

An alternative procedure for selecting a good value for the parameter c in RBF-interpolation

Michael Scheuerer

Received: 16 March 2009 / Accepted: 28 January 2010 /

Published online: 4 June 2010

© The Author(s) 2010. This article is published with open access at Springerlink.com

Abstract The impact of the scaling parameter c on the accuracy of interpolation schemes using radial basis functions (RBFs) has been pointed out by several authors. Rippa (Adv Comput Math 11:193–210, 1999) proposes an algorithm based on the idea of cross validation for selecting a good such parameter value. In this paper we present an alternative procedure, that can be interpreted as a refinement of Rippa’s algorithm for a cost function based on the euclidean norm. We point out how this method is related to the procedure of maximum likelihood estimation, which is used for identifying covariance parameters of stochastic processes in spatial statistics. Using the same test functions as Rippa we show that our algorithm compares favorably with cross validation in many cases and discuss its limitations. Finally we present some computational aspects of our algorithm.

Keywords Radial basis functions · Shape parameter · Cross validation · Maximum likelihood

Mathematics Subject Classifications (2010) 65D05 · 65D15

Communicated by Joe Ward.

The author has been supported by the Graduiertenkolleg 1023 “Identification in Mathematical Models: Synergy of Stochastic and Numerical Methods“, Universität Göttingen, in form of a DFG scholarship.

M. Scheuerer (✉)

Institut für Mathematische Stochastik, Georg-August-Universität Göttingen,
Goldschmidtstr. 7, 37077 Göttingen, Germany
e-mail: scheuer@math.uni-goettingen.de

1 Introduction

Suppose that $f : \Omega \rightarrow \mathbb{R}$ with $\Omega \subset \mathbb{R}^d$ is a d -variate, real valued function that we want to reconstruct based on its values $\mathbf{f} = (f_1, \dots, f_N)'$ at locations $X = \{x_1, \dots, x_N\}$.

In Radial Basis Function (RBF) interpolation the interpolant $s_{f,X}$ of f is of the form

$$s_{f,X}(x) = \sum_{i=1}^N \alpha_i \phi(\|x - x_i\|) + \sum_{k=1}^Q \beta_k p_k(x), \quad x \in \Omega \tag{1}$$

where $\phi(\|\cdot\|) : \Omega \rightarrow \mathbb{R}$ is a conditionally positive definite function on Ω with respect to a finite-dimensional subspace \mathcal{P} of $C(\Omega)$ with basis p_1, \dots, p_Q . Typically \mathcal{P} is the space of all polynomials of a certain degree but in principle other choices are also possible. In the special case where ϕ is positive definite ($\mathcal{P} = \{0\}$) the second term in (1) is missing.

The coefficients $\alpha = (\alpha_1, \dots, \alpha_N)'$ and $\beta = (\beta_1, \dots, \beta_Q)'$ are determined by the interpolation conditions

$$s_{f,X}(x_i) = f_i, \quad i = 1, \dots, N \tag{2}$$

and, for conditionally positive definite RBFs, by the additional requirement

$$\sum_{i=1}^N \alpha_i p_j(x_i) = 0 \quad j = 1, \dots, Q \tag{3}$$

If X is \mathcal{P} -unisolvent, then the linear system defined by (2) and (3) is solvable. The solution is unique, if \mathcal{P} is the space of polynomials of a certain degree [19, Sec. 8.5].

Each pair (ϕ, \mathcal{P}) can be associated with some native Hilbert space $(\mathcal{N}_{\phi(\Omega)}, |\cdot|_{\mathcal{N}_{\phi(\Omega)}})$ [19, Ch. 10], and it is usually assumed that $f \in \mathcal{N}_{\phi(\Omega)}$. The RBF interpolant $s_{f,X}$ has the special property, that it has minimal (semi-)norm $|\cdot|_{\mathcal{N}_{\phi(\Omega)}}$ under all functions $s \in \mathcal{N}_{\phi(\Omega)}$ that interpolate the data \mathbf{f} at the data points X [19, Sec. 13.1].

Table 1 shows some examples of commonly used RBFs. Usually, a scaled version $\phi_c(\|\cdot\|) = \phi(\frac{1}{c} \|\cdot\|)$ of these prototype RBFs is used in (1) and

Table 1 Some commonly used RBFs

Gaussians	$\phi(r) = e^{-r^2}$	
Matérn functions	$\phi(r) = \frac{K_{\gamma-d/2}(r) r^{\gamma-d/2}}{2^{\gamma-1} \Gamma(\gamma)}$	$\gamma > \frac{d}{2}$
Inverse multiquadrics	$\phi(r) = (1+r^2)^{\gamma}$	$\gamma < 0$
Multiquadrics	$\phi(r) = (-1)^{\lceil \gamma \rceil} (1+r^2)^{\gamma}$	$\gamma > 0, \gamma \notin \mathbb{N}$
Powers	$\phi(r) = (-1)^{\lceil \gamma/2 \rceil} r^{\gamma}$	$\gamma > 0, \gamma \notin 2\mathbb{N}$
Thin-plate splines	$\phi(r) = (-1)^{\lceil \gamma/2+1 \rceil} r^{\gamma} \log(r)$	$\gamma \in 2\mathbb{N}$

several authors [1, 4, 8, 14] have pointed out the big impact the choice of c can have on the accuracy of the interpolant.

This choice is limited by the fact that the system of Eqs. 2 and 3 becomes ill-conditioned for big values of c . Several articles [6–8] have been dedicated to developing algorithms that are able to compute a stable solution of these systems for any value of the scaling parameter. They also give illustrations of how RBF interpolation errors vary with c for some test cases where results are available both with ill-conditioning present and with its effect eliminated. From these it can be seen that for very big values of the scaling parameter big interpolation errors are caused mainly by ill-conditioning. If this effect cannot be eliminated, the choice of c is limited to values that allow a stable calculation of the interpolant but this interpolant need not be optimal.

When ill-conditioning is not an issue for the relevant range of parameter values, there usually is a value of c that minimizes the interpolation errors. These typically go down rapidly if c is increased until the trend is reversed [12] due to fact that RBF interpolants near the flat limit (i.e. $c \rightarrow \infty$) behave very similar to polynomial interpolants. This can produce Runge-type oscillations or other undesired artifacts near the boundaries blowing up the approximation errors. The curves in the left plot of Fig. 1 show how the root mean square (RMS) interpolation errors for Franke’s test function (F1 from the Appendix) depend on c . Interpolation is carried out with inverse multiquadrics based on 81 data points with three different alignments (see Section 4 for details). All curves assume a minimum at a value of c for which the condition number is moderate. Note the big impact of the choice of node locations on interpolation accuracy: alignment C-81 concentrates many node locations near the boundaries which reduces the undesired oscillations for big c ’s dramatically. The price for this, however, is a reduced node density in the interior which for F1 entails a lower optimal interpolation accuracy than obtained for E-81 and H-81.

The right plot of Fig. 1 shows an instance of a test function (F6 from the Appendix) where the optimal c is outside the range of values for which stable

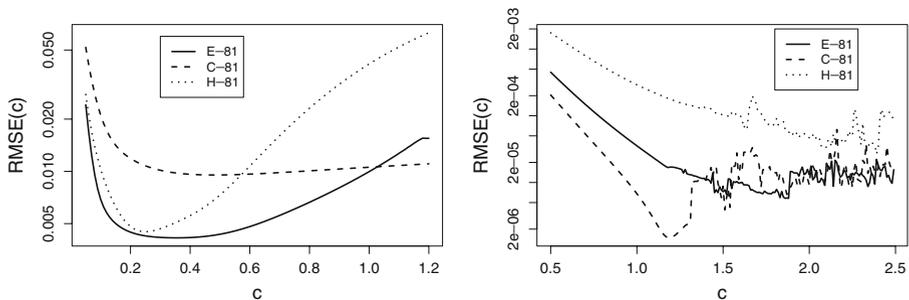


Fig. 1 RMSE curves for test function F1 (left) and F6 (right) with inverse multiquadrics for different alignments of 81 node locations: equidistant points, tensor-product Chebyshev point and Halton points

solutions are obtained. In this case a good procedure should select the best c that guarantees a stable solution.

