception in the sense that although the dimensionality is only 8, the improvement in accuracy is significantly high. Since the ratio of sample size to dimensionality is only 1.9, bias might have been introduced. A further analysis of the results and data revealed that all the three global optima excluded the last feature along which all three classes overlap considerably. Thus, this feature can be considered as irrelevant. Since one of the optima excluded only this feature, we feel that the bias introduced is small. In addition to the last feature, the second feature from the first and the fourth feature from another optimal subsets were excluded. They can be treated as redundant.

The results obtained on the sonar data set are particularly interesting. This data set is highly undersampled considering the dimensionality of 60 and a sample size of 104 for training and test sets. The best result obtained, using a multilayer perceptron, was 90.4[17]. We however, have obtained 100% accuracy. Of course, MLP used only the training set as the design set, whereas our optimization of NNC accuracy uses both training and test sets as the design set. However, we feel that there was no unwanted bias introduced as indicated by the results in Table III. Also, the best found subset consisted features selected at almost regular intervals, *i.e.*, it selected two or three and rejected two or three features repeatedly across the entire spectrum. The features were obtained from sampling apertures offset temporally. Since this roughly corresponds to taking fewer number of wider apertures with greater offset, we feel that the bias introduced by NNC is small. Hence, the features can be considered as redundant.

The convergence plots shown in Fig. 1 clearly indicate that genetic algorithm is better than a random search. Although convergence has not taken place, the trend is clearly seen. Also, that such a trend has been shown in all cases of parameter values it indicates the robustness of the GA in the selection of a good subset of features.

The results shown in Table III, which include the results obtained by GA (column 6) and reported in Table II, indicate that the improvement from all features to subsets is similar across all columns which correspond to different experiments. This trend can be observed for all the data sets except the vowel data set. Hence, we feel that a bias has been introduced only in the vowel data set. We feel that there is no bias introduced in the case of 8OX, soybean and sonar data sets based on the results shown in Table III along with manual analysis done. According to the guideline in PR practice, bias must have been introduced in these and not in the vowel data set. Hence, we feel that further investigation is desirable to resolve this conflict.

5. Summary and Conclusion

We have investigated the effectiveness of selecting an optimal subset of features which maximizes the classification accuracy. We found that as the number of features increases, the improvement in classification accuracy increases too. Intuitively this is appealing because the chances of the presence of redundant or irrelevant features increases with increasing dimensionality. Irrelevant features may either correspond to features which have no relevance to the classification intent as in the case of some of the features in soybean data, or may correspond to features which are highly noisy as in the case of 8O X data. Redundant features may arise out of multivariate relationships among features, or out of negligible variation in their values across different classes (compared to variation in other features), or may even arise out of finer resolution as in the case of sonar data.

Genetic algorithms have been found to be effective in finding good solutions fast. Also, they appear to be robust for our problem since a variety of parameter values gave almost the same results on all data sets.

We have used the NNC directly to compute the classifier accuracy for optimization. As far as the possibility of a bias is concerned, our analysis contradicts, for some data sets, the guideline in PR practice of using a sample size at least five times the dimensionality. Finally, we conclude that a considerable improvement that can be achieved by our approach of feature selection makes a further investigation of bias worth before the results are either accepted or rejected.

Note: After submitting the paper, we came to know of a similar attempt made earlier. This work on the GAs was reported in the Neural Information Processing Systems - Vol. II}.

References

1. Jain, A. K. and Chandrasekaran, R. Dimensionality and sample size considerations in pattern recognition practice. In *Handbook of statistics*, (Krishnaiah, P. R. and Kanal L. N., eds), (Classification, pattern recognition and reduction of dimensionality), Vol. 2, 1982, North-Holland.

2. Hand, D. J. *Discrimination and classification*, 1981, Wiley.

3. Siedlecki, W. and Sklansky, J. A note on genetic algorithm for large-scale feature selection, *Pattern Recongition Lett.*, 1989, **10**, 335–347.

4. Raveendran, P. and Sigeru Omatu Performance of an optimal subset of zernike features for pattern classification. *Inf. Sci.*, 1993, **1**, 133–147.

5. Chang, C. Y. Dynamic programming as applied to feature subset selection in pattern recognition system, *IEEE Trans.*, 1973, **SMC-3**, 116–171.

6. Dattatreya, G. R. and Sarma, V. V. S. Bayesian and decision tree approaches for pattern recognition including feature measurement costs, *IEEE Trans.*, 1981, **PAMI-3**, 293–298.

7. Narendra, P. M. and Fukunaga, K. A branch and bound algorithm for feature subset selection, *IEEE Trans.*, 1977, **C-26**, 917–922.

8. Yu, B. and Yuan, B. A more efficient branch and bound algorithm for feature selection, *Pattern recognition*, 1993, **26**, 883–889.

9. Mucciadri, A. N. and Gose, E. E. A comparison of seven techniques for choosing subsets of pattern recognition properties, *IEEE Trans.*, 1971, **C-20**, 1023–1013.

10. Cover, T. M. The best two independent measurements are not the two best, *IEEE Trans.*, 1974, **SMC**, 116–117.

11. Van Campenhout, J. M. Topics in measurement selection. In *Handbook of statistics*, Vol. 2 (Krishnaiah, P. R. and Kanal L. N., eds) (Classification, pattern recognition and reduction of dimensionality), 1982, North-Holland.

12. Kudo, M. and Shimbo, M. Feature selection based on the structural indices of categories, *Pattern Recognition*, 1993, **26**, 891–901.

13. Goldberg, D. *Genetic algorithms in search, optimization and machine learning*, 1989, Addison-Wesley.

14. Sudhanva, D. and Chidananda Gowda, K. Dimensionality reduction using geometric projections: a new technique. *Pattern Recognition*, 1992, **25**, 809–817.

15. Chien, Y. *Interactive pattern recognition*, 1978, Marcel Dekker.

16. Jain, A. K. and Dubes, R. *Algorithms for clustering data*, 1988, Prentice Hall.

17. Gorman, R. P. and Sejnowski, J. Learned classification of sonar targets using a massively parallel network, *IEEE Trans.*, 1988, **ASSP-36**, 1135–1140.