Scattered Data Approximation of Noisy Data via Iterated Moving Least Squares

Gregory E. Fasshauer and Jack G. Zhang

Abstract. In this paper we focus on two methods for multivariate approximation problems with non-uniformly distributed noisy data. The new approach proposed here is an iterated approximate moving least-squares method. We compare our method to ridge regression which filters out noise by using a smoothing parameter. Our goal is to find an optimal number of iterations for the iterative method and an optimal smoothing parameter for ridge regression so that the corresponding approximants do not exactly interpolate the given data but are reasonably close. For both approaches we implement variants of leave-one-out cross-validation in order to find these optimal values. The shape parameter for the basis functions is also optimized in our algorithms.

§1. Introduction

In multivariate data fitting problems we are usually given data (x_j, f_j), $j = 1, \dots, N$ with distinct $x_j \in \mathbb{R}^s$ and $f_j \in \mathbb{R}$, and we want to find a (continuous) function $\mathcal{P}_f : \mathbb{R}^s \to \mathbb{R}$ such that

$$\mathcal{P}_f(x_j) = f_j, \qquad j = 1, \dots, N. \tag{1}$$

For classical interpolation methods we assume $\mathcal{P}_f$ to be a linear combination of a set of basis functions ϕ_j, i.e.,

$$\mathcal{P}_f(x) = \sum_{j=1}^{N} c_j \phi_j(x). \tag{2}$$

The coefficients c_j are determined by satisfying the constraint (1). To guarantee existence of a unique set of c_j for arbitrary distinct x_j, it is known that, when $s > 1$ and $N > 1$, ϕ_j must be x_j-dependent. In the

Conference Title
Editors pp. 1–6.

ISBN 0-0-9728482-x-x

literature, the ϕ_j are frequently chosen to be basis functions generated by shifts of a single *strictly positive definite* basic function ϕ, i.e.,

$$\phi_j(x) = \phi(x - x_j). \tag{3}$$

Correspondingly, (2) is now rewritten as

$$\mathcal{P}_f(x) = \sum_{j=1}^{N} c_j \phi(x - x_j). \tag{4}$$

To express the problem in matrix-vector form we let

$$\mathbf{c} = [c_1, \dots, c_N]^T, \quad \mathbf{f} = [f_1, \dots, f_N]^T, \quad \Phi_{ij} = \phi(x_i - x_j).$$

Then, enforcing the interpolation constraint (1) with a $\mathcal{P}_f$ in the form of (4) leads to

$$\Phi\mathbf{c} = \mathbf{f}. \tag{5}$$

The fact that ϕ is assumed to be strictly positive definite guarantees that the interpolation matrix Φ is invertible. Therefore, $\mathbf{c} = \Phi^{-1}\mathbf{f}$.

Note that, in this classical setup, there is no restriction on the distribution of the data sites x_j except for being pair-wise distinct. However, distribution of the x_j is an important issue in the so-called *approximate moving least squares* (AMLS) approximation method. More details will be given later.

The definition of the basic function ϕ often involves a multiplicative parameter ε, e.g., the Gaussian $\phi(x) = e^{-\varepsilon^2 x^2}$. This shape parameter can be used to control the flatness of ϕ, and finding a good value for ε is a major issue of data approximation (see e.g., the recent paper [3]). Throughout the rest of this paper, we frequently omit ε in our notation, but the reader should be aware that some quantities, e.g., ϕ, Φ and $\mathbf{c}$ may be ε-dependent.

The function $\mathcal{P}_f$ based on (5) exactly interpolates the given data. When N is large, solving this (often dense and ill-conditioned) linear system can be rather time-consuming. In many practical applications the data values f_j will contain a certain amount of inaccuracy such as experimental noise or computational error (e.g., if the data come from some other previous computation). This fact demonstrates that we may not want to invest too much effort into obtaining a solution that exactly interpolates the data. Usually one attempts to overcome these difficulties by giving up some of the exactness of the solution function. For more details on scattered data approximation by positive definite functions we refer the reader to either [1] or [7].

