

# Improved Quantification of Prediction Error for Kriging Response Surfaces

*Don Jones  
Technical Fellow  
General Motors*

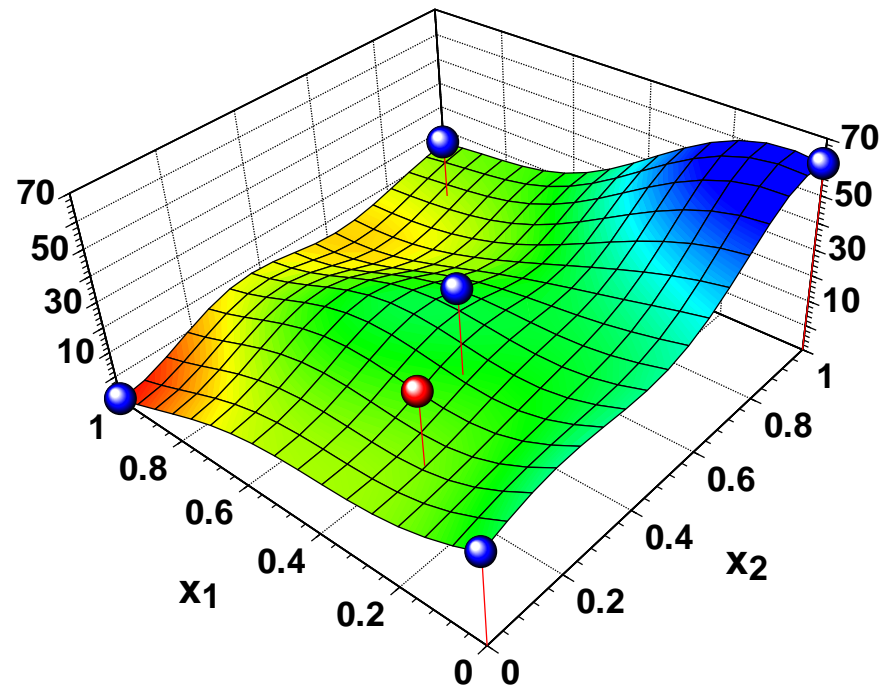
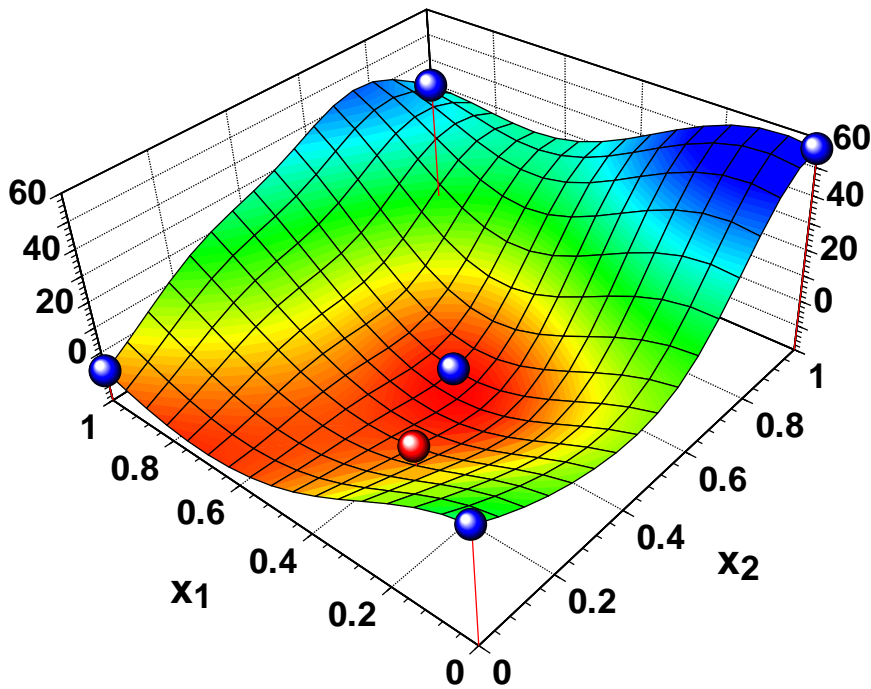
*Presented at the workshop  
“Uncertainty Quantification in Industrial and Energy Applications”  
Institute for Mathematics and its Applications, University of Minnesota*

