The extended Rippa's algorithm in RBF interpolation

Francesco Marchetti
francesco.marchetti@math.unipd.it
Dipartimento di Matematica "Tullio Levi-Civita"
2GGALN, Napoli, 15/02/2022

Outline

1. Introduction: shape parameter's tuning in RBF interpolation
2. The Extended Rippa's Algorithm (ERA)
3. Numerical experiments
4. Related works and future directions

Kernel-based interpolation

Let $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, $\kappa_\varepsilon : \Omega \times \Omega \longrightarrow \mathbb{R}$ be a strictly positive definite radial kernel, $\kappa_\varepsilon(\boldsymbol{x}, \boldsymbol{y}) = \Phi_\varepsilon(\|\boldsymbol{x} - \boldsymbol{y}\|)$, possibly depending on a shape parameter $\varepsilon > 0$, and let $\mathcal{X} = \{\boldsymbol{x}_i,\ i = 1, \ldots, n\} \subset \Omega$ be a subset of distinct nodes, $n \in \mathbb{N}$. Suppose that we wish to reconstruct a function $f : \Omega \longrightarrow \mathbb{R}$ by knowing its samples at $\mathcal{X}$, i.e. $f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)$. We take a function

$$S_{f,\mathcal{X}}(\mathbf{x}) = \sum_{i=1}^{n} c_i \kappa_\varepsilon(\boldsymbol{x}, \boldsymbol{x}_i), \quad \boldsymbol{x} \in \Omega,$$

where $\boldsymbol{c} = (c_1, \ldots, c_n)^\intercal \in \mathbb{R}^n$ is determined by imposing interpolation conditions $S_{f,\mathcal{X}}(\boldsymbol{x}_i) = f(\boldsymbol{x}_i)$, $i = 1, \ldots, n$, i.e. $K_\varepsilon \boldsymbol{c} = \boldsymbol{f}$, with $(K_\varepsilon)_{ij} = \kappa_\varepsilon(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{f}$ the vector of function evaluations. We may denote $\kappa_\varepsilon = \kappa$ and $K_\varepsilon = K$.

CV methods

In order to tune the shape parameter, a Cross Validation (CV) scheme can be used.

1. The dataset is divided into $k \in \mathbb{N}$ (possibly equal-sized) disjoint subsets, $k \leq n$
2. Then, k different models are built upon $k-1$ *training* folds and evaluated on the respective remaining *validation* fold.
3. A validation error e_i is assigned to each node. Then, we take the vector $\boldsymbol{e} = (e_1, \dots, e_n) \in \mathbb{R}^n$ and we compute $\|\boldsymbol{e}\|$ as the *global* CV error.

By setting $p \approx n/k$, this procedure is an approximation of *Leave-p-Out* CV (LpOCV).

Rippa's scheme

The computational cost related to a k-fold CV scheme is $\mathcal{O}(k(n-p)^3) \approx \mathcal{O}(n^3 k)$. In particular, it is $\mathcal{O}(n^4)$ for LOOCV, where $k = n$.

In [Rippa (1999)], a fast LOOCV algorithm has been proposed: the vector of CV errors $\boldsymbol{e}$ can be computed in $\mathcal{O}(n^3)$ by the formula

$$\boldsymbol{e} = \boldsymbol{c}./\mathrm{diag}(K^{-1})$$

where ./ is the pointwise division between vectors.

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

The ERA

Let $\boldsymbol{p} = (p_1, \dots, p_v)^\intercal$, $p_i \in \{1, \dots, n\}$, $v \in \mathbb{N}$, $v < n$, and let $\mathcal{V} = \{\boldsymbol{x}_{p_1}, \dots, \boldsymbol{x}_{p_v}\}$ be the validation set at a certain iteration of the CV scheme. In [M. (2021)], Rippa's scheme has been extended for any $k < n$.

Theorem [Theorem 1, M. (2021)] The vector of v validation errors $\boldsymbol{e_p}$ related to $\boldsymbol{x}_{p_1}, \dots, \boldsymbol{x}_{p_v}$ is the unique solution of the linear system $K^{-1}_{\boldsymbol{p},\boldsymbol{p}} \boldsymbol{z} = \boldsymbol{c_p}, \quad \boldsymbol{z} \in \mathbb{R}^v$, where $K^{-1}_{\boldsymbol{p},\boldsymbol{p}} = (K^{-1}_{i,j})_{i,j \in \boldsymbol{p}}$ and $\boldsymbol{c_p} = (c_i)_{i \in \boldsymbol{p}}$.

With this Extended Rippa's Algorithm (ERA), the computational complexity of k-fold CV is $\mathcal{O}(n^3) + \mathcal{O}\left(\frac{n^3}{k^2}\right)$. Thus, if $k \approx n$ (equivalently $p \ll n$), ERA is to be preferred over the classical approach, which is $\approx \mathcal{O}(n^3 k)$.