While a variety of ad-hoc criteria for the choice of c have been proposed in the literature (e.g. [4, 11]), Rippa was the first to propose an algorithm based on the idea of leave-one-out cross validation (LOOCV), which is used in the statistics literature for a variety of parameter identification problems. We will give a brief review of this algorithm in Section 2 below and point out two aspects that could be criticized with this procedure.

In Section 3 we suggest a modification of the LOOCV-principle that takes these points into account and we show the connection to maximum likelihood estimation, a procedure used for identifying covariance parameters of stochastic processes in spatial statistics.

In Section 4, using some of the test functions from [14], we compare the performance of our procedure with that of two LOOCV-procedures (based on l_1 - and l_2 -norm of the LOOCV-errors) in selecting a value c that yields near-minimal RMSE.

In Section 5 we comment on the implementation of our algorithm and discuss a computationally efficient, approximative variant.

2 A short review of LOOCV and criticism

Cross validation is a very general idea that has long been used in the statistic literature. The algorithm proposed by Rippa corresponds to one of its variants called “leave-one-out” cross validation (LOOCV). In the RBF context the idea is to split off one single data point x_k at a time, calculate the partial fit $s_{f,[-k]}$ to all data pairs except (x_k, f_k) and from it the approximation error ε_k at x_k :

$$\varepsilon_k := f_k - s_{f,[-k]}(x_k), \quad k = 1, \dots, N$$

The parameter c is then chosen as the minimizer of some norm l_p of the error vector $\varepsilon = (\varepsilon_1, \dots, \varepsilon_N)'$, having in mind that $\|\varepsilon\|_p$ should behave similar to $\|f - s_{f,X}\|_{L_p(\Omega)}$. Indeed, [14] notes that in his experiments the graphs of the RMS error (which can be interpreted as a discrete approximation of $\|f - s_{f,X}\|_{L_2(\Omega)}$) are very similar to the graphs of $\|\varepsilon\|_1$ and $\|\varepsilon\|_2$.

A more recent paper by [3] discusses extensions of Rippa’s algorithm that can be applied in the setting of iterated approximate moving least squares approximation of function value data and for RBF pseudo-spectral methods for the solution of partial differential equations.

Another way of looking at LOOCV in the RBF context would be to interpret ε_k as the value at x_k of the error function $e_k := s_{f,X} - s_{f,[-k]}$, which is the error function of the surrogate problem of interpolating $s_{f,X}$ based on the data at the locations $X \setminus \{x_k\}$. This will be the starting point for a modification of LOOCV in the next section.

Although LOOCV was shown to work well for a variety of test functions and usually picks a value of c quite close to the optimal one, it might be possible

to further improve on it, since it does not fully account for the geometry of X , which can imply a loss of efficiency:

1. The fact that the same data (X, \mathbf{f}) is used for the calculation of every component ε_k , either as value to be predicted or as data the interpolant is fitted to, can lead to distortions, especially with irregular patterns of data points.
 To see this assume that the distance between two data points x_i and x_j is small compared with the average distance between data points. Then $s_{f,[-i]}(x_i)$ is determined mainly by f_j and vice versa, so that the components ε_i and ε_j basically contain the same information about $f - s_{f,X}$. This “redundancy” is not accounted for by LOOCV.
2. As already noted in the introduction (see especially Fig. 1), the accuracy of $s_{f,[-k]}$ as a predictor for f_k does not only depend on c , but strongly depends on the geometry of X . Prediction at data points near the boundaries or at isolated data points is more difficult (this is nicely illustrated in [2], Ch. 1) and hence, even for a good choice of c , the corresponding ε_k is usually bigger than for data points in the interior or in densely sampled areas of Ω . This is also not taken into account by LOOCV.

3 Variations to the LOOCV-algorithm

In this section we make two proposals on how the two aspects just mentioned can be taken into account, which will lead to an alternative procedure for selecting c .

As a first step we propose to replace the leave-one-out principle by a sequential approach, i.e. instead of using the data at all points $X \setminus \{x_k\}$ to predict f_k , we only use the data at the points $\{x_1, \dots, x_{k-1}\} =: X_{k-1}$.

Denoting by s_{f,X_k} the corresponding interpolant with the convention that $s_{f,X_0} \equiv 0$, the approximation errors now considered are

$$\tilde{\varepsilon}_k := f_k - s_{f,X_{k-1}}(x_k), \quad k = Q + 1, \dots, N$$

$\tilde{\varepsilon}_k$ can be interpreted as the value at x_k of the error function $\tilde{e}_k := s_{f,X_k} - s_{f,X_{k-1}}$ which is now only defined for $k > Q$. In this sequence of $N - Q$ surrogate interpolation problems each data pair is still used several times, but it now holds that

$$(\tilde{e}_i, \tilde{e}_j)_{\mathcal{N}_\phi(\Omega)} = 0, \quad \text{for all } Q < i \neq j \leq N \tag{4}$$

which follows from Lemma 10.24 in [19] by noting that for $i < j$

$$\tilde{e}_i \in \left\{ \sum_{n=1}^{j-1} \alpha_n \phi(\|\cdot - x_n\|) : \sum_{n=1}^{j-1} \alpha_n p(x_n) = 0 \text{ for all } p \in \mathcal{P} \right\} + \mathcal{P}$$

In other words: the error functions \tilde{e}_k are pairwise orthogonal with respect to the native space norm corresponding to ϕ and hence can be expected to

yield essentially different information about the interpolation behavior of the given data.

As a second variation to LOOCV we propose to take into account the magnitude of the approximation errors that can be expected on the basis of the geometry of X , and use it to weight the components of ε accordingly. The need for such an adjustment becomes even more necessary when passing from ε to $\tilde{\varepsilon}$ since the interpolants used to calculate the latter are based on a different amount of data which also heavily affects the potential accuracy.

A natural way to do so is to weight each component by the reciprocal power function that corresponds to the respective prediction, i.e. to pass from $\tilde{\varepsilon}_k$ to

$$\tilde{\tilde{\varepsilon}}_k := \frac{\tilde{\varepsilon}_k}{P_{\phi, X_{k-1}}(x_k)}, \quad k = Q + 1, \dots, N \tag{5}$$

where

$$P_{\phi, X_{k-1}}(x_k) = \sup_{\substack{g \in \mathcal{N}_{\phi(\Omega)} \\ |g|_{\mathcal{N}_{\phi(\Omega)}} \neq 0}} \frac{|(g - s_{g, X_{k-1}})(x_k)|}{|g|_{\mathcal{N}_{\phi(\Omega)}}} = \sup_{\substack{g \in \mathcal{N}_{\phi(\Omega)} \\ s_{g, X_{k-1}} = 0}} \frac{|g(x_k)|}{|g|_{\mathcal{N}_{\phi(\Omega)}}} \tag{6}$$

is the norm of the pointwise error functional of the interpolation process (see Definition 4.1.1 and Lemma 4.1.5 of [15]). It does not depend on f but gives information about the potential interpolation accuracy due to the geometry of X_k , and is therefore an ideal measure for the magnitude of $\tilde{\varepsilon}_k$.

Moreover, when the euclidean norm of $\tilde{\tilde{\varepsilon}}$ is used as the cost function to be minimized, one gets a nicely interpretable, and conveniently computable expression:

$$\left\| \tilde{\tilde{\varepsilon}} \right\|_2^2 = \sum_{n=Q+1}^N \frac{(\tilde{\varepsilon}_n(x_n))^2}{P_{\phi, X_{n-1}}^2(x_n)} = \sum_{n=Q+1}^N |\tilde{\varepsilon}_n|_{\mathcal{N}_{\phi(\Omega)}}^2 = |s_{f, X}|_{\mathcal{N}_{\phi(\Omega)}}^2. \tag{7}$$

The second equality follows from the fact, that $\tilde{\varepsilon}_n$ is of the form (1) with $X = X_n$ and hence is known to minimize the native space norm over all functions g with the same values on X_n , so that it attains the maximum in (6). The last equality follows by writing $s_{f, X}$ as a telescope sum and repeated use of (4).

A particular consequence of (7) is that $\left\| \tilde{\tilde{\varepsilon}} \right\|_2$ has the desirable property that it does not depend on the ordering of x_1, \dots, x_N . Therefore in the following we will concentrate on the l_2 -norm of our weighted errors.