*June 2, 2011*

# Outline

- Review & Motivation
  - Standard presentation of the Kriging predictor
  - Why prediction error is underestimated in small samples
- How problem addressed in literature
- New ways to compensate for underestimation of prediction error
  - Simple method based on cross-validation
  - Complex method based on likelihood-ratio tests
  - Hybrid approach combining the above
- Conclusion

# Kriging as the Best Linear Unbiased Predictor (BLUP)



● Sampled  
Points  
 $(X_i, Y_i), i=1, N$

● Point where  
making  
prediction  
 $(X_0, Y_0)$

- We show two realizations of a stochastic process on 17x17 grid.
- Derivation of BLUP assumes we know the process parameters and, hence, the smoothness, variability, etc. of typical realizations
- We have sampled **BLUE** points  $(X_i, Y_i), i=1, N$
- We want to predict  $Y$  at the **RED** point  $(X_0, Y_0)$
- The predictor of  $Y_0$  is weighted sum (linear function) of  $Y_i, i=1, N$ .
- The weights minimize sum of squared errors *over all possible realizations* (minimum variance) subject to being correct on average (unbiased)

# Formulas

Our function is  
 $Y(x, \bar{\omega})$  for some  $\bar{\omega}$

This is just one of many possible covariance functions. The parameters  $\theta_k$ ,  $p$ , and  $\sigma^2$  determine the smoothness & variability of the realizations of the stochastic process.

$$Y(\mathbf{x}, \omega) = \sum_{j=1}^k \beta_j f_j(\mathbf{x}) + Z(\mathbf{x}, \omega)$$

$\omega \sim$  state of the world or "realization"

$$Z(\mathbf{x}, \omega) = \text{Normal}(0, \sigma^2)$$

$$\text{Cov}(\mathbf{x}_i, \mathbf{x}_j) \equiv E_{\omega} [Z(\mathbf{x}_i, \omega)Z(\mathbf{x}_j, \omega)] = \sigma^2 \exp \left[ -\sum_{k=1}^d \theta_k |x_{ik} - x_{jk}|^p \right]$$

Given  $n$  sampled points  $(\mathbf{x}_i, y_i)$  the BLUP is

$$\hat{y}(\mathbf{x}) = \sum_{j=1}^k \hat{\beta}_j f_j(\mathbf{x}) + \mathbf{r}' \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F} \hat{\boldsymbol{\beta}})$$

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}' \mathbf{R}^{-1} \mathbf{F})^{-1} \mathbf{F}' \mathbf{R}^{-1} \mathbf{y}$$

$\hat{\boldsymbol{\beta}}$  is a linear function of  $\mathbf{y}$  and so  $\hat{y}(\mathbf{x})$  is a linear function of  $\mathbf{y}$

But note that weights on  $\mathbf{y}$  are nonlinear functions of  $\mathbf{x}$  and  $(\mathbf{x}_i, y_i), i = 1, \dots, n$ .

$$\text{MSE}(x) = \sigma^2 \left[ 1 - \mathbf{r}' \mathbf{R}^{-1} \mathbf{r} + (\mathbf{f} - \mathbf{F}' \mathbf{R}^{-1} \mathbf{r})' (\mathbf{F}' \mathbf{R}^{-1} \mathbf{F})^{-1} (\mathbf{f} - \mathbf{F}' \mathbf{R}^{-1} \mathbf{r}) \right]$$

where  $\mathbf{R}_{ij} = \text{Cov}(\mathbf{x}_i, \mathbf{x}_j)$ ,  $\mathbf{r}_i = \text{Cov}(\mathbf{x}_i, \mathbf{x})$ ,  $\mathbf{F}_{ij} = f_j(\mathbf{x}_i)$

# The BLUP is BEST ... if its assumptions are met!

- Mean is linear in parameters with known regression terms
  - **Reasonable** because coefficients not assumed known and because the stochastic process can compensate for errors in the regression.
- Know distribution & functional form of covariance function
  - **Reasonable** because the functional form of the covariance function is very flexible...it can capture very different functional behavior using different parameters.
- Know parameters of covariance function (sigma, theta and p)
  - **NOT Reasonable.** We have no clue!