§2. Interpolation By Iterated AMLS Approximation

In the standard approach to *moving least squares* (MLS) approximation one takes a local (usually polynomial) approximation space, and determines the point-wise approximation as the solution of a locally weighted norm minimization problem. This and many more details of MLS and AMLS approximation are described in [1]. The dimension of this polynomial space is usually far smaller than the data size. This polynomial approach can be equivalently interpreted as a quasi-interpolation approach, namely Backus-Gilbert's method. Similarly to the local polynomial approach, Backus-Gilbert's method avoids solving a full size linear system. However, its generating functions are point-wise determined, that is, unless a polynomial space of low order is used, evaluation of the Backus-Gilbert solution at each point still requires the solution of a linear system whose size is based on the dimension of the polynomial space. The advantage of the Backus-Gilbert formulation is that it suggests an extension to a continuous approximant whose evaluation no longer requires solution of any linear systems.

2.1. From the Backus-Gilbert Method to AMLS Approximation

Assume a quasi-interpolant $\mathcal{Q}_f : \mathbb{R}^s \to \mathbb{R}$ of the form

$$\mathcal{Q}_f(x) = \sum_{j=1}^{N} f_j \phi(x - x_j). \tag{6}$$

It is known that if ϕ is a cardinal function, then $\mathcal{Q}_f$ interpolates the given data and minimizes the point-wise error. In the Backus-Gilbert formulation of MLS approximation we do not intend to use a cardinal ϕ, but instead we determine the $\phi(x-x_j)$ at a fixed x via the following setup.

Assume

$$\phi(x - x_j) = w(x - x_j)q(x - x_j), \qquad q \in \Pi_d^s, \tag{7}$$

with a positive weight function $w : \mathbb{R}^s \to \mathbb{R}$ and polynomial q of degree at most d in s variables which satisfies the point-wise conditions

$$\sum_{j=1}^{N} (x - x_j)^\alpha \phi(x - x_j) = \delta_{\alpha,0}, \qquad 0 \leq |\alpha| \leq d. \tag{8}$$

It will become apparent later why we use the same notation ϕ for the MLS generating function as we used to define the interpolant $\mathcal{P}_f$ in (4). However, at this point the Backus-Gilbert method itself does not require ϕ to be strictly positive definite.

Condition (8) presents a set of *discrete moment conditions.* These moment conditions and the form (7) of the generating function can be derived from the Gram matrix associated with the standard MLS approximation. For $d > 1$, it is quite complicated to obtain an explicit analytic formulation for ϕ. However, if x_j are (coordinate-wise) uniformly distributed, then (8) may be approximately converted to an integral form via construction of a Riemann sum, i.e.,

$$\int_{\mathbb{R}^s} \tilde{x}^\alpha \phi(\tilde{x}) d\tilde{x} = \delta_{\alpha 0}, \qquad 0 \le |\alpha| \le d. \tag{9}$$

In analogy to (8), condition (9) now represents a set of *continuous moment conditions*. These conditions are the key ingredient of AMLS approximation — a method closely related to the general approximate approximation paradigm first suggested by Maz'ya in the early 1990s (see, e.g., [5] and references therein). The advantage of the continuous formulation (9) is that we can now explicitly construct the basic function ϕ. In fact, it is clear that any appropriately normalized ϕ will satisfy (9) for $d = 0$.

Condition (9) also implies a polynomial reproduction via function convolution, i.e.,

$$\int_{\mathbb{R}^s} \tilde{x}^\alpha \phi(x - \tilde{x}) d\tilde{x} = \int_{\mathbb{R}^s} (x - \tilde{x})^\alpha \phi(\tilde{x}) d\tilde{x} = x^\alpha, \qquad 0 \le |\alpha| \le d. \tag{10}$$

Consequently,

$$p(x) = \int_{\mathbb{R}^s} p(\tilde{x}) \phi(x - \tilde{x}) d\tilde{x}, \qquad p \in \Pi_d^s. \tag{11}$$

This reproduction of degree-d polynomials is what ensures good approximation properties of the scheme. Again, in the case of a uniform center distribution, the polynomial reproduction (11) can be approximately converted to a discrete form, i.e.,

$$p(x) \approx \sum_{j=1}^{N} p(x_j) \phi(x - x_j) \qquad p \in \Pi_d^s. \tag{12}$$

This interpretation leads us to believe that the (discrete) AMLS approximation method should also contain properties of function convolution. In particular, convolution is known to be a smoothing operation when it does not exactly reconstruct the convoluted function. Consequently, we expect $\mathcal{Q}_f$ to carry some high level smoothing effects.

