Multi-Objective Genetic Algorithms to Create Ensemble of Classifiers

Luiz S. Oliveira, Marisa Morita, Robert Sabourin, and Flávio Bortolozzi

[1] Pontifícia Universidade Católica do Paraná, Curitiba, BRAZIL,
[2] Universidade Tuiuti do Paraná, Curitiba, BRAZIL
[3] Ecole de Technologie Superiéure, Montreal, CANADA
soares@ppgia.pucpr.br

Abstract. Feature selection for ensembles has shown to be an effective strategy for ensemble creation due to its ability of producing good subsets of features, which make the classifiers of the ensemble disagree on difficult cases. In this paper we present an ensemble feature selection approach based on a hierarchical multi-objective genetic algorithm. The algorithm operates in two levels. Firstly, it performs feature selection in order to generate a set of classifiers and then it chooses the best team of classifiers. In order to show its robustness, the method is evaluated in two different contexts: supervised and unsupervised feature selection. In the former, we have considered the problem of handwritten digit recognition while in the latter, we took into account the problem of handwritten month word recognition. Experiments and comparisons with classical methods, such as Bagging and Boosting, demonstrated that the proposed methodology brings compelling improvements when classifiers have to work with very low error rates.

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. It has been 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 [7]. In other words, an ideal ensemble consists of good classifiers (not necessarily excellent) that disagree as much as possible on difficult cases.

The literature has shown that varying the feature subsets used by each member of the ensemble should help to promote this necessary diversity [6,15,18]. Traditional feature selection algorithms aim at finding the best trade-off between features and generalization. On the other hand, ensemble feature selection has the additional goal of finding a set of feature sets that will promote disagreement among the component members of the ensemble. The Random Subspace Method (RMS) proposed by Ho in [6] was one early algorithm that constructs an ensemble by varying the subset of features. Strategies based on genetic algorithms (GAs) also have been proposed [5,15]. All these strategies claim better results than those produced by traditional methods for creating ensembles such as Bagging and Boosting. In spite of the good results brought by GA-based methods, they still can be improved in some aspects, e.g., avoiding classical methods such as

the weighted sum to combine multiple objective functions. 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.

It has been demonstrated that feature selection through multi-objective genetic algorithm (MOGA) is a very powerful tool for finding a set of good classifiers [4,14], since GA is quite effective in rapid global search of large, non-linear and poorly understood spaces [17]. Besides, it can overcome problems such as scaling and sensitivity towards the weights. Kudo and Sklansky [8] have compared several algorithms for feature selection and concluded that GAs are suitable when dealing with large-scale feature selection (number of features is over 50). This is the case of most of the problems in handwriting recognition, which is the test problem in this work.

In this light, we propose an ensemble feature selection approach based on a hierarchical MOGA. The underlying paradigm is the "overproduce and choose" [16]. The algorithm operates in two levels. The former is devoted to generate a set of good classifiers by minimizing two criteria: error rate and number of features. The latter combines these classifiers in order to find an ensemble by maximizing the following two criteria: accuracy of the ensemble and a measure of diversity. We demonstrated through experimentation that using diversity jointly with performance to guide selection can avoid overfitting during the search.

In order to show robustness of the proposed methodology, it was evaluated in two different contexts: supervised and unsupervised feature selection. In the former, we have considered the problem of handwritten digit recognition and used three different feature sets and multi-layer perceptron (MLP) neural networks as classifiers. In the latter, we took into account the problem of handwritten month word recognition and used three different feature sets and hidden Markov models (HMM) as classifiers. We demonstrate that it is feasible to find compact clusters and complementary high-level representations (codebooks) in subspaces without using the recognition results of the system. Experiments and comparisons with classical methods, such as Bagging and Boosting, demonstrated that the proposed methodology brings compelling improvements when classifiers have to work with very low error rates.

2 Methodology Overview

In this section we outline the hierarchical approach proposed. As stated before, it is based on an "overproduce and choose" paradigm where the first level generates several classifiers by conducting feature selection and the second one chooses the best ensemble among such classifiers. Figure 1 depicts the proposed methodology. Firstly, we carry out feature selection by using a MOGA. It gets as inputs a trained classifier and its respective data set. Since the algorithm aims at minimizing two criteria during the search[4], it will produce at the end a 2-dimensional Pareto-optimal front, which contains a set of classifiers (trade-offs between the criteria being optimized). The final step of this first level consists in training such classifiers.

