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. This method is easy to compute and can be implemented straightforwardly in standard computational environments like R and Matlab. The effectiveness of the proposed method is successfully illustrated by several examples.

KEY WORDS: Computer experiment; Kriging; Hypersphere decomposition

1. INTRODUCTION

Computer models are now ubiquitous in almost all fields. The standard statistical framework for the design and analysis of computer experiments assumes that all the factors are quantitative (Santner, Williams, and Notz 2003; Fang, Li, and Sudjianto 2005). In many areas, however, computer models can contain both qualitative and quantitative factors. For example, computational fluid-dynamics program 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, Wu, and Wu 2008). Rawlinson, Furman, Li, Wright and Bartel (2006), and Han, Santner, Notz, and Bartel (2009) discussed knee models in biomechanical engineering for investigating wear mechanisms of total knee replacements that can have qualitative factors such as "prosthesis design" and "force pattern". Furthermore, a set of multi-fidelity computer models with the same quantitative factors (Kennedy and O'Hagan 2000; Qian, Seepersad, Joseph, Allen, and Wu 2006; Qian and Wu 2008) can be treated collectively as a computer model with a common set of quantitative factors and a qualitative factor to describe the different accuracy of the original models (Han et al. 2009).

Several methods are now available for building Gaussian process based emulators with qualitative and quantitative factors. Qian et al. (2008) proposed a general framework for building Gaussian process models with qualitative and quantitative factors. Their method uses an unrestrictive correlation structure for the quantitative factors and requires the use of special optimization methods in the estimation to guarantee the positive-definiteness of the assumed correlation structure. It is possible to significantly simplify the computational complication of their method by taking a restrictive correlation function for the qualitative factors (McMillian, Sacks, Welch, and Gao 1999; Joseph and Delaney 2007; Qian et al. 2008), but these restrictive correlation functions lacks the flexibility of capturing various types of correlations of the qualitative factors. Different from Qian et al. (2008), Han et al. (2009) introduced some hierarchical Bayesian Gaussian process models with these two types of factors and used Markov chain Monte Carlo (MCMC) methods for the computation.

We propose a flexible yet computationally efficient approach to this emulation problem. Our approach inherits the flexibility of the unrestrictive correlation structure for the qualitative factors used in Qian et al. (2008) but replaces their complicated estimation procedure with a clever parameterization using the hypersphere decomposition, originally proposed in Rebonato and Jackel (1999) for modeling the correlations of some financial models. This new parameterization essentially turns some optimization problems with positive-definite constraints into standard and easy to compute optimization problems with box constraints. The proposed method is easy to compute and can be implemented straightforwardly in standard computational environments like R and Matlab.

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

2. THE GENERAL MODEL

Consider a computer model with an input vector $\boldsymbol{w} = (\boldsymbol{x}^t, \boldsymbol{z}^t)^t$, where $\boldsymbol{x} = (x_1, \dots, x_I)^t$ consists of all the quantitative factors, $\boldsymbol{z} = (z_1, \dots, z_J)^t$ 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 $\boldsymbol{z}$ are assumed to be categorical but not ordinal. The response of the computer model at an input value $\boldsymbol{w}$ is modeled as

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

where $\boldsymbol{f}(\boldsymbol{w}) = (f_1(\boldsymbol{w}), \dots, f_p(\boldsymbol{w}))^{\boldsymbol{t}}$ 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 $\varepsilon(\boldsymbol{w})$ is assumed to be a stationary Gaussian process with mean 0 and variance σ^2.

The model in (1) has a similar flavor to the standard Gaussian process model with quantitative factors, which models the response y at an input value $\boldsymbol{x}$ as

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

where $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \dots, f_p(\boldsymbol{x}))^{\boldsymbol{t}}$ 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 $\varepsilon(\boldsymbol{x})$ is assumed to be a stationary Gaussian process with mean 0 and variance σ^2, and some correlation function $cor\big(\varepsilon(\boldsymbol{x}_1), \varepsilon(\boldsymbol{x}_2)\big) = K(\boldsymbol{x}_1, \boldsymbol{x}_2)$. A popular choice of the correlation function for model (2) is the Gaussian correlation function

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

The model in (2) has been implemented in various packages such as the Matlab toolbox DACE

(Lophaven, Nielsen, and Sondergaard 2002b).

