PADOVA UNIVERSITY PRESS

Dolomites Research Notes on Approximation

Volume 17 · 2024 · Pages 48–55

Comparing Deterministic and Statistical Optimization Techniques for the Shape Parameter Selection in RBF Interpolation

Roberto Cavoretto[1,4] · Alessandra De Rossi[1,4] · Adeeba Haider[1,4] · Sandro Lancellotti[1,4]

Dedicated to our colleague and friend Len Bos on the occasion of his retirement

Abstract

In this paper, we compare two novel approaches for effectively determining the optimal value of the shape parameter in Radial Basis Function (RBF) interpolation, a crucial factor for numerical method accuracy. We analyze the results of applying the deterministic Leave-One-Out Cross Validation (LOOCV) method in combination with Lipschitz Global Optimization with Pessimistic Improvement (GOPI) and Optimistic Improvement (GOOI), contrasting them with the statistical Bayesian Optimization (BO). Both techniques yield similar validation errors, underlining their effectiveness in shape parameter search. However, the deciding factor in technique selection lies in computational time, which is contingent upon the cardinality of the interpolation set.

Keywords: RBF interpolation, Shape parameter, Global optimization, Bayesian Optimization

1 Introduction

Over the past few decades, Radial Basis Functions (RBFs) have emerged as vital tools in scientific computing. Their applications span diverse areas, including interpolation and approximation, machine learning, and solution of partial differential equations [28, 21]. RBF interpolation aims to reconstruct unknown functions based on available data, sometimes exhibiting exponential convergence errors. In the context of RBFs, the identification of an appropriate shape (or scale) parameter has posed a significant challenge over the years, since the shape parameter significantly influences the accuracy of corresponding numerical methods, see e.g. [9, 11].

For the selection of scale parameter, conventional methods often rely on empirical trial and error to assign a specific value. However, these approaches prove inadequate for certain applications. On the other hand, cross validation techniques are fundamental in statistical analysis, with the Leave-One-Out Cross Validation (LOOCV) being a prominent method, particularly when selecting the optimal shape parameter for RBF in scattered data interpolation, as suggested by Rippa [24].

In this article, we focus on the LOOCV method combined with Lipschitz Global Optimization with Pessimistic Improvement (GOPI) and Global Optimization with Optimistic Improvement (GOOI) techniques, thus resulting in the so-called LOOCV-GOOI and LOOCV-GOPI methods, introduced in [7] and later further extended in [4]. The LOOCV-GOOI method alternates between local and global optimization steps, halting when a local descent fails to find a new promising minimizer and a local stopping condition is met. The LOOCV-GOPI method instead first continues the local optimization until the desired accuracy is achieved, and then shifts to the global phase incorporating the local trials [26]. These methods offer substantial benefits, including enhanced generalization performance and improved predictive accuracy, making them in some specific situations superior choices for our current analysis. The main goal of this work is nevertheless to provide a comparison between the improved LOOCV-based optimization method and the Bayesian Optimization (BO) technique [19, 27]. BO is a global optimization method renowned for its efficacy in determining the optimal shape parameter value for scattered data approximation based on RBFs [5, 6]. More precisely, this technique is a strategy arising from machine learning to streamline the optimization of intricate functions that are challenging to evaluate directly. It is particularly effective in hyperparameter tuning, where it bypasses the exhaustive computation and assessment of suboptimal parameter sets. By leveraging statistical models, BO predicts and prioritizes the evaluation of parameter combinations, focusing on those with the highest potential, thus enhancing the efficiency of the optimization process.

[1] Department of Mathematics “Giuseppe Peano”, University of Torino, via Carlo Alberto 10, 10123 Torino, Italy

[4] Member of the INdAM Research group GNCS

This approach not only saves computational resources but also accelerates the convergence towards optimal solutions in complex optimization landscapes.

The remaining paper is organized into several sections as follows. Section 2 delves into kernel-based interpolation, providing a comprehensive analysis of its methodologies and results. Section 3 is dedicated to optimizers, briefly describing the deterministic LOOCV technique (Subsection 3.1), the Lipschitz GOOI and GOPI strategies (Subsection 3.2), and the statistical BO model-based approach for global optimization (Subsection 3.3). Following these, Section 4 presents several numerical results, showcasing the experimental findings obtained from the comparison of LOOCV-GOOI and LOOCV-GOPI with BO methods. Finally, Section 5 concludes the paper by highlighting the potential of the different methods and the cases where they are recommended.

2 Kernel-based Interpolation

Kernel-based methods are powerful and effective tools for scattered data interpolation. In this section we introduce some basic notations and results for radial kernel-based interpolation. For further details on basic theory, we refer the reader to [3, 10, 29].