[4] Error rate and number of features in the case of supervised feature selection and a clustering index and the number of features in the case of unsupervised feature selection.

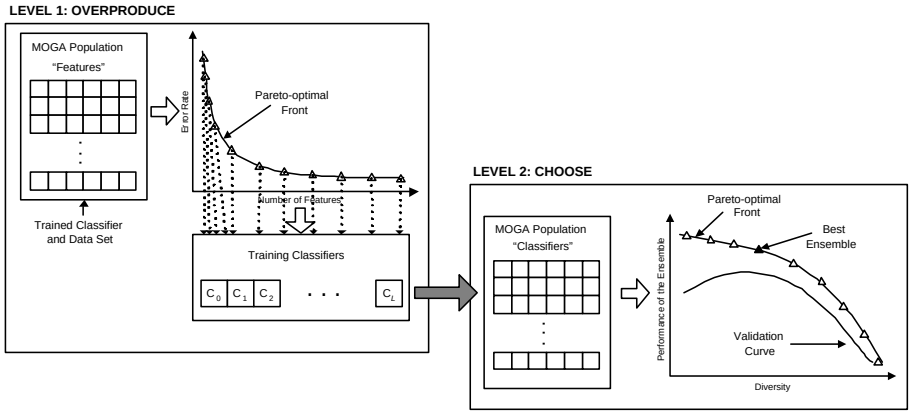


Fig. 1. An overview of the proposed methodology.

Once the set of classifiers have been trained, the second level is suggested to pick the members of the team which are most diverse and accurate. Let $A = \{C_1, C_2, \ldots, C_L\}$ be a set of L classifiers extracted from the Pareto-optimal and B a chromosome of size L 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. Therefore, the algorithm will produce a 2-dimensional Pareto-optimal front which is composed of several ensembles (trade-offs between accuracy and diversity). In order to choose the best one, we use a validation set, which points out the most diverse and accurate team among all. Later in this paper, we will discuss the issue of using diversity to choose the best ensemble.

In both cases, MOGAs are based on bit representation, one-point crossover, and bit-flip mutation. In our experiments, MOGA used is a modified version of the Non-dominated Sorting Genetic Algorithm (NSGA) [2] with elitism.

3 Classifiers and Feature Sets

As stated before, we have carried out experiments in both supervised and unsupervised contexts. The remaining of this section describes the feature sets and classifiers we have used.

3.1 Supervised Context

To evaluate the proposed methodology in the supervised context, we have used three base classifiers trained to recognize handwritten digits of NIST SD19. Such classifiers were trained with three well-known feature sets: Concavities and Contour (CCsc) [13], Distances (DDDsc), and Edge Maps (EMsc). All classifiers here are MLPs trained with the gradient descent applied to a sum-of-squares error function.

The training (TRDBsc) and validation (VLDB1sc) sets are composed of 195,000 and 28,000 samples from hsf_0123 series respectively while the test set (TSDBsc) is composed of 30,089 samples from the hsf_7. We consider also a second validation set (VLDB2sc), which is composed of 30,000 samples of hsf_7. This data is used to select the best ensemble of classifiers. Table 1 reports the performance of all classifiers at zero-rejection level and error rates fixed at low levels (0.10 and 0.50%). These numbers are much more meaningful when dealing with real applications since they describe the recognition rate in relation to a specific error rate, including implicitly a corresponding reject rate. They also corroborates that recognition of handwritten digits is still an open problem when very low error rates are required.

Table 1. Description and performance of the classifiers on TSDB (zero-rejection level) .

Feature Set	Number. of Features	Units in the Hidden Layer	Rec. Rate (%)	Rec. Rate Err=0.1%	Rec. Rate Err=0.5%
CCsc	132	80	99.13	91.83	98.50
DDDsc	96	60	98.17	75.11	92.80
EMsc	125	70	97.04	60.11	85.10

3.2 Unsupervised Context

To evaluate the proposed methodology in unsupervised context we have used three HMM-based classifiers trained to recognize handwritten Brazilian month words (“Janeiro”, “Fevereiro”, “Março”, “Abril”, “Maio”, “Junho”, “Julho”, “Agosto”, “Setembro”, “Outubro”, “Novembro”, “Dezembro”). The training (TRDBuc), validation (VLDB1uc), and testing (TSDBuc) sets are composed of 1,200, 400, and 400 samples, respectively. In order to increase the training and validation sets, we have also considered 8,300 and 1,900 word images, respectively, extracted from the legal amount database. This is possible because we are considering character models. We consider also a second validation set (VLDB2uc) of 500 handwritten Brazilian month words. Such data is used to select the best ensemble of classifiers.

