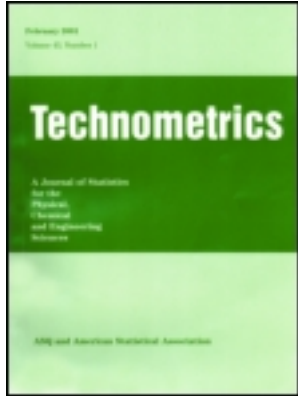


This article was downloaded by: [University of Wisconsin - Madison]

On: 21 December 2012, At: 07:29

Publisher: Taylor & Francis

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



Technometrics

Publication details, including instructions for authors and subscription information:
<http://amstat.tandfonline.com/loi/utch20>

A Simple Approach to Emulation for Computer Models With Qualitative and Quantitative Factors

Qiang Zhou, Peter Z. G. Qian and Shiyu Zhou

Department of Industrial and Systems Engineering, University of Wisconsin-Madison, Madison, WI 53706

Department of Statistics, University of Wisconsin-Madison, Madison, WI 53706

Department of Industrial and Systems Engineering, University of Wisconsin-Madison, Madison, WI 53706

Version of record first published: 24 Jan 2012.

To cite this article: Qiang Zhou, Peter Z. G. Qian and Shiyu Zhou (2011): A Simple Approach to Emulation for Computer Models With Qualitative and Quantitative Factors, Technometrics, 53:3, 266-273

To link to this article: <http://dx.doi.org/10.1198/TECH.2011.10025>

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: <http://amstat.tandfonline.com/page/terms-and-conditions>

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae, and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand, or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

A Simple Approach to Emulation for Computer Models With Qualitative and Quantitative Factors

Qiang ZHOU

Department of Industrial and Systems Engineering
University of Wisconsin–Madison
Madison, WI 53706
(qzhou3@wisc.edu)

Peter Z. G. QIAN

Department of Statistics
University of Wisconsin–Madison
Madison, WI 53706
(peterq@stat.wisc.edu)

Shiyu ZHOU

Department of Industrial and Systems Engineering
University of Wisconsin–Madison
Madison, WI 53706
(szhou@engr.wisc.edu)

We propose a flexible yet computationally efficient approach for building Gaussian process models for computer experiments with both qualitative and quantitative factors. This approach uses the hypersphere parameterization to model the correlations of the qualitative factors, thus avoiding the need of directly solving optimization problems with positive definite constraints. The effectiveness of the proposed method is successfully illustrated by several examples.

KEY WORDS: Computer experiment; Hypersphere decomposition; Kriging.

1. INTRODUCTION

Computer models are now ubiquitous in engineering and science. The standard statistical framework for the design and analysis of computer experiments assumes that all the factors are quantitative (Sacks, Schiller, and Welch 1989; Sacks et al. 1989; Currin et al. 1991; Santner, Williams, and Notz 2003; Fang, Li, and Sudjianto 2005). In many applications, however, computer models can contain both qualitative and quantitative factors. For example, computational fluid-dynamics programs in the IT industry for studying data center thermal dynamics can involve qualitative factors such as “air diffuser unit location,” “hot air return vent location,” and “power unit type” (Qian and Wu 2008). Rawlinson et al. (2006) and Han et al. (2009) reported knee models having qualitative factors such as “prosthesis design” and “force pattern” for investigating wear mechanisms of total knee replacements in biomechanical engineering. Furthermore, a set of multi-fidelity computer codes with quantitative factors (Kennedy and O’Hagan 2000; Qian et al. 2006; Qian and Wu 2008) can be treated collectively as a computer code with the same quantitative factors and a qualitative factor to describe different accuracy of the codes (Han et al. 2009).

Several methods are now available for building Gaussian process based emulators with qualitative and quantitative factors. Qian, Wu, and Wu (2008) proposed a general framework for building Gaussian process models with qualitative and quantitative factors. Their method uses an unrestrictive correlation structure for the qualitative factors and requires the use of modern optimization methods in the estimation to ensure that the correlation structure of the qualitative factors is positive definite. Though it is possible to significantly simplify the computational complexity of their method by taking a restrictive

correlation function for the qualitative factors (McMillian et al. 1999; Joseph and Delaney 2007; Qian, Wu, and Wu 2008), these restrictive correlation functions in general lack the flexibility of capturing various types of correlations of the qualitative factors. In an approach different from that of Qian, Wu, and Wu (2008), Han et al. (2009) introduced hierarchical Bayesian Gaussian process models to accommodate both qualitative and quantitative factors, using Markov chain Monte Carlo (MCMC) methods for computation.

In this article, we propose a new approach to the emulation of computer codes with qualitative and quantitative factors. Our approach inherits the flexibility of the unrestrictive correlation structure for qualitative factors used by Qian, Wu, and Wu (2008) but replaces their complicated estimation procedure with a clever parameterization using the hypersphere decomposition, originally proposed by Rebonato and Jackel (1999) for modeling correlations in financial models. This new parameterization essentially turns the required optimization problems with positive definite constraints in the article by Qian, Wu, and Wu (2008) into standard nonlinear optimization problems with box constraints.

The remainder of the article is organized as follows. Section 2 gives the general model structure. Section 3 presents estimation and prediction procedures. Section 4 discusses computational issues in the proposed estimation method. Section 5 provides several examples to illustrate the effectiveness of the proposed method. Section 6 provides a brief summary and concluding remarks.

2. THE GENERAL MODEL

Consider a computer model with an input vector $\mathbf{w} = (\mathbf{x}^t, \mathbf{z}^t)^t$, where $\mathbf{x} = (x_1, \dots, x_I)^t$ consists of all the quantitative factors, $\mathbf{z} = (z_1, \dots, z_J)^t$ consists of all the qualitative factors, and z_j has b_j levels. Let $m = \prod_{j=1}^J b_j$. Throughout, the factors in \mathbf{z} are assumed to be categorical but not ordinal. Methods for emulation of computer codes with ordinal factors can be found in section 4.4 of the article by Qian, Wu, and Wu (2008). The response of the computer model at an input value \mathbf{w} is modeled as

$$y(\mathbf{w}) = \mathbf{f}^t(\mathbf{w})\boldsymbol{\beta} + \varepsilon(\mathbf{w}), \tag{1}$$

where $\mathbf{f}(\mathbf{w}) = (f_1(\mathbf{w}), \dots, f_p(\mathbf{w}))^t$ is a set of p user-specified regression functions, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^t$ is a vector of unknown coefficients, and the residual $\varepsilon(\mathbf{w})$ is assumed to be a stationary Gaussian process with mean 0 and variance σ^2 .

