A Novel Orthogonal Direction Mesh Adaptive Direct Search Approach for SVM Hyperparameter Tuning

Alexandre R. Mello\textsuperscript{a,b}, Jonathan de Matos\textsuperscript{c}, Marcelo R. Stemmer\textsuperscript{b}, Alceu de S. Britto Jr.\textsuperscript{c}, Alessandro L. Koerich\textsuperscript{a}

\textsuperscript{a}École de Technologie Supérieure - Université du Québec 1100 Notre-Dame West, Montréal, QC, H3C 1K3, Canada.
\textsuperscript{b}University of Santa Catarina Campus Reitor João David Ferreira Lima, s/n - Trindade, Florianópolis, SC, 88040-900, Brazil.
\textsuperscript{c}Pontifícia Universidade Católica do Paraná, Curitiba, PR, 80215-901, Brazil.

Abstract

In this paper, we propose the use of a black-box optimization method called deterministic Mesh Adaptive Direct Search (MADS) algorithm with orthogonal directions (Ortho-MADS) for the selection of hyperparameters of Support Vector Machines with a Gaussian kernel. Different from most of the methods in the literature that exploit the properties of the data or attempt to minimize the accuracy of a validation dataset over the first quadrant of $(C, \gamma)$, the Ortho-MADS provides convergence proof. We present the MADS, followed by the Ortho-MADS, the dynamic stopping criterion defined by the MADS mesh size and two different search strategies (Nelder-Mead and Variable Neighborhood Search) that contribute to a competitive convergence rate as well as a mechanism to escape from undesired local minima. We have investigated the practical selection of hyperparameters for the Support Vector Machine with a Gaussian kernel, i.e., properly choose the hyperparameters $C$ (bandwidth) and $\gamma$ (trade-off) on several benchmark datasets. The experimental results have shown that the proposed approach for hyperparameter tuning consistently finds comparable or better solutions, when using a common configuration, than other methods. We have also evaluated the accuracy and the number of function evaluations of the Ortho-MADS with the Nelder-Mead search strategy and the Variable Neighborhood Search strategy using the mesh size as a stopping criterion, and we have achieved accuracy that no other method for hyperparameters optimization could reach.

Keywords: NOMAD, hyperparameter tuning, model selection, SVM, MADS, VNS

1. Introduction

Support Vector Machine (SVM) \cite{1,2} is a popular algorithm used in machine learning for statistical pattern recognition tasks, originally designed for binary classification \cite{3}. The so-called maximal margin classifier optimizes the bounds of the generalization error of linear machines by separating the data with a strategy that attempts to find the maximal margin hyperplane in an appropriately chosen kernel-introduced feature space \cite{4}. An alternative to increase the SVM flexibility and improve its performance is a nonlinear kernel function such as the radial basis function (RBF or Gaussian). The SVM with Gaussian kernel has two important hyperparameters that impact greatly in the performance of the learned model: the soft-margin $C$ and the kernel hyperparameter $\gamma$. Therefore, the application of the SVM with Gaussian kernel to a classification problem requires an appropriate selection of hyperparameters, called hyperparameter tuning or model selection. Given these two hyperparameters and a training dataset, an SVM solver can find a unique solution of the constrained quadratic optimization problem and return a classifier model. Unfortunately, there is no standard procedure for hyperparameter tuning, and a common approach is to split the dataset into training and validation set, and for each $C$ and $\gamma$ from a suitable set, select the pair that results in an SVM model that when trained on the training set has the lowest error rate over the corresponding validation set \cite{5}.

Black-box optimization (BBO) is the study of the design and analysis of algorithms that assume that the objective and/or constraint functions are given by a black-box \cite{6}. Some of the most used methods to solve BBO problems are: (i) the naive methods such as the exhaustive search, grid search, and coordinate search; (ii) the heuristic methods, such as the genetic algorithm (and its variations) and the Nelder-Mead search \cite{7}; and (iii) the direct search algorithms, such as the Generalized Pattern Search (GPS) and the Mesh Adaptive Direct Search (MADS) \cite{8}. The naive and the heuristic methods do not guarantee convergence, while the direct search methods combine a flexible framework with proof of convergence \cite{6}.

Considering the relationship between the data geometric structure in the feature space and the kernel function (that is not relevant to $C$), Jiancheng Sun et al. \cite{9} determine $C$ and $\gamma$ separately by two stages. Chen et al. \cite{10} proposed another two-step procedure for efficient parameter selection by exploiting the geometry of the training data to select $\gamma$ directly via nearest neighborhood and choosing the $C$ value with an elbow method that finds the smallest $C$ leading to the highest possi-
ble validation accuracy. This approach reduces the number of candidate points to be checked, while maintaining comparable accuracy to other classical methods. Chung et al. [11] and Gold et al. [12] introduced the Bayesian Optimization (BO) approach for tuning the kernel hyperparameters, which constructs a probabilistic model to find the minimum function \( f(x) \) on some bounded set \( X \), that exploits the model to make decisions about where in \( X \) to next evaluate the function while integrating out uncertainty [13]. Acerbi and Ji [14] proposed the Bayesian Adaptive Direct Search (BADS) which combines a Bayesian optimization with the MADS framework via a local Gaussian surrogate process, implemented with a number of heuristics. The BADS’ goal is fitting moderately expensive computational models, and it achieved state-of-the-art accuracy on computational neuroscience models (such as convolution neural networks). Chang and Chou [15] proposed a two-stage method for hyperparameter tuning. The first step consists of tuning the \( \gamma \) using the generalization error of a k-NN classifier with the aim of maximizing the margins and extend the class separation. The second stage defines the value of \( C \) by an analytic function obtained with the jackknife technique.

Although many efforts to properly tuning the hyperparameters of a SVM with Gaussian kernel have been made, most of the BBO-based methods lack convergence proof. The BBO problems may present a limited precision or may be corrupted by numerical noise, which leads to an invalid output. Considering a fixed starting point, a BBO algorithm may provide different outputs, and there are unreliable properties frequently encountered in real problems [16]. Furthermore, most of BBO-based methods use the time or number of function evaluations as a stopping criterion, which creates an uncertainty on the achieved local minimum.

The NOMAD software (Nonlinear Optimization by MADS) [17, 18] is a C++ implementation of the MADS algorithm which can efficiently explore a design space in search of better solutions for a large spectrum of BBO problems as described by Audet [19], and inspired by cases of success as [14] and [19]. In this paper, we use the Ortho-MADS [20] (that is a MADS improvement), with two different search strategies Nelder-Mead [7] and the Variable Neighborhood Search (VNS) [21], to tune hyperparameters of a SVM with Gaussian kernel considering a dynamic stopping criterion. The proposed method relies on the MADS convergence properties and it combines different search strategies to reach the desired local minimum (that is set by the mesh size) and escape from undesired local minimum. The experimental results on benchmark datasets have shown that the proposed approach achieves state-of-the-art accuracy, besides presenting several interesting properties such as the guarantee of convergence and a dynamic stopping criterion. Furthermore, it also provides many other tools that aid to adjust the hyperparameters regarding the particularities of each application.

