

# Intelligent Feature Extraction for Ensemble of Classifiers

†Paulo V. W. Radtke<sup>1,2</sup>, Robert Sabourin<sup>1,2</sup>, Tony Wong<sup>1</sup>

<sup>1</sup>École de Technologie Supérieure - Montreal, Canada

<sup>2</sup>Pontifícia Universidade Católica do Paraná - Curitiba, Brazil

†e-mail: radtke@livia.etsmtl.ca

## Abstract

*This paper presents a two-level approach to create ensemble of classifiers based on intelligent feature extraction and multi-objective genetic optimization. The first stage optimizes a set of representations, which is used to create classifiers. The second stage then optimizes the ensemble's aggregated classifiers. To assess the approach's feasibility, a set of tests with isolated handwritten digits is performed. The experimental results encourage further researches in this direction, as the optimized ensemble of classifiers outperforms the single classifier approach.*

## 1 Introduction

The task to create a classifier is usually done by a human expert, who creates several candidate designs to select the most performing classifier for a given problem. In the context of isolated handwritten symbols, this process can be modeled as a multi-objective optimization problem, using domain knowledge to optimize *representations* based on features extracted from zones to train classifiers. The *Intelligent Feature Extraction* (IFE) methodology optimizes a representation set  $RS$ , used to train a classifier set  $K$  where the most performing classifier is selected.

For higher accuracy, we use a wrapper approach during optimization to evaluate candidate representations. To reduce the computational burden we need a classifier which is both fast to train and accurate. The projection distance (PD) classifier [1] satisfies these conditions, creating a compact representation based on separate hyperplanes, while requiring a reduced data set for training, 50000 observations for learning and 15000 to optimize the hyperplanes. Our first hypothesis is that the set  $RS$  optimized with the IFE can be used to train more discriminant classifiers, such as a multi layer perceptron (MLP), and yet select in  $K$  a classifier that outperforms a classifier designed by traditional approaches.

When optimizing this type of supervised learning problems, the objective function space during optimization does not match the objective space on actual unseen observa-

tions, hence an optimization algorithm adapted to these problems is needed.

An Ensemble of Classifiers (EoC) is typically created by running several times a learning algorithm to create a set of classifiers, which are combined by an aggregation function. The key issue in this process is to generate a set of diverse and fairly accurate classifiers [2]. Our second hypothesis assumes that the set  $RS$  optimized by the IFE creates a diverse set  $K$  of classifiers, whose combination can be optimized to create an EoC. This problem is also modeled to be solved by means of multi-objective optimization.

Thus, we propose in this paper a two-level approach to generate an EoC to recognize isolated handwritten symbols with supervised learning. Section 2 discusses the IFE methodology, while Sect. 3 introduces the *Multi-Objective Memetic Algorithm* (MOMA) used by the IFE. Section 4 presents the strategy to optimize an EoC from the IFE results. Section 5 details the experiments performed to validate the proposed hypothesis, with the results presented in Sect. 6. Finally, Sect. 7 discusses the results and their implications.

## 2 Intelligent Feature Extractor

In order to create a classifier, isolated handwritten symbols are modeled as representations, based on features extracted from specific *foci* of attention on images using *zoning*. Three operators are used to generate representations: a *zoning operator* based on a zoning mechanism, a *feature extraction operator* to apply transformations in zones, and a *feature subset selection operator* that removes irrelevant features. The IFE user's domain knowledge is introduced in the choice of transformations for the feature extraction operator.

The *zoning operator* defines the zoning strategy  $Z = \{z^1, \dots, z^n\}$ , where  $z^i, 1 \leq i \leq n$  is a zone in the image  $I$  and  $n$  the number of zones. The pixels inside the zones in  $Z$  are transformed by the *feature extraction operator* to a representation  $F = \{f^1, \dots, f^n\}$ , where  $f^i$  is the feature vector extracted from  $z^i$ .  $F$  has the irrelevant features elim-

inated by the *feature subset selection operator*, producing the representation  $G = \{g^1, \dots, g^n\}$ ,  $g^i$  being the feature subset of  $f^i$ . This process is illustrated in Fig. 1.

