Method Article

Nature inspired optimization tools for SVMs - NIOTS

Carlos Eduardo da Silva Santos\textsuperscript{a,b,*}, Leandro dos Santos Coelho\textsuperscript{b,c}, Carlos Humberto Llanos\textsuperscript{a}

\textsuperscript{a}Universidade de Brasília – UnB, Brasília - DF - Brasil
\textsuperscript{b}Instituto Federal de Educação, Ciência e Tecnologia do Tocantins - IFTO, Palmas - TO - Brasil
\textsuperscript{c}Pontífica Universidade Católica do Paraná - PUC-Curitiba - PR, Brasil

A B S T R A C T

Support Vector Machines (SVMs) technique for achieving classifiers and regressors. However, to obtain models with high accuracy and low complexity, it is necessary to define the kernel parameters as well as the parameters of the training model, which are called hyperparameters. The challenge of defining the more suitable value to hyperparameters is called the Parameter Selection Problem (PSP). However, minimizing the complexity and maximizing the generalization capacity of the SVMs are conflicting criteria. Therefore, we propose the Nature Inspired Optimization Tools for SVMs (NIOTS) that offers a method to automate the search process for the best possible solution for the PSP, allowing the user to quickly obtain several sets of good solutions and choose the one most appropriate for his specific problem.

- The PSP has been modeled as a Multiobjective Optimization Problem (MOP) with two objectives: (1) good precision and (2) low complexity (low number of support vectors).
- The user can evaluate multiple solutions included in the Pareto front, in terms of precision and low complexity of the model.
- Apart from the Adaptive Parameter with Mutant Tournament Multiobjective Differential Evolution (APMT-MODE), the user can choose other metaheuristics and also among several kernel options.

© 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)
The development of models based on Machine Learning (ML) is trendy due to their ability to generate efficient models from a set of data [5]. The ML techniques are subdivided into two large groups depending on the available data, namely, supervised and unsupervised ones. Our proposed method is based on a multiobjective metaheuristic optimization approach to select Support Vector Machines and Support Vector Regressor (SVM/SVR’s) hyperparameters such as described in [11].

In this way, we present a method for solving the Parameter Selection Problem (PSP) for both SVM/SVR that uses supervised training. In this case, we seek to reach an optimal set of hyperparameters to minimize the number of support vectors and maximize generalization capacity. A significant problem is that SVM/SVR requires choosing a specific kernel to work with a particular dataset. In this way, the PSP consists of both obtaining the model parameters C (regularization parameter) for SVMs and the ε-insensitive tube width and C for SVR, apart from the parameters for the chosen kernel. Both sets of parameters together (SVM/SVR and kernel) are denoted as hyperparameter sets.

For solving the PSP we have created several steps and strategies into the Nature-Inspired Optimization Tool for SVM/SVR (NIOTS), which was developed to automate the definition of SVM/SVR hyperparameters, seeking a balance between complexity and accuracy. Thus, the PSP is treated as a multi-objective optimization to minimize both the empirical risk and the complexity, which are contradictory criteria models. Apart from the NIOTS automating the solution of PSP, the same allows the user to select a series of steps for each target problem, choosing the metaheuristics to be used, the kernel type, and generating a Pareto front to be analyzed. Once the user chooses among a set of options, NIOTS generates a report containing the Pareto front and Pareto set. Bearing in mind that the NIOTs allows us to choose from a tree of possibilities, our method empowers the designer to make the most appropriate decision for his problem, comparing the results of each possibility offered by NIOTS.

SVMs were developed by Vapnik [3]. The training of SVMs consists of solving a convex quadratic optimization problem, making it possible to use algorithms based on a gradient to obtain the optimal solution [2], for instance, the algorithm Sequential Minimal Optimization (SMO). The training process of an SVM is a convex quadratic optimization problem, as shown by Eq. 1, which is a necessary and sufficient condition to guarantee a global optimal solution.

\[
\max \ L_c(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j)
\]

