Unsupervised Feature Selection for Ensemble of Classifiers

Marisa Morita, Luiz S. Oliveira, and Robert Sabourin
Pontifícia Universidade Católica do Paraná, Curitiba, Brazil
École de Technologie Supérieure - Montreal, Canada

Abstract

In this paper we discuss a strategy to create ensemble of classifiers based on unsupervised features selection. It takes into account a hierarchical multi-objective genetic algorithm that generates a set of classifiers by performing feature selection and then combines them to provide a set of powerful ensembles. The proposed method is evaluated in the context of handwritten month word recognition, using three different feature sets and Hidden Markov Models as classifiers. Comprehensive experiments demonstrates the effectiveness of the proposed strategy.

Keywords: Keywords: Ensemble of Classifiers, Unsupervised Feature Selection, Handwriting Recognition, Multi-objective Optimization, Genetic Algorithms.

1 Introduction

The choice of features to represent the patterns affects several aspects of the pattern recognition problem such as accuracy, required learning time, and the necessary number of samples. In this way, the selection of the best discriminative features plays an important role when constructing classifiers. Most of works concerning feature selection have been carried out under the supervised learning paradigm [14, 18], paying little attention to unsupervised learning tasks [4, 8]. Supervised feature selection algorithms are used when class labels of the data are available, otherwise unsupervised feature selection algorithms are employed.

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. This is a difficult task because to find the subset of features that maximizes the performance criterion, the clusters have to be defined. The problem is made more difficult when the number of clusters is unknown beforehand which happens in most real-life situations. Hence, it is necessary to explore different numbers of clusters using traditional clustering methods such as the K-Means algorithm and its variants. Thus, 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.

In this context, unsupervised feature selection presents a multi-criterion optimization function, e.g., the number of features and a validity index to measure the quality of the clusters. In light of this, Multi-Objective Genetic Algorithm (MOGA) offers a particularly attractive approach to solve this kind of problems since they can cope with several objectives in a very clever way and are generally quite effective in rapid global search of large, non-linear, and poorly understood spaces. Another advantage of using MOGA lies in the fact that a set of alternative solutions (different trade-offs between the objective functions being optimized) is available at the end, instead of one single solution.

Such solutions can be helpful in several different ways, but in this paper we are particulary interested in using then to create an ensemble of classifiers, which are characterized by the fact that they produce several classifiers out of one given base classifier automatically. The literature shows us different techniques of ensemble creation, such as Bagging [1], Boosting [5], Random Subspace [7], Input Decimation [17], and Feature Selection [15, 6].

The latter is the strategy adopted in this work and it is based on the work presented by Oliveira et al [15], which generates ensembles of classifiers based on feature selection in the context of supervised learning where the base classifier is a neural network. This approach operates in two different levels. The first one generates a set of classifiers by conducting feature selection and the second one searches the best ensemble among such classifiers. In this paper we propose a methodology for creation of ensembles of Markovian classifiers based on unsupervised feature selection [12]. To the knowledge of the authors, this is the first study that applies unsupervised feature selection to create ensemble of classifiers.

We demonstrate the robustness of the methodology through experimentation on handwritten word recognition, where both recognition and reliability rates were considerably improved.

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 IEEE

2 Multi-Objective Optimization using GA

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. In our experiments, the algorithm adopted is the Non-dominated Sorting Genetic Algorithm (NSGA) with elitism proposed by Srinivas and Deb in [19].

The idea behind NSGA is that a ranking selection method is applied 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 adopted 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 [19, 3].

3 Classifiers and Feature Sets

To evaluate the proposed methodology 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 (TRDB), validation (VLDB), and testing (TRDB) 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 (VLDB2) of 500 handwritten Brazilian month words [16]. Such data is used to select the best ensemble of classifiers.

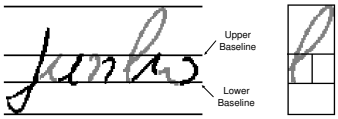

Figure 1. Zoning based on the reference baselines: (a) baselines and (b) 4-region zoning.

