A Quantitative Comparison Between MOGAs and the RRT Algorithm on Classification Systems Optimization

Paulo V. W. Radtke[1], **Robert Sabourin**[2], **Tony wong**[2]

[1]Programa de Pós-Graduação em Informática Aplicada
Pontifícia Universidade Católica do Paraná (PUCPR)
Rua Imaculada Conceição, 1155 – 80215-901 – Curitiba – PR – Brazil

[2]Département de Génie de la Production Automatisée
École de Technologie Supérieure (ETS)
1100, Rue Notre-Dame Ouest – H3C 1K3 – Montréal - QC - Canada

`pvwradtke@gmail.com, {robert.sabourin, tony.wong}@etsmtl.ca`

Abstract. *Genetic algorithms are powerful population based optimization methods. Their multi-objective counterparts have been often used to effectively optimize classification systems, but little is discussed on their computational cost to solve such problems. To better understand this issue, an annealing based approach to optimize a classification system is proposed and discussed. Results are then compared to results obtained with a multi-objective genetic algorithm in the same problem. The experiments performed with isolated handwritten digits demonstrate both the effectiveness and lower computational cost of the annealing based approach.*

Resumo. *Algoritmos genéticos são métodos de otimização baseados em população. Seus equivalentes multi-critério são usados freqüentemente na otimização de sistemas de classificação, mas pouco se discute sobre o custo computacional ao solucionar tais problemas. Para entender melhor esta relação, é proposta a utilização de uma abordagem baseada em* simulated annealing. *Os resultados são comparados com os obtidos por algoritmos genéticos multi-critério no mesmo problema. Os experimentos com dígitos manuscritos isolados indicam a eficácia e baixo custo computacional da abordagem baseada em* simulated annealing.

1. Introduction

Classification systems will usually require that the image pixel information be first transformed into an abstract representation (a feature vector) suitable for recognition with classifiers, a process known as *feature extraction*. This process is performed to transform the the data used for classification into a more discriminant representation. It has also been observed that partitioning the image provides better results than when extracting features from the whole image [Li and Suen 2000]. Thus, a methodology that extract features must select the spatial location to apply feature transformations on the image. This choice regards the *domain context*, the type of symbols to classify, and the *domain knowledge*, what was previously done in similar problems. An human expert is usually responsible for the trial-and-error approach used during traditional feature extraction. The problem is also exacerbated by the fact that changes in the domain context requires a new feature set for proper classification. This context mandates a semi-automated approach that uses the expert's domain knowledge to optimize the classification system.

To minimize the human intervention in defining and adapting classification systems, this problem is modeled as an optimization problem, using the domain knowledge and the domain context. This paper discusses the two-level approach to optimize classification systems in Fig. 1. The first level employs the *Intelligent Feature Extraction* (IFE) methodology to extract feature sets that are used on the second level to optimize an *ensemble of classifiers* (EoC) to improve accuracy.

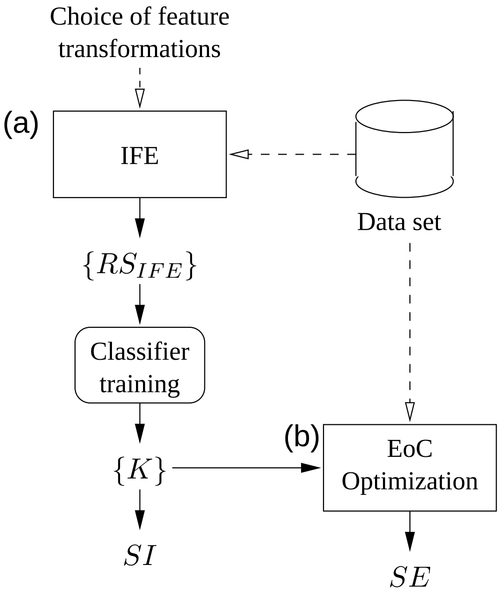


Figure 1. Classification system optimization approach. Representations obtained with IFE are used to further improve accuracy with EoCs.