The model in (1) is a more general form of the standard Gaussian process model with only quantitative factors in \mathbf{x} :

$$y(\mathbf{x}) = \mathbf{f}^t(\mathbf{x})\boldsymbol{\beta} + \varepsilon(\mathbf{x}), \tag{2}$$

where $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_p(\mathbf{x}))^t$ is a set of p user-specified regression functions, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^t$ is a vector of unknown coefficients, and the residual $\varepsilon(\mathbf{x})$ is assumed to be a stationary Gaussian process with mean 0 and variance σ^2 , and some correlation function $\text{cor}(\varepsilon(\mathbf{x}_1), \varepsilon(\mathbf{x}_2)) = K(\mathbf{x}_1, \mathbf{x}_2)$. A popular choice of the correlation function for model (2) is the *Gaussian correlation function*

$$K(\mathbf{x}_1, \mathbf{x}_2) = \exp\left\{-\sum_{i=1}^I \phi_i(x_{1i} - x_{2i})^2\right\}. \tag{3}$$

The model in (2) has been implemented in various packages such as the Matlab toolbox DACE (Lophaven, Nielsen, and Sondergaard 2002a).

We now discuss how to specify a valid correlation structure for $\varepsilon(\mathbf{w})$ associated with the model in (1). This specification is challenging because \mathbf{w} involves both qualitative and quantitative factors. For convenience, let c_1, \dots, c_m denote m categories, corresponding to all level combinations of the factors in \mathbf{z} . Here, $\mathbf{w} = (\mathbf{x}^t, c_q)^t$ ($q = 1, \dots, m$) is used to denote all the factors involved in the model. Following Qian, Wu, and Wu (2008), for two input values $\mathbf{w}_i = (\mathbf{x}_i^t, c_i)^t$ ($i = 1, 2$), the correlation between $\varepsilon(\mathbf{w}_1)$ and $\varepsilon(\mathbf{w}_2)$ is defined to be

$$\begin{aligned} \text{cor}(\varepsilon(\mathbf{w}_1), \varepsilon(\mathbf{w}_2)) &= \text{cor}(\varepsilon_{c_1}(\mathbf{x}_1), \varepsilon_{c_2}(\mathbf{x}_2)) \\ &= \tau_{c_1, c_2} K(\mathbf{x}_1, \mathbf{x}_2), \end{aligned} \tag{4}$$

where $\tau_{c_1, c_2} = \tau_{c_2, c_1}$ is the *cross-correlation* between categories c_1 and c_2 . In numerical examples in Section 5, the Gaussian correlation function in (3) is used and (4) becomes

$$\text{cor}(\varepsilon(\mathbf{w}_1), \varepsilon(\mathbf{w}_2)) = \tau_{c_1, c_2} \exp\left\{-\sum_{i=1}^I \phi_i(x_{1i} - x_{2i})^2\right\}, \tag{5}$$

where the unknown *roughness parameters* ϕ_i will be collectively denoted as $\boldsymbol{\Phi} = \{\phi_i\}$.

For (5) to be a valid correlation function, the $m \times m$ matrix $\mathbf{T} = \{\tau_{r,s}\}$ must be a *positive definite matrix with unit diagonal elements* (PDUDE) (Qian, Wu, and Wu 2008). Departing from the work of Qian, Wu, and Wu (2008), here \mathbf{T} is modeled by

using the *hypersphere decomposition*, originally introduced by Rebonato and Jackel (1999) for modeling correlations in financial applications.

This parameterization provides a simple yet flexible way to model a PDUDE matrix. It consists of two steps. In Step 1, a Cholesky-type decomposition is applied to \mathbf{T} given by

$$\mathbf{T} = \mathbf{L}\mathbf{L}^t, \tag{6}$$

where $\mathbf{L} = \{l_{r,s}\}$ is a lower triangular matrix with strictly positive diagonal entries. In Step 2, each row vector $(l_{r,1}, \dots, l_{r,r})$ in \mathbf{L} is modeled as the coordinates of a surface point on an r -dimensional unit hypersphere described as follows. For $r = 1$, let $l_{1,1} = 1$ and for $r = 2, \dots, m$, use the following spherical coordinate system:

$$\begin{cases} l_{r,1} = \cos(\theta_{r,1}) \\ l_{r,s} = \sin(\theta_{r,1}) \cdots \sin(\theta_{r,s-1}) \cos(\theta_{r,s}) \\ \text{for } s = 2, \dots, r-1 \\ l_{r,r} = \sin(\theta_{r,1}) \cdots \sin(\theta_{r,r-2}) \sin(\theta_{r,r-1}), \end{cases} \tag{7}$$

where $\theta_{r,s} \in (0, \pi)$. Collectively, denote by $\boldsymbol{\Theta}$ all $\theta_{r,s}$ involved in (7). Because each $\theta_{r,s}$ is restricted to take values in $(0, \pi)$, the diagonal entries $l_{r,r}$ in \mathbf{L} are strictly positive, thus guaranteeing that \mathbf{T} is a positive definite matrix. In addition, $\tau_{r,r} = \sum_{s=1}^r l_{r,s}^2 = 1$ ($r = 1, \dots, m$) by (7), implying that \mathbf{T} must have unit diagonal elements. Thus, the matrix \mathbf{T} under this parameterization is always a PDUDE.

