Applied Mathematics and Computation 307 (2017) 113–123

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Kernel-based adaptive approximation of functions with discontinuities

Licia Lenarduzzi [a], Robert Schaback [b,*]

[a] C.N.R., I.M.A.T.I. Milano, Via Bassini 15, Milano 20133, Italy
[b] Institut für Numerische und Angewandte Mathematik, Universität Göttingen, Lotzestraße 16–18, Göttingen D-37083, Germany

ARTICLE INFO

MSC:
65D05
62H30
68T05

Keywords:
Kernels
Classification
Localized approximation
Adaptivity
scattered data

ABSTRACT

One of the basic principles of Approximation Theory is that the quality of approximations increase with the smoothness of the function to be approximated. Functions that are smooth in certain subdomains will have good approximations in those subdomains, and these *sub-approximations* can possibly be calculated efficiently in parallel, as long as the subdomains do not overlap. This paper proposes an algorithm that first calculates sub-approximations on non-overlapping subdomains, then extends the subdomains as much as possible and finally produces a global solution on the given domain by letting the subdomains fill the whole domain. Consequently, there will be no Gibbs phenomenon along the boundaries of the subdomains. The method detects faults and gradient faults with good accuracy. Throughout, the algorithm works for fixed scattered input data of the function itself, not on spectral data, and it does not resample.

1. Introduction

Assume that a large set $\{(\mathbf{x}_i, f_i), i = 1, \ldots, N\}$ of data is given, where the points $\mathbf{x}_i$ are scattered in $\mathbb{R}^d$ and form a set X. We want to find a function u that recovers the data on a domain Ω containing the points, i.e.

$$\begin{aligned} u &: \Omega \to \mathbb{R}, \\ u(\mathbf{x}_i) &\approx f_i, \ i = 1, \ldots, N. \end{aligned}$$

We are particularly interested in situations where the data have smooth interpolants in certain non-overlapping subdomains Ω_j, but not globally. The reason may be that there are discontinuities in the function itself or its derivatives. Thus a major goal is to identify subdomains $\Omega_j \subseteq \Omega, 1 \le j \le J$ and smooth functions $u_j, 1 \le j \le J$ such that

$$\begin{aligned} u_j &: \Omega_j \to \mathbb{R}, \\ u_j(\mathbf{x}_i) &\approx f_i \text{ for all } \mathbf{x}_i \in X \cap \Omega_j. \end{aligned}$$

The solution to the problem is piecewise defined as

$$u(\mathbf{x}) := u_j(\mathbf{x}) \text{ for all } \mathbf{x} \in \Omega_j, \ 1 \le j \le J.$$

* Corresponding author.
E-mail addresses: licia.lenarduzzi@mi.imati.cnr.it (L. Lenarduzzi), schaback@math.uni-goettingen.de, robert@schaback.com (R. Schaback).

http://dx.doi.org/10.1016/j.amc.2017.02.043
0096-3003/

Our motivation is the well-known fact that errors and convergence rates in Approximation Theory always improve with increasing smoothness. Thus on each subdomain we expect to get rather small errors, much smaller than if the problem would have been treated globally, where the non-smoothness is a serious limiting effect.

From the viewpoint of Machine Learning (see [7,20,21] for details) this is a mixture of classification and regression. The domain points have to be classified in such a way that on each class there is a good regression model. The given training data are used for both classification and regression, but in this case the classification is dependent on the regression, and the regression is dependent on the classification.

Furthermore, there is a serious amount of geometry hidden behind the problem. The subdomains should be connected, their interiors should be disjoint, and the union of their closures should fill the domain completely. This is why a black-box machine learning approach is not pursued here. Instead, Geometry and Approximation Theory play a dominant part. For the same reason, we avoid to calculate edges or fault lines first, followed by local approximations later. The approximation properties should determine the domains and their boundaries, not the other way round.

