Feature Selection Using Multi-Objective Genetic Algorithms for Handwritten Digit Recognition

† L. S. Oliveira[1,2], R. Sabourin[1−3], F. Bortolozzi[3], and C. Y. Suen[2]
[1]Ecole de Technologie Supérieure - Montreal, Canada
[2]Centre for Pattern Recognition and Machine Intelligence - Montreal, Canada
[3]Pontifícia Universidade Católica do Paraná - Curitiba, Brazil

† soares@livia.etsmtl.ca

Abstract

This paper discusses the use of genetic algorithm for feature selection for handwriting recognition. Its novelty lies in the use of a multi-objective genetic algorithms where sensitivity analysis and neural network are employed to allow the use of a representative database to evaluate fitness and the use of a validation database to identify the subsets of selected features that provide a good generalization. Comprehensive experiments on the NIST database confirm the effectiveness of the proposed strategy.

1 Introduction

The feature selection problem in automated design of pattern classifiers refers to the task of identifying and selecting an effective subset of features to be used to represent patterns from a larger set of often mutually redundant or even irrelevant features. Therefore, the main goal of feature selection is to reduce the number of features used in classification while maintaining an acceptable classification accuracy. We can classify feature selection algorithms into two categories based on whether or not feature selection is performed independently of the learning algorithm used to construct the classifier. If feature selection is done independently of the learning algorithm, the technique is said to follow a filter approach. Otherwise, it is said to follow a wrapper approach [4].

In the context of practical applications such as handwriting recognition, feature selection presents a multi-criterion optimization function, e.g. number of features and accuracy of classification. Genetic algorithms offer a particularly attractive approach to solve this kind of problems since they are generally quite effective in rapid global search of large, non-linear, and poorly understood spaces.

In this paper we discuss the use of the multi-objective genetic algorithms as a means to search for subsets of features [1], which contain discriminatory information to perform the classification of handwritten digits. The strategy proposed takes into account an efficient multi-objective genetic algorithm [7] to generate a set of alternative solutions and the use of a cross-validation method to indicate the best accuracy/complexity trade-off. We demonstrate that the cross-validation is very important when working with a set of alternative solutions and it can not be neglected as in [1]. The classification accuracy is supplied by multi-layer perceptron neural networks in conjunction with the sensitivity analysis [5]. Such an approach makes it feasible to deal with huge databases in order to better represent the pattern recognition problem during the fitness evaluation. In order to show the robustness of the proposed strategy, we carried out comprehensive experiments on the NIST database.

2 Multi-Objective Optimization using Genetic Algorithms

A general multi-objective optimization problem consists of a number of objectives and is associated with a number of inequality and equality constraints. Solutions to a multi-objective optimization problem can be expressed mathematically in terms of nondominated points, i.e., a solution is dominant over another only if it has superior performance in all criteria. A solution is said to be Pareto-optimal if it cannot be dominated by any other solution available in the search space.

A common difficulty with multi-objective optimization problem is the conflict between the objectives. In general, none of the feasible solutions allow simultaneous optimal solutions for all objectives. Thus, mathematically the most favorable Pareto-optimum is the solution that offers the least objective conflict. In order to find such solutions, classical methods scalarize the objective vector into one objective.

The simplest of all classical techniques is the weighted

1051-4651/02 $17.00 (c) 2002 IEEE

sum method. It aggregates the objectives into a single and parameterized objective through a linear combination of the objectives. However, setting up an appropriate weight vector also depends on the scaling of each objective function. It is likely that different objectives take different orders of magnitude. When such objectives are weighted to form a composite objective function, it would be better to scale them appropriately so that each has more or less the same order or magnitude. Moreover, the solution obtained through this strategy largely depends on the underlying weight vector.

2.1 Pareto-based Approach

In order to overcome such difficulties, Pareto-based evolutionary optimization has become an alternative to classical techniques such as weighted sum method. This approach was first proposed by Goldberg in [2] and it explicitly uses Pareto dominance in order to determine the reproduction probability of each individual. Basically, it consists of assigning rank 1 to the nondominated individuals and removing them from contention, then finding a new set of nondominated individuals, ranked 2, and so forth.

