Stably Computing with Gaussian RBFs

Michael McCourt
Center for Applied Mathematics
Cornell University

International Conference on Scientific Computing and Applications
University of Nevada at Las Vegas
April 2, 2012

Joint work with Greg Fasshauer
Department of Applied Mathematics, Illinois Institute of Technology

mccomic@mcs.anl.gov (Cornell)
RBFs in applications: Why aren’t there more?

For many problems, Kernel-based approximations are appropriate; sometimes they would be optimal. Why are they not more prevalent?
RBFs in applications: Why aren’t there more?

For many problems, Kernel-based approximations are appropriate; sometimes they would be optimal. Why are they not more prevalent?

Possible barriers to widespread RBF usage

1. Mathematical complexity
2. Stability concerns
3. Computational cost
4. Presence of one or more free parameters
Fundamental Application (Scattered Data Fitting)

Given data \((x_j, y_j), j = 1, \ldots, N\), with \(x_j \in \mathbb{R}^d, y_j \in \mathbb{R}\), find a (continuous) function \(s_f\) such that \(s_f(x_j) = y_j, j = 1, \ldots, N\).

Consider here multivariate kernel-based interpolation using a data-dependent linear function space

\[
s_f(x) = \sum_{j=1}^{N} c_j K(x, x_j), \quad x \in \Omega \subseteq \mathbb{R}^d
\]

with \(K : \Omega \times \Omega \rightarrow \mathbb{R}\) a positive definite kernel.

To find \(c_j\) solve the interpolation equations

\[
s_f(x_i) = f(x_i) = y_i, \quad i = 1, \ldots, N
\]

which gives a linear system \(Kc = y\); \(K\) is symmetric positive definite.
Gaussian Kernel Interpolation

One (common?) choice of kernel is the Gaussian

$$K(x, z) = \exp\left(-\varepsilon^2 \|x - z\|^2\right),$$

where the $\varepsilon$ value determines the locality of the basis functions.
Gaussian Kernel Interpolation

One (common?) choice of kernel is the Gaussian

\[ K(x, z) = \exp \left( -\varepsilon^2 \|x - z\|^2 \right), \]

where the \( \varepsilon \) value determines the locality of the basis functions.

Opportunity

Gaussians are great choices for interpolating very smooth functions.

Problem

For some (potentially very accurate) choices of \( \varepsilon \), the interpolation matrix \( K \) may be very ill-conditioned.
As $\varepsilon \to 0$ the basis functions become flatter, which produces a matrix which looks increasingly like a matrix of all ones (a low rank matrix). 

\[
\begin{pmatrix}
1 & \cdots & 1 \\
\vdots & \ddots & \vdots \\
1 & \cdots & 1
\end{pmatrix}
\]
Better Gaussian Interpolation Results

This is what we want to see ... but how can we make it happen?
Eigenfunction (Mercer) Series Using Hilbert-Schmidt Theory

\[ e^{-\varepsilon^2(x-z)^2} = \sum_{n=0}^{\infty} \lambda_n \varphi_n(x) \varphi_n(z) \]

where

\[ \lambda_n = \sqrt{\alpha^2 + \delta^2 + \varepsilon^2 \left( \alpha^2 + \delta^2 + \varepsilon^2 \right)^n}, \quad \varphi_n(x) = \gamma_n e^{-\delta^2 x^2} H_n(\alpha \beta x) \]

with \( H_n \) Hermite polynomials, \( \beta = (1 + (2\varepsilon/\alpha)^2)^{1/4} \), \( \gamma_n = \sqrt{\beta^2 n \Gamma(n+1)} \), \( \delta^2 = \alpha^2 (\beta^2 - 1) \)

and \( \{\varphi_n\}_{n=0}^{\infty} \) (weighted) \( L_2 \)-orthonormal, i.e.,

\[ \int_{-\infty}^{\infty} \varphi_m(x) \varphi_n(x) \alpha \sqrt{\pi e^{-\alpha^2 x^2}} dx = \delta_{mn} \]
Eigenfunction (Mercer) Series Using Hilbert-Schmidt Theory

\[ e^{-\varepsilon^2(x-z)^2} = \sum_{n=0}^{\infty} \lambda_n \varphi_n(x) \varphi_n(z) \]