This paper is organized as follows: Section 2 introduces the basic definitions of SVM, notation and the formulation of a BBO problem. Section 3 presents the proposed approach for hyperparameter tuning. In Section 4, beyond describing other hyperparameter optimization methods used as benchmarks, we present our experimental protocol and the observed results. The conclusions and perspectives for future research are stated in the last section.

2. Basic Concepts of SVM

Let’s consider a training dataset \( H = \{ h_1, \ldots, h_n \in \mathbb{R}^{d \times n} \} \), where each \( h_i \) consists of \( n \) features and one possible class label \( y_i = \pm 1 \). We can reduce the SVM problem to a convex optimization form, i.e., minimize a quadratic function under linear inequality constraints\(^1\). Given a kernel function \( k(h, h') \) that must satisfy the distance relationship between transformed and original space, i.e., the kernel function satisfies \( k(h, h') = \Phi(h)^\top\Phi(h') \), it can map the training instances into some feature space \( \mathcal{F} : \mathbb{R}^d \to \mathcal{F} \). The Gaussian Radial Basis Function (RBF) kernel is a usual choice for the kernel function, and it is defined as:

\[
\begin{align*}
 k(h, h') := \exp\left(-\frac{||h - h'||^2}{2\sigma^2}\right) \\
 = \exp(\gamma||h - h'||^2), \quad \forall h, h' \in \mathbb{R}^d
\end{align*}
\]

where \( ||\cdot|| \) represents the \( l_2 \) norm and \( (\gamma, \sigma) \) are fixed constants with \( \gamma = \frac{1}{2\sigma^2} \). The maximum margin separating hyperplane in the feature space is defined as \( \omega^\top\Phi(h) + b = 0 \), and the quadratic programming problem that represents the nonlinear soft-margin SVM is described by Eq. 2 in its primal form. The minimization of Eq. 2 results in the maximum margin separating hyperplane.

\[
\begin{align*}
 \min_{\omega,h,b} \frac{1}{2}||\omega||^2 + C \sum_{i=1}^{n} \eta_i \\
 \text{subject to } y_i(\omega^\top\Phi(h_i) + b) \geq 1 - \eta_i \\
 \eta_i \geq 0, \forall i
\end{align*}
\]

where \( \eta \) is the slack variable that indicates tolerance of misclassification, the constant \( C > 0 \) is a trade-off hyperparameter. We calculate the classification score using Eq. 3, regarding the distance from an observation \( h \) to the decision boundary.

\[
f(h) = \sum_{i=1}^{d} \alpha_i y_i k(h_i, h) + b
\]

where \( \alpha_1, \ldots, \alpha_d \) and \( b \) are the estimated SVM parameters. For the cases where we have a multiclass problem, we can extend the SVM by reducing the classification problem with three or more classes to a set of binary classifiers using the one-vs-one or the one-vs-rest approach [3]. In this paper, we use the one-vs-one error-correcting output codes (ECOC) model [22]. Although, the choice of the multiclass approach does not interfere in our proposed solution.

A practical difficulty in the nonlinear SVM is to proper tune the hyperparameters \( \gamma \) and \( C \) from Eqs. 1 and 2 respectively. For the BBO-based methods (that are not based on internal metrics), the tuning process maximizes the accuracy on a training data usually using a cross-validation procedure, which is

\(^1\)All proves and deductions can be found in [4].
equivalent to minimizing a loss function ($L$), as described by Eq. 4.

$$\max_{y > 0, C > 0} \text{Accuracy}(y, C) \equiv \min L$$  \hspace{1cm} (4)

Almost all loss functions commonly used in the literature meet convexity assumption, however, different loss functions lead to different theoretical behaviors, i.e., different convergence rates [23]. It is possible to use any loss function that meets convexity with the proposed approach, but based on Rosasco et al. [23] and considering the loss function options for classification tasks, we have chosen the Hinge loss (HL). The HL leads to a convergence rate practically equal to the convergence rate of a logistic loss and better than the convergence rate of a square loss. However, considering a hypothesis space sufficiently rich, the thresholding stage has little impact on the obtained bounds. We define the weighted average classification hinge loss to tune the hyperparameters of the SVM as Eq. 5.

$$L = \sum_{i=1}^{d} w_i \max[0.1 - m_i]$$  \hspace{1cm} (5)

where $w_i$ is the normalized weight for an observation $i$ ($\sum_{i=1}^{d} w_i = 1$) and $m_i = f(h_i)$ is the scalar classification score when the model predicts true for the instance $h_i$. We can formulate the SVM with Gaussian kernel hyperparameter tuning problem as a BBO function $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ in the sense that we obtain a function value from a given $x$ in $\mathbb{R}$ (in this case $x$ corresponds to the hyperparameters $C$ and $\gamma$) for which the analytic form of $f$ is unknown. The objective function $f$ and the different functions defining the set $\Omega$ are provided as a bounded optimization problem of the form:

$$\min_{x \in \Omega \subseteq \mathbb{R}^n} f(x)$$  \hspace{1cm} (6)

where $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$, $\Omega = \{x \in X : c_j(x) \leq 0, j = 1, 2, \ldots, m\}$, and $X \subseteq \mathbb{R}^n$ represents closed constraints which are bounded by $L \leq x \leq U$, with the lower ($L$) and the upper ($U$) bounds in $[\mathbb{R} \cup \{\infty\}]^n$, and the functions $c_j$ represent the other $m$ open constraints [21, 24].

3. The Ortho-MADS with NM and VNS

In this section, we briefly introduce the methods that we propose for tuning the hyperparameters of a SVM with Gaussian kernel, which are the MADS and the Ortho variation, as well as the two search methods used with the Ortho-MADS: the Nelder-Mead (NM) search and the Variable Neighborhood Search (VNS).

3.1. Mesh Adaptive Direct Search (MADS)

The Mesh Adaptive Direct Search (MADS) [8] is an iterative algorithm to solve problems of the form Eq. 6, in which the goal at each iteration is to replace the current best feasible points (called incumbent solution, or the incumbent) by a better one. The algorithm starts from a finite collection of initial points $V_0 \subseteq \mathbb{R}^n$, and the incumbent $x_h$ at iteration $k$ is defined as $f(x) : x \in V^k \cap \Omega$, where $V^k$ is the set of trial points where the black-box was previously evaluated. To achieve a better solution at each iteration it uses a mesh $M^k$ to generate trial points on a discretized space, defined by Eq. 7.

$$M^k := \{x^k + \delta^k D y : y \in \mathbb{N}^p\}$$  \hspace{1cm} (7)

where $\delta^k$ defines the mesh size, $D$ is the set of directions, and $p$ is the number of directions. At iteration $k$, the algorithm attempts to improve the incumbent solution by executing the search or poll stage, allowing an attempt of improving the current solution or escaping from an undesired local minimum. The search stage attempts to improve the incumbent by evaluating points that are generated from a finite trial points of the subset $M^k$, based on the $D$ set of directions, and that are not necessarily close to the current incumbent. When the search stage does not succeed, the poll stage is executed to evaluate points inside a frame defined by Eq. 8.

$$F^k := \{x \in M^k : \|x - x^k\|_\infty < \Delta^k b\}$$  \hspace{1cm} (8)

where $x$ is the point being evaluated, $\Delta^k$ is the frame size that must satisfy $0 < \delta^k \leq \Delta^k$, and $b$ is defined by Eq. 9.

$$b = \max\{\|d'\|_\infty : d' \in D\}$$  \hspace{1cm} (9)

where $d'$ is a direction.