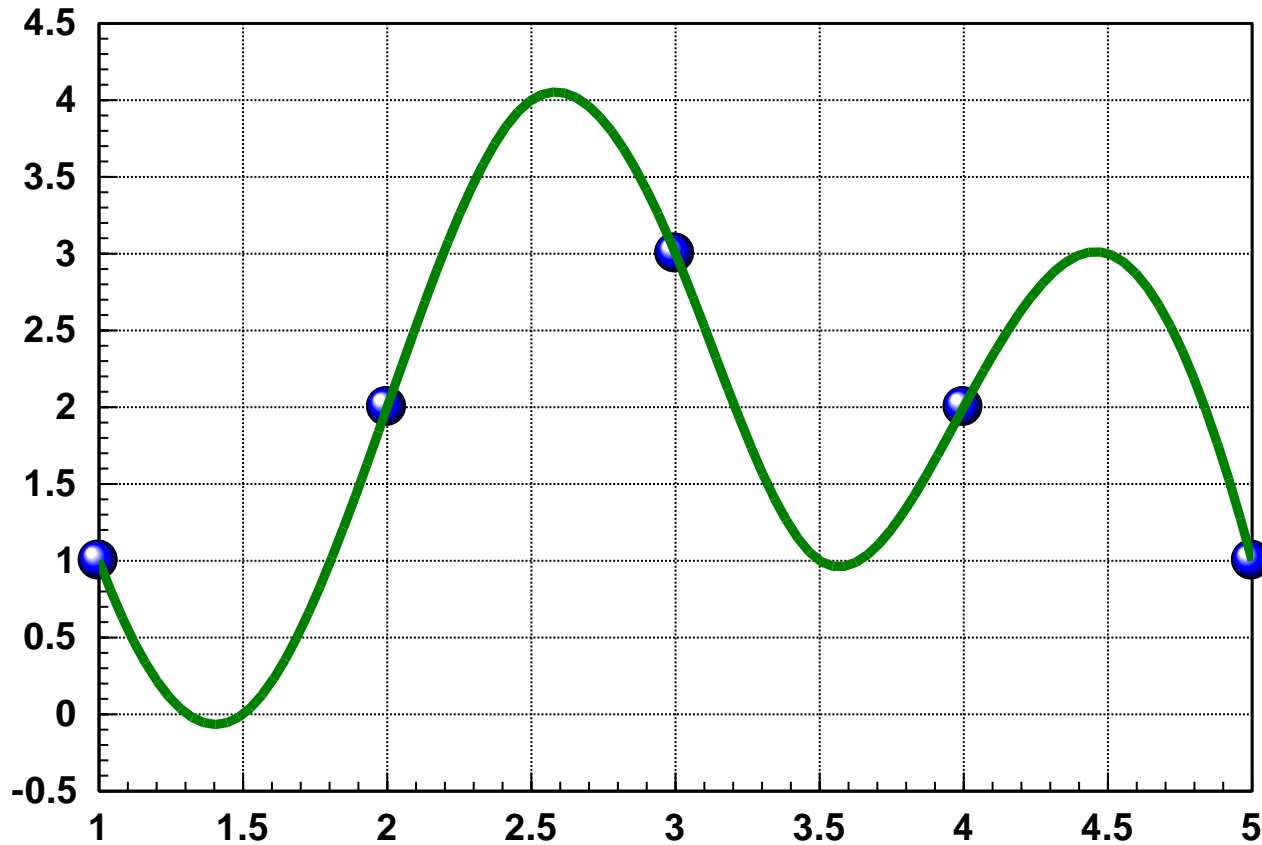
## In practice, we use a 2-Step process

1. Use maximum likelihood estimation (MLE) to estimate the parameters  $\sigma$ ,  $\theta$ , and  $p$ .
2. Substitute these values into the formulas for the BLUP.

