Multi-Step Fitting of Large-Scale Emulators: Exploring the Interface between Design and Modeling

Peter Z. G. Qian
Joint work with Ben Haaland and Xu Xu

1 University of Wisconsin-Madison
2 National University of Singapore

Nov 9, 2011
Outline

• Latin hypercube designs
• Emulators for computer experiments
• Error decomposition
• Multi-step interpolator
• Numeric illustration
• Theoretical bounds
• Conclusions
Latin Hypercube Designs (Mckay et al., 1979)
Sampling Theory (Stein, 1987)

• Consider estimating $\mu = E[f(x)]$ for a computer model, where the distribution of $x$ is the uniform measure on $[0, 1)^q$. Let $\sigma^2 = \text{var}[f(x)]$.

• Compute $\hat{\mu}$ using a Latin hypercube design with $n$ runs. As $n \to \infty$,

$$\text{var}(\hat{\mu}) = n^{-1} \sigma^2 - n^{-1} \sum_{k=1}^{q} \int_0^1 [f_k(x_k)]^2 dx_k + o(n^{-1}).$$

where $f_{-k}(x_k) = \int \{f(x) - \mu\} dF_{-k}$, $dF_{-k} = \prod_{j \neq k} dF_j$, and $F_k$ is the uniform distribution $U[0, 1)$ in the $k$th dimension.
Emulators for Computer Experiments

- When using a computer experiment to study a real system, a thorough exploration of the unknown surface is typically wanted.

- Input/output pairs often too expensive for a complete exploration.

- A solution is to evaluate the computer experiment at several well-distributed data sites.

- Then, build an interpolator which can be used as a stand-in, or emulator, for the actual computer experiment.

- Thorough exploration can then be carried out on the emulator.
Gaussian Process Interpolators

- Gaussian process or Reproducing Kernel Hilbert Spaces (RKHS) interpolators are often used to build emulators for computer experiments.

- Building block is a symmetric, positive definite basis function \( \Phi: \Omega \times \Omega \rightarrow \mathbb{R} \).

- **Simple Form:** \( P(x) = \sum_{i=1}^{n} \alpha_i \Phi(x, x_i) \), with \( P(x_i) = f(x_i) \).

- Associated with each \( \Phi \) (and \( \Omega \)) is a Hilbert space of functions, \( \mathcal{N}_\Phi(\Omega) \), whose norm measures size and smoothness.
Numerical Problems in Interpolation

- Many systems which scientists, engineers, and medical researchers use computer experiments to study exhibit extremely complex behavior in portions of the input space.
- Understanding these regions requires many input sites which are potentially very close together.
- Most interpolation procedures suffer from increasingly severe numeric problems as the number of data sites becomes larger.
- The problem of finding an interpolator becomes ill-conditioned as data sites becomes too near to one another.
- Popular techniques for numerically stabilizing an interpolator: nugget effect, compactly supported kernels and correlation decomposition. These methods fit approximate models.
New View: Presence of Numeric and Nominal Error

- A computed quantity, $\tilde{x}$, which is subject to floating point rounding, is distinguished from the idealized quantity, $x$.

- Key idea:

$$\|f - \tilde{P}\| \leq \|f - P\| + \|P - \tilde{P}\|$$ (1)

- The first and second terms on the right-hand side are called the *nominal* and *numeric* portions of the error, respectively.

- The error separation (1) is true for any norm $\| \cdot \|$.

- Often, there is a trade-off between the nominal and numeric error.

Nominal and Numeric Trade-Off

- If \( f \in \mathcal{N}_\phi(\Omega) \) and \( X_1 \subseteq X_2 \), then
  \[
  \| f - P_2 \|_{\mathcal{N}_\phi(\Omega)} \leq \| f - P_1 \|_{\mathcal{N}_\phi(\Omega)},
  \]
  (2)

  where \( P_1 \) and \( P_2 \) denote the RKHS interpolators on the sets \( X_1 \) and \( X_2 \), respectively.

- The nominal error of an interpolator is always reduced (with respect to the native space norm) by the addition of data sites.

- On the other hand, the numeric error is potentially increased to an arbitrary degree by the addition of data sites.
Example 1

- Consider building an interpolator for the below Michaelewicz function on 925 points.
Point Set #1
Point Set #2
Point Set #3
Interpolator with Very Poor Accuracy

- The separation distance is $10^{-10}$.

- The crosses do not contribute much information about the unknown surface, but make the interpolation problem much less stable.

- The best possible mean squared error (over all widths of kernels) for traditional interpolation on the third point set is $\approx 0.15$, the $L_2$ norm of the function.
Multi-Step Interpolator (Floater and Iske 1996)

- Form well-spread nested subsets of the data.

- Interpolate the first subset with a wide and smooth kernel and form residuals on the next subset.