Pareto-based ranking correctly assigns all nondominated individuals the same fitness, however, this does not guarantee that the Pareto set be uniformly sampled. In order to avoid such a problem, Goldberg and Richardson in [3] propose the additional use of fitness sharing. The main idea behind this is that individuals in a particular niche have to share the available resources. The more individuals are located in the neighborhood of a certain individual, the more its fitness value is degraded.

In this work, we have used the Nondominated Sorting Genetic Algorithm NSGA (with elitism) proposed by Srinivas and Deb in [7]. The idea behind NSGA is that a ranking selection method is used to emphasize good points and a niche method is used to maintain stable subpopulations of good points. It varies from simple genetic algorithm only in the way the selection operator works. The crossover and mutation remain as usual. Before the selection is performed, the population is ranked on the basis of an individual's nondomination. The nondominated individuals present in the population are first identified from the current population. Then, all these individuals are assumed to constitute the first nondominated front in the population and assigned a large dummy fitness value. The same fitness value is assigned to give an equal reproductive potential to all these nondominated individuals.

In order to maintain the diversity in the population, these classified individuals are then shared with their dummy fitness values. Sharing is achieved by performing selection operation using degraded fitness values obtained by dividing the original fitness value of an individual by a quantity proportional to the number of individuals around it. Thereafter, the population is reproduced according to the dummy fitness values. Since individuals in the first front have the maximum fitness value, they get more copies than the rest of the population. The efficiency of NSGA lies in the way multiple objectives are reduced to a dummy fitness function using nondominated sorting procedures. More details about NSGA can be found in [7].

3 Handwritten Digit Classifier

In order to evaluate the effect of the proposed feature selection scheme, we have used a handwritten digit classifier. Such a classifier is a neural network (Multi-layer Perceptron) trained with the backpropagation algorithm. The database considered in our work is the NIST SD19.

The the training (TRDB) and validation (VDB1) sets were composed of 195,000 and 28,000 samples from the hsf_{0,1,2,3} series respectively while the test (TSDB) set was composed of 30,089 samples from the hsf_7 series. The recognition rates (zero-rejection level) achieved by this classifier were 99.66%, 99.65%, and 99.13% on TRDB, VDB1, and TSDB sets respectively. It is feed with a feature vector composed of 132 components based on concavity measures and contour information. More details about this classifier and its architecture can be found in [6].

4 Methodology

In our experiments, NSGA is based on bit representation, one-point crossover, bit-flip mutation, and roulette wheel selection (with elitism). The following parameter settings were employed: population size = 128, number of generations = 1000, probability of crossover = 0.8, probability of mutation = 0.007, and niche distance (σ_{share}) = 0.5.

In order to define the probabilities of crossover and mutation, we have used the one-max problem, which is probably the most frequently-used test function in research on genetic algorithms because of its simplicity. This function measures the fitness of an individual as the number of bits set to one on the chromosome. The parameter σ_{share} can be calculated as follows [7]:

$$\sigma_{share} \approx \frac{0.5}{\sqrt[p]{q}} \tag{1}$$

where q is the desired number of distinct Pareto-optimal solutions and p is the number of decision variables. Although the calculation of σ_{share} depends on this parameter q, it has been shown [7] that the use of the above equation with $q \approx 10$ works in many test problems.

As discussed previously, our practical problem consists of optimizing two objectives: minimization of the number of features and minimization of the error rate of the classifier on VDB1. Computing the first one is simple, i.e., the

1051-4651/02 $17.00 (c) 2002 IEEE

number of selected features (bit = 1). The problem lies in computing the second one, i.e., the error rate supplied by the classifier. Regarding a wrapper approach, in each generation, evaluation of a chromosome (a feature subset) requires training the corresponding neural network and computing its accuracy. This evaluation has to be performed for each of the chromosomes in the population. Since such a strategy is not feasible due to the limits imposed by the learning time of the huge training set considered in this work, we have adopted the strategy proposed by Moody and Utans in [5], who use the sensitivity of the network to estimate the relationship between the input features and the network performance.