We now discuss how to specify a valid correlation structure for $\varepsilon(\boldsymbol{w})$ associated with the model in (1). This specification is challenging because $\boldsymbol{w}$ involves both qualitative and quantitative factors. For convenience, let $c_1, \dots, c_m$ denote m categories, corresponding to the m level combinations of the factors in $\boldsymbol{z}$. Without confusion, we use $\boldsymbol{w} = (\boldsymbol{x}^t, c_q)^t$ $(q = 1, \dots, m)$ to denote the factors involved in the computer model under consideration. Following Qian et al. (2008), for two input values $\boldsymbol{w}_i = (\boldsymbol{x}_i^t, c_i)^t$ $(i = 1,2)$, the correlation between $\varepsilon(\boldsymbol{w}_1)$ and $\varepsilon(\boldsymbol{w}_2)$ is defined to be

$$cor\big(\varepsilon(\boldsymbol{w}_1), \varepsilon(\boldsymbol{w}_2)\big) = cor\left(\varepsilon_{c_1}(\boldsymbol{x}_1), \varepsilon_{c_2}(\boldsymbol{x}_2)\right) = \tau_{c_1,c_2} K(\boldsymbol{x}_1, \boldsymbol{x}_2), \tag{4}$$

where τ_{c_1,c_2} is the cross-correlation between categories c_1 and c_2. In our numerical examples, we use the Gaussian correlation function in (3) and (4) becomes

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

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 $\boldsymbol{T} = \{\tau_{r,s}\}$ must be a positive definite matrix with unit diagonal elements (PDUDE) (Qian et al. 2008). Departing from Qian et al. (2008), here $\boldsymbol{T}$ is modeled by using the *hypersphere decomposition*, originally introduced by Rebonato and Jackel (1999) for modeling correlations for 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 $\boldsymbol{T}$ given by

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

where $\boldsymbol{L} = \{l_{r,s}\}$ is a lower triangular matrix with strictly positive diagonal entries. In step 2, each row vector $(l_{r,1}, \dots, l_{r,r})$ in $\boldsymbol{L}$ is modeled as the coordinate 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

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

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, π), the diagonal entry $l_{r,r}$ in $\boldsymbol{L}$ is strictly positive, thus guaranteeing that $\boldsymbol{T}$ 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 $\boldsymbol{T}$ must have unit diagonal elements. Thus, the matrix $\boldsymbol{T}$ under this parameterization is always a PDUDE. For illustration, consider the case with $m = 3$. In step 1, a 3×3 PDUDE

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

is decomposed as

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

In step 2, (l_{21}, l_{22}) are transformed into a 2D spherical coordinate system as

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

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

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

where $\theta_{r,s}$ can be calculated based on the following relations:

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

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).