Given a compact domain $\Omega \subset \mathbb{R}^d$, we assume that the N distinct data points (or nodes) are defined by the set $X = \{\boldsymbol{x}_i\}_{i=1}^N \subseteq \Omega$. The corresponding data values associated with $\boldsymbol{x}_i$, $i = 1,\dots,N$, are given by measurements $y_i \in \mathbb{R}$, or simply obtained by sampling some (unknown) function $f : \Omega \to \mathbb{R}$ at $\boldsymbol{x}_i$, i.e. $y_i = f(\boldsymbol{x}_i) \in \mathbb{R}$. In this framework, we want to solve a scattered data interpolation problem finding a function $s_X : \Omega \to \mathbb{R}$ satisfying the following N interpolation conditions

$$s_X(\boldsymbol{x}_i) = y_i, \qquad i = 1,\dots,N. \tag{1}$$

We express the interpolant s_X as a linear combination of kernels $\kappa_\varepsilon : \Omega \times \Omega \to \mathbb{R}$ depending on the so-called *shape parameter* $\varepsilon > 0$, i.e.

$$s_X(\boldsymbol{x}) = \sum_{j=1}^{N} c_j \kappa_\varepsilon(\boldsymbol{x}, \boldsymbol{x}_j), \qquad \boldsymbol{x} \in \Omega. \tag{2}$$

Assuming that the kernel κ_ε is symmetric and strictly positive definite (SPD), the interpolation (or kernel) matrix K_ε with the entries $(\mathsf{K}_\varepsilon)_{ij} = \kappa_\varepsilon(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $i, j = 1,\dots,N$, is positive definite for any set X. Thus, the coefficients c_j in (2) are uniquely determined by enforcing the interpolation conditions (1) and can be obtained by solving the symmetric linear system

$$\mathsf{K}_\varepsilon \boldsymbol{c} = \boldsymbol{y}, \tag{3}$$

where $\boldsymbol{c} = (c_1,\dots,c_N)^T$ and $\boldsymbol{y} = (y_1,\dots,y_N)^T$.

Associated with the kernel κ_ε in (2) we may define a SPD RBF $\phi : \mathbb{R}_0^+ \to \mathbb{R}$ such that

$$\kappa_\varepsilon(\boldsymbol{x}, \boldsymbol{x}_j) = \phi_\varepsilon(||\boldsymbol{x} - \boldsymbol{x}_j||_2) = \phi_\varepsilon(r) := \phi(\varepsilon r), \qquad \forall \boldsymbol{x}, \boldsymbol{x}_j \in \Omega,$$

where $||\cdot||_2$ denotes the Euclidean norm on $\mathbb{R}^d$. Moreover, we know that the choice of a “good” value of ε is generally a crucial task for kernel-based interpolation, but at the same time also a big issue (see e.g. [7, 8, 12], or [10, Chapter 14]). Some examples of popular SPD RBFs (or radial kernels) together with their smoothness degrees and abbreviations are listed as follows (see [9, 29]):

$$\phi_\varepsilon(r) = \begin{cases} \exp(-\varepsilon^2 r^2), & \text{Gaussian } C^\infty, & \text{GA} \\ (1+\varepsilon^2 r^2)^{-1/2}, & \text{Inverse MultiQuadric } C^\infty, & \text{IMQ} \\ \exp(-\varepsilon r)(\varepsilon^3 r^3 + 6\varepsilon^2 r^2 + 15\varepsilon r + 15), & \text{Matérn } C^6, & \text{M6} \\ \max(1-\varepsilon r, 0)^6 (35\varepsilon^2 r^2 + 18\varepsilon r + 3), & \text{Wendland } C^4, & \text{W4} \\ \max(1-\varepsilon r, 0)^4 (4\varepsilon r + 1), & \text{Wendland } C^2, & \text{W2} \end{cases}$$

When we solve the linear system (3), the solution is often very sensitive to changes in the data. This sensitivity is additionally influenced by the determination of ε. A way to evaluate the computational stability of a kernel-based interpolant consists in numerically calculate the 2-norm condition number (cond_2) of the kernel matrix K_ε, i.e.,

$$\text{cond}_2(\mathsf{K}_\varepsilon) = ||\mathsf{K}_\varepsilon||_2 ||\mathsf{K}_\varepsilon^{-1}||_2 = \frac{\lambda_{\max}}{\lambda_{\min}}, \tag{4}$$

$\lambda_{\max}$ and $\lambda_{\min}$ being the largest and smallest eigenvalues of the symmetric and positive definite matrix K_ε.

