Using the RRT Algorithm to Optimize Classification Systems for Handwritten Digits and Letters

Paulo V. W. Radtke
Pontifícia Universidade Católica do Paraná
R. Imaculada Conceição, 1155
Curitiba, PR, Brazil
paulo.radtke@pucpr.br

Robert Sabourin
École de Technologie Supérieure
1100, rue Notre-Dame ouest
Montréal, QC, Canada
robert.sabourin@etsmtl.ca

Tony Wong
École de Technologie Supérieure
1100, rue Notre-Dame ouest
Montréal, QC, Canada
tony.wong@etsmtl.ca

ABSTRACT

Multi-objective genetic algorithms have been often used to optimize classification systems, but little is discussed on their computational cost to solve such problems. This paper optimizes a classification system with an annealing based approach, the Record-to-Record Travel algorithm. Results obtained are compared to those obtained with a multi-objective genetic algorithm in the same approach. Experiments are performed with isolated handwritten digits and uppercase letters, demonstrating both the effectiveness and lower computational cost of the annealing based approach.

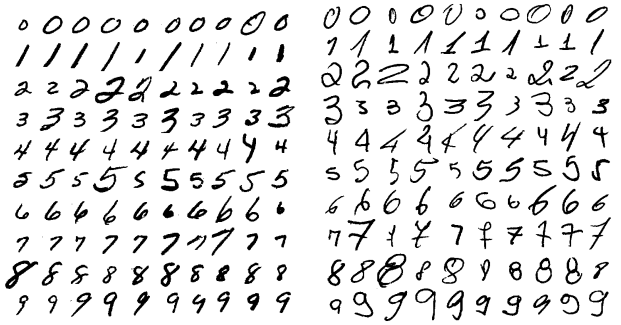
Keywords

local search, RRT algorithm, classification systems, ensemble of classifiers

1. INTRODUCTION

Image pixel information is traditionally transformed by a feature extraction process prior to classification. A process performed to reduce data complexity and select the most relevant information from images. Zoning is often used to improve the features discriminatory power [1, 6], instead of using the whole image for classification. Both feature extraction and zoning have important roles in the classification stage, but are defined on a trial and error basis by an human expert. Associated to this burden, one classification system is adapted to a specific domain. Figure 1 details the difficulties faced by changing the handwriting style. Thus, the same classification system can not be used on another problem with the same reliability, unless the classification system is properly adapted to the new context.

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 expert's domain knowledge and information from the domain context – actual images. This paper discusses the two-level approach to optimize classification systems in Fig. 2. The first level employs the *Intelligent Feature Extraction* (IFE) methodology to extract feature sets. These feature sets are then used on the second level to optimize an *ensemble of classifiers* (EoC) to improve classification accuracy.



(a) (b)

Figure 1: Handwritten digits: (a) NIST SD-19 and (b) Brazilian checks.

The literature demonstrates that genetic based approaches are frequently used to optimize classification systems [2, 5, 12, 13], specially *multi-objective genetic algorithms* (MOGAs). It is now understood that the advantage of MOGAs lies in the inherent diversity of the optimized solution set, avoiding the population convergence to a single local optimum. However, population based approaches evaluate a large number of candidate solutions. When using a wrapper approach, training and testing solutions takes a considerable time. Hence, the use of other algorithms may provide comparable solutions associated to a lower computational burden. The algorithm chosen for a comparative study is the *Record-to-Record Travel* (RRT) algorithm [8], 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 [9] using MOGAs. The new contribution is to investigate the 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 Section 5 the results obtained. Finally, Section 6 discusses the goals attained and future works.

SAC'08 March 16-20, 2008, Fortaleza, Ceará, Brazil

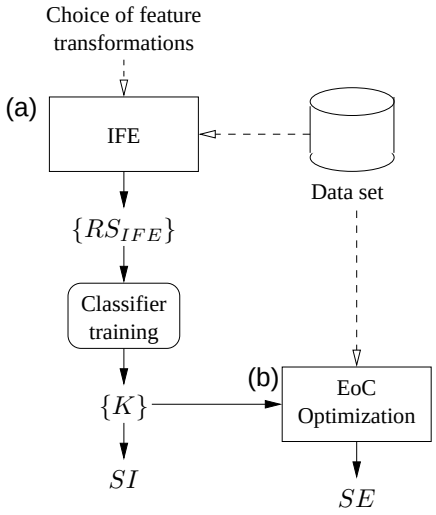


Figure 2: Classification system optimization approach.

2. CLASSIFICATION SYSTEMS OPTIMIZATION

Classification systems are modeled in a two-level process (Fig. 2). The first level uses the IFE methodology to obtain the representation set RS_{IFE} (Fig. 2.a) – feature sets. 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. 2.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. 3. 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).

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 [7] is considered. This baseline representation was defined on a traditional trial and error basis. Its zoning strategy, detailed in Fig. 4.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. Figure 4.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).

