Kernel Approximation: From Regression to Interpolation

Lulu Kang[1] and V. Roshan Joseph[2]

[1]Department of Applied Mathematics

Illinois Institute of Technology, Chicago, IL 60616, USA

lkang2@iit.edu

[2]The H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology, Atlanta, GA 30332, USA

roshan@gatech.edu

Abstract

In this paper we introduce a new interpolation method, known as *kernel interpolation* (KI), for modeling the output from expensive deterministic computer experiments. We construct it by repeating a generalized version of the classic Nadaraya-Watson kernel regression an infinite number of times. Although this development is numerical, we are able to provide a statistical framework for KI using a nonstationary Gaussian process. This enables us to quantify the uncertainty in the predictions as well as estimate the unknown parameters in the model using empirical Bayes method. Through some theoretical arguments and numerical examples, we show that KI has better prediction performance than the popular kriging method in certain situations.

Keywords: Computer experiments; Gaussian process; Kernel regression; Kriging.

1 Introduction

Consider the regression model

$$y = f(\boldsymbol{x}) + \epsilon, \tag{1}$$

and the problem of estimating $f(\boldsymbol{x})$ from the data $(\boldsymbol{x}_i, y_i)$, $i = 1, \cdots, n$, where the vector $\boldsymbol{x}_i$ belongs to an input space $\Omega \subset \mathbb{R}^p$. We assume ϵ to be independently and identically distributed with mean 0 and variance σ^2, although these assumptions can easily be relaxed.

Arguably this is one of the most important problems in statistics and the literature on it is huge. Here we touch upon some of the relevant parts of the literature to motivate the work done in this paper.

We focus on nonparameteric approaches. Among all the nonparametric regression methods, Nadaraya-Watson kernel regression ([17] and [26]) has the simplest prediction form, which is given by

$$\hat{y}(\boldsymbol{x}) = \frac{\sum_{i=1}^{n} K(\boldsymbol{x}, \boldsymbol{x}_i) y_i}{\sum_{i=1}^{n} K(\boldsymbol{x}, \boldsymbol{x}_i)}, \tag{2}$$

where $K(\cdot, \cdot) : \Omega \times \Omega \to \mathbb{R}$ is called the *kernel* function. Several improvements to kernel regression have appeared in the literature such as local polynomial smoothing ([1], [2], [7], and [22]). The kernel function contains some unknown parameters known as bandwidth parameters which are estimated from the data using, for example, cross validation [11].

An alternative approach to the nonparametric regression was being developed in geostatistics known as *kriging* [13], which can be written in the form (see [24], pp. 63-64)

$$\hat{y}(\boldsymbol{x}) = \mu + \sum_{i=1}^{n} K(\boldsymbol{x}, \boldsymbol{x}_i) b_i,$$

where μ and $\boldsymbol{b} = (b_1, \cdots, b_n)'$ are unknown parameters. When μ is a constant rather than a function of $\boldsymbol{x}$, it is called *ordinary kriging* (OK) predictor, which is the most commonly used kriging method due to its simplicity. Here the kernel function is usually known as the correlation function. The values of μ and $\boldsymbol{b}$ are chosen to make $\hat{y}(\boldsymbol{x})$ the best linear unbiased predictor. The resulting predictor has a simple form and is given by

$$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{r}(\boldsymbol{x})'(\boldsymbol{R} + \frac{\sigma^2}{\tau^2}\boldsymbol{I}_n)^{-1}(\boldsymbol{y} - \hat{\mu}\boldsymbol{1}), \qquad \hat{\mu} = \frac{\boldsymbol{1}'(\boldsymbol{R} + \sigma^2/\tau^2 \boldsymbol{I}_n)^{-1}\boldsymbol{y}}{\boldsymbol{1}'(\boldsymbol{R} + \sigma^2/\tau^2 \boldsymbol{I}_n)^{-1}\boldsymbol{1}}, \tag{3}$$

where $\boldsymbol{1}$ is an $n \times 1$ vector of 1's, $\boldsymbol{r}(\boldsymbol{x})' = (K(\boldsymbol{x}, \boldsymbol{x}_1), \cdots, K(\boldsymbol{x}, \boldsymbol{x}_n))$, $\boldsymbol{R}$ is an $n \times n$ matrix with elements $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\boldsymbol{I}_n$ is the identity matrix, and τ^2 is another unknown parameter that needs to be estimated from the data. One distinguishing feature of the kriging predictor from the other kernel regression methods is that it explicitly uses the noise variance σ^2. As the noise reduces to zero ($\sigma^2 \to 0$), the kriging predictor interpolates the data, which makes sense. This is not achieved by the kernel regression methods. It is a crucial difference between these two approaches, which will be exploited in this work.

Because interpolation has a longer history in mathematics, applied mathematicians have independently developed a closely related technique to kriging known as *radial basis function* (RBF) method. Recently [4] has made some important connections between approximate moving least squares and RBF. Our work is closely related to theirs but uses kernel regression instead of approximate moving least squares.

In this work we show that by iteratively applying a generalized version of the kernel regression on the residuals, we can obtain a predictor that interpolates the data. We show that this new predictor has several desirable properties and has better prediction performance than kriging under some conditions (discussed in Section 5). Although our initial development is numerical, using its connections with kriging, we postulate a stochastic model underlying our new predictor. This enables us to obtain empirical Bayes estimates of the unknown parameters as well as the prediction intervals.

However, the foregoing advantages of the predictor come at the expense of increased computations and therefore, we are able to demonstrate its performance only for small to moderate sized data. Hence, in this work we primarily focus on its applications to the modeling of expensive functions, where the data are hard to obtain. Extensions to large data can be handled in the future after developing more efficient methods for computation.

Modeling of expensive functions, also known as computer experiments, is an emerging field in statistics ([23]). Even when the functions are deterministic, there are uncertainties in predictions at the unobserved locations and therefore a statistical approach is useful for quantifying the uncertainty. Kriging is the major technique used in the computer experiment literature owing to its probabilistic framework. See [24] and [3] for details. Because our proposed predictor has close connections to kriging, the problems in computer experiments can be a good test bed for us to evaluate its performance.

This article is organized as follows. In Section 2, we start with a generalized version of the kernel regression and develop our new predictor known as *kernel interpolation* (KI) using an iterative implementation of the kernel regression. In Section 3, we provide a statistical framework for our method using a nonstationary Gaussian process (GP) model. The estimation of the unknown parameters in the model is discussed in Section 4. Some properties of the KI are described in Section 5. In Section 6, we apply the new predictor on two simulated examples and compare its performance with some popular interpolation and regression methods. In Section 7, the stochastic model for KI is extended to the case where there is random noise involved in the observations, i.e., $\sigma^2 > 0$ and we conclude the paper with some remarks and future research directions in Section 8.

2 Kernel Interpolation: A Numerical Approach

Consider a more general version of the kernel regression given by

$$\hat{y}(\boldsymbol{x}) = \mu + \frac{\boldsymbol{r}(\boldsymbol{x})'(\boldsymbol{y} - \mu\mathbf{1})}{s(\boldsymbol{x})}, \tag{4}$$

where $s(\boldsymbol{x})$ is a positive scaling function and μ is an unknown mean parameter. For the kernel regression in (2), $s(\boldsymbol{x}) = \sum_{i=1}^{n} K(\boldsymbol{x}, \boldsymbol{x}_i) = \boldsymbol{r}(\boldsymbol{x})'\mathbf{1}$, which restricts the predicted values $\hat{y}(\boldsymbol{x}_i)$ to be between the smallest and largest values of y_i's. To overcome this restriction and make (4) more flexible, we can use other types of positive functions such as

$$s(\boldsymbol{x}) = \sum_{i=1}^{n} K(\boldsymbol{x}, \boldsymbol{x}_i) c_i = \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{c}, \tag{5}$$

where $\boldsymbol{c}$ is a positive vector in $\mathbb{R}^n$.

