Feature Selection for Ensembles: A Hierarchical Multi-Objective Genetic Algorithm Approach

[†] 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@cenparmi.concordia.ca

Abstract

Feature selection for ensembles has shown to be an effective strategy for ensemble creation. In this paper we present an ensemble feature selection approach based on a hierarchical multi-objective genetic algorithm. The first level performs feature selection in order to generate a set of good classifiers while the second one combines them to provide a set of powerful ensembles. The proposed method is evaluated in the context of handwritten digit recognition, using three different feature sets and neural networks (MLP) as classifiers. Experiments conducted on NIST SD19 demonstrated the effectiveness of the proposed strategy.

1 Introduction

Ensemble of classifiers has been widely used to reduce model uncertainty and improve generalization performance. Developing techniques for generating candidate ensemble members is a very important direction of ensemble of classifiers research. Both theoretical [6] and empirical [11] research has demonstrated that a good ensemble is one where the individual classifiers in the ensemble are both accurate and make their errors on different parts of the input space (there is no gain in combining identical classifiers). In other words, an ideal ensemble consists of good classifiers (not necessarily excellent) that disagree as much as possible.

The most popular methods for ensembles creation are Bagging and Boosting. The effectiveness of such methods comes primarily from the diversity caused by re-sampling the training set while using the complete set of features to train the component classifiers. In addition, some attempts have been made to incorporate the diversity into ensemble creation methods. The Random Subspace Method (RMS) proposed by Ho in [4] was one early algorithm that construct an ensemble by varying the subset of features. In this case, the diversity is promoted through different subsets of features.

In the same vein, methods based on feature selection have been proposed for ensembles. The key idea is to promote diversity among the classifiers by performing feature selection. Gerra-Salcedo and Withley [2] used a simple genetic algorithm (GA) to explore the space of all possible feature subsets, and then create an ensemble based on them. In their experiments, this GA-based approach outperformed classical methods such as Bagging and Boosting. In spite of the fact they achieved interesting results, they did not consider any measure of diversity. A more elaborate method, also based on GA, was proposed by Optiz [11]. In his work, he stresses the importance of a diversity measure by including one into the fitness calculation. The drawback of this method is that the objective functions are combined through the weighted sum. It is well known that when dealing with this kind of combination, one should deal with problems such as scaling and sensitivity towards the weights. More recently Günter and Bunke [3] have applied different feature selection algorithms to create ensemble of classifiers for the field of handwriting recognition. They have used the problem of word recognition where the base classifier was an HMM.

It has been demonstrated that feature selection through multi-objective genetic algorithm (MOGA) is a very powerful tool to find a set (Pareto-optimal) of good classifiers [10]. Besides, it can overcome problems such as scaling and sensitivity towards the weights. In this light, we propose a new methodology for creating ensembles of classifiers which is able to cope with multiple ensembles simultaneously. Such a strategy is based on a hierarchical MOGA where the first level is devoted to generate a set of good classifiers while the second one combines these classifiers in order to find an ensemble. The use of MOGA in both levels is justified by the fact that in both cases we have to cope with multi-objective optimization problems. In order to show the robustness of the proposed methodology, we carried out comprehensive experiments in the context of handwritten digit recognition using the NIST SD19 database.

2 Multi-Objective Optimization using Genetic Algorithms

Since the concept of multi-objective optimization will be explored in the remaining of this paper, this section briefly introduces it. For more details see [12].

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.

3 Classifiers and Feature Sets

To evaluate the proposed methodology we have used three classifiers trained to recognize handwritten digits of NIST SD19. Such classifiers were trained with three well-known feature sets: Concavities and Contour (CC) [9], Distances (DDD) [8], and Edge Maps (EDM) [1]. All classifiers are neural networks (MLP) trained with the back-propagation algorithm. The training (TRDB) and validation (VDB1) sets are composed of 195,000 and 28,000 samples from hsf_0123 series respectively while the test set (TSDB) is composed of 30,089 samples from the hsf_7. Table 1 reports the performance of all classifiers, where "Rec.Rate" means the recognition rate at zero-rejection level and "Rec.Rate 0.5%" means the recognition rate achieved for an error rate fixed at 0.5%. The latter is much more meaningful when dealing with real applications since it describes the recognition rate in relation to a specific error rate, including implicitly a corresponding reject rate. This rate also allows us to compute the reliability of the system for a given error rate. It can be done by using Equation 1.

$$\text{Reliability} = \frac{\text{Rec.Rate}}{\text{Rec.Rate} \quad + \quad \text{Error Rate}} \times 100 \qquad (1)$$

Though all systems reach a reliability close to 99.5%, it is clear that there is enough room (see Table 1), especially for the last two feature sets, to improve "Rec.Rate 0.5%".