The frame is always larger than the mesh and provides a broader sampling. If the poll stage obtains a new incumbent, the frame moves to this point and increases its size, otherwise, it decreases its size and maintains the current incumbent. We define the initial poll size as $\Delta^0 = \frac{U - L}{10}$ according to [25], and the initial mesh size as $\delta^0 = \Delta^0$. The poll stage selects points based on a subset $D^k$ of directions from $D$, $\delta$ and $\Delta$, and at each iteration, the mesh size $\delta$ is updated as $\delta = \min(\Delta^k, \Delta^k \tau^2)$, changing the frame size by $r^{-1} \Delta$ when increased and $r^{1} \Delta$ when decreased, where $r \in [0, 1]$. This process allows a continuous refinement of the mesh around the incumbent, and consequently reducing the objective function value.

The Ortho-MADS [20] is an improvement in the poll stage of the MADS algorithm, where the choice of the polling directions is deterministic and orthogonal to each other. This leads, at each iteration, to convex cones of missed directions that are minimal in a reasonable measure [20]. The Ortho-MADS changes the evaluation point selection method during the poll stage by creating $D^k$ set of orthogonal directions, which define a dense sphere that increases with the number of iterations $k^2$.

Fig. 1a depicts a mesh and frame of size one, where the poll stage points $p^0$, $p^1$, and $p^2$ are limited by the frame and the mesh size, which in this case, the search and poll stage would find both the same points, performing similarly. In Fig. 1b, the mesh size is half of the frame size, and the poll stage can use 24 points defined by the mesh intersections (excluding $x^k$), allowing a broader exploration of the search space than the searching step.

Among the traditional stopping criteria, such as time and number of function evaluations, the MADS can terminate the

---

2For further information refers to [20].
optimization process when it achieves a desired minimum mesh size value, that corresponds to achieve a desired local minimum. During the MADS optimization process, it continuously updates the mesh (δ) and the frame (Δ) sizes. Both δ and Δ become smaller as the algorithm moves towards the minimum of f(x). This mesh size verification is the last step in the MADS optimization process when it achieves a desired minimum mesh size value, that corresponds to achieve a desired local minimum.

3.2. Search Methods

The search stage is not necessary for the convergence analysis and can be done using different strategies such as the Variable Neighborhood Search (VNS) [21] or the Nelder-Mead (NM) search [7]. The MADS algorithm does not dictate the selection of points in the search stage.

The NM [7] is a method for function minimization proposed by Audet and Tribes [26] to be used in the search stage of the MADS algorithm. The NM continuously replaces the worst point x^o from a set of n points defined as X = \{x^0, x^1, \ldots, x^n\}, where X is a set of n vertices from a simplex problem for the minimization function f(x). A point x_{new} is considered better, or is said to dominate another point x, if f(x_{new}) < f(x). The domination defines the function Best(x_{new}, x) as in Eq. 10.

\[
\text{Best}(x_{\text{new}}, x) = \begin{cases} x_{\text{new}}, & \text{if } f(x_{\text{new}}) < f(x) \\ x, & \text{otherwise} \end{cases}
\] (10)

At each iteration, the NM evaluates f(x) at the points given by the simplex and replaces x^o according to the following criteria:

\[
x^o = \begin{cases} \text{shrink}(X) & \text{if } x^c \in \text{inside contraction zone} \\ x^c & \text{if } x^c \notin \text{inside contraction zone} \\ \text{Best}(x', x^e) & \text{if } x' \in \text{expansion zone} \\ x' & \text{if } x' \in \text{reflection zone} \\ \text{Best}(x', x^{ic}) & \text{if } x' \in \text{outside contraction zone} \end{cases}
\] (11)

where x', x^e, x^{oc}, and x^{ic} are defined by Eqs. 13 to 16 as follows:

\[
x^e = \frac{1}{n} \sum_{i=0}^{n-1} x^i
\] (12)

\[
x' = x^e + (x^e - x^o)
\] (13)

\[
x^c = x^e Q'(x^e - x^o)
\] (14)

\[
x^{oc} = x^e Q^{oc}(x^c - x^o)
\] (15)

\[
x^{ic} = x^e Q^{ic}(x^c - x^o)
\] (16)

\[
\text{shrink}(X) = x^0, x^0 + \zeta(x^1 - x^0), x^0 + \zeta(x^2 - x^0), \ldots, x^0 + \zeta(x^n - x^0)
\] (17)

where Q', Q^{oc}, and Q^{ic} are expansion, outside contraction and inside contraction respectively, being defined as Q' = 2, Q^{oc} = \frac{-1}{2}, and Q^{ic} = \frac{1}{2}. ζ is the shrinking parameter usually defined as ζ = \frac{1}{2}.

The zone definition of a new point x is given as follows:
and in the last case, \( x \) dominates none or one point of \( \mathcal{X} \).

The NM method improves the search stage by selecting new points in a more controlled way than e.g., a random selection. The MADS algorithm can also use other methods in the search stage, such as the Bayesian Optimization method, which is the basis of the BADS algorithm. The convergence rate of the MADS algorithm depends on the quality of the search method.

In order to include a far-reaching search step to escape from an undesired local minimum, Audet et al. [21] also incorporates the Variable Neighborhood Search (VNS) [27, 28] as a search stage in the MADS algorithm. The VNS algorithm complements the MADS poll stage, i.e., when current iteration results in no success, which means that it could not find points with a smaller \( f(x) \), the next poll stage generates trial points closer to the poll center, while the VNS explores a more distant region with a larger perturbation amplitude. The VNS uses a random perturbation method to attempt to escape from a local optimum solution so that a new descent method from the perturbed point leads to an improved local optimum. The VNS requires a neighborhood structure that defines all possible trial points reachable from the current solution, and a descent method that acts in the structure. The VNS amplitude of iteration \( k \) is parameterized by a non-negative scalar \( \xi_k \in \mathbb{N} \) that gives the order of the perturbation.

The MADS mesh provides the required neighborhood structure to the VNS, and by adding a VNS exploration in the search step, it introduces two new parameters: one is related to the VNS shaking method \( \Delta_k > 0 \), and the other defines a stopping criterion for the descent \( \rho > 0 \) [21]. The shaking of iteration \( k \) generates a point \( x' \) belonging to the current mesh \( M(k, \Delta_k) \), and the amplitude of the perturbation is relative to a coarser mesh, which is independent of \( \Delta_k \). The VNS mesh size parameter defined as \( \Delta_k > 0 \) and the VNS mesh \( M(k, \Delta_k) \) are constant and independent on the iteration number \( k \) not to be influenced by a specific MADS behavior (the perturbation amplitude \( \xi_k \) is updated outside the VNS search step). The shaking function is defined as:

\[
\text{shaking} : (M(k, \Delta_k), \mathbb{N}) \rightarrow M(k, \Delta_k) \subseteq M(k, \Delta_k) \\
(x, \xi_k) \mapsto x' = \text{shaking}(x, \xi_k)
\]  

(18)

The VNS descent function generates a finite number of mesh points and it is defined as:

\[
\text{descent} : M(k, \Delta_k) \rightarrow M(k, \Delta_k) \\
x' \mapsto x'' = \text{descent}(x')
\]  

(19)

where \( x' \) is the point resultant from the previous shaking and \( x'' \) is a point based on \( x' \) but with improved \( f(x) \) value. The improvement is important because \( x' \) has low probability to generate good optimization results due to its random choice by the shaking.