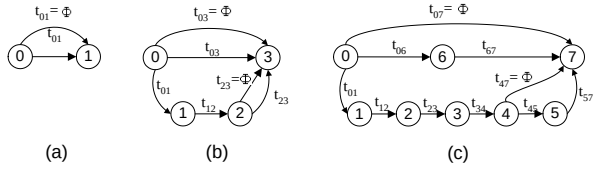


Fig. 2. Topologies of (a) space, (b), and (c) letter models

Given a discrete HMM-based approach, each word image is transformed as a whole into a sequence of observations by the successive application of preprocessing, segmentation, and feature extraction. Preprocessing consists of correcting the average character slant. The segmentation algorithm uses the upper contour minima and some heuristics to split the date image into a sequence of segments (graphemes), each of which consists of a correctly segmented, an under-segmented, or an over-segmented character. A detailed description of the preprocessing and segmentation stages is given in [12].

The word models are formed by the concatenation of appropriate elementary HMMs, which are built at letter and space levels. The topology of space model shown in Figure 2(a) consists of two states linked by two transitions that encode a space (transition t_{01}) or no space (transition $t_{01} = \Phi$).

Two topologies of letter models were chosen based on the output of our grapheme-based segmentation algorithm which may produce a correct segmentation of a letter, a letter under-segmentation or a letter over-segmentation into two, three, or four graphemes depending on each letter. In order to cope with these configurations of segmentations, we have designed topologies with three different paths leading from the initial state to the final state. Considering uppercase and lowercase letters, we need 42 models since the legal amount alphabet is reduced to 21 letter classes and we are not considering the unused ones. Thus, regarding the two topologies, we have 84 HMMs which are trained using the Baum-Welch algorithm with the Cross-Validation procedure.

The feature set that feeds the first classifier is a mixture of concavity and contour features (CCuc) [13]. In this case, each grapheme is divided into two equal zones (horizontal) where for each region a concavity and contour feature vector of 17 components is extracted. Therefore, the final feature vector has 34 components. The other two classifiers make use of a feature set based on distances. The former uses the same zoning discussed before (two equal zones), but in this case, for each region a vector of 16 components is extracted. This leads to a final feature vector of 32 components (DDD32$_u c$). For the latter we have tried a different zoning. Table 2 reports the performance of all classifiers on the test set at zero-rejection level and error rates fixed at 1 and 4%. We have chosen higher error rates in this case due to the size of the database we are dealing with.

Table 2. Performance of the classifiers on the test set.

Feature Set	Number of Features	Codebook Size	Rec. Rate (%)	Rec. Rate Err=1%	Rec. Rate Err=4%
CCuc	34	80	86.1	61.0	79.2
DDD32uc	32	40	73.0	30.5	48.4
DDD64uc	64	60	64.5	24.9	37.0

It can be observed from Table 2 that the recognition rates with error fixed at 1 and 4% are very poor, hence, the number of rejected patterns is very high. We will see in the next sections that the proposed methodology can improve these results considerably.

4 Implementation

This section introduces how we have implemented both levels of the proposed methodology. First we discuss the supervised context and then the unsupervised.

4.1 Supervised Feature Subset Selection

The feature selection algorithm used in here was introduced in [14]. To make this paper self-contained, a brief description is included in this section.

As stated elsewhere, the idea of using feature selection is to promote diversity among the classifiers. To tackle such a task we have to optimize two objective functions: minimization of the number of features and minimization of the error rate of the classifier. Computing the first one is simple, i.e., the number of selected features. 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 [9], who use the sensitivity of the network to estimate the relationship between the input features and the network performance.

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. Such a scheme 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 the number of features and the accuracy of the classifier, and generate the Pareto-optimal front in the first run of the algorithm.

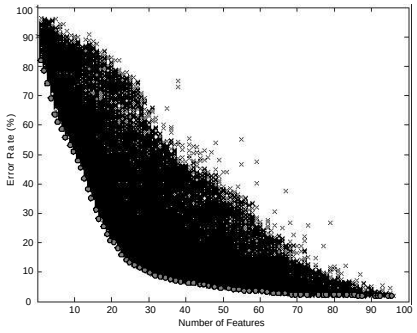


Fig. 3. Evolution of the population in the objective plane

It can be observed in Figure 3 that the Pareto-optimal front is composed of several different classifiers. To find out which classifiers of the Pareto-optimal front compose the best ensemble, we carried out a second level of search. 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 (1).