It should be noted, though, that the original feature set DDD proposed by Oh and Suen [8] contains 256 features. After carrying out some experiments with different strategies of zoning, we realized that using 96 features (6 zones: 3 horizontal and 2 vertical) we could achieve the same results using 256 features (16 symmetrical zones).

Table 1. Performance of the classifiers on TSDB.

Feature Set	No. of Features	Units H.Layer	Rec. Rate (%)	Rec.Rate 0.5% (%)
CC	132	80	99.13	98.50
DDD	96	60	98.17	92.80
EDM	125	70	97.04	85.10

4 Proposed Methodology

In this section we describe the hierarchical approach proposed. As stated before, it is based on a 2-level MOGA where the first level generates a set of good classifiers by conducting feature selection and the second one searches the best ensemble among such classifiers. In both cases, MOGAs are based on bit representation, one-point crossover, bit-flip mutation. In our experiments, MOGA used is the Non-dominated Sorting Genetic Algorithm (NSGA) with elitism proposed by Srinivas and Deb in [12].

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 GA 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 made to share 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. After sharing, these nondominated individuals are ignored temporarily to process the rest of population in the same way to identify individuals for the second nondominated front. These new set of points are then assigned a new dummy fitness value which is kept smaller than the minimum shared dummy fitness of the previous front. This process is continued until the entire population is classified into several fronts.

Thereafter, the population is reproduced according to the dummy fitness values. A stochastic remainder proportionate selection is used here. 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 [12].

4.1 Feature Selection

Feature selection is conducted through the strategy presented in [10]. It takes into account a MOGA where the classification accuracy is supplied by multi-layer perceptron (MLP) networks in conjunction with the sensitivity analysis. Such an approach makes it feasible to deal with huge databases in order to better represent the pattern recognition problem during the fitness evaluation. Moreover it can accommodate multiple criteria such as number of features and accuracy of the classifier, and generate the Pareto-optimal front in the first run of the algorithm.

Figure 1a depicts an example of the evolution of the population in the objective plane while Figure 1b shows the corresponding Pareto-optimal front. Once we did not train the models during the search (the training step is replaced by the sensitivity analysis), the last step of feature selection consists of training the solutions provided by the Pareto-optimal front.

It can be observed in Figure 1b that the Pareto-optimal front is composed of several different classifiers. In order to get a better insight about them, they were classified into 3 different groups: weak, medium, and strong. It can be observed that among all those classifiers there are very good ones. To find out which classifiers of the Pareto-optimal front compose the best ensemble, we carried out a second level of search.

4.2 Finding the Best Ensemble

Let $A = C_1, C_2, \dots, C_n$ be a set of n classifiers extracted from the Pareto-optimal (Figure 1b) and B a chromosome of size n of the population. The relationship between A and B is straightforward, i.e., the gene i of the chromosome B is represented by the classifier C_i from A. Thus, if a chromosome has all bits selected, all classifiers of A will be included in the ensemble.

In order to find the best ensemble of classifiers, i.e., the most diverse set of classifiers that brings a good generalization, we have used two objective functions during this level of the search, namely, maximization of the recognition rate of the ensemble and maximization of the ambiguity as proposed in [6]. We have tried other measures of diversity such

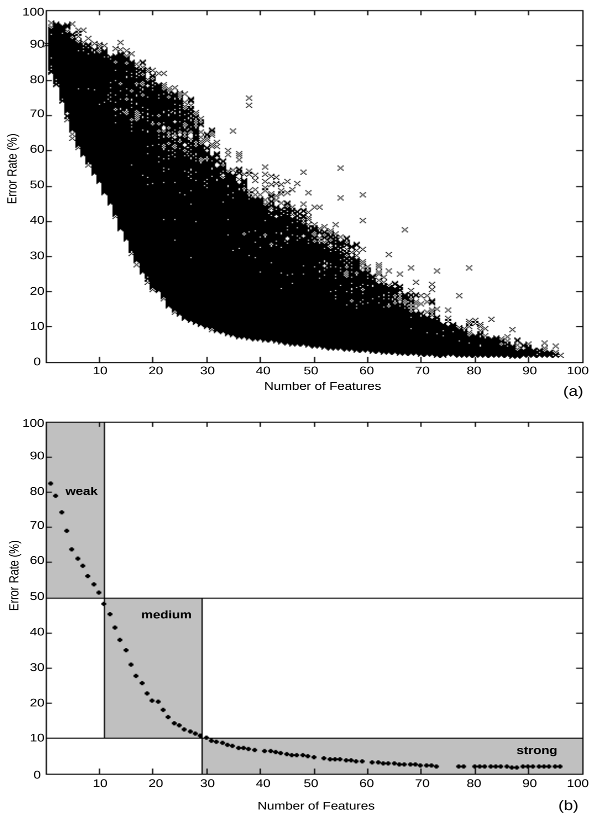

Figure 1. Feature selection using a Pareto-based approach (a) Evolution of the population in the objective plane, (b) Pareto-optimal front found by the NSGA

overlap and entropy [7], but the ambiguity yielded better results in our experiments.