Moreover, for the kernel κ_ε there exists the so-called *native space*, which is a Hilbert space $\mathcal{N}_{\kappa_\varepsilon}(\Omega)$ with inner product $(\cdot,\cdot)_{\mathcal{N}_{\kappa_\varepsilon}(\Omega)}$ in which the kernel κ_ε is reproducing, i.e., for any $f \in \mathcal{N}_{\kappa_\varepsilon}(\Omega)$ we have the identity $f(\boldsymbol{x}) = (f, \kappa_\varepsilon(\cdot, \boldsymbol{x}))_{\mathcal{N}_{\kappa_\varepsilon}(\Omega)}$, with $\boldsymbol{x} \in \Omega$. Then, if we introduce a pre-Hilbert space $H_{\kappa_\varepsilon}(\Omega) = \text{span}\{\kappa_\varepsilon(\cdot, \boldsymbol{x}),\ \boldsymbol{x} \in \Omega\}$, with reproducing kernel κ_ε and equipped with the bilinear form $(\cdot,\cdot)_{H_{\kappa_\varepsilon}(\Omega)}$, the native space $\mathcal{N}_{\kappa_\varepsilon}(\Omega)$ of κ_ε is its completion w.r.t. the norm $||\cdot||_{H_{\kappa_\varepsilon}(\Omega)} = \sqrt{(\cdot,\cdot)_{H_{\kappa_\varepsilon}(\Omega)}}$. Specifically, for all $f \in H_{\kappa_\varepsilon}(\Omega)$ we have $||f||_{\mathcal{N}_{\kappa_\varepsilon}(\Omega)} = ||f||_{H_{\kappa_\varepsilon}(\Omega)}$ (see [29]). Then, we can now provide an error bound in terms of the *power function* $P_{\kappa_\varepsilon, X}$ (see e.g. [9, Theorem 14.2]):

Theorem 2.1. *Let $\Omega \subseteq \mathbb{R}^d$, $\kappa_\varepsilon \in C(\Omega \times \Omega)$ be strictly positive definite on $\mathbb{R}^d$, and suppose that $X = \{\boldsymbol{x}_i\}_{i=1}^N$ has distinct points. Then, for all $f \in \mathcal{N}_{\kappa_\varepsilon}(\Omega)$, we have*

$$|f(\boldsymbol{x}) - s_X(\boldsymbol{x})| \le P_{\kappa_\varepsilon,X}(\boldsymbol{x}) ||f||_{\mathcal{N}_{\kappa_\varepsilon}(\Omega)}, \quad \boldsymbol{x} \in \Omega.$$

The generic error estimate of Theorem 2.1 can further be refined as in [9, Theorem 14.5]:

Theorem 2.2. *Let $\Omega \subseteq \mathbb{R}^d$ be bounded and satisfy an interior cone condition. Suppose that $\kappa_\varepsilon \in C^{2k}(\Omega \times \Omega)$ is symmetric and strictly positive definite. Then, for all $f \in \mathcal{N}_{\kappa_\varepsilon}(\Omega)$, there exist constants h_0, $C > 0$ (independent of $\boldsymbol{x}$, f and κ_ε) such that*

$$|f(\boldsymbol{x}) - s_X(\boldsymbol{x})| \le C h_{X,\Omega}^k \sqrt{C_{\kappa_\varepsilon}(\boldsymbol{x})}\, ||f||_{\mathcal{N}_{\kappa_\varepsilon}(\Omega)},$$

provided $h_{X,\Omega} \le h_0$. Here

$$C_{\kappa_\varepsilon}(\boldsymbol{x}) = \max_{|\boldsymbol{\beta}|=2k,\ \boldsymbol{w},\boldsymbol{z} \in \Omega \cap B(\boldsymbol{x}, c_2 h_{X,\Omega})} \max \left| D_2^{\boldsymbol{\beta}} \kappa_\varepsilon(\boldsymbol{w}, \boldsymbol{z}) \right|$$

with $B(\boldsymbol{x}, c_2 h_{X,\Omega})$ denoting the ball of radius $c_2 h_{X,\Omega}$ centred at $\boldsymbol{x}$, and $h_{X,\Omega}$ being the fill distance

$$h_{X,\Omega} = \sup_{\boldsymbol{x} \in \Omega} \min_{\boldsymbol{x}_j \in X} ||\boldsymbol{x} - \boldsymbol{x}_j||_2.$$

Theorem 2.2 states that interpolation with a C^{2k} smooth kernel κ_ε has approximation order k. Thus, we deduce that: (i) for C^∞ SPD kernels, the approximation order k is arbitrarily high; (ii) for SPD kernels with limited smoothness, the approximation order is limited by the smoothness of the kernel. For more refined error estimates, we refer the reader to the monograph [29].

3 Optimizers

The problem of selecting the best shape parameter attracted the attention of the community for its strong dependence on the fit accuracy of the RBF interpolant. In this section we introduce some optimizers that can be used for the search of the shape parameter in RBF interpolation. Specifically, we consider the deterministic LOOCV-based technique, suitably combined with Lipschitz global optimizers, and the statistical BO-based method.

3.1 Leave-One-Out Cross Validation

A common method for estimating the RBF shape parameter ε in RBF interpolation is the LOOCV. It consists in splitting the dataset into training and validation sets, used for the partial fit and the computation of the error. The training set consists of all the data but one that composes the validation set. This subdivision is performed for each point in X and at the end we retrieve an error estimate vector and we use a cost function to determine the best ε parameter, see [9].