where

\[ \lambda_n = \sqrt{\frac{\alpha^2}{\alpha^2 + \delta^2 + \varepsilon^2}} \left( \frac{\varepsilon^2}{\alpha^2 + \delta^2 + \varepsilon^2} \right)^n, \quad \varphi_n(x) = \gamma_n e^{-\delta^2 x^2} H_n(\alpha \beta x) \]

with \( H_n \) Hermite polynomials,

\[ \beta = \left( 1 + \left( \frac{2\varepsilon}{\alpha} \right)^2 \right)^{\frac{1}{4}}, \quad \gamma_n = \sqrt{\frac{\beta}{2^n \Gamma(n+1)}}, \quad \delta^2 = \frac{\alpha^2}{2} \left( \beta^2 - 1 \right) \]

and \( \{\varphi_n\}_{n=0}^{\infty} \) (weighted) \( L_2 \)-orthonormal, i.e.,

\[ \int_{-\infty}^{\infty} \varphi_m(x) \varphi_n(x) \frac{\alpha}{\sqrt{\pi}} e^{-\alpha^2 x^2} \, dx = \delta_{mn} \]
Properties of Eigenfunction Approach

Analytic properties
- Orthogonality (for appropriate parameter ranges)
- Best least squares approximation
- Recovers polynomials in the $\varepsilon \to 0$ "flat limit"
- Multiple dimensions via tensor product

Computational properties
- Stable basis, rather than standard basis
- Recurrence relation drawn from Hermite recurrence
The RBF-QR Algorithm

The eigenfunction structure is

\[ K(x, z) = \sum_{m=0}^{\infty} \lambda_m \varphi_m(x) \varphi_m(z) \]

But we can’t compute with an infinite matrix, so we choose a truncation value \( M > N \) (aided by \( \lambda_m \to 0 \) as \( m \to \infty \)) and write

\[
K = \begin{pmatrix}
K(x_1, x_1) & \ldots & K(x_1, x_N) \\
\vdots & & \vdots \\
K(x_N, x_1) & \ldots & K(x_N, x_N)
\end{pmatrix}
\]

\[
= \begin{pmatrix}
\varphi_1(x_1) & \ldots & \varphi_M(x_1) \\
\vdots & & \vdots \\
\varphi_1(x_N) & \ldots & \varphi_M(x_N)
\end{pmatrix}
\begin{pmatrix}
\lambda_1 \\
\vdots \\
\lambda_M
\end{pmatrix}
\begin{pmatrix}
\varphi_1(x_1) & \ldots & \varphi_1(x_N) \\
\vdots & & \vdots \\
\varphi_M(x_1) & \ldots & \varphi_M(x_N)
\end{pmatrix}
\]

\[
= \Phi \Lambda \Phi^T
\]
Find invertible $X$ such that $\Psi = KX^{-1}$ is better conditioned than $K$
Find invertible $X$ such that $\Psi = KX^{-1}$ is better conditioned than $K$.

For $M > N$, $\Phi$ is “short and fat”, so its QR decomposition is

$$
\begin{pmatrix}
\varphi_1(x_1) & \ldots & \varphi_N(x_1) & \varphi_{N+1}(x_1) & \ldots & \varphi_M(x_1) \\
\vdots & & \vdots & & & \vdots \\
\varphi_1(x_N) & \ldots & \varphi_N(x_N) & \varphi_{N+1}(x_N) & \ldots & \varphi_M(x_N)
\end{pmatrix}
= Q
\begin{pmatrix}
R_1 & R_2 \\
N \times N & N \times (M-N)
\end{pmatrix}
$$
Find invertible $X$ such that $\Psi = KX^{-1}$ is better conditioned than $K$

For $M > N$, $\Phi$ is “short and fat”, so its QR decomposition is

\[
\begin{pmatrix}
\varphi_1(x_1) & \cdots & \varphi_N(x_1) & \varphi_{N+1}(x_1) & \cdots & \varphi_M(x_1) \\
\vdots & & \vdots & & \vdots & \\
\varphi_1(x_N) & \cdots & \varphi_N(x_N) & \varphi_{N+1}(x_N) & \cdots & \varphi_M(x_N)
\end{pmatrix}
= Q
\begin{pmatrix}
R_1 \\
R_2
\end{pmatrix}
\]