For illustration, consider the case with $m = 3$. In Step 1, a 3×3 PDUDE

$$\mathbf{T}_3 = \begin{bmatrix} 1 & \tau_{12} & \tau_{13} \\ \tau_{12} & 1 & \tau_{23} \\ \tau_{13} & \tau_{23} & 1 \end{bmatrix} \tag{8}$$

is decomposed as

$$\mathbf{T}_3 = \mathbf{L}_3 \mathbf{L}_3^t = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} 1 & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}. \tag{9}$$

In Step 2, (l_{21}, l_{22}) are transformed into a two-dimensional (2D) spherical coordinate system

$$\begin{cases} l_{21} = \cos(\theta_{21}) \\ l_{22} = \sin(\theta_{21}), \end{cases} \tag{10}$$

and (l_{31}, l_{32}, l_{33}) are transformed into a 3D spherical coordinate system

$$\begin{cases} l_{31} = \cos(\theta_{31}) \\ l_{32} = \sin(\theta_{31}) \cos(\theta_{32}) \\ l_{33} = \sin(\theta_{31}) \sin(\theta_{32}), \end{cases} \tag{11}$$

where $\theta_{r,s} \in (0, \pi)$ can be calculated based on the following equations:

$$\begin{cases} \tau_{12} = \cos(\theta_{21}) \\ \tau_{13} = \cos(\theta_{31}) \\ \tau_{23} = \cos(\theta_{21}) \cos(\theta_{31}) + \sin(\theta_{21}) \sin(\theta_{31}) \cos(\theta_{32}). \end{cases} \tag{12}$$

In (10) and (11), (l_{21}, l_{22}) are the coordinates of a point on the half unit circle given by $l_{21}^2 + l_{22}^2 = 1$ and $l_{22} > 0$ as shown in Figure 1(a); (l_{31}, l_{32}, l_{33}) are the coordinates of a surface point on the unit hemisphere given by $l_{31}^2 + l_{32}^2 + l_{33}^2 = 1$ and $l_{33} > 0$ as shown in Figure 1(b).

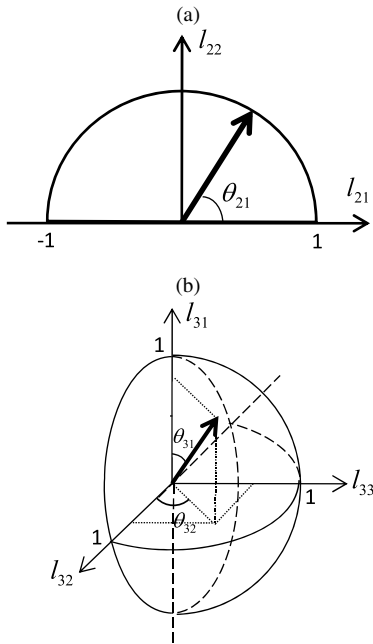


Figure 1. (a) Point (l_{21}, l_{22}) on the half unit circle; (b) point (l_{31}, l_{32}, l_{33}) on the unit hemisphere.

The proposed parameterization has several advantages. First, it turns the complicated PDUDE constraint on \mathbf{T} into box constraints $\theta_{r,s} \in (0, \pi)$. Second, because $\theta_{r,s}$ take values in $(0, \pi)$, the entries in \mathbf{T} can be either positive or negative and thus can capture various correlations across different categories. Third, there is a one-to-one correspondence between a PDUDE matrix and Θ . That is, a PDUDE matrix with an arbitrary structure can be parameterized by using a particular Θ value and any given Θ always gives a PDUDE matrix.

Based on the *hypersphere decomposition*, the number of parameters required to construct a PDUDE matrix is $m(m-1)/2$. For situations with multiple qualitative factors, instead of using the correlation function in (5), one may take the following product form:

$$\begin{aligned} \text{cor}(\varepsilon(\mathbf{w}_1), \varepsilon(\mathbf{w}_2)) &= \text{cor}(\varepsilon(\mathbf{x}_1, \mathbf{z}_1), \varepsilon(\mathbf{x}_2, \mathbf{z}_2)) \\ &= \left[\prod_{j=1}^J \tau_{j, z_{j1}, z_{j2}} \right] \exp \left\{ - \sum_{i=1}^I \phi_i (x_{1i} - x_{2i})^2 \right\}, \end{aligned} \quad (13)$$

where each matrix $\mathbf{T}_j = \{\tau_{j,r,s}\}$ ($r, s = 1, \dots, m_j$) is a PDUDE that is modeled by using the parameterization in (4) and (5). Though the product correlation function in (13) is not as flexible as the general correlation function in (5), it can significantly reduce the number of parameters that need to be estimated. For example, for a computer experiment consisting of three qualitative factors, each with three levels, the correlation function in (5) would require $3^3(3^3 - 1)/2 = 351$ parameters to construct the PDUDE matrix, whereas the correlation function in (13) would involve only $\sum_{j=1}^3 3(3 - 1)/2 = 9$ parameters. Note that correlation functions with similar product structures like the product Gaussian correlation function are widely used in fitting Gaussian process models with quantitative factors (Santner, Williams, and Notz 2003).

3. ESTIMATION AND PREDICTION

Suppose the computer model under consideration is conducted at n different input values, $\mathbf{D}_w = (\mathbf{w}_1^0, \dots, \mathbf{w}_n^0)$, and the corresponding response values are denoted by $\mathbf{y} = (y_1, \dots, y_n)^t$. The parameters in model (1) to be estimated are σ^2 , β , Φ , and Θ . We use the maximum likelihood to estimate these parameters and denote the resulting estimators by $\hat{\sigma}^2$, $\hat{\beta}$, $\hat{\Phi}$, and $\hat{\Theta}$. The log-likelihood function of \mathbf{y} , up to an additive constant, is

$$-\frac{1}{2}[n \log(\sigma^2) + \log |\mathbf{R}| + (\mathbf{y} - \mathbf{F}\beta)' \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\beta) / \sigma^2], \quad (14)$$

where $\mathbf{F} = (\mathbf{f}(\mathbf{w}_1^0), \dots, \mathbf{f}(\mathbf{w}_n^0))^t$ is an $n \times p$ matrix and \mathbf{R} is the correlation matrix whose (i, j) th entry is $\text{cor}(\varepsilon(\mathbf{w}_i^0), \varepsilon(\mathbf{w}_j^0))$ defined in (5) or (13). Given Φ and Θ , $\hat{\beta}$ and $\hat{\sigma}^2$ are

$$\hat{\beta} = (\mathbf{F}' \mathbf{R}^{-1} \mathbf{F})^{-1} \mathbf{F}' \mathbf{R}^{-1} \mathbf{y}, \quad (15)$$

$$\hat{\sigma}^2 = (\mathbf{y} - \mathbf{F}\hat{\beta})' \mathbf{R}^{-1} (\mathbf{y} - \mathbf{F}\hat{\beta}) / n.$$