The descent step must lead towards a local optimum, and in the MADS context the local optimality is defined with respect to the mesh, so the descent step acts with respect to the current step size \( \Delta_k \) and the directions used. To reduce the number of function evaluations and to avoid exploring a previously visited region, the descent step stopping criterion is defined as \( \| x - x_{new} \|_2 \leq \rho \), where \( x_{new} \) is a trial point close to another point \( x \) considered previously\(^1\). The VNS generally brings about a higher number of black-box evaluations, but these additional evaluations lead to better results [21].

We choose to use the mesh size parameter \( \delta_k \) as stopping criterion because it corresponds to a situation where new refinements could not find a better solution, meaning a local minimum, and according to Audet and Hare [6], the mesh size parameter goes to zero faster than the poll size parameter \( \Delta_k \). The NM improves the quality of the solutions in the search stage [26], while the VNS allows the far-reaching exploration from current incumbent [21]. The pseudo-code in Alg. 1 describes the high-level procedure of the MADS technique considering the Ortho-MADS algorithm with the Nelder-Mead and the VNS search strategies and the minimum mesh size as stopping criterion implemented by NOMAD.

To run the NOMAD software with the Ortho-MADS, NM, and VNS, we need to set the lower bound \( L = [l_1, l_2] \), the upper bound \( U = [u_1, u_2] \), and the initial poll and mesh size are defined as \( \Delta_k^0 = \delta_k^0 = \frac{u_j - l_j}{10} \), where the mesh size parameter is associated with the variable \( j \in \{1, 2, \ldots, n\} \). The algorithm (Alg. 1) starts evaluating \( f(x) \) within the initial point. The first iteration executes a search step, and in case of a failure, that is, not finding a \( f(x) \) value smaller than \( f(x_k) \), it executes a poll step with Ortho-MADS direction. In case of a successful iteration of the NM-Search and the poll step, the next iteration runs the VNS-search to try escaping from an eventual and undesired local minimum.

4. Experimental Protocol and Results

We use the NOMAD black-box optimization software [17, 18] (version 3.9.1), which provides several interfaces to run the Ortho-MADS and its variations, including MATLAB. The other approaches used for comparison are also implemented in MATLAB, and we developed an experimental protocol to compare the black-box optimization methods in a machine learning classification context based on Audet and Hare [6] and Moré and Wild [29]. The experimental protocol consists of three steps:

1. Select datasets;
2. Algorithm comparison using a common configuration;
3. Evaluation of the proposed strategy.

4.1. Datasets

We have selected thirteen benchmark datasets with different numbers of instances and dimensions to evaluate the proposed

\(^1\)Further information of the VNS and the MADS integration can be found at [21]
Algorithm 1 NOMAD high-level procedure

Input: Initial point \(x_0 = [c_0, y_0]\), VNS amplitude parameter \(\xi\), Minimum mesh size \((\delta_{\min}, \delta_{\min-y})\)
Output: Best point \(x_{\text{best}} = [c_{\text{best}}, y_{\text{best}}]\)
Initialization: \(k \leftarrow 0\)

1: while \(\delta_k^C > \delta_{\min}\) and \(\delta_k^D > \delta_{\min-y}\) do
2: \(x' \leftarrow\) shaking\((x_k, \Delta_k)\)
3: \(x'' \leftarrow\) descent\((x')\)

Search stage
4: \(S_k \leftarrow\) finite number of points of \(M(k, \Delta_k) \Rightarrow M(k, \Delta_k)\) is the current mesh size
5: Evaluates \(f(i)\) on \(S_k \cup x'' \Rightarrow f(i)\) is the sub-step evaluation inside the search stage
6: if \(f(t) < f(x_k)\) for some \(t\) in a finite subset of \(S_k \subset M_k\) using NM-
SEARCH then
7: \(x_{k+1} \leftarrow t\)
8: goto 20
9: end if

Poll stage
10: Compute \(p\) MADS directions \(D_k \in \mathbb{R}^{\text{exp}} \Rightarrow p\) is the number of Poll stage points and \(D_k\) is the directions set at \(k\)
11: Construct the frame \(P_k \subseteq M(k, \Delta_k)\)
12: Evaluates \(f(p)\) on \(p\) points of \(P_k\)
13: if \(f(p) < f(x_k)\) then
14: \(x_{k+1} \leftarrow f\)
15: \(\Delta_{k+1} \leftarrow r^{-1}\Delta_k\)
16: else
17: \(x_{k+1} \leftarrow x_k\)
18: \(\Delta_{k+1} \leftarrow r^{\Delta}_k\)
19: end if
20: Update VNS amplitude \((\xi_{k+1} \leftarrow \xi_0\) or \(\xi_{k+1} \leftarrow \xi_k + \delta)\)
21: Updates of solution and mesh
22: \(k \leftarrow k + 1\)
23: end while
24: return \(x_k\)

approach, and to compare it with other strategies available in the literature. These datasets are publicly available at the LIB-SVM website\(^4\). Tab. 1 summarizes the main characteristics of each dataset.

| Dataset       | #Class | #Features | #Train | #Test | #Valid |
|---------------|--------|-----------|--------|-------|--------|
| Astroparticle | 30     | 2         | 4,069  | 4,000 | NA     |
| Car           | 30     | 2         | 1,243  | 41    | NA     |
| DNA           | 3      | 180       | 1,400  | 1,186 | 600    |
| Letter        | 31     | 26        | 10,500 | 5,000 | 4,500  |
| Madelon       | 32     | 2         | 2,000  | 600   | NA     |
| Pendigits     | 31     | 16        | 7,494  | 3,498 | NA     |
| Protein       | 33     | 3         | 14,895 | 6,621 | 2871   |
| Satimage      | 3      | 6         | 3,104  | 2,000 | 1,331  |
| Shuttle       | 3      | 7         | 30,450 | 14,500| 13,050 |
| Splice        | 31     | 2         | 60,000 | 2,175 | NA     |
| SVMguide4     | 30     | 4         | 300    | 312   | NA     |
| USPS          | 34     | 10        | 256    | 7,291 | 2,007  |
| Vowels        | 31     | 10        | 958    | 462   | NA     |

\(^4\)https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/

4.2. Other BBO methods

We have selected five widely used BBO-based methods of hyperparameter tuning for comparison purposes: Bayesian optimization (BO), Bayesian Adaptive Direct Search (BADS), Simulated Annealing (SA), Grid Search (GS), and Random Search (RS). These methods are briefly described as follows:

- The Bayesian optimization (BO) \([13]\) uses a probabilistic function \(f(x)\) as a model for the problem. It benefits from previous information in contrast with other methods that use gradients or Hessians. Bayesian optimization uses prior, which is the probabilistic model of the objective function, and the acquisition function, which defines the next points to evaluate. In this comparison, we used the Gaussian process as prior and the expected improvement as acquisition function. We use the \texttt{bayesopt} function from MATLAB.