S.I.
Table 1

| Kernel                | Kernel function                                                                 | Parameters |
|-----------------------|---------------------------------------------------------------------------------|------------|
| Polinomial            | $K(x,y) = (\beta \mathbf{x} \cdot \mathbf{y} + \kappa)^d$                      | $\beta$, $\kappa$ and $d$ |
| Gaussian              | $K(x,y) = \exp(-\frac{|x-y|^2}{\gamma})$                                        | $\gamma$ |
| Arc cosine (Cho e Saul 2009) | $K(x,y) = 1 - \frac{1}{\gamma} \cos^{-1}\left(\frac{y^T x}{\sqrt{|y|^2 |x|^2}}\right)$ | none |
| Cauchy (Drewnik e Pasternak-Winiarski 2017) | $K(x,y) = 1/(1 + |x-y|^2\gamma)$ | $\lambda$ |

$$
\sum_{i=1}^{N} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, N, \quad \xi_i \geq 0, \quad (1)
$$

$$
\xi_i \geq 0. \quad (\text{This equation was joined in equation (1)})
$$

where $C$ is a regularization parameter, $N$ is the training set cardinality, $\alpha$’s are the Lagrange multipliers, $\xi_i$ are the slack variables and $K(\cdot)$ is the symmetric Positive Semidefinite (PSD) matrix. This matrix must satisfy the importance conditions and is called kernel matrix or kernel function [4].

The optimal solution to the optimization problem of Eq. 1 is a sparse vector of weights $\mathbf{\alpha}$ (the Lagrangian coefficients) associated with each training set element (see Eq. 2).

$$
\hat{f}(x) = \sum_{i=1}^{\#SV} \alpha_i K(x, x_i) + \rho \quad (2)
$$

where $\hat{f}(x)$ is the predict value of the vector input $x$, $\#SV$ is the support vectors set cardinality, $\alpha_i$ are the Lagrange coefficients, $K(x, x_i)$ is the kernel function, $x_i$ is the $i$-th training vector and $\rho$ is the bias. Thus, the greater the sparsity of the optimal vector, the lower the complexity of the SVMs. Therefore, we seek SVM/SVRs with the least complexity possible and with the greatest accuracy. Both complexity and accuracy of an SVM are controlled indirectly by the $C$ (see Eq. 1) and the kernel parameters (see Table 1), which together are called hyperparameters of the model. The suitable choice of hyperparameters is strongly related to implicit characteristics to each set of training data and the chosen kernel, requiring the designer to handle hyperparameters and considerable effort in training models with different parameters.

Fig. 1 shows a flow chart diagram of the steps to be followed to obtain models based on machine learning. The steps (a) choice of training technique, (b) features selection, and (c) hyperparameters selection deserve to be highlighted because they consist of complex problems and a low-quality solution compromise the final model’s performance [1]. Due to these problems’ complexity, it is common to use metaheuristics to find solutions.

To develop a data-based model the data must be prepared. The initial set of steps we call the pre-processing block, as shown in Fig. 1. Pre-processing consists of the following steps: (a) data acquisition or generation; (b) data cleaning; (c) selection of characteristics; (d) normalization or standardization; and (e) dividing the data into three sets, e.g., training, validation, and testing. In preprocessing we highlight step (e), which is a complex problem and of fundamental importance for the final result of the model in question [8].

From the data sets obtained in the pre-processing block, there are the tasks included in the learning block (details in Fig. 1), which consist of the following steps: (a) choice of the training algorithm, (b) selection of hyperparameters, (c) training the model and, (d) final evaluation. In this block all steps are critical.

Within the learning box, there is a blue NIOTS sub-block that selects SVM hyperparameters iterating over the model refinement cycle. The available metaheuristics in the tool e.g., (1) Adaptive Parameters - Multi-objective Differential Evolution (AP-MODE), (2) Adaptive Parameters with Mutation Tournament - Multi-objective Differential Evolution (APMT-MODE), and (3) Multi-objective Particle Swarm Optimization (MOPSO) allow evaluating the models with the data from the validation set through the metrics ACC (Accuracy) or MSE (Mean Squared Error), as well as the complexity of the
Fig. 1. Machine learning flowchart.
model (support vector set cardinality \#SV). This strategy produces, as a result, a report with the set of solutions non-dominant, that is, the best models according to the criteria of accuracy and complexity.

The MOPSO employs the basic Particle Swarm Optimization (PSO) designed by Kennedy, [10] to update particles' velocity and position along the optimization process. We pick the global best particle and individual best particle from a non-dominated set with the same technique as in Santos [11]. In addition, the user has to choose the initial and final inertia, social and cognitive coefficient, the particle max velocity, and the population size.