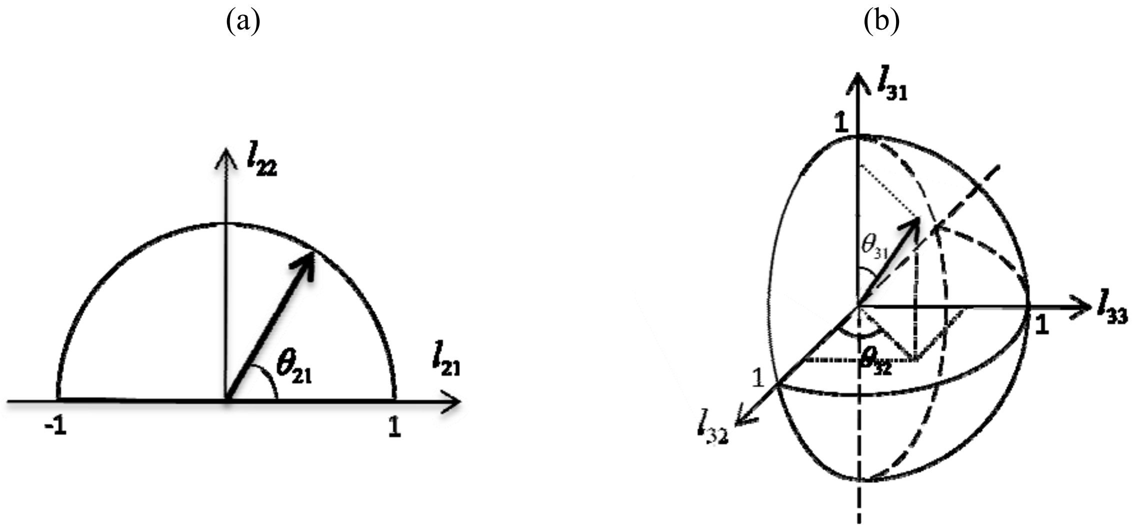

Figure 1. (a) Point (l_{21}, l_{22}) on the half unit circle. (b) Point (l_{31}, l_{32}, l_{33}) on the unit hemisphere.

The proposed parameterization has several major advantages. First, it turns the complicated PDUDE constraint on $\boldsymbol{T}$ into simple box constraints $\theta_{r,s} \in (0, \pi)$. Second, because $\theta_{r,s}$ take values in $(0, \pi)$, the entries in $\boldsymbol{T}$ can be either positive or negative, thus possible to capture various correlations across different categories. Third, any PDUDE matrix and $\boldsymbol{\Theta}$ has a one-to-one correspondence, i.e., a PDUDE matrix with any arbitrary structure can be parameterized using a set of $\boldsymbol{\Theta}$ values and any given $\boldsymbol{\Theta}$ always gives a PDUDE matrix.

For situations with multiple qualitative factors, instead of using the correlation function in (5), one can take a product form

$$cor\big(\varepsilon(\boldsymbol{w}_1), \varepsilon(\boldsymbol{w}_2)\big) = cor\big(\varepsilon(\boldsymbol{x}_1, \boldsymbol{z}_1), \varepsilon(\boldsymbol{x}_2, \boldsymbol{z}_2)\big) = \left[\prod_{j=1}^{J} \tau_{j,z_{j1},z_{j2}}\right] \exp\{-\sum_{i=1}^{I} \phi_i (x_{1i} - x_{2i})^2\}, \tag{13}$$

where each matrix $\boldsymbol{T}_j = \{\tau_{j,r,s}\}$ $(r, s = 1, \dots, m_j)$ is a PDUDE that is modeled by using the parameterization given by (4) and (5). This formulation can significantly reduce the number of parameters when the number of qualitative factors involved in the model is not small.

3. ESTIMATION AND PREDICTION

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

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

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

$$\begin{aligned} \widehat{\boldsymbol{\beta}} &= (\boldsymbol{F}^t\boldsymbol{R}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^t\boldsymbol{R}^{-1}\boldsymbol{y}, \\ \hat{\sigma}^2 &= \left(\boldsymbol{y} - \boldsymbol{F}\widehat{\boldsymbol{\beta}}\right)^t\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F}\widehat{\boldsymbol{\beta}})/n. \end{aligned} \tag{15}$$

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

$$(\widehat{\boldsymbol{\Phi}}, \widehat{\boldsymbol{\Theta}}) = \operatorname{argmin}_{(\boldsymbol{\Phi},\boldsymbol{\Theta})}\{n\log(\hat{\sigma}^2) + \log|\boldsymbol{R}|\}. \tag{16}$$

This optimization problem only involves box constraints $\theta_{r,s} \in (0, \pi)$ for $\boldsymbol{\Theta}$, much simpler than the positive definite constraints of the estimation procedure in Qian et al. (2008). Note that the problem in (16) can be solved by using standard non-linear optimization algorithms in R or Matlab.