- The Bayesian Adaptive Direct Search (BADS) \([14]\) uses a Mesh Adaptive Direct Search (MADS) \([8]\) with a Bayesian optimization in the search step. The search step performs the discovering of the points with the intention of inserting domain-specific information and improving the quality of the points. When the search stage does not find suitable points, the poll stage of MADS broadly evaluates new points. The poll step is a computational expensive process and explores the objective function’s shape for new points. We use the MATLAB implementation provided by Acerbi and Ji \([14]\)\(^5\).

- The Simulated Annealing (SA) mimics the process of annealing in metal. The temperature value dictates the probability function of the distance to a new random point based on the current point, while the distance to new points reduces as the temperature decreases with time. This procedure does not limit the new points to minimal points, this means that new points can have higher objective function value, helping to avoid the local minimum. We use the \texttt{simulannealbnd} function from MATLAB.

- The Grid Search (GS) algorithm consists of testing a combination of values for all the hyperparameters. This is a naive method that needs \(N^M\) evaluations where \(M\) is the number of hyperparameters and \(N\) is the number of values for each hyperparameter. The advantage of this method is that it can be easily parallelized, however, the method itself does not define a maximum number of evaluations. Therefore, we have split the search space based on the lower and upper bounds of \(C\) and \(\gamma\) to obtain the same number of evaluations allowed for other methods. We implemented the method in MATLAB.

- The Random Search (RS) algorithm starts by generating a set of points in the pre-defined search space. Subsequently, it gets the minimum among them and refines the search around it. It repeats this process until achieving the stop criterion. We implemented the method in MATLAB.

As one may see we do not compare the proposed method with derivative or evolutionary methods. In the framework of

\(^5\)Available at https://github.com/lacerbi/bads
BBO, which includes the hyperparameter optimization problem, the derivative is unavailable, making the former unsuitable, while the later consists of global search heuristic methods (e.g., genetic algorithm and particle swarm optimization) in which the emphasis is on finding a decent global solution instead of finding an accurate local solution that provides a stopping criterion with some assurance of optimality.

We have evaluated the hinge-loss output value and the classification accuracy in the test set after 100 evaluations using a common configuration for all methods and the Ortho-MADS with the Nelder-Mead search. We used the functions *fitcsvm* and *fitcecoc* from the MATLAB Statistics and Machine Learning Toolbox to train the SVM for the binary or multiclass cases respectively, and the function *predict* to evaluate the learned model in the test sets. We defined the lower bound of the search space as $L = [0.01, 0.01]$ and its upper bound as $U = [100, 100]$. Considering that the starting point plays an important role in BBO optimization, we have compared the methods using six different initialization points $x_0 = (C, \gamma) = [(C, 0.5), (10, 10), (50, 50), (90, 90), (1, 90), (90, 1)]$. We have used the pre-defined validation set when it was available on the hinge-loss function, and for the cases where there was not a pre-defined validation set, we have used the training set with a stratified 3-fold cross validation strategy. For the RS and the GS, there is no pre-defined initialization point, and we have set a linear search of 100 iterations, i.e., $C \in [1, 10 \times 1, 10 \times 2, \ldots, 10 \times 9]$ and $\gamma \in [1, 10 \times 1, 10 \times 2, \ldots, 10 \times 9]$.

Tab. 2 shows the mean accuracy, standard deviation and maximum accuracy for all hyperparameter tuning methods. Both Ortho-MADS and BADS have shown to be more consistent to achieve a competitive mean accuracy for all datasets. The BO, SA, and RS present competitive results in ten datasets, and the GS shows competitive results in six datasets. The BO, SA, and RS present competitive results in ten datasets, and the GS shows competitive results in six datasets. Tab. 3 presents the mean loss $L$, its standard deviation and the minimum loss $L_{\text{min}}$. For most of the datasets, the behavior is similar to that presented in Tab. 2. The Bayesian, RS, and GS provided the lowest $L_{\text{min}}$ for the Splice dataset. However, this result does not necessarily translate into high accuracy because in this case, the function might be overfitting.

Another aspect to analyze is the convergence rate and the trajectory of the algorithms. For all datasets, the Ortho-MADS presents a competitive convergence rate, and in many cases (as exemplified in Figs. 2a and 2b), both the Ortho-MADS and the BADS (that also uses the MADS algorithm) have the fastest convergence rate to reach a minimum. In some cases, the Ortho-MADS may not have the fastest convergence, as depicted in Figs. 3a and 3b. However, it is still competitive with other methods, and may pass over its convergence rate to reach a lower local minimum, as shown in Fig. 3b. The BADS achieved the best results overall, with better accuracy in eight out of thirteen datasets (Astro, Car, DNA, Letter, Pendigits, Splice, Svmguide4, and USPS), and competitive accuracy for all other datasets with a low standard deviation. The Ortho-MADS has the second-best results, with best results in five out of thirteen datasets (Astro, Pendigits, Splice, Svmguide4, and USPS), and competitive accuracy for all other datasets with a low standard deviation.

The Bayesian and the SA methods presented the best results in four out of thirteen datasets, but sometimes the results are not competitive, as observed for the Splice and Svmguide4 datasets when applying the Bayesian method, and for the DNA, Splice, and Svmguide4 datasets when using the SA method. In addition, both methods have presented standard deviations higher than Ortho-MADS and BADS. The RS achieved the best accuracy in the Madelon dataset, however, RS depends on the randomness that leads to more iterations to achieve a good result, and GS depends on the grid choice, which creates unreachable spaces.

A close look at the Ortho-MADS standard deviation (this extends to other methods as well) from Tab. 2 indicates that in some cases we do not reach the best point, and this fact could be related to the choice of the starting point, as it has an important influence on the result or the method randomness that falls into a local minimum. Figs. 3a and 3b exemplify the influence of the starting point on the effectiveness of the algorithms. From the starting point $x_0 = (0.5, 0.5)$, depicted in Fig. 3a, the Ortho-MADS, Bayesian, BADS, SA, and GS achieved worse objective function value when compared to the starting point $x_0 = (90, 1)$ from Fig. 3b.

We generate an ordering of the methods based on the mean, maximum and worst accuracy reported in Tab. 2. Tab. 4 summarizes the comparison between all methods through an average ranking [35] according to the measured accuracy mean, worst case, and best case. The BADS has the best rank among all considered methods, followed by the Ortho-MADS, Bayesian, SA, GS, and RS. The most consistent methods are the BADS and the Ortho-MADS, ranking 1 and 2 for the best mean and the best maximum accuracy, and 6 and 5 for the worst mean accuracy respectively.