For each ε and for each $j \in \{1, \dots, N\}$, the error

$$e_j(\varepsilon) = f(\boldsymbol{x}_j) - s_{X \setminus \{x_j\}}(\boldsymbol{x}_j)$$

is computed at the validation point $\boldsymbol{x}_j$ that is not used to construct the partial RBF interpolant

$$s_{X \setminus \{x_j\}}(\boldsymbol{x}) = \sum_{k=1,\ k \neq j}^{N} c_k \kappa_\varepsilon(\boldsymbol{x}, \boldsymbol{x}_k). \tag{5}$$

The coefficients c_k in (5) are determined by interpolating only the set $X \setminus \{x_j\}$, i.e.,

$$s_{X \setminus \{x_j\}}(\boldsymbol{x}_k) = f(\boldsymbol{x}_k), \quad k = 1, \dots, j-1, j+1, \dots, N.$$

The optimal value of ε is found as

$$\varepsilon^* = \text{argmin}_\varepsilon ||\boldsymbol{e}(\varepsilon)||, \qquad \boldsymbol{e} = (e_1, \dots, e_N)^T,$$

where $|| \cdot ||$ is any norm used in the minimization problem, for instance, the ∞-norm. To avoid solving N linear system for each ε parameter, the Rippa rule [24] is used:

$$e_j(\varepsilon) = \frac{c_j}{(\mathsf{K}_\varepsilon^{-1})_{jj}},$$

where c_j is the jth coefficient of the solution vector $\boldsymbol{c} = \mathsf{K}_\varepsilon^{-1} \boldsymbol{y}$ in (3), and $(\mathsf{K}_\varepsilon^{-1})_{jj}$ is the jth diagonal element of the inverse of the *full* RBF matrix K_ε.

It follows immediately that the optimal value ε^* for the shape parameter is the one that minimizes the error function $Er(\varepsilon)$ defined as follows:

$$Er(\varepsilon) = \max_{j=1,\dots,N} \left| \frac{c_j}{(\mathsf{K}_\varepsilon^{-1})_{jj}} \right|. \tag{6}$$

3.2 Lipschitz Global Optimization

The Lipschitz global optimization in LOOCV for RBF interpolation is a specialized technique which aims at determining the optimal shape parameter leveraging between stability and accuracy of interpolation results. To enhance the reliability and efficiency of the interpolation process, it is proposed to integrate global optimization with both pessimistic and optimistic modifications. This integration, referred to as LOOCV-GOPI and LOOCV-GOOI, can lead to significant advancements in the interpolation methodology, as shown in [7].

For a given search interval $I = [0, \varepsilon_{max}]$, ε_{max} being sufficiently large, our aim is to determine the optimal shape parameter ε^* and the corresponding value

$$Er^* = Er(\varepsilon^*) = \min Er(\varepsilon), \qquad \varepsilon \in I, \tag{7}$$

where $Er(\varepsilon)$ coming from (6) is the error function, which is supposed to be characterized by its non-differentiable nature, multiple extreme values, and the complexity of its evaluation, even for an individual data point. Additionally, this function is expected to adhere to Lipschitz continuity within the interval I, i.e.,

$$|Er(\varepsilon_1) - Er(\varepsilon_2)| \le L|\varepsilon_1 - \varepsilon_2|, \qquad \varepsilon_1, \varepsilon_2 \in I, \tag{8}$$

L denoting the Lipschitz constant (bounded by $0 < L < \infty$), which assesses the function smoothness.

This kind of problems are crucial due to their widespread real-world applications in various scientific and engineering fields (see e.g. [14, 16, 20, 22, 25]). Researchers have indeed developed several methods to solve problems (7)-(8), including deterministic nature-inspired and metaheuristic algorithms. In this study, we address the optimization problems presented by equations (7)-(8) by employing the Lipschitz global optimization method developed in previous work [7]. This approach combined with effective strategies of local improvement (i.e. GOOI and GOPI) has been rigorously validated in [7], standing out for its accuracy and efficiency in determining the RBF shape parameter.

3.3 Bayesian Global Optimization

The BO, first introduced in the late ’70s by Mockus [19], is a widespread technique for optimizing hyperparameters in the machine learning landscape. It is an iterative approach that at each iteration exploits the previously obtained results. Suppose to search for a maximum of a function g on a given set, BO offers a valuable approach to conduct the search. The method involves constructing a probabilistic model of g, known as a surrogate model, and leveraging it to select sampling points in X via an acquisition function. This allows for the evaluation of the target function at these selected points. With each iteration, the surrogate model is updated based on collected data and then utilized to inform the selection of the next sampling point. Although the process involves computational overhead for selecting the next evaluation point, it proves beneficial in scenarios where evaluating g is resource-intensive. By iteratively refining the sampling strategy, BO can efficiently converge to the maximum. In this section, we provide a brief overview of BO. For a more comprehensive understanding, readers are encouraged to refer to [2].