**But the BLUP derivation assumes  $\sigma$ ,  $\theta$ ,  $p$  are KNOWN!  
Thus, we are violating one of the assumptions.**

- Two-step process ignores our uncertainty about the MLE estimates
- As a result, we usually underestimate the error in our predictions.
- Damage from the violated assumption is greatest in small samples.
- As  $n \rightarrow \text{infinity}$ , the MLE estimates converge to the true values, and underestimation of prediction error is not too bad

## Another intuitive reason for underestimating error...

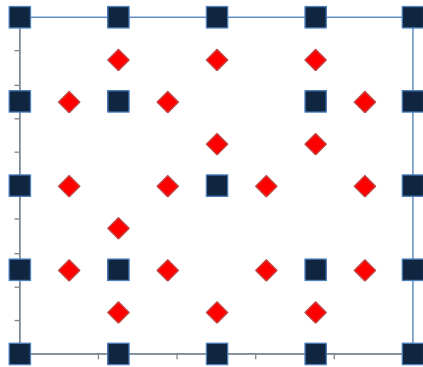


If the function is active, a small sample is unlikely to sample the highest peaks and lowest valleys.

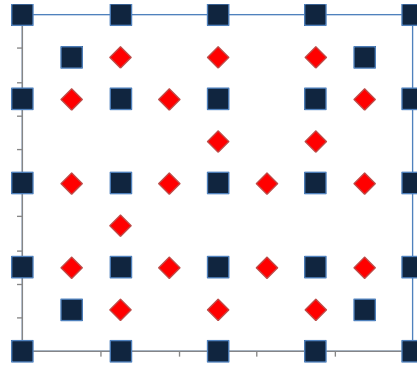
So the variability will seem lower than it really is.

# Numerical Example

21



29

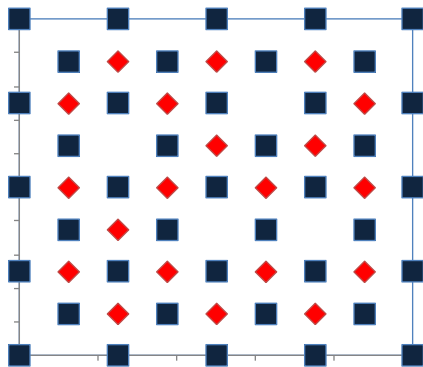


Generate 1000 realizations of Gaussian stochastic process over a 9x9 grid using:  $p=1.95$ ,  $\theta(1)=1$ ,  $\theta(2)=2$ ,  $\sigma=20$ , and regression model  $= 10 - 3 \cdot x_1 + 7 \cdot x_2$

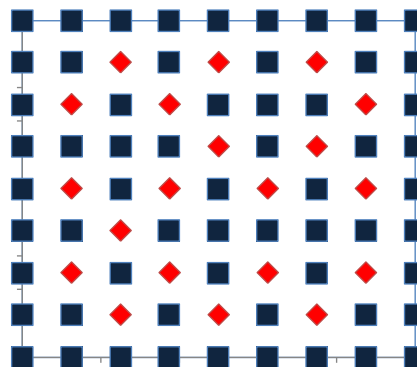
For each realization, select some points for fitting (**BLUE**) and some for predicting (**RED**).

Try  $N=21, 29, 41$ , and  $61$  **BLUE** points. Use same  $20$  **RED** points in all cases.

41



61



Do MLE on BLUE points allowing  $\theta$  in  $[0.01, 10]$  with  $p$  fixed at  $1.95$ . Substitute MLE estimates in BLUP. Get 90% confidence interval.

Track fraction of time true value at RED point is in the confidence interval. Should be 90%.

Numerical evidence verifies that problem exists

<i><b><math>n</math></b></i>	<i><b>Percent of Time Data in 90% Confidence Interval</b></i>	
	<b>Using MLE Estimates</b>	<b>Using True Values</b>
<b>21</b>	80.32%	90.00%
<b>29</b>	80.81%	90.19%
<b>41</b>	83.78%	89.88%
<b>61</b>	88.34%	89.96%

# Adjustment factors

In the simulations, can compute how much the confidence intervals need to be enlarged so that they do cover 90% of the data.

This is the **Adjustment Factor**.

The proper adjustment factor depends upon the number of points, the number of variables, and the “true” thetas used to generate realizations. Experiments show it does not depend upon sigma.

