Anisotropic weights for RBF-PU interpolation with subdomains of variable shapes

R. Cavoretto[1], A. De Rossi[2], G. E. Fasshauer[3], M. J. McCourt[4], and E. Perracchione[5]

[1] Department of Mathematics "Giuseppe Peano", University of Torino, Italy, roberto.cavoretto@unito.it
[2] Department of Mathematics "Giuseppe Peano", University of Torino, Italy, alessandra.derossi@unito.it
[3] Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden, CO, USA, fasshauer@mines.edu
[4] SigOpt, Inc., San Francisco, CA, USA, mccourt@sigopt.com
[5] Department of Mathematics "Tullio Levi-Civita", University of Padova, Italy, emma.perracchione@math.unipd.it

Abstract. The partition of unity (PU) method, performed with local radial basis function (RBF) approximants, has already been proved to be an effective tool for solving interpolation or collocation problems when large data sets are considered. It decomposes the original domain into several *subdomains* or *patches* so that only linear systems of relatively *small* size need to be solved. In research on such partition of unity methods, such subdomains usually consist of spherical patches of a fixed radius. However, for particular data sets, such as track data, ellipsoidal patches seem to be more suitable. Therefore, in this paper, we propose a scheme based on *a priori* error estimates for selecting the sizes of such variable ellipsoidal subdomains. We jointly solve for both these domain decomposition parameters and the anisotropic RBF shape parameters on each subdomain to achieve superior accuracy in comparison to the standard partition of unity method.

1 Introduction

Radial basis function (RBF)-based methods [4] find their natural applications in various fields, such as image reconstruction, resolution of partial differential equations and population dynamics. Two common computational issues arising when we deal with real situations involve creating approximations from a *large* number of scattered points and the one of producing accurate approximations despite ill-conditioned linear systems.

In this article, we attack the first issue with an efficient computation by means of the Partition of Unity (PU) method [8]. It enables us to decompose the original interpolation problem (involving a matrix of the same size as the amount of data) into many small ones defined on subdomains/patches of the original domain. However, the design of these subdomains (and consequently the number of points lying on each patch) affects the accuracy of the approximation.

Generally, when the PU method is used for scattered data interpolation, the PU subdomains are assumed to be balls of a fixed radius [2]. In [3], a local approach was proposed via the PU method that selects *optimal local approximants*: both the shape parameter and the patch radius were selected such that error estimates were minimized. This strategy was more effective at dealing with points that were inconsistently distributed throughout the domain.

This previous work allowing varying patch radii was limited to only spherical patches. Such a scheme is not completely suitable for particular data distributions, such as track data [1]. To address this situation we propose to use ellipsoidal subdomains, allowing one free domain parameter per dimension. This adaptation requires careful selection of the PU weights to guarantee consistency, which we detail in our proposal.

To match the anisotropic structure of these subdomains, we use anisotropic Wendland's functions [7] to form our local approximants. The values of the shape parameters and of the semi-axes of patches are selected by minimizing theoretical error estimates. In particular, as in [3], we focus our attention on the Leave One Out Cross Validation (LOOCV) scheme [6]. By jointly optimizing for both the domain and RBF parameters, we are able to improve on the computational cost from [3]. We also factor in the impact of ill-conditioning during this optimal parameter search to balance accuracy and stability.

The outline of the paper is as follows. In Section 2, we briefly review the main theoretical aspects of the RBF-PU method. Section 3 is devoted to the presentation of the proposed scheme which makes use of ellipsoidal patches. Numerical experiments are presented in Section 4. Section 5 deals with conclusions and work in progress.

2 The RBF-based Partition of Unity method

The approximation problem considered in this paper is formulated as follows. Consider a set $\mathcal{X}_N = \{\boldsymbol{x}_i,\ i = 1, \ldots, N\} \subseteq \Omega$ of distinct data points (or data sites or nodes), arbitrarily distributed on a domain $\Omega \subseteq \mathbb{R}^M$, with an associated set $\mathcal{F}_N = \{f_i = f(\boldsymbol{x}_i),\ i = 1, \ldots, N\}$ of data values (or measurements or function values), which are obtained by sampling some (unknown) function $f : \Omega \longrightarrow \mathbb{R}$ at the nodes $\boldsymbol{x}_i$. The *scattered data interpolation* problem consists of finding a function $R : \Omega \longrightarrow \mathbb{R}$ such that $R\left(\boldsymbol{x}_i\right) = f_i$, $i = 1, \ldots, N$.