Minimizing $\left\| \tilde{\tilde{\varepsilon}} \right\|_2$ is however not yet the final solution, because the weighting by the reciprocal power function entails a serious problem: while it accounts for the different magnitudes of the $\tilde{\varepsilon}_k$, it induces a bias towards selecting small values of c .

This can be explained by noting that $P_{\phi, X_{k-1}}(x_k)$ itself depends on ϕ (and hence on c), and so $\tilde{\tilde{\varepsilon}}_k$ gets small not only if the interpolation is good, but also if the power function is big. Numerical experiments (see e.g. [2], Sec. 17.1.2) suggest that the power function gets smaller if c is increased (at least as long as ill-conditioning does not play a role). From heuristic arguments in

the framework of spatial statistics, where $P_{\phi, X_{k-1}}(x_k)$ also appears as a measure of accuracy and bigger c 's can be interpreted as a stronger similarity of the function values at close-by data sites, it can be assumed that this relation between c and $P_{\phi, X_{k-1}}(x_k)$ holds in general. This explains the preference of small values of c .

Example 1 This bias is most explicit if ϕ is of the power type (see Table 1). Noting that in this case $\phi_c(\cdot) = c^{-\gamma} \phi(\cdot)$ one can easily verify that

$$\alpha^{(c)} := c^\gamma \cdot \alpha, \quad \beta^{(c)} := \beta$$

solves the corresponding system (2), (3). Now c^γ cancels out in (1), so the interpolants $s_{f, X}^{(c)}$ corresponding to ϕ_c do not depend on c , and a good procedure should automatically account for that. However, noting (cf. [19], Ch. 10) that

$$|s_{f, X}|_{\mathcal{N}_{\phi(\Omega)}} = \sum_{i,j=1}^N \alpha_i \alpha_j \phi(\|x_i - x_j\|) \stackrel{(2)}{=} \sum_{j=1}^N \alpha_j \left(f_j - \sum_{k=1}^Q \beta_k P_k(x_j) \right) \stackrel{(3)}{=} \alpha' \mathbf{f},$$

one obtains for the scaled interpolant

$$|s_{f, X}^{(c)}|_{\mathcal{N}_{\phi(\Omega)}} = (\alpha^{(c)})' \mathbf{f} = c^\gamma \cdot \alpha' \mathbf{f} = c^\gamma \cdot |s_{f, X}|_{\mathcal{N}_{\phi(\Omega)}}$$

so a minimization of $\|\tilde{\varepsilon}\|_2$ would select $c = 0$.

The above arguments show, that an additional factor $\psi(c)$ is needed that penalizes big values of the power functions involved. A first idea might be to use the arithmetic mean

$$\psi_A^2(c) = \frac{1}{N - Q} \sum_{n=Q+1}^N P_{\phi, X_{n-1}}^2(x_n). \tag{8}$$

However, the values of $P_{\phi, X_{k-1}}^2(x_k)$, $k = Q + 1, \dots, N$ have very different magnitudes so $\partial \psi_A^2 / \partial c$ would be dominated by the first terms in (8). Using the geometric mean

$$\psi_G^2(c) = \sqrt[N-Q]{\prod_{n=Q+1}^N P_{\phi, X_{n-1}}^2(x_n)} \tag{9}$$

instead, amounts to averaging on a logarithmic scale, and this seems much more appropriate since the terms

$$\frac{\partial}{\partial c} \log(P_{\phi, X_{k-1}}^2(x_k)) = \frac{\frac{\partial}{\partial c} P_{\phi, X_{k-1}}^2(x_k)}{P_{\phi, X_{k-1}}^2(x_k)}, \quad k = Q + 1, \dots, N$$

can more reasonably be assumed to be of the same magnitude, so that the dependence of the penalty factor ψ_G on c is uniformly based on all the power functions involved.

Using ψ_G is plausible, but seems to be an ad hoc choice only. Yet, it has another desirable property:

Proposition 1 *Let $A = (\phi_c(\|x_i - x_j\|)) \in \mathbb{R}^{N \times N}$ and $P = (p_j(x_i)) \in \mathbb{R}^{N \times Q}$ the matrices corresponding to the linear system (2), (3). Then ψ_G has the representation*

$$\psi_G^2(c) \sim \sqrt[N-Q]{(-1)^Q \begin{vmatrix} A & P \\ P' & \mathbf{0} \end{vmatrix}} = \sqrt[N-Q]{|A| |P' A^{-1} P|} \tag{10}$$

with a proportionality constant that does not depend on c .

In particular ψ_G is independent of the ordering of x_1, \dots, x_N .

Proof We write the system (2), (3) in the form

$$\begin{pmatrix} \mathbf{0} & P' \\ P & A \end{pmatrix} \begin{pmatrix} \beta \\ \alpha \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{f} \end{pmatrix}$$

and prove the assertion of the proposition by induction.

Let $A_n \in \mathbb{R}^{n \times n}$ and $P_n \in \mathbb{R}^{n \times Q}$ denote the sub-matrices of A and P that correspond to the system for the first n data points only, further let

$$M_n := \begin{pmatrix} \mathbf{0} & P'_n \\ P_n & A_n \end{pmatrix},$$

$$a_n := (\phi_c(\|x_n - x_1\|), \dots, \phi_c(\|x_n - x_{n-1}\|))' \text{ and } p_n := (p_1(x_n), \dots, p_Q(x_n))'.$$

Let's first assume, that $\{p_1, \dots, p_Q\}$ is a Lagrange basis of \mathcal{P} . In this case $P_Q = \text{Id}_Q$ and we have

$$|M_Q| = \begin{vmatrix} \mathbf{0} & \text{Id}_Q \\ \text{Id}_Q & A_Q \end{vmatrix} = (-1)^Q \begin{vmatrix} \text{Id}_Q & A_Q \\ \mathbf{0} & \text{Id}_Q \end{vmatrix} = (-1)^Q$$

Now let $Q < n \leq N$. If M_n is written in block form as M_{n-1} augmented by the n -th row and n -th column

$$M_n = \begin{pmatrix} \mathbf{0} & P'_{n-1} & p_n \\ P_{n-1} & A_{n-1} & a_n \\ P'_n & a'_n & \phi(0) \end{pmatrix}$$

with Schur complement

$$S_n = \phi(0) - \begin{pmatrix} p_n \\ a_n \end{pmatrix}' \begin{pmatrix} \mathbf{0} & P'_{n-1} \\ P_{n-1} & A_{n-1} \end{pmatrix}^{-1} \begin{pmatrix} p_n \\ a_n \end{pmatrix}.$$

Then, using basic results about block matrices (cf. e.g. [13]), we obtain $|M_n| = |M_{n-1}| \cdot S_n$, and it remains to show that $S_n = P_{\phi, x_{n-1}}^2(x_n)$.

To see this look at the interpolant s_{g, X_n} of the values $g_k = \delta_{kn}$, $k = 1, \dots, n$ at the data points in X_n . Its native space norm is then found to be

$$|s_{g, X_n}|_{\mathcal{N}_{\phi(\Omega)}}^2 = \alpha'_g \mathbf{g} = \begin{pmatrix} \beta_g \\ \alpha_g \end{pmatrix}' \begin{pmatrix} \mathbf{0} \\ \mathbf{g} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{g} \end{pmatrix}' M_n^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{g} \end{pmatrix} = S_n^{-1}$$

where the last equality is a result about the inversion of block matrices.

For s_{g, X_n} however, we can use the same arguments as for \tilde{e}_n in (7) to see that it attains the maximum in (6) and we obtain

$$P_{\phi, X_{n-1}}(x_n) = \frac{|s_{g, X_n}(x_n)|}{|s_{g, X_n}|_{\mathcal{N}_{\phi(\Omega)}}} = \frac{1}{S_n^{-1/2}} = \sqrt{S_n}$$

and hence, by induction, the first representation in (10) follows after horizontally and vertically switching the four blocks of M_N to the usual form.

The second representation in (10) follows by noting that the Schur complement with respect to these four blocks is given by $S = -P'A^{-1}P$ and applying once again the calculation rule for determinants of block matrices.

