Combining Meta-Learning and Optimization Algorithms for Parameter Selection

T. Gomes and **P. Miranda** and **R. Prudêncio**[1] and **C. Soares**[2] and **A. Carvalho**[3]

Abstract. In this article we investigate the combination of meta-learning and optimization algorithms for parameter selection. We discuss our general proposal as well as present the recent developments and experiments performed using Support Vector Machines (SVMs). Meta-learning was combined to single and multi-objective optimization techniques to select SVM parameters. The hybrid methods derived from the proposal presented better results on predictive accuracy than the use of traditional optimization techniques.

1 Introduction

The induction of a machine learning model with a good predictive accuracy to solve a learning problem is influenced by a variety of aspects, such as data pre-preprocessing, algorithm selection, parameter optimization and training procedure. The study presented in this paper focuses on a specific and relevant step of modeling: *parameter selection*. Once a learning algorithm is chosen, the user has to define its parameter values. Learning performance is usually affected by a poor selection of these values. For instance, the performance of SVMs depends on the adequate choice of its kernel function, kernel parameters, regularization constant, among other aspects [2].

Parameter selection is treated by many authors as an optimization problem in which a search technique is employed to find the configuration of parameters which maximizes the learning performance estimated on the problem at hand. There is an extensive literature applying optimization algorithms for parameter selection, especially for Artificial Neural Networks. Although it represents a systematic approach to parameter selection, this approach can be very expensive, since a large number of candidate parameter configurations must be evaluated to ensure that an optimal, or at least reasonably good, set of parameters is found [6].

Meta-learning, originally proposed for algorithm selection, has also been adapted to parameter selection (e.g., for SVM [6, 1]). In this approach, the choice of parameter values for a given task is based on parameter values sucessfully adopted in similar problems. Each *meta-example* in this solution includes: (1) a set of characteristics (called *meta-features*) describing a learning problem; and (2) the best configuration of parameters (among a set of candidates) tested on that problem. A *meta-learner* is used to acquire knowledge from a set of such meta-examples in order to recommend (predict) adequate parameters for new problems based on their meta-features.

Compared to the optimization approach, meta-learning tends to be computationally more efficient, at least at the moment when the recommendation of parameters is made. It must be observed that meta-learning however is very dependent on the quality of its meta-examples. It is usually difficult obtaining good results since meta-features are in general very noisy and the number of problems available for meta-example generation is commonly limited.

As discussed in [4], good solutions to a particular search problem can be used to indicate promising regions of the search space for similar problems. Related ideas have been applied to improve optimization tasks but in very different contexts (e.g. job shop scheduling [4]). The positive results in these contexts motivated us to apply similar ideas for optimizing learning parameters. Here, we present the combination of optimization techniques and meta-learning for the problem of parameter selection. Meta-learning is used to suggest an initial set of solutions, which are then refined by a search technique. In previous work, the search process starts by evaluating random solutions from the parameter space. In the proposed hybrid approach, the search process starts with successful solutions from previous similar problems. Hence, we expect that meta-learning guides the search directly to promising regions of the search space, thus speeding up the convergence to good solutions.

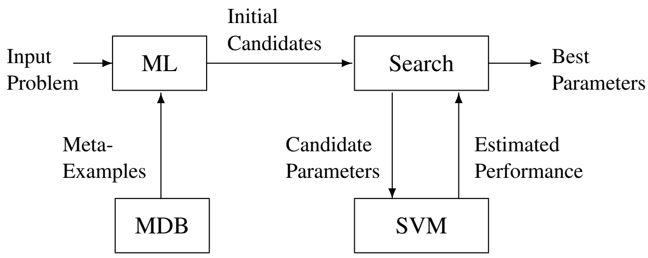

Figure 1. General Architecture

2 Developed Research

Figure 1 shows the general architecture of the proposed solution. Initially, the Meta-Learner (ML) module retrieves a predefined number of past meta-examples stored in a Meta-Database (MDB), selected on the basis of their similarity to the input problem. Next, the Search module adopts as initial search points the configurations of successful parameter values on the retrieved meta-examples. In the Search module, a search process iteratively generates new candidate values for the SVM parameters. The final solution which is recommended by the system is the best one generated by the Search module up to its convergence or other stopping criteria.

[1] Universidade Federal de Pernambuco, Brazil, email: {tafg,pbcm,rbcp}@cin.ufpe.br
[2] Universidade do Porto, Portugal, email: csoares@fep.up.pt
[3] Universidade de São Paulo, São Carlos, Brazil, email: andre@icmc.usp.br

In [3], we performed experiments that evaluated the proposed hybrid method using Particle Swarm Optimization (PSO) in the Search module. The system was empirically tested on the selection of two parameters for SVMs on regression problems: the γ parameter of the RBF kernel and the regularization constant C, which may have a strong influence in SVM performance. A database of 40 meta-examples was produced from the evaluation of a set of 399 configurations of (γ, C) on 40 different regression problems. Each meta-example refers to a single regression problem, which was described in our work by 17 meta-features (see [3] for details).