The kernel regression in (4) only smooths the data and does not interpolate, which is not suitable if σ^2 is small in (1). Therefore, to improve the predictions we may consider applying another kernel regression of the form (4) to the residuals. Let $\boldsymbol{e}_1$ be the vector of residuals from the kernel regression, i.e., $\boldsymbol{e}_1 = \boldsymbol{y} - \hat{\boldsymbol{y}}_1$, where $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}_1$ are the vector of responses and predictions using (4). Thus, we obtain a new predictor:

$$\hat{y}_2(\boldsymbol{x}) = \mu + \frac{\boldsymbol{r}(\boldsymbol{x})'(\boldsymbol{y} - \mu\mathbf{1})}{s(\boldsymbol{x})} + \frac{\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{e}_1}{s(\boldsymbol{x})}.$$

For simplicity, we have set the mean parameter to be zero and used the same kernel and scaling functions as before in the second application of the kernel regression, which will be generalized at a later stage. Extending this idea, we can repeatedly apply the kernel regression to the residuals and add it back to the predictor. Suppose we repeat this N times, then the new predictor becomes:

$$\hat{y}_N(\boldsymbol{x}) = \mu + \frac{\boldsymbol{r}(\boldsymbol{x})'(\boldsymbol{y} - \mu\mathbf{1})}{s(\boldsymbol{x})} + \frac{\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{e}_1}{s(\boldsymbol{x})} + \cdots + \frac{\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{e}_{N-1}}{s(\boldsymbol{x})}, \tag{6}$$

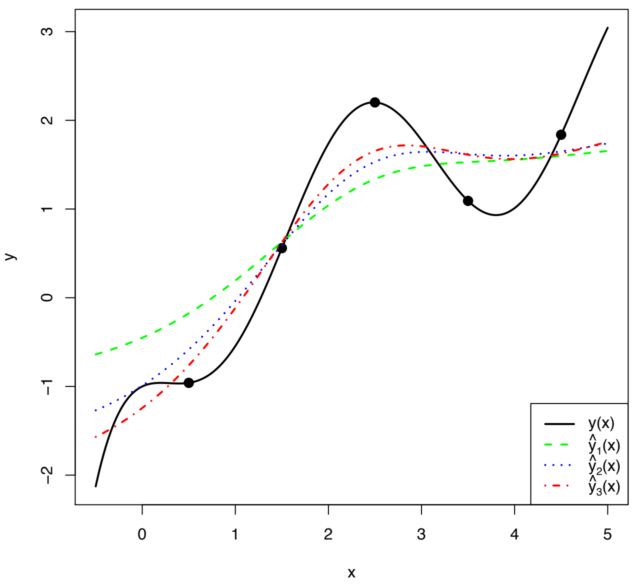

Figure 1: Comparison between the true function $y(x) = 0.5x - \sin(2x) - \exp(-2x)$ and iterated kernel regressions $\hat{y}_1(x)$, $\hat{y}_2(x)$, and $\hat{y}_3(x)$ with $\theta = 0.50$.

where $\boldsymbol{e}_j = \boldsymbol{y} - \hat{\boldsymbol{y}}_j$ are the residuals from the jth stage. Let $\boldsymbol{U} = \boldsymbol{S}^{-1}\boldsymbol{R}$, where $\boldsymbol{R}_{rs} = K(\boldsymbol{x}_r, \boldsymbol{x}_s)$ and $\boldsymbol{S}$ be the diagonal matrix $\boldsymbol{S} = \operatorname{diag}\{s(\boldsymbol{x}_1), \ldots, s(\boldsymbol{x}_n)\}$. Then

$$\boldsymbol{e}_j = \boldsymbol{y} - \hat{\boldsymbol{y}}_j = \boldsymbol{e}_{j-1} - \boldsymbol{U}\boldsymbol{e}_{j-1} = (\boldsymbol{I}_n - \boldsymbol{U})\boldsymbol{e}_{j-1} = (\boldsymbol{I}_n - \boldsymbol{U})^j(\boldsymbol{y} - \mu\boldsymbol{1}).$$

Throughout the paper, we define $\boldsymbol{e}_0 = \boldsymbol{y} - \mu\boldsymbol{1}$ and $\boldsymbol{A}^0 = \boldsymbol{I}_n$ for any matrix $\boldsymbol{A}$. Thus (6) can be written succinctly as

$$\hat{y}_N(\boldsymbol{x}) = \mu + \boldsymbol{u}(\boldsymbol{x})' \sum_{i=0}^{N-1} (\boldsymbol{I}_n - \boldsymbol{U})^i(\boldsymbol{y} - \mu\boldsymbol{1}), \tag{7}$$

where the weight vector $\boldsymbol{u}(\boldsymbol{x})$ is defined as

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})/s(\boldsymbol{x}). \tag{8}$$

Figure 1 shows an example of these iterative fittings with the Gaussian kernel

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\sum_{k=1}^{p} \theta_i (x_{k,i} - x_{k,j})^2\right)$$

and $s(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})'\mathbf{1}$. We can see that overall the fitted curves get closer to the observations as N is increased. Naturally this raises the question: will the kernel regression in (7) interpolate the observations as N becomes infinitely large? The answer to this question is positive. As $N \to \infty$, the kernel regression becomes an interpolator as stated in Theorem 1. To prove it, we need the condition that the kernel functions are symmetric positive definite functions, which is often assumed in nonparametric regression literature and is a necessary condition for valid correlation functions. Under this assumption, $\boldsymbol{R}$ becomes a symmetric positive definite matrix.

Theorem 1. *If the kernel function $K(\cdot,\cdot)$ is symmetric positive definite and $s(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathbb{R}^p$, as $N \to \infty$, the kernel regression* (7) *converges to*

$$\hat{y}_{KI}(x) = \boldsymbol{u}(\boldsymbol{x})'\boldsymbol{U}^{-1}\boldsymbol{y}, \tag{9}$$

which interpolates all the observations $(\boldsymbol{x}_i, y_i)$ for $i = 1, \ldots, n$, if and only if

$$\lambda_i(\boldsymbol{U}) < 2, \ \textit{for } i = 1, \ldots, n. \tag{10}$$

where $\lambda_i(\boldsymbol{U})$ denotes the ith eigenvalue of $\boldsymbol{U}$.

All the proofs are given in the Appendix. Interestingly, the unknown mean parameter μ disappears in the limit. Thus we obtain a new interpolator, which we name as *kernel interpolation* (KI). In Figure 2, we use the same test function and the five equally spaced design points as in Figure 1, and fit KI with $\boldsymbol{s}(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})'\mathbf{1}$ and OK. Figure 2 shows that both of the predictors interpolate the data, but KI gives better fitting which is closer to the true test function.

Note that $\lambda_i(\boldsymbol{U}) < 2$ for $i = 1, \ldots, n$ is the sufficient and necessary condition for the convergence $\hat{y}_N(\boldsymbol{x}) \to \hat{y}_{KI}(\boldsymbol{x})$. This condition looks quite restrictive, but the following proposition shows that it can be easily relaxed through a simple modification of the initial predictor. Introduce a new parameter η in the kernel regression as $\hat{y}(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{y}/(\eta s(\boldsymbol{x}))$. Interestingly, η disappears in the limit, leaving the KI predictor unaffected.

Proposition 1. *Given $\boldsymbol{s}(\boldsymbol{x}) > 0$ for any $\boldsymbol{x}$, there exists a scalar $\eta > 0$ such that $\lambda_i(\boldsymbol{U}) < 2\eta$ for $i = 1, \ldots, n$. The kernel regression of the form*

$$\hat{y}_N(\boldsymbol{x}) = \mu + \eta^{-1}\boldsymbol{u}(\boldsymbol{x})' \sum_{i=0}^{N-1} (\boldsymbol{I}_n - \eta^{-1}\boldsymbol{U})^i (\boldsymbol{y} - \mu\mathbf{1})$$

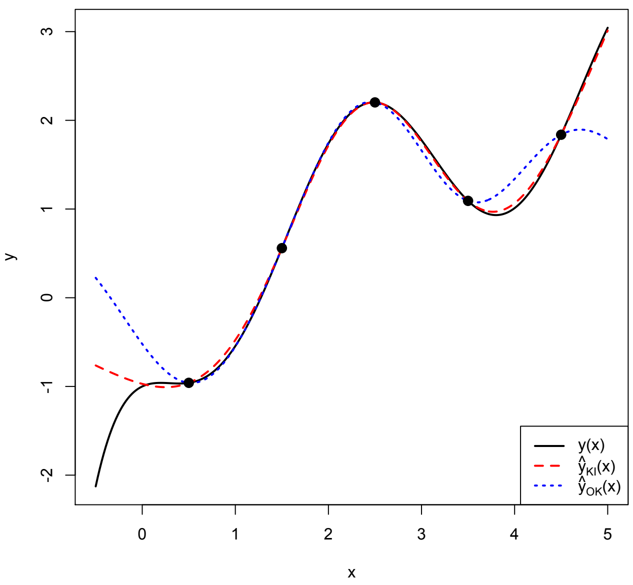

Figure 2: Comparison between KI and OK for the test function $y(x) = 0.5x - \sin(2x) - \exp(-2x)$ with $\hat{\theta}_{OK} = 0.97$, $\theta_{KI} = 0.50$ and $\boldsymbol{c} = \boldsymbol{1}$

converges to the kernel interpolator $\hat{y}_{KI}(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x})'\boldsymbol{U}^{-1}\boldsymbol{y}$ *as* $N \to \infty$*.*

KI can be extended to a more general form with different kernel and scaling functions. Let $K_i(\cdot,\cdot)$ be a symmetric positive definite kernel function and $s_i(\boldsymbol{x}) > 0$ a positive scaling function, for $i = 1, \ldots, N$. Then, similar to the development of (7), N iterations of the kernel regression on the residuals gives

$$\hat{y}_N(\boldsymbol{x}) = \mu + \left(\sum_{i=1}^{N} \boldsymbol{u}_i(\boldsymbol{x})' \prod_{j=0}^{i-1} (\boldsymbol{I}_n - \boldsymbol{U}_j) \right) (\boldsymbol{y} - \mu \boldsymbol{1}),$$

where $\boldsymbol{U}_0 = \boldsymbol{0}$, $\boldsymbol{u}_i(\boldsymbol{x}) = \boldsymbol{r}_i(\boldsymbol{x})/s_i(\boldsymbol{x})$ and $\boldsymbol{U}_i = \boldsymbol{S}_i^{-1}\boldsymbol{R}_i$. Here $\boldsymbol{r}_i(\boldsymbol{x})' = (K_i(\boldsymbol{x},\boldsymbol{x}_1), \ldots, K_i(\boldsymbol{x},\boldsymbol{x}_n))$, $\boldsymbol{S}_i = \text{diag}\{s_i(\boldsymbol{x}_1), \ldots, s_i(\boldsymbol{x}_n)\}$, and $\boldsymbol{R}_i$ is a $n \times n$ symmetric positive definite matrix with elements $K_i(\boldsymbol{x}_s, \boldsymbol{x}_k)$. As given below, the corresponding KI also has a more general form.