4.2 Choosing the Best Ensemble

As defined in Section 2 each gene of the chromosome is represented by a classifier produced in the previous level. Therefore, if a chromosome has all bits selected, all

classifiers will compose the team. 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 a measure of diversity. We have tried different measures such as overlap, entropy, and ambiguity [7]. The results achieved with ambiguity and entropy were very similar. In this work we have used ambiguity as diversity measure. The ambiguity is defined as follows:

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

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 classifiers on input x. Thus the ambiguity of an ensemble measured on a set of M samples is

$$\overline{A} = \frac{1}{N} \sum \frac{1}{M} \sum_{k=1}^{M} a_i(x_k) \tag{2}$$

where N is the number of classifiers. So, if the classifiers implement the same functions, the ambiguity $\overline{A}$ will be low, otherwise it will be high.

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. Other combination rules such as product, min, and max have been tested but the simple average has produced slightly better results. In order to evaluate the objective functions during the search described above we have used the validation set VLDB1sc.

4.3 Unsupervised Feature Subset Selection

A lot of work done in the field of handwritten word recognition take into account discrete HMMs as classifiers, which have to be fed with a sequence of discrete values (symbols). This means that before using a continuous feature vector, we must convert it to discrete values. A common way to do that is through clustering. The problem is that for the most of real-life situations we do not know the best number of clusters, what makes it necessary to explore different numbers of clusters using traditional clustering methods such as the K-means algorithm and its variants. In this light, clustering can become a trial-and-error work. Besides, its result may not be very promising especially when the number of clusters is large and not easy to estimate.

Unsupervised feature selection emerges as a clever solution to this problem. The literature contains several studies on feature selection for supervised learning, but only recently, the feature selection for unsupervised learning has been investigated [3]. The objective in unsupervised feature selection is to search for a subset of features that best uncovers “natural” groupings (clusters) from data according to some criterion. In this way, we can avoid the manual process of clustering and find the most discriminative

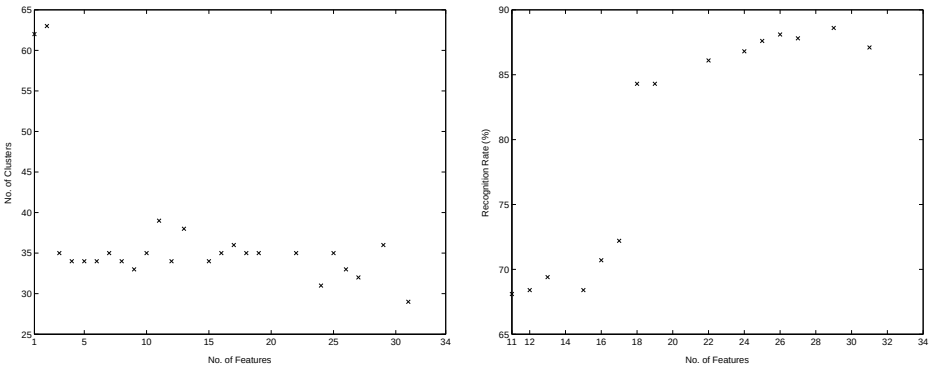
features in the same time. Hence, we will have at the end a more compact and robust high-level representation (symbols).

In the above context, unsupervised feature selection also presents a multi-criterion optimization function, where the objective is to find compact and well separated hyper-spherical clusters in the feature subspaces. Differently of the supervised feature selection, here the criteria optimized by the algorithm are a validity index and the number of features. [11].

In order to measure the quality of clusters during the clustering process, we have used the Davies-Bouldin (DB)-index [1] over 80,000 feature vectors extracted from the training set of 9,500 words. To make such an index suitable for our problem, it must be normalized by the number of selected features. This is due to the fact that it is based on geometric distance metrics and therefore, it is not directly applicable here because it is biased by the dimensionality of the space, which is variable in feature selection problems.

We have noticed that the value of DB index decreases as the number of features increases. We have correlated this effect with the normalization of DB-index by the number of features. In order to compensate this, we have considered as second objective the minimization of the number of features. In this case, one feature must be set at least. Figure 4 depicts the Pareto-optimal front found after the search, the relationship between the number of clusters and number of features and the relationship between the recognition rate on the validation set and the number of features.