- The residuals are then interpolated using a narrower and less smooth kernel.

- The residual interpolator is added to the previous step interpolator to form the current stage interpolator.

- Residuals are successively formed and interpolated with narrower and less smooth kernels.

- The final interpolator is the sum of the interpolators formed at each stage.

- A general theory is developed in Haaland and Qian (2011).
Illustration of Multi-Step Algorithm

- Consider using the multi-step algorithm to interpolate the below function using data at the plotted sites.

**Figure**: True function and data sites
Step 1

- Interpolate at every other point using a wide kernel.

Figure: True function (solid) and first-stage interpolator (dashed)
Step 2

- Interpolate residuals with narrow kernel.

**Figure:**

- **Top Panel:** Residuals (solid) and residual interpolator (dashed)
- **Bottom Panel:** True function (solid) and multi-scale interpolator (dashed)
Multi-Step Interpolator: Details

• Here, $J$ denotes the number of steps, $\Phi_j$ denotes the kernel used in step $j$, and $X_1 \subset \cdots \subset X_J = X$. Initialize $\mathcal{P}^0 \equiv 0$.

• Then, for $j = 1, \ldots, J$, let

$$\mathcal{P}^j(x) = \sum_{u=1}^{n_j} \alpha^j_u \Phi_j(x - x_u),$$

$$\alpha^j = A_{X_j, \Phi_j}^{-1} (f - \sum_{k=0}^{j-1} \mathcal{P}^k)|_{X_j},$$

$$A_{X_j, \Phi_j} = \{ \Phi_j(x_u - x_v) \}, \ u, v = 1, \ldots, n_j,$$

$$n_j = \text{card } X_j.$$  

• Then, the multi-step interpolator is

$$\mathcal{P}(x) = \sum_{j=1}^{J} \mathcal{P}^j(x).$$
Data Collection

- Use *nested space-filling designs* such as nested Latin hypercube designs (Qian, 2009), nested lattice samples (Qian and Ai, 2010) and nested nets (Haaland and Qian, 2010).
- Such designs were originally proposed for multi-fidelity computer experiments.
- Optimal nested Latin hypercube designs maximize the separation distance of nested Latin hypercube designs.
- The thinning algorithm (Floater and Iske, 1996) can generate a nested sequence of subsets with nearly space-filling properties from any design set.
Nested Latin Hypercube Designs (Qian, 2009)
Optimal Nested Latin Hypercube Designs
Illustration of the Thinning Algorithm

X4 with 2000 points from U(0,1)

X3 with 1000 points

X2 with 500 points

X1 with 250 points
Example 2: Function

- Consider a two-dimensional function

\[
f(x, y) = 15 \exp\left\{ -\frac{1}{1 - (2x - 1)^2} - \frac{1}{1 - (2y - 1)^2} \right\} \times \\
\left[ 0.75 \exp\left\{ -\frac{(9x - 2)^2 + (9y - 2)^2}{4} \right\} + 0.75 \exp\left\{ -\frac{(9x + 1)^2}{49} + \frac{(9y - 2)^2}{10} \right\} + 0.5 \exp\left\{ -\frac{(9x - 7)^2 + (9y - 3)^2}{4} \right\} - 0.2 \exp\left\{ -(9x - 4)^2 - (9y - 7)^2 \right\} \right].
\]

- Compute the root mean square error (RMSE) for a testing sample on the 40 \times 40 grid.
### Example 2: Results

<table>
<thead>
<tr>
<th>Grids (# of steps)</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Multi-step</td>
</tr>
<tr>
<td>33 × 33 (5)</td>
<td>1.21e-04</td>
</tr>
<tr>
<td>65 × 65 (6)</td>
<td>4.03e-05</td>
</tr>
<tr>
<td>129 × 129 (7)</td>
<td>1.53e-06</td>
</tr>
<tr>
<td>257 × 257 (8)</td>
<td>5.72e-08</td>
</tr>
<tr>
<td>513 × 513 (9)</td>
<td>1.97e-08</td>
</tr>
<tr>
<td>1025 × 1025 (10)</td>
<td>6.88e-09</td>
</tr>
</tbody>
</table>
Example 3

• Consider

\[ f(x) = \sum_{j=1}^{10} x_j \left( c_j + \log \frac{x_j}{x_1 + \cdots + x_{10}} \right), \]

where \((c_1, \ldots, c_{10}) = (-6.089, -17.164, -34.054, -5.914, -24.721, -14.986, -24.100, -10.708, -26.662, -22.179)\).

• Generate training data from:
  • Nested Latin hypercube designs (Qian, 2009)
  • Optimal nested Latin hypercube designs by maximizing the separation distance of nested Latin hypercube designs
• Compute the RMSE of a Latin hypercube design of 1000 runs.