Proposition 2. *If* $||\boldsymbol{I}_n - \boldsymbol{U}_i|| < 1$ *for* $i = 1, 2, \ldots,$ *as* $N \to \infty$*, the kernel regression converges to*

$$\hat{y}_{KI}(\boldsymbol{x}) = \lim_{N \to \infty} \hat{y}_N(x) = \mu + \left(\sum_{i=1}^{\infty} \boldsymbol{u}_i(\boldsymbol{x})' \prod_{j=0}^{i-1} (\boldsymbol{I}_n - \boldsymbol{U}_j) \right) (\boldsymbol{y} - \mu \boldsymbol{1}), \tag{11}$$

which interpolates all the observations $(\boldsymbol{x}_i, y_i)$ *for* $i = 1, \ldots, n$. *Here* $||\cdot||$ *can be any matrix norm and the product* $\prod_{i=1}^{N} \boldsymbol{A}_i = \boldsymbol{A}_N \boldsymbol{A}_{N-1} \ldots \boldsymbol{A}_1$ *denotes left matrix multiplication.*

KI has interesting connections to some of the existing interpolators. For example, as stated below, the kriging predictor in (3) with $\sigma^2 = 0$ can be seen as a special case of the KI.

Corollary 1. *Let* $\boldsymbol{s}(\boldsymbol{x}) \equiv 1$ *and* $\eta > 0$. *Then as* $N \to \infty$,

$$\hat{y}_N(\boldsymbol{x}) = \mu + \eta^{-1}\boldsymbol{r}(\boldsymbol{x})' \sum_{i=0}^{N-1} (\boldsymbol{I}_n - \eta^{-1}\boldsymbol{R})^i (\boldsymbol{y} - \mu\boldsymbol{1}), \tag{12}$$

converges to the kriging predictor $\hat{y}(\boldsymbol{x}) = \mu + \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}(\boldsymbol{y} - \mu\boldsymbol{1})$, *if and only if* $\lambda_i(\boldsymbol{R}) < 2\eta$ *for* $i = 1, \ldots, n$.

A similar representation to (12) was used by [20] for the purpose of achieving numerical stability. Another interesting connection is with the *limit kriging* predictor proposed in [10] as stated below in Corollary 2. Note, however, that in the original definition of limit kriging $\boldsymbol{s}(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{1}$ is not guaranteed to be positive.

Corollary 2. *If* $\boldsymbol{s}(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{c}$ *and* $\boldsymbol{c} = \boldsymbol{R}^{-1}\boldsymbol{1}$, *kernel interpolation in* (9) *becomes*

$$\hat{y}_{LK}(\boldsymbol{x}) = \frac{\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{y}}{\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{1}},$$

which is the limit kriging predictor.

3 Kernel Interpolation: A Statistical Approach

By exploiting the connections between the OK predictor and kernel interpolator, we can develop a statistical framework for KI. This will enable us to obtain empirical Bayes estimates of the unknown parameters and construct prediction intervals.

The GP model for OK is

$$y(\boldsymbol{x}) - \mu = Z(\boldsymbol{x}), \quad \text{where } Z(\boldsymbol{x}) \sim GP(0, \tau^2 K(\cdot,\cdot)),$$

where μ and τ^2 are the unknown mean and variance parameters. Denote the unknown correlation parameters in $K(\cdot,\cdot)$ by $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_p)'$. Then, the posterior distribution of $y(\boldsymbol{x})$ given the data $\boldsymbol{y}$ is

$$y(\boldsymbol{x})|\boldsymbol{y},\mu,\tau^2,\boldsymbol{\theta} \sim N\left(\mu + \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}(\boldsymbol{y}-\mu\boldsymbol{1}), \tau^2\left\{1-\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x})\right\}\right). \tag{13}$$

Integrating out μ by assuming a noninformative prior distribution: $p(\mu) \propto 1$, we obtain

$$y(\boldsymbol{x})|\boldsymbol{y},\tau^2,\boldsymbol{\theta} \sim N\left(\hat{\mu} + \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}(\boldsymbol{y}-\hat{\mu}\boldsymbol{1}), \tau^2\left\{1-\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) + \frac{(1-\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{1})^2}{\boldsymbol{1}'\boldsymbol{R}^{-1}\boldsymbol{1}}\right\}\right), \tag{14}$$

where $\hat{\mu} = \boldsymbol{1}'\boldsymbol{R}^{-1}\boldsymbol{y}/\boldsymbol{1}'\boldsymbol{R}^{-1}\boldsymbol{1}$. It is also easy to integrate out τ^2 [24], but we will not do it here for the ease of comparison with the new predictor.

Now consider the KI predictor in (9) with $s(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{c}$. Since $\boldsymbol{u}(\boldsymbol{x})'\boldsymbol{U}^{-1}\boldsymbol{1} = 1$ for all $\boldsymbol{x}$, we can write (9) in the form:

$$\hat{y}_{KI}(\boldsymbol{x}) = \mu + \frac{\boldsymbol{r}(\boldsymbol{x})'}{s(\boldsymbol{x})}\boldsymbol{R}^{-1}\boldsymbol{S}\left(\boldsymbol{y}-\mu\boldsymbol{1}\right), \tag{15}$$

where $\boldsymbol{S} = \text{diag}\{\boldsymbol{Rc}\}$. Comparison of (15) with the posterior mean in (13) motivates us to propose the following GP model for KI:

$$s(\boldsymbol{x})\{y(\boldsymbol{x})-\mu\} = Z(\boldsymbol{x}), \quad \text{where } Z(\boldsymbol{x}) \sim GP(0,\tau^2 K(\cdot,\cdot)). \tag{16}$$

Here, $\text{var}\{y(\boldsymbol{x})|\mu,\boldsymbol{c},\boldsymbol{\theta},\tau^2\} = \tau^2/s(\boldsymbol{x})^2$ is no longer a constant with respect to $\boldsymbol{x}$, which makes it a nonstationary GP model. This is quite different from the stationary GP model for kriging. The joint distribution of $(y(\boldsymbol{x}),\boldsymbol{y}')'$ given $\mu,\tau^2,\boldsymbol{\theta},\boldsymbol{c}$ is $N(\mu\boldsymbol{1}_{n+1},\boldsymbol{\Sigma})$, where

$$\boldsymbol{\Sigma} = \tau^2\begin{bmatrix} s(\boldsymbol{x})^{-2}, & \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{S}^{-1}s(\boldsymbol{x})^{-1} \\ s(\boldsymbol{x})^{-1}\boldsymbol{S}^{-1}\boldsymbol{r}(\boldsymbol{x}), & \boldsymbol{S}^{-1}\boldsymbol{R}\boldsymbol{S}^{-1}\end{bmatrix}.$$

It can be easily shown that the posterior distribution of $y(\boldsymbol{x})$ given the data is

$$y(\boldsymbol{x})|\boldsymbol{y},\mu,\tau^2,\boldsymbol{\theta},\boldsymbol{c} \sim N\left(\frac{\boldsymbol{r}(\boldsymbol{x})'}{s(\boldsymbol{x})}\boldsymbol{R}^{-1}\boldsymbol{S}\boldsymbol{y}, \frac{\tau^2}{s^2(\boldsymbol{x})}\left\{1-\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x})\right\}\right). \tag{17}$$

Since the right side of (17) is independent of μ, the unconditional distribution of $y(\boldsymbol{x})|\boldsymbol{y},\tau^2,\boldsymbol{\theta},\boldsymbol{c}$ is the same as the normal distribution in (17). Also note that $\boldsymbol{r}(\boldsymbol{x}_i)'\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}_i) = 1$ for $i = 1,2,\ldots,n$. Thus the posterior variance becomes 0 at all of the observed locations.