Once we have a limited space here, we opted by not showing the Pareto-optimal front for unsupervised case. However, it is very similar to that presented in Figure 3. Figure 4 shows the relationship between the number of clusters and the number of features and the relationship between the recognition rate and the number of features. The way of choosing the best ensemble is exactly the same as introduced in Section 4.2.



(a) (b)

Fig. 4. (a)Relationship between the number of clusters and the number of features and (b) Relationship between the recognition rate and the number of features.

5 Experimental Results

All experiments 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. We have used a Beowulf cluster with 17 (one master and 16 slaves) PCs (1.1Ghz CPU, 512Mb RAM) to execute our experiments.

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 (σ_{share}) = [0.25,0.45]. 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. We have used a standard genetic algorithm with a single-point crossover and the maximum generations of 1000. The fixed crossover and mutation rates are used in a run, and the combination of the crossover rates 0.0, 0.4, 0.6, 0.8 and 1.0 and the mutation rates of $0.1/L$, $1/L$ and $10/L$, where L is the length of the chromosome. The best results were achieved with $P_c = 0.8$ and $P_m = 1/L$. The parameter σ_{share} was tuned empirically.

5.1 Experiments in the Supervised Context

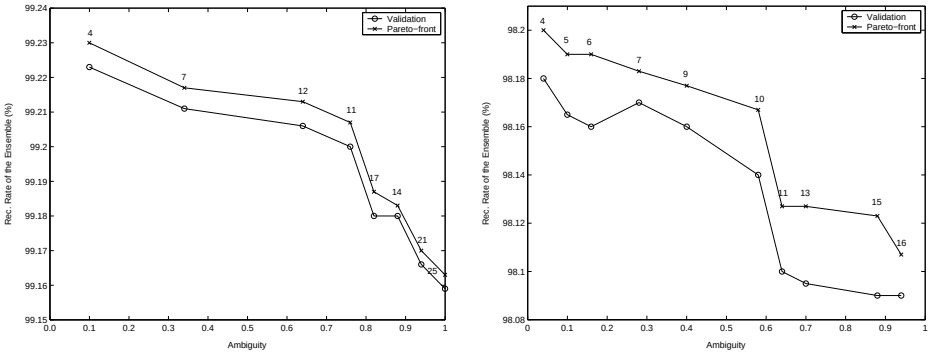
Once all parameters have been defined, the first step, as described in Section 4.1, consists of performing feature selection for a given feature set. As depicted in Figure 3, this procedure produces quite a large number of classifiers, which should be trained for use in the second level. After some experiments, we found out that the second level never chooses those classifiers with poor performance (e.g., error $> 60\%$) to compose the ensemble. Thus, in order to speed up the training process and the second level of search as well, we decide not to use them in the second level. To train such classifiers, the same databases reported in Section 3.1 were used. Table 3 summarizes the classifiers that undergoes to the second level for the three feature sets we have considered.

Table 3. Summary of the classifiers produced by the first level.

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

Considering for example the feature set CC_{sc}, the first level of the algorithm provided 81 classifiers which have the number of features ranging from 24 to 125 and

recognition rates ranging from 90.5% to 99.1% on TSDB$_{sc}$. This shows the great diversity of the classifiers produced by the feature selection method. Based on Table 3 we define four sets of base classifiers as follows: $S_1 = \{CCsc_0, \ldots, CCsc_{80}\}$, $S_2 = \{DDDsc_0, \ldots, DDDsc_{53}\}$, $S_3 = \{EMsc_0, \ldots, EMsc_{77}\}$, and $S_4 = \{S_1 \bigcup S_2 \bigcup S_3\}$. All these sets could be seem as ensembles, but in this work we reserve the word ensemble to characterize the results yielded by the second-level of the algorithm. In order to assess the objective functions of the second-level of the algorithm (generalization of the ensemble and diversity) we have used the validation set (VLDB1sc).



(a) (b)

Fig. 5. The Pareto-optimal front produced by the second-level MOGA: (a) S_1 and (b) S_2

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 the most accurate ensemble among all. Figure 5 depicts the variety of ensembles yielded by the second-level of the algorithm for S_1 and S_2. The number over each point stands for the number of classifiers in the ensemble. In order to decide which ensemble to choose we validate the Pareto-optimal front using VLDB2sc, which was not used so far. Since we are aiming at performance, the direct choice will be the ensemble that provides better generalization on VLDB2$_{sc}$. Table 4 summarizes the best ensembles produced for the four sets of base classifiers and their performance at zero-rejection level on the test set. For facility, we reproduce in this table the results of the original classifiers.

