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 $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, $\kappa_\varepsilon : \Omega \times \Omega \rightarrow \mathbb{R}$ be a strictly positive definite radial kernel, $\kappa_\varepsilon(x, y) = \Phi_\varepsilon(\|x - y\|)$, possibly depending on a shape parameter $\varepsilon > 0$, and let $\mathcal{X} = \{x_i, i = 1, \ldots, n\} \subset \Omega$ be a subset of distinct nodes, $n \in \mathbb{N}$. Suppose that we wish to reconstruct a function $f : \Omega \rightarrow \mathbb{R}$ by knowing its samples at $\mathcal{X}$, i.e. $f(x_1), \ldots, f(x_n)$. We take a function

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

where $c = (c_1, \ldots, c_n)^{\top} \in \mathbb{R}^n$ is determined by imposing interpolation conditions $S_{f,\mathcal{X}}(x_i) = f(x_i), i = 1, \ldots, n$, i.e. $K_\varepsilon c = f$, with $(K_\varepsilon)_{ij} = \kappa_\varepsilon(x_i, x_j)$ and $f$ 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 $e = (e_1, \ldots, e_n) \in \mathbb{R}^n$ and we compute $\|e\|$ as the global CV error.

By setting $p \approx n/k$, this procedure is an approximation of Leave-$p$-Out CV (L$p$OCV).
Rippa’s scheme

The computational cost related to a $k$-fold CV scheme is $\mathcal{O}(k(n-p)^3) \approx \mathcal{O}(n^3k)$. 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 $\mathbf{e}$ can be computed in $\mathcal{O}(n^3)$ by the formula

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

where ./ is the pointwise division between vectors.

The ERA

Let $\mathbf{p} = (p_1, \ldots, p_v)\trans$, $p_i \in \{1, \ldots, n\}$, $v \in \mathbb{N}$, $v < n$, and let $\mathcal{V} = \{\mathbf{x}_{p_1}, \ldots, \mathbf{x}_{p_v}\}$ 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 $\mathbf{e}_\mathbf{p}$ related to $\mathbf{x}_{p_1}, \ldots, \mathbf{x}_{p_v}$ is the unique solution of the linear system

$$K_{\mathbf{p},\mathbf{p}}^{-1}\mathbf{z} = \mathbf{c}_\mathbf{p}, \quad \mathbf{z} \in \mathbb{R}^v,$$

where $K_{\mathbf{p},\mathbf{p}}^{-1} = (K_{i,j}^{-1})_{i,j \in \mathbf{p}}$ and $\mathbf{c}_\mathbf{p} = (c_i)_{i \in \mathbf{p}}$.

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 $\approx \mathcal{O}(n^3k)$.

*The extension of Rippa’s algorithm beyond LOOCV,*
Remark Let \( \alpha, \beta \in \mathbb{R}^n \). If \( \alpha_p = (\alpha_{p_1}, \ldots, \alpha_{p_v})^\top \equiv 0 \), then \( A\alpha = \beta \implies A^{p_p} \alpha^p = \beta^p \), where \( \alpha^p = (\alpha_i)_{i \notin p} \), \( \beta^p = (\beta_i)_{i \notin p} \).

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

\[
S^{(p)}(x_{p_j}) = \sum_{i \notin p} c_i^{(p)} \Phi_\varepsilon(\|x_{p_j} - x_i\|) = \sum_{i=1}^{n} b_i \Phi_\varepsilon(\|x_{p_j} - x_i\|) = (K \, b)_{p_j} = f_{p_j} - \gamma_j.
\]

Hence, \( \gamma_j = f_{p_j} - S^{(p)}(x_{p_j}) = e_{p_j} \) and \( \gamma = e_p \). Finally, we have \( K_{p,p}^{-1} \gamma = c_p \).

Indeed, \( b = K^{-1}\left( f - \sum_{i=1}^{v} \gamma_i I_{:i:p_i} \right) = c - \sum_{i=1}^{v} \gamma_i K_{:i:p_i}^{-1} \), and therefore \( 0 \equiv b_p = c_p - K_{p,p}^{-1} \gamma \).
Tests setting

1. Interpolation set $\mathcal{E}_n$, $30 \times 30$ equispaced grid in $\Omega$, $n = 900$;

   \[
   \text{Wendland } C^2 \Phi_{W, \varepsilon}(r) = (1 - \varepsilon r)^4_+(4 \varepsilon r + 1);
   \]

   test function $f_1(\bm{x}) = \frac{\sin(x_1) \cos(x_2)}{(x_1^2 + 1)(x_2^2 + 1)}$, $\bm{x} = (x_1, x_2)$.

2. Interpolation set $\mathcal{H}_n$, quasi-random Halton points, $n = 900$;

   \[
   \text{Inverse multiquadric } \Phi_{I, \varepsilon}(r) = \frac{1}{\sqrt{1 + (\varepsilon r)^2}};
   \]

   test function $f_2(\bm{x}) = e^{-(x_1+1)^2 + x_2^2}$, $\bm{x} = (x_1, x_2)$.

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

Test 1

Test 2

Left: varying $k \in k$, 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 $\varepsilon^* \in \varepsilon$. Right: the value $\varepsilon^*$ 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,*

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!