2.2. Iterated AMLS Approximation

We now choose a ϕ that not only meets the continuous moment conditions but also is strictly positive definite. With such a ϕ, the quasi-interpolant

$\mathcal{Q}_f$ may in fact be pushed as close to the interpolant $\mathcal{P}_f$ as possible via an iterative process initialized with $\mathcal{Q}_f$. Details of this approach are presented in [2]. A basic algorithm is summarized below:

$$\mathcal{Q}_f^{(0)}(x) = \sum_{j=1}^{N} f_j \phi(x - x_j), \tag{13}$$

$$\mathcal{Q}_f^{(n+1)}(x) = \mathcal{Q}_f^{(n)}(x) + \sum_{j=1}^{N} \left[f_j - \mathcal{Q}_f^{(n)}(x_j) \right] \phi(x - x_j), \tag{14}$$

with residual functions

$$\begin{aligned} \mathcal{R}_f^{(n)}(x) &= \mathcal{Q}_f^{(n+1)}(x) - \mathcal{Q}_f^{(n)}(x) \\ &= \sum_{j=1}^{N} \left[f_j - \mathcal{Q}_f^{(n)}(x_j) \right] \phi(x - x_j). \end{aligned} \tag{15}$$

According to the results presented in [2], if the matrix with entries $\Phi_{ij} = \phi(x_i - x_j)$ satisfies $\|\Phi\|_2 < 1$, then $\mathcal{Q}_f^{(n)} \to \mathcal{P}_f$, i.e., $\mathcal{R}_f^{(n)}(x) \to 0$ as $n \to \infty$.

This iterative process will eventually reach the exact interpolant simply by reducing the residuals at the data points during each iteration. Note that the residuals themselves are also quasi-interpolants of the same construction as the initial $\mathcal{Q}_f^{(0)}$ is.

We now remind the reader of the data site distribution issue. AMLS approximation in general is best formulated for a uniform data distribution. However, since the iterative $\mathcal{Q}_f^{(n)}$ is guaranteed to arrive at the interpolant $\mathcal{P}_f$ it is not necessary to ask for evenly spaced data in the construction of $\mathcal{Q}_f^{(0)}$ or the residual function $\mathcal{R}_f^{(n)}$ in each iteration.

The initial AMLS approximant will usually be smooth but rather far away from the exact true function. As the iteration proceeds, the residuals decrease, but noise is also being picked up (if there is any). Thus, one might expect that during the iteration there is a moment when the approximant is relatively ideal meaning it is smooth and also close to the true function. This optimal moment is known to be problem dependent and also ϕ-dependent. We will come back to this discussion in detail later. To end this section, let us make sure that there are some basic functions ϕ available for this iterative AMLS method. For example,

- any normalized strictly positive definite function (for $d = 0$) will do, e.g.,
  - inverse multiquadrics $\phi(x) = (1 + x^2)^{-\beta}$, $\beta > 0$ for all s,
  - Wendland's compactly supported functions for specific s,
- normalized Laguerre-Gaussians for arbitrary d and s (see [2]), e.g.,

- $\phi(x) = e^{-x^2}$ for $d = 0$ and all s,
- $\phi(x) = (\frac{3}{2} - x^2)e^{-x^2}$ for $d = 1$ and $s = 1$.

§3. Ridge Regression for Noise Filtering

A well known approach to dealing with noisy data is ridge regression or smoothing spline approximation (see, e.g., the seminal paper [4] by Kimeldorf and Wahba). The basic idea is to give up some accuracy and then obtain a solution via a minimization process that balances a tradeoff between smoothness and closeness to the data. If we again assume the solution $\mathcal{P}_f$ to be of the form (4) (but not subject to constraint (1)), then using a smoothing parameter $\gamma > 0$, the coefficient vector $\mathbf{c}$ is determined by

$$\min \left\{ \mathbf{c}^T \Phi \mathbf{c} + \gamma \sum_{j=1}^{N} \left(\mathcal{P}_f(x_j) - f_j\right)^2 \right\}. \tag{16}$$

Here the quadratic form $\mathbf{c}^T\Phi\mathbf{c}$ is an indicator of the smoothness of $\mathcal{P}_f$ and clearly, $\sum_{j=1}^N \left(\mathcal{P}_f(x_j) - f_j\right)^2$ measures the closeness to the data.