We can notice from Table 4 that the ensembles and base classifiers have very similar performance at zero-rejection level. On the other hand, it also shows that the ensembles respond better for error rates fixed at very low levels than single classifiers. The most expressive result was achieved for the ensemble S_3, which attains a reasonable performance at zero-rejection level but performs very poorly at low error rates. In such a case, the ensemble of classifiers brought an improvement of about 8%. 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 the number of samples rejected is reduced.

Thus, aiming for a small error rate we have to consider the important role of the ensemble. Another fact worth noting though, is the performance of S_4 at low error rates. For the error rate fixed at 1% it reached 95.0% against 93.5% of S_1. S_4 is composed of 14, 6, and 4 classifiers from S_1, S_2, and S_3, respectively. This emphasizes the ability of the algorithm in finding good ensembles when more original classifiers are available.

Table 4. Performance of the ensembles on the test set.

		Ensembles			Original		
Feature	No.	Rec. Rate			Rec. Rate		
Set	Classif.	no Rej.	Err=0.1%	Err=0.5%	no Rej.	Err=0.1%	Err=0.5%
S_1	4	99.22	93.49	98.86	99.13	91.83	98.50
S_2	4	98.18	79.22	95.28	98.17	75.11	92.80
S_3	7	97.10	68.50	89.00	97.04	60.11	85.10
S_4	24	99.25	95.03	98.94			

5.2 Experiments in the Unsupervised Context

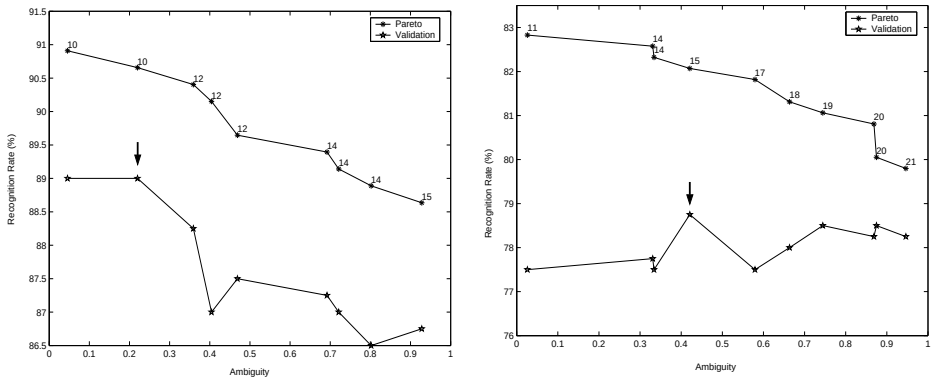
The experiments in the unsupervised context follow the same vein of the supervised one. As discussed in Section 4.3, the main difference lies in the way the feature selection is carried out. In spite of that, we can observe that the number of classifiers produced during unsupervised feature selection is quite large as well. To train the classifiers, the same databases reported in Section 3.2 were considered. Table 5 summarizes the classifiers (after training) produced by the first level for the three feature sets we have considered.

Table 5. Summary of the classifiers produced by the first level.

Feature Set	Number of Classifiers	Range of Features	Range of Codebook	Range of Rec. Rates (%)
CCuc	15	10-32	29-39	68.1 - 88.6
DDD32uc	21	10-31	20-30	71.7 - 78.0
DDD64uc	50	10-64	52-80	60.6 - 78.2

Considering for example the feature set CCuc, the first level of the algorithm provided 15 classifiers which have the number of features ranging from 10 to 32 and recognition rates ranging from 68.1% to 88.6% on VLDB1uc . This shows the great diversity of the classifiers produced by the feature selection method. Based on the classifiers reported in Table 5 we define four sets of base classifiers as follows: $F = \{CCuc_0, \ldots, CCuc_{14}\}$, $F_2 = \{DDD32uc_0, \ldots, DDD32uc_{20}\}$, $F_3 = \{DDD64uc_0, \ldots, DDD64uc_{49}\}$, and $F_4 = \{F_1 \bigcup F_2 \bigcup F_3\}$.