One trend for these classification problems is to use genetic based approaches [Kuncheva and Jain 2000, Tremblay et al. 2004, Tsymbal et al. 2005, Handl and Knowles 2006], specially *multi-objective genetic algorithms* (MOGAs). These approaches have been found effective to solve these problems. It is now understood that the advantage of MOGA lies in the inherent diversity of the optimized solution set, avoiding the population convergence to a single local optimum. However, how efficient are these algorithms is an unanswered question. Population based approaches evaluate a large number of candidate solutions. If using a wrapper approach, training and testing solutions may take a considerable time. Hence, the use of other algorithms may provide comparable solutions associated to a lower computational burden. The algorithm chosen for this comparative study is the *Record-to-Record Travel* (RRT) algorithm [Pepper et al. 2002], an annealing based heuristic. This local search algorithm features a strategy to avoid local optimum solutions, a feature often required to optimize classification problems.

This paper extends the work in [Radtke et al. 2006a]. The new contribution is to investigate an annealing based approach to optimize classification systems for a quantitative comparison with MOGA results. The paper has the following structure. The approach to optimize classification systems is discussed in Section 2, and Section 3 discusses the RRT algorithm. Section 4 details the experimental protocol and the results obtained. Finally, Section 5 discusses the goals attained.

2. Classification System Optimization

Classification systems are modeled in a two-level process (Fig. 1). The first level uses the IFE methodology to obtain the representation set RS_{IFE} (Fig. 1.a). The representations in RS_{IFE} are then used to train the classifier set K that is considered for aggregation on an EoC SE for improved accuracy (Fig. 1.b). Otherwise, if a single classifier is desired for limited hardware, such as embedded devices, the most accurate single classifier SI may be selected from K. The next two subsections details both the IFE and EoC optimization methodologies.

2.1. Intelligent Feature Extraction

The goal of IFE is to help the human expert define representations in the context of isolated handwritten symbols, using a wrapper approach to optimize solutions. IFE models handwritten symbols as features extracted from specific *foci* of attention on images using *zoning*. Two operators are used to generate representations with IFE: a *zoning operator* to define foci of attention over images, and a *feature extraction* operator to apply transformations in zones. The choice of transformations for the feature extraction operator constitutes the domain knowledge. The domain context is introduced as actual observations in the *optimization* data set used to evaluate and compare solutions. Hence, the zoning operator is optimized by the IFE to the domain context and domain knowledge.

The IFE structure is illustrated in Fig. 2. The zoning operator defines the zoning strategy $Z = \{z^1, \ldots, z^n\}$, where $z^i, 1 \leq i \leq n$ is a zone in the image I and n the total number of zones. Pixels inside the zones in Z are transformed by the feature extraction operator in the representation $F = \{f^1, \ldots, f^n\}$, where $f^i, 1 \leq i \leq n$ is the partial feature vector extracted from z^i. At the end of the optimization process, the optimization algorithm has explored the representation set $RS_{IFE} = \{F^1, \ldots, F^p\}$ (for MOGAs, RS_{IFE} is the optimal set at the last generation).