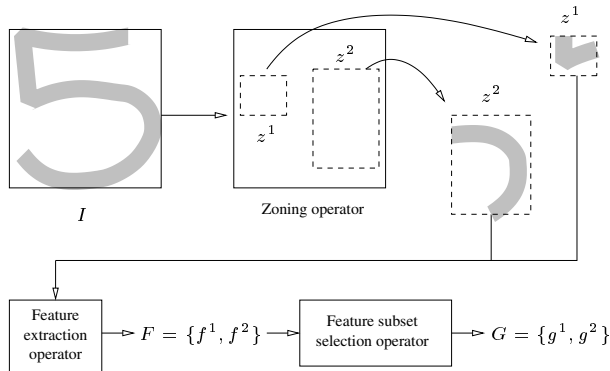


Figure 1. IFE hierarchical structure

Candidate solutions are represented on a hierarchical genetic coding, with three different parts, each related to an IFE operator. This strategy is indicated in Fig. 2, where parts are hierarchical in the sense that the coding in one part will determine the data manipulated by another. The following sections discuss the optimization strategy, IFE operators, candidate solution evaluation and the selection procedure after optimization.

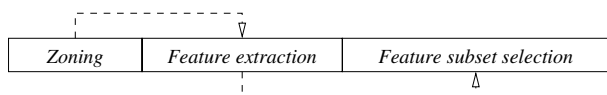


Figure 2. IFE candidate solution coding

## 2.1 Dividers Zoning Operator

To compare the IFE against the traditional approaches we consider a baseline representation known to achieve high accuracy with isolated handwritten digits [4]. The zoning on this representation can be defined as a set of three dividers, where the intersection of image borders and dividers defines zones as 4-sided polygons. We expand this concept into a set of 5 horizontal and 5 vertical dividers that can be either *active* or *inactive*. Figure 3.a details the operator template, represented by a 10 bits binary string, each bit associated to a divider. This operator produces zoning strategies with 1 to 36 zones, and the baseline zoning in Fig. 3.b is obtained by activating only  $d_2$ ,  $d_6$  and  $d_8$ . Figure 3.c depicts a zoning strategy optimized by the IFE.

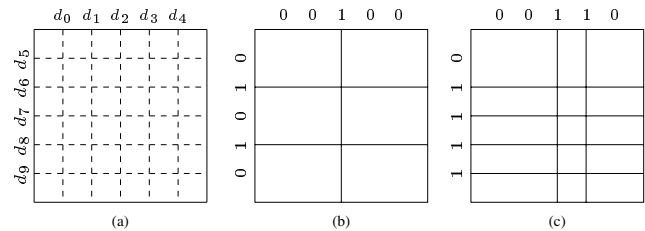


Figure 3. Dividers zoning operator

## 2.2 Feature Extraction Operator

In [4], Oliveira et al. used a mixture of concavities, contour and surface transformations, extracting 22 features per zone – 13 for concavities, 8 for contour and 1 for surface. With three different transformations the operator is encoded as a three bits binary string, where each bit indicates the state of the associated transformation. When all transformations are inactive, the zone becomes a missing part [3], a zone with no features extracted.

## 2.3 Feature Subset Selection Operator

This operator selects the most relevant features in the feature vector  $F = \{f^1, \dots, f^n\}$ , creating a final representation  $G = \{g^1, \dots, g^n\}$ . This task is performed with a binary string associated to each feature vector  $f^i$ . Each bit in the string indicates if the associated feature is active or not. Thus a 22 bits binary string is required to encode the feature extraction operator described in the previous section.

## 2.4 Solutions Evaluation and Selection

We search for small, yet accurate representations. Hence two objectives guide the optimization process, classifier error rate and feature set cardinality. Error rate is evaluated on a wrapper approach using the PD classifier, while the cardinality is simply the count of features extracted. The PD classifier is chosen due to its accuracy when trained with a smaller training set, as well as for the training speedup when compared to more discriminant approaches, such as MLP or SVM classifiers.

Pareto based multi-objective evolutionary approaches uses the dominance relation to emphasize solutions. However, optimization of supervised learning problems is plagued by non matching objective function spaces between optimization and actual classification stages – good performance during optimization does not imply in good generalization. Figure 4 illustrates this effect on classifiers optimized by the IFE. During optimization in Fig. 4.a, a Pareto-based approach emphasizes solution *B*, which dominates solutions *A* and *C*.