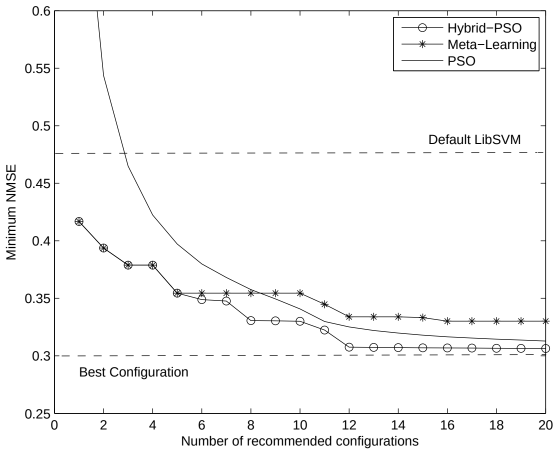

Figure 2. NMSE result obtained at each recommended configuration

Figure 2 compares the minimum NMSE (averaged over the 40 problems) obtained by SVM using the parameters suggested by combining meta-learning and PSO, referred to as Hybrid-PSO (using 5 initial solutions recommended by meta-learning), and the two methods individually, PSO (with random initialization, population size = 5) and meta-learning (which recommends the best configuration of each retrieved meta-example). We also present in Figure 2 the average NMSE achieved by the default heuristic adopted by the LibSVM tool (γ = inverse of the number of attributes and C=1). Finally, Figure 2 shows the average NMSE that would be achieved if the best parameter configuration had been chosen on each problem.

By comparing PSO and meta-learning, we identified a trade-off in their relative performances. Meta-learning is better than PSO for a small number of recommended parameter configurations. It is also better than the default LibSVM parameters. Hence, meta-learning alone would be indicated in situations in which the SVM user had strong resources constraints. In these situations, meta-learning could recommend a lower number of configurations with intermediate performance levels. PSO in turn is able to find better configurations along its search and then it is more adequate if a higher number of configurations can be tested.

The Hybrid-PSO was able to combine the advantages of its components. The performance of the Hybrid-PSO in the initial five recommended configurations is of course the same as the performance of meta-learning (since the initial configurations are recommended by meta-learning). From that point of the curve, the Hybrid-PSO consistently achieves better results compared to both the PSO and the meta-learning. It converges earlier to solutions with similar NMSE values compared to the best configurations observed in the 40 problems. There is an additional cost in recommending the configurations by the hybrid approach which is the cost of the meta-learning initialization (specially the cost of computing the meta-features). However, we deployed meta-features with a low computational cost.

In [5], we extended the previous work to perform Multi-Objective Optimization (MOO) of SVM parameters. The Multi-Objective PSO (MOPSO) algorithm was used to optimize the parameters (γ, C) regarding two conflicting objectives: complexity (number of support vectors) and success rate. We evaluated the MOPSO in two different versions: (1) MOPSO with initial population suggested by ML (Hybrid MOPSO) and (2) MOPSO with random initial population. In the meta-learning module, for each similar problem retrieved, we generated a Pareto Front (a set of non-dominated solutions) by applying the dominance evaluation to the 399 configurations of SVM parameters considered. In order to suggest an initial population, we select one random solution of each produced Pareto Front.

In our experiments, the final Pareto Fronts optimized by the MOPSO and the Hybrid MOPSO were evaluated using three metrics for MOO problems: Spacing, Hypervolume and Spread. The proposed hybrid approach was able to generate better comparative results, considering the Spacing and Hypervolume metrics. Regarding the maximum Spread, our approach lost in first generations, but was similar to MOPSO in the last generations.

3 Conclusion

The combination of meta-learning and optimization techniques showed promising results for SVM parameter values selection. The proposed approach can be easily adapted to other learning algorithms (e.g., Artificial Neural Networks). A number of aspects need to be investigated in our proposed solution such as alternative strategies to integrate meta-learning in the optimization process. For instance, not only the best solutions to similar problems can be considered, but also diverse solutions in the search space. Additionally, the limitations of the individual components (as usual in hybrid systems) need to be dealt with. For instance, new strategies to augment the number of datasets for meta-learning can improve the learning performance in our context.

Acknowledgments: The authors would like to thank CNPq, CAPES, FAPESP and FACEPE (Brazilian Agencies) and FCT (Portuguese Agency) for their financial support.

REFERENCES

[1] S. Ali and K. Smith-Miles, 'A meta-learning approach to automatic kernel selection for support vector machines', *Neurocomputing*, **70**(1-3), 173–186, (2006).

[2] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, 2000.

[3] T. Gomes, R. B. C. Prudêncio, C. Soares, A. Rossi, , and A. Carvalho, 'Combining meta-learning and search techniques to select parameters for support vector machines', *Neurocomputing*, **75**, 3–13, (2012).

[4] S. Louis and J. McDonnell, 'Learning with case-injected genetic algorithms', *IEEE Trans. on Evolutionary Computation*, **8**, 316–328, (2004).

[5] P. Miranda, R. B. C. Prudêncio, C. Soares, and A. Carvalho, 'Multi-objective optimization and meta-learning for svm parameter selection', in *IJCNN 2012 (to appear)*, (2012).

[6] C. Soares, P. Brazdil, and P. Kuba, 'A meta-learning approach to select the kernel width in support vector regression', *Machine Learning*, **54**(3), 195–209, (2004).