To this end, we take $R \in H_{\Phi}(\mathcal{X}_N) = \mathrm{span}\{\Phi(\cdot, \boldsymbol{x}_i),\ \boldsymbol{x}_i \in \mathcal{X}_N\}$, where $\Phi : \Omega \times \Omega \longrightarrow \mathbb{R}$ is a strictly positive definite and symmetric kernel. More specifically, we take RBFs (radial kernels), and thus, we suppose that there exist a function $\phi : [0, \infty) \to \mathbb{R}$ and a shape parameter $\varepsilon > 0$ such that for all $\boldsymbol{x}, \boldsymbol{y} \in \Omega$ we have $\boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{y}) = \phi_\varepsilon(||\boldsymbol{x} - \boldsymbol{y}||_2) := \phi(r)$. In Table 1, we list the strictly positive definite RBFs that will be used later. Note that the RBFs depend on a shape parameter $\varepsilon > 0$ that significantly affects the accuracy of the approximation. We will later refer to the functions reported in Table 1 as isotropic kernels, meaning that ε is a scalar.

Table 1. Examples of strictly positive definite isotropic radial kernels.

RBF	$\phi(r)$
Inverse MultiQuadric C^{∞} (IMQ)	$(1+\varepsilon^2 r^2)^{-1/2}$
Matérn C^2 (M2)	$\mathrm{e}^{-\varepsilon r}(\varepsilon r+1)$
Wendland C^2 (W2)	$\max\left(1-\varepsilon r, 0\right)^4 \left(4\varepsilon r+1\right)$

By using RBFs, the interpolant assumes the form

$$R(\boldsymbol{x}) = \sum_{k=1}^{N} \alpha_k \phi(||\boldsymbol{x}-\boldsymbol{x}_k||_2), \quad \boldsymbol{x} \in \Omega. \tag{1}$$

The coefficients $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)^T$ in (1) are determined by solving the linear system $\mathsf{A}\boldsymbol{\alpha} = \boldsymbol{f}$, where the entries of the matrix $\mathsf{A} \in \mathbb{R}^{N\times N}$ are given by $(\mathsf{A})_{ik} = \phi(||\boldsymbol{x}_i - \boldsymbol{x}_k||_2)$, $i,k = 1, \ldots, N$, and $\boldsymbol{f} = (f_1, \ldots, f_N)^T$. The uniqueness of the solution is ensured by the fact that the kernel Φ is strictly positive definite and symmetric.

One drawback of this method is the computational cost associated with the solution of potentially large linear systems. The PU method, presented below, enables us to overcome such issue. At first, we cover the domain Ω with d overlapping subdomains Ω_j. To be more precise, we require a regular covering, i.e., $\{\Omega_j\}_{j=1}^d$ must fulfill the following properties:

i. for each $\boldsymbol{x} \in \Omega$, the number of subdomains Ω_j, with $\boldsymbol{x} \in \Omega_j$, is bounded by a global constant C_1,
ii. each subdomain Ω_j satisfies an interior cone condition,
iii. the local fill distances $h_{\mathcal{X}_{N_j}}$ are uniformly bounded by the global fill distance $h_{\mathcal{X}_N}$, where $\mathcal{X}_{N_j} = \mathcal{X}_N \cap \Omega_j$.