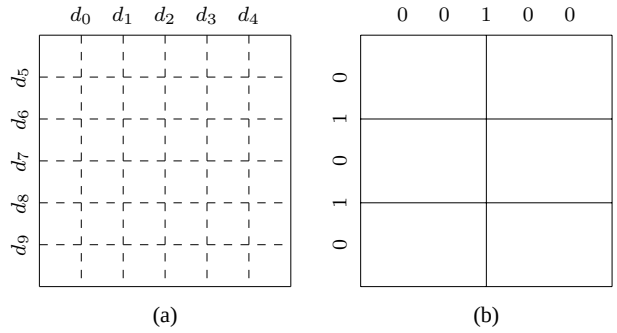


Figure 4: Divider zoning operator (a). The baseline representation in (b) is obtained by setting only d_2, d_6 and d_8 as active.

2.1.2 Feature Extraction Operator

The classification system in [7] used and detailed 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 [3].

2.2 EoC Optimization

Several classifiers may be combined 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 [5]. The key issue is to generate a set of diverse and fairly accurate classifiers for aggregation [4].

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

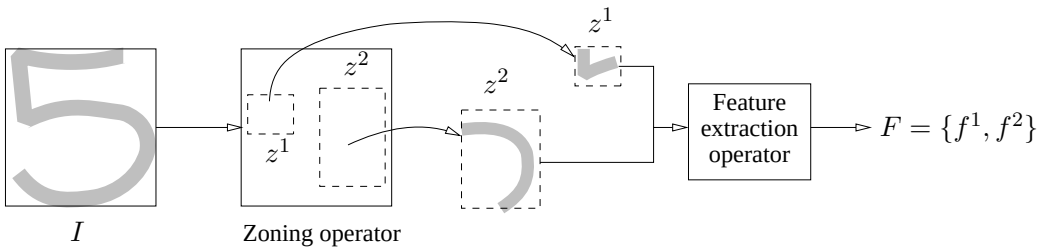


Figure 3: IFE structure.

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 [11]. 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

The *Record-to-Record Travel* (RRT) algorithm [8], is an annealing based heuristic. It is said to be a local search algorithm, searching for new solutions in the vicinity of the current solution. 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 algorithm, detailed in Algorithm 1, produces after a number of iterations the record solution r. The algorithm is similar to a hill climbing approach, 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 [10] is used to avoid this effect, and Algorithm 1 includes support for this strategy.

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.

Given the initial 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. 5.a has solutions in Figs. 5.b and 5.c as two possible neighbors.

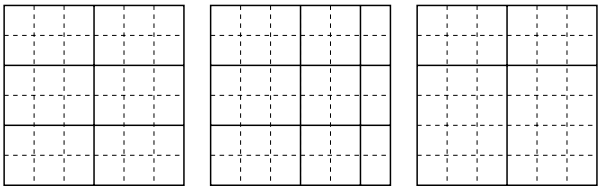


Figure 5: Zoning strategy (a) and two neighbors (b and c) using the proposed divider zoning operator.

4. EXPERIMENTAL PROTOCOL

The tests are performed as in Fig. 2. The IFE methodology is solved to obtain the representation set RS_{IFE}, which is used to train the classifier set K . For a single classifier system, the most accurate classifier $SI, SI \in K$ is selected. EoCs are then created with K, producing the ensemble SE. To select the most accurate solution in the result set S, we use the global validation approach detailed in [10]. The full process is performed for both digits and uppercase letters. Solutions obtained are compared to the baseline representation defined in [7] and to solutions obtained with MOGAs in [9]. 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 [9] and to the best result obtained in 30 runs.

The data sets in Tables 1 and 2 are used in the experiments – isolated handwritten digits and letters from NIST SD-19. 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 used with the global validation strategy to select solutions. The *test* databases are used to compare solutions, where $test_b$ (handwritten digits) is known to require a robust classifier for higher accuracies.

Table 1: Handwritten digits data sets extracted from NIST SD-19.

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 2: Handwritten uppercase letters data sets extracted from NIST SD-19.

Data set	Size	Origin	Sample range
training	43160	hsf_0123	1 to 43160
validation	3980	hsf_4	1 to 3980
optimization	3980	hsf_4	3981 to 7960
selection	3980	hsf_4	7961 to 11940
$test$	12092	hsf_7	1 to 12092

Both the IFE and EoC have initial solutions associated to empty strings. Thus, there are no active dividers 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 RRT experiments were performed on a Athlon64 3000+ processor with 1GB of RAM memory.

Solutions are compared as follows. The baseline representation is compared directly with solutions SI and SE obtained with the RRT algorithm. 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.

5. RESULTS

Handwritten digits results are detailed in Table 3, 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 are the best results obtained with MOGAs in 30 replications, whereas $\overline{SI_M}$ and $\overline{SE_M}$ are average values for the 30 replications. The baseline representation is included for comparison purposes.

Table 3: Digits optimization results – confidence interval lower and upper values in parenthesis for average values.