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 $\boldsymbol{w}_0$ is

$$\hat{y}(\boldsymbol{w}_0) = \boldsymbol{f}^t(\boldsymbol{w}_0)\widehat{\boldsymbol{\beta}} + \hat{\boldsymbol{r}}_0^t\widehat{\boldsymbol{R}}^{-1}(\boldsymbol{y} - \boldsymbol{F}\widehat{\boldsymbol{\beta}}), \tag{17}$$

where $\hat{\boldsymbol{r}}_0 = (\widehat{cor}(\varepsilon(\boldsymbol{w}_0), \varepsilon(\boldsymbol{w}_1^0), \ldots, \widehat{cor}(\varepsilon(\boldsymbol{w}_0), \varepsilon(\boldsymbol{w}_n^0))^t$ and $\widehat{\boldsymbol{R}}$ is the estimated correlation matrix of $\boldsymbol{y}$. Similar to its counterpart for the standard Gaussian process model in (2) with quantitative factors (Santner et al. 2003; Fang et al. 2005), the EBLUP in (17) smoothly interpolates all the observed data points. The features of the function $y(\boldsymbol{w})$ can be visualized by plotting the estimated functional main effects and interactions of the predictor $\hat{y}(\boldsymbol{w})$. In estimating these functional ANOVA effects, for a qualitative factor, it suffices to average over the predicted response values for all the levels of that factor.

4. COMPUTATIONAL ISSUES

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

$$\boldsymbol{R} = \boldsymbol{T} \otimes \boldsymbol{H}, \tag{18}$$

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

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

and

$$\log|\boldsymbol{R}| = \log|\boldsymbol{T} \otimes \boldsymbol{H}| = \log(|\boldsymbol{T}|^{n_c} \cdot |\boldsymbol{H}|^{m}) = n_c \log|\boldsymbol{T}| + m\log|\boldsymbol{H}|. \tag{20}$$

Plugging (19) and (20) into the objective function in (16) can significantly simplify the complication of the computation of the problem. Given the close connection between the proposed model in Section 2 and the standard Gaussian process model with quantitative factors in (2), available numerical techniques for the latter such as those dealing with singularity of correlation matrices and inverting correlation matrices (Lophaven, Nielsen, and Sondergaard 2002a) can be readily adapted to the former.

5. EXAMPLES

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

a. 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 et al. 2003).

b. The exchangeable correlation method, denoted by **EC**. This method fits a single integrated Gaussian process model with qualitative and quantitative factors. For the quantitative factors, the Gaussian correlation function is used and for the qualitative factors, the exchangeable correlation function, i.e., $\tau_{r,s} = c \ (0 < c < 1)$ for $r \neq s$, is used. This correlation function was used by Joseph and Delaney (2007) and Qian et al. (2008).

c. The multiplicative correlation method, denoted by **MC**. The multiplicative correlation function (McMillian et al. 1999; Qian et al. 2008) has the following form: $\tau_{r,s} = exp\{-(\theta_r + \theta_s)I[r \neq s]\} \ (\theta_r, \theta_s > 0)$.

d. The proposed method discussed in Section 3, denoted by **UC,** which stands for unrestrictive correlation.

Some comparisons of these methods are worth mentioning. First, the methods in *b* to *d* all use a *single* Gaussian process model to analyze all available data, whereas the **IK** method, also called the independent analysis in Qian et al. (2008), fits distinct Gaussian process models in (2) to the data collected at different level combinations of the qualitative factors, thus ignoring possible correlations among different categories. Second, the **UC** method takes a far more flexible correlation structure than the **EC** and **MC** methods and thus is expected to produce superior results. Third, the correlation function for the qualitative factors of the **UC** method is essentially “structure free” and hence can capture both positive and negative cross-correlations across different categories, which cannot be modeled by the **MC** method.