Once we select weight functions W_j, $j = 1, \ldots, d$, the PU interpolant can be defined as

$$\mathcal{I}\left(\boldsymbol{x}\right) = \sum_{j=1}^{d} R_j\left(\boldsymbol{x}\right) W_j\left(\boldsymbol{x}\right), \quad \text{with} \quad R_j\left(\boldsymbol{x}\right) = \sum_{k=1}^{N_j} \alpha_k^j \phi(||\boldsymbol{x}-\boldsymbol{x}_k^j||_2),$$

where R_j is defined on the subdomain Ω_j, N_j indicates the number of points on Ω_j and $\boldsymbol{x}_k^j \in \mathcal{X}_{N_j}$, with $k = 1, \ldots, N_j$. Therefore, the problem leads to solving d linear systems of the form $\mathsf{A}_j\boldsymbol{\alpha}_j = \boldsymbol{f}_j$, where $\boldsymbol{\alpha}_j = (\alpha_1^j, \ldots, \alpha_{N_j}^j)^T$, $\boldsymbol{f}_j = (f_1^j, \ldots, f_{N_j}^j)^T$ and the entries of $\mathsf{A}_j \in \mathbb{R}^{N_j\times N_j}$ are given by $(\mathsf{A}_j)_{ik} = \phi(||\boldsymbol{x}_i^j - \boldsymbol{x}_k^j||_2)$, $i,k = 1, \ldots, N_j$.

Since the coefficients of the local interpolants are determined by imposing the local interpolation conditions, the functions W_j, $j = 1, \ldots, d$, must form a partition of unity. Moreover, we also require that such partition of unity is k-stable (see, e.g. [7] Def. 15.16, p. 276), which in particular implies that $\mathrm{supp}(W_j) \subseteq \Omega_j$. For instance, such conditions are satisfied for the well-known *Shepard's weights*; refer e.g. to [8] for further details.

3 Optimal local interpolants for the RBF-based PU method

We now focus on the selection of the PU patches, and we remove the standard assumption that they consist of balls of a fixed radius. Therefore we consider ellipsoidal patches, i.e., each Ω_j is defined through its semi-axes $\boldsymbol{\delta}_j = (\delta_1^j, \ldots, \delta_M^j)$. Moreover, in what follows, we use anisotropic kernels. We remark that any isotropic radial kernel can be turned into an anisotropic one by using a weighted 2-norm instead of an unweighted one. Thus, to fix the ideas, on a subdomain Ω_j it is enough to replace the scalar value of the shape parameter ε_j with a symmetric positive definite matrix E_j. More precisely, we consider the special case for which $\mathsf{E}_j = \mathrm{diag}(\varepsilon_1^j, \ldots, \varepsilon_M^j)$. This allows us to choose a different scaling along the dimensions of the problem.

Our parametrization strategy consists of selecting both $\boldsymbol{\varepsilon}_j = (\varepsilon_1^j, \ldots, \varepsilon_M^j)$ and $\boldsymbol{\delta}_j$ such that the error estimates on Ω_j are minimized. This study is motivated by the fact that the PU approximation error is governed by the local ones ([7] Th. 15.19, p. 277).

3.1 Local error estimates

For a general overview about error estimates refer e.g. to [4]. Here we focus on error predictions that are popular in statistics, and precisely on cross validation schemes. We describe the cross validation algorithm that is applied on a given Ω_j to get a local *a priori* error estimate [4]. At first, we split the set $\mathcal{X}_{N_j}$ into two disjoint subsets: a training set $\mathcal{T}_{N_j^t}$ and a validation set $\mathcal{V}_{N_j^v}$ such that $N_j^t + N_j^v = N_j$. The set $\mathcal{T}_{N_j^t}$ is used to construct a surrogate or partial approximation that is validated via the set $\mathcal{V}_{N_j^v}$. To simplify the following discussion, on Ω_j we introduce the following block decomposition of the local interpolation matrix

$$\mathsf{A}_j = \begin{pmatrix} \mathsf{A}_j^{tt} & \mathsf{A}_j^{tv} \\ \mathsf{A}_j^{vt} & \mathsf{A}_j^{vv} \end{pmatrix},$$