The extension of Rippa's algorithm beyond LOOCV,
F. Marchetti - Appl. Math. Lett., 120 (2021), 107262

Proof

Remark Let $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^n$. If $\boldsymbol{\alpha}_{\boldsymbol{p}} = (\alpha_{p_1}, \ldots, \alpha_{p_v})^\intercal \equiv \mathbf{0}$, then $A\boldsymbol{\alpha} = \boldsymbol{\beta} \implies A^{\boldsymbol{p},\boldsymbol{p}}\boldsymbol{\alpha}^{\boldsymbol{p}} = \boldsymbol{\beta}^{\boldsymbol{p}}$, where $\boldsymbol{\alpha}^{\boldsymbol{p}} = (\alpha_i)_{i \notin \boldsymbol{p}}$, $\boldsymbol{\beta}^{\boldsymbol{p}} = (\beta_i)_{i \notin \boldsymbol{p}}$.

Let $\boldsymbol{b} = (b_1, \ldots, b_d)^\intercal \in \mathbb{R}^d$ be such that $\boldsymbol{b}_{\boldsymbol{p}} \equiv \mathbf{0}$ and $K\boldsymbol{b} = \boldsymbol{f} - \sum_{i=1}^{v} \gamma_i I_{:,p_i}$, where $I_{:,\ell}$ denotes the ℓ-th column of the $n \times n$ identity matrix I and $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_v)^\intercal \in \mathbb{R}^v$. Then, by virtue of Remark, $\boldsymbol{c}^{(\boldsymbol{p})} = \boldsymbol{b}^{\boldsymbol{p}}$, which implies for any $p_j \in \boldsymbol{p}$, $j = 1, \ldots, v$,

$$S^{(\boldsymbol{p})}(\boldsymbol{x}_{p_j}) = \sum_{i \notin \boldsymbol{p}} c_i^{(\boldsymbol{p})} \Phi_\varepsilon(\|\boldsymbol{x}_{p_j} - \boldsymbol{x}_i\|) = \sum_{i=1}^{n} b_i \Phi_\varepsilon(\|\boldsymbol{x}_{p_j} - \boldsymbol{x}_i\|) = (K\boldsymbol{b})_{p_j} = f_{p_j} - \gamma_j.$$

Hence, $\gamma_j = f_{p_j} - S^{(\boldsymbol{p})}(\boldsymbol{x}_{p_j}) = e_{p_j}$ and $\boldsymbol{\gamma} = \boldsymbol{e}_{\boldsymbol{p}}$. Finally, we have $K^{-1}_{\boldsymbol{p},\boldsymbol{p}}\boldsymbol{\gamma} = \boldsymbol{c}_{\boldsymbol{p}}$.

Indeed, $\boldsymbol{b} = K^{-1}\left(\boldsymbol{f} - \sum_{i=1}^{v} \gamma_i I_{:,p_i}\right) = \boldsymbol{c} - \sum_{i=1}^{v} \gamma_i K^{-1}_{:,p_i}$, and therefore

$\mathbf{0} \equiv \boldsymbol{b}_{\boldsymbol{p}} = \boldsymbol{c}_{\boldsymbol{p}} - K^{-1}_{\boldsymbol{p},\boldsymbol{p}}\boldsymbol{\gamma}$.

Tests setting

1. Interpolation set $\mathcal{E}_n$, 30×30 equispaced grid in Ω, $n = 900$;
   Wendland C^2 $\Phi_{W,\varepsilon}(r) = (1 - \varepsilon r)_+^4 (4\varepsilon r + 1)$;
   test function $f_1(\boldsymbol{x}) = \frac{\sin(x_1)\cos(x_2)}{(x_1^2+1)(x_2^2+1)}$, $\boldsymbol{x} = (x_1, x_2)$.