Substituting (15) into (14), $\hat{\Phi}$ and $\hat{\Theta}$ can be obtained as

$$(\hat{\Phi}, \hat{\Theta}) = \underset{(\Phi, \Theta)}{\text{argmin}} \{n \log(\hat{\sigma}^2) + \log |\mathbf{R}|\}. \quad (16)$$

The optimization problem in (16) involves the constraints $\theta_{r,s} \in (0, \pi)$ for Θ and $\phi_i \geq 0$ for Φ . It can be solved by using standard nonlinear optimization algorithms in R or Matlab and is much simpler than the optimization problems with positive definite constraints in the estimation procedure of Qian, Wu, and Wu (2008). The fitted model can be used to predict the response value y at any untried point in the design space. Given all the estimated parameters, the *empirical best linear unbiased predictor* (EBLUP) of y at any input value \mathbf{w}_0 is

$$\hat{y}(\mathbf{w}_0) = \mathbf{f}'(\mathbf{w}_0) \hat{\beta} + \hat{\mathbf{r}}_0' \hat{\mathbf{R}}^{-1} (\mathbf{y} - \mathbf{F}\hat{\beta}), \quad (17)$$

where $\hat{\mathbf{r}}_0 = (\widehat{\text{cor}}(\varepsilon(\mathbf{w}_0), \varepsilon(\mathbf{w}_1^0)), \dots, \widehat{\text{cor}}(\varepsilon(\mathbf{w}_0), \varepsilon(\mathbf{w}_n^0)))^t$ and $\hat{\mathbf{R}}$ is the estimated correlation matrix of \mathbf{y} . Similarly to its counterpart for the standard Gaussian process model in (2) with only quantitative factors (Santner, Williams, and Notz 2003; Fang, Li, and Sudjianto 2005), the EBLUP in (17) smoothly interpolates all the observed data points. The features of the function $y(\mathbf{w})$ can be visualized by plotting the estimated functional main effects and interactions of the predictor $\hat{y}(\mathbf{w})$. Details of performing ANOVA decompositions can be found in the work of Schonlau and Welch (2006).

4. COMPUTATIONAL ISSUES

If the design set \mathbf{D}_w has some cross-array structure (Wu and Hamada 2009) between the design for the quantitative factors \mathbf{x} and the design for the qualitative factors \mathbf{z} , the optimization problem in (16) can be further simplified. This simplification is in the same spirit of the simplification of the iterative estimation procedure in the article by Qian, Wu, and Wu (2008) using well-structured designs. First consider the model in (5) where the same set of input values $(\mathbf{x}_1, \dots, \mathbf{x}_{n_c})^t$ is chosen for the quantitative factors \mathbf{x} , across the m categories defined in Section 2. Hence, \mathbf{D}_w can be expressed as a *cross-array* of $\mathbf{D}_x = (\mathbf{x}_1, \dots, \mathbf{x}_{n_c})$ and $\mathbf{D}_c = (1, \dots, m)$. Then \mathbf{R} can be simplified to the Kronecker product of two smaller matrices given by

$$\mathbf{R} = \mathbf{T} \otimes \mathbf{H}, \quad (18)$$

where \mathbf{H} is the $n_c \times n_c$ matrix whose (j_1, j_2) th entry is $K(\mathbf{x}_{j_1}, \mathbf{x}_{j_2})$ and \otimes denotes the Kronecker product. By the positive definiteness of the three matrices in (18) and properties of the Kronecker product (Graham 1981), we have that

$$\mathbf{R}^{-1} = \mathbf{T}^{-1} \otimes \mathbf{H}^{-1} \tag{19}$$

and

$$\begin{aligned} \log |\mathbf{R}| &= \log |\mathbf{T} \otimes \mathbf{H}| = \log(|\mathbf{T}|^{n_c} \cdot |\mathbf{H}|^m) \\ &= n_c \log |\mathbf{T}| + m \log |\mathbf{H}|. \end{aligned} \tag{20}$$

Plugging (19) and (20) into the objective function in (16) can significantly simplify computations. Given the close connections between the proposed model in Section 2 and the standard Gaussian process model with quantitative factors in (2), numerical techniques developed to mitigate numerical issues for the latter can be adapted here. For the examples presented in Section 5, the proposed method is implemented by modifying the Matlab toolbox DACE, Version 2.5, for fitting kriging models with quantitative factors. More information about DACE can be found in the work of Lophaven, Nielsen, and Sondergaard (2002a, 2002b). This toolbox uses a gradient-free pattern search algorithm for optimization and includes a very small nugget term to the diagonal elements of the correlation matrix \mathbf{R} to avoid ill conditioning. If a gradient based optimizer is used, a penalized likelihood approach can be helpful (Li and Sudjianto 2005) in cases when the likelihood function is flat near its optimum. Other robust methods for inverting correlation matrices can be found in the works of Jones, Schuoulay, and Welch (1998), Booker (2000), and Ranjan, Haynes, and Karsten (2010).