A $(1-\alpha)$ confidence interval for the prediction at any $\boldsymbol{x}$ can be obtained as

$$\hat{y}_{KI}(\boldsymbol{x}) \pm z_{\alpha/2}\frac{\tau}{s(\boldsymbol{x})}\{1-\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x})\}^{1/2}. \tag{18}$$

It is plotted as a shaded region in Figure 3 for the test function $y(x) = 0.5 - \sin(2x) - \exp(-2x)$, where the unknown parameters are chosen by empirical Bayes methods as discussed in the next section. The confidence interval for the ordinary kriging is given by $\hat{y}(\boldsymbol{x})_{OK} \pm z_{\alpha/2}\tau_{OK}\{1-\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) + (1-\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{1})^2/\boldsymbol{1}'\boldsymbol{R}^{-1}\boldsymbol{1}\}^{1/2}$, which is also plotted in the same figure as a dotted line. We can see that the confidence interval for KI is narrower in the middle than that of OK and it becomes wider at both ends. This is because $s(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{c}$ decreases as $\boldsymbol{x}$ is moved away from the observed locations, whereas the width of the OK confidence interval converges to a constant $2z_{\alpha/2}\tau_{OK}$. Overall, the KI confidence interval looks narrower than that of OK, but it could be because of not incorporating the uncertainties in $\boldsymbol{c}$. Unfortunately, it is not easy to account for the uncertainties in $\boldsymbol{c}$ because of the high-dimensionality and nonlinearity of the parameters.

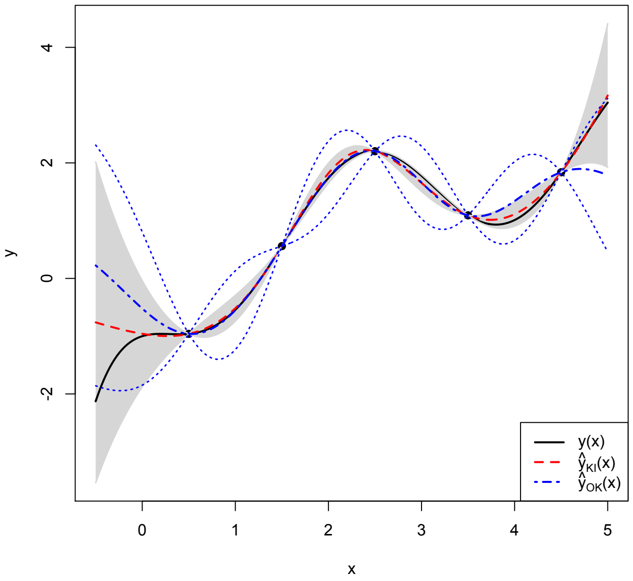

Figure 3: Confidence intervals for KI and OK predictions for the test function $y(x) = 0.5x - \sin(2x) - \exp(-2x)$ with $\hat{\theta}_{OK} = 0.97$, $\hat{\theta}_{KI} = 0.34$, and $\hat{\boldsymbol{c}} = (0.86, 0.0, 0.55, 0.04, 0.83)'$.

4 Empirical Bayes Estimation

Under (16), the marginal distribution of the responses is $\boldsymbol{y} \sim N(\mu\mathbf{1}, \tau^2\boldsymbol{S}^{-1}\boldsymbol{R}\boldsymbol{S}^{-1})$. Thus the likelihood (omitting the proportionality constant) is

$$L = \frac{1}{\tau^n|\boldsymbol{S}^{-1}\boldsymbol{R}\boldsymbol{S}^{-1}|^{1/2}} \exp\left(-\frac{1}{2\tau^2}(\boldsymbol{y}-\mu\mathbf{1})'\boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}(\boldsymbol{y}-\mu\mathbf{1})\right),$$

which gives

$$-2\log L = n\log\tau^2 + \log|\boldsymbol{R}| - 2\log|\boldsymbol{S}| + \frac{1}{\tau^2}(\boldsymbol{y}-\mu\mathbf{1})'\boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}(\boldsymbol{y}-\mu\mathbf{1}). \tag{19}$$

The empirical Bayes estimates of μ, τ^2, $\boldsymbol{c}$, and $\boldsymbol{\theta}$ are the solution of

$$\min_{\mu,\tau^2,\boldsymbol{c},\boldsymbol{\theta}} -2\log L, \qquad \text{subject to } \boldsymbol{\theta} \geq \mathbf{0}, \boldsymbol{c} > \mathbf{0}.$$

This is a nonlinear optimization problem in $n+p+2$ dimensions, which can be hard to solve for large n and/or p. We can simplify the optimization procedure as follows.

First, differentiating $-2\log L$ with respect to τ^2 and equating to 0, we obtain

$$\hat{\tau}^2 = \frac{1}{n}(\boldsymbol{y}-\mu\mathbf{1})'\boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}(\boldsymbol{y}-\mu\mathbf{1}). \tag{20}$$

Substituting for τ^2, $-2\log L$ becomes

$$-2\log L = n\log\left\{\frac{1}{n}(\boldsymbol{y}-\mu\mathbf{1})'\boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}(\boldsymbol{y}-\mu\mathbf{1})\right\} + \log|\boldsymbol{R}| - 2\log|\boldsymbol{S}|.$$

Thus,

$$L^{-2/n} \propto \frac{(\boldsymbol{y}-\mu\mathbf{1})'\boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}(\boldsymbol{y}-\mu\mathbf{1})}{\left(\prod_{i=1}^n \boldsymbol{r}(\boldsymbol{x}_i)'\boldsymbol{c}\right)^{2/n}}|\boldsymbol{R}|^{1/n}. \tag{21}$$

Now consider minimizing this expression with respect to $\boldsymbol{c}$. This is again not an easy problem because $\boldsymbol{c}$ appears in the numerator and denominator of (21). However, the numerator can be written as

$$Q = (\boldsymbol{y}-\mu\mathbf{1})'\boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}(\boldsymbol{y}-\mu\mathbf{1}) = \boldsymbol{c}'\boldsymbol{P}\boldsymbol{c},$$

where $\boldsymbol{P} = \boldsymbol{R}\text{diag}\{\boldsymbol{y}-\mu\mathbf{1}\}\boldsymbol{R}^{-1}\text{diag}\{\boldsymbol{y}-\mu\mathbf{1}\}\boldsymbol{R}$. Moreover,

$$\frac{Q}{\left(\prod_{i=1}^n \boldsymbol{r}(\boldsymbol{x}_i)'\boldsymbol{c}\right)^{2/n}}|\boldsymbol{R}|^{1/n} \leq Q|\boldsymbol{R}|^{1/n}$$

if $\boldsymbol{r}(\boldsymbol{x}_i)'\boldsymbol{c} \geq 1$ for $i = 1, \ldots, n$. Thus, an approximate solution of $\boldsymbol{c}$ can be obtain by solving the quadratic programming problem:

$$\min_{\boldsymbol{c}} Q = \boldsymbol{c}'\boldsymbol{P}\boldsymbol{c}, \qquad \text{s.t.} \quad \boldsymbol{R}\boldsymbol{c} \geq \boldsymbol{1}, \quad \boldsymbol{c} > \boldsymbol{0}. \tag{22}$$

The foregoing simplification brings tremendous reduction in computation, because quadratic programming is polynomial time solvable. In the quadratic programming, we specify a very small lower bound $\nu > 0$ for c_i $i = 1, \ldots, n$, so that the constraint $\boldsymbol{c} > \boldsymbol{0}$ can be replaced by $\boldsymbol{c} \geq \nu\boldsymbol{1}$. Note that since $\boldsymbol{R}$ is positive definite, $\boldsymbol{P}$ is also positive definite assuming that $\boldsymbol{y} \neq \mu\boldsymbol{1}$. Furthermore, a small nugget is added to the diagonal of $\boldsymbol{R}$ to avoid any ill-conditioning problem of the $\boldsymbol{R}$ matrix [19].

Now minimizing $-2\log L$ with respect to μ, we obtain

$$\hat{\mu} = \frac{\boldsymbol{1}'\boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}\boldsymbol{y}}{\boldsymbol{1}'\boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}\boldsymbol{1}} = \frac{\hat{\boldsymbol{c}}'\boldsymbol{S}\boldsymbol{y}}{\hat{\boldsymbol{c}}'\boldsymbol{R}\hat{\boldsymbol{c}}}. \tag{23}$$

Because $\hat{\boldsymbol{c}}$ depends on the value of μ, this needs to be solved iteratively. We start with an initial value of $\mu = \bar{y}$ and then iterate between (22) and (23) until convergence.

For any given value of the correlation parameters $\boldsymbol{\theta}$, the foregoing description shows how to obtain the optimal τ^2, μ, and $\boldsymbol{c}$. Substituting them in (19), we obtain $-2\log L$ as a function of only $\boldsymbol{\theta}$. Thus, the optimal $\boldsymbol{\theta}$ can be obtained as the solution of

$$\min_{\boldsymbol{\theta} \geq \boldsymbol{0}} n \log \hat{\tau}^2 + \log|\boldsymbol{R}| - 2\log|\boldsymbol{S}|. \tag{24}$$

We use iterative procedures such as the classical L-BFGS-B algorithm to solve this p-dimensional nonlinear optimization problem. Multiple random starts are used in the optimization algorithm for finding the global optimum. In each iteration, evaluation of the objective function involves estimation of τ^2, μ and $\boldsymbol{c}$ as illustrated above. Thus, the optimization procedure is still computationally very intensive, but much simpler than the initial $(n + p + 2)$-dimensional optimization problem.