It can be shown that solving (16) leads to the linear system

$$\left(\Phi + \frac{1}{\gamma}\mathbf{I}\right)\mathbf{c} = \mathbf{f}. \tag{17}$$

Note that this is just a regularized version of the interpolation system (5). It is clear that (17) will have a unique solution provided ϕ is strictly positive definite, since then the matrix $\left(\Phi + \frac{1}{\gamma}\mathbf{I}\right)$ is also positive definite for $\gamma > 0$. The function $\mathcal{P}_f$ found by solving (17) will no longer exactly fit the given data simply because of the use of γ. The parameter γ balances the tradeoff between closeness of fit and smoothness. Obviously, a large γ makes $\mathcal{P}_f$ fit the data more closely. In the limiting case, $\left(\Phi + \frac{1}{\gamma}\mathbf{I}\right) \to \Phi$ as $\gamma \to \infty$. This recovers the standard interpolation formulation as given by (5). We will use cross-validation to get a good value of the smoothing parameter. A different approach was suggested in the recent paper [8].

§4. Leave-One-Out Cross-Validation

As mentioned earlier, the optimal number of iterations for iterated AMLS approximation and the optimal smoothing parameter γ for ridge regression are ϕ-dependent. For a given problem, in addition to the optimal smoothing parameter γ used in ridge regression and the optimal number of iterations for iterated AMLS approximation, we seek to find an optimal shape parameter ε for the basic function ϕ in both methods. Indeed, an optimal ε is also required for standard interpolation. In order to find these optimal values we use a variation of the so-called *leave-one-out cross-validation* (LOOCV) algorithm. We now describe LOOCV along with its implementation.

4.1. LOOCV for Ridge Regression

For $k = 1, \ldots, N$, denote $\mathbf{x}^{[k]} = [x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_N]^T$, $\mathbf{f}^{[k]} = [f_1, \ldots, f_{k-1}, f_{k+1}, \ldots, f_N]^T$, and correspondingly, $\Phi^{[k]}_{ij} = \phi(\varepsilon(\mathbf{x}^{[k]}_i - \mathbf{x}^{[k]}_j))$, $\mathbf{c}^{[k]} = [c^{[k]}_1, \ldots, c^{[k]}_{N-1}]^T$. We do this to indicate that in forming the approximation we do not take the whole data set into consideration, but "leave out" the k^{th} element. Hence, all involved quantities are now of size $(N-1)$ instead of N. We implement the ridge regression method to find $\mathbf{c}^{[k]}$. That is,

$$\mathbf{c}^{[k]} = \left(\Phi^{[k]} + \frac{1}{\gamma}\mathbf{I}\right)^{-1} \mathbf{f}^{[k]}. \tag{18}$$

Correspondingly, the ridge regression approximation for leaving out the k^{th} data value becomes

$$\mathcal{P}^{[k]}_f(x) = \sum_{j=1}^{N-1} c^{[k]}_j \phi(\varepsilon(x - \mathbf{x}^{[k]}_j)). \tag{19}$$

In order to obtain a cost function for the optimization of γ (and ε) we compute the residual on the left-out data point x_k, i.e.,

$$e_k = \left| f_k - \mathcal{P}^{[k]}_f(x_k) \right|, \tag{20}$$

and denote $\mathbf{e} = [e_1, \ldots, e_N]^T$.

Judging from the formulation presented thus far, this optimization demands a large amount of computation — especially when the set of γ candidates and the set of ε candidates are large. We adapt a formula given by Rippa for standard RBF interpolation that will significantly reduce the computation (see [6]). We denote, $\tilde{\Phi} = \Phi + \frac{1}{\gamma}\mathbf{I}$, where Φ is the interpolation matrix based on the whole data set. Following Rippa's analysis in [6] one can show that e_k given by (20) can actually be computed by

$$e_k = \frac{\left(\tilde{\Phi}^{-1}\mathbf{f}\right)_k}{\tilde{\Phi}^{-1}_{kk}}. \tag{21}$$

For each pair of γ and ε, (20) requires computation of a matrix inverse and function evaluation for every different k, whereas with (21) we need to compute and store the inverse of the full size matrix only once for all $k = 1, \ldots, N$.