### 4.3. Proposed Approach

The Ortho-MADS has shown to be competitive with other state-of-the-art methods (Tab. 2) to tune the hyperparameters of the SVM with Gaussian kernel, however, we propose to combine the Ortho-MADS convergence properties with two different search algorithms to enhance the stability and reachability, and to use the mesh size as stopping criterion. The NM search strategy leads to a faster convergence when compared to regular Ortho-MADS search strategy, i.e., it requires fewer function evaluations to reach the pre-defined minimum mesh size. The VNS explores regions far from the incumbent (increasing the number of function evaluations), helping to escape from an eventual undesired local minimum. The VNS counterbalance the NM fast convergence; however, it explores more regions from the search space and it mitigates the initial point influence. The Ortho-MADS attempts to find an accurate local solution and using the mesh-size as stopping criterion translates into stopping the algorithm when achieving a local solution (mesh-size) that satisfies the user needs.

We evaluate the accuracy and the number of function evaluations for the Ortho-MADS with two search algorithms (NM
Table 2: Mean accuracy, standard deviation and maximum accuracy for all hyperparameter optimization methods in 13 datasets. The best results are underlined.

| Method   | Astro       | Car         | DNA          | Letter       | Madelon     | Pendigits   | Protein     | Satimage    | Shuttle     | Splice       | Svmguide4   | USPS        | Vowels      |
|----------|-------------|-------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|--------------|-------------|-------------|-------------|
| Ortho-MADS | 0.970±0.001 | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.969±0.000 | 0.970±0.000 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 |
| Bayesian  | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.969±0.000 | 0.970±0.000 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 |
| SA       | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.969±0.000 | 0.970±0.000 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 |
| RS       | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.969±0.000 | 0.970±0.000 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 |
| GS       | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.969±0.000 | 0.970±0.000 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 |
| BADS     | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.969±0.000 | 0.970±0.000 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 |

Table 3: Mean loss ($L$), its standard deviation and minimum loss ($L_{\text{min}}$) for all hyperparameter optimization methods in 13 datasets. The best results are underlined.

| Method   | Astro       | Car         | DNA          | Letter       | Madelon     | Pendigits   | Protein     | Satimage    | Shuttle     | Splice       | Svmguide4   | USPS        | Vowels      |
|----------|-------------|-------------|--------------|--------------|-------------|-------------|-------------|-------------|-------------|--------------|-------------|-------------|-------------|
| Ortho-MADS | 0.970±0.001 | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.969±0.000 | 0.970±0.000 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 |
| Bayesian  | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.969±0.000 | 0.970±0.000 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 |
| SA       | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.969±0.000 | 0.970±0.000 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 |
| RS       | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.969±0.000 | 0.970±0.000 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 |
| GS       | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.969±0.000 | 0.970±0.000 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 |
| BADS     | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.970±0.000 | 0.969±0.000 | 0.970±0.000 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 | 0.955±0.004 |

Figure 2: Satimage dataset with $x_0 = [0.5, 0.5]$: (a) convergence plot comparison; (b) trajectory plot comparison.
and VNS, both with non-opportunistic strategy\(^6\), using the minimum mesh size as stopping criterion. Because the MADS direction is randomly chosen, for each dataset we run 50 times using the following default configuration (empirically defined): the lower bound \((L)\) as \([0.01, 0.01]\), the upper bound \((U)\) as \([100.01, 100.01]\), the starting point \((x_0)\) as \([50, 50]\), the minimum mesh size \((\delta_{\text{min}})\) as 0.009 (that corresponds to three shrinking executions from the initial mesh size), and the perturbation amplitude is \(\xi = 0.25\). From the initial configuration, we further analyze the impact of changing the starting point \(x_0\), the minimum mesh size \(\delta_{\text{min}}\) and the perturbation amplitude \(\xi\).

Tab. 5 presents the mean accuracy, the standard deviation, and the maximum and median accuracy. For each measure we also present the corresponding number of function evaluations. As stated before, the incorporation of VNS in the search step aids the Ortho-MADS to escape from local minimum, which may lead to better results, and the NM counterbalance the number of function evaluations needed.

Tab. 5 presents the results of the proposed approach, and comparing with Tab. 2 we observe several improvements. Using the minimum mesh-size as stopping criterion may avoid unnecessary function evaluations. In our previous experiment\(^6\), for each search iteration, the algorithm does not finish when a better incumbent is found. It only finishes when all points are evaluated.

\(^6\)For each search iteration, the algorithm does not finish when a better incumbent is found. It only finishes when all points are evaluated.

---

Table 4: Average ranking (AR) considering the best mean accuracy, the worst mean accuracy and the maximum accuracy for the 13 datasets [35].

| Algorithm   | Best Mean Accuracy Rank | Worst Mean Accuracy Rank | Best Maximum Accuracy Rank |
|-------------|-------------------------|--------------------------|---------------------------|
| BADS        | 1.77                    | 1                        | 2.08                      | 1 |
| Ortho-MADS  | 2.07                    | 2                        | 2.69                      | 2 |
| Bayesian    | 2.31                    | 3                        | 3.23                      | 4 |
| SA          | 2.78                    | 4                        | 2.92                      | 3 |
| GS          | 3.54                    | 5                        | 4.38                      | 6 |
| RS          | 3.77                    | 6                        | 3.92                      | 5 |

Table 5: Accuracy | No. of function evaluations for the proposed approach. Mean, standard deviation, maximum (and the number of function evaluations regarding the best result), and median for all 13 datasets. * indicates higher mean accuracy.

| Dataset     | Mean | Std  | Max  | Median | No. of function evaluations |
|-------------|------|------|------|--------|-----------------------------|
| Astro       | 0.968 | 0.001 | 0.971 | 0.969 | 162                         |
| Car*        | 0.720 | 0.027 | 0.829 | 0.707 | 137                         |
| DNA         | 0.942 | 0.000 | 0.942 | 0.942 | 51                          |
| Letter*     | 0.954 | 0.000 | 0.954 | 0.954 | 111                         |
| Madelon*    | 0.598 | 0.007 | 0.607 | 0.602 | 75                          |
| Pendigits   | 0.972 | 0.000 | 0.973 | 0.972 | 113                         |
| Protein     | 0.691 | 0.000 | 0.691 | 0.691 | 73                          |
| Satimage    | 0.913 | 0.000 | 0.913 | 0.913 | 108                         |
| Shuttle*    | 0.999 | 0.000 | 0.999 | 0.999 | 73                          |
| Splice      | 0.897 | 0.001 | 0.901 | 0.897 | 118                         |
| Svmguide4*  | 0.847 | 0.008 | 0.865 | 0.848 | 112                         |
| USPS        | 0.943 | 0.001 | 0.945 | 0.943 | 97                          |
| Vowels      | 0.624 | 0.011 | 0.643 | 0.621 | 123                         |

Figure 3: Astroparticle dataset comparison convergence plot with different starting points: (a) at \(x_0 = [0.5, 0.5]\); (b) at \(x_0 = [90, 1]\).
0.641 to 0.643, however, with an increase in the number of function evaluations (from 100 to 128 of mean).