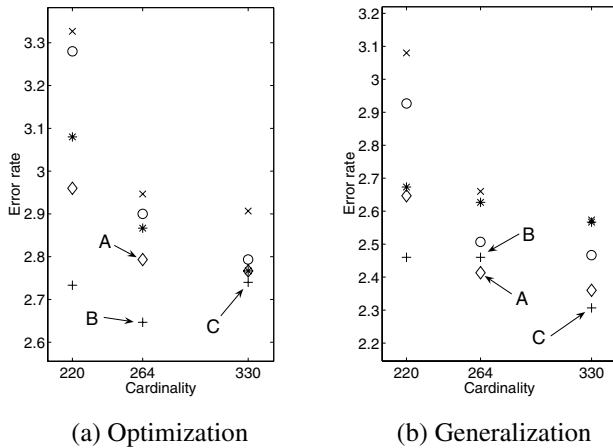


Figure 4. Objective function space

When classifying unseen observations in Fig. 4.b, solutions *A* dominates solution *B*, while solution *C* is non-dominated. To overcome this effect we need a post-processing stage, using a different database to test optimized solutions and then select a representation with good generalization on unseen data.

To perform the post-processing stage, the optimization algorithm must comply to two requirements. For each possible feature set cardinality, it must archive a set of most performing solutions. Also, we need to emphasize the best error rate for each cardinality value, regardless of the dominance relation.

### 3 Multi-Objective Memetic Algorithm

Traditional *Multi-Objective Genetic Algorithms* (MOGA) are based on the Pareto dominance concept. Section 2.4 demonstrated that solutions overlooked by these approaches may in fact have better generalization properties than non-dominated solutions. To optimize properly the IFE methodology we proposed the MOMA algorithm [6], which combines a traditional Multi-objective Genetic Algorithm (MOGA) with a local search (LS) algorithm, featuring modified selection and archiving strategies suitable for the IFE methodology.

To archive solutions as defined in Sect. 2.4, objective functions are divided in two categories, *objective function one* ( $o_1$ ) in the integer domain, defining the archive's *slots*  $S$ , and *objective function two* ( $o_2$ ), optimized for each  $o_1$  value. Each slot  $S^l$  is a set of  $max_{S^l}$  solutions, associated to a possible  $o_1$  value. For the IFE,  $o_1$  is feature set cardinality and  $o_2$  is the error rate.

### 3.1 Algorithm Overview

The MOMA structure is depicted in Fig. 5. It evolves a population  $P$  of size  $m$ , archiving good solutions in the slots  $S$  at the end of each generation.  $P$  is initialized in two steps, the first creates candidate solutions with a Bernoulli distribution, while the second generates individuals to initialize the slots. For each slot, we choose one random solution that is admissible in the slot and insert it in the population.

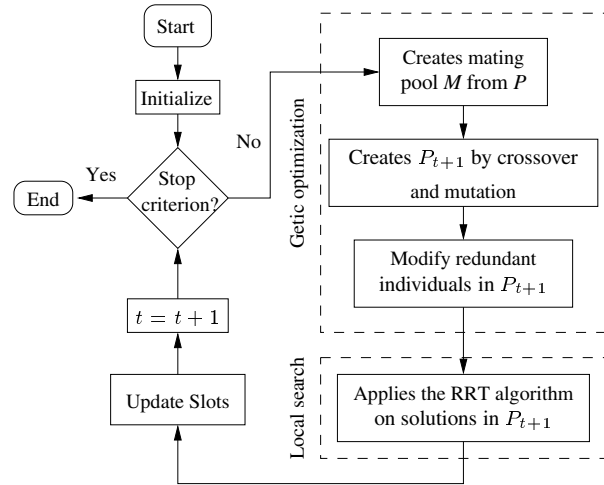


Figure 5. The MOMA algorithm

At each iteration  $t$  individuals in  $P_t$  are ranked by frontier to create a mating pool  $M$  by tournament selection, followed by crossover and mutation to create the offspring population  $P_{t+1}$ . In case of a draw in the tournament selection, one of the solutions is chosen randomly. To avoid genetic overtake, redundant individuals are mutated until the population has no redundant individuals.

After genetic operations solutions are improved by the Record-to-Record Travel (RRT) algorithm [5], an annealing based heuristic. The RRT improves solutions in  $P_{t+1}$  by searching in its neighborhood for  $n$  potential solutions during  $NI$  iterations, allowing a decrease in the current performance of  $a\%$  to avoid local optimal solutions. The last step updates the archive  $S$  with good solutions from  $P_{t+1}$ . At this point, we verify the stopping criterion, deciding if the algorithm should continue to the next iteration or stop the optimization process.

### 3.2 Algorithm Discussion

To provide the features discussed in Sect. 2.4, the MOMA algorithm differs from traditional MOGAs in the archiving and selection strategies. The archive is a set  $S = \{S^1, \dots, S^j\}$ , which stores a set of solutions that contains the best  $o_2$  value for  $o_1$  associated to  $S^l$ , as well as close neighbor solutions, limited by  $max_{S^l}$ .