If $\{p_1, \dots, p_Q\}$ is not a Lagrange basis, there exists a regular matrix T so that

$$(\tilde{p}_1, \dots, \tilde{p}_Q) := (p_1, \dots, p_Q) T$$

is a Lagrange basis, and hence $\tilde{P} = PT$ has the desired form.

But then, we have

$$|A| |P'A^{-1}P| = |A| |T' P'A^{-1}PT| / |T|^2 = |A| |\tilde{P}'A^{-1}\tilde{P}| / |T|^2$$

so the determinant only changes by a factor that is independent of c . □

We can now give a short explicit formulation for our proposed procedure for selecting a good value c^* of the scaling parameter:

$$\begin{aligned} c^* &= \operatorname{argmin}_c \left\{ \psi_G^2 \cdot |s_{f, X}|_{\mathcal{N}_{\phi(\Omega)}}^2 \right\} \\ &= \operatorname{argmin}_c \left\{ \sqrt[(-1)^Q]{(-1)^Q} \left| \begin{matrix} A & P \\ P' & \mathbf{0} \end{matrix} \right| \cdot \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}' \begin{pmatrix} A & P \\ P' & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix} \right\} \end{aligned} \tag{11}$$

Remark 1 (Example 1 continued)

With the same arguments used to explain (7) we see that for the power type RBFs from Example 1 it holds that

$$P_{\phi, X_{k-1}}^{(c)}(x_k) = \frac{|\tilde{e}_k^{(c)}(x_k)|}{|\tilde{e}_k^{(c)}|_{\mathcal{N}_{\phi(\Omega)}}} = \frac{|\tilde{e}_k(x_k)|}{c^\gamma \cdot |\tilde{e}_k|_{\mathcal{N}_{\phi(\Omega)}}} = c^{-\gamma} \cdot P_{\phi, X_{k-1}}(x_k)$$

where we used that $\tilde{e}_k^{(c)}$ is of the form (1) and therefore $|\tilde{e}_k^{(c)}|_{\mathcal{N}_{\phi(\Omega)}} = c^\gamma \cdot |\tilde{e}_k|_{\mathcal{N}_{\phi(\Omega)}}$.

As a consequence, the cost function in (11) is constant for power type RBFs, which reasonably reflects the fact that their corresponding interpolants are scale invariant.

Remark 2 (Interpretation)

Both factors in (11) can be interpreted with respect to their role in the minimization procedure. While the minimization of ψ_G aims at choosing c so that the power functions in the corresponding native space become small, the minimization of $|s_{f,X}|_{\mathcal{N}_{\phi(\Omega)}}$ aims at choosing values of c for which f has small native space norm. Both objectives are desirable in the light of the standard bound on approximation errors

$$|f(x) - s_{f,X}(x)| \leq P_{\phi,X}(x) \cdot |f|_{\mathcal{N}_{\phi(\Omega)}}$$

and (11) realizes a compromise between the two objectives for the sequence of orthogonal surrogate interpolation problems created from the data.

Such a compromise seems reasonable also in the light of the discussion in Sec. 17.1.2 of [2], where it is noted that the power function is a useful indicator for a good scale parameter but that the dependence of $|f|_{\mathcal{N}_{\phi(\Omega)}}$ on c should somehow be acknowledged.

Remark 3 (Connection to spatial statistics)

In spatial statistics, a procedure called “Kriging” is used for the interpolation of paths of spatial stochastic processes (cf. e.g. [16] for details). Kriging is identical to RBF interpolation, but in the statistical context the term $\tau \cdot \phi(\|x_i - x_j\|)$ has the additional interpretation as the covariance of the function values at x_i and x_j where the additional parameter $\tau \in (0, \infty)$ denotes the process variance.

One popular approach taken there to identify τ and the Kernel parameters is the method of *maximum likelihood estimation* (MLE) which under the additional assumption of Gaussian distribution and for given covariance parameters assigns a likelihood to the observed values of f . The logarithm of this likelihood is given by (see [16], pp. 169–171)

$$\begin{aligned} l(c, \tau; \mathbf{f}) &= -\frac{N-Q}{2} \log(2\pi) - \frac{1}{2} \log(|\tau A| |P'(\tau A)^{-1} P|) \\ &\quad - \frac{1}{2} \mathbf{f}' \left((\tau A)^{-1} + (\tau A)^{-1} P (\tau^{-1} S)^{-1} P' (\tau A)^{-1} \right) \mathbf{f} \\ &= -\frac{N-Q}{2} \log(2\pi) - \frac{N-Q}{2} \log(\tau) - \frac{1}{2} \log(|A| |P' A^{-1} P|) \\ &\quad - \frac{1}{2\tau} \mathbf{f}' (A^{-1} + A^{-1} P S^{-1} P' A^{-1}) \mathbf{f} \end{aligned} \tag{12}$$

where $S = -P' A^{-1} P$ is again the Schur complement of the matrix $\begin{pmatrix} A & P \\ P' & \mathbf{0} \end{pmatrix}$.

The parameters are then chosen as the maximizers of this (log-)likelihood. Now for any fixed value of c the maximizer of (12) in τ is given by

$$\tau^* = \frac{1}{N-Q} \mathbf{f}' (A^{-1} + A^{-1} P S^{-1} P' A^{-1}) \mathbf{f}$$

Plugging this back into (12) yields the so-called *profile log-likelihood*

$$\begin{aligned}
 l(c; \mathbf{f}) = & -\frac{N-Q}{2} \log(2\pi) - \frac{N-Q}{2} (1 - \log(N-Q)) \\
 & - \frac{N-Q}{2} \log(\mathbf{f}' (A^{-1} + A^{-1}PS^{-1}P'A^{-1}) \mathbf{f}) \\
 & - \frac{1}{2} \log(|A| |P'A^{-1}P|)
 \end{aligned} \tag{13}$$

and since

$$|s_{f,X}|_{\mathcal{N}_{\phi(\Omega)}}^2 = \alpha' \mathbf{f} = \mathbf{f}' (A^{-1} + A^{-1}PS^{-1}P'A^{-1}) \mathbf{f} \tag{14}$$

one can see that maximizing (13) in c is equivalent to (11). In the context of spatial statistics the MLE is usually considered superior in terms of accuracy compared to other common methods.

Remark 4 (Limits of applicability)

A crucial difference to the situation in spatial statistics arises when $f \in \mathcal{N}_{\phi(\Omega)}$ (note that this is a common, but not a necessary assumption in RBF interpolation). While in spatial statistics the adjusted error components $\tilde{\varepsilon}_k$ in (5) are indeed all of the same magnitude, this cannot be true if $f \in \mathcal{N}_{\phi(\Omega)}$. In fact this implies $\tilde{\varepsilon}_k \rightarrow 0$ as $k \rightarrow \infty$ which follows from (7) and from the fact that $|s_{f,X}|_{\mathcal{N}_{\phi(\Omega)}} \leq |f|_{\mathcal{N}_{\phi(\Omega)}}$.

If the sequence $(|s_{f,X_n}|_{\mathcal{N}_{\phi(\Omega)}})_{n \in \mathbb{N}}$ converges to $|f|_{\mathcal{N}_{\phi(\Omega)}}$ only slowly, then for moderate N and $1 \leq n \leq N$ the increase should be approximately linear and so the weighting in (5) is reasonable. If the convergence is quite fast however, then the decrease of the magnitude of $\tilde{\varepsilon}_k$ in k is clearly visible even for moderate N , as shown in Section 4. As a consequence, the first components have a stronger influence on the minimization process, and since they are based on fewer data points, they will typically favor bigger values of c .

Remark 5 (Disadvantages of our weighting of the errors)

There is another substantial difference to the situation in spatial statistics. In that framework the model assumptions usually imply that the behavior of f does not radically differ from one subdomain of Ω to another. In the framework of numerical analysis, however, there is no such assumption, and so it may happen that just near the boundaries, where approximation is generally more difficult anyway, the function f has a structure that is particularly hard to recover. In this case LOOCV is likely to perform better than our method since the big approximation errors near the boundaries will be reflected in big LOOCV error components corresponding to data sites in this area. Due to the weighting in (5) the dominance of those error components is reduced in procedure (11) and a value of c is typically selected that works well in the interior but worse than the LOOCV choice near the boundaries where the really big errors occur.

