Effective & Efficient Handling of Ill-Conditioned Correlation Matrices in Kriging & Gradient Enhanced Kriging Emulators Through Pivoted Cholesky Factorization

Keith Dalbey, PhD
Sandia National Labs, Dept 1441
Optimization & Uncertainty Quantification

David Day, PhD, Dept 1442, Numerical Analysis
Mark Hoemmen, Dept 1426, Scalable Algorithms

April 2, 2012
Outline

• Introduction
• Kriging Emulators
• Gradient Enhanced Kriging (GEK)
• Ways to Handle Ill-Conditioning
• Pivoted Cholesky for Kriging
• Pivoted Cholesky for GEK
• Pivoted Cholesky Details
• Results
• Conclusions
• Current / Future Work
Introduction

• Emulators built from small amount of build data can be used as fast surrogates for computationally expensive simulators.

• Kriging/GP emulators are popular because they produce:
  – Best guess estimate that interpolates build data (as long as correlation matrix, $R$, is real symmetric and positive definite) &
  – Estimate of prediction error away from build points.

• But poorly spaced or large amounts of build data can make $R$ ill-conditioned (read as numerically singular) which “voids the warranty” so we need to handle ill-conditioned $R$.

• Gradient Enhanced Kriging (GEK) is becoming popular because of “new” simulation techniques (e.g. Automatic Differentiation, Ryan Starkey) that produce function value + gradient for cost of about 2 or fewer function values (higher info/cost ratio).
Kriging Emulators

Also known as
- Gaussian Process Emulators
- Bayes Linear Method
- “BLUP”
- “BLUE”

Differences among them are minor. All have:
- unadjusted mean (frequently a least squares fit)
- correction/adjustment to mean based on data
- estimated distribution about adjusted mean of possible true surfaces
Kriging Emulators

\[ E[y(x)|\underline{Y}(X)] = \underline{g}(x)^T \underline{\beta} + \underline{r}(x)^T \underline{R}^{-1}(\underline{Y} - \underline{G} \underline{\beta}) \]

\[ \text{Var}[y(x)|\underline{Y}(X)] = \hat{\sigma}^2 \left(1 - \underline{r}(x)^T \underline{R}^{-1} \underline{r}(x)\right) + \ldots \]

\[ \left(\underline{g}(x)^T - \underline{r}(x)^T \underline{R}^{-1} \underline{G}\right)\left(\underline{G}^T \underline{R}^{-1} \underline{G}\right)^{-1}\left(\underline{g}(x)^T - \underline{r}(x)^T \underline{R}^{-1} \underline{G}\right)^T \]

\[ r_j(x) = r\left(x, \underline{X}_j \right) = \exp\left(-\sum_{d=1}^{M} \theta_d \left(x_d - X_{j,d}\right)^2\right) \]

\[ R_{i,j} = R_{j,i} = r\left(\underline{X}_i, \underline{X}_j \right) \quad R_{i,i} = 1 \quad G_{i,j} = g_j(\underline{X}_i) \]

\[ \underline{\beta} = \left(\underline{G}^T \underline{R}^{-1} \underline{G}\right)^{-1}\left(\underline{G}^T \underline{R}^{-1} \underline{Y}\right) \quad \hat{\sigma}^2 = \left(\underline{Y} - \underline{G} \underline{\beta}\right)^T \underline{R}^{-1}\left(\underline{Y} - \underline{G} \underline{\beta}\right)/(N - N_\beta) \]

Correlation parameters, \( \theta \), related to correlation lengths, \( L \), by \( \theta_d = 1/(2L_d^2) \)
Pros:
More info (equations) per cost
Much better conditioned for same # of equations