where, for example, the block A_j^{tv} is generated using training points to evaluate and validation data as centers of the kernels. Similarly, we partition $\boldsymbol{\alpha}_j = (\boldsymbol{\alpha}_j^t, \boldsymbol{\alpha}_j^v)^T$, $\boldsymbol{f}_j = (\boldsymbol{f}_j^t, \boldsymbol{f}_j^v)^T$. With this notation, the prediction at the points in the validation set using the training set is $\mathsf{A}_j^{vt}(\mathsf{A}_j^{tt})^{-1}\boldsymbol{f}_j^t$. In other words, $|\boldsymbol{f}_j^v - \mathsf{A}_j^{vt}(\mathsf{A}_j^{tt})^{-1}\boldsymbol{f}_j^t|$ provides information about the accuracy of the fit on the j-th subdomain. Then, we take q partitions of $\mathcal{V}_{N_j^v}$, $\mathcal{V}_{N_j^v} = \{\mathcal{V}_{N_j^v}^{(1)}, \ldots, \mathcal{V}_{N_j^v}^{(q)}\}$, such that

$$\mathcal{V}^{(k)}_{N^v_j} \cap \mathcal{V}^{(i)}_{N^v_j} = \emptyset, \text{ for } i \neq k, \text{ with } \cup_{k=1}^{q} \mathcal{V}^{(k)}_{N^v_j} = \mathcal{X}_{N_j}, \text{ and } \mathcal{T}^{(k)}_{N^t_j} = \mathcal{X}^{(k)}_{N_j} \backslash \mathcal{V}^{(k)}_{N^v_j}.$$

Thus, as error estimate we consider the residual left over by the interpolants evaluated at the validation sets. The LOOCV scheme is a particular case of the general setting presented above for which $q = N_j$ and each $\mathcal{V}^{(k)}_{N^v_j} = \boldsymbol{x}^j_k$. Moreover for the LOOCV scheme, A^{vv}_j is the diagonal element of A^{-1}_j and thus, being a scalar, the computation simplifies. Indeed, as error estimate for the j-th subdomain we have (see also [6])

$$e_j = ||(\alpha^j_1/(\mathsf{A}^{-1}_j)_{11}, \, \dots, \, \alpha^j_{N_j}/(\mathsf{A}^{-1}_j)_{N_j N_j})||_p,$$

where in what follows we fix the index of the discrete norm to $p = 2$.

In our PU context, using both anisotropic kernels and ellipsoidal patches, we have that $e_j = e_j(\boldsymbol{\varepsilon_j}, \boldsymbol{\delta_j})$. In fact, the shape parameter affects the accuracy of the RBF approximant and, for the PU method, the accuracy also depends on which points are involved in the computation of the local interpolants.

3.2 Description of the PU-LOOCV method

To minimize the LOOCV error estimates we use a multivariate optimization tool. This allows us to reduce the computational cost of the procedure presented in [3]. To be more precise, we consider the Nelder-Mead simplex algorithm [5]. Without going into details, we remark that on Ω_j, given an initial guess $(\boldsymbol{\varepsilon}^0_j, \boldsymbol{\delta}^0_j)$, it provides an approximation of the optimal values by computing subsequent simplices and only needs function evaluations of the objective function [5]. In particular, for the implementation we use the Matlab software and the `fminsearch.m` routine. Of course, we also need to impose the following constraints on the parameters we optimize:

$$\varepsilon^j_k > 0, \quad \text{and} \quad \delta^+ \geq \delta^j_k \geq \delta^*, \quad k = 1, \dots, M, \quad j = 1, \dots, d,$$

where $\delta^* \in \mathbb{R}$ is chosen so that patches form a covering of the domain and $\delta^+ \in \mathbb{R}$ is selected so that for each $\boldsymbol{x} \in \Omega$ the number of subdomains Ω_j, with $\boldsymbol{x} \in \Omega_j$, is bounded.

In this way, after optimizing the parameters $(\boldsymbol{\varepsilon}_j, \boldsymbol{\delta}_j)$, $j = 1, \dots, d$, we have a PU covering made of ellipsoids that is also regular. In fact, each of the patches satisfies an interior cone condition. This assumption is trivially verified for balls. However, it is also true for ellipsoids (see, e.g. [7] Pr. 11.26, p. 195).