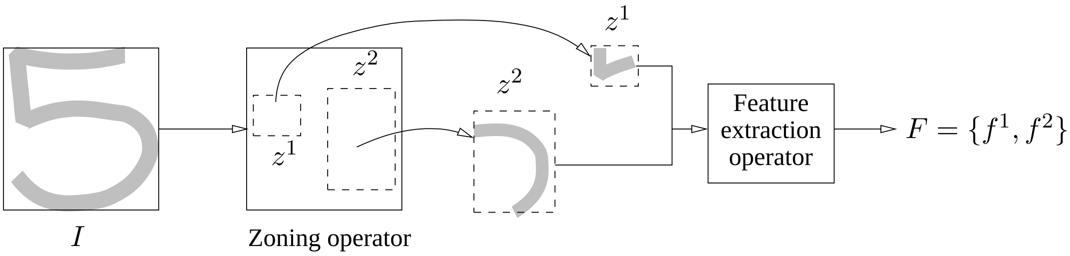
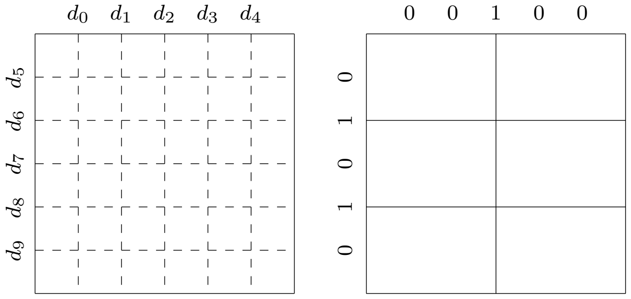


Figure 2. IFE structure.

The result set RS_{IFE} is used to train the classifier set $K = \{K^1, \ldots, K^p\}$, where K^i is the classifier trained with representation F^i. The first hypothesis is to select the most accurate classifier $SI, SI \in K$ for a single classifier system. The second hypothesis is to use K to optimize an EoC for higher accuracy, an approach discussed in Section 2.2. The remainder of this section discusses the IFE operators chosen for experimentation with isolated handwritten digits and the candidate solution evaluation.

2.1.1. Zoning Operator

To compare performance to the traditional human aproach, a *baseline* representation with a high degree of accuracy on handwritten digits with a *multi-layer Perceptron* (MLP) classifier [Oliveira et al. 2002] is considered. Its zoning strategy, detailed in Fig. 3.b, is defined as a set of three image dividers, producing 6 zones. The *divider zoning operator* expands the baseline zoning concept into a set of 5 horizontal and 5 vertical dividers that can be either *active* or *inactive*, producing zoning strategies with 1 to 36 zones. Fig. 3.a details the operator template, encoded by a 10-bit binary string. Each bit is associated with a divider's state (1 for active, 0 for inactive).



(a) (b)

Figure 3. Divider zoning operator (a). The baseline representation in (b) is obtained by setting only d_2, d_6 and d_8 as active.

2.1.2. Feature Extraction Operator

Oliveira *et al.* used and detailed in [Oliveira et al. 2002] a mixture of concavities, contour directions and black pixel surface transformations, extracting 22 features per zone (13 for concavities, 8 for contour directions and 1 for surface). To allow a direct comparison between IFE and the baseline representation, the same feature transformations (the domain knowledge) are used to assess the IFE.

2.1.3. Candidate Solution Evaluation

Candidate solutions are evaluated with respect to their classification accuracy. Thus, the objective is to minimize the classification error rate on the *optimization* data set (the domain context). To compare optimization methods, candidate solutions are evaluated with the *projection distance* (PD) classifier [Kimura et al. 1998].

2.2. EoC Optimization

A recent trend in PR has been to combine several classifiers to improve their overall performance. Algorithms for creating EoCs will usually fall into one of two main categories. They either manipulate the training samples for each classifier in the ensemble (like Bagging and Boosting), or they manipulate the feature set used to train classifiers [Kuncheva and Jain 2000]. The key issue is to generate a set of diverse and fairly accurate classifiers for aggregation [Kittler et al. 1998].

We create EoCs on a two-level process. The first level creates a classifier set K with IFE, and the second level optimizes the classifiers aggregated. We assume that RS_{IFE} generates a set K of p diverse and fairly accurate classifiers. To realize this task, the classifiers in K are associated with a binary string E of p bits, which is optimized to select the best combination of classifiers using an optimization algorithm. The classifier K^i is associated with the i^{th} binary value in E, which indicates whether or not the classifier is active in the EoC.

The optimization process minimizes the EoC classification error on the *optimization* data set. This is supported by [Ruta and Gabrys 2005]. Evaluating the EoC error rate requires actual classifier aggregation. PD classifiers are aggregated by majority voting, and votes are calculated once and stored in memory to speed up the opimization process.