It should be noted, however, that varying the scale parameter is not the most appropriate way to deal with the challenges of approximation near the boundaries anyway. Various RBF edge correction strategies have been proposed (see [5] for an overview) and shown to be quite effective.

Remark 6 (Selecting the other RBF parameters)

Many of the examples of RBFs in Table 1 contain another parameter γ , the choice of which is sometimes even more crucial, e.g. in the case of Matérn type RBFs, where it determines the smoothness of the native space f is supposed to belong to. In the absence of prior knowledge about an appropriate value for γ , both (11) and LOOCV can of course be used to select a good value for both c and γ simultaneously. For (11) this is successful practice in spatial statistics.

4 Numerical examples

In this section we compare the performance of procedure (11), denoted by **MLE**, with that of Rippa's LOOCV algorithm based on the l_1 -norm (**CV1**), and the l_2 -norm (**CV2**) of ε .

4.1 Experimental design

Experiments are conducted for several test functions f on $[0, 1]^2$ (see appendix for definitions and plots), mostly taken from [14]. Since the choice of the node locations is another important factor with big impact on accuracy and on the appropriate choice of c we use different numbers N and different alignments of data points:

- equidistant points (E-N)
- tensor-product Chebyshev points (C-N)
- Halton points (H-N)

The tensor-product Chebyshev points are taken from [2] and have the advantage that they provide more information about f near the boundaries where the approximation accuracy is often lower. Halton points are an example of a quasi-random number sequence and we use them to represent the situation of scattered data. For details about their definition we refer to [10] and [20].

For a given point set X and given RBF ϕ the corresponding interpolant $s_{f,X}$ is calculated and its values on an equidistant 100×100 grid on $[0, 1]^2$ are compared with those of f . As a criteria for the quality of the interpolation we calculate the RMSE, the root of the mean squared interpolation errors $f(x) - s_{f,X}(x)$, on this grid for different choices of c within a reasonable range. From this RMSE-curve the optimal c can be read off.

Following the observation of Rippa that the results are comparable for Gaussian, multiquadric and inverse multiquadric RBFs, we have only used the latter in our experiments.

Table 2 Relative performance of MLE over CV2 with respect to the RMSE achieved by using inverse multiquadrics with c_{MLE} or c_{CV2} , respectively

	F0	F1	F3	F5	F6	F9	F14	F15
E-25			+	+	---			--
C-25		++	+	+	---		+	
H-25		+	+	+	---			-
E-81	--		+++	++	ic	++	+++	ic
C-81	--		++	+++	ic		+	ic
H-81	+	-	-	---	ic			ic
E-289	ic	++	-	ic	ic		+	ic
C-289	ic			ic	ic	+		ic
H-289	ic	--	ic	ic	ic			ic

4.2 General remarks on the results

The different test functions we study exhibit very different qualitative behaviors with respect to the optimal choice of the scaling parameter. As already discussed in the introduction, F6 is an example of a function that calls for relatively big values of c which leads to ill-conditioned systems even for $N = 81$. The same problem arises with F15 (for $N = 81/289$), F0, F5 (for $N = 289$), and F3 (for H-289). Whenever we suspected from the appearance of the RMSE-curve that ill-conditioning is present for the relevant range of c , we did not use it for the comparison of the algorithms and marked these cases with “ic” instead.

4.3 Comparison of the different algorithms

In Table 2 we give an overview of the relative performance of MLE and CV2 (which has, according to Section 3, the closest relation to MLE) by comparing the RMSEs of the interpolants corresponding to c_{MLE} and c_{CV2} . The coding is as follows: if $RMSE(c_{CV2})$ is more than 5%/20%/100% larger than $RMSE(c_{MLE})$ (i.e. MLE performs better) we put +/ ++/ +++. Conversely, if

Table 3 Optimal c and c selected by the different algorithms for test function F5 together with the RMSEs of the corresponding approximations

	c_{opt}	$RMSE(c_{opt})$	c_{MLE}	$RMSE(c_{MLE})$	c_{CV1}	$RMSE(c_{CV1})$	c_{CV2}	$RMSE(c_{CV2})$
E-25	0.27	$1.468 \cdot 10^{-3}$	0.20	$4.104 \cdot 10^{-3}$	0.20	$4.104 \cdot 10^{-3}$	0.40	$4.747 \cdot 10^{-3}$
C-25	0.18	$7.560 \cdot 10^{-3}$	0.28	$1.073 \cdot 10^{-2}$	0.16	$8.259 \cdot 10^{-3}$	0.37	$1.258 \cdot 10^{-2}$
H-25	0.31	$5.789 \cdot 10^{-3}$	0.34	$5.882 \cdot 10^{-3}$	0.42	$6.719 \cdot 10^{-3}$	0.39	$6.330 \cdot 10^{-3}$
E-81	0.58	$4.012 \cdot 10^{-6}$	0.59	$5.355 \cdot 10^{-6}$	0.62	$2.002 \cdot 10^{-5}$	0.69	$7.297 \cdot 10^{-5}$
C-81	0.39	$5.230 \cdot 10^{-5}$	0.54	$1.180 \cdot 10^{-4}$	0.77	$2.357 \cdot 10^{-4}$	0.95	$3.644 \cdot 10^{-4}$
H-81	0.45	$1.006 \cdot 10^{-5}$	0.61	$4.268 \cdot 10^{-5}$	0.46	$1.050 \cdot 10^{-5}$	0.48	$1.255 \cdot 10^{-5}$
E-289	0.66	$2.345 \cdot 10^{-10}$	0.78	$3.633 \cdot 10^{-10}$	0.71	$1.422 \cdot 10^{-9}$	0.71	$1.422 \cdot 10^{-9}$
C-289	0.78	$7.099 \cdot 10^{-11}$	0.77	$1.054 \cdot 10^{-10}$	0.72	$3.009 \cdot 10^{-10}$	0.72	$3.009 \cdot 10^{-10}$
H-289	0.71	$1.004 \cdot 10^{-9}$	0.76	$1.946 \cdot 10^{-9}$	0.67	$1.039 \cdot 10^{-9}$	0.67	$1.039 \cdot 10^{-9}$

Table 4 Optimal c and c selected by the different algorithms for test function F14 together with the RMSEs of the corresponding approximations

	c_{opt}	$\text{RMSE}(c_{\text{opt}})$	c_{MLE}	$\text{RMSE}(c_{\text{MLE}})$	c_{CV1}	$\text{RMSE}(c_{\text{CV1}})$	c_{CV2}	$\text{RMSE}(c_{\text{CV2}})$
E-25	0.15	$1.723 \cdot 10^{-1}$	0.46	$1.914 \cdot 10^{-1}$	0.67	$2.065 \cdot 10^{-1}$	0.57	$1.992 \cdot 10^{-1}$
C-25	0.52	$2.452 \cdot 10^{-1}$	0.55	$2.453 \cdot 10^{-1}$	7.50	$2.781 \cdot 10^{-1}$	7.50	$2.781 \cdot 10^{-1}$
H-25	0.27	$1.320 \cdot 10^{-1}$	0.25	$1.330 \cdot 10^{-1}$	0.21	$1.396 \cdot 10^{-1}$	0.21	$1.396 \cdot 10^{-1}$
E-81	0.17	$4.640 \cdot 10^{-2}$	0.29	$4.887 \cdot 10^{-2}$	1.37	$1.547 \cdot 10^{-1}$	1.39	$1.565 \cdot 10^{-1}$
C-81	0.44	$1.024 \cdot 10^{-1}$	0.38	$1.026 \cdot 10^{-1}$	1.54	$1.111 \cdot 10^{-1}$	1.54	$1.111 \cdot 10^{-1}$
H-81	0.27	$4.006 \cdot 10^{-2}$	0.23	$4.009 \cdot 10^{-2}$	0.21	$4.050 \cdot 10^{-2}$	0.24	$4.006 \cdot 10^{-2}$
E-289	0.21	$3.576 \cdot 10^{-3}$	0.23	$3.611 \cdot 10^{-3}$	0.31	$4.240 \cdot 10^{-3}$	0.29	$3.997 \cdot 10^{-3}$
C-289	0.47	$1.655 \cdot 10^{-2}$	0.30	$1.705 \cdot 10^{-2}$	0.50	$1.663 \cdot 10^{-2}$	0.57	$1.675 \cdot 10^{-2}$
H-289	0.22	$4.058 \cdot 10^{-3}$	0.21	$4.154 \cdot 10^{-3}$	0.21	$4.154 \cdot 10^{-3}$	0.21	$4.154 \cdot 10^{-3}$

$\text{RMSE}(c_{\text{MLE}})$ exceeds $\text{RMSE}(c_{\text{CV2}})$ by 5%/20%/100% we put -/ -/ - ---. If both methods perform similar the cell is left empty.