Gaussian Processes (GP) are often favored as the surrogate model in BO due to their cost-effective evaluations and their capacity to integrate prior beliefs regarding the objective function. They are constituted by a set of random variables, where any subset of these variables follows a joint Gaussian distribution. These processes are entirely defined by a mean function $m : \mathcal{X} \to \mathbb{R}$ and a positive definite covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, where $\mathcal{X} \subseteq \mathbb{R}$ (see [23] for more detailed explainations). When we model the target function with a Gaussian process as $g(x) \sim \mathcal{GP}\big(m(x), k(x, x')\big)$, we are imposing the following conditions:

- $\mathbb{E}\big[g(x)\big] = m(x)$;
- $\mathbb{E}\big[\big(g(x) - m(x)\big)\big(g(x') - m(x')\big)\big] = k(x, x')$.

An acquisition function $a : \mathcal{X} \to \mathbb{R}$ plays a crucial role in selecting the subsequent point for evaluation by the objective function. The chosen point is the maximum of this acquisition function, and its assessment by the objective function serves to update the surrogate model. In this study, we employ the "Expected Improvement" acquisition function [15], which considers both the likelihood of improvement of the candidate point compared to the previous maximum and the extent of this improvement. In particular, we use a closed form proposed in [18] that trades off exploration and exploitation:

$$EI(x) = \begin{cases} (\mu(x) - g(\hat{x}) - \xi)\Phi(Z) + \sigma(x)\phi(Z), & \text{if } \sigma(x) > 0, \\ 0, & \text{if } \sigma(x) = 0, \end{cases} \tag{9}$$

where $Z = \frac{\mu(x) - g(\hat{x}) - \xi}{\sigma(x)}$ and ξ is the non-negative parameter that allows trading between exploration and exploitation.

4 Numerical Results

In this section we report the numerical results obtained by applying the BO, LOOCV-GOPI and LOOCV-GOOI to find the optimal shape parameter using different kernels, interpolating the Franke’s and Valley test functions using various sets of data points. The mathematical formulations of these two test functions [1, 17] are as follows:

$$
\begin{aligned}
f_1(\boldsymbol{x}) = {} & 0.75\exp\left[-\frac{(9x_1-2)^2}{4}-\frac{(9x_2-2)^2}{4}\right]+0.75\exp\left[-\frac{(9x_1-2)^2}{49}-\frac{9x_2+1}{10}\right]+\\
& +0.5\exp\left[-\frac{(9x_1-7)^2}{4}-\frac{(9x_2-3)^2}{4}\right]-0.2\exp\left[-(9x_1-4)^2-(9x_2-7)^2\right],\\
f_2(\boldsymbol{x}) = {} & \frac{1}{2}x_2\left[\cos(4x_1^2+x_2-1)\right]^4.
\end{aligned}
$$

We used some sets of Halton points, i.e. $N = 80$, 160, 320, 640, for the interpolation and a grid of $M = 40 \times 40$ points in $\Omega = [0,1]^2$ for the evaluation. For all methods the search of ε has been carried out in the interval $I = [0, \varepsilon_{max}]$, with $\varepsilon_{max} = 20$. The ξ parameter for the BO was set to 0.01 because the analysis reported in [5] shows that for this kind of problem there are no significant improvements in tweaking it. The metric used for the evaluation of the models is the Root Mean Squared Error (RMSE) defined as follows:

$$
\text{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(s_X(\boldsymbol{x}_i)-y_i\right)^2}.
$$

All the experiments were performed using MATLAB online with processor Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz.

Tables 1-10 show that the obtained precision specified by RMSE is comparable in almost every case, while the CPU Time computed in seconds (s) is the discriminator for the methods. Notably, for smaller number of interpolation points the LOOCV-GOOI and LOOCV-GOPI methods are preferable whereas, if we increase the size of interpolation dataset, BO seems to be a more suitable choice.

	RMSE			CPU Time (s)		
N	BO	LOOCV-GOPI	LOOCV-GOOI	BO	LOOCV-GOPI	LOOCV-GOOI
80	1.1289e-02	1.2296e-02	1.2296e-02	1.4421e+00	2.3807e-01	1.5142e-01
160	3.0200e-03	2.8216e-03	2.8216e-03	1.3317e+00	2.7791e-01	1.3040e-01
320	2.3085e-03	5.4031e-04	5.4031e-04	1.4199e+00	1.3305e+00	4.9235e-01
640	1.1535e-05	4.6602e-05	4.6602e-05	1.9452e+00	2.0486e+00	1.8549e+00

Table 1: Comparative analysis of RMSE and CPU Time using GA kernel on f_1.

	RMSE			CPU Time (s)		
N	BO	LOOCV-GOPI	LOOCV-GOOI	BO	LOOCV-GOPI	LOOCV-GOOI
80	5.4518e-03	5.3374e-03	5.3374e-03	1.2424e+00	3.9665e-02	3.5078e-02
160	8.0557e-04	7.3814e-04	7.3814e-04	1.3082e+00	1.1375e-01	1.0692e-01
320	7.0361e-04	8.6893e-05	8.6893e-05	1.0940e+00	2.4894e+00	1.5453e+00
640	6.9506e-05	3.0039e-06	4.7761e-06	1.6180e+00	1.6074e+01	1.1391e+01

Table 2: Comparative analysis of RMSE and CPU Time using IMQ kernel on f_1.