5 KI vs OK

Consider again the example presented in Section 2. Figure 4 shows the OK and KI predictors with a large correlation parameter value ($\theta_{OK} = \theta_{KI} = 40$). We can see that the OK

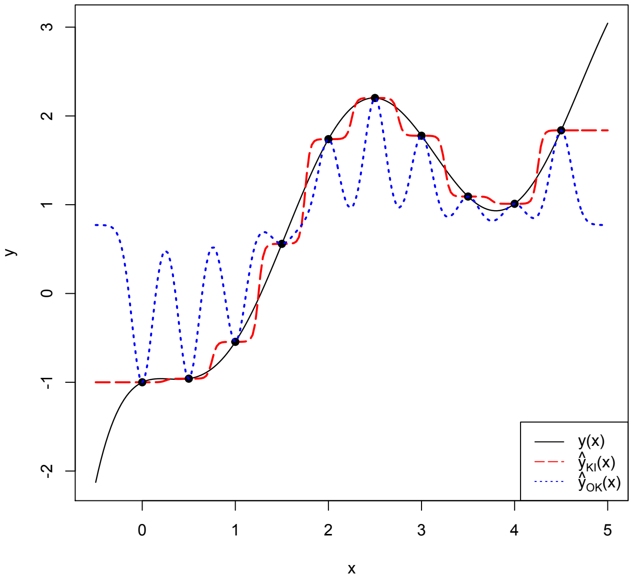

Figure 4: Comparison of KI and OK for a large θ ($\theta_{OK} = \theta_{KI} = 40$).

predictor is pulled towards the mean, whereas the KI predictor becomes almost a step function. Overall, the deterioration in the prediction performance is much larger for OK than KI. This indicates robustness of the KI predictor against misspecification of the correlation parameters. The behaviors of the two predictors are shown in more general terms in the following proposition.

Proposition 3. *Assume the isotropic power exponential correlation function of the form*

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\theta d(\boldsymbol{x}_i, \boldsymbol{x}_j))$$

for both the kernel interpolation and the ordinary kriging predictor. Here $\theta > 0$ *and* $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{l=1}^{p} |x_{i,l} - x_{j,l}|^{\alpha_l}$*, with* $0 < \alpha_l \leq 2$ *for* $l = 1, \ldots, p$*. The following results are true.*

(i) For kernel interpolation,

$$\lim_{\theta \to \infty} \hat{y}_{KI}(\boldsymbol{x}) = \frac{\sum_{i \in \xi} c_i y_i}{\sum_{i \in \xi} c_i},$$

where $\xi = \{i : d(\boldsymbol{x}, \boldsymbol{x}_i) = \delta$ *where* $\delta \triangleq \min_j d(\boldsymbol{x}, \boldsymbol{x}_j)\}$*.*

(ii) For ordinary kriging,

$$\lim_{\theta\to\infty} \hat{y}_{OK}(\boldsymbol{x}) = \begin{cases} \hat{\mu}, & \boldsymbol{x} \neq \boldsymbol{x}_j, \quad j = 1, \ldots, n, \\ y_j, & \boldsymbol{x} = \boldsymbol{x}_j, \quad j = 1, \ldots, n. \end{cases}$$

Here δ is the minimum distance between $\boldsymbol{x}$ and all the design points $\{\boldsymbol{x}_i\}_{i=1}^n$, and ξ is the index set of the points whose distance to $\boldsymbol{x}$ is equal to the minimum distance. When there is only one design point (which is often the case), which has the minimum distance to $\boldsymbol{x}$, we obtain

$$\lim_{\theta\to\infty} \hat{y}_{KI}(\boldsymbol{x}) = y_i, \quad \text{where } i = arg \min_j d(\boldsymbol{x}, \boldsymbol{x}_j).$$

Thus, the KI prediction converges to the nearest observation, whereas the OK prediction converges to the mean value (for $\boldsymbol{x} \neq \boldsymbol{x}_i$). This property of avoiding reversion to mean is also captured by the fractional Brownian fields, another non-stationary Gaussian process recently proposed by [27]. Thus, KI is expected to perform better than OK for the class of functions where the nearest neighbor gives a better prediction than using the overall mean. For example, this holds true for the class of monotone functions.

Although the foregoing asymptotic result gives some insights on the class of functions in which KI is expected to perform better than OK, we cannot use it to choose between KI and OK for a given function. One option to decide between KI and OK is to use cross validation. Let $\hat{y}_{-k}^{OK}(\boldsymbol{x})$ and $\hat{y}_{-k}^{KI}(\boldsymbol{x})$ be the OK and KI predictors estimated without using the point $(\boldsymbol{x}_k, y_k)$ respectively. Neglecting the changes in $\hat{\boldsymbol{\theta}}_{OK}$, the leave-one-out cross validation error for OK can be easily shown to be

$$y_k - \hat{y}_{-k}^{OK}(\boldsymbol{x}_k) = \frac{1}{(\boldsymbol{R}^{-1})_{k,k}} \left(\boldsymbol{R}^{-1}(\boldsymbol{y} - \hat{\mu}_{-k}\mathbf{1}_{n-1})\right)_k,$$

where

$$\hat{\mu}_{-k} = \frac{\mathbf{1}\boldsymbol{R}^{-1}\boldsymbol{y} - (\boldsymbol{R}^{-1}\boldsymbol{y})_k/(\boldsymbol{R}^{-1})_{k,k}}{\mathbf{1}\boldsymbol{R}^{-1}\mathbf{1} - (\boldsymbol{R}^{-1}\mathbf{1})_k/(\boldsymbol{R}^{-1})_{k,k}}.$$

Similarly, when computing the leave-one-out cross validation error for KI, we use the same $\hat{\boldsymbol{\theta}}_{KI}$ estimated from the complete data. However, the coefficients $\boldsymbol{c}$ should be re-estimated for each k, following the same iterative estimation procedure described in Section 4.

6 Examples

Example 1. *Borehole Example ([16]):*

The flow rate (y) through a borehole is given by

$$y = \frac{2\pi T_u(H_u - H_l)}{\ln(r/r_w)\left[1 + \frac{2LT_u}{\ln(r/r_w)r_w^2K_w} + \frac{T_u}{T_l}\right]},$$

where the ranges of interest for the eight variables are: $r_w \in [0.05, 0.15]$, $r \in [100, 50000]$, $T_u \in [63070, 115600]$, $H_u \in [990, 1110]$, $T_l \in [63.1, 116]$, $H_l \in [700, 820]$, $L \in [1120, 1680]$, and $K_w \in [9855, 12045]$. Denote them by x_1 through x_8, respectively.

We generated a maximin Latin hypercube design (LHD) ([15]) with 80 runs (10 times the number of variables) and fitted KI. Since the true function is known in this example, we can compute its prediction errors. For this purpose we generated a 1000-run random LHD from the same design region and computed the root mean-squared prediction error (RMSPE). Two other interpolating methods (OK and limit kriging) and four nonparametric regression methods treed GP ([5]), local linear regression, multivariate additive regression splines (MARS), and neural network ([21]) were chosen for comparison. They are fitted using the R packages *tgp* ([6]), *np* ([9]), *mda* ([8]), and *nnet* ([25]) using the same 80-run training dataset. The RMSPE's of the seven methods are given in Table 1. In general, the interpolation methods give superior performance over the four nonparametric regression methods, which is expected because of the deterministic nature of the data. Among the interpolation methods, KI gives the smallest RMSPE, which is about one-third of the two kriging methods. This excellent performance of KI comes with a price: the CPU time for fitting KI was 96 seconds which is about 40 times more than that of the kriging methods. However, when approximating computationally expensive functions, it is worth spending a few extra minutes to obtain better prediction.

We were able to compute the RMSPE in this example because the function was known. In real applications where the function is expensive to compute, we should compute cross validation errors for choosing a method. As explained in the previous section, the root mean-squared cross validation error for OK can be computed as 0.8708 and that of KI as 0.2474. This shows that KI is expected to work significantly better than OK in this example.

Table 1: RMSPE's of different predictors for Examples 1 and 2.

Example	Interpolation			Nonparametric Regression			
	KI	OK	Limit Kriging	TGP	Local Linear Regression	MARS	Neural Network
Borehole	0.2158	0.6478	0.6091	7.9725	2.3978	5.1961	3.5684
Cyclone	0.0197	0.0360	0.0370	0.1114	0.0723	0.1477	0.0407

To exclude the effects of experimental design, we generated 100 more training datasets of size $n = 80$ using random LHDs and fitted both KI and OK. Figure 5 compares the RMSPE's of the two methods on the same test dataset of size 1000. All the points fall above the $y = x$ line indicating that OK has larger prediction errors than KI.