5. EXAMPLES

In this section, we provide numerical illustration to demonstrate the effectiveness of the proposed method. We compare the following four methods for modeling computer experiments with qualitative and quantitative factors:

1. The individual Kriging method, denoted by IK. This method fits the data associated with every level combination of the qualitative factors separately using the standard Gaussian model in (2) and (3) with a constant mean (Santner, Williams, and Notz 2003).
2. The exchangeable correlation method, denoted by EC. This method fits a single Gaussian process model with qualitative and quantitative factors, where the Gaussian correlation function is used for the quantitative factors and the exchangeable correlation function, $\tau_{r,s} = c$ ($0 < c < 1$) for $r \neq s$ (Joseph and Delaney 2007; Qian, Wu, and Wu 2008), is used for the qualitative factors.
3. The multiplicative correlation method, denoted by MC. The multiplicative correlation function (McMillian et al. 1999; Qian, Wu, and Wu 2008) has the following form:

$$\tau_{r,s} = \exp\{-(\theta_r + \theta_s)I[r \neq s]\} \quad (\theta_r, \theta_s > 0).$$

4. The proposed method discussed in Section 2, denoted by UC, which stands for unrestrictive correlation.

We note some differences between these methods. First, methods 2, 3, 4 all use a *single* Gaussian process model to analyze all available data, whereas the IK method, also called the independent analysis in the article by Qian, Wu, and Wu (2008), fits distinct Gaussian process models in (2) to data collected at different level combinations of the qualitative factors, thus ignoring possible correlations across the categories. Second, because the correlation structure of the qualitative factors for the UC method, using the hypersphere decomposition, is much more flexible than those of the EC and MC methods, the UC method is expected to produce more accurate results than the latter two methods. Third, the correlation function for the qualitative factors of the UC method is “structure free” and can capture both positive and negative cross-correlations between different categories, which cannot be modeled by the MC method.

From the modeling perspective, these four methods are interconnected. All of them fit Kriging type emulators with qualitative and quantitative factors but have different degrees of flexibility in modeling the correlations of the qualitative factors. The IK model does not borrow information across data from different categories. It is refined by the EC method based on a simple function for capturing cross-correlations among the categories, which in turn is enhanced by the MC method with a more flexible correlation structure. The correlation function of the UC method improves upon those of the EC and MC methods. The connection between these methods provides a basis for setting initial correlation parameter values in fitting the methods. We found that parameter estimates of Φ from the IK method provide good initial values of roughness parameters for the EC method, and estimates of \mathbf{T} from the EC method provide good initial values of correlation parameters of the MC or UC method.

Below we present four numerical examples with varying numbers of qualitative factors. In fitting the UC method to these examples, a constant term is used for $\mathbf{f}(\mathbf{w})$ in (1) unless described otherwise and the product correlation function in (13) is used. Table 1 summarizes the parameters of these examples, where n is the sample size, I is the number of quantitative factors, J is the number of qualitative factors, and m is the number of categories.

5.1 An Example With Both Positive and Negative Cross-Correlations

This example considers an experiment with one quantitative factor, x_1 , taking values on $[0, 1]$ and one qualitative factor, z_1 , with three levels. The response of the experiment takes the following form:

$$y = \begin{cases} \cos(6.8\pi x_1/2), & \text{if } z_1 = 1 \\ -\cos(7\pi x_1/2), & \text{if } z_1 = 2 \\ \cos(7.2\pi x_1/2), & \text{if } z_1 = 3. \end{cases} \tag{21}$$

Figure 2 compares three curves of the function at different levels of z_1 . In the absolute scale, these curves are similar to one another. Since the second equation in (21) contains a negative sign, the curve with $z_1 = 2$ is negatively correlated with those when $z_1 = 1$ and $z_1 = 3$, and the curves with $z_1 = 1$ and $z_1 = 3$ are positively correlated with each other.

For each level of z_1 , a training sample is obtained by using a Latin hypercube design on $[0, 1]$ of eight runs for x_1

Table 1. Summary of the number of parameters of the four examples

Example	n	I	J	m	Parameters in \mathbf{T}				Total correlation parameters			
					IK	EC	MC	UC	IK [†]	EC	MC	UC
5.1	24	1	1	3	–	1	3	3	1	2	4	4
5.2	13	1	1	3	–	1	3	3	1	2	4	4
5.3	67	5	3	24	–*	3*	8*	10	5*	8*	13*	15
5.4	216	4	3	27	–*	3	9	9	4*	7	13	13

[†]For IK, correlation parameters are roughness parameters for a single category.

*These methods are not computed in examples; listed for reference only.

(McKay, Beckman, and Conover 1979), and a testing sample is taken at 20 equally spaced points of the interval $[0, 1]$: $0, 1/19, 2/19, \dots, 1$. The root mean squared errors (RMSEs) of the testing sample are calculated for the four methods described in the beginning of Section 5 to assess their prediction accuracy. This procedure of data generation, modeling fitting, and assessment of prediction accuracy was repeated 100 times. Figure 3 compares the RMSEs of the testing sample for the four methods.

The mean values of the 100 RMSEs for the four models are 0.0392, 0.0391, 0.0365, and 0.0191, respectively, indicating that the UC method achieves the best performance. Average estimates of cross-correlations parameters $\tau_{1,2}$, $\tau_{1,3}$, and $\tau_{2,3}$ across the 100 simulations are (0.01, 0.88, 0.01) for the MC method and (–0.95, 0.81, –0.93) for the UC methods, with minimal variation. This example demonstrates that the UC method correctly captures both the positive and negative cross correlations, whereas the MC method fails to catch the two negative correlations due to its positiveness constraint in the cross-correlation matrix.