	RMSE			CPU Time (s)		
N	BO	LOOCV-GOPI	LOOCV-GOOI	BO	LOOCV-GOPI	LOOCV-GOOI
80	7.2507e-03	6.9058e-03	6.9058e-03	1.1259e+00	4.1105e-02	5.1959e-02
160	6.6278e-04	8.6947e-04	8.6947e-04	1.0314e+00	1.2380e+00	1.6924e-01
320	8.8553e-05	3.3847e-04	3.3847e-04	1.2129e+00	6.9026e+00	5.7679e-01
640	7.8870e-06	1.4182e-05	1.4182e-05	2.1707e+00	1.9895e+00	1.7967e+00

Table 3: Comparative analysis of RMSE and CPU Time using M6 kernel on f_1.

Figure 1 summarizes the results of the previous tables, demonstrating that LOOCV-based methods combined with Lipschitz global optimization tools are better for interpolation problems of small size, because computational cost strongly correlates with the cardinality of the set on which optimization is performed. On the other hand, since the computational expense of BO is mainly due to surrogate model fitting, it maintains a relatively constant level of computational cost thus turning out less influenced

	RMSE			CPU Time (s)		
N	BO	LOOCV-GOPI	LOOCV-GOOI	BO	LOOCV-GOPI	LOOCV-GOOI
80	7.4787e-03	7.0838e-03	7.0838e-03	1.1087e+00	1.7831e-01	5.4054e-02
160	1.5801e-02	2.3901e-03	2.3901e-03	1.2040e+00	8.0776e-01	1.4245e-01
320	1.6295e-03	1.7567e-03	1.7567e-03	1.3793e+00	3.0430e+00	5.2081e-01
640	4.5363e-04	4.3827e-04	4.3827e-04	1.6942e+00	1.2186e+01	2.1261e+00

Table 4: Comparative analysis of RMSE and CPU Time using W2 kernel on f_1.

	RMSE			CPU Time (s)		
N	BO	LOOCV-GOPI	LOOCV-GOOI	BO	LOOCV-GOPI	LOOCV-GOOI
80	8.4409e-03	8.6778e-03	8.6778e-03	8.9213e-01	1.7128e-01	4.5174e-02
160	3.7877e-03	1.1496e-03	1.1496e-03	1.1134e+00	7.7839e-01	1.3751e-01
320	1.0393e-02	3.9007e-04	3.9007e-04	1.2974e+00	2.9886e+00	7.1863e-01
640	1.6257e-04	3.7756e-05	3.7756e-05	1.7459e+00	1.1315e+01	1.8754e+00

Table 5: Comparative analysis of RMSE and CPU Time using W4 kernel on f_1.

	RMSE			CPU Time (s)		
N	BO	LOOCV-GOPI	LOOCV-GOOI	BO	LOOCV-GOPI	LOOCV-GOOI
80	1.8894e-02	3.2335e-02	3.2335e-02	9.3042e-01	1.2881e-01	1.3507e-01
160	4.4386e-03	6.8314e-03	6.8314e-03	9.3453e-01	1.2718e-01	1.5204e-01
320	4.5508e-04	4.8270e-04	4.8270e-04	8.9626e-01	5.4048e-01	4.9732e-01
640	4.1555e-05	5.4715e-05	5.4715e-05	1.3416e+00	1.7654e+00	1.5634e+00

Table 6: Comparative analysis of RMSE and CPU Time using GA kernel on f_2

	RMSE			CPU Time (s)		
N	BO	LOOCV-GOPI	LOOCV-GOOI	BO	LOOCV-GOPI	LOOCV-GOOI
80	2.4888e-02	4.5869e-02	4.5869e-02	8.2484e-01	4.1061e-02	3.6245e-02
160	4.0723e-03	5.2407e-03	8.5460e-03	9.1687e-01	7.0580e-01	1.1351e-01
320	7.1197e-04	1.2350e-02	1.2350e-02	9.5668e-01	2.6689e+00	4.1033e-01
640	1.0726e-04	1.7244e-04	1.8652e-04	1.3264e+00	2.4916e+00	1.9077e+00

Table 7: Comparative analysis of RMSE and CPU Time using IMQ kernel on f_2.

	RMSE			CPU Time (s)		
N	BO	LOOCV-GOPI	LOOCV-GOOI	BO	LOOCV-GOPI	LOOCV-GOOI
80	2.6825e-02	3.8603e-02	4.3552e-02	1.0154e+00	3.9888e-01	9.0818e-02
160	4.4334e-03	5.3896e-03	5.3896e-03	9.6398e-01	1.8130e-01	2.9467e-01
320	2.2620e-03	2.2586e-03	2.2586e-03	1.1008e+00	4.3941e-01	5.8553e-01
640	7.2221e-04	1.4191e-03	1.4206e-03	1.8800e+00	4.5927e+00	2.8442e+00

Table 8: Comparative analysis of RMSE and CPU Time using M6 kernel on f_2.

by the growth of N. In light of these findings, we can conclude that the compared methods are not competing, but they are rather complementary. Indeed, the LOOCV-based approach is better when one interpolates a "small" number of points, while the BO is more efficient for "large" datasets. The choice between them should be guided by the complexity of the problem and its dimensionality.