Overall MLE performs slightly superior but it does not outperform CV2 as clearly as it often does in statistical applications. Some of the effects leading to an inferior performance can be explained via Remarks 4 and 5 from above.

For test function F5 and F14 results are given in detail (Tables 3 and 4) since this pair of test functions can be used to illustrate the caveat about our algorithm we pointed out in Remark 4. We observe that F5 can be interpolated well even with few data points. The consequence is a fast convergence of the native-space norms of interpolants which can be seen in Fig. 2 for the point set C-81 (to avoid big jumps that blur the overall trend, the sequence of data points was reordered: for $k = 2, \dots, N$ we let x_k the maximizer of $|f(x) - s_{f, X_{k-1}}(x)|$). While for F14 the native-space norms increase almost linearly and the increments $\tilde{\varepsilon}_k^2$ are really of the same magnitude, those for F5 noticeably tend to 0 and hence have decreasing influence on the selection of c . The resulting bias towards bigger values of c , that can be expected according

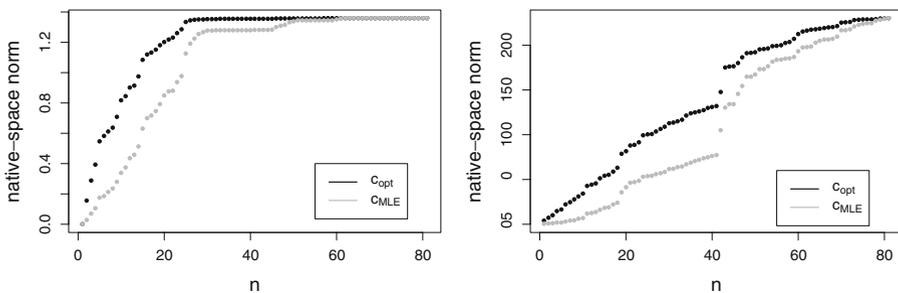


Fig. 2 Sequences of the native-space norms $(|s_{f, X_n}|_{\mathcal{N}_{\phi}(\Omega)})_{n=1, \dots, N}$ for F5 (left) and F14 (right). The norms are calculated for point set C-81 with $c = c_{\text{opt}}$ and $c = c_{\text{MLE}}$. The c_{MLE} -curves are rescaled so that their values for $n = 81$ coincide with those of the c_{opt} -curves

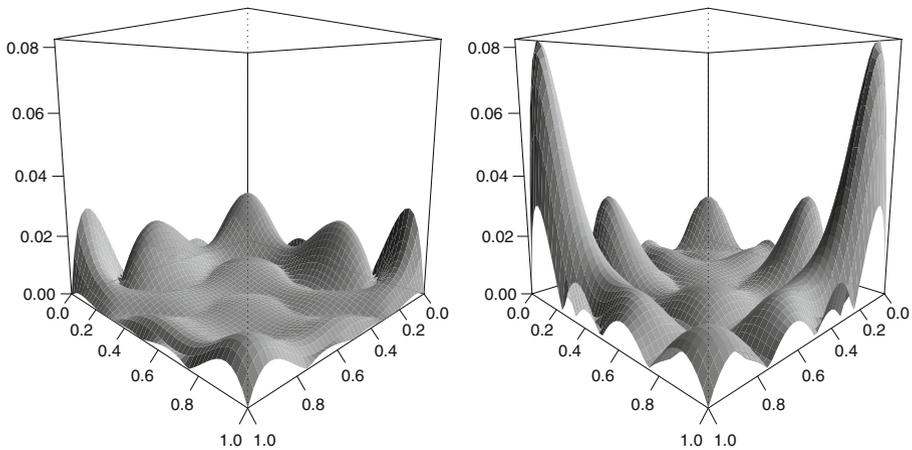


Fig. 3 Approximation errors of the interpolation of F_{15} on 25 equidistant points for $c = c_{\text{opt}}$ ($= 1.45$) (left) and $c = c_{\text{MLE}}$ ($= 0.71$) (right)

to Remark 4, can be observed in Table 3. The sequence of the native-space norms calculated with $c = c_{\text{MLE}}$ (also depicted in Fig. 2) illustrates how MLE tries to “correct” the different magnitudes of $\tilde{\varepsilon}_k^2$. It should be noted, however, that this systematic error is often compensated by the smaller dispersion c_{MLE} around the optimal c compared to c_{CV1} and c_{CV2} .

The bad performance of MLE with test function F_6 and F_{15} can be partly explained via Remark 5. While a good reconstruction near the corners requires relatively big values of c this does not necessarily hold for the interior. For reasons pointed out in Remark 5 our procedure reduces the emphasis of error components near the boundaries, and therefore the selected values of c tend to be too small (see Fig. 3).

5 Computational issues

First of all, we shall explain how the target function in (11) can be calculated in a stable and efficient way. It is equivalent (but more convenient) to minimize its logarithm:

Algorithm 1 (Calculation of the log target function for given c)

1. Calculate the singular value decomposition $A = UDV'$ of A
2. Issue a warning message if the condition of A exceeds a certain threshold
3. Calculate $U'P$, $P'V$, $U'\mathbf{f}$, $\mathbf{f}'V$, and from it
 - $S = -P'A^{-1}P = -(P'V)D^{-1}(U'P)$
 - $|s_{f,X}|_{\mathcal{N}_{\phi}(\Omega)}^2 = (\mathbf{f}'V) (D^{-1} + D^{-1}(U'P)S^{-1}(P'V)D^{-1}) (U'\mathbf{f})$ (see (14))

4. Calculate $\log(\psi_G^2(c)) = \frac{1}{N-Q} (\log |A| + \log |-S|) = \frac{1}{N-Q} (\sum_i \log(D_{ii}) + \log |-S|)$
5. The log target function is then $\log(|s_{f,X}|^2_{\mathcal{N}_{\phi(\Omega)}}) + \log(\psi_G^2(c))$

When $\mathcal{P} = \{0\}$ all terms involving P or P' vanish. Minimization of the log target function should be carried out such that only those values of c are considered that do not lead to ill-conditioned systems.

Computational costs and stability The computationally most expensive step is the singular value decomposition of A . Such a step, however, is also needed for the calculation of RBF expansion coefficients when direct solvers are used, and so the computational costs are of the same order $O(N^3)$ (which is also the order of the LOOCV-procedures).

The log determinant of A can then be obtained as a sum of logarithms and its calculation is just as stable as the calculation of $|s_{f,X}|_{\mathcal{N}_{\phi(\Omega)}}$. In fact we even found that the MLE target function (11) depends more smoothly on c than those of the LOOCV-procedures (see Fig. 4), which encourages the use of gradient based optimization procedures.