To create the mating pool  $M$ , individuals are ranked by *frontier*. Given that  $A(S^l, C)$  denotes the subset of solutions in  $C$  that are admissible in  $S^l$ , solutions belonging to the first rank are defined by  $R^1 = \bigcup_{l=1}^j \{B(A(S^l, P))\}$ , where  $B(C)$  denotes the solution with the lowest  $o_2$  value. The solution set belonging to the second rank  $R^2$  is obtained as the first rank of  $P \setminus R^1$ , and so on.

For genetic operations, we use hierarchical versions of the traditional single-point crossover and bitwise mutation. Instead of applying these operators on the whole binary string encoding the hierarchical representation in Fig. 2, they are applied independently on each operator.

#### 4 Ensemble of Classifiers Optimization

To create an EoC our hypothesis is to optimize which classifiers to aggregate from the set  $K = \{K^1, \dots, K^p\}$ , where  $K^i$  is the classifier trained with the representation  $G^i$  in the result set  $RS = \{G^1, \dots, G^p\}$  optimized by the IFE. As this problem is an example based learning problem, in the same context as the IFE, we use the MOMA algorithm to avoid the problems discussed in Sect. 2.4.

To realize this task, the classifiers in  $K$  are associated to a binary string  $E$  of  $p$  bits, which is optimized to select the best combination of classifiers using the MOMA algorithm. The classifier  $K^i$  is associated to the  $i^{th}$  binary value in  $E$ , which indicates if the classifier is active in the EoC or not. The diversity of classifiers in the EoC impacts directly on the ensemble error rate. Thus, instead of optimizing a diversity metric, we use a wrapper approach where the actual combined EoC error rate is associated to objective  $o_2$  during optimization. As we wish for an EoC that is both accurate and has a lower cost associated to the classification stage, we associate objective  $o_1$  to the number of active classifiers in the EoC.

Our initial hypothesis for the IFE methodology is that the representations in  $RS$  can be used to train more discriminant classifiers. Therefore, we test this EoC optimization approach with both the PD classifier, as a proof of concept, and with a MLP classifier. To aggregate the PD classifiers we use majority voting, while MLP classifiers are aggregated by the MLPs outputs average [2].

#### 5 Experimental Protocol

To compare the results with the baseline representation, we extract the same features from all zones as in [4]. To achieve this, both feature extraction and feature subset selection operators are fixed, and only the zoning operator is optimized. All experiments use the databases in Table 1, isolated handwritten digits extracted from the NIST SD-19 database. The first stage of our experiments optimizes the

representation set  $RS$  with the IFE methodology using the MOMA algorithm. To evaluate solutions in the wrapper approach we train the PD classifier using *learn'* as the learning samples, and *validation* to configure the number of hyperplanes. To calculate  $o_2$  we use the classifier's error rate on the *optimization* database.

Database	Size	Origin	Initial sample
<i>learn</i>	150000	hsf_0123	1
<i>learn'</i>	50000	hsf_0123	1
<i>validation</i>	15000	hsf_0123	150001
<i>optimization</i>	15000	hsf_0123	165001
<i>selection</i>	15000	hsf_0123	180001
<i>test</i>	60089	hsf_7	1

Table 1. Digits databases

The parameters used on the MOMA algorithm to optimize the set  $RS$  with the IFE are the following. Crossover probability is set to  $p_c = 80\%$ , while mutation is set to  $p_m = 1/L$ , where  $L$  is the length of the mutated binary string. During 1000 generations, the local search will look in  $NI = 3$  iterations for  $n = 1$  neighbors, with deviation  $a = 0\%$ . Each slot is allowed to store  $max_{sl} = 5$  solutions, and the population size is  $m = 64$ . As we store 5 solutions per slot, we keep in the archive a total of  $p = 82$  solutions. At the last generation the archive is the set  $RS$ .

Once the representation set  $RS$  is optimized we test our hypotheses. We first train classifiers with the representations in  $RS$  to create the set  $K$ . Next we select the single best classifier  $SB$  from  $K$ , using the *selection* database to evaluate the classifiers error rate to select  $SB$  as the most accurate classifier. We then compare  $SB$  against the baseline classifier on the *test* database.

To assess the second hypothesis we use the classifiers in  $K$  to optimize an EoC, as described in Sect. 4. The MOMA algorithm parameters are the same as when optimizing the IFE, except for the population size  $m = 186$ . To evaluate the error rate of candidate EoCs in the optimization process, we also use the *optimization* database. After optimization, all EoCs in the result set have their performance evaluated on the *selection* database to select the most accurate EoC.