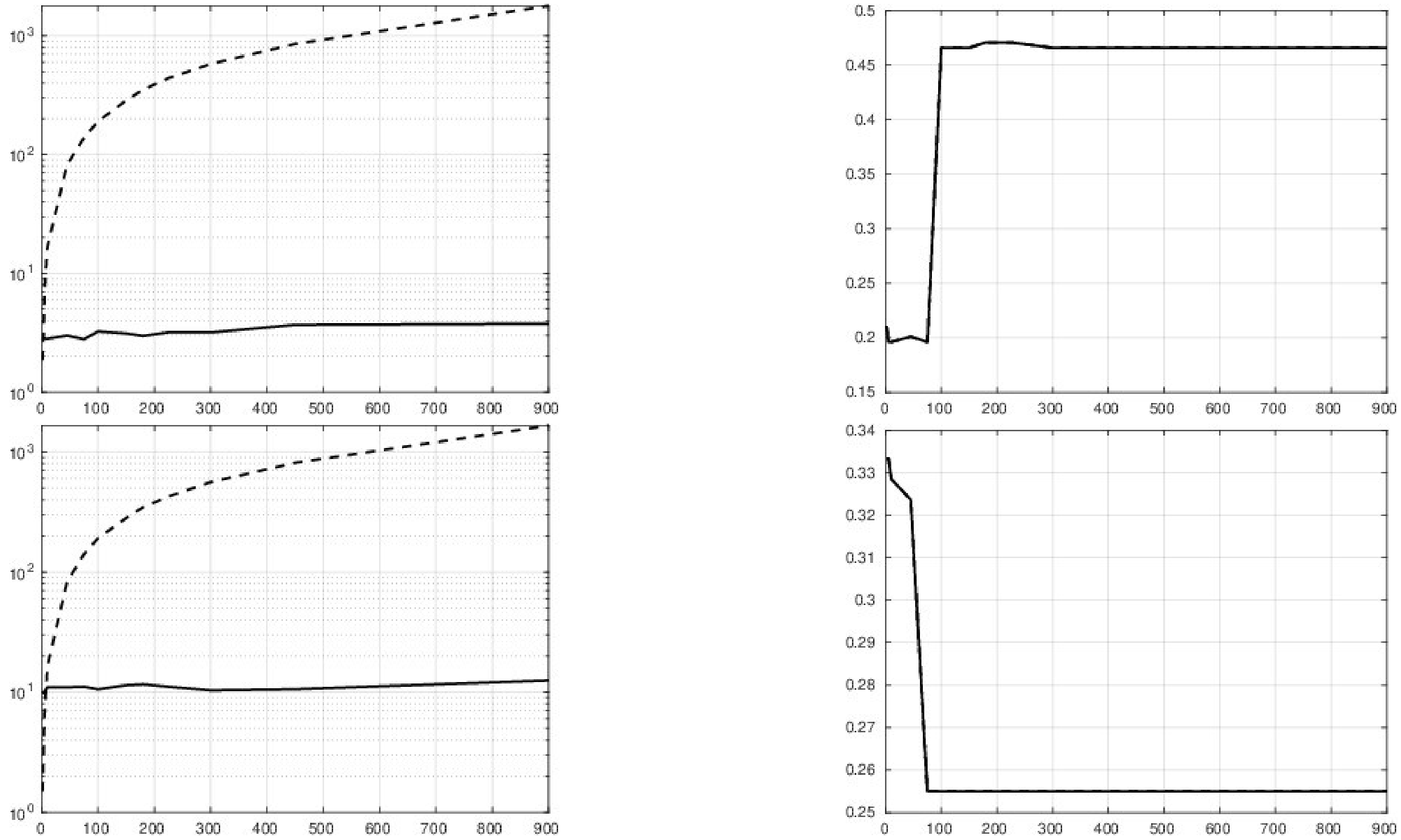
2. Interpolation set $\mathcal{H}_n$, quasi-random *Halton points*, $n = 900$;
   Inverse multiquadric $\Phi_{I,\varepsilon}(r) = \frac{1}{\sqrt{1+(\varepsilon r)^2}}$;
   test function $f_2(\boldsymbol{x}) = e^{-(x_1+1)^2} + x_2^2$, $\boldsymbol{x} = (x_1, x_2)$.

Letting $\Omega = [-1, 1]^2$, we take $\boldsymbol{k} = (2, 5, 10, 45, 75, 100, 150, 180, 225, 300, 450, 900)$ as folds vector and $\boldsymbol{\varepsilon} = (a + (b - a)i/\nu)_{i=0,\dots,\nu}$ with $a = 1/100$, $b = 1/2$, $\nu = 100$ as shape parameters vector. We consider k-fold CV with $k \in \boldsymbol{k}$ in order to choose the optimal $\varepsilon^\star \in \boldsymbol{\varepsilon}$. More precisely, the optimal value is selected by evaluating the CV error in infinity norm.

Results

Test 1

Test 2



Left: varying $k \in \boldsymbol{k}$, the time (in seconds) employed by the standard implementation (dashed line) and by the proposed extended Rippa's scheme (solid line) in performing $\nu + 1$ repetitions of k-fold CV to select $\varepsilon^\star \in \boldsymbol{\varepsilon}$. Right: the value $\varepsilon^\star$ chosen by both the methods, which coincides.

Concluding remarks

We studied a stochastic low-rank approximation of the kernel matrix inverse for further speeding up the calculations (joint work with L. Ling)

A stochastic extended Rippa's algorithm for LpOCV,
L. Ling, F. Marchetti - Appl. Math. Lett. 129 (2022), 107955

We presented a knot-removal scheme which is built upon the ERA (joint work with E. Perracchione, within RITA)

Efficient Reduced Basis Algorithm (ERBA) for kernel-based approximation,
F. Marchetti, E. Perracchione - submitted (2021)

Could the ERA be adapted/employed in different contexts?

Thanks for the attention!