Here again, we generate an ordering of the methods but now replacing the Ortho-Mads by the proposed approach. Tab. 6 summarizes the comparison between all methods through an average ranking [35] according to the measured accuracy mean, worst case, and best case. The proposed approach has the best rank among all considered methods, followed by the BADS, Bayesian, SA, GS, and RS. The most consistent methods are the proposed approach and the BADS, ranking 1 and 2 for the best mean and the best maximum accuracy, and 5 and 6 for the worst mean accuracy respectively.

Table 6: Average ranking (AR) considering the best mean accuracy, the worst mean accuracy and the maximum accuracy for the 13 datasets [35].

| Algorithm      | Best Mean Accuracy | Worst Mean Accuracy | Best Maximum Accuracy |
|----------------|--------------------|--------------------|-----------------------|
|                | t  | Rank  | t  | Rank  | t  | Rank  |
| Proposed Approach | 1.85 | 1     | 4.31 | 5     | 2.15 | 1     |
| BADS           | 1.92 | 2     | 4.46 | 6     | 2.38 | 2     |
| Bayesian        | 2.61 | 3     | 3.38 | 4     | 3.46 | 4     |
| SA             | 3.08 | 4     | 3.08 | 3     | 3.15 | 3     |
| GS             | 3.85 | 5     | 2.31 | 2     | 4.46 | 6     |
| RS             | 4.15 | 6     | 2.00 | 1     | 4.08 | 5     |

The Friedman rank sum test shows a p-value of 0.00024, and Tab. 7 presents the Nemenyi test using Tabs. 2 and 5. The results indicate that the proposed approach (Ortho-MADS+VNS+NM) presents similar results to BADS, and the concordance of results between Ortho-MADS and BADS are smaller than the proposed approach. We can conclude that both BADS and Ortho-MADS+VNS+NM present superior performance compared to other methods. Furthermore, the advantage of the proposed approach when compared to BADS is the false minimum avoidance and the stopping criterion. Fig. 4 depicts the critical difference graph comparing all methods, illustrating the results of Tab. 7. The confidence interval is 95% for the null hypotheses of $H_0 : \theta_i = \theta_j$ and the alternative hypotheses of $H_1 : \theta_i \neq \theta_j$. Considering the used confidence interval the p-values close to one do not reject the null hypotheses, while the p-values close to zero reject the $H_0$ and do not reject $H_1$.

Table 7: Nemenyi test between all methods using the 13 datasets.

| Proposed Approach | Ortho-MADS | Bayesian | SA | RS | GS | Ortho-MADS | Bayesian | SA | RS | GS | Ortho-MADS | Bayesian | SA | RS | GS |
|-------------------|------------|----------|----|----|----|------------|----------|----|----|----|------------|----------|----|----|----|
|                   | 0.9637     | 0.8446   | 0.3597 | 0.0037 | 0.0497 | 1.0000       | 0.9780   | 0.8844 | 0.4163 | 0.0052 | 0.0642 |
| Mean accuracy     | 0.738       | 0.729    | 0.716   | 0.036  | 0.756   | 0.738       | 0.729    | 0.716   | 0.036  | 0.756   |
| Std               | 0.054      | 0.042    | 0.029   | 0.035  | 0.084   | 0.054      | 0.042   | 0.029   | 0.035  | 0.084   |
| Max               | 0.829      | 0.829    | 0.829   | 0.805  | 0.829   | 0.829      | 0.829   | 0.829   | 0.829  | 0.829   |
| Median            | 0.707      | 0.707    | 0.707   | 0.707  | 0.707   | 0.707      | 0.707   | 0.707   | 0.707  | 0.707   |

Tab. 9 shows that by decreasing $\delta_{min}$, the proposed approach needs more function evaluations to reach the desired local minimum, as the mesh size reduction occurs in sequence (it is not possible to execute two mesh reduction operations in the same Ortho-MADS with NM and VNS iteration). For $\delta_{min(C)} = 9e - 1$ we need fewer function evaluations to reach the stopping criterion, however, the maximum accuracy achieved was lower than the results from $\delta_{min(C)} = 9e - 3$. A smaller minimum mesh size, $\delta_{min(C)} = 9e - 7$, does not guarantee a good performance, but for sure it increases the number of function evaluations needed to reach the stopping criterion. In this case the model can discard good points or over-fit the model.

Table 8: Mean accuracy, standard deviation, maximum accuracy and median accuracy for the Car dataset for different permutation amplitudes.

We choose the Car dataset (as it has the largest standard deviation among all datasets reported in Tab. 5) to analyze the impact of changing the VNS, the starting point and the minimum mesh size. We start considering different perturbation amplitudes $\xi = [0.25, 0.5, 0.75, 0.9]$, and Tab. 8 shows the results after 50 runs for each $\xi$. By increasing $\xi$, the algorithm can reach regions more distant from the current incumbent, however, it requires more function evaluations to achieve a high mean accuracy. A large $\xi$ does not translate into a better accuracy, as increasing $\xi$ it creates sparser points to evaluate which may skip good regions, as the results for $\xi = 0.75$ and $\xi = 0.9$ indicate.

We have also evaluated the influence of the starting point for the Car dataset using five pre-defined and five random starting points $x_0 = (0.5, 0.5), (50, 50), (90, 90), (1, 90), (90, 1), (70.93, 75.21), (50, 60, 64.29), (25.21, 79.05), (89.59, 13.49), (2.37, 57.91))$. Tab. 10 shows that the mean value may be similar to the solution without VNS (as shown in Tab. 2), however, we could reach at least 0.805 of accuracy for all starting points, which is higher than the best solution from Tab. 2 (0.780 using BADS). Thus, the VNS mitigates the starting point effect and the MADS randomness. Fig. 5 exemplifies a comparison example between the Ortho-MADS with and without the VNS, both using the same starting point at $x_0 = (50, 50)$ and the number of
function evaluations. The Ortho-MADS without VNS reached $L_{\text{min}} = 1.1904$ and the accuracy of 0.7073 on the test set, while the Ortho-MADS with VNS reached $L_{\text{min}} = 1.1876$ and the accuracy of 0.8048.

| $x_0 = (C, \gamma)$ | Mean | Max | Min |
|---------------------|------|-----|-----|
| (0.5, 0.5)          | 0.717 | 1.191 | 0.805 | 1.187 | 0.780 | 1.185 |
| (50, 50)            | 0.735 | 1.189 | 0.829 | 1.187 | 0.805 | 1.183 |
| (90, 90)            | 0.760 | 1.190 | 0.805 | 1.188 | 0.732 | 1.183 |
| (1, 90)             | 0.690 | 1.192 | 0.805 | 1.194 | 0.780 | 1.185 |
| (90, 1)             | 0.721 | 1.192 | 0.805 | 1.196 | 0.707 | 1.118 |
| (70.93, 75.21)      | 0.698 | 1.192 | 0.829 | 1.188 | 0.707 | 1.186 |
| (50.60, 64.29)      | 0.728 | 1.191 | 0.805 | 1.196 | 0.780 | 1.184 |
| (25.21, 79.05)      | 0.726 | 1.190 | 0.805 | 1.189 | 0.707 | 1.187 |
| (89.59, 13.49)      | 0.727 | 1.191 | 0.805 | 1.193 | 0.732 | 1.185 |
| (2.37, 57.91)       | 0.727 | 1.190 | 0.805 | 1.189 | 0.780 | 1.186 |

Table 10: Mean accuracy, maximum accuracy and minimum accuracy and the corresponding $L$ value for the Car dataset for different starting points $x_0$.