The ambiguity is defined as follows:

$$a_i(x_k) = \frac{1}{N}[V_i(x_k) - \overline{V}(x_k)]^2 \tag{2}$$

where a_i is the ambiguity of the i^{th} classifier on the example x_k, randomly drawn from an unknown distribution, while V_i and $\overline{V}$ are the i^{th} classifier and the ensemble predictions respectively. In other words, it is simply the variance of ensemble around the mean, and it measures the disagreement among the networks on input x.

In this scenario the error from the ensemble is:

$$E = \overline{E} - \overline{A} \tag{3}$$

where $\overline{E}$ is the average of the single classifier errors and $\overline{A}$ is the ambiguity of the ensemble. Equation 3 expresses the trade-off between bias and variance in the ensemble, but in a different way than the common bias-variance relation in which the averages are over possible training sets instead of ensemble averages. If the ensemble is strongly biased the ambiguity will be small, because the networks implement very similar functions and thus agree in inputs even outside the training set.

At this level of the strategy we want to maximize the generalization of the ensemble, therefore, it will be necessary to use a way of combining the outputs of all classifiers to get a final decision. To do this, we have used the average, which is a simple and effective scheme of combining predictions of the neural networks [5]. Other combination rules such as product, min, and max have been tested but the simple average has produced better results. In order to evaluate the objective functions described above we have used a database composed of 30,000 samples extracted from hsf_7 (VDB2).

Different from other methodologies for ensemble creation based on feature selection where only one ensemble is considered, our approach considers w ensembles simultaneously, where w is the population size used by MOGA in the second level. This is due to the fact that each chromosome of the population represents a potential ensemble. Moreover, we will see in the experiments that this strategy produces more compact ensembles than other methods.

5 Experiments and Discussion

All experiments we have carried out in this work were based on a single-population master-slave MOGA. 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).

The following parameter settings were employed in both levels: population size = 128, number of generations = 1000, probability of crossover = 0.8, probability of mutation = $1/L$ (where L is the length of the chromosome), and niche distance = 0.45. The length of the chromosome in the first level is the number of components in the feature set (see Table 1), while in the second level is the number of classifiers picked from the Pareto-optimal front in the previous level. 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 niche distance was determined empirically.

Once all parameters have been defined, the first step, as described in Section 4.1, consists of performing feature selection for a given feature set. As depicted in Figure 1b, this procedure produces quite a large number of classifiers, which should be trained for use in the second level. After some experiments, we found out that the second level always chooses "strong" classifiers to compose the ensemble. Thus, in order to speed up the training process and the second level of search as well, we decide to train and use in the second level just the "strong" classifiers. This decision was made after we realize that in our experiments the "weak" and "medium" classifiers did not cooperate with the ensemble at all. To train such classifiers, the same databases reported in Section 3 were used. Table 2 summarizes the "strong" classifiers produced by the first level for the three feature sets we have considered.

Table 2. Summary of the classifiers produced by the first level.

Feature Set	No. of Classifiers	Range of Features	Range of Rec. Rates (%)
CC	81	24-125	90.5 - 99.1
DDD	54	30-84	90.6 - 98.1
EM	78	35-113	90.5 - 97.0

Considering for example the feature set CC, the first level of the algorithm provided 81 "strong" classifiers which have the number of features ranging from 24 to 125 and recognition rates ranging from 90.5% to 99.1% on TSDB. This shows the great diversity of the classifiers produced by the feature selection method. In order to assess the objective functions of the second-level MOGA (generalization of the ensemble and diversity) we have used VDB2, which was not used so far.

Like the first level, the second one also generates a set of possible solutions which are the trade-offs between the generalization of the ensemble and its diversity. Thus the problem now lies in choosing one ensemble among all. Figure 2 depicts the variety of ensembles yielded by the second-level MOGA for the feature set CC. The number over each point stands for the number of classifiers in the ensemble. Such information can be used to support the decision about which ensemble should be selected. Since we are aiming at performance, the direct choice will be the ensemble that provides better generalization. In our experiments, the ensemble that presents better performance has also the smallest number of classifiers. Table 3 reports the results of the ensembles for the three different feature sets on TSDB.