Figure 6 depicts the variety of ensembles yielded by the second-level of the algorithm for F_1 and F_2. The number over each point stands for the number of classifiers in the ensemble. Like in the previous experiments, the second validation set (VLDB2uc)



(a) (b)

Fig. 6. The Pareto-optimal front (and validation curves where the best solutions are highlighted with an arrow) produced by the second-level MOGA: (a) F_1 and (b) F_2

was used to select the best ensemble. After selecting the best ensemble the final step is to assess them on the test set. Table 6 summarizes the performance of the ensembles on the test set. For the sake of comparison, we reproduce in Table 6 the results presented in Table 2.

Table 6. Comparison between ensembles and original classifiers.

		Ensembles			Original		
Feature	No.	Rec. Rate			Rec. Rate		
Set	Classif.	no Rej.	Err=1%	Err=5%	no Rej.	Err=1%	Err=4%
F_1	10	89.2	66.0	81.0	86.1	61.0	79.0
F_2	15	80.2	45.0	60.2	73.0	29.5	48.5
F_3	36	80.7	43.7	62.5	64.5	24.0	36.5
F_4	45	90.2	70.2	77.0			

Like in the previous experiments (supervised context), the result achieved by the ensemble F_4 shows the ability of the algorithm in finding good ensembles when more base classifiers are considered. The ensemble F_4 is composed of 9, 11, and 25 classifiers from F_1, F_2, and F_3, respectively. In light of this, we decided to introduce a new feature set, which, based on our experience, has a good discrimination power when combined with other features such as concavities. This feature set, which we call "global features", is composed of primitives such as ascenders, descenders, and loops. The combination of these primitives plus a primitive that determines whether a grapheme does not contain ascender, descender, and loop produces a 20-symbol alphabet. For more details, see Ref. [10]. In order to train the classifier with this feature set, we have used the same databases described in Section 3.2. The recognition rates at zero-rejection level are 86.1% and

87.2% on validation and testing sets, respectively. This performance compares with the CCuc classifier.

Since we have a new base classifier, our sets of base classifiers must be modified to cope with it. Thus, $F_{1G} = \{F_1 \bigcup G\}$, $F_{2G} = \{F_2 \bigcup G\}$, $F_{3G} = \{F_3 \bigcup G\}$, and $F_{4G} = \{F_1 \bigcup F_2 \bigcup F_3 \bigcup G\}$. In such cases, G stands for the classifier trained with global features. Table 7 summarizes the ensembles found using these new sets of base classifiers. It is worthy of remark the reduction of the size of the teams and the improvement in the recognition rates. This shows the ability of the algorithm in finding not just diverse but also uncorrelated classifiers to compose the ensemble [19]. Besides, it corroborates to our claim that the classifier G when combined with other features bring an improvement to the performance.

Table 7. Performance of the ensembles with global features.

Base Classifiers	Number of Classifiers	Rec. Rate (%) no Rej.	Err=1%	Err=4%
F_{1G}	2	92.2	69.0	87.5
F_{2G}	2	89.7	53.2	80.2
F_{3G}	7	85.5	55.0	75.0
F_{4G}	23	92.0	75.0	88.7

Like the results at zero-rejection level, the improvement observed here also are quite impressive. Table 7 shows that F_{1G} and F_{4G} reach similar results on the test set at zero-rejection level, however, F_{1G} contains just two classifiers against 23 of F_{4G}. On the other hand, the latter features a slightly better error-reject trade-off in the long run.

Based on the experiments reported so far we can affirm that the unsupervised feature selection is a good strategy to generate diverse classifiers. This is made very clear in the experiments regarding the feature set DDD64. In such a case, the original classifier has a poor performance (about 65% on the test set), but when it is used to generate the set of base classifiers, the second-level MOGA was able to produce a good ensemble by maximizing the performance and the ambiguity measure. Such an ensemble of classifiers brought an improvement of about 15% in the recognition rate at zero-rejection level.

6 Discussion and Conclusion

The results obtained here attest that the proposed strategy is able to generate a set of good classifiers in both supervised and unsupervised contexts. To better evaluate our results, we have used two traditional ensemble methods (Bagging and Boosting) in the supervised context. Figure 7 reports the results for CC_{sc}. As we can see, the proposed methodology achieved better results, especially when considering very low error rates.

Diversity is an issue that deserves some attention when discussing ensemble of classifiers. As we have mentioned before, some authors advocated that diversity does not help at all. In our experiments, most of the time, the best ensembles of the Pareto-optimal also were the best for the unseen data. This could lead one to agree that diversity is not important when building ensembles, since even using a validation set the selected team is always the most accurate and with less diversity.