The feature set that feeds the first classifier is a mixture of concavity and contour features (CC) [14]. 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 [13]. 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). For the latter we have tried a different zoning. The grapheme is divided into four zones using the reference baselines (see Figure 1), hence, we have a final feature vector composed of 64 components (DDD64). Table 1 reports the performance of all classifiers on the test set, where "Rec.Rate" means the recognition rate at zero-rejection level and "Rec.Rate 1.0%" means the recognition rate achieved for an error rate fixed at 1.0%. 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 rejection 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.

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

It can be observed from Table 1 that the recognition rates with error fixed at 1% 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.

Table 1. Performance of the classifiers on the test set.

Feature Set	No. of Features	Codebook Size	R.R. (%)	R.R. (1%)
CC	34	80	86.1	61.0
DDD32	32	40	73.0	30.0
DDD64	64	60	64.5	24.7

R.R. stands for Recognition Rate.

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 classifiers by conducting unsupervised 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, and bit-flip mutation. The elitism here is implemented using a generational procedure [3]. In summary, the methodology follows five steps: (i) Unsupervised Feature Selection using TRDB, (ii) Train the HMMs produced during feature selection using TRDB and VLDB as training and validation sets, respectively, (iii) Search for the best ensemble of classifiers using VLDB, (iv) Select the best ensemble using VLDB2, (v) Apply the best ensemble on TSDB. In the next subsection we discuss the foregoing steps in details.

4.1 Feature Selection

The unsupervised feature selection algorithm used in this work was introduced in [12]. However, to make this paper self-contained, a brief description is included in this section. It takes into account a MOGA to optimize two criteria: minimization of a validity index and the minimization of the number of features.

In order to measure the quality of clusters during the clustering process, we have used the Davies-Bouldin (DB) index [2] 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 correlated this effect by the normalization of such an 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 2 depicts the Pareto-optimal front found after the search.

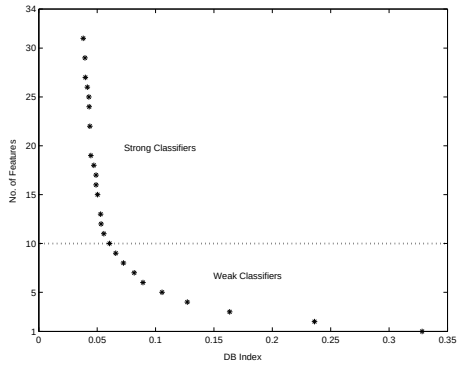

Figure 2. Pareto-optimal front found during feature selection.

After discussing the unsupervised feature selection strategy, let us concentrate on its usage to produce an ensemble of classifiers. To find out which classifiers of the Pareto-optimal front compose the best ensemble, we first train all classifiers of the Pareto and then perform a second level of search. Here is important to mention, though, that in order to speed up the second level of search we have decided to use just those classifiers with more than 10 features. We have realized that those classifiers with very few features are not selected to compose the ensemble. In Section 5 we discuss this issue in more detail. Figure 2 shows the Pareto-optimal front where a line divides the classifiers into two different groups: weak (less than 10 features) and strong (more than 10 features).

4.2 Finding the Best Ensemble

Let $A = C_1, C_2, \ldots, C_n$ be a set of n classifiers extracted from the Pareto-optimal (Figure 2) 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 [10]. We have tried other measures of diversity such as overlap and entropy [11], but the choice of ambiguity yielded better results in our experiments.

The ambiguity is defined as follows:

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

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 classifiers on input x. Thus the contribution to diversity of an ensemble member i as measured on a set of M samples is:

$$A_i = \frac{1}{M}\sum_{k=1}^{M} a_i(x_k) \tag{3}$$

and the ambiguity of the ensemble is

$$\overline{A} = \frac{1}{N}\sum A_i \tag{4}$$

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. In this scenario the error from the ensemble is

$$E = \overline{E} - \overline{A} \tag{5}$$

where $\overline{E}$ is the average errors of the single classifiers and $\overline{A}$ is the ambiguity of the ensemble. Equation 5 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 classifiers implement very similar functions and thus agree in inputs even outside the training set [10].

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 [9]. 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 VLDB.

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.