Solution	Z	$\|S\|$	e_{test_a}	e_{test_b}
Baseline	6	132	2.96%	6.83%
$\overline{SI_M}$	15	330	(2.18%) 2.18% (2.18%)	(5.47%) 5.47% (5.47%)
SI_M	15	330	2.18%	5.47%
SI	15	330	2.18%	5.47%
$\overline{SE_M}$	–	24.67	(2.00%) 2.02% (2.06%)	(5.14%) 5.19% (5.22%)
SE_M	–	23	1.96%	5.06%
SE	–	23	2.05%	5.20%

Solutions SI and SE obtained with the RRT algorithm outperform the baseline representation defined by the human expert. Figure 6.a details the zoning strategy associated to SI and SI_M. Comparing solutions SI and SE to solutions obtained with the same approach with an MOGA, we conclude that the RRT had a similar performance. Solution SI obtained by the RRT has the same zoning strategy as SI_M (the same on 30 replications), and the error rate for SE is comparable to $\overline{SE_M}$ (error rate is within the confidence interval).

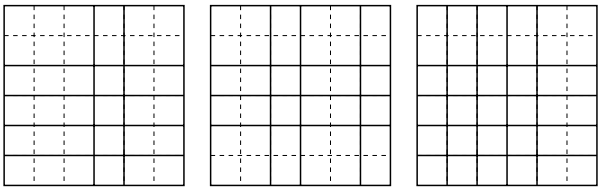


Figure 6: Zoning strategies for handwritten digits (a) and uppercase letters – MOGA (b) and RRT (c).

The same experiments are performed with uppercase letters, a more complex problem, and results are detailed in Table 4. Zoning strategies for SI_M and SI in Table 4 are presented in Figs. 6.b and 6.c respectively. The RRT algorithm outperforms again the human based approach, confirming the approach to optimize classification systems. Comparing the IFE and EoC solutions performance, we observe that solutions are similar, and no significant out performance is verified – solution SE_M in Table 4 is an outlier. Solution cardinality is smaller when using MOGAs, owing to the multi-objective approach used that emphasized both accuracy and smaller cardinality.

Whereas performance results indicate a tie between the RRT and MOGA approaches, the computational cost is significantly smaller when using the RRT. Table 5 details the solutions evaluated to solve each problem with both algorithm types. To optimize solution SI_M, the MOGA was actually modified to memorize solutions, so that they were evaluated only once. This was faster than training the PD classifier for the same representation multiple times – each representation required about 30 minutes on a Beowulf cluster. Solution SE_M was optimized with a traditional MOGA, as the solution evaluation procedure was very fast.

Table 4: Letters optimization results – confidence interval lower and upper values in parenthesis for average values.

Solution	Z	$\|S\|$	e_{test}
Baseline	6	132	9.20%
$\overline{SI_M}$	16	352	(7.19%) 7.19% (7.19%)
SI_M	16	352	7.19%
SI	25	550	7.14%
$\overline{SE_M}$	–	14.41	(6.33%) 6.43% (6.54%)
SE_M	–	12	6.22%
SE	–	29	6.53%

Table 5: Solutions evaluated through the IFE and EoC optimization process.

Result	Digits	Letters
SI_M	450	623
SI	76	100
SE_M	166000	166000
SE	13580	33067

6. DISCUSSION

The RRT algorithm produced solutions comparable to traditional MOGAs when optimizing classification systems in two different problems. The computational burden was significantly lower when using the RRT algorithm. The higher burden is explained by the global approach used by the MOGA to optimize solutions. Thus, the RRT is more appropriate to optimize classification systems when processing time is a concern.

The discussed semi-automatic approach to optimize classification systems outperformed the human based approach. The zoning strategy was adapted to both optimization problems using the same feature set on each zone. The different zoning strategies found also justifies the claim that classification systems have to be adapted to the problem domain.

Solutions obtained with the RRT algorithm were also over-fitted to the *optimization* data set. The global validation strategy detailed in [10] selected better results in S than simply selecting the record solution r obtained at the end of the optimization process. This reinforces the conclusion in [10] that the optimization of classification systems using wrapped classifiers is prone to solution over-fit.

The higher computational burden associated to MOGAS made it infeasible to optimize classification systems with more discriminant classifiers, such as MLP or SVM classifiers. Such classifiers take more time than the PD classifier to train with large data sets. Thus, a future research direction is to optimize classification systems using a wrapped MLP classifier with the RRT algorithm.

7. 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 Católica do Paraná (PUCPR, Brazil) for supporting this research. The other authors would like to acknowledge the NSERC (Canada) for supporting this research.

8. REFERENCES

[1] V. di Lecce, G. Dimauro, S. Impedovo, G. Pirlo, and A. Salzo. Zoning Design for Hand-Written Numeral Recognition. In *Proceedings of the Seventh international Workshop on Frontiers in Handwriting Recognition – IWFHR-7*, pages 583–588, Amsterdan, 2000. Nijmegenl: International Unipen Foundation.

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

[3] 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.

[4] 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.

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

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

[7] 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.

[8] 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, Mand and Cybernetics – Part A: Systems and Humans*, 32(1):72–77, 2002.

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

[10] P. V. W. Radtke, T. Wong, and R. Sabourin. 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, 2006.

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

[12] G. Tremblay, R. Sabourin, and P. Maupin. 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., August 2004. IEEE Computer Society.

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