Cons:
Much worse conditioned for same number of points

How?: Derive with assumptions similar to regular Kriging or replace

\[
Y \rightarrow Y_v, \quad G \rightarrow G_v, \quad r \rightarrow r_v, \quad R \rightarrow R_v, \quad N \rightarrow N_v = N \cdot (1 + M)
\]

\[
Y_v = \begin{bmatrix}
\frac{\partial Y}{\partial X_{:,1}} \\
\vdots \\
\frac{\partial Y}{\partial X_{:,M}}
\end{bmatrix}, \quad G_v = \begin{bmatrix}
\frac{\partial G}{\partial X_{:,1}} \\
\vdots \\
\frac{\partial G}{\partial X_{:,M}}
\end{bmatrix}, \quad r_v = \begin{bmatrix}
\frac{\partial r}{\partial X_{:,1}} \\
\vdots \\
\frac{\partial r}{\partial X_{:,M}}
\end{bmatrix}
\]
Gradient Enhanced Kriging (GEK)

\[
R_\nabla = R_\nabla^T = \begin{bmatrix}
\frac{\partial R}{\partial X_{:,1}} & \frac{\partial R}{\partial X_{:,1}^2} & \frac{\partial R}{\partial X_{:,2}} & \cdots & \frac{\partial R}{\partial X_{:,M}} \\
\frac{\partial^2 R}{\partial X_{:,1} \partial X_{:,2}} & \frac{\partial^2 R}{\partial X_{:,1} \partial X_{:,2}^2} & \frac{\partial^2 R}{\partial X_{:,1} \partial X_{:,2}} & \cdots & \frac{\partial^2 R}{\partial X_{:,1} \partial X_{:,M}^2} \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
\frac{\partial R}{\partial X_{:,M}} & \frac{\partial R}{\partial X_{:,M} \partial X_{:,2}} & \frac{\partial R}{\partial X_{:,M} \partial X_{:,2}} & \cdots & \frac{\partial R}{\partial X_{:,M} \partial X_{:,M}^2}
\end{bmatrix}
\]

\[
\frac{\partial R}{\partial X_{:,1}^T} = \left(\frac{\partial R}{\partial X_{:,1}}\right)^T = \left(\frac{\partial R}{\partial X_{:,2}}\right)^T = -\frac{\partial R}{\partial X_{:,1}^2}
\]

\[
\frac{\partial^2 R}{\partial X_{:,1} \partial X_{:,2}^T} = \left(\frac{\partial^2 R}{\partial X_{:,1} \partial X_{:,2}}\right)^T = \frac{\partial^2 R}{\partial X_{:,1} \partial X_{:,2}^T} = \left(\frac{\partial^2 R}{\partial X_{:,1} \partial X_{:,2}}\right)^T
\]

SANDIA NATIONAL LABORATORIES
Ways to Handle Ill-Conditioning

Some Options:

• **Shrink correlation lengths**, $L$: but adjusted mean may degenerate into unadjusted mean too quickly

• **Add nugget to diagonal of** $R$: but causes Kriging or GEK to approximate (smooth) instead of interpolate (ask yourself if really need to interpolate, you might not)

• **Use mixture correlation model**: weighted sum of $R_S$ with Short $L$ and $R_L$ with Long $L$ (weight $r()$’s too)

• **Use “good” subset of available build data**: will interpolate subset you keep but how do you efficiently select good subset? **New Answer**: Pivoted Cholesky Factorization (checks different “optimal” subset for each candidate $L$ or $\theta$)
Pivoted Cholesky for Kriging

- Pivoted Cholesky sorts equations in $\mathbf{R}$ according to how much new information they contain. When sorted, equations with more information come before equations with less information.

- Can then use LAPACK “rcond()” estimate and bisection search to efficiently determine how many low info equations need to be dropped off the end.

- Dropped equations are ones that contain the least new info and so are easiest to predict.


- Lucas’ level 3 can (and often does) default to level 2 and is not cost competitive with highly optimized level 3 LAPACK (non-pivoted) Cholesky for large matrices.
Pivoted Cholesky for GEK

- $R_\nabla$ can be much larger than $R$ so Pivoted Cholesky on $R_\nabla$ is very slow (compared to LAPACK’s Cholesky)

- Pivoted Cholesky on $R_\nabla$ also prefers derivative equations (higher information content) over response values (bad because function values are more reliable than derivatives)

- **Solution**: Do pivoted Cholesky on $R$ not $R_\nabla$, then apply same ordering to whole points (a point’s function value immediately followed by its derivatives) in $R_\nabla$. Then do LAPACK Cholesky on reordered $R_\nabla$. Reduces Pivoted Cholesky cost by factor of $(1+M)^3$
Pivoted Cholesky Details