From the modeling perspective, these models are interconnected to each other. 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 among the data from different categories. It is refined by the **EC** method based on a simple function for capturing cross-correlations among categories, which in turn is enhanced by the **MC** method using a more flexible correlation structure. The correlation function of the **UC** method drastically improves those of the **EC** and **MC** methods, rendering the most flexible model among all the methods. This viewpoint provides a logical basis for setting initial correlation parameter values for these methods. In particular, we found the following modeling fitting procedures work well in our numerical examples. (1) In analyzing an example with the **EC** method, one is recommended to first use the **IK** method to model the data and then use the estimated parameters to determine initial values of the roughness parameters when fitting the same data using the **EC** method. (2) When fitting an **MC** or **UC** model, initial values of the correlation parameters can be set based on the estimated parameters from a fitted **EC** model to the same data.

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 is known to take 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 the 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 the curves with $z_1 = 1$ and $z_1 = 3$, while the curve with $z_1 = 1$ is positively correlated with the curve with $z_1 = 3$.

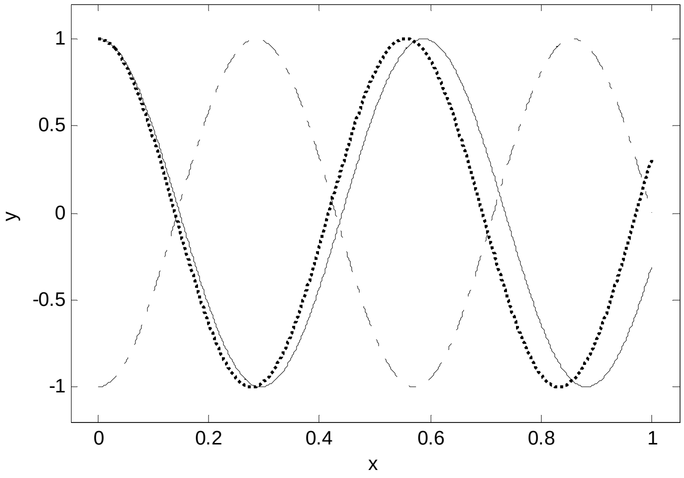

Figure 2. Three curves of the function values with $z_1 = 1$ (—), $z_1 = 2$ (— ·) and $z_1 = 3$ (···).

For each level of z_1, the training data were obtained by using a Latin hypercube design of eight runs for x_1 on [0, 1], and the testing data were then taken at 20 equally spaced points of 0, 1/19, 2/19, …, 1. The root mean squared errors (RMSEs) of the testing data were calculated for the four models to assess prediction accuracy. This procedure of data generation, modeling fitting and prediction accuracy assessment was repeated 100 times. Figure 3 compares the RMSEs of the four methods.

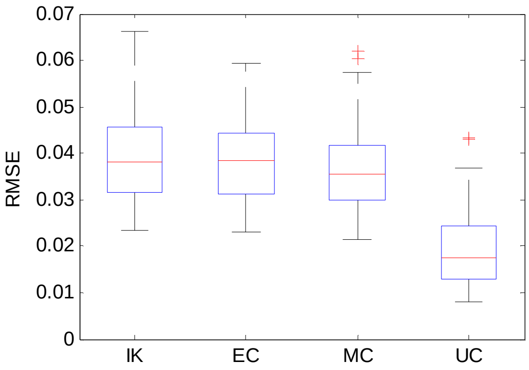

Figure 3. Boxplots of the RMSEs of the four methods for Example 5.1.

The mean values of the 100 RMSEs for the four models described in the beginning of Section 5 are 0.0392, 0.0391, 0.0365 and 0.0191, respectively, indicating that the **UC** method achieves the best prediction performance. Figure 4 depicts the boxplots of the cross-correlation parameters

$\tau_{1,2}$, $\tau_{1,3}$ and $\tau_{2,3}$ estimated in the **UC** and **MC** methods.

