ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

I/S-Race: An iterative Multi-Objective Racing Algorithm for the SVM Parameter Selection Problem

Péricles B.C. Miranda[1,2], Ricardo M. Silva[1] and Ricardo B. C. Prudêncio[1]

1- Centro de Informática - Universidade Federal de Pernambuco - Brazil
2- DEINFO - Universidade Federal Rural de Pernambuco - Brazil

Abstract. Finding appropriate values for the parameters of an algorithm is an important and time consuming task. Recent studies have shown that racing algorithms can effectively handle this task. This paper presents a multi-objective racing algorithm called iterative S-Race (I/S-Race), which efficiently addresses multi-objective model selection problems in the sense of Pareto optimality. We evaluate the I/S-Race for selecting parameters of SVMs, considering 20 widely-used classification datasets. The results revealed that the I/S-Race is an efficient and effective algorithm for automatic model selection, when compared to a brute-force multi-objective selection approach and the S-Race algorithm.

1 Introduction

A great number of algorithms needs adequate parameters to perform well [1]. Many researchers have faced the parameter selection as an optimization problem and a variety number of optimization algorithms have been used [6, 5]. Besides the cited algorithms, we highlight the Racing Algorithms (RAs)[1]. RAs are iterative procedures that start with a pool of candidate models. At each iteration, the candidates are evaluated considering an objective function. If a statistical evidence is reached, the candidates that were outperformed are eliminated from the race.

Although parameter selection often involves more than one objective [1], most researches used single objective RAs to refine candidate models. We highlight that there is only one RA for Multi-Objective Model Selection (MORA), called S-Race [7]. The S-Race demonstrated to be an efficient and effective algorithm for model selection, when compared to a brute-force selection approach. The main drawback of S-Race is that it can become computationally prohibitive when the number of models to start the race is large.

This work presents a novel MORA called I/S-Race which intends to avoid the S-Race's weakness and be able to: 1) make the tuning tasks suitable with a very large number of initial candidate parameter settings and, 2) allow a significant reduction of the number of function evaluations without any major loss in solution quality. We evaluated the proposed modification for the parameter selection problem considering 20 classification problems, and we show the effectiveness of our proposal. In Section 2 introduces the S-Race algorithm. Section 3 details the developed work. Section 4 describes the experiments and obtained results. Finally, Section 5 presents some conclusions and the future work.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

2 MO Racing Algorithms

The S-Race algorithm inspired the creation of the proposed work. All details of the algorithm will be presented to facilitate the understanding of the I/S-Race. The S-Race uses the same procedure of the RA [7], it considers a pool of m candidate models, $C^1, C^2, ..., C^m$, at the beginning of racing. At each racing' step, the performance of each candidate is evaluated on a batch (usually one or several test instances from the validation set) and it is represented as a performance vector, where each element represents an objective. After that, we perform a statistical test to determine whether the difference in performance between the candidates of the pool is significant. The statistical test adopted by the S-Race's authors to establish dominance between a pair of candidate models is the Sign Test (ST)[8]. Besides, in order to avoid false rejections and, therefore, useful models may be excluded from further consideration, S-Race employs multiple hypothesis testing of dominance relationships whenever warranted. However, the multiple hypothesis test can be overly conservative. Thus, a testing approach called Holm's Step-Down procedure was used to control the Family-Wise Error Rate (FWER) [9]. It guarantees that the FWER will not exceed a user-specified threshold. In addition, it makes no assumptions on the normality or independence of the test statistics involved [9].

Along the algorithm' steps, when the statistical evidence becomes significant, the outperformed candidates are excluded from the race. As we are considering a multi-objective scenario, the process of comparing a pair of candidates works by checking if the Pareto dominance between the candidates is statistically significant. The models to be excluded are those, whose performances are dominated by the corresponding performances of at least one other model in the racing pool. At the end of the race, a Pareto-front of candidates is generated and this Pareto may contain optimal solutions.

3 I/S-Race

In order to avoid the problems identified in S-Race, we created an iterative racing algorithm which follows three steps. First, construct a candidate solution based on some probability model; second, evaluate all candidates in the pool; third, update the probability model biasing the next sampling towards the better candidate solutions. These three steps are iterated, until some stopping criterion is satisfied. An outline of the I/S-Race algorithm is given in Algorithm 1.