Finally, in order to build a consistent PU setting, we also need to carefully choose the compactly supported functions for the PU Shepard's weights, which are here constructed with the W2 function. Since we have ellipsoids, we need to select the anisotropic form of the W2 function. Thus, $\boldsymbol{\varepsilon}_j$, which identifies the support of the compactly supported RBF, is taken so that $\text{supp}(W_j) = \Omega_j$, $j = 1, \dots, d$.

4 Numerical experiments

Our experiments focus on bivariate interpolation. In Subsection 4.1, we show the numerical results obtained by considering known functions and artificial track data [1]. Then, in Subsection 4.2 we also take into account real data by analyzing an application to Earth's topography.

4.1 Experiments with artificial data

To illustrate the accuracy of the proposed method, we evaluate the interpolant on a grid of $s = 40^2$ points $\tilde{\boldsymbol{x}}_i$, $i = 1, \dots, s$, on $\Omega = [0,1]^2$ and we calculate Root Mean Square Error (RMSE). The patch centers are constructed as a grid of t^2 points on Ω, where t is the number of tracks. Of course this design will affect the accuracy of the approximation. Nevertheless, the scheme proposed here, allowing to choose variable subdomains, is consequently less sensitive to this choice.

We show numerical results obtained by considering five sets of track data on $\Omega = [0,1]^2$ sampled from the *2D Franke's function*, see e.g. [4]. In particular, the results of using LOOCV to optimize the semi-axes of the patches and the shape parameters of the local basis functions are reported in Table 2. They are compared with the classical PU method obtained by taking a grid of t^2 points on Ω as PU centers and a fixed patch radius $\delta = \delta^*$. Furthermore, we consider the isotropic IMQ kernel with shape parameter equal to 1. We select such shape parameter arbitrarily. Indeed, one of the main advantages of the proposed method is the one of automatically choosing *safe* values for the shape parameters. Of course, different values might lead to more accurate approximations, but it is not possible to provide a priori optimal or safe shape parameters. Moreover, we also report the CPU times. Tests have been carried out with the Matlab software on an Intel(R) Core(TM) i7-6500U CPU 2.59 GHz processor. In Figure 1 (left), we show an example of 2000 track data [1] and the ellipsoidal patches obtained via PU-LOOCV.

From the numerical experiments we can note that the classical PU method, which makes use of circular patches, is not able to accurately fit the data, especially when a large number of points is involved. This might be due to a non-optimal selection of the shape parameter and/or of the patch size for the classical PU. In this sense, the PU-LOOCV reveals its robustness, selecting optimal values for those parameters and providing accurate results also when N grows. Finally, we point out that the proposed scheme, besides extending the idea of the method presented in [3] to subdomains having different shapes and to anisotropic kernels, thanks to the use of an optimization routine for the minimization problem, it also speeds up the procedure. For instance, using the same scheme outlined in [3] would take about 300 s for 1000 data. Moreover, note that Table 2 shows that the PU-LOOCV with 1000 points is more than twice as accurate *and* more than twice as fast than the classical PU method with 160000 points.

Table 2. RMSEs, and CPU times obtained by using the PU-LOOCV and classical PU methods.

N	RMSE		CPU time	
	PU	PU-LOOCV	PU	PU-LOOCV
1000 (20 × 50)	3.27E–3	8.02E–5	0.5	43.3
2000 (25 × 80)	1.08E–3	2.58E–5	1.2	64.7
4000 (40 × 100)	1.28E–3	4.67E–6	7.2	136.0
8000 (50 × 160)	3.48E–4	1.25E–6	17.4	189.0
16000 (80 × 200)	3.49E–4	4.55E–7	109.0	475.0

4.2 Experiments with real data

To test the method with real data, we consider points extracted from maps. We take, as example, a map of Korea (plotted in the right frame of Figure 1) and we extract 40 tracks containing 100 points. The function values, being real samples of the elevation above sea level, are truly oscillating and thus the interpolation problem is particularly challenging. We also extract a grid of 40^2 points to evaluate the error.