3. Optimization Algorithm

A local search algorithm is used to optimize the IFE and EoC. The algorithm chosen is the *Record-to-Record Travel* (RRT) algorithm [Pepper et al. 2002], an annealing based heuristic. The RRT algorithm improves an initial solution i by searching in its neighborhood for better solutions based on their evaluation (classification error rate). The RRT algorithmn, detailed in Algorithm 1, produces after a number of iterations the record solution r. The algorithm is similar to a hill climbing aproach, but avoids local optimum solutions by allowing the search towards non-optimal solutions with a fixed deviation D. Earlier experiments indicated that the RRT algorithm over-fitted solutions during the optimization process. The global validation strategy discussed in [Radtke et al. 2006b] is used to avoid this effect, and Algorithm 1 includes support for this strategy.

Given the inicial solution i, the algorithm will copy it to the record solution r and store its evaluation value in $RECORD$. It also copies i as the current solution p. Next it will repeat the following process during a number of iterations, until the current solution is worse than the record solution plus the allowed deviation. First it will find the set P, solutions neighbor to p, and select the best neighbor $p', p' \in P$. To avoid cyclic optimization, solutions already evaluated are not considered for p'. If evaluating p' yields results within the allowed deviation, it is copied as p for the next iteration. Solution p' replaces the record solution r only if it yields better results. If p' is worse than the allowed deviation, the optimization process stops. The explored solution set S is responsible to store solutions tested by the RRT algorithm for the global validation strategy. At each iteration, the algorithm inserts into S the solutions in the neighbor set P. At the end of the optimization process, solutions in S are validated and the most accurate solution is selected. For the IFE process, S is the result set RS_{IFE} used to create the classifier set K.

Neighbors to solution X^i are created by swapping bits in the binary string with their complement. For a binary string E with p bits, a set of p neighbors is created by complementing each bit $i, 1 \leq i \leq p$ on solution E^i. For the IFE, solution in Fig. 4.a has solutions in Figs. 4.b and 4.c as two possible neighbors.

4. Experimental Protocol and Results

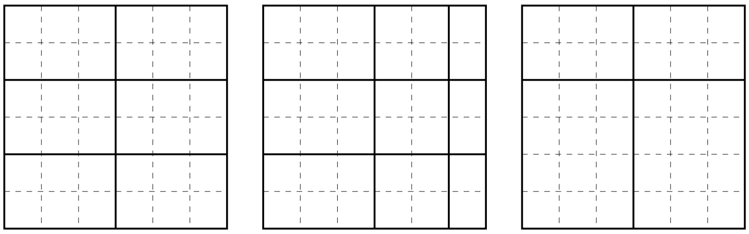
The tests are performed as in Fig. 1. The IFE methodology is solved to obtain the representation set RS_{IFE}, which is used to train the classifier sets K . For a single classifier system, the most accurate classifier $SI, SI \in K$ is selected. EoCs are then created

```
Data: Initial solution $i$
Data: Deviation $D$
Result: Record solution $r$
Result: Explored solution set $S$
$r = i$;
$RECORD = eval(r)$;
$p = i$;
$S = \emptyset$;
repeat
    Create the solution set $P$, neighbor to $p$;
    $S = S \cup P$
    Select the best solution $p' \in P$ such as that $p'$ has not yet been evaluated;
    if $eval(p') < RECORD + RECORD \times D$ then
        $p = p'$;
        if $eval(p') < RECORD$ then
            $RECORD = eval(p')$;
            $r = p'$;
        end
    end
until $eval(p) <= RECORD + RECORD \times D$ ;
```

Algorithm 1: Modified record to record travel (RRT) algorithm used to optimize classification systems with global validation.

with K, producing SE. To select resulting solutions, we use the global validation approach detailed in [Radtke et al. 2006b]. Solutions obtained are compared to the baseline representation defined in [Oliveira et al. 2002] and to solutions obtained with MOGAs in [Radtke et al. 2006a]. Unlike MOGAs which may produce different solutions on each run, the RRT algorithm will yield the same result set S for the same initial solution i. Thus, solutions obtained with the RRT are compared to both average results in [Radtke et al. 2006a] and to the best result obtained in 30 runs.