Approximative schemes For big N even the order $O(N^3)$ of the computational costs is too high and moreover the calculations become increasingly unstable. In the spatial statistics literature a number of approximative schemes for the statistical counterpart of (11) have been proposed, and we shall briefly outline one of them that can easily be transferred to the present setup. Its main idea is to turn back to the original representation of the cost function with the two factors

$$\| \text{err} \|_2^2 = \sum_{n=Q+1}^N \frac{(f_n - s_{f,X_{n-1}}(x_n))^2}{P_{\phi, X_{n-1}}^2(x_n)} \quad \text{and} \quad \psi_G^2 = \sqrt{\prod_{n=Q+1}^N P_{\phi, X_{n-1}}^2(x_n)} \tag{15}$$

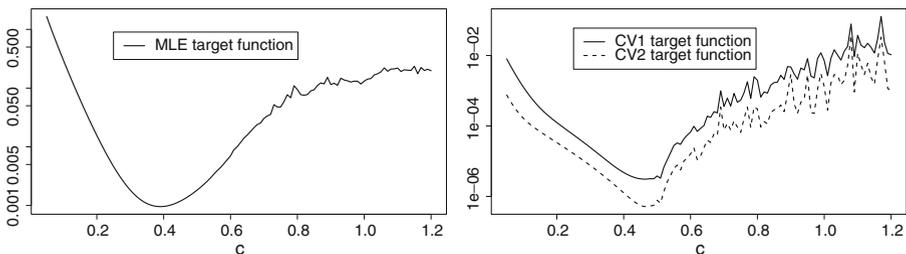


Fig. 4 Target functions of MLE, CV1, and CV2 for test function F1 and point set E-289

and to assume that $s_{f, X_{n-1}}$ and $P_{\phi, X_{n-1}}^2$ can be approximated well by $s_{f, \bar{X}_{n-1}}$ and $P_{\phi, \bar{X}_{n-1}}^2$, where \bar{X}_{n-1} is a subset of X_{n-1} containing at most M (some fixed number) points:

Algorithm 2 (Calculation of the approximate log target function for given c and M)

1. For $n = Q + 1, \dots, N$:
 - Choose an interpolation set $\bar{X}_{n-1} \subset X_{n-1}$ for f_n of size $\min(M, n - 1)$
 - Calculate the interpolant $s_{f, \bar{X}_{n-1}}$ and the power function $P_{\phi, \bar{X}_{n-1}}$ at x_n
2. The approximate log target function is then

$$\log \left(\sum_{n=Q+1}^N \frac{(f_n - s_{f, \bar{X}_{n-1}}(x_n))^2}{P_{\phi, \bar{X}_{n-1}}^2(x_n)} \right) + \frac{1}{N - Q} \sum_{n=Q+1}^N \log \left(P_{\phi, \bar{X}_{n-1}}^2(x_n) \right)$$

The computational cost can thus be reduced to an order of $O(NM^3)$ which can be a substantial saving if $M \ll N$, and the smaller equation systems of size $M + Q$ are also more stable than the original one. The M data points chosen from X_{n-1} are typically those closest to x_n . If the optimal scale is expected to be large, however, the approximation can be improved considerably if some points further apart are also included (see [17] for a detailed discussion).

Note that the approximate cost function is no longer independent of the ordering of X . Our experiments with test functions F9 and F14 (for these examples the optimal c is still within a range of values that lead to stable interpolants for $N = 1,089$) do not entirely confirm the observation of [18] and [17] that the differences for different orderings are small. When we keep the original, very systematic ordering (row by row in our experiment with E-1089) the approximation in the test case F14 is very bad. However, when using some random permutation of the original ordering, the values of c selected by Algorithm 2 are usually close to c_{MLE} .

In Table 5 we give results for point set E-1089. The interpolation sets X_{n-1} in Algorithm 2 are chosen to be the 50 closest neighbors of x_n from X_{n-1} . For the random orderings the intervals containing all c 's from 20 different experiments are given.

Table 5 Optimal c and c selected by the different algorithms (Algorithm 2 is denoted by AMLE) for test function F9 and F14 and point set E-1089

	c_{opt}	c_{MLE}	c_{AMLE} (orig. ord.)	c_{AMLE} (rand. ord.)	c_{CV1}	c_{CV2}
F9	0.08	0.1	0.1	{0.1}	0.14	0.215
F14	0.21	0.21	0.35	[0.21, 0.23]	0.24	0.25

6 Discussion

In this paper we have introduced a procedure for selecting a near-optimal value for the parameter c in RBF interpolation. Our procedure (11) was motivated as an extension of the LOOCV-algorithm proposed by Rippa and was shown to be equivalent to the so-called maximum likelihood estimator that is used for a similar problem in spatial statistics.

Numerical experiments were presented that showed that our procedure is superior to LOOCV in some test cases but that it can fail in situations where either the native space norm of the sequence of interpolants converges very fast to that of the function f to be interpolated, or where the behavior of f in the interior of the domain of interest is very different from its behavior near the boundaries of this domain. Overall it performs comparable to the LOOCV procedures, but it has a more solid mathematical background.

The importance of a good choice of c was motivated by showing that one often needs to compromise between a good approximation accuracy in the interior, which often improves as c increases, and near the boundaries, where big c 's can produce undesired artifacts. Another way of dealing with these conflicting objectives is to let the shape parameter vary spatially, rather than assigning a single value to it (cf. e.g. [9]). This idea is very effective in combination with local node clustering allowing to improve the resolution wherever this might be needed while the parameter c in these areas is chosen such that the local refinement does not entail any Runge-type errors.

It may be worthwhile goal to extend procedure (11) so that it can be used to select a scale parameter that may vary spatially (e.g. depending on the local node density) but is adapted to the particular function f to be approximated. This seems to be quite challenging because with spatially varying c the solid mathematical ground of reproducing kernel Hilbert spaces is left, and it is not even clear how the definition (6) of the power function can be extended to this framework. The discussion in Section 2, however, suggests that procedure (11) may be more adapted to the situation of clustered nodes than LOOCV, and may help to achieve a further improvement of approximation accuracy with minimal number of data points.

We conclude that (11) is a good alternative to Rippa's algorithm for selecting the parameter c in RBF interpolation with further potential of improvement. Whenever prior knowledge about the nature of f suggests that the caveats mentioned in Remarks 4 and 5 do not apply its use can definitely be recommended.

Acknowledgements The author is grateful to Martin Schlather, Robert Schaback and Stefan Müller for hints and helpful discussions. The feedback of the referees is also highly acknowledged. Their comments were throughout very useful and helped to improve this article significantly.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

Appendix: The test functions

The test functions used above are a selection of those used in [14] with the same enumeration. Examples F2 and F4 have been omitted due to their similarity to F14 and F5 respectively, in lieu thereof an additional test function F0 was added which is a superposition of some trigonometric functions and therefore quite different from the other examples (Figs. 5, 6, 7 and 8).

Note that not all of the test functions (F9 is an obvious exception) are in the native space of inverse multiquadrics.

$$F1 = 0.75 \exp\left(-\frac{(9x - 2)^2 + (9y - 2)^2}{4}\right) + 0.75 \exp\left(-\frac{(9x + 1)^2}{49} - \frac{9y + 1}{10}\right) + 0.5 \exp\left(-\frac{(9x - 7)^2 + (9y - 3)^2}{4}\right) - 0.2 \exp(-(9x - 4)^2 - (9y - 7)^2)$$

$$F0 = \sum_{i=1}^5 (-1)^i \cos((-1.5)^i x + 3^{i/2} y), \quad F3 = \frac{1.25 + \cos(5.4 y)}{6(1 + (3x - 1)^2)}$$

$$F5 = \frac{\exp\left(-\frac{81}{4}((x - 0.5)^2 + (y - 0.5)^2)\right)}{3} \quad (\text{Scaled Gaussian Kernel})$$