Initially, a limited number of configurations is selected by using an uniform random sampling. Once we have an initial population, we apply the S-Race algorithm to determine the solutions that must continue in the race. At each step of the S-Race, the candidates are evaluated on a single instance. These solutions are called Elite configurations Θ^{elite}, and they are used to create a probabilistic model. So that, at each iteration, a small set of candidates is generated according to the model. In other words, the Θ^{elite} is used to update the model in order to bias the search around the high quality candidates. After each step, those

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

Algorithm .1: I/S-Race Pseudo-code.

```
1: Require: I = {i_1, i_2, ...}
        Parameter space: Θ,
        Fitness Function: ζ(C, i) ∈ ℝ,
        Θ_1 ∼ Sample(Θ)
        Θ^elite :=S-Race(Θ_1)
        j := 2
        count := 0
2: while count < |I| or |Pool| = 1 do
        Θ^new ∼ Sample(X, Θ^elite)
        Θ_j := Θ^new ∪ Θ^elite
        Θ^elite :=S-Race(Θ_j)
        j := j + 1
        count := count + 1
   end
   return Θ^elite
```

solutions that perform statistically worse than at least another one are discarded, and the race continues with the remaining surviving solutions. Once S-Race terminates, the candidates within the Θ^{elite} are then weighted according to their ranks. We use N^{elite} which denotes the number of candidates that survived the race. The weight of an elite configuration with rank $r_z(z = 1, ..., N^{elite})$ is given by:

$$w_z = \frac{N^{elite} - r_z + 1}{N^{elite}(N^{elite} + 1)/2}. \tag{1}$$

In next iteration, the $|\Theta_{k+1}| - N^{elite}$ new candidates are iteratively sampled around one of the elite configuration consider. To do so, for sampling each new candidate configuration, first one elite solution $E_z(z \in 1, ..., N^{elite})$ is chosen with a probability proportional to its weight w_z and next a value is sampled for each parameter. Here, we adopt, as sampling procedure, a d-variate normal distribution [2]. Aiming to center the search around the promising configurations into the Θ^{elite}, this distribution is defined on each configuration which survived from the previous iteration.

4 Experiments

In this section, we present the experiments which evaluated the I/S-Race for the SVM parameter problem. As a basis of comparison, we implemented the S-Race and we used as a baseline the BFA. All algorithms were evaluated on the set of 20 different numerical classification datasets available in [4].

The algorithms performed a search in a space represented by a discrete grid of SVM configurations. By following the guidelines provided in [3], γ can assume values from 2^{-15} to 2^{3} and the parameter C can assume values from 2^{-5} to 2^{15}. In this work, we decided to discretize the values of C and γ so that both parameters assume, each one, 500 values into the interval. Thus yielding $500 \times 500 = 250,000$ different combinations of values. In this work, the configurations were optimized in two aspects, success rate (SR) and the number of support

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

vectors (NSV). Obviously, greater SR and lower NSV are preferred in order to generate SVM models with high performance rate and low complexity. The Pareto dominance was defined considering these two objectives.

In the experiments, the S-Race was executed using a pool size with all the $250,000$ models. When the validation batches are exhausted or $\mid Pool \mid = 1$ the algorithm stops. On the other hand, the I/S-Race was executed using a pool size limited to 200 models and it used the same stop criterion presented in Algorithm 1. As it can be seen, the maximum number of candidate evaluations that the I/S-Race can perform is (200 models $\times 20$ S-Race executions) $\times$ 20 iterations $= 80,000$ evaluations, which represents, in the worst case, only 32% of exploration in the search space. Our goal by using a reduced number of models is to demonstrate that I/S-Race is able to identify the same Pareto-optimal models as a BFA and be more efficient than S-Race. In order to guarantee a fair simulation, the simulations were executed 30 times and the average results were recorded.

4.1 Performance Metrics

Let P_{BFA} be the final models obtained via a BFA. Similarly, let P_{SR} and P_{ISR} be the final models obtained via S-Race and I-S/Race respectively. Note that, in an ideal scenario, we would like to have $P_{ALG} = P_{BFA}$ at a much less computational cost than a BFA, where P_{ALG} is a general MORA (i.e. P_{SR} or P_{ISR}). In our experimental analysis, we considered two quantities to evaluate the quality of P_{ALG}. These were the retention R, $R = \mid P_{ALG} \cap P_{BFA} \mid / \mid P_{BFA} \mid$, and the excess E, $E = (1 - \mid P_{ALG} \cap P_{BFA} \mid) / \mid P_{ALG} \mid$.

As their names somewhat imply, R measures the algorithm's ability to retain models obtained via the BFA. On the other hand, E measures the algorithm's ability to correctly identify dominated models that do not appear in P_{BFA}. Optimally, the MORA should behave like the BFA and, therefore, exhibit $R = 1$ and $E = 0$. Apart from R and E, we also consider a measure T of I/S-Race's computational effort. It is defined as the ratio of total running time of the I/S-Race over the running time of the S-Race.

4.2 Results

In order to analyze our results adequately we performed statistical analysis. As the data does not follow a normal distribution, we applied the Wilcoxon test to verify our hypothesis: the I/S-Race algorithm is a competitive approach for the problem at hand. All the following analysis used this methodology.