Substituting this back in for $\Phi^T$ we get

\[
K = \Phi \Lambda \Phi^T = \Phi \begin{pmatrix} \Lambda_1 \\ \Lambda_2 \end{pmatrix} \begin{pmatrix} R_1^T \\ R_2^T \end{pmatrix} Q^T = \Phi \begin{pmatrix} \Lambda_1 R_1^T \\ \Lambda_2 R_2^T \end{pmatrix} Q^T
\]

\[
= \Phi \begin{pmatrix} I_N \\ \Lambda_2 R_2^T R_1^{-T} \Lambda_1^{-1} \end{pmatrix} \begin{pmatrix} \Lambda_1 R_1^T \\ Q^T \end{pmatrix}
= \Psi
= X
\]

Stable Gaussians
April 2, 2012 11 / 27
Find invertible $X$ such that $\Psi = KX^{-1}$ is better conditioned than $K$

For $M > N$, $\Phi$ is “short and fat”, so its QR decomposition is

$$
\begin{pmatrix}
\varphi_1(x_1) & \cdots & \varphi_N(x_1) & \varphi_{N+1}(x_1) & \cdots & \varphi_M(x_1) \\
\vdots & & \vdots & \vdots & & \vdots \\
\varphi_1(x_N) & \cdots & \varphi_N(x_N) & \varphi_{N+1}(x_N) & \cdots & \varphi_M(x_N)
\end{pmatrix}
= Q
\begin{pmatrix}
R_1 \\
N \times N \\
R_2 \\
N \times (M-N)
\end{pmatrix}
$$

Substituting this back in for $\Phi^T$ we get

$$
K = \Phi \Lambda \Phi^T = \Phi \begin{pmatrix}
\Lambda_1 \\
\Lambda_2
\end{pmatrix}
\begin{pmatrix}
R_1^T \\
R_2^T
\end{pmatrix}Q^T = \Phi \begin{pmatrix}
\Lambda_1 R_1^T \\
\Lambda_2 R_2^T
\end{pmatrix}Q^T
$$

$$
= \Phi \begin{pmatrix}
I_N \\
\Lambda_2 R_2^T R_1^{-T} \Lambda_1^{-1}
\end{pmatrix} \underbrace{\Lambda_1 R_1^T Q^T}_{=\Psi}
$$

New basis

$$
\Psi = (\Phi_1 \Phi_2)
\begin{pmatrix}
I_N \\
\Lambda_2 R_2^T R_1^{-T} \Lambda_1^{-1}
\end{pmatrix}
= \underbrace{\Phi_1}_{\text{first } N \text{ eigenfunctions}} + \underbrace{\Phi_2}
\begin{pmatrix}
\Lambda_2 R_2^T R_1^{-T} \Lambda_1^{-1}
\end{pmatrix}
$$
Remark

- The structure of the correction term $\Phi_2 \left[ \Lambda_2 R_2^T R_1^{-T} \Lambda_1^{-1} \right]$ is important.
  - Since $\lambda_n \to 0$ as $n \to \infty$, the eigenvalues in $\Lambda_2$ are smaller than those in $\Lambda_1$ and so $R_2^T R_1^{-T}$ is not blown up.

- The multiplications by $\Lambda_2$ and $\Lambda_1^{-1}$ can be done analytically.
  - This avoids stability issues associated with possible underflow (entries in $\Lambda_2$ are as small as $\varepsilon^{2M-2}$) or overflow (entries in $\Lambda_1^{-1}$ are as large as $\varepsilon^{-2N-2}$).

- This QR approach should be applicable to any positive definite kernels whose eigenfunction expansion is known.
  - Spherical (or zonal) basis functions (Fornberg et al).
  - Recent progress for Matérn functions.
Stable Interpolation Results

\[ f(x) = \sin(x/2) - 2\cos(x) + 4\sin(\pi x), \quad x \in [-4, 4] \]

Things to note:
- Structure exists which can’t be resolved with RBF-Direct
- RBF-QR can get the same accuracy as RBF-Direct for fewer points

mccomic@mcs.anl.gov (Cornell)

April 2, 2012 13 / 27
Problems with RBF-QR

There is significant cost involved in computing

$$\psi = \phi_1 + \phi_2 \left[ \Lambda_2 R_2^T R_1^{-T} \Lambda_1^{-1} \right]$$

both in size of the problem (since $M > N$) and in time required for the QR factorization.