• Adaptive least squares trend order

• If using maximum likelihood to choose $\theta$, need to optimize the per equation log likelihood

$$\text{obj}(\theta) = \log(\hat{\sigma}^2) + \frac{\log(\det(R)) + \log(\det(G^T R^{-1} G))}{N - N_\beta}$$

• $R$ only contains information about $\theta$ and inputs $X$, it does not contain information about outputs $Y$ so…
  
  – discontinuity in output not taken into consideration (but can compare difference between “predicted” and actual discarded $Y$ using Mahalanobis distance*)

  – Pivoted Cholesky can be used for sample design

* Using Mahalanobis distance was suggested by Tony O’Hagan
GEK Results for Rosenbrock

Kriging & GEK predictions of Rosenbrock from a nested LHS design

<table>
<thead>
<tr>
<th>Sample Design</th>
<th>Kriging: [error]</th>
<th>GEK: [error]</th>
<th>[error]</th>
</tr>
</thead>
<tbody>
<tr>
<td>N=16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N=32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N=64</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All equations GEK vs GEK for Rosenbrock from a nested LHS design

<table>
<thead>
<tr>
<th>Sample Design</th>
<th>GEK all Eqns: [error]</th>
<th>GEK: [error]</th>
<th>[error]</th>
</tr>
</thead>
<tbody>
<tr>
<td>N=16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N=32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N=64</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

• Pivoted Cholesky can handle ill-conditioning of Kriging’s correlation matrix $R$ by efficiently selecting an “optimal” subset of available build data points.

• Dropped points are ones that contain the least new information and so are easiest to predict.

• Pivoted Cholesky can be used:
  – to detect discontinuities in the response
  – for sample design
  – to determine good set of “Long” correlation lengths in mixture model Kriging / GP
Current / Future Work

• Gaussian Process guided Adaptive mixture Importance Sampling (GPAIS, doesn’t need to interpolate, does need all data so add nugget to handle ill-conditioning) to be presented Thursday 9:30-9:55 AM in MS67 by Laura Swiler & at Quality & Productivity Research Conference June 4 - 7, 2012

• “Synopsis: A tool for managing V&V / UQ activities and data” GUI front end for SVN repository that implements formal “best practices” process (including incremental review and approval, assessment, etc.) for V&V and UQ

Calculating Probability of Failure with GPAIS

• Importance Sampling reduces Monte Carlo’s error variance by drawing more samples from “important” regions & appropriately down-weighting

\[ P_{MC} = \frac{1}{N} \sum_{i=1}^{N} I(x_i), \quad x_i \sim \rho(x_i) \rightarrow P_{IS} = \frac{1}{N} \sum_{i=1}^{N} \left( I(x_i) \frac{\rho(x_i)}{\rho'(x_i)} \right), \quad x_i \sim \rho'(x_i) \]

• \( I(x) \) unknown so optimal importance density \( \rho^*(x) \propto I(x)\rho(x) \) is unknown

• Gaussian Process Adaptive Importance Sampling uses series of improving GP approximations of \( \rho^*(x) \), \( \rho_j'(x) \approx \rho^*(x) \quad j \geq 1, \quad \rho_0'(x) = u(x) \)

in a mixture approximation of \( \rho^*(x) \), \( \rho'(x) = \rho^M(x) = \sum_{j=0}^{\infty} w_j \rho_j'(x) \approx \rho^*(x) \)

• Mixture importance sampling “is … not much worse than importance sampling from the best of the mixture components” [Owen & Zhou 2000]

• j-th component GP approximation is \( \rho_j'(x) \propto \text{E}_j(I(x))\rho(x) \)

• Real valued Expected Indicator \( 0 \leq \text{E}_j(I(x)) \leq 1 \) is the point-wise portion of GP’s Gaussian CDF past the failure threshold
Herbie Function

2D "herbie"

2D "smooth_herbie"
GPAIS On Herbie Function

Herbie2D $P(y<-1.065) \approx 1.5\% \quad P_{\text{isfail}} = 0.01447$ 62.5\% of 200 pts failed
GPAIS On Herbie Function

Herbie2D $P(y<-1.12656) \approx 0.01\%$ $\left| S_{\text{fail}} \right| = 9.73 \times 10^{-5}$ 59.5% of 200 pts failed