![Convergence plot comparison between Ortho-MADS with and without VNS for the Car dataset.](image)

Table 11: Best maximum accuracy and No. of function evaluations for the Car dataset.

| Approach  | Accuracy | # Evaluations |
|-----------|----------|---------------|
| Proposed Approach | 0.607 | 70 |
| Bayesian  | 0.568 | 30 |
| BADS     | 0.602 | 106 |
| GS       | 0.571 | 57 |
| RS       | 0.603 | 30 |
| SA       | 0.601 | 19 |

Table 12: Best maximum accuracy and No. of function evaluations for the Car dataset with stopping condition of either 0.8 of accuracy or 1,000 function evaluations.

**5. Conclusion**

We presented the Ortho-MADS with the Nelder-Mead and the Variable Neighborhood Search (VNS), and the mesh size as a stopping criterion for tuning the hyperparameters $C$ and $\gamma$ of a SVM with Gaussian kernel. We have shown on benchmark datasets that the proposed approach outperforms widely used and state-of-the-art methods for model selection. The alternation between the search and pool stage provides a robust performance, and the use of two different search methods attenuate the randomness of the MADS and the starting point choice. Besides that, the proposed approach has convergence proof, and using the mesh size as stopping criterion gives to the user the possibility of setting a specific local minimum region instead of using the number of evaluations, time, or fixed-size grid as a stopping criterion. In the cases where a test set is available, the evolution of the mesh size during the BBO iterations can help identify under and over fitting behaviors.

The proposed approach gives the user the flexibility of choosing parameters to explore different strategies and situations. From our experiments, we recommend starting with the proposed default configuration, and from there the user can customize the Ortho-MADS parameters if necessary, which may improve the quality of the results. Therefore, we strongly recommend using the Ortho-MADS with NM and VNS search for tuning the hyperparameters of a SVM with Gaussian kernel, which also provides many other functionalities. For future work, we expect to analyze the NOMAD with SVM variations, which includes incremental formulations and different kernels.
References

[1] B. E. Boser, I. M. Guyon, V. N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT’92, ACM, New York, NY, USA, 1992, pp. 144–152.

[2] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (1995) 273–297.

[3] Chih-Wei Hsu, Chih-Jen Lin, A comparison of methods for multiclass support vector machines, IEEE Transactions on Neural Networks 13 (2002) 412–425.

[4] N. Cristianini, J. Shawe-Taylor, An introduction to support vector machines: and other kernel-based learning methods, Cambridge University Press, Cambridge, 2000.

[5] J. Wainer, G. Cawley, Empirical Evaluation of Resampling Procedures for Optimising SVM Hyperparameters, Journal of Machine Learning Research 18 (2017) 1–35.

[6] C. Audet, W. Hao, Derivative-Free and Blackbox Optimization, Springer Series in Operations Research and Financial Engineering, Springer International Publishing, Berlin, 2017.

[7] J. A. Nelder, R. Mead, A simplex method for function minimization, The Computer Journal 7 (1965) 308–313.

[8] C. Audet, J. E. Dennis, Jr., Mesh adaptive direct search algorithms for constrained optimization, SIAM J. on Optimization 17 (2006) 188–217.

[9] Jiancheng Sun, Chongxun Zheng, Xiaohe Li, Yatong Zhou, Analysis of the Distance Between Two Classes for Tuning SVM Hyperparameters, IEEE Transactions on Neural Networks 21 (2010) 305–318.

[10] G. Chen, W. Floreiro-Salinas, D. Li, Simple, fast and accurate hyperparameter tuning in Gaussian-kernel SVM, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 348–355.

[11] K.-M. Chung, W.-C. Kao, L.-L. Wang, C.-J. Lin, Radius Margin Bounds for Support Vector Machines with the RBF Kernel, Neural Computation 15 (2003) 2643–2681.

[12] C. Gold, A. Holub, P. Sollich, Bayesian approach to feature selection and parameter tuning for support vector machine classifiers, Neural Networks 18 (2005) 693–701.

[13] J. Snoek, H. Larochelle, R. P. Adams, Practical Bayesian optimization of machine learning algorithms, in: F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 2951–2959.

[14] L. Acerbi, W. Ji, Practical Bayesian optimization for model fitting with bayesian adaptive direct search, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 1836–1846.

[15] C.-C. Chang, S.-H. Chou, Tuning of the hyperparameters for L2-loss SVMs with the RBF kernel by the maximum-margin principle and the jackknife technique, Pattern Recognition 48 (2015) 3983–3992.

[16] C. Audet, A Survey on Direct Search Methods for Blackbox Optimization, Journal of Computational Theory 92 (1999) 133–158.

[17] C. Audet, C. Tribes, Mesh-based nelder-mead algorithm for inequality constrained optimization, Computational Optimization and Applications (2018).

[18] M. A. Abramson, C. Audet, J. E. Dennis, Jr., S. L. Digabel, Orthomads: A deterministic mads instance with orthogonal directions, Society for Industrial and Applied Mathematics Journal on Optimization 20 (2009) 948–966.

[19] C. Audet, V. Béchard, S. L. Digabel, Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search, Journal of Global Optimization 41 (2008) 299–318.

[20] M. A. Abramson, C. Audet, J. E. D. Jr., S. L. Digabel, Orthomads: A deterministic mads instance with orthogonal directions, Society for Industrial and Applied Mathematics Journal on Optimization 20 (2009) 948–966.

[21] C. Audet, S. Le Digabel, C. Tribes, Dynamic scaling in the mesh adaptive direct search algorithm for blackbox optimization, Optimization and Engineering 17 (2016) 333–358.

[22] C. Audet, C. Tribes, Dynamic scaling in the mesh adaptive direct search algorithm for blackbox optimization, Optimization and Engineering 16 (2015) 449–467.

[23] J. J. Moré, S. M. Wild, Benchmarking Derivative-Free Optimization Algorithms, SIAM Journal on Optimization 20 (2009) 172–191.

[24] T. G. Dietterich, G. Bakiri, Solving Multiclass Learning Problems via Error-Correcting Output Codes, Journal of Artificial Intelligence Research (1994).

[25] L. Rosasco, E. D. Vito, A. Caponnetto, M. Piana, A. Verri, Are Loss Functions All the Same?, Neural Computation 16 (2004) 1063–1076.

[26] W. Zegel, N. Essaddam, J. Brimberg, A New VNS Metaheuristic Using MADS as a Local Optimizer, Journal of Multi-Criteria Decision Analysis 19 (2012) 257–262.

[27] P. Hansen, N. Mladenović, Variable neighborhood search, Computers & Operations Research 24 (1997) 1097–1100.

[28] H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 2951–2959.