4.2. LOOCV for Iterated AMLS Approximation

As mentioned earlier, the iterated AMLS approximant carries a strong smoothing property which we will use for noise filtering purposes. Again, we employ a variant of LOOCV to find an optimal shape parameter ε and an optimal number of iterations. An interpretation of the iterated AMLS method is that the iterative approximant $\mathcal{Q}_f^{(n)}$ approximates the interpolant $\mathcal{P}_f$ by representing the inverse of the interpolation matrix Φ by a truncated Neumann series (see [2]), i.e.,

$$\Phi^{-1} \approx \sum_{i=0}^{n} (\mathbf{I} - \Phi)^i . \tag{22}$$

To optimize $\mathcal{Q}_f^{(n)}$ with respect to n and ε using LOOCV, we again define a cost function based on Rippa's formula (21) which was originally derived for the interpolation process. This leads to the following procedure.

- For a fixed ε formulate a cost function:
  - During each iteration, apply an approximate version of Rippa's formula based on (22), i.e.,

    $$e_k^{(n)} = \frac{\left(\sum_{i=0}^n (\mathbf{I} - \Phi)^i \mathbf{f}\right)_k}{\left(\sum_{i=0}^n (\mathbf{I} - \Phi)^i\right)_{kk}}. \tag{23}$$

    and denote $\mathbf{e}^{(n)} = [e_1^{(n)}, \ldots, e_N^{(n)}]^T$.
  - If $\|\mathbf{e}^{(n)}\| - \|\mathbf{e}^{(n-1)}\|$ falls below a specified tolerance or the maximal iteration number is reached, stop the iteration and store $\mathbf{e}(\varepsilon) = \mathbf{e}^{(n)}$.
- An optimal ε is obtained by minimizing $\|\mathbf{e}(\varepsilon)\|$ with respect to ε.

Since the matrix $\mathbf{I} - \Phi$ is non-singular (see [2]) we can perform an eigen decomposition, i.e., $\mathbf{I} - \Phi = \mathbf{X}\Lambda\mathbf{X}^{-1}$, so that for $n \ll N$ the cost of updating (23) during iterations can be reduced to $\mathcal{O}(N^2)$ from $\mathcal{O}(N^3)$ as would be required by usual direct matrix multiplication.

§5. Discussion of Numerical Results

In our numerical experiments we use Gaussian basic functions. All computations are performed on the unit square $[0,1]^2$. The scaling of the basic functions is achieved by using two parameters: the shape parameter ε defines the general shape, and a second parameter h that reflects the data spacing is used to mimic stationary approximation. Thus, we end up using expansions such as

$$\mathcal{P}_f(x) = \sum_{j=1}^{N} c_j e^{-\frac{\varepsilon^2}{h^2}\|x-x_j\|^2} \quad \text{or} \quad \mathcal{Q}_f(x) = \sum_{j=1}^{N} f_j e^{-\frac{\varepsilon^2}{h^2}\|x-x_j\|^2}.$$

Test data is taken from a modified Franke function. In Figure 1 we compare the convergence behavior of Gaussian RBF interpolation to that of the iterated AMLS algorithm for a problem without noise. The interpolation error is represented by the highly oscillatory curve caused by the associated numerical instabilities. The other two curves represent the errors for iterated AMLS approximation. The dash-dotted curve is for the initial iterate $\mathcal{Q}_f^{(0)}$ and the solid one corresponds to $\mathcal{Q}_f^{(10)}$. When ε is small the iterated AMLS method successfully overcomes the numerical difficulties associated with the interpolant since the residual iteration acts as a smoothing operation. The graph on the right is a zoom-in of the left graph in a small ε range.