In our proposal, NIOTS (see Fig. 1) automates the steps in purples (into the blue square), where the metaheuristics act over Refine model cycle (red arrows) and, as well as in the evaluation of the Final models (Final model evaluation cycle, see purple arrows). Usually, the metaheuristics have one metric to address for a solution, but this strategy induces overfitting. Therefore, to find the balance between generalization capacity and complexity, we model the PSP as a multi-objective problem taking as objective functions the support vectors cardinality (#SV) and the accuracy (ACC/MSE).

The three metaheuristics available in the NIOTS (e.g., APMT-MODE, AP-MODE, and the MOPSO) were coded in m language (from Matlab), offering to the user a Graphical User Interface Development Environment (GUIDE). In this way, our system has four tabs: (a) Optimization, (b) Ensembles, (c) Prediction, for testing the generated models and, (d) Statistics, for evaluating the results via statistical significance tests (see Fig. 2). Besides, NIOTS offers the user a certain degree of flexibility by the possibility to choose among several possible paths through the options described in Table 2.

In the NIOTS optimization environment, it is possible to optimize the hyperparameters of classifiers and regressors, using the MOPSO and APMT-MODE metaheuristic to solve the parameter selection problem formulated as a MOOP. The solutions obtained by the MOPSO and APMT-MODE metaheuristics are the suitable hyperparameters for the SVM/SVR.

The graphical interface of NIOTS (see Fig. 2) was developed to facilitate the several configuration possibilities that the system provides. The optimization environment is split into three panels: (1) Machine Definition, (2) General Parameters, and (3) Algorithm Parameters. Table 2 presents the description of each of the panel options.

In addition to the panels described in Table 2, the bottom has three buttons. The load-training-data button reads the file containing the training data, as shown in Fig. 3. To load the validation data, the load-validation-data button is used. This option is available if the cross-validation [9] option is not active. Both the name and location where the reports will be saved are chosen using the save-report button, and the run button starts the optimization process.

The AP/APMT-MODE parameters do not appear in Fig. 2, because only parameters of metaheuristic chosen in the Machine definition panel are displayed. This behavior of the program prevents the user from confusing the metaheuristics parameters.
Table 2
NIOTS description options on optimization panel

| Panel                        | Options          | Descriptions                                      |
|-----------------------------|------------------|---------------------------------------------------|
| Machine definition          | Model            | SVM/LibSVM                                        |
|                             |                  | SVR/LibSVM                                        |
|                             |                  | Grid Search SVR                                   |
| Optimization Algorithm      | Grid Search SVM  |                                                   |
| Kernel                      |                  |                                                   |
|                             |                  |                                                   |
| General parameters          | Samples          | Independent experiments amount.                  |
|                             | Iterations       | Optimization algorithm iterations amount.         |
|                             | Cross-validation | Enable the cross-validation training/validation process. |
|                             | Random distribution | Uniform                                      |
|                             |                  | Normal                                            |
|                             |                  | Cauchy                                            |
| Search space                |                  | The $C$ and $\gamma$ upper bounds                 |
|                             |                  | The $C$ and $\gamma$ lower bounds                 |
|                             |                  | The $\epsilon$-insensitive tube upper bound       |
| Metaheuristics parameters:  | The swarm size   | The swarm particles amount                        |
| MOPSO                       | Initial inertia  | The PSO Initial inertia factor                    |
|                             | Final inertia    | The PSO Final inertia factor                      |
|                             | Cognitive coefficient | The particle confidence level is in its own best position. |
|                             | Social coefficient | The particle confidence level is the best particle position overall. |
|                             | The particle max speed | The particle max speed allowed by PSO. |
| Metaheuristic parameters:   | Population size  | The Differential Evolution (DE) individuals amount.|
| AP/APMT-MODE                | Scale factor     | Initial scale factor.                             |
|                             | Crossover rate   | Initial crossover rate.                          |

Fig. 3. NIOTS input file example

The LibSVM (a Library for Support Vector Machines) tool (Chang, 2011) conducts the training using the APMT-MODE or MOPSO multi-objective metaheuristics for obtaining the optimized hyperparameters. These models are evaluated with the data from the validation set, returning the metrics of the model (generalization capacity and complexity). In Fig. 1 this is represented by the refine-model cycle.
In the flowchart of Fig. 4, the variable $\omega$ is the vector of the hyperparameters and the functional values $E(\omega)$ and $C_{SV}(\omega)$ are the metrics of generalization capacity and complexity, respectively. The filled arrows, in black, represent the data flow between the blocks, while the white arrows indicate the possible options that the user can adopt, also presented in Table 2.