Question

What can we do about this?
Approximation vs. Interpolation

Answer

Choose a truncation point $M < N$ and conduct a regression rather than interpolation.

Because the eigenfunction expansion is the $M$-term optimal approximation to the Gaussian (in the $L_2(\mathbb{R}^d, \rho)$ space) we can expect good results if the eigenvalues decay quickly enough.

$$\lambda_n = \sqrt{\frac{\alpha^2}{\alpha^2 + \delta^2 + \epsilon^2}} \left( \frac{\epsilon^2}{\alpha^2 + \delta^2 + \epsilon^2} \right)^n$$

$$\lambda_n \to \epsilon^2, \quad \text{as } \epsilon \to 0.$$
Approximation vs. Interpolation

Answer

Choose a truncation point $M < N$ and conduct a regression rather than interpolation.

Because the eigenfunction expansion is the $M$-term optimal approximation to the Gaussian (in the $L_2(\mathbb{R}^d, \rho)$ space) we can expect good results if the eigenvalues decay quickly enough.

$$\lambda_n = \sqrt{\frac{\alpha^2}{\alpha^2 + \delta^2 + \varepsilon^2}} \left( \frac{\varepsilon^2}{\alpha^2 + \delta^2 + \varepsilon^2} \right)^n$$

$$\lambda_n \to \varepsilon^2, \text{ as } \varepsilon \to 0.$$ 

Basically all we need is $\varepsilon$ small, which is the main motivation for this research.
The RBF-QRr Algorithm

The original interpolation system was

$$Kb = y$$

Choosing $M < N$ produces the low rank system

$$K \approx \Phi \Lambda \Phi^T,$$

which when plugging this into the system gives

$$\Phi \Lambda \Phi^T b = y \quad \Phi c = y$$

where $c = \Lambda \Phi^T b$ is an $M$-vector.
The RBF-QRr Algorithm

The original interpolation system was

\[ K \mathbf{b} = \mathbf{y} \]

Choosing \( M < N \) produces the low rank system

\[ K \approx \Phi \Lambda \Phi^T, \]

which when plugging this into the system gives

\[ \Phi \Lambda \Phi^T \mathbf{b} = \mathbf{y} \]
\[ \Phi \mathbf{c} = \mathbf{y} \]

where \( \mathbf{c} = \Lambda \Phi^T \mathbf{b} \) is an \( M \)-vector.

- This problem is now a *regression* problem rather than an interpolation problem
- RBF-QRr will only reproduce the “true” RBF interpolant for small \( \varepsilon \)
Numerical Results in 2D

\[ f(x, y) = \frac{1}{1 + x^2 + y^2}, \quad M = 0.7N \]
Numerical Results in 5D

\[ f(x) = \cos \left( \frac{1}{5} \sum_{k=1}^{5} x_k \right), \quad M = 0.6N \]
Applications: Meshfree Multiphysics Coupling

Two models, coupled through an interface. Each model has three regions: interior, coupling and interface. Interior values can be handled internally by the model, but values in the coupling region and interface require knowledge of the other model as well.

\[
\begin{pmatrix}
F_1(x) \\
F_2(x) \\
F_C(x)
\end{pmatrix} = 0
\]
Applications: Meshfree Multiphysics Coupling

\[
\begin{pmatrix}
J_1(F_1) & J_1^C(F_1) & J_1^C(F_1) & J_1^C(F_1) \\
J_2(F_2) & J_2^C(F_2) & J_2^C(F_2) & J_2^C(F_2) \\
J_1(F_1^C) & J_1^C(F_1^C) & J_1^C(F_1^C) & J_1^C(F_1^C) \\
J_2(F_2^C) & J_2^C(F_2^C) & J_2^C(F_2^C) & J_2^C(F_2^C)
\end{pmatrix}
\begin{pmatrix}
\Phi_{D_2\Phi_1}^{-1} \\
\Phi_{D_1\Phi_2}^{-1} \\
\Phi_{I_2\Phi_1}^{-1} \\
\Phi_{I_1\Phi_2}^{-1}
\end{pmatrix}
\begin{pmatrix}
u_1 \\
u_2 \\
u_1^C \\
u_2^C
\end{pmatrix}
\begin{pmatrix}
u_1 \\
u_2 \\
u_1^I \\
u_2^I
\end{pmatrix}
\]