$$F6 = \frac{\sqrt{64 - 81((x - 0.5)^2 + (y - 0.5)^2)}}{9} - 0.5$$

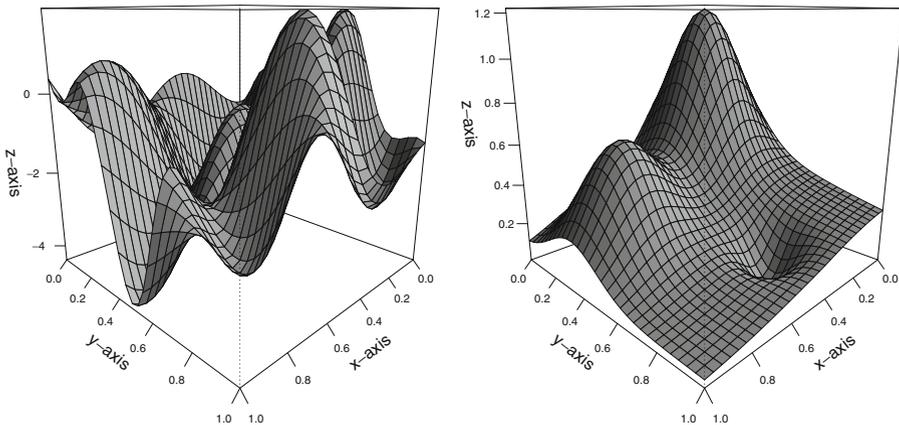


Fig. 5 Perspective view of F0 and F1

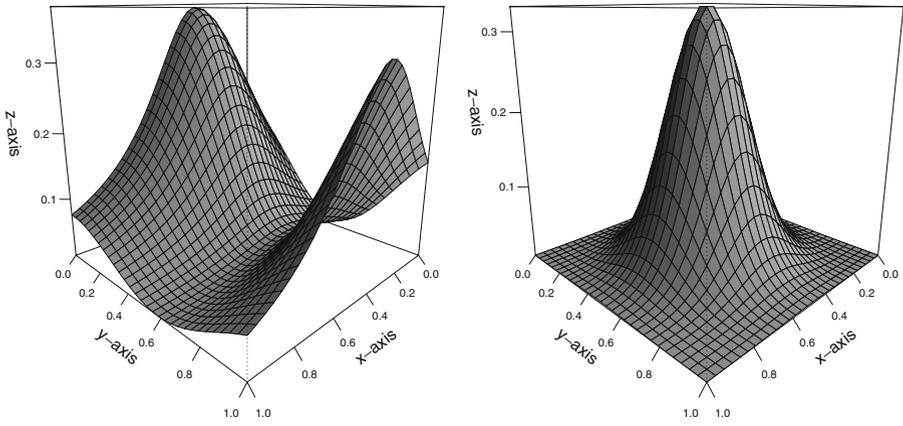


Fig. 6 Perspective view of F_3 and F_5

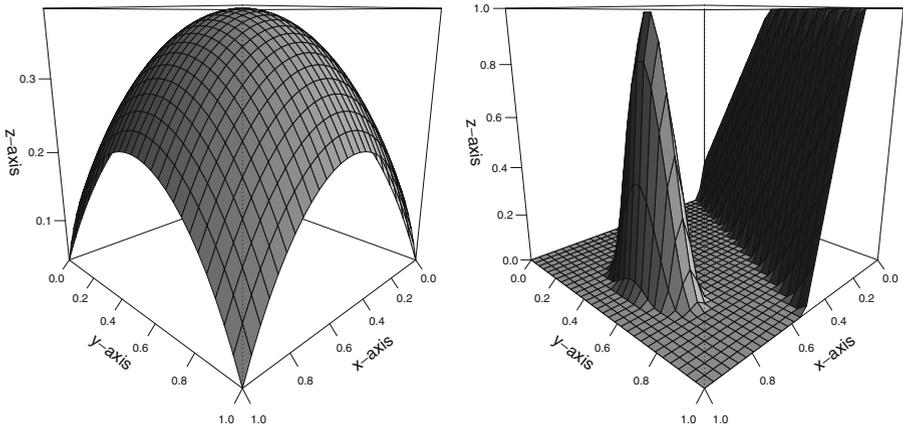


Fig. 7 Perspective view of F_6 and F_9

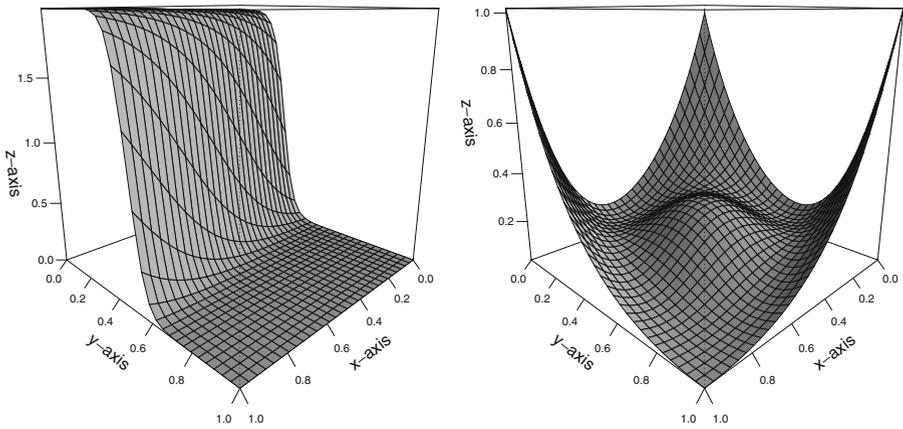


Fig. 8 Perspective view of F_{14} and F_{15}

$$F9 = \begin{cases} 1 & \text{if } y - \xi \geq 1/2, \\ 2(y - \xi) & \text{if } 0 \leq y - \xi \leq 1/2, \\ (\cos(4\pi r(\xi, y)) + 1)/2 & \text{if } r(\xi, y) \leq 1/4, \\ 0, & \text{otherwise} \end{cases}$$

where

$$r(\xi, y) = \sqrt{\left(\xi - \frac{3}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2}, \quad \xi = 2.1x - 0.1$$

$$F14 = \tanh\left(-3(0.595576(y + 3.79762)^2 - x - 10)\right) + 1$$

$$F15 = \left(1 - \frac{x}{2}\right)^6 \left(1 - \frac{y}{2}\right)^6 + 1000(1-x)^3 x^3 (1-y)^3 y^3 + y^6 \left(1 - \frac{x}{2}\right)^6 + x^6 \left(1 - \frac{y}{2}\right)^6$$

References

- Carlson, R.E., Foley, T.A.: The parameter R^2 in multiquadric interpolation. *Comp. Math. Appl.* **21**, 29–42 (1991)
- Fasshauer, G.E.: *Meshfree Approximation Methods with Matlab*. World Scientific Publishers, Singapore (2007)
- Fasshauer, G.E., Zhang, J.G.: On choosing “optimal” shape parameters for RBF approximation. *Numer. Algorithms* **45**, 345–368 (2007)
- Franke, R.: Scattered data interpolation: tests of some methods. *Math. Comput.* **38**, 181–200 (1982)
- Fornberg, B., Driscoll, T.A., Wright, G., Charles, R.: Observations on the behaviour of radial basis function approximations near boundaries. *Comput. Math. Appl.* **43**, 473–490 (2002)
- Fornberg, B., Piret, C.: A stable algorithm for flat radial basis functions on a sphere. *SIAM J. Sci. Comput.* **30**, 60–80 (2007)
- Fornberg, B., Piret, C.: On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere. *J. Comput. Phys.* **227**, 2758–2780 (2008)
- Fornberg, B., Wright, G.: Stable computation of multiquadric interpolants for all values of the shape parameter. *Comput. Math. Appl.* **47**, 497–523 (2004)
- Fornberg, B., Zuev, J.: The Runge phenomenon and spatially variable shape parameters in RBF interpolation. *Comput. Math. Appl.* **54**, 379–398 (2007)
- Halton, J.H.: On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.* **2**, 84–90 (1960)
- Hardy, R.L.: Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res.* **76**, 1905–1915 (1971)
- Larsson, E., Fornberg, B.: Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions. *Comput. Math. Appl.* **49**, 103–130 (2005)
- Rao, C.R., Rao, M.B.: *Matrix Algebra and Its Applications to Statistics and Econometrics*. World Scientific, Singapore (1998)
- Rippa, S.: An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Adv. Comput. Math.* **11**, 193–210 (1999)
- Schaback, R.: *Reconstruction of multivariate functions from scattered data*. Monograph (1997)
- Stein, M.L.: *Interpolation of Spatial Data: Some Theory of Kriging*. Springer, New York (1999)
- Stein, M.L., Chi, Z., Welty, L.J.: Approximating likelihoods for large spatial data sets. *J. R. Stat. Soc. B* **66**, 275–296 (2004)

18. Vecchia, A.V.: Estimation and model identification for continuous spatial processes. *J. R. Stat. Soc. B* **50**, 297–312 (1988)
19. Wendland, H.: *Scattered Data Approximation*. Cambridge University Press, Cambridge (2005)
20. Wong, T.-T., Luk, W.-S., Heng, P.-A.: Sampling with Hammersley and Halton points. *J. Graphics Tools* **2**, 9–24 (1997)