5.2 An Example From Han et al. (2009)

We now compare the four methods using an example from the work of Han et al. (2009). This example uses the following quadratic function with one quantitative factor $x_1 \in [0, 1]$ and

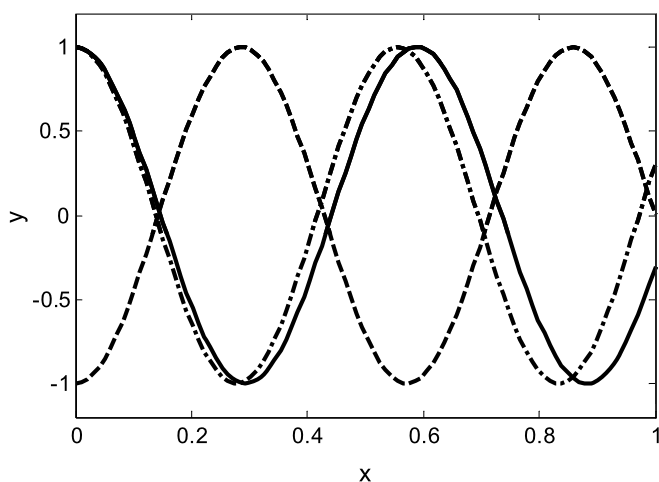


Figure 2. Example 5.1: function curves at $z_1 = 1$ (–), $z_1 = 2$ (– –), and $z_1 = 3$ (– ·).

one three-level qualitative factor z_1 :

$$y = \begin{cases} b_{01} + b_{11}x_1 + b_{21}x_1^2, & \text{if } z_1 = 1 \\ b_{02} + b_{12}x_1 + b_{22}x_1^2, & \text{if } z_1 = 2 \\ b_{03} + b_{13}x_1 + b_{23}x_1^2, & \text{if } z_1 = 3. \end{cases} \quad (22)$$

Following Han et al. (2009), training data and testing data are generated as follows. The first two levels of the function are evaluated at $\{0, 0.25, 0.5, 0.75, 1\}$ and the third level at $\{0.5, 0.75, 1\}$. Three scenarios are used to produce true quadratic curves in the testing. In each scenario, the nine coefficients of the three testing functions are randomly drawn from normal distributions with the same standard deviation of 0.01. The expected values of $(b_{01}, b_{02}, b_{03}, b_{11}, b_{12}, b_{13}, b_{21}, b_{22}, b_{23})$ for the three scenarios are $(1, 0, -1, 6, 4, 5, -6, -6, -6)$, $(1, 0, -1, 0, 6, 5, 2, -6, -6)$, and $(1, 0, -1, 6, 6, 6, -6, -6, -6)$, respectively. The testing data are obtained for $z_1 = 3$ at interpolation points, $\{0.5, 0.51, \dots, 1.00\}$, and extrapolation points, $\{0, 0.01, \dots, 0.50\}$. The above steps of data generation, modeling fitting, and assessment of prediction accuracy were repeated 30 times for all three scenarios, where independent random samples of b_{ij} 's were drawn each time.

Tables 2 and 3 compare RMSE quantiles for the testing data. The UC method performs the best except under extrapolation for Scenarios 1 and 2 with $z_1 = 3$. Compared with the boxplots reported in figures 4 and 5 of the article by Han et al. (2009) for this example, but with different data realizations for the quadratic curve coefficients, the UC method appears to outperform the following methods: (1) SHB: a “surfacewise hierarchical Bayes” predictor; (2) KOH: the autoregressive model

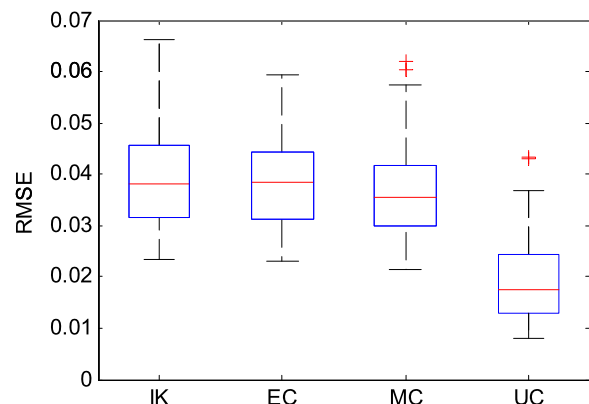


Figure 3. Boxplots of the RMSEs of the four methods for Example 5.1.

Table 2. Example 5.2: RMSE quantiles ($\times 10^4$) at interpolation testing points, $z_1 = 3$

Quantile:	Scenario 1			Scenario 2			Scenario 3		
	25%	50%	75%	25%	50%	75%	25%	50%	75%
IK	949.92	966.58	984.67	950.33	951.95	983.73	930.82	934.41	936.41
EC	2.64	2.76	2.90	176.51	177.17	180.07	5.22	5.69	5.81
MC	11.73	12.12	12.36	153.58	154.39	189.56	7.10	7.36	7.47
UC	2.53	2.68	4.37	73.75	74.26	75.32	0.04	0.07	0.10

developed by Kennedy and O'Hagan (2000); and (3) HQQV: the hierarchical Bayesian model proposed by Han et al. (2009).

To expand the testing set discussed above, the UC method is used to predict the other two levels of z_1 at the two sets of testing points in Table 4. This table indicates that compared with the results for $z_1 = 3$, predictions at the other two levels of z_1 are better because of the use of a larger set of training sample. Compared with the other three methods, the UC method produces significantly smaller RMSE values in Scenarios 1 and 3, and has similar results with EC and MC in Scenario 2 (results not shown here).

5.3 A Data Center Computer Experiment

Here we use the UC method to reanalyze the data of a data center computer experiment from section 7 of the article by Qian, Wu, and Wu (2008). This experiment studies the thermal dynamics of an air-cooled data center using a computational fluid-dynamics program in Flotherm. The goal of the experiment is to predict airflow and heat transfer in the electronic equipment of the data center. Each run of the experiment takes hours or even days to complete. This experiment contains five quantitative factors, x_1, x_2, x_3, x_4, x_5 , and three qualitative factors, z_1, z_2, z_3 , with two, four, and three levels, respectively. The response variable, y , is the temperature at a selected location of the system. Since there are 67 observations for 24 level combinations of three qualitative factors, on average each level combination of the qualitative factors has less than three observations. In the work of Qian, Wu, and Wu (2008), this example is analyzed by fitting the model in (13) with a semi-definite programming approach that does not use the hypersphere decomposition for \mathbf{T}_j .