The sensitivity of the network model to variable β is defined as:

$$S_\beta = \frac{1}{N}\sum_{j=1}^{N} ASE(\bar{x}_\beta) - ASE(x_\beta) \tag{2}$$

with

$$\bar{x}_\beta = \frac{1}{N}\sum_{j=1}^{N} x_{\beta_j} \tag{3}$$

where x_{β_j} is the βth input variable of the jth exemplar. S_β measures the effect on the training ASE (average square error) of replacing the βth input x_β by its average $\bar{x}_\beta$. Moody and Utans show that when variables with small sensitivity values with respect to the network outputs are removed, they do not influence the final classification. So, in order to evaluate a given feature subset we replace the unselected features by their averages. In this way, we avoid training the neural network and hence turn the wrapper approach feasible for our problem. We call this strategy modified-wrapper. Such a kind of scheme has been employed also by Emmanouilidis et al in [1].

The last step of our strategy consists of choosing the best solution from the Pareto-optimal front. After several experiments, we realized that the Pareto-optimal front by itself does not provide enough information to select the best solution. Often, the best solution found in the Pareto-optimal front does not have good generalization power on a different database. In order to overcome this kind of problem, we propose the use of a second validation database, which is not used during the optimization, to verify the generalization power of the Pareto-optimal front. The second validation set (VDB2) is composed of 30,000 samples from hsf_7 series.

5 Experiments and Discussion

All experiments we have carried out in this work were based in a single-population master-slave genetic algorithm. In this strategy, one master node executes the genetic operators (selection, crossover and mutation), and the evaluation of fitness is distributed among several slave processors. In order to execute our experiments, we have used a cluster with 17 (one master and 16 slaves) PCs (1.1Ghz CPU, 512Mb RAM). In order to test the effectiveness of the proposed methodology we have applied it to optimize the classifier described in section 3.

We have used both the weighted-sum approach and NSGA to generate the potential solutions. The results achieved by the former presented a premature convergence to a specific region of the search space instead of maintaining a diverse population. Hence, after several trials we did not succeed in finding the Pareto-optimal front but rather than an approximation of the Pareto-optimal solutions. This kind of behavior can be explained by the sensitivity towards weight presented by the weighted-sum approach. Since we have chosen weights to favor solutions with a small error rate rather than a small number of features, the selection pressure drove the search to the region where the error rates are smaller.

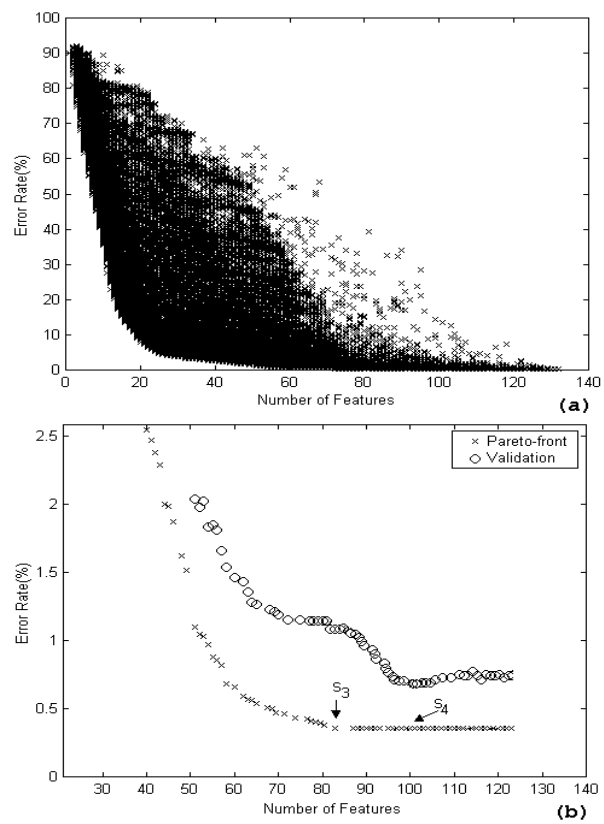

Figure 1. Pareto-based approach (a) Evolution of the population in the objectives plane, (b) Pareto-optimal front found by the NSGA and its correspondent validation curve.

As we have discussed in section 2.1 the Pareto-based approach was designed to overcome this kind of problem.