By using the M2 kernel, the RMSE for the PU-LOOCV is equal to 2.58E−2. The classical PU with a fixed radius $\delta = \delta^*$ completely fails. However, for the classical PU, taking anisotropic kernels with variable parameters and $\delta = 3\delta^*$ as fixed size of the circular patches gives RMSE=4.21E−2. In other words, we reach about the same accuracy, but this is not completely satisfying. Indeed, if we take circular subdomains of radius $3\delta^*$, the average of points on each patch is about 52, while the PU-LOOCV only requires on average 16 data per patch. Therefore, especially with real data, the approximation by means of an optimized PU method becomes essential.

5 Conclusions

In this paper we presented a scheme for the optimal selection of local approximants in the PU method. The scheme, based on ellipsoidal patches, is particularly suitable for track data. Work in progress consists in comparing the LOOCV scheme with other a priori error estimates and in selecting suitable locations for the patch centers. Indeed, here they are taken as grids of points, but this might be restrictive. Further investigations for more efficient optimization routines are also needed.

Acknowledgments

This research has been accomplished within RITA (Rete ITaliana di Approssimazione) and partially supported by GNCS-INδAM. The first and second authors were partially supported by the 2016-2017 project *Metodi numerici e*

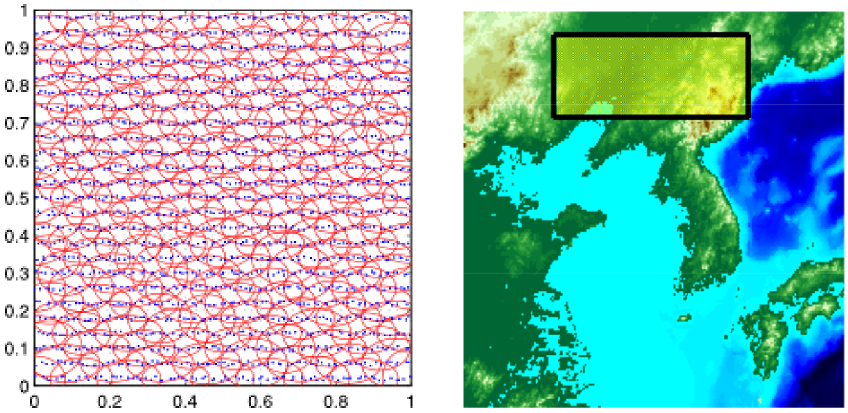

Fig. 1. Left: an illustrative example with 2000 track data that shows how patches are selected by means of the PU-LOOCV method. Right: the Korea's map and the extracted tracks.

computazionali per le scienze applicate of the Department of Mathematics of the University of Torino. The third author was partially supported by grant NSF-DMS #1522687. The last author is supported by the research project *Radial basis functions approximations: stability issues and applications*, No. BIRD167404.

References

1. G. Allasia, R. Besenghi, R. Cavoretto and A. De Rossi, *Scattered and track data interpolation using an efficient strip searching procedure*, Appl. Math. Comput. **217** (2011), 5949–5966.
2. R. Cavoretto, A. De Rossi and E. Perracchione, *Efficient computation of partition of unity interpolants through a block-based searching technique*, Comput. Math. Appl. **71** (2016), 2568–2584.
3. R. Cavoretto, A. De Rossi and E. Perracchione, *Optimal selection of local approximants in RBF-PU interpolation*, to appear in J. Sci. Comput.
4. G. E. Fasshauer and M. J. McCourt, *Kernel-based Approximation Methods using* Matlab, World Scientific, Singapore, 2015.
5. J. C. Lagarias, J. A. Reeds, M. H. Wright and P. E. Wright, *Convergence properties of the Nelder-Mead simplex method in low dimensions*, SIAM J. Optimiz. **9** (1998), 112–147.
6. S. Rippa, *An algorithm for selecting a good value for the parameter c in radial basis function interpolation*, Adv. Comput. Math. **11** (1999), 193–210.
7. H. Wendland, *Scattered data approximation*, Cambridge Monogr. Appl. Comput. Math., vol. 17, Cambridge Univ. Press, Cambridge, 2005.
8. H. Wendland, *Fast evaluation of radial basis functions: Methods based on partition of unity*, Approximation Theory X: Wavelets, Splines, and Applications, (C. K. Chui et al., eds.), Vanderbilt Univ. Press, Nashville, 2002, pp. 473–483.