We use the same form of $\mathbf{f}(\mathbf{w})'\boldsymbol{\beta}$ given in by Qian, Wu, and Wu (2008). Following Qian, Wu, and Wu (2008), a leave-one-out cross-validation procedure is employed to assess prediction accuracy of our method, where the model correlation parameters and correlation matrices are obtained based on all data points and hence not recomputed each time. The RMSE of our

method is 1.70, similar to 1.88 of the method developed by Qian, Wu, and Wu (2008). However, the proposed method gains significant computational efficiency. It took less than 20 seconds on a PC with Intel Core2 Duo CPU at 2.00 GHz to fit the proposed method to this example, whereas fitting the method in the article by Qian, Wu, and Wu (2008) with 400 iterations to the same example took more than 3 hours on a double-core PC running a Linux system.

5.4 The Borehole Example

The borehole function, introduced by Morris, Mitchell, and Ylvisaker (1993), is widely used for illustrating various methods in computer experiments. This function models the flow rate of water through a borehole. Table 5 presents the eight input variables of the function.

The response y of the function is defined as

$$y = 2\pi T_u(H_u - H_l)$$

$$/ \left(\log(r/r_w) \left[1 + \frac{2LT_u}{\log(r/r_w)r_w^2 K_w} + \frac{T_u}{T_l} \right] \right). \quad (23)$$

We now use this function to create an experiment with four quantitative factors x_1, x_2, x_3 , and x_4 , and three qualitative factors z_1, z_2 , and z_3 . Note that although eight inputs are listed in the table, we will treat $H_u - H_l$ as a single factor in this example. By using four input variables in Table 5, the three qualitative factors are defined by Table 6, each of which has three levels. The four input variables in Table 5 that are not used in Table 6 are left as the four quantitative factors, x_1, x_2, x_3 , and x_4 , of this experiment.

A training sample of 216 runs is generated in which the design set for the three qualitative factors is obtained by repeating a 3^3 full factorial design eight times and the design set for the quantitative factors is a Latin hypercube design with 216 levels. Data from the training sample are used to fit prediction models based on the EC, MC, and UC methods. Since the training sample only has eight runs for each level combination of the qualitative factors, the IK method is not used. The accuracy of

Table 3. Example 5.2: RMSE quantiles ($\times 10^4$) at extrapolation testing points, $z_1 = 3$

Quantile:	Scenario 1			Scenario 2			Scenario 3		
	25%	50%	75%	25%	50%	75%	25%	50%	75%
IK	2715.90	2727.40	2746.28	2723.09	2738.23	2748.88	2756.83	2789.11	2809.16
EC	67.54	72.33	79.09	633.52	634.84	647.33	195.59	203.47	207.11
MC	943.93	969.46	984.78	457.08	462.25	693.49	72.83	75.32	82.79
UC	110.96	116.23	189.62	665.49	689.93	712.60	0.56	1.16	1.86

Table 4. Example 5.2: RMSE quantiles ($\times 10^4$) for UC method at $z_1 = 1$ and $z_1 = 2$

	Quantile:	Interpolation			Extrapolation		
		25%	50%	75%	25%	50%	75%
Scenario 1	$z_1 = 1$	0.15	0.16	0.28	0.15	0.16	0.28
	$z_1 = 2$	0.32	0.34	0.60	0.05	0.05	0.09
Scenario 2	$z_1 = 1$	30.29	30.49	31.93	11.07	11.18	11.63
	$z_1 = 2$	21.26	21.44	22.39	21.34	21.44	22.40
Scenario 3	$z_1 = 1$	0.04	0.04	0.04	0.04	0.04	0.04
	$z_1 = 2$	0.04	0.04	0.04	0.04	0.04	0.04

Table 5. Input variables in the borehole example

Input variable	Description	Range	Unit
r_w	radius of borehole	0.05 ~ 0.15	m
r	radius of influence	100 ~ 50,000	m
T_u	transmissivity of upper aquifer	63,070 ~ 115,600	m^2/yr
H_u	potentiometric head of upper aquifer	990 ~ 1110	m
T_l	transmissivity of lower aquifer	63.1 ~ 116	m^2/yr
H_l	potentiometric head of lower aquifer	700 ~ 820	m
L	length of borehole	1120 ~ 1680	m
K_w	hydraulic conductivity of borehole	9855 ~ 12,045	m/yr

the prediction model for each of the three methods is assessed on a testing sample of 405 runs in which the design set for the three qualitative factors is obtained by repeating a 3^3 full factorial design 15 times and the design set for the quantitative factors is a Latin hypercube design with 405 levels.

The above procedure of data generation, modeling fitting, and model assessment was repeated 100 times for each of the three methods mentioned above. Figure 4 compares the RMSEs of the testing data, where the UC method, once more, outperforms the EC and MC methods.

6. SUMMARY AND DISCUSSION

We have proposed a new method for modeling computer experiments with qualitative and quantitative factors. By employing a parameterization for modeling the correlations of the quantitative factors, this method turns complicated constrained optimization problems with positive-definite constraints into much simpler problems with box constraints. The proposed method is developed under the usual assumption that computer experiments are deterministic in nature; that is, running such an experiment twice at the same input configuration yields identical response values (Santner, Williams, and Notz 2003). A possible extension of this work is to model stochastic computer experiments with qualitative and quantitative factors. For a sim-

Table 6. Borehole example: three qualitative factors and their levels

Level	$z_1 = r$	$z_2 = H_u - H_l$	$z_3 = K_w$
1	10,000	200	10,000
2	20,000	300	11,000
3	30,000	400	12,000

ple stochastic computer model in which the output variance is stationary in the inputs, such an extension entails including a larger nugget term in (1). For a more complicated stochastic computer code like a discrete event analysis program, both the variance σ^2 in (1) and the correlation structure can vary from one input value to another and thus become functions of the inputs (Ankenman, Nelson, and Staum 2010). Extending our method to deal with such nonstationarity poses significant challenges.

This work is focused on the modeling aspect of computer experiments with both qualitative and quantitative factors. For the corresponding data collection issue, the interested reader is referred to the work of Qian and Wu (2009) for an experimental design framework for computer experiments with mixed types of inputs. All computations used in this article are done in MATLAB programs, which are available from the authors. A Bayesian alternative of the proposed methodology will be developed and reported elsewhere.