Since NSGA uses a niching technique to preserve the diversity in the population, this algorithm is able to deal with the problem of converging prematurely to a specific region of the search space. Therefore, it can guide the search towards the Pareto-optimal set. Figure 1a depicts the evolution of the population in the objectives plane from the first generation to the last one. This plot demonstrates the efficacy of NSGA in converging close to the Pareto-optimal front with a wide variety of solutions.

As discussed in section 4, after finding the Pareto-optimal front the next step is to choose a solution. In order to perform this task we have used VDB2. Figures 1b shows the Pareto-optimal front as well as its correspondent validation curve, which depicts the performance of all Pareto-optimal front on VDB2. After analyzing the validation curve plotted in Figure 1b, we selected a solution with 100 features (solution S_4 in Figure 1b) and error rate on the new validation set smaller than 1% to retrain the general-purpose recognizer.

Thereafter, we trained a new classifier using such a solution using the same databases presented in section 3. The recognition rates achieved by this new classifier were 99.66%, 99.65%, 99.16% on TRDB, VDB1, and TSDB sets respectively. As we can verify, the optimized classifier produced an error rate slightly lower than the original classifier but with about 25% less features (see Table 1). This confirms the efficiency of the proposed methodology in selecting a powerful subset of features. In order to show the importance of the second validation set to select a good solution, we retrained the best trade-off of the Pareto-front without regarding the validation curve (Solution S_3 in Figure 1b). The recognition rate reached by this solution was 96.8% on TSDB.

Table 1. Comparision between the original and optimized classifiers.

Original Classifier [6]		Optimized Classifier	
Features	RR (%)	Features	RR (%)
132	99.13	100	99.16

In spite of the fact that the Pareto-based approach presents several advantages when compared to the classical one, we have seen through the experiments that both strategies found similar solutions. In our first experiment, we observed that the classical approach converged the search to the space where are located the most probable solutions due to the weights we have chosen. However, for problems where the solutions are located along of the Pareto-front, the classical approach does not work properly. Moreover, to achieve part of the Pareto-front, the weighted-sum method was run several times with different weight vectors.

For the problem of feature selection for handwriting recognition we can observe that the main advantage of the Pareto-based approach is the ability of dealing with different databases with no need of dealing with problems such as scaling and finding the suitable values for the weight vector. Moreover, Pareto-based approaches have the ability of finding the Pareto-optimal front in the first run of the algorithm.

6 Conclusion

In this study we have proposed a methodology for feature selection which uses a Pareto-based approach to generate the Pareto-optimal front where sensitivity analysis and neural network enable the use of a representative database to evaluate fitness. Afterwards, the entire Pareto-front is validated on a different database in order to provide the solutions that present better generalization on different databases. Finally, the selected solution is trained and applied to the recognition system.

The use of a Pareto-based approach instead of a classical one is supported by the theory as well as the experiments carried out. We also have shown the importance of using a second validation set in order to avoid selecting subsets of features with poor generalization ability. We have demonstrated that the proposed methodology succeeded in reducing the complexity of the feature set used by the classifier and also that such a classifier even using less features achieved recognition rates at the same level than reached by the original classifier.

References

[1] C. Emmanouilidis, A. Hunter, and J. MacIntyre. A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. In *Proc. of Congress on Evolutionary Computation*, volume 1, pages 309–316, 2000.

[2] D. Goldberg. *Genetic Algorithms in search, optimization and machine learning*. Reading, Mass., Addison-Wesley, 1989.

[3] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multi-modal function optimisation. In *Proc. of* 2nd *International Conference on Genetic Algorithms and Their Applications*, pages 41–49, 1987.

[4] G. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problems. In *Proc. of* 11th *International Conference on Machine Learning*, pages 121–129, 1994.

[5] J. Moody and J. Utans. Principled architecture selection for neural networks: Application to corporate bond rating prediction. In J.Moody, S.J.Hanson, and R.P.Lippmann, editors, *Advances in Neural Information Processing Systems 4*. Morgan Kaufmann, 1991.

[6] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Automatic recognition of handwritten numerical strings: A recognition and verification strategy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, to appear.

[7] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1995.

1051-4651/02 $17.00 (c) 2002 IEEE