Adjustment Factor to 90% Confidence Interval				
		Low Theta	Medium Theta	High Theta
n	21	1.60	1.33	1.15
	29	1.39	1.30	1.13
	41	1.28	1.19	1.10
	61	1.05	1.05	1.03

One could, in principle, compute a big table of adjustment factors indexed by number of points, number variables, and theta.

But how would you know which entry to look up????  
You don't know the true theta values!

# How problem addressed in the literature

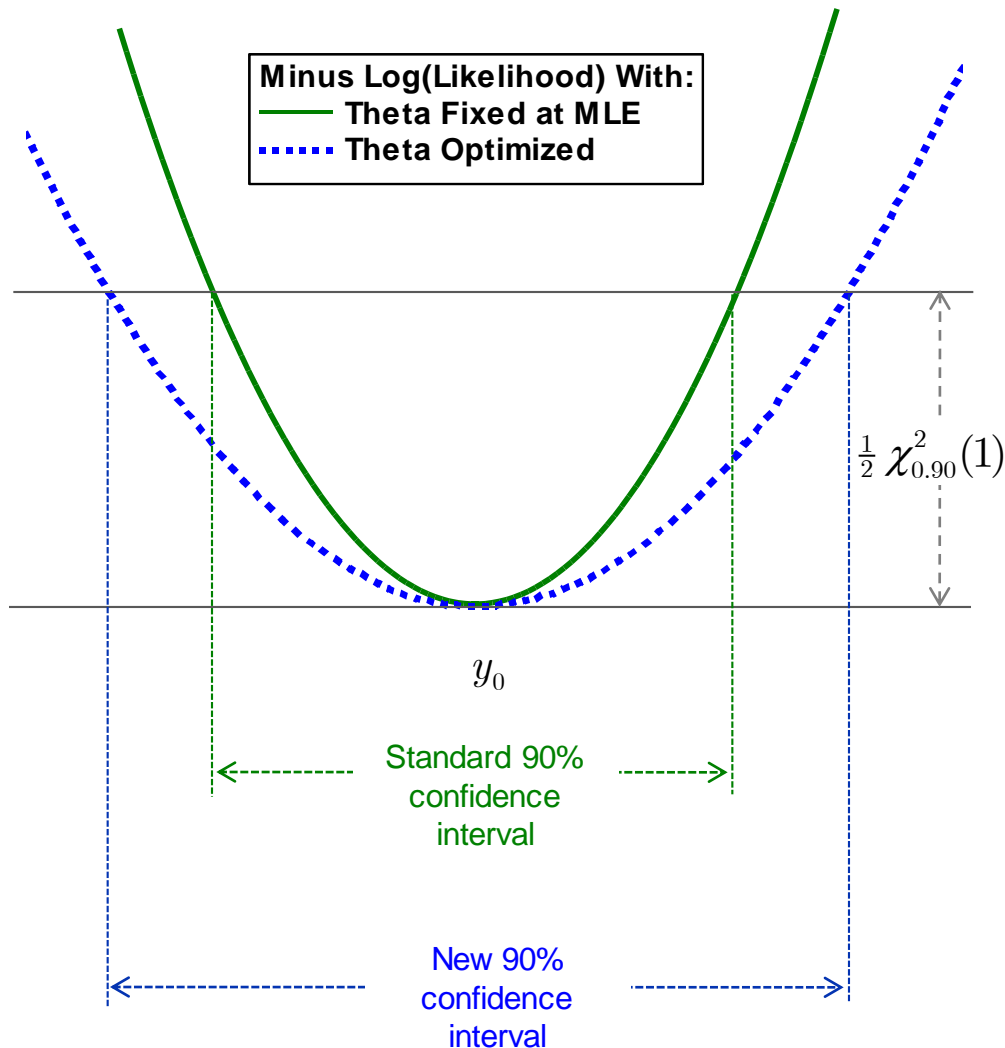
- Bootstrapping
  - Do simulation like I have done with the MLE estimates in place of the assumed true values. Moreover, do this for each point at which you want to make a prediction. Amounts to looking up the Adjustment Factor using the estimated thetas as if they were true.
  - Because get Adjustment Factor  $> 1$ , it helps, but doesn't solve problem because the Adjustment Factor is wrong (based on estimated, not true thetas)
    - Jack Kleijnen, Wim van Beers, Inneke van Nieuwenhuyse, "Expected improvement in efficient global optimization through bootstrapped kriging," to appear in the *Journal of Global Optimization*.
- Priors on correlation parameters
  - Goal is to avoid "bad estimates" in small samples
  - Effect of priors function diminishes as  $n$  gets large
  - Usual motivation is to avoid large theta parameters because these give rough surfaces...but low thetas will underestimate error!
    - Runzi Li, "Analysis of Computer Experiments Using Penalized Likelihood in Gaussian Kriging Models." *Technometrics*, vol 47, no. 2, pp. 111-120.
    - Daniel Lizotte, "An Experimental Methodology for Response Surface Optimization Methods," to appear in *Journal of Global Optimization*