5 Conclusions

The paper aimed to bring light on and compare two recently used methods for the search of the shape parameter in RBF interpolation. As the numerical results showed in Section 4 there exists an inverse trend in computational cost between the

	RMSE			CPU Time (s)		
N	BO	LOOCV-GOPI	LOOCV-GOOI	BO	LOOCV-GOPI	LOOCV-GOOI
80	2.0007e-02	5.3561e-02	5.3561e-02	8.7842e-01	2.3875e-01	4.7427e-02
160	1.2276e-02	1.2358e-02	1.2358e-02	9.0036e-01	1.6587e+00	1.5310e-01
320	3.2437e-03	3.2455e-03	3.2517e-03	1.0256e+00	2.9376e+00	8.3872e-01
640	1.2760e-03	1.3674e-03	1.3674e-03	1.6090e+00	1.9748e+01	2.2449e+00

Table 9: Comparative analysis of RMSE and CPU Time using W2 kernel on f_2.

	RMSE			CPU Time (s)		
N	BO	LOOCV-GOPI	LOOCV-GOOI	BO	LOOCV-GOPI	LOOCV-GOOI
80	3.4742e-02	4.1797e-02	4.1797e-02	9.5569e-01	2.7706e-01	6.5227e-02
160	4.5122e-03	1.9048e-02	4.6621e-03	1.0826e+00	1.2377e+00	1.5665e-01
320	4.8468e-03	2.1142e-03	2.1142e-03	1.3506e+00	5.1854e+00	5.9573e-01
640	8.5735e-04	1.0629e-03	1.0629e-03	2.0742e+00	2.6742e+01	2.1460e+00

Table 10: Comparative analysis of RMSE and CPU Time using W4 kernel on f_2.

deterministic (LOOCV-GOPI and LOOCV-GOOI) and the statistical (BO) optimization methods; in other words, here it is meant that, even though both methods increase their cost when the number of points grows, the BO increases slightly with respect to the LOOCV-based technique. Notably, from Figure 1 it can be seen that the GOOI method is less expensive than GOPI for each set of data points, while optimizing on sets over roughly $N = 400$ points (leading to a linear system of dimension N) makes the BO preferable as optimization method. In conclusion, from this analysis one can deduce that both methods are valid, and in particular the choice of which one to use should be based on the cardinality of the set on which the RBF interpolant is to be constructed.

Acknowledgments

The authors sincerely thank editor and reviewer for their constructive and valuable comments that enabled to improve the paper. This work has been supported by the INdAM Research group GNCS. The work of R.C. and A.D. has been supported by the Spoke 1 “Future HPC & BigData” of the Italian Research Center on High-Performance Computing, Big Data and Quantum Computing (ICSC) funded by MUR Missione 4 Componente 2 Investimento 1.4: Potenziamento strutture di ricerca e creazione di “campioni nazionali di R&S (M4C2-19)” – Next Generation EU (NGEU). Moreover, the work has been supported by the Fondazione CRT, project 2022 “Modelli matematici e algoritmi predittivi di intelligenza artificiale per la mobilità sostenibile”. This research has been accomplished within the RITA “Research ITalian network on Approximation”, the UMI Group TAA “Approximation Theory and Applications”, and the SIMAI Activity Group ANA&A “Numerical and Analytical Approximation of Data and Functions with Applications”.

References

[1] M. Bozzini, L. Lenarduzzi, M. Rossini, Polyharmonic splines: An approximation method for noisy scattered data of extra-large size, Appl. Math. Comput. 216, 317–331 (2010)

[2] E. Brochu, V.M. Cora, N. De Freitas, A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, arXiv:1012.2599 (2010)

[3] M.D. Buhmann, Radial Basis Functions: Theory and Implementation, Cambridge Monogr. Appl. Comput. Math., vol. 12, Cambridge Univ. Press, Cambridge (2003)

[4] R. Cavoretto, A. De Rossi, A. Haider, Y.D. Sergeyev, Adaptive hyperparameter selection in kernel-based partition of unity methods by global optimization techniques, 2024; https://doi.org/10.21203/rs.3.rs-4293620/v1

[5] R. Cavoretto, A. De Rossi, S. Lancellotti, Bayesian approach for radial kernel parameter tuning, J. Comput. Appl. Math. 441, 115716 (2024)

[6] R. Cavoretto, A. De Rossi, S. Lancellotti, F. Romaniello, Parameter tuning in the radial kernel-based partition of unity method by Bayesian optimization, J. Comput. Appl. Math. 451, 116108 (2024)

[7] R. Cavoretto, A. De Rossi, M.S. Mukhametzhanov, Y.D. Sergeyev, On the search of the shape parameter in radial basis functions using univariate global optimization methods, J. Global Optim. 79, 305–327 (2021)