<table>
<thead>
<tr>
<th>Training size</th>
<th>Multi-step(# of steps)</th>
<th>Single-step with compact support</th>
</tr>
</thead>
<tbody>
<tr>
<td>576</td>
<td>0.53 (4)</td>
<td>0.87</td>
</tr>
<tr>
<td></td>
<td>0.36(4)</td>
<td>0.83</td>
</tr>
<tr>
<td>2304</td>
<td>0.2 (5)</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td>0.13 (5)</td>
<td>0.36</td>
</tr>
</tbody>
</table>
Numeric Accuracy Results

- The only complete and rigorous bound on the numeric error of the multi-step interpolator.
- The results make very weak and general assumptions.
- Approximations to the terms in the bound can be computed.
The matrix 2-norm $\| \cdot \|_2$ is defined as $\| A \|_2 = \sqrt{\lambda_{\text{max}}(A' A)}$.

Suppose $Ax = b$ and $\tilde{A}\tilde{x} = \tilde{b}$ with $\| A - \tilde{A} \|_2 \leq \delta_A \| A \|_2$, $\| b - \tilde{b} \|_2 \leq \delta_b \| b \|_2$, and $\kappa(A) = r/\delta_A < 1/\delta_A$ for some $\delta_A, \delta_b > 0$. Then, $\tilde{A}$ is non-singular,

$$\frac{\| \tilde{x} \|_2}{\| x \|_2} \leq \frac{1 + r(\delta_b/\delta_A)}{1 - r},$$

$$\frac{\| x - \tilde{x} \|_2}{\| x \|_2} \leq \frac{\delta_A + \delta_b}{1 - r} \kappa(A),$$

where $\kappa(A) = \| A \|_2 \| A^{-1} \|_2$. 

**(5)**
Numeric Accuracy of Single Step Interpolator

- Suppose that $\|A_X,\Phi - \tilde{A}_X,\Phi\|_2 \leq \delta_A\|A_X,\Phi\|_2$, $\|f|_X - \tilde{f}|_X\|_2 \leq \delta_f\|f|_X\|_2$, $\kappa(A_X,\Phi) = r/\delta_A < 1/\delta_A$, and
  
  $$\sup_{x,y \in \Omega} |\Phi(x, y) - \tilde{\Phi}(x, y)| < D\delta_A$$
  
  for some $\delta_A, \delta_f, D > 0$, then
  
  $$|P(x) - \tilde{P}(x)| \leq \|f|_X/\sqrt{n}\|_2\frac{(\delta_A + \delta_f)}{1 - r} g(X, \Phi),$$
  
  $$g(X, \Phi) = \frac{n}{\lambda_{\min}(A_X,\Phi)}(\kappa(A_X,\Phi)\Phi(0) + D),$$

  where $\kappa(\cdot)$ is defined in the previous lemma.

- To bound error of single step interpolator, take $\delta_A = \delta_f = 10^{-15}$, restrain $\kappa(A_X,\Phi)$. 

Numeric Accuracy of Multi-Step Interpolator

- Suppose that for \( j = 1, \ldots, J \), \( \| A_{X_j, \Phi_j} - \tilde{A}_{X_j, \Phi_j} \|_2 \leq \delta_j \| A_{X_j, \Phi_j} \|_2 \), 
  \( \| f |_{X_j} - \tilde{f} |_{X_j} \|_2 \leq \delta \| f |_{X_j} \|_2 \), \( \kappa(A_{X_j, \Phi_j}) \leq r/\delta_j < 1/\delta_j \), and 
  \( \sup_{x,y \in \Omega} | \Phi_j(x - y) - \tilde{\Phi}_j(x - y) | < D\delta \) for some \( \delta_j, \delta, D > 0 \) with 
  \( \delta_j \| (f - \sum_{k=1}^{j-1} P^k) |_{X_j/\sqrt{n_j}} \|_2 \leq \delta \| f |_{X_j/\sqrt{n_j}} \|_2 \), then 

\[
\left| \sum_{j=1}^{J} \mathcal{P}^j(x) - \sum_{j=1}^{J} \tilde{\mathcal{P}}^j(x) \right| \\
\leq \delta \| f |_{X_j/\sqrt{n_j}} \|_2 \left[ \sum_{M=1}^{J} C^M \sum_{i \in S_J(M)} \prod_{k=1}^{M} \rho(X_{i_k}, X_{i_{k+1}}) g(X_{i_k}, \Phi_{i_k}) \right],
\]

where \( C = 2/(1 - r) \), 
\( S_J(M) = \{ i \in \mathbb{N}^{M+1} : 1 \leq i_1 < \cdots < i_M \leq i_{M+1} = J \} \) and 
\( \rho(X, Y) = \| f |_{X/\sqrt{n_X}} \|_2 / \| f |_{Y/\sqrt{n_Y}} \|_2 \).
Bound in Terms of Separation Distance