At the end of the optimization process, three types of reports are generated, represented in Fig. 2 by the reports file block. For each hyperparameter of the Pareto Front (PF) set (in details Fig. 4), a report file is generated, called model-data, which has all the data necessary to implement the model in question. Fig. 5 is an example of this report.

The Pareto Front report (details in Fig. 6) contains the user-defined AP/APMT-MODE/MOPSO parameters and the elements of the Pareto Set where the index column represents each element of the set, which is composed of obtained hyperparameter ($C$, order, and $\varepsilon$) for an Hermite polynomial kernel, as well as their respective PF objectives (MSE and SV columns).

In the SVM/SVR design, a kernel must be defined as a priori, which has several parameters. Due to their particular characteristics, each kernel combined with each benchmark generates a different PSP. Each combination of kernel and training set generates a distinct optimization problem with several local minima, making choosing the optimal hyperparameters a complex task. However, for this task, NIOTS makes hyperparameters smart search through efficient metaheuristics, highlighting in this context the APMT-MODE. Besides, the PSP is modeled as a MOOP approach providing the designer with a set of high-quality solutions that should contemplate several constraints for SVM/SVR applications. In the NIOTS the PF quality, obtained for each metaheuristic, is measured with the
Fig. 5. The models report file

Fig. 6. NIOTS report file for SVR, first three columns are Pareto set and the two last one are Pareto front.

Inverted Generational Distance (IGD) metric (Ishibuchi, 2015). Eq. 2 shows as IGD is calculated.

\[
IGD(S, P^*) = \frac{1}{|P^*|} \sum_{y \in P^*} \text{dist}(y, S)
\]
where $P^*$ is the true PF, $\text{dist}(y, S)$ is the Euclidean distance between $y \in P^*$ from the nearest point in $S$, which is the set of solutions obtained by the metaheuristics, and $|P^*|$ the cardinality of true PF. The IGD measures the convergence and spread of the $S$ set and the lower the IGD the better is the $S$ set.

Taking into account that in practice we do not have an optimal PF, in the NIOTS each metaheuristic generates several non-dominant sets that are used for obtaining a general non-dominant set of solutions by achieving the union of these partial sets, that we adopt as the true PF for PSP [6] [7].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Christian. Igel, Multi-Objective Model Selection for Support Vector Machines, Lecture Notes in Comput. Sci. Evol. Multi-Criterion Optim. (2005) 534–546, doi:10.1007/978-3-540-31880-4_37.
[2] S.S. Keerthi, et al., Improvements to Platt’s SMO Algorithm for SVM Classifier Design, Neural Comput. 13 (3) (2001) 637–649, doi:10.1162/089976601300014493.
[3] Vapnik, V.N. “The Nature of Statistical Learning Theory.” 1995, doi:10.1007/978-1-4757-2440-0.
[4] C.J. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining Knowl. Discov. 2 (2) (1998) 121–167.
[5] Jonathan Waring, et al., Automated Machine Learning: Review of the State-of-the-Art and Opportunities for Healthcare, Artif. Intell. Med. 104 (2020) 101822, doi:10.1016/j.artmed.2020.101822.
[6] H. Ishibuchi, H. Masuda, Y. Tanigaki, Y. Nojima, Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems, in: 2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM), IEEE, 2014, pp. 170–177.
[7] Xi-nye Cai, et al., A grid-based inverted generational distance for multi/many-objective optimization, IEEE Trans. Evol. Comput. 25 (1) (2020) 21–34.
[8] J. Schmidt, M.R. Marques, S. Botti, M.A. Marques, Recent advances and applications of machine learning in solid-state materials science, npj Comput. Mater. 5 (1) (2019) 1–36.
[9] Y. Bengio, Y. Grandvalet, No unbiased estimator of the variance of k-fold cross-validation, J. Mach. Learn. Res. 5 (Sep) (2004) 1089–1105.
[10] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN95-international conference on neural networks, IEEE, 1995, pp. 1942–1948, Vol. 4.
[11] C.E.D.S. Santos, R.C. Sampaio, L.D.S. Coelho, G.A. Bestard, C.H. Llanos, Multi-objective adaptive differential evolution for svm/svr hyperparameters selection, Pattern Recognit. (2021) 110.