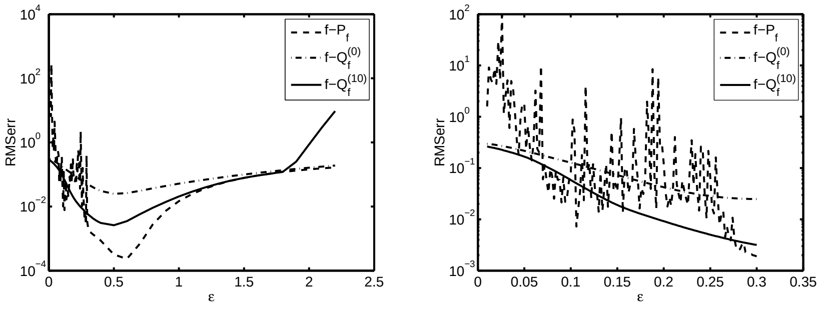

Fig. 1. RMS-errors vs. ε and illustration of instabilities. 33×33 Halton points.

Results based on data with 3% noise added are summarized in Table 1. In addition to the two methods discussed in this paper we include results for an iterated Shepard approximation. Clearly, simple RBF interpolation (with a fixed ε) is not recommended for noisy data. All three methods produce similar error drops. However, searching for optimal parameters for ridge regression is considerably more time consuming than for the two iterative methods which both do not require solutions of linear systems during the iterations. We note that for the two iterative methods the initial approximant based on an optimal ε is flat and does not approximate the data very well. However, convergence is rather fast during early iterations, and then slows down once the approximant starts feeding on the noise. We hope to detect this point at which the noise is optimally filtered with the LOOCV algorithm.

In summary, both iterative methods will produce reliably smooth results in significantly less time than the ridge regression method, and therefore perform quite well in this noise filtering application.

$N =$		9	25	81	289	1089
AMLS	RMSerr	4.80e-3	1.53e-3	6.42e-4	4.39e-4	2.48e-4
	ε	1.479865	1.268158	0.911530	0.652600	0.468866
	no. iter.	7	6	6	4	3
	time	0.2	0.4	1.0	5.7	254
Shepard	RMSerr	5.65e-3	1.96e-3	8.21e-4	5.10e-4	2.73e-4
	ε	2.194212	1.338775	0.895188	0.656272	0.468866
	no. iter.	7	7	7	5	3
	time	0.2	0.4	2.1	7.0	225
Ridge	RMSerr	3.54e-3	1.62e-3	7.20e-4	4.57e-4	2.50e-4
	ε	2.083918	0.930143	0.704802	0.382683	0.181895
	γ	100.0	100.0	47.324909	26.614484	29.753487
	time	0.3	1.2	1.1	21.3	672

Tab. 1. Comparison of different methods using Gaussians for noisy data sampled at Halton points.

References

1. Fasshauer, G. E., *Meshfree Approximation Methods with MATLAB*, World Scientific Publishers, Singapore, to appear.
2. Fasshauer, G. E., and J. G. Zhang, Iterated approximate moving least squares approximation, Proceedings of Meshless Methods (Lisbon 2005), Springer, in press.
3. Fornberg, B., and J. Zuev, The Runge phenomenon and spatially variable shape parameters in RBF interpolation, preprint, 2006.
4. Kimeldorf, G., and G. Wahba, Some results on Tchebycheffian spline functions, J. Math. Anal. Applic. **33** (1971), 82–95.
5. Maz'ya, V., and G. Schmidt, On quasi-interpolation with non-uniformly distributed centers on domains and manifolds, J. Approx. Theory **110** (2001), 125–145.
6. Rippa, S., An algorithm for selecting a good value for the parameter c in radial basis function interpolation, Adv. Comp. Math **11** (1999), 193–210.
7. Wendland, H., *Scattered Data Approximation*, Cambridge University Press, Cambridge, 2004.
8. Wendland, H., and C. Rieger, Approximate interpolation with applications to selecting smoothing parameters, Numer. Math. **101**, 29–748.

Gregory E. Fasshauer and Jack G. Zhang
Illinois Institute of Technology
Chicago, IL
`fasshauer@iit.edu   http://www.math.iit.edu/∼fass`
`zhangou@iit.edu   http://www.iit.edu/∼zhangou`