RBF-QR allows for fewer coupling points while maintaining a high order coupled simulation.
Applications: Solving BVP with RBF-QRr

Applying nonsymmetric (Kansa) collocation with RBF-QRr is much the same as it is with direct RBF methods: given a problem

\[ L(u) = f \]
\[ B(u) = g \]

make the assumption that \( u \) is a linear combination of \( N \) Gaussians. In turn, use the approximation that each Gaussian is a linear combination of \( M \) eigenfunctions and the system becomes

\[
\begin{pmatrix}
\Phi_L \\
\Phi_B
\end{pmatrix}
\begin{pmatrix}
c
\end{pmatrix}
=
\begin{pmatrix}
f \\
g
\end{pmatrix}.
\]
Applications: Solving BVP with RBF-QR

One point of comparison for RBF-QR, would be the method of fundamental solutions:

Laplacian on square domain, with Dirichlet BC

RMS error vs. collocation points
Applications: Solving BVP with RBF-QR

One point of comparison for RBF-QR, would be the method of fundamental solutions:

Here, obviously, MFS puts GaussQR to shame. :(
Applications: Solving BVP with RBF-QRr

Maybe if we choose a more complicated domain, Gaussians will stand a chance?

Laplacian on L-shaped domain, with Dirichlet BC

RMS error

collocation points
Applications: Solving BVP with RBF-QRr

Maybe if we choose a more complicated domain, Gaussians will stand a chance?

Only for very (unnecessarily) large values of $N$. 

Laplacian on L-shaped domain, with Dirichlet BC

RMS error

collocation points

10^{-16}

10^{-10}

10^{-5}

MFS

GaussQR
Applications: Solving BVP with RBF-QRr

Would a change in boundary conditions make GaussQR more competitive?

![Graph showing comparison between MFS and GaussQR methods.](chart.png)
Applications: Solving BVP with RBF-QR

Would a change in boundary conditions make GaussQR more competitive?

No longer embarrassed, but ...
Applications: Solving BVP with RBF-QR

How about choosing a harder problem?

Biharmonic on L-shaped domain, with mixed BC

- MFS
- GaussQR
Applications: Solving BVP with RBF-QR

How about choosing a harder problem?

Now there is finally an example where GaussQR is appropriate.
To Do List

- Analyze Gaussian interpolants in $\mathbb{R}^d$. ✓
- Find applications where stable Gaussians are useful. ✓
- Begin solving boundary value problems. ✓
- Develop a fast solver based on recurrence relations. ✓
To Do List

- Analyze Gaussian interpolants in $\mathbb{R}^d$. ✓
- Find applications where stable Gaussians are useful. ✓
- Begin solving boundary value problems. ✓
- Develop a fast solver based on recurrence relations. ✓
- Determining eigenfunction expansions for kernels other than Gaussians.
- Producing a fast solver based on minres-QLP, using fast summation.
- Using statistics (maximum likelihood estimation, inference, cross-validation) to determine $\varepsilon$, $\alpha$, $M$.
- Considering anisotropic shape parameters, i.e., non radial kernels.

Matlab implementation available at

http://math.iit.edu/~mccomic/gaussqr
\( \varepsilon \) and \( \alpha \) relationship

\[
\lim_{\varepsilon \to 0} \varphi_k(x) = \gamma_k e^{-\varepsilon^2 x^2} H_{n-1}(\alpha x)
\]

\[
\lim_{\varepsilon \to \infty} \varphi_k(x) = \gamma_k e^{-\varepsilon \alpha x^2} H_{n-1}(\sqrt{2\varepsilon \alpha} x)
\]