In Figures 1 and 2 we show how the metrics R and E vary as the batches are executed during the race. First of all, in Figure 1 we can see that the I/S-Race has a low proportion of retention in the initial batches. It is because the I/S-Race starts with a sample of solution which represents only 0.8% of Θ. Hence, the probability of a solution from P_{BFA} is in the I/S-Race's initial sample is very low. However, as it can be seen, the I/S-Race was able to find solutions of the P_{BFA} quickly. It happened due to the I/S-Race's replacement

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

procedure where dominated solutions are replaced by new solutions around the elite configurations. Although the I/S-Race had started with a low retention value, after the 12^{th} iteration it achieved 100% of retention. This result overcame the S-Race's result which was 95% of retention.

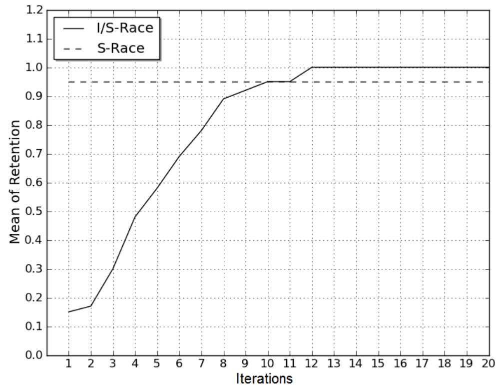

Fig. 1: Mean of Retention × iteration considering 20 classification problems.

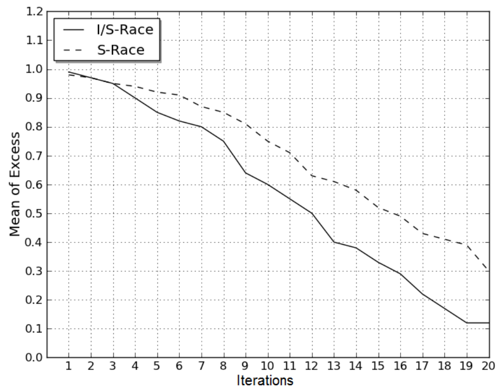

Fig. 2: Mean of Excess × iteration considering 20 classification problems.

In Figure 2 we also can see that the I/S-Race was able to replace most of its Pareto's solutions by P_{BFA}' solutions. In the first iteration the I/S-Race started with no solutions from the P_{BFA}. However, in the following iterations the I/S-Race quickly removed undesirable solutions from its Pareto achieving 12% of excess value. The S-Race algorithm reduced drastically its Pareto's size but the excess value stagnated in 30%. By applying statistical tests, we confirm that I/S-Race outperformed the S-Race regarding to E, with 95% of confidence.

Besides the analysis of E and R, we also performed a comparison between I/S-Race and S-Race considering the computational effort T. In average, the I/S-Race consumed only 15% of S-Race's total consumption. Other interesting

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

point is that the S-Race explored 100% of the search space, while the I/S-Race explored 32%. This shows that an iterative racing strategy to solve the parameter selection problem can be less computationally expensive and bring good results.

5 Conclusion

In this current work, we created an iterative MORA, I/S-Race, to select SVM parameters. The results showed I/S-Race produced a close Pareto front as a Brute-Force Approach (BFA) would yield, but at a fraction of computational cost. Besides, the I/S-Race was able to generate a better Pareto front when compared to the S-Race's one, performing less evaluations. In future work, we intend to create an exploration mechanism to avoid a possible stagnation in local minima, and investigate how the I/S-Race algorithm works with many objectives in other learning problems.

Acknowledgments: The authors would like to thank CNPq and CAPES.

References

[1] M. Birattari, T. Stutzle, L. Paquete and K. Varrentrapp. A Racing Algorithm for Configuring Metaheuristics. In GECCO 2002, p. 11-18.

[2] M. Birattari, Z. Yuan, P. Balaprakash, and T. Stutzle. F-race and iterated f-race: An overview. Technical report, IRIDIA, 2011.

[3] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines. In ACM Transac. on Intel. Systems and Technology, vol. 2(3), 2011.

[4] A. Frank and A. Asuncion. UCI Mach. Learning Repository, Avaiable in: http://archive.ics.uci.edu/ml. Irvine, CA: University of California, School of Information and Computer Science.

[5] F. Friedrichs and C. Igel. Evolutionary tuning of multiple svm parameters. In Neurocomputing, vol. 64, p. 107–117 2005.

[6] B. de Souza, A. de Carvalho, and R. Ishii. Multiclass svm model selection using particle swarm optimization. In 6th Int. Conf. on Hybrid Intelligent Systems, 2006.

[7] T. Zhang, M. Georgiopoulos and G. C. Anagnostopoulos. S-Race: A Multi-objective Racing Algorithm. In Proceedings of the 15th GECCO, p. 1565–1572, 2013.

[8] W. Mendenhall, D. D. Wackerly, and R. L. Scheaffer. Mathematical Statistics with Applications, p. 674679. PWS-Kent, 1989.

[9] S. Holm. A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics, 2:6570, 1979.