5 Experiments and Discussion

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}) = 0.35. 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 considered 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 2, this procedure produces quite a large number of classifiers, which should be trained to be used in the second level. After some experiments, we found out that the second level always chooses "strong" classifiers (see Figure 2) 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 "strong" classifiers. This decision was made after we realized that in our experiments the "weak" classifiers did not cooperate with the ensemble at all. To train such classifiers, the same databases reported in Section 3 were considered. Table 2 summarizes the "strong" classifiers (after training) 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	15	10-32	68.1 - 88.6
DDD32	21	10-31	71.7 - 78.0
DDD64	50	10-64	60.6 - 78.2

Considering for example the feature set CC, the first level of the algorithm provided 15 "strong" classifiers which have the number of features ranging from 10 to 32 and recognition rates ranging from 68.1% to 88.6% on VLDB . This shows the great diversity of the classifiers produced by the feature selection method. Based on the classifiers reported in Table 2 we define four sets of base classifiers as follows: $F_1 = \{CC_0, \ldots, CC_{14}\}$,

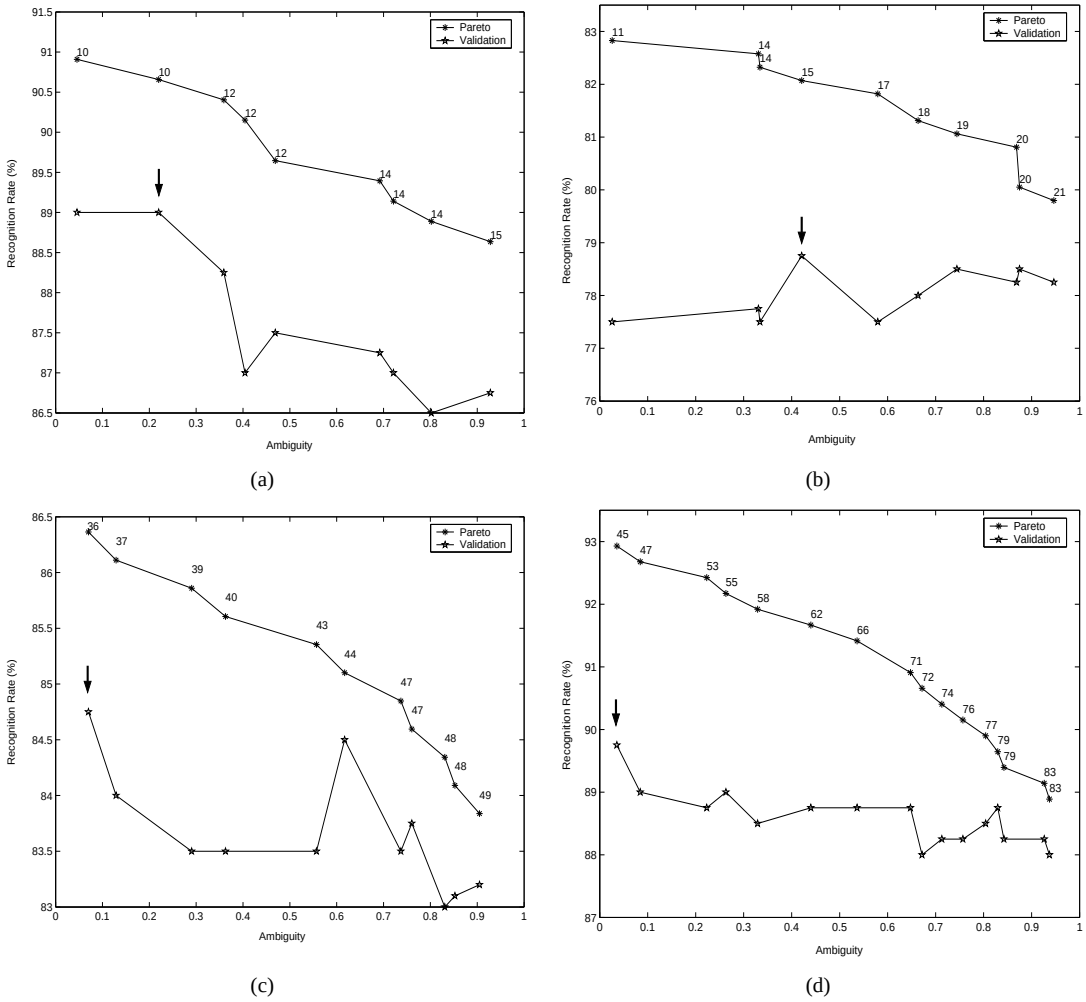

Figure 3. The Pareto-optimal front (and validation curves where the best solutions are highlighted with an arrow) produced by the second-level MOGA: (a) F_1, (b) F_2, (c) F_3, and (d) F_4.