In particular, *localized approximation* will combine Geometry and Approximation Theory and provide a central tool, together with *adaptivity*. The basic idea is that in the interior of each subdomain, far away from its boundary, there should be a good and simple approximation to the data at each data point from the data of its neighbors.

There is quite an amount of literature on the subject, e.g. [1,2,4,5,9,10,12–15], but this approach differs in several aspects. There will be no a-priori assumptions on the form and placement of faults or singularities, and we expect the overall approximation quality to be not significantly reduced by these, because we focus on the errors in the subdomains, not on the total error. The data may be rather irregularly distributed, and we use a graph-theoretic method for finding disjoint subdomains. For the localized approximations, we use kernel-based methods instead of moving least squares or Shepard-type techniques because they automatically adapt to the best possible convergence rate under an unknown smoothness [16].

2. A basic tool

The following serves as an underpinning of various aspects of the adaptive algorithm that we shall describe below. We need a reliable criterion that tells us whether a data point $\mathbf{x}_i$ belongs to a subdomain Ω_j or another subdomain. As long as we construct a local interpolant on a subdomain Ω_j where the data come from a single smooth function, we have all tools of kernel-based interpolation at our disposal, and we can predict to some extent how the interpolation error will behave. If such a prediction fails on a new data point $\mathbf{x}_i$, this model is violated, and we shall try to assign $\mathbf{x}_i$ to another subdomain. We use the Hilbert space norms of the local interpolants as a criterion, and we now show why.

Assume that we have a kernel-based interpolant $s_{X,f}$ to data of a function f on a finite set X of centers, and let $P_X(x)$ be the associated Power Function. See the standard literature (e.g. [6,8,18,22]) on this subject for details. In particular, there is a strict error bound

$$|f(x) - s_{X,f}(x)| \le P_X(x)\|f\|_K$$

in the norm $\|.\|_K$ of the "native" Hilbert space where the kernel K is reproducing, and there is orthogonality

$$\|f\|_K = \|s_{X,f}\|_K^2 + \|f - s_{X,f}\|_K^2$$

due to the standard norm-minimality of the interpolant. If the set X gets large, the norms $\|s_{X,f}\|_K$ grow towards $\|f\|_K$. This is what is to be expected in this model situation for "good" local interpolation.

Furthermore, there is an exact update formula

$$\|s_{X\cup\{x\},f}\|_K^2 = \|s_{X,f}\|_K^2 + \frac{(f(x) - s_{X,f}(x))^2}{P_X^2(x)} \tag{1}$$

for the Hilbert space norms. See [19], but the basic recursion arguments are already in [17]. This shows that the increase of the norm of the interpolant after adding a new interpolation point x is small if the old interpolant already predicts the new function value $f(x)$ well. But if the new value comes from a different local function, the norm can be expected to jump up dramatically. This is confirmed by the numerical results at the end of this section.

If the set X is already large enough to let $\|s_{X,f}\|_K$ come close to $\|f\|_K$, then

$$\|s_{X,f}\|_K^2 \le \|s_{X,f}\|_K^2 + \frac{(f(x) - s_{X,f}(x))^2}{P_X^2(x)} = \|s_{X\cup\{x\},f}\|_K^2 \le \|f\|_K^2$$

does not leave much leeway in (1) for the increase of $\|s_{X,f}\|_K^2$ towards $\|f\|_K^2$, and any larger increase signalizes that the current model is violated, i.e. that the data do not come from a single function f in the native space of the kernel.

The Power Function value in the denominator is independent of f and cares for the geometric fact that if x is close to X, the new error in x must be considerably smaller than for points far from X, if the same increase of the norm of the interpolant is to be attained. Thus the norm of the interpolant allows a "geometry-corrected" evaluation of what will happen if x is added to X.

We add a characteristic example here. The function

$$f_4(x,y) := ((x-0.5)^2 + (y-0.5)^2)^{0.35} + 0.05\cdot(x-0.5)_+^0 \tag{2}$$