Example 2. *Cyclone Example ([14] and [12]):*

A cyclone is used for separating solid mass and gaseous mass. The critical parameter of particles (y) is related to seven input variables by

$$y = 174.42 \left(\frac{x_1}{x_5}\right) \left(\frac{x_3}{x_2 - x_1}\right)^{0.85} \left(\frac{1 - 2.62[1 - 0.36(x_4/x_2)^{-0.56}]^{3/2}(x_4/x_2)^{1.16}}{x_6 x_7}\right)^{1/2}.$$

The ranges of the variables are $x_1 \in [0.075, 0.125]$, $x_2 \in [0.225, 0.375]$, $x_3 \in [0.075, 0.125]$, $x_4 \in [0.075, 0.125]$, $x_5 \in [1.125, 1.875]$, $x_6 \in [12.0, 20.0]$, and $x_7 \in [0.5625, 0.9375]$.

The comparisons of KI with the other methods are performed in the same way as in the Borehole example, except that the size of the training dataset is $n = 70$. The second row of Table 1 shows the RMSPE's of the seven different methods. The KI predictor again gives the smallest RMSPE, which is about half of the other kriging methods. The performance of the nonparametric regression methods is relatively better in this example, but still the best performing method (neural network) has three times larger RMSPE than that of KI. Figure 6 shows the RMSPE's of the KI and OK predictors fitted on 100 training datasets generated using random LHDs. We can clearly see that KI is much more accurate than the OK predictor for approximating the cyclone function.

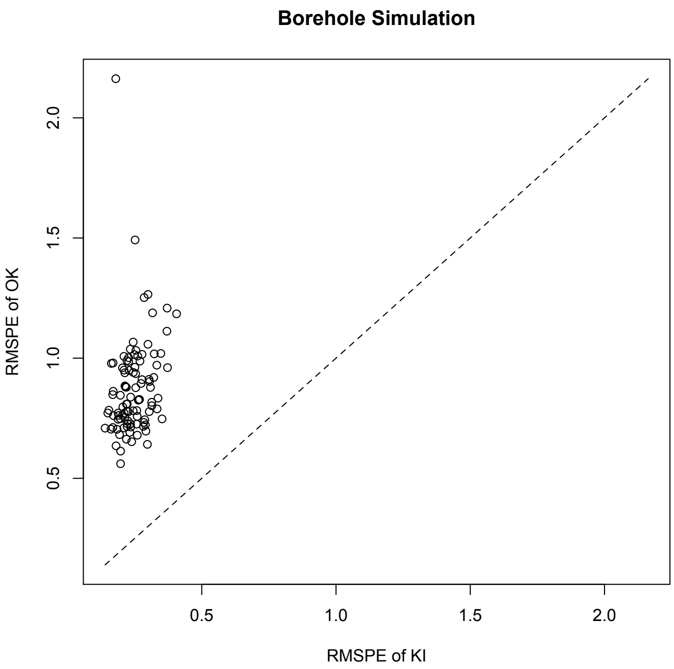

Figure 5: RMSPEs of KI and OK in the Borehole example for 100 datasets.

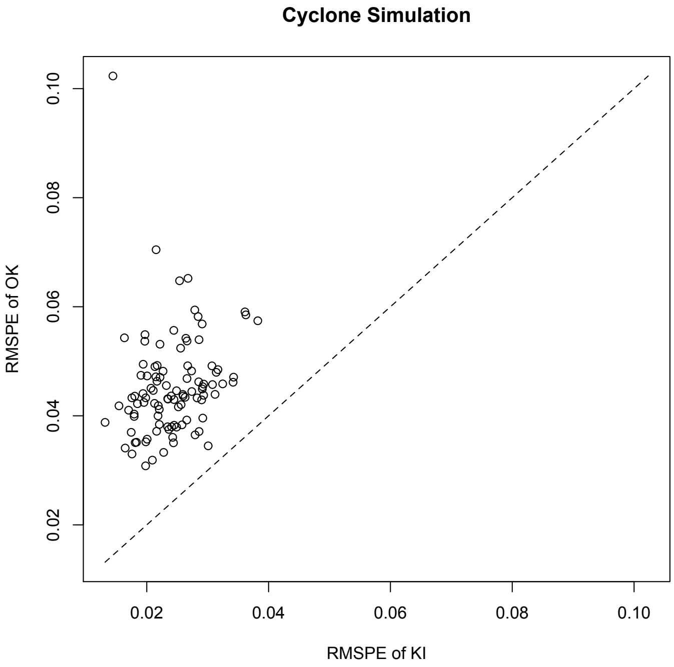

Figure 6: RMSPEs of KI and OK in the Cyclone example for 100 datasets.

7 Extension to Noisy Data

In this section, we extend the KI method to deal with data having random noise. Consider the model in (1) with

$$f(\boldsymbol{x}) = \mu + Z(\boldsymbol{x})/s(\boldsymbol{x}).$$

For the noisy data, replicates of observations are necessary to estimate the noise variance. If we have n_i replicates at the design point $\boldsymbol{x}_i$ for $i = 1, \ldots, m$, then we can estimate the noise variance as $\hat{\sigma}^2 = \sum_{i=1}^m (n_i - 1)s_i^2/(n-m)$, where $n = \sum_{i=1}^m n_i$ and s_i^2 is the sample variance of the responses at $\boldsymbol{x}_i$. Denote the vector $\boldsymbol{y} = (\bar{y}_1, \ldots, \bar{y}_m)$. Its sampling distribution is

$$\boldsymbol{y}|\mu, \tau^2, \sigma^2, \boldsymbol{c}, \boldsymbol{\theta} \sim N\left(\mu\boldsymbol{1}, \tau^2 \boldsymbol{S}^{-1}\boldsymbol{R}\boldsymbol{S}^{-1} + \sigma^2 \boldsymbol{L}\right),$$

where $\boldsymbol{L} = \operatorname{diag}\{1/n_1, \ldots, 1/n_m\}$. Assuming a noninformative prior for μ as in Section 3, the posterior distribution of $y(\boldsymbol{x})$ can be obtained to be normal with mean

$$E\left\{y(\boldsymbol{x})|\tau^2, \sigma^2, \boldsymbol{c}, \boldsymbol{\theta}, \boldsymbol{y}\right\} = \hat{\mu} + \frac{\boldsymbol{r}(\boldsymbol{x})'}{s(\boldsymbol{x})}\left(\boldsymbol{S}^{-1}\boldsymbol{R} + \frac{\sigma^2}{\tau^2}\boldsymbol{L}\boldsymbol{S}\right)^{-1}(\boldsymbol{y} - \hat{\mu}\boldsymbol{1}) \tag{25}$$

and variance

$$\begin{aligned} \operatorname{var}\left(y(\boldsymbol{x})|\tau^2, \sigma^2, \boldsymbol{c}, \boldsymbol{\theta}, \boldsymbol{y}\right) &= \frac{\tau^2}{s^2(\boldsymbol{x})}\left\{1 - \boldsymbol{r}(\boldsymbol{x})'\left(\boldsymbol{R} + \frac{\sigma^2}{\tau^2}\boldsymbol{L}\boldsymbol{S}^2\right)^{-1}\boldsymbol{r}(\boldsymbol{x})\right\} \\ &+ \tau^2\left\{\left(1 - \frac{\boldsymbol{r}(\boldsymbol{x})'}{s(\boldsymbol{x})}\left(\boldsymbol{S}^{-1}\boldsymbol{R} + \frac{\sigma^2}{\tau^2}\boldsymbol{L}\boldsymbol{S}\right)^{-1}\boldsymbol{1}\right)^2 \frac{1}{\boldsymbol{1}'\left(\boldsymbol{S}^{-1}\boldsymbol{R}\boldsymbol{S}^{-1} + \frac{\sigma^2}{\tau^2}\boldsymbol{L}\right)^{-1}\boldsymbol{1}}\right\} + \sigma^2, \end{aligned}$$

where

$$\hat{\mu} = \frac{\boldsymbol{1}'\left(\boldsymbol{S}^{-1}\boldsymbol{R}\boldsymbol{S}^{-1} + \frac{\sigma^2}{\tau^2}\boldsymbol{L}\right)^{-1}\boldsymbol{y}}{\boldsymbol{1}'\left(\boldsymbol{S}^{-1}\boldsymbol{R}\boldsymbol{S}^{-1} + \frac{\sigma^2}{\tau^2}\boldsymbol{L}\right)^{-1}\boldsymbol{1}}.$$

The posterior mean of $y(\boldsymbol{x})$ in (25) no longer interpolates the data, and therefore, we call it *kernel approximator* (KA).

