Overfitting in the Selection of Classifier Ensembles: a Comparative Study Between PSO and GA

Eulanda M. Dos Santos
Ecole de technologie superieure
Montreal, Canada
eulanda@livia.etsmtl.ca

Luiz S. Oliveira
Pontifical Catholic University of Parana
Curitiba, Brazil
soares@ppgia.pucpr.br

Robert Sabourin
Ecole de technologie superieure
Montreal, Canada
Robert.Sabourin@etsmtl.ca

Patrick Maupin
Defence Research and Development Canada
Val-Bélair, Canada
Patrick.Maupin@drdc-rddc.gc.ca

ABSTRACT

Classifier ensemble selection may be formulated as a learning task since the search algorithm operates by minimizing/maximizing the objective function. As a consequence, the selection process may be prone to overfitting. The objectives of this paper are: (1) to show how overfitting can be detected when the selection is performed by two classical search algorithms: Genetic Algorithm and Particle Swarm Optimization; and (2) to verify which algorithm is more prone to overfitting. The experimental results demonstrate that GA appears to be more affected by overfitting.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms

Keywords

Classifier ensembles selection, overfitting, GA, PSO

1. INTRODUCTION

Given a pool of classifiers $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$ generated using any ensemble creation method, the objective of the classifier ensemble selection method, called *overproduce-and-choose strategy* (OCS), is to find the most relevant subset of classifiers, based on the assumption that classifiers in $\mathcal{C}$ are redundant. Several search algorithms have been applied in the literature for OCS. *Genetic algorithms* (GAs) are attractive since they allow the fairly easy implementation of OCS as optimization processes. However, it has been shown that such stochastic search algorithms, when used in conjunction with Machine Learning techniques, are prone to overfitting.

The objective of this paper is to present the results of a comparative study between Particle Swarm Optimization (PSO) and GA in order to show how these algorithms are affected by overfitting and which one is more prone to overfitting. The experiments were conducted using the minimization of the error rate (ϵ) as the objective function. This comparative study is focused on detecting and controlling overfitting in GA and PSO by applying the global validation strategy (GV) [2] as overfitting control method.

Copyright is held by the author/owner(s).

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

2. GLOBAL VALIDATION (GV)

Let C_j^* and $C_j^{*'}$ be the best performing candidate ensembles found through calculating ϵ for each element of $\mathcal{P}(\mathcal{C})$ over samples contained in an optimization $\mathcal{O}$ and in a validation $\mathcal{V}$ sets respectively. $\mathcal{P}(\mathcal{C})$ is the powerset of $\mathcal{C}$ defining the population of all possible candidate ensembles C_j. Consider the classification error ϵ of these two candidate ensembles measured using samples from $\mathcal{V}$. We will denote this classification error by $\epsilon(\mathcal{V}, C_j^*)$ and $\epsilon(\mathcal{V}, C_j^{*'})$. In this setting, C_j^* is said to overfit on $\mathcal{O}$ if an alternative candidate ensemble $C_j^{*'} \in \mathcal{P}(\mathcal{C})$ can be found such that $\epsilon(\mathcal{V}, C_j^*) > \epsilon(\mathcal{V}, C_j^{*'})$.

GV relies on using $\mathcal{V}$ to create an auxiliary archive $\mathcal{A}$ to store in it the solution $C_j^{*'}$ found before overfitting starts to occur. The idea is to use ϵ measured on $\mathcal{V}$ as an estimation of the generalization error. To do this, it is sufficient to validate all solutions at each generation/iteration (g), find out the best solution $C_j^{*'}(g)$ and compare it to $C_j^{*'}$ stored in $\mathcal{A}$. When $C_j^{*'}(g)$ is better than $C_j^{*'}$, $\mathcal{A}$ is updated. In this way we keep the real $C_j^{*'}$ stored in $\mathcal{A}$.