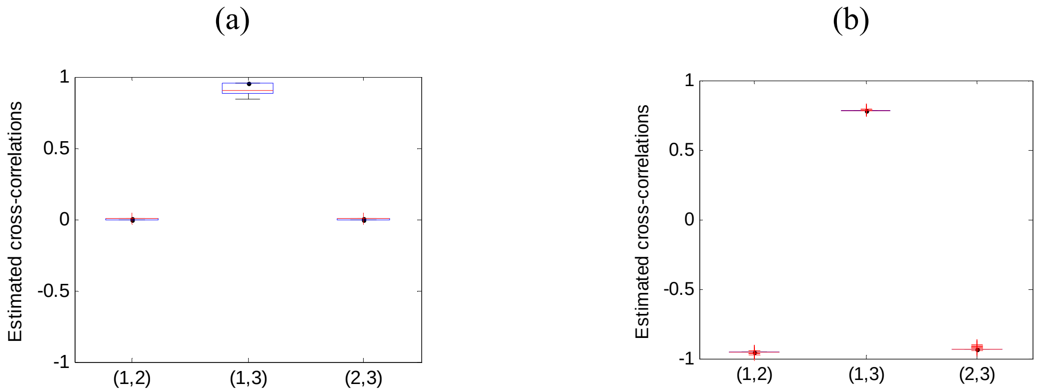

Figure 4. Boxplots of the estimated cross-correlation parameters of the **MC** method, given in (a)**,** and those of the **UC** method, given in (b), where (i, j) denotes the cross-correlation of the ith and jth z_1 levels.

As indicated by Figure 4, the **UC** method correctly captures both the positive and negative cross-correlations of this example, whereas the **MC** method fails to catch the two negative correlations of the second level of z_1 due to the positiveness constraint in the cross-correlation matrix of this method. This flexibility of the method **UC** explains its superior performance in this example.

5.2 An example from Han et al. (2009)

We now compare the **UC** method with some existing methods using an example from Han et al. (2009). This example uses the following quadratic function with one quantitative factor $x_1 \in [0,1]$ and one three-level qualitative factor z_1

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

Following Han et al. (2009), the training data and testing data were generated as follows. The first two levels of the function were evaluated at {0, 0.25, 0.5, 0.75, 1} and the third level at {0.5, 0.75, 1}. There were three processes to produce true quadratic curves for the testing. In each process, nine coefficients of the three testing function were 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 processes were (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 were obtained only for $z_1 = 3$ at points {0.5, 0.51, …, 1.00} and RMSEs were calculated. The above procedure of data generation, modeling fitting and assessment of prediction accuracy was repeated 30 times for all three processes with independent random samples of $\mathrm{b}_{ij}\mathrm{s}$ drawn each time.

Tables 1~3 compare the RMSEs of the four methods discussed in the beginning of Section 5 together with (1) **SHB**: a "surfacewise hierarchical bayes" predictor; (2) **KOH** method; (3) **HQQV**: the hierarchical Bayesian model proposed by Han et al. (2009). The results presented in the tables for the last three methods were extracted from the corresponding figures in Han et al. (2009).

Table 1. Some quantiles of the RMSEs for process 1 of the seven methods

Quantile	**SHB**	**KOH**	**HQQV**	**IK**	**EC**	**MC**	**UC**
25%	0.102	0.012	0.029	0.0951	2.73E-4	1.16E-3	2.51E-4
50%	0.120	0.034	0.030	0.0969	2.84E-4	1.21E-3	2.66E-4
75%	0.130	0.039	0.031	0.0984	3.02E-4	1.24E-3	4.37E-4

Table 2. Some quantiles of the RMSEs for process 2 of the seven methods

Quantile	**SHB**	**KOH**	**HQQV**	**IK**	**EC**	**MC**	**UC**
25%	0.090	0.008	0.028	0.0950	0.0177	0.0154	7.41E-3
50%	0.112	0.009	0.031	0.0952	0.0179	0.0155	7.49E-3
75%	0.126	0.010	0.032	0.0980	0.0180	0.0157	7.54E-3