• Let

\[ q_X = \frac{1}{2} \min_{x_i, x_j \in X} \| x_i - x_j \|_2. \quad (8) \]

• For \( f \in L_1(\mathbb{R}^d) \) define the Fourier transform

\[ \hat{f}(\omega) = (2\pi)^{-d/2} \int_{\mathbb{R}^d} f(x) e^{-i\omega'x} \, dx. \quad (9) \]

• Let \( \varphi_*(M, \Phi) = \inf_{\|\omega\|_2 \leq 2M} \hat{\Phi}(\omega). \)

• Then,

\[ g(X_j, \Phi_j) \leq \kappa_{\text{upper}}(X_j, \Phi_j)(\kappa_{\text{upper}}(X_j, \Phi_j)\Phi(0) + D), \]

\[ \kappa_{\text{upper}}(X_j, \Phi_j) = \frac{n_j q_X^d}{C_d \varphi_*(M_d / q_X, \Phi_j)} \quad (10) \]
Point-Wise Nominal Bound

• For a nonsingular $\Theta$, define $\Phi_\Theta(x) = \Phi(\Theta x)$.

• Let $h_X$ denote the fill distance

$$h_X = \sup_{x \in \Omega} \min_{x_u \in X} \|x - x_u\|_2.$$  \hspace{1cm} (11)

• Suppose that $\Omega$ is bounded and convex, $\Phi$ is continuous, positive definite, and has $k$ continuous derivatives, and $\Theta$ is non-singular. Then,

$$|f(x) - P(x)| \leq C_\Phi \|\Theta\|_2^{k/2} h_X^{k/2} \|f\|_{N_\Phi(\Theta)(\Omega)}.$$  \hspace{1cm} (12)
Under the assumptions of the previous theorem,

\[ \| f - P \|_{\mathcal{N}_{\Phi_0}(\Omega)} \leq C_\Phi \| \Theta \|_2^{k/2} h_\chi^{k/2} \| f \|_{\mathcal{N}_{\Phi_0 \Phi_0}(\Omega)}. \]  \hspace{1cm} (13)

To allow for different re-scalings at each stage, some additional development and assumptions are necessary.
Notation and Assumptions

- Define $\Psi_k$ recursively as
  
  $\Psi^0 = \Phi,$
  
  $\Psi^k = \Psi^{k-1} \ast \Psi^{k-1},$ \hspace{1cm} (14)

  for $k \in \mathbb{N}$.

- For the kernel on step $j$, take
  
  $\Phi_j = \Psi_{\Theta_j}^{j-j}.$ \hspace{1cm} (15)

- Take $c_2 \geq c_1 > 0$ and $\hat{\Upsilon}$ with
  
  $\omega' \omega \leq \nu' \nu \Rightarrow \hat{\Upsilon}(\omega) \geq \hat{\Upsilon}(\nu),$ \hspace{1cm} (16)

  $c_1 \hat{\Upsilon}(\omega) \leq \hat{\Phi}(\omega) \leq c_2 \hat{\Upsilon}(\omega).$
If $\Phi$ is continuous and positive definite, the above assumptions are satisfied, and $\Theta_{j-1}, \Theta_j$ are non-singular with respective inverses $\Xi'_{j-1}, \Xi'_j$, then

$$\lambda_{\text{max}}(\Theta'_{j-1} \Theta_{j-1} \Xi'_{j-1} \Xi_j) \leq 1$$

$$\implies \|f\|^2_{\mathcal{N}_{\Phi_{j-1}^* \Phi_j}(\Omega)} \leq \left(\frac{C_2}{C_1}\right)^{2^{J-(j-1)}} \frac{|\det(\Xi_{j-1})|}{|\det(\Xi_j)|^2} \|f\|^2_{\mathcal{N}_{\Phi_{j-1}}(\Omega)}$$

for $1 \leq j \leq J$
Nominal Error for Multi-Step Interpolator

- Under the nominal error assumptions,

\[
|f(x) - \sum_{j=1}^{J} \mathcal{P}_j(x)|
\]

\[
\leq C_{\Phi,J} \|f\|_{\mathcal{N}_{\Phi_0}(\Omega)} \|\Theta J\|_2^{k/2} h_{X_j}^{k/2} \prod_{j=1}^{J} \left\{ \frac{\sqrt{|\det(\Xi_{j-1})|}}{|\det(\Xi_j)|} \left( \|\Theta_j\|_2 h_{X_j}^k \right)^{2^{J-j-1}} \right\}
\]

(18)
Conclusions

- The interface between design and modeling: nested space-filling designs vs. the multi-step method.

- Applicable to general data structures by using the thinning algorithm.

- The multi-step method is also useful for fitting other kernel models like support vector machines with massive data.