The empirical Bayes estimation is slightly different when $\sigma^2 > 0$ from Section 4. Let $\gamma = \hat{\sigma}^2/\tau^2$. Then the likelihood L can be derived similarly as in (21):

$$L^{-2/n} \propto \frac{(\boldsymbol{y} - \mu\boldsymbol{1})'\boldsymbol{S}\boldsymbol{V}^{-1}\boldsymbol{S}(\boldsymbol{y} - \mu\boldsymbol{1})}{\left(\prod_{i=1}^n \boldsymbol{r}(\boldsymbol{x}_i)'\boldsymbol{c}\right)^{2/n}}|\boldsymbol{V}|^{1/n},$$

where $\boldsymbol{V} = \boldsymbol{R} + \gamma\boldsymbol{L}\boldsymbol{S}^2$. Similar to Section 4 we also need to solve $\boldsymbol{c}$ and μ iteratively. Since $\boldsymbol{V}$ involves $\boldsymbol{c}$, we approximate $\boldsymbol{V}$ by using the value of $\boldsymbol{c}$ obtained in the previous iteration.

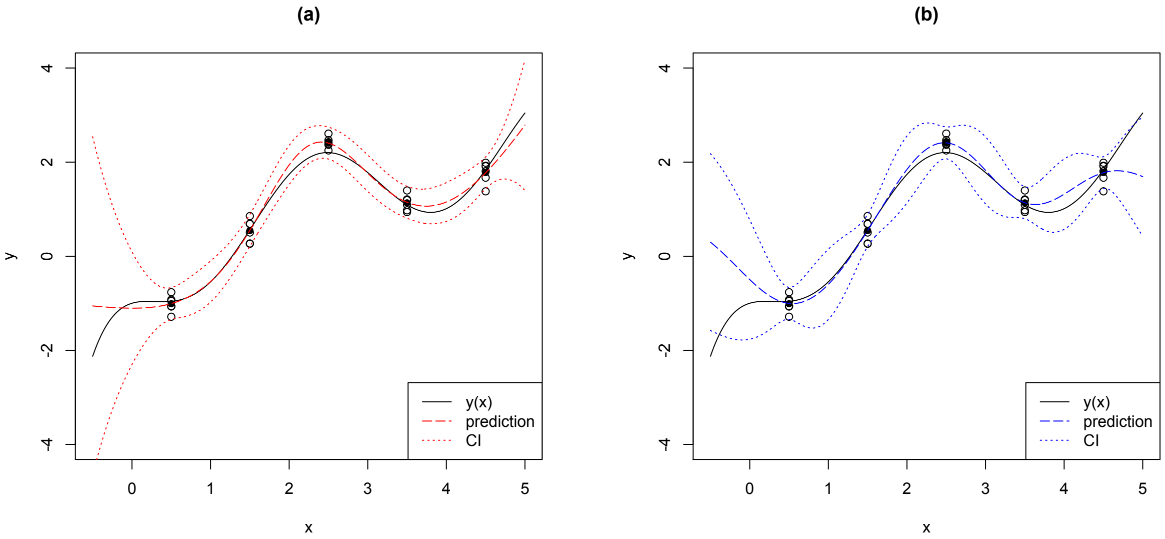

Figure 7: Prediction and confidence intervals with noisy data. (a) KA and (b) OK.

Specifically, for given $\boldsymbol{\theta}$ value, set the initial value $\mu_0 = \bar{y}$ and $\boldsymbol{c}_0 = \mathbf{1}$. It is necessary to specify the initial value for $\boldsymbol{c}$ because of $\boldsymbol{V}$, whereas in Section 4 we only need to specify μ_0. Then iteratively update $\boldsymbol{c}$ and μ as follows until convergence.

$$\boldsymbol{c}_i = arg \min_{\boldsymbol{c}} \boldsymbol{c}'\boldsymbol{P}_{i-1}\boldsymbol{c}, \quad \text{s.t} \quad \boldsymbol{Rc} \geq \mathbf{1}, \quad \boldsymbol{c} > \mathbf{0},$$
$$\text{and } \hat{\mu}_i = \frac{\mathbf{1}'\boldsymbol{S}_i\boldsymbol{V}_i^{-1}\boldsymbol{S}_i\boldsymbol{y}}{\mathbf{1}'\boldsymbol{S}_i\boldsymbol{V}_i^{-1}\boldsymbol{S}_i\mathbf{1}},$$

where $\boldsymbol{P}_{i-1} = \boldsymbol{R}\text{diag}\{\boldsymbol{y} - \hat{\mu}_{i-1}\mathbf{1}\}\boldsymbol{V}_{i-1}^{-1}\text{diag}\{\boldsymbol{y} - \hat{\mu}_{i-1}\mathbf{1}\}\boldsymbol{R}$, $\boldsymbol{V}_i = \boldsymbol{R} + \gamma\boldsymbol{L}\boldsymbol{S}_i^2$, and $\boldsymbol{S}_i = \text{diag}\{\boldsymbol{Rc}_i\}$. In each iteration, γ is also updated along with μ and $\boldsymbol{c}$. To start the iterations, we choose a reasonable initial value for γ by setting $\tau^2 = \sum_{i=1}^m n_i(\bar{y}_i - \bar{y})^2/m$.

For illustration, we chose five equally spaced design points and generated 6 replications from the model $y_{ij} = 0.5x_i - \sin(2x_i) - \exp(-2x_i) + \epsilon_{ij}$, where $\epsilon_{ij} \sim^{iid} N(0, .2^2)$. Figure 7 shows the prediction and confidence intervals of KI and OK. We can see that the KI continues to perform better near the boundaries for this function, however, there is no clear improvement in the middle region.

8 Conclusions

In this paper, we proposed an interpolator for analyzing deterministic computer experiments, called kernel interpolation. It is developed by iteratively applying the kernel regression on the residuals infinite number of times. We also proposed a nonstationary GP model for KI, which helped in estimating the unknown parameters, constructing confidence intervals, and extending the method to noisy observations. Compared with the popular kriging method, KI seems to be more robust to the choice of correlation parameters and better in approximating functions having monotonic global trends.

The main disadvantage of KI is its high computational cost for estimation of the unknown parameters compared to kriging. However, the prediction using KI is as fast as that of kriging. Therefore, the disadvantage in estimation diminishes as the function becomes more and more expensive to evaluate. However, faster computational tools need to be developed for dealing with large datasets and high dimensional problems.

Acknowledgments

The research was supported in part by U.S. National Science Foundation grant CMMI-1030125 and a grant from the U. S. Army Research Laboratory and the U. S. Army Research Office under contract number W911NF-14-1-0024. We thank the Associate Editor and two reviewers for their valuable comments and suggestions.

Appendix: Proofs

Proof of Theorem 1.

Proof. To show that

$$\hat{y}_{KI}(\boldsymbol{x}) = \lim_{N\to\infty} \hat{y}_N(\boldsymbol{x}) = \mu + \boldsymbol{u}(\boldsymbol{x})' \sum_{i=0}^{\infty} (\boldsymbol{I}_n - \boldsymbol{U})^i (\boldsymbol{y} - \mu\boldsymbol{1})$$

interpolates all the observations, we only need to show for any $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{U}\sum_{i=0}^{\infty}(\boldsymbol{I}_n - \boldsymbol{U})^i(\boldsymbol{y} - \mu\boldsymbol{1}) = \boldsymbol{y} - \mu\boldsymbol{1}$, or equivalently, $\sum_{i=0}^{\infty}(\boldsymbol{I}_n - \boldsymbol{U})^i = \boldsymbol{U}^{-1}$. This convergence is true if and only if $\rho(\boldsymbol{I}_n - \boldsymbol{U}) < 1$, where $\rho(\cdot)$ denotes the spectral radius of a matrix. Because

$$\rho(\boldsymbol{I}_n - \boldsymbol{U}) = \max_j |\lambda_j(\boldsymbol{I}_n - \boldsymbol{U})| = \max_j |1 - \lambda_j(\boldsymbol{U})| < 1,$$

the condition $\rho(\boldsymbol{I}_n - \boldsymbol{U}) < 1$ is equivalent to $0 < \lambda_j(\boldsymbol{U}) < 2$ for $j = 1, \dots, n$. The matrix $\boldsymbol{R}$ is always symmetric positive definite since the kernel function is symmetric positive definite. The entries of the diagonal matrix $\boldsymbol{S}$ are all strictly positive. Since both $\boldsymbol{S}$ and $\boldsymbol{R}$ are symmetric positive definite matrices, $\boldsymbol{U} = \boldsymbol{S}^{-1}\boldsymbol{R}$ is weakly positive definite. Therefore, $\lambda_i(\boldsymbol{U}) > 0$ for $i = 1, \dots, n$ ([18]). Thus, $\hat{y}_N(\boldsymbol{x}) \to \hat{y}_{KI}(\boldsymbol{x})$ as $N \to \infty$, if and only if $\lambda_i(\boldsymbol{U}) < 2$ for $i = 1, \dots, n$.