Both hypotheses are tested with the PD classifier, as a proof of concept, and with the MLP classifier, a well known discriminant classifier. We call them *Test A*, with the PD classifier, and *Test B*, with the MLP classifier. The PD classifier is trained as when optimizing the IFE, while the MLP is trained using *learn* as the training samples and *validation* to verify the MLP generalization power when optimizing the number of hidden nodes (HN).

## 6 Results

After optimizing the set  $RS$  with the IFE, we proceeded to *Test A*, using the PD classifier. We created  $K_{PD}$  and evaluated these classifiers with the *selection* database. The classifier with the smallest error rate is labeled  $SB_{PD}$ , and its zoning operator is indicated in Fig. 3.c. We also optimized the EoC and evaluated the result set with the *selection* database, labeling the most accurate as  $EoC_{PD}$ .

We then compared the performance of the baseline PD classifier with  $SB_{PD}$  and  $EoC_{PD}$  on the *test* database. These results are in Table 2, which details the feature set cardinality ( $|G|$ ), the error rate on the *selection* and *test* databases ( $e_{sel}$  and  $e_{test}$ ), and the number of classifiers aggregated in the EoC ( $|EoC|$ ).

Classifier	$ G $	$e_{sel}$	$e_{test}$	$ EoC $
Baseline	132	3.01%	2.96%	–
$SB_{PD}$	330	2.31%	2.18%	–
$EoC_{PD}$	–	1.51%	1.94%	19

**Table 2. PD classifier results**

These results confirm our hypotheses in the proof of concept using the PD classifier. The IFE optimized a representation that outperforms the baseline representation, created using the traditional human expert approach. The EoC optimized also shows improved accuracy against the single classifier approaches, which confirms the hypothesis behind the proposed two-level approach to create the EoC.

It is known in the literature that the MLP classifier is more accurate than the PD classifier, therefore we test our hypotheses with this classifier in *Test B*. The same steps are made, selecting the best single classifier  $SB_{MLP}$  and the ensemble  $EoC_{MLP}$ . The results for this test are in Table 3, where  $HN$  is the MLP's number of hidden nodes, confirming both hypotheses with the MLP classifier.

Classifier	$ G $	$HN$	$e_{sel}$	$e_{test}$	$ EoC $
Baseline	132	70	0.44%	0.89%	–
$SB_{MLP}$	330	130	0.41%	0.79%	–
$EoC_{MLP}$	–	–	0.37%	0.74%	4

**Table 3. MLP classifier results**

## 7 Discussion

The representations optimized by the IFE methodology using the PD classifier have good generalization with a more discriminant classifier, as  $SB_{MLP}$  has lower error

rates than the baseline MLP classifier. As  $EoC_{MLP}$  outperforms  $SB_{MLP}$ , we confirm in a more robust classifier the proposed methodology for EoC creation.

One aspect to note is that the representation used to train  $SB_{MLP}$  is the same used to train  $SB_{PD}$  in Fig. 3.c. This suggests that the selection procedure is capable of finding representations that have good generalization power for a single classifier, using only the simple and fast PD classifier. Further experiments with the complete IFE methodology, optimizing all operators, will investigate this property.

## 8 Acknowledgments

The first author would like to acknowledge the CAPES and the Brazilian government for supporting this research through scholarship grant BEX 2234/03-3.

## References

- [1] F. Kimura, S. Inoue, T. Wakabayashi, S. Tsuruoka, and Y. Miyake. Handwritten Numeral Recognition using Autoassociative Neural Networks. In *Proceedings of the International Conference on Pattern Recognition*, pages 152–155, 1998.
- [2] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.
- [3] Z.-C. Li and C. Y. Suen. The partition-combination method for recognition of handwritten characters. *Pattern Recognition Letters*, (21):701–720, 2000.
- [4] 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.
- [5] J. W. Pepper, B. L. Golden, and E. A. Wasil. Solving the traveling salesman problem with annealing-based heuristics: A computational study. *IEEE Trans. on Systems, Man and Cybernetics – Part A: Systems and Humans*, 32(1):72–77, 2002.
- [6] P. V. W. Radtke, T. Wong, and R. Sabourin. A Multi-Objective Memetic Algorithm for Intelligent Feature Extraction. In *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 3410 of *Lecture Notes in Computer Science*, Berlin, 2005. Springer-Verlag.