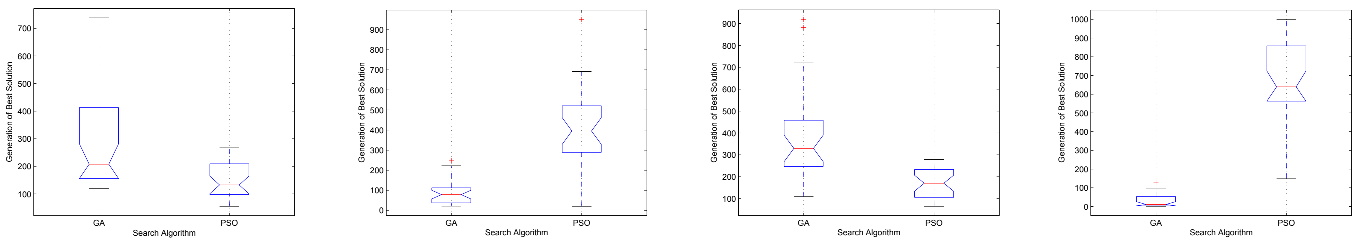
3. EXPERIMENTS

We used in our experiments kNN as the base classifier and we set $k = 1$ without fine-tuning this parameter in order to avoid additional experiments. The random subspace method (RSS) was used to generate one pool of 100 homogeneous classifiers. The size of the subsets of features used by RSS is shown in Table 1. Moreover, in order to provide the same experimental conditions for GA and PSO, we set: number of individuals/particles=128 and maximum number of generations/iterations=$1,000$. The optimization processes were conducted by GA and PSO using binary vectors.

Table 1 shows that two databases were used: NIST digits Special Database 19 (NIST SD19), called NIST-digits here, and the NIST SD19 handwritten uppercase letters, called NIST-letters. We employ the representation proposed in [1]. There are two test sets from NIST-digits, called here test1

Table 1: Specifications of the data sets used in the experiments.

Data set	# of features	Training	Optimization	Validation	Test	Features RSS	Pool size
NIST-digits	132	5,000	10,000	10,000	test1 60,089 test2 58,646	32	100
NIST-letters	132	43,160	3,980	7,960	12,092	32	100



(a) Optimization-Digits (b) Validation-Digits (c) Optimization-Letters (d) Validation-Letters

Figure 1: The convergence points of GA and PSO obtained on 30 replications. The generation when the best solution was found on optimization (1(a) and 1(c)) and on validation (1(b) and 1(d)).

Table 2: Mean values of the error rates obtained on 30 replications comparing GA and PSO.

Dataset	GA		PSO	
	NV	GV	NV	GV
NIST-Digits test1	3.60	**3.55**	3.62	3.61
NIST-Digits test2	7.91	7.80	8.02	7.88
NIST-letters	6.49	6.44	6.54	6.49

and test2. In addition, the databases were divided into four datasets: training, optimization, validation and test.

3.1 Results

All the experiments were replicated 30 times and the results were tested on multiple comparisons using the Kruskal-Wallis nonparametric statistical test with confidence level was 95% ($\alpha = 0.05$), and the Dunn-Sidak correction was applied to the critical values. Table 2 shows the comparative results. Values are shown in bold when GV decreased the generalization error significantly. The results with no overfitting control, denoted NV, are also included in this table.

These results show that, for NIST-digits, although both search algorithms are prone to overfitting, GA appears to be more affected. It is important to note that this behavior confirms Reunamen's work [3]. He pointed out that the degree of overfitting increases as the intensity of the search increases. Figure 1(a) shows that GA needs more generations than PSO to find C_j^* on $\mathcal{O}$ during the optimization process. On the other hand, the solution $C_j^{*'}$ stored in $\mathcal{A}$ is found much earlier using GA than PSO (Figure 1(b)). This result can be explained by the fact that PSO keeps searching for solutions even worse than the particle with the best fitness ($gBest$), during the optimization process. Thus, it is possible to generate solutions that are better than $gBest$ in generalization. For NIST-letters, both search algorithms appear to be equally affected by overfitting. Again, the results confirm that GA conducts a more intense search process, since GA needs more generations to find C_j^* (Figure 1(c)) while it finds $C_j^{*'}$ much earlier than PSO (Figure 1(d)).

4. CONCLUSIONS

This paper presented the experimental results of a study comparing GA and PSO in terms of overfitting, when both search algorithms are applied in classifier ensemble selection. The results showed that, although overfitting can be detected in both search algorithms, GA appears to be more prone to overfitting. Our results indicated that the possible reason for this is that GA performs a more intense search process, i.e. it needs more generations to find the best solution during the optimization process. These results confirm the literature in that the degree of overfitting increases as the intensity of search increases.

5. ACKNOWLEDGMENTS

This work was supported by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), Brazil, and Defence Research and Development Canada, DRDC-Valcartier under the contract W7701-2-4425.

6. REFERENCES

[1] L. Oliveira, R. Sabourin, F. Bortolozzi, and C. Suen. Automatic recognition of handwritten numerical strings: A recognition and verification strategy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1438–1454, 2002.

[2] P. Radtke, R. Sabourin, and T. Wong. Impact of solution over-fit on evolutionary multi-objective optimization for classification systems. In *Proceedings of the IEEE International Joint Conference on Neural Network*, Vancouver, Canada, 2006.

[3] J. Reunanen. Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, 3:1371–1382, 2003.