By comparing the results of Tables 3 and 1, it can be observed that the ensembles provided a compelling improvement in the recognition rates when the error rate is fixed at 0.5%, especially for those feature sets (EDM and DDD) where there is more room for improvement. We have noticed that the ensemble reduces the high outputs of some outliers so that the threshold used for rejection can be reduced and consequently “Rec.Rate 0.5%” is improved. This is an important issue in handwriting recognition where real applications have to work with very low error rates.

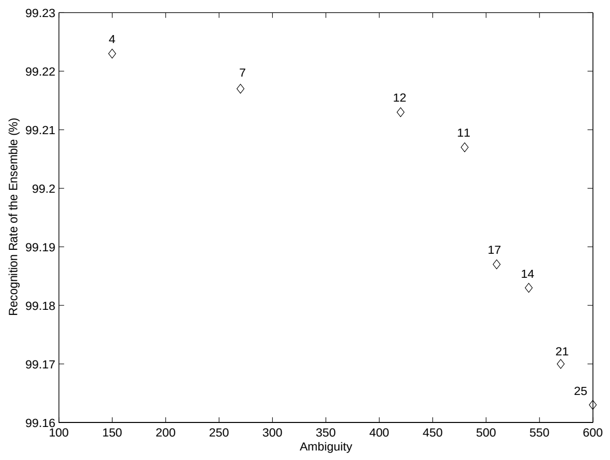

Figure 2. The Pareto-optimal front produced by the second-level MOGA (feature set CC).

Table 3. Performance of the ensembles on TSDB.

Feature Set	Nb. of Classif.	Rec. Rate (%)	Rec. Rate 0.5% (%)	Improv. (%)
CC	4	99.23	98.86	0.36
DDD	4	98.16	95.28	2.48
EDM	7	97.16	89.00	3.90

In order to better evaluate our methodology we have implemented the method proposed by Optiz in [11]. We have chosen Optiz’s method because it seems to be more robust than the others we have found in the literature. Basically, our methodology brought slightly better results but with considerably smaller ensembles. Regarding Optiz’s methodology, the best results were achieved with ensembles composed of about 20 classifiers, while in our’s the ensembles were composed of about 5 classifiers. Moreover, the feature selection method we have applied is designed to tackle huge databases so that the pattern recognition problem can be better represented. On the other hand, our method is more time consuming, since a two-level optimization is necessary.

6 Conclusion

In this paper we have proposed a methodology for ensemble creation based on feature selection. It takes into account a hierarchical MOGA where the first level carries out the feature selection to yield a set of good classifiers while the second one combines them in order to provide a set of powerful ensembles.

The experiments on three different feature sets have demonstrated the validity and efficiency of the proposed strategy by finding small ensembles, which succeed in improving the recognition rates for classifiers working with a very low error rate (0.5%). For future works we plan to use the reliability rate as an objective function in the two levels of the algorithm. We also plan to study the behavior of the ensembles to recognize strings of digits [9].

References

[1] Y. C. Chim, A. A. Kassim, and Y. Ibrahim. Dual classifier system for handprinted alphanumeric character recognition. *PAA*, 1(3):155–162, 1998.

[2] C. Gerra-Salcedo and D. Whitley. Genetic approach to feature selection for ensemble creatin. In *Proc. of Genetic and Evolutionary Computation Conference*, pages 236–243, 1999.

[3] S. Günter and H. Bunke. Creation of classifier ensembles for handwritten word recogntion using feature selection algorithms. In *Proc. of* 8th *IWFHR*, pages 183–188, Niagara-on-the-Lake, Canada, 2002.

[4] T. K. Ho. The random subspace method for constructing decision forests. *IEEE PAMI*, 20(8):832–844, 1998.

[5] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE PAMI*, 20(3):226–239, 1998.

[6] A. Krogh and J. Vedelsby. Neural networks ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems 7*, pages 231–238. 1995.

[7] L. I. Kuncheva and C. J. Whitaker. Ten measures of diversity in classifier ensembles:limits for two classifiers. In *Proc. of IEE Workshop on Intelligent Sensor Processing*, pages 1–10, 2001.

[8] I.-S. Oh and C. Y. Suen. Distance features for neural network-based recognition of handwritten characters. *IJDAR*, 1(2):73–88, 1998.

[9] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Automatic recognition of handwritten numerical strings: A recognition and verification strategy. *IEEE PAMI*, 24(11):1438–1454, 2002.

[10] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Feature selection using multi-objective genetic algorithms for handwritten digit recognition. In 16th *ICPR*, pages 568–571, 2002.

[11] D. W. Optiz. Feature selection for ensembles. In *Proc. of* 16th *International Conference on Artificial Intelligence*, pages 379–384, 1999.

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