# Intuitive approach based on Cross-validation HELPS

- Get predictions & standard errors from leave-one-out cross-validation.
- See what Adjustment Factor is needed to make 90% of the data fall in the confidence interval on cross-validation.
  - NOTE: If Adjustment Factor comes out less than 1, just use 1.
  - In my example, comes out greater than 1 only 34% of time
- Use this Adjustment Factor for out-of-sample predictions

<i>n</i>	<i>Percent of Time Data in 90% Confidence Interval</i>	
	Using MLE Estimates	MLE with Crossvalidation Adjustment
21	80.32%	81.88%
29	80.81%	82.55%
41	83.78%	85.50%
61	88.34%	90.26%

# More complex approach that HELPS



- Add new “pseudo” observation at  $x_0$  with **assumed value**  $y_0$ .
- Look at likelihood of augmented sample, now with  $N+1$  points, as function of assumed  $y_0 = f(x_0)$
- If use MLE estimates of correlation parameters, augmented likelihood is optimized at  $y_0 = \text{Kriging predictor}$
- As vary  $y_0$  from this value, likelihood degrades. Likelihood Ratio Test says 90% confidence interval generated if accept degradation  $\leq \text{ChiSqr}(1, 0.90)$
- New approach:
  - As vary  $y_0$ , also optimize over correlation parameters!
  - In this way we avoid assuming the MLE estimates are true.
  - Confidence interval will be wider

# Results

Both the cross-validation and augmented log-likelihood help.

Hybrid approach uses the min of the lower limits from the two approaches and the max of the two upper limits.

<i>n</i>	<i>Percent of Time Data in 90% Confidence Interval</i>			
	Using MLE Estimates	MLE with Crossvalidation Adjustment	Augmented Loglikelihood Adjustment	Hybrid: Max of Crossvalidation and Augmented LogLik
21	80.32%	81.88%	81.54%	82.86%
29	80.81%	82.55%	82.16%	83.52%
41	83.78%	85.50%	84.63%	86.14%
61	88.34%	90.26%	88.86%	90.53%

# Conclusion

- Underestimation in small samples is hard to correct. Data may simply not have shown us the true variability of function. No fancy statistical dancing can solve that.
- But we can do some checks via cross-validation.
- And we can avoid assuming MLE estimates are true using the new augmented log-likelihood approach.
- A combination of these two approaches does a fairly good job.
- Caveat:
  - Results shown were for synthetic realizations of stochastic process in low dimensions
  - But conceptual reasoning makes sense, and expect approach to generalize to more variables and real-world data sets

# Kleijnen's results with bootstrapping

- Uses 6-hump Camel test function in 2D
- $n$  is the number of sampled points
- In the table below, “Classic” is the BLUP with estimated thetas
- In the table below “Bootstrap” means the following.  
He generates many realizations of stochastic process with the MLE parameters at the points in the initial sample and the point being predicted. For each realization, a fresh MLE is done and prediction made. The mean squared error over these realizations is taken as the “bootstrap” error and used to make confidence intervals.

$n$	5	20	50	80
Classic	0.7198	0.8065	0.8637	0.8866
Bootstrap	0.7643	0.8459	0.8747	0.8903

Results are on a par with the cross-validation based adjustment, but more computationally intensive.