[8] R. Cavoretto, A. De Rossi, A. Sommariva, M. Vianello, RBFCUB: A numerical package for near-optimal meshless cubature on general polygons, Appl. Math. Lett. 125, 107704 (2022)

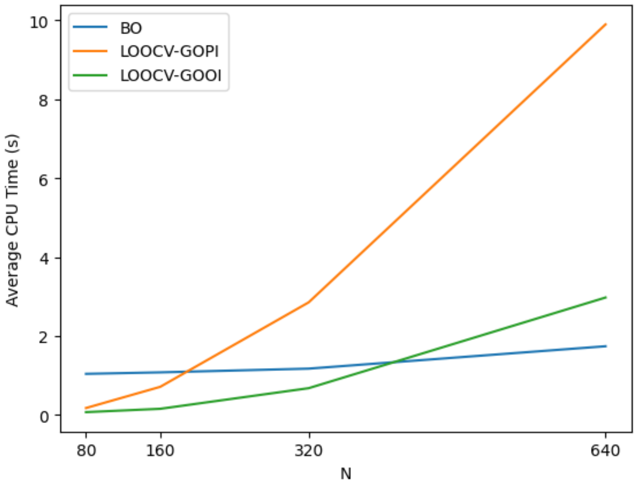

Figure 1: Average CPU Time computed on different kernels versus N.

[9] G.E. Fasshauer, Meshfree Approximation Methods with MATLAB, Interdisciplinary Mathematical Sciences, vol. 6, World Scientific Publishing Co., Singapore (2007)

[10] G.E. Fasshauer, M.J. McCourt, Kernel-based Approximation Methods using MATLAB, Interdisciplinary Mathematical Sciences, Vol. 19, World Scientific Publishing Co., Singapore (2015)

[11] T.A. Foley, Near optimal parameter selection for multiquadric interpolation, J. Appl. Sci. Comput. 1, 54–69 (1994)

[12] A. Golbabai, E. Mohebianfar, H. Rabiei, On the new variable shape parameter strategies for radial basis functions, Comput. Appl. Math. 34, 691–704 (2015)

[13] K. Hamacher, On stochastic global optimization of one-dimensional functions, Physica A: Statistical Mechanics and its Applications 354, 547–557 (2005)

[14] D.E. Johnson, Introduction to Filter Theory, Prentice Hall Inc., New Jersey (1976)

[15] D.R. Jones, M. Schonlau, W.J. Welch, Efficient Global Optimization of Expensive Black-Box Functions, J. Global Optim. 13, 455–492 (1998)

[16] H.Y.F. Lam, Analog and Digital Filters-Design and Realization, Prentice Hall Inc., New Jersey (1979)

[17] D. Lazzaro, L.B. Montefusco, Radial basis functions for the multivariate interpolation of large scattered data sets, J. Comput. Appl. Math. 140, 521–536 (2002)

[18] D. Lizotte, Practical Bayesian Optimization, PhD thesis, University of Alberta, Edmonton, Alberta, Canada (2008)

[19] J. Mockus, V. Tiesis, A. Zilinskas, The application of Bayesian methods for seeking the extremum, Towards Global Optimization 2 (1978) 117–129.

[20] A. Molinaro, Y.D. Sergeyev, Finding the minimal root of an equation with the multiextremal and nondifferentiable left-hand part, Numer. Alg. 28, 255–272 (2001)

[21] A. Noorizadegan, C.-S. Chen, R. Cavoretto, A. De Rossi, Efficient truncated randomized SVD for mesh-free kernel methods, Comput. Math. Appl. 164, 12–20 (2024)

[22] J.D. Pinter, Global Optimization in Action – Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications, Kluwer Academic Publishers, Dordrecht (1996)

[23] C.E. Rasmussen, C. Williams, Gaussian Processes for Machine Learning, MIT Press (2006)

[24] S. Rippa, An algorithm for selecting a good value for the parameter c in radial basis function interpolation, Adv. Comput. Math. 11, 193–210 (1999)

[25] Y.D. Sergeyev, D.E. Kvasov, F.M.H. Khalaf, A one-dimensional local tuning algorithm for solving GO problems with partially defined constraints, Optim. Lett. 1(1), 85–99 (1995)

[26] Y.D. Sergeyev, M.S. Mukhametzhanov, D.E. Kvasov, D. Lera, Derivative-free local tuning and local improvement techniques embedded in the univariate global optimization, J. Optim. Theory Appl. 171, 186–208 (2016)

[27] J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian optimization of machine learning algorithms, Advances in Neural Information Processing Systems 25, 2960–2968 (2012)

[28] R. Schaback, H. Wendlandm Kernel techniques: from machine learning to meshless methods, Acta Numer. 15, 543–639 (2006)

[29] H. Wendland, Scattered Data Approximation, Cambridge Monogr. Appl. Comput. Math., vol. 17, Cambridge Univ. Press, Cambridge (2005)