Furthermore, since

$$\boldsymbol{u}(\boldsymbol{x})'\boldsymbol{U}^{-1}\boldsymbol{1} = \frac{\boldsymbol{r}(\boldsymbol{x})'}{s(\boldsymbol{x})}\boldsymbol{R}^{-1}\text{diag}\{\boldsymbol{Rc}\}\boldsymbol{1} = \frac{\boldsymbol{r}(\boldsymbol{x})'}{\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{c}}\boldsymbol{R}^{-1}\boldsymbol{Rc} = 1$$

for all $\boldsymbol{x}$,

$$\hat{y}(\boldsymbol{x}) = \mu + \boldsymbol{u}(\boldsymbol{x})'\boldsymbol{U}^{-1}(\boldsymbol{y} - \mu\boldsymbol{1}) = \boldsymbol{u}(\boldsymbol{x})'\boldsymbol{U}^{-1}\boldsymbol{y}.$$

□

Proof of Proposition 2.

Proof. To show that $\hat{y}_N(\boldsymbol{x})$ converges to an interpolator, we need to show $||\boldsymbol{y} - \hat{\boldsymbol{y}}_N|| \to 0$ as $N \to \infty$ for any matrix norm. We have

$$\begin{aligned}\boldsymbol{y} - \lim_{N\to\infty}\hat{\boldsymbol{y}}_N &= \left(\boldsymbol{I}_n - \sum_{i=1}^{\infty}\boldsymbol{U}_i\prod_{j=0}^{i-1}(\boldsymbol{I}_n - \boldsymbol{U}_j)\right)(\boldsymbol{y} - \mu\boldsymbol{1}) \\ &= \left(\prod_{i=1}^{\infty}(\boldsymbol{I}_n - \boldsymbol{U}_i)\right)(\boldsymbol{y} - \mu\boldsymbol{1}).\end{aligned}$$

$$||\boldsymbol{y} - \lim_{N\to\infty}\hat{\boldsymbol{y}}_N|| \leq \prod_{i=1}^{\infty}||\boldsymbol{I}_n - \boldsymbol{U}_i|| \cdot ||\boldsymbol{y} - \mu\boldsymbol{1}||.$$

If $||\boldsymbol{I}_n - \boldsymbol{U}_i|| < 1$ for all i, then $\prod_{i=1}^{\infty}||\boldsymbol{I}_n - \boldsymbol{U}_i|| = 0$. Therefore, $\boldsymbol{y} = \lim_{N\to\infty}\hat{\boldsymbol{y}}_N$. □

Proof of Proposition 3.

Proof. The KI prediction form is:

$$\hat{y}_{KI}(\boldsymbol{x}) = \frac{\boldsymbol{r}(\boldsymbol{x})'}{\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{c}}\boldsymbol{R}^{-1}\boldsymbol{Sy}.$$

The jth element of the vector $\boldsymbol{r}(\boldsymbol{x})/\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{c}$ is

$$u_j(\boldsymbol{x}) = \frac{\exp\{-\theta d(\boldsymbol{x}, \boldsymbol{x}_j)\}}{\sum_m c_m \exp\{-\theta d(\boldsymbol{x}, \boldsymbol{x}_m)\}}.$$

Define the notations δ and ξ as in the Proposition 3. Then

$$u_j(\boldsymbol{x}) = \frac{\exp\{-\theta(d(\boldsymbol{x}, \boldsymbol{x}_j) - \delta)\}}{\sum_m c_m \exp\{-\theta(d(\boldsymbol{x}, \boldsymbol{x}_m) - \delta)\}}.$$

If $j \notin \xi$, then $d(\boldsymbol{x}, \boldsymbol{x}_j) > \delta$. As $\theta \to \infty$, we have $u_j(\boldsymbol{x}) \to 0$. Similarly, if $j \in \xi$, then $d(\boldsymbol{x}, \boldsymbol{x}_j) = \delta$, and thus $u_j(\boldsymbol{x}) \to 1/\sum_{i\in\xi} c_i$ as $\theta \to \infty$. Meanwhile, if $\theta \to \infty$, $K(\boldsymbol{x}_i, \boldsymbol{x}_j) \to 0$ for $i \neq j$, and thus $\boldsymbol{R}$ pointwise converges to the identity matrix $\boldsymbol{I}_n$. The vector $\boldsymbol{R}^{-1}\boldsymbol{S}\boldsymbol{y}$ pointwise converges to the vector diag$\{\boldsymbol{c}\}\boldsymbol{y}$. Therefore,

$$\frac{\boldsymbol{r}(\boldsymbol{x})'}{\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{c}} \boldsymbol{R}^{-1}\boldsymbol{S}\boldsymbol{y} \to \frac{\sum_{i\in\xi} c_i y_i}{\sum_{i\in\xi} c_i}, \quad \text{as } \theta \to \infty.$$

The result of OK can be proved similarly. □

References

[1] W.S. Cleveland, *Robust locally weighted regression and smoothing scatterplots*, Journal of the American statistical association, (1979), pp. 829–836.

[2] J. Fan and I. Gijbels, *Local Polynomial Modelling and Its Applications*, Chapman & Hall/CRC, London, 1996.

[3] K.T. Fang, R. Li, and A. Sudjianto, *Design and Modeling for Computer Experiments*, Chapman & Hall/CRC, Boca Raton, FL USA, 2006.

[4] G.E. Fasshauer and J.G. Zhang, *Iterated approximate moving least squares approximation*, in Advances in Meshfree Techniques, V. M. A. Leitao, C. Alves, and C. A. Duarte, eds., Springer, 2007, pp. 221–240.

[5] R.B. Gramacy and H.K.H. Lee, *Bayesian treed gaussian process models with an application to computer modeling*, Journal of the American Statistical Association, 103 (2008), pp. 1119–1130.

[6] Robert B. Gramacy, *tgp: An R package for bayesian nonstationary, semiparametric nonlinear regression and design by treed gaussian process models*, Journal of Statistical Software, 19 (2007), pp. 1–46.

[7] T. Hastie and C. Loader, *Local regression: Automatic kernel carpentry*, Statistical Science, 8 (1993), pp. 120–129.

[8] Trevor Hastie and Robert Tibshirani, *mda: Mixture and flexible discriminant analysis*, R package version 0.4-2, 2011.

[9] Tristen Hayfield and Jeffrey S. Racine, *Nonparametric econometrics: The np package*, Journal of Statistical Software, 27 (2008), pp. 1–32.

[10] V. R. Joseph, *Limit kriging*, Technometrics, 48 (2006), pp. 458–466.

[11] Q. Li and J.S. Racine, *Nonparametric Econometrics: Theory and Practice*, Princeton University Press, Princeton, NJ, 2007.

[12] W. Li and C. F. J. Wu, *An integrated method of parameter design and tolerance design*, Quality Engineering, 11 (1999), pp. 417–425.

[13] G. Matheron, *Le krigeage universel. Vol. 1*, Cahiers du Centre de Morphologie Mathematique, Ecole des Mines de Pairs, Fontainebleau, France, 1969.

[14] T Mori, *Case Studies in Experimental Design*, Management System Research Laboratory (in Japanese), Tokyo, 1985.

[15] M.D. Morris and T.J. Mitchell, *Exploratory designs for computational experiments*, Journal of Statistical Planning and Inference, 43 (1995), pp. 381–402.

[16] M.D. Morris, T.J. Mitchell, and D. Ylvisaker, *Bayesian design and analysis of computer experiments: Use of derivatives in surface prediction*, Technometrics, 35 (1993), pp. 243–255.

[17] E. A. Nadaraya, *On estimating regression*, Theory of Probability and Its Applications, 9 (1964), pp. 141–142.

[18] T.K. Nilssen, *Weakly positive definite matrices*, research report, Simula Research Laboratory, 2005.

[19] Chien-Yu Peng and CF Jeff Wu, *On the choice of nugget in kriging modeling for deterministic computer experiments*, Journal of Computational and Graphical Statistics, 23 (2014), pp. 151–168.

[20] Pritam Ranjan, Ronald Haynes, and Richard Karsten, *A computationally stable approach to gaussian process interpolation of deterministic computer simulation data*, Technometrics, 53 (2011), pp. 366–378.

[21] R. Rojas, *Neural Networks: A Systematic Introduction*, Springer, New York, 1996.

[22] D. Ruppert and MP Wand, *Multivariate locally weighted least squares regression*, The Annals of Statistics, 22 (1994), pp. 1346–1370.

[23] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn, *Design and analysis of computer experiments*, Statistical Science, 4 (1989), pp. 409–423.

[24] T.J. Santner, B.J. Williams, and W. Notz, *The Design and Analysis of Computer Experiments*, Springer, New York, 2003.

[25] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, Springer, New York, fourth ed., 2002.

[26] G.S. Watson, *Smooth regression analysis*, Sankhyā: The Indian Journal of Statistics, Series A, 26 (1964), pp. 359–372.

[27] N. Zhang and D. W. Apley, *Fractional brownian fields for response surface metamodeling*, Journal of Quality Technology, 46 (2014), pp. 285–301.