The data sets in Table 1 are used in the experiments – isolated handwritten digits from NIST-SD19. Classifier training is performed with the *training* data set. The *validation* data set is used to adjust the classifier parameters (PD hyper planes). The optimization process is performed with the *optimization* data set, and the *selection* data set is



(a) (b) (c)

Figure 4. Zoning strategy (a) and two neighbors (b and c) using the proposed divider zoning operator.

used with the global validation strategy to select solutions.

Data set	Size	Origin	Sample range
training	50000	hsf_0123	1 to 50000
validation	15000	hsf_0123	150001 to 165000
optimization	15000	hsf_0123	165001 to 180000
selection	15000	hsf_0123	180001 to 195000
$test_a$	60089	hsf_7	1 to 60089
$test_b$	58646	hsf_4	1 to 58646

Table 1. Handwritten digits data sets extracted from NIST-SD19.

Both the IFE and EoC have initial solutions associated to empty strings. Thus, there are no dividers active in the initial IFE solution, and no classifiers associated to the initial EoC. The deviation D is set empirically to $D = 5\%$. The RRT is a deterministic algorithm, hence a single run is performed with both processes. All experiments were performed on a Athlon64 3000+ processor with 1GB of RAM memory.

Results obtained are detailed in Table 4, where *Z* is the solution zone number, $|S|$ is the solution cardinality (either feature number or classifier number), e_{test_a} and e_{test_b} are classification error rates on $test_a$ and $test_b$. Solutions SI_M and SE_M where obtained with MOGAs, and the baseline representation is defined in [Oliveira et al. 2002]. These solutions are included for comparison purposes.

To compare solutions, the following procedure is used. The baseline representation is compared directly with solutions SI and SE. Moga solutions are observations with 30 samples. Thus, we calculate the confidence interval lower and upper values with $\alpha = 0.05$ (95% of confidence) for MOGA solutions. One solution is said comparable to a MOGA solution only if its error rate is within the confidence interval. Otherwise, the solution may be better if it is bellow the confidence interval, and worse if it is above.

Solution	Z	$\|S\|$	e_{test_a}	e_{test_b}
Baseline	6	132	2.96%	6.83%
SI_M	15	330	(2.18%) 2.18% (2.18%)	(5.47%) 5.47% (5.47%)
SI	15	330	2.18%	5.47%
SE_M	–	24.67	(2.00%) 2.02% (2.06%)	(5.14%) 5.19% (5.22%)
SE	–	23	2.05%	5.20%

Table 2. IFE and EoC optimization results – mean values on 30 MOGA replications (confidence interval lower and upper values in parenthesis) and actual error rates for remaining solutions.

Solutions SI and SE obtained with the RRT algorithm outperform the baseline representation defined by the human expert. Figure 5 details the zoning strategy associated to SI and SI_M. Comparing these solutions to SI_M and SE_M, we conclude that the RRT performed similarly to an MOGA. Solution SI obtained by the RRT has the same zoning strategy as SI_M, and the error rate for SE is comparable to SE_M. These results indicate that the RRT algorithm is effective to optimize classification systems.

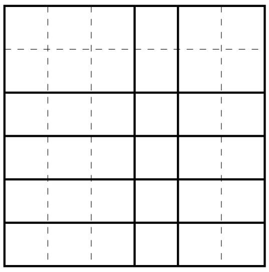


Figure 5. Zoning strategy associated to SI and SI_M.

The RRT algorithm had a smaller computational cost to solve these problems. To optimize the IFE a total of 76 candidate solutions were evaluated by the RRT, whereas the MOGA evaluates 64 candidate solutions only on its initial population in [Radtke et al. 2006a] to optimize the same problem. The same is observed with EoCs, the RRT evaluated a total of 14000 solutions to optimize the EoC, and the MOGA evaluated 166000 solutions thoughout the same optimization process.