However, if we look carefully the results, we will observe that there are cases where the validation curve does not have the same shape of the Pareto-optimal. In such cases diversity is very useful to avoid selecting overfitted solutions.

One can argue that using a single-objective GA and considering the entire final population, perhaps the similar solutions found in the Pareto-optimal produced by the MOGA will be there. To show that it does not happen, we have carried out some experiments with a single-objective GA where the fitness function was the maximization of the ensemble´s accuracy. Since a single-objetive optimization algorithm searches for an optimum solution, it is natural to expect that it will converge towards the fittest solution, hence, the diversity of solutions presented in the Pareto-optimal is not present in the final population of the single-objective GA.

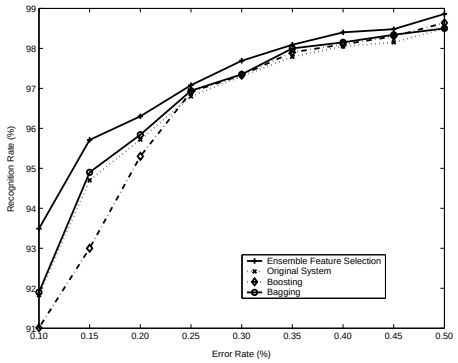


Fig. 7. Comparison among feature selection for ensembles, bagging, and boosting for CC_sc

We have described a methodology for ensemble creation underpinned on the paradigm "overproduce and choose". It takes two levels of search where the first level overproduces a set of classifiers by performing feature selection while the second one chooses the best team of classifiers.

The feasibility of the strategy was demonstrated through comprehensive experiments carried out in the context of handwriting recognition. The idea of generating classifiers through feature selection was proved to be successful in both supervised and unsupervised contexts. The results attained in both situations and using different feature sets and base classifiers demonstrated the efficiency of the proposed strategy by finding powerful ensembles, which succeed in improving the recognition rates for classifiers working with a very low error rates. Such results compare favorably to traditional ensemble methods such as Bagging and Boosting.

Finally we have addressed the issue of using diversity to build ensembles. As we have seen, using diversity jointly with the accuracy of the ensemble as selection criterion might be very helpful to avoid choosing overfitted solutions. Our results certainly brings some contribution to the field, but this still is an open problem.

Acknowledgements

This research has been supported by The National Council for Scientific and Technological Development (CNPq) grant 150542/2003-8.

References

1. D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1(224-227):550–554, 1979.
2. K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Sons Ltd, 2^{nd} edition, April 2002.
3. J. G. Dy and C. E. Brodley. Feature subset selection and order identification for unsupervised learning. In *Proc. 17^{th} International Conference on Machine Learning*, Stanford University-CA, July 2000.
4. 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.
5. 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.
6. T. K. Ho. The random subspace method for constructing decision forests. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
7. A. Krogh and J. Vedelsby. Neural networks ensembles, cross validation, and active learning. In G.Tesauro et al, editor, *Advances in Neural Information Processing Systems 7*, pages 231–238. MIT Press, 1995.
8. M. Kudo and J. Sklansky. Comparision of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33(1):25–41, 2000.
9. 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.
10. M. Morita, L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. An HMM-MLP hybrid system to recognize handwritten dates. In *Proc. of International Joint Conference on Neural Networks*, pages 867–872, Honolulu-USA, 2002. IEEE Press.
11. M. Morita, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition. In *Procs of the 7^{th} ICDAR*, pages 666–670. IEEE Computer Society, 2003.
12. M. Morita, A. El Yacoubi, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Handwritten month word recognition on Brazilian bank cheques. In *Proc. 6^{th} ICDAR*, pages 972–976, 2001.
13. 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*, 24(11):1438–1454, 2002.
14. L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(6):903–930, 2003.
15. D. W. Optiz. Feature selection for ensembles. In *Proc. of* 16^{th} *International Conference on Artificial Intelligence*, pages 379–384, 1999.
16. D. Partridge and W. B. Yates. Engineering multiversion neural-net systems. *Neural Computation*, 8(4):869–893, 1996.
17. W. Siedlecki and J. Sklansky. A note on genetic algorithms for large scale on feature selection. *Pattern Recognition Letters*, 10:335–347, 1989.
18. A. Tsymbal, S. Puuronen, and D. W. Patterson. Ensemble feature selection with the simple Bayesian classification. *Information Fusion*, 4:87–100, 2003.
19. K. Tumer and J. Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(3-4):385–404, 1996.