Table 3. Some quantiles of the RMSEs for process 1 of the seven methods

Quantile	**SHB**	**KOH**	**HQQV**	**IK**	**EC**	**MC**	**UC**
25%	0.083	0.051	0.025	0.0928	5.01E-4	7.02E-4	3.70E-6
50%	0.098	0.058	0.026	0.0933	5.40E-4	7.26E-4	5.13E-6
75%	0.103	0.059	0.027	0.0936	5.68E-4	7.52E-4	6.58E-6

These tables suggest that the **UC** method consistently outperforms all competing methods in this example except that it gives an RMSE similar to that of the EC method for the 75% quantile in Table 1. In particular, the **UC** method gives far better results in Table 2 than all other methods for process 2, which involves both positive and negative cross-correlations among the levels of the qualitative factor.

5.3 A data center computer experiment

Here we reanalyzed the data of the data center computer experiment given in Qian et al. (2008) by using the **UC** method with the product correlation structure in (13). This experiment studies the thermal dynamics of an air-cooled data center system 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 this 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 2, 4, and 3 levels, respectively. The response variable, y, is the temperature at a selected location of the system. There are 67 observations and 24 level combinations of three qualitative factors; which implies that on average each level combination has less than 3 observations.

The form of $\boldsymbol{f}(\boldsymbol{w})^t\boldsymbol{\beta}$ used in Qian et al. (2008) is retained in our analysis. For this example, we take the product correlation function in (13). Following Qian et al. (2008), we use a leave-one-out cross-validation procedure to assess prediction accuracy of our method where the model correlation parameters and correlation matrices are obtained based on all data points and are not recomputed each time. The RMSE of our method is 1.70, similar to 1.88 of the method in Qian et al. (2008). However, our method gains significant computational efficiency. It took less than 20 seconds on a PC with Intel Core2 Duo CPU at 2.00GHz to fit the proposed method to this example, whereas fitting the method in Qian et al. (2008) with 400 iterations to the same example took more than 3 hours on a double-core PC running a Linux system.

6. SUMMARY AND DISCUSSION

We have proposed a new method for modeling computer experiments with qualitative and quantitative factors. This method uses a new parameterization for modeling the correlations among the levels of the quantitative factors, which simplifies complicated constrained optimization problems to easy to compute problems. Several examples clearly demonstrate that this method compares favorably with existing methods. This work deals with modeling computer experiments with these two types of factors. The interested reader is referred to Qian and Wu (2009) for a framework on design construction for such experiments. All computations used in this paper are done in MATLAB programs, which are available from the authors.

ACKNOWLEDGEMENTS

This work is supported by Grants #0545600, #0705206 and #0856222 from the National Science Foundation and a faculty reward from IBM.

REFERENCES

Fang, K. T., Li, R., and Sudjianto, A. (2005), *Design and Modeling for Computer Experiments*, New York: Chapman & Hall/CRC Press.

Graham, A. (1981), *Kronecker Products and Matrix Calculus With Applications*, Chichester, U.K.: Ellis Horwood Limited.

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.

Joseph, V. R., and Delaney, J. D. (2007), “Functionally Induced Priors for the Analysis of Experiments,” *Technometrics*, 49, 1-11.

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.

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.

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.

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.

Qian, P. Z. G. and Wu, C. F. J. (2009), "Sliced Space-Filling Designs," *Biometrika*, 96, 945-956.

Lophaven, S. N., Nielsen, H. B., and Sondergaard, J. (2002a), "Aspects of the Matlab Toolbox DACE," Technical Report IMM-REP-2002-13, Technical University of Denmark.

Lophaven, S. N., Nielsen, H. B., and Sondergaard, J. (2002b), *Matlab Kriging toolbox DACE*, Version 2.5, available at *http://www2.imm.dtu.dk/~hbn/dace/.*

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.

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.

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.

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.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003), *The Design and Analysis of Computer Experiments*, New York: Springer.

Wu, C. F. J., and Hamada, M. (2009), *Experiments: Planning, Analysis, and Optimization*, New York: Wiley.