5. Discussion

Solutions obtained with the RRT are comparable to solutions obtained with MOGAs. The advantage of the RRT algorithm is the lower computational burden to optimize the classification system discussed. The results obtained indicate that the RRT algorithm is a more appropriate choice to optimize classification systems with the proposed approach.

Solutions obtained with the RRT algorithm were also over-fitted to the *optimization* data set. The global validation strategy detailed in [Radtke et al. 2006b] selected better results in S than simply selecting the record solution r obtained at the end of the optimization process. This reinforces the conclusion that the optimization of classification systems using wrapped classifiers is prone to solution over-fit.

We will pursue two directions to further develop this research. The first is to optimize the IFE and EoC using MLP classifiers, a more discriminant classifier. The experiments in [Radtke et al. 2006a] used the PD as a meta-classifier to select MLP classifier. With the current hardware, optimizing classification systems using an actual MLP classifier with MOGAs is unfeasible for a large data set. Thus, the interest on using the RRT algorithm for this task. The second direction is to compare algorithmic performance with other classification problems, such as handwritten letters, a different domain, or feature subset selection.

Acknowledgments

The first author would like to acknowledge the CAPES and the Brazilian government for supporting part of this research through scholarship grant BEX 2234/03-3. The first author also would like to acknowledge the Pontifícia Universidade Catolica do Parana (PUCPR, Brazil) for supporting this research. The other authors would like to acknowledge the NSERC (Canada) for supporting this research.

References

Handl, J. and Knowles, J. (2006). Feature subset selection in unsupervised learning via multiobjective optimization. *International Journal on Computational Intelligence Research*, 3(1):217–238.

Kimura, F., Inoue, S., Wakabayashi, T., Tsuruoka, S., and Miyake, Y. (1998). Handwritten Numeral Recognition using Autoassociative Neural Networks. In *Proceedings of the International Conference on Pattern Recognition*, pages 152–155.

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

Kuncheva, L. I. and Jain, L. C. (2000). Design classifier fusion systems by genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4(4):327–336.

Li, Z.-C. and Suen, C. Y. (2000). The partition-combination method for recognition of handwritten characters. *Pattern Recognition Letters*, 21(8):701–720.

Oliveira, L. S., Sabourin, R., Bortolozzi, F., and Suen, C. Y. (2002). Automatic Recognition of Handwritten Numerical Strings: A Recognition and Verification Strategy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(11):1438–1454.

Pepper, J. W., Golden, B. L., and Wasil, E. A. (2002). Solving the traveling salesman problem with annealing-based heuristics: A computational study. *IEEE Trans. on Systems, Mand and Cybernetics – Part A: Systems and Humans*, 32(1):72–77.

Radtke, P. V. W., Wong, T., and Sabourin, R. (2006a). Classification system optimization with multi-objective genetic algorithms. In *Proceedings of the 10th International Workshop on Frontiers in Handwriten Recognition (IWFHR 2006)*, pages 331–336. IAPR.

Radtke, P. V. W., Wong, T., and Sabourin, R. (2006b). An evaluation of over-fit control strategies for multi-objective evolutionary optimization. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2006)*, pages 6359–6366. IEEE Computer Society.

Ruta, D. and Gabrys, B. (2005). Classifier Selection for Majority Voting. *Information fusion*, 6:63–81.

Tremblay, G., Sabourin, R., and Maupin, P. (2004). Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm. In *17th International Conference on Pattern Recognition – ICPR2004*, pages 208–211, Cambridge, U.K. IEEE Computer Society.

Tsymbal, A., Pechenizkiy, M., and Cunningham, P. (2005). Sequential genetic search for ensemble feature selection. In *Proceddings of International Joint Conference on Artificial Intelligence*, pages 877–882.