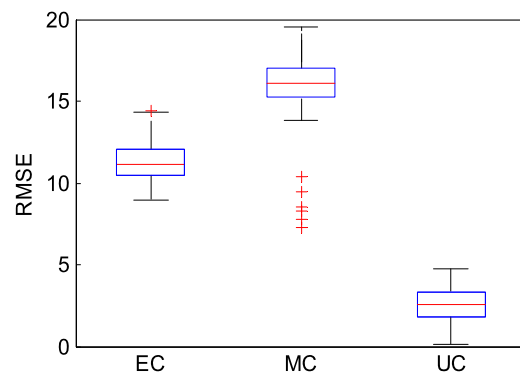


Figure 4. Borehole example: RMSE boxplots of the UC, EC, and MC method.

ACKNOWLEDGMENTS

This work is supported by grants 0545600, 0705206, 0856222, and 0969616 from the National Science Foundation and an IBM Faculty Award. The authors thank the editor, the associate editor, and two referees for their helpful comments that have led to improvements in the article.

[Received February 2010. Revised March 2011.]

REFERENCES

- Ankenman, B., Nelson, B. L., and Staum, J. (2010), "Stochastic Kriging for Simulation Metamodeling," *Operations Research*, 58, 371–382. [272]
- Booker, A. (2000), "Well-Conditioned Kriging Models for Optimization of Computer Simulations," Mathematics and Computing Technology Phantom Works, M&CTECH-00-002, The Boeing Company. [269]
- Curran, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991), "Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments," *Journal of the American Statistical Association*, 86, 953–963. [266]
- Fang, K. T., Li, R., and Sudjianto, A. (2005), *Design and Modeling for Computer Experiments*, New York: Chapman & Hall/CRC Press. [266,268]
- Graham, A. (1981), *Kronecker Products and Matrix Calculus With Applications*, Chichester, U.K.: Ellis Horwood Limited. [269]
- Han, G., Santner, T. J., Notz, W. I., and Bartel, D. L. (2009), "Prediction for Computer Experiments Having Quantitative and Qualitative Input Variables," *Technometrics*, 51, 278–288. [266,270,271]
- Jones, D., Schonlau, M., and Welch, W. (1998), "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, 13, 455–492. [269]
- Joseph, V. R., and Delaney, J. D. (2007), "Functionally Induced Priors for the Analysis of Experiments," *Technometrics*, 49, 1–11. [266,269]
- Kennedy, M. C., and O'Hagan, A. (2000), "Predicting the Output From a Complex Computer Code When Fast Approximations Are Available," *Biometrika*, 87, 1–13. [266,271]
- Li, R., and Sudjianto, A. (2005), "Analysis of Computer Experiments Using Penalized Likelihood in Gaussian Kriging Models," *Technometrics*, 47, 111–120. [269]
- Lophaven, S. N., Nielsen, H. B., and Sondergaard, J. (2002a), "Matlab Kriging Toolbox DACE," version 2.5, available at <http://www2.imm.dtu.dk/~hbn/dace/>. [267,269]
- (2002b), "Aspects of the Matlab Toolbox DACE," Technical Report IMM-REP-2002-13, Technical University of Denmark. [269]
- McKay, M. D., Beckman, R. J., and Conover, W. J. (1979), "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code," *Technometrics*, 21, 239–245. [270]
- McMillian, N. J., Sacks, J., Welch, W. J., and Gao, F. (1999), "Analysis of Protein Activity Data by Gaussian Stochastic Process Models," *Journal of Biopharmaceutical Statistics*, 9, 145–160. [266,269]
- Morris, M. D., Mitchell, T. J., and Ylvisaker, D. (1993), "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction," *Technometrics*, 35, 243–255. [271]
- Qian, P. Z. G., and Wu, C. F. J. (2008), "Bayesian Hierarchical Modeling for Integrating Low-Accuracy and High-Accuracy Experiments," *Technometrics*, 50, 192–204. [266]
- (2009), "Sliced Space-Filling Designs," *Biometrika*, 96, 945–956. [272]
- Qian, P. Z. G., Wu, H., and Wu, C. F. J. (2008), "Gaussian Process Models for Computer Experiments With Qualitative and Quantitative Factors," *Technometrics*, 50, 283–396. [266-269,271]
- Qian, Z., Seepersad, C., Joseph, R., Allen, J., and Wu, C. F. J. (2006), "Building Surrogate Models With Detailed and Approximate Simulations," *ASME Journal of Mechanical Design*, 128, 668–677. [266]
- Ranjan, P., Haynes, R., and Karsten, R. (2010), "Gaussian Process Models and Interpolators for Deterministic Computer Simulators," available at arXiv:1003.1315. [269]
- Rawlinson, J. J., Furman, B. D., Li, S., Wright, T. M., and Bartel, D. L. (2006), "Retrieval, Experimental, and Computational Assessment of the Performance of Total Knee Replacements," *Journal of Orthopaedic Research*, 24, 1384–1394. [266]
- Rebonato, R., and Jackel, P. (1999), "The Most General Methodology for Creating a Valid Correlation Matrix for Risk Management and Option Pricing Purposes," *The Journal of Risk*, 2, 17–27. [266,267]
- Sacks, J., Schiller, S. B., and Welch, W. J. (1989), "Designs for Computer Experiments," *Technometrics*, 31, 41–47. [266]
- Sacks, J., William, J. W., Mitchell, T. J., and Wynn, H. P. (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, 4, 409–423. [266]
- Santner, T. J., Williams, B. J., and Notz, W. I. (2003), *The Design and Analysis of Computer Experiments*, New York: Springer. [266,268,269,272]
- Schonlau, M., and Welch, W. J. (2006), "Screening the Input Variables to a Computer Model via Analysis of Variance and Visualization," in *Screening Methods for Experimentation in Industry, Drug Discovery and Genetics*, eds. A. M. Dean and S. Lewis, New York: Wiley, pp. 308–327. [268]
- Wu, C. F. J., and Hamada, M. (2009), *Experiments: Planning, Analysis, and Optimization*, New York: Wiley. [268]