$F_2 = \{DDD32_0, \ldots, DDD32_{20}\}$, $F_3 = \{DDD64_0, \ldots, DDD64_{49}\}$, and $F_4 = \{F_1 \bigcup F_2 \bigcup F_3\}$. 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 MOGA.

Like the first level, the second one also generates a set of possible solutions (Pareto-optimal front) which are the trade-offs between the generalization of the ensemble and its diversity. Thus the problem now lies in choosing the best ensemble among all. Figure 3 depicts the variety of ensembles yielded by the second-level MOGA for the four sets of base classifiers. 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 VLDB2, 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.

After selecting the best ensemble the final step is to assess them on the test set. Table 3 summarizes the performance of the ensembles on the test set.

Figure 3 also shows the performance of the ensembles generated with all base classifiers available, i.e., Ensemble F_4. In this experiment we would expect the algorithm to achieve at least the performance presented by the most powerful ensemble (Ensemble F_1). In fact, it did better (see Table 3). The result achieved by the ensemble F_4 shows the ability of the algorithm in finding good ensembles when more base classifiers are considered.

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

Table 3. Performance of the ensemble on the test set.

Base Classifiers	Number of Classifiers	Rec. Rate (%)	Rec. Rate (1%)
F_1	10	89.2	70.0
F_2	15	80.2	45.9
F_3	36	80.7	43.7
F_4	45*	90.2	73.2

*This ensemble is composed of 9, 11, and 25 classifiers from F_1, F_2, and F_3, respectively.

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 Conclusion

We have proposed a methodology for ensemble creation based on unsupervised feature selection. It considers a hierarchical MOGA where the first level performs unsupervised feature selection to produce a set of 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 ensembles, which succeed in improving the recognition rates for classifiers working with low error rates (1%). Our next efforts will be devoted to reduce the computational cost of this strategy, which is quite expensive.

Acknowledgements

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

References

[1] L. Breiman. Stacked regressions. *Machine Learning*, 24(1):49–64, 1996.

[2] 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.

[3] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Sons Ltd, 2001.

[4] 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*, pages 247–254, 2000.

[5] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Proc. of 13^{th} International Conference on Machine Learning*, pages 148–156, 1996.

[6] S. Gunter and H. Bunke. Creation of classifier ensembles for handwritten word recogntion using feature selection algorithms. In *Proc. of 8^{th} IWFHR*, pages 183–188, 2002.

[7] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[8] Y. S. Kim, W. N. Street, and F. Menczer. Feature selection in unsupervised learning via evolutionary search. In *Proc. 6^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 365–369, 2000.

[9] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[10] A. Krogh and J. Vedelsby. Neural networks ensembles, cross validation, and active learning. In G. et al, editor, *Advances in Neural Information Processing Systems 7*, pages 231–238. MIT Press, 1995.

[11] 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.

[12] 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, 2003.

[13] I.-S. Oh and C. Y. Suen. Distance features for neural network-based recognition of handwritten characters. *International Journal on Document Analysis and Recognition*, 1(2):73–88, 1998.

[14] 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.

[15] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Feature selection for ensembles: A hierarchical multi-objective genetic algorithm approach. In *Proceedings of the 7^{th} ICDAR*, pages 676–680, 2003.

[16] J. J. Oliveira-Jr., J. M. Carvalho, C. O. A. Freitas, and R. Sabourin. Evaluating NN and HMM classifiers for handwritten word recognition. In *Proceedings of the 15^{th} Brazilian Symposium on Computer Graphics and Image Processing*, pages 210–217, 2002.

[17] N. C. Oza and K. Tumer. Input decimation ensembles: Decorrelation through dimensionality reduction. In *Proc. of the 2^{nd} International Workshop on Multiple Classifier Systems*, pages 238–247, Cambridge, UK, 2001.

[18] W. Siedlecki and J. Sklansky. A note on genetic algorithms for large scale on feature selection. *Pattern Recognition Letters*, 10:335–347, 1989.

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