Mixed-input Gaussian process emulators for computer experiments with a large number of categorical levels

Qiong Zhang
School of Mathematical and Statistical Sciences, Clemson University
Peter Chien
Department of Statistics, University of Wisconsin-Madison
Qing Liu
Wisconsin School of Business, University of Wisconsin-Madison
Li Xu and Yili Hong
Department of Statistics, Virginia Tech

Abstract

Computer models with both quantitative and qualitative inputs frequently arise in science, engineering and business. Mixed-input Gaussian process models have been used for emulating such models. The key in building this emulator is to accurately estimate the covariance between different categorical levels of the qualitative inputs. This problem is challenging when the number of categorical levels is large. We propose a sparse covariance estimation approach to estimating the covariance matrix with a large number of categorical levels for the mixed-input Gaussian process emulator. The effectiveness of this approach is illustrated with an application of IO operation modes in high performance computing systems.

Keywords: Computer experiments, qualitative and quantitative inputs, uncertainty quantification.

1 Introduction

1.1 Motivation

Gaussian process models with both quantitative and qualitative inputs have been widely used for emulation of computer models in many applications. A key for building such models is to accurately estimate the covariance structure among different categorical levels of the qualitative inputs. This problem poses challenges when the number of categorical levels is large. This work is motivated by a computer experiment of high performance computing (HPC) systems. This example aims at studying the input/output (IO) performance variability of HPC systems under different system configurations. The output of the experiment is the variability of the IO performance under different configurations. The inputs are the system configurations, including quantitative inputs like the CPU frequency and an important qualitative input, the IO operation mode. The IO operation mode contains 13 categorical levels, e.g., Initial write, Fwrite and Random write. Due to the large number of categorical levels, an unstructured covariance matrix of a mixed-input Gaussian process emulator is 13×13 with 91 parameters to be estimated. This becomes a challenging estimation problem. Accurately assessing the correlation between each pair of operation modes is critical for constructing a statistical emulator of the relationship between HPC performance variability and different system configurations. Therefore, this experiment requires emulators that can incorporate quantitative inputs as well as qualitative inputs with a large number of categorical levels.

1.2 Related literature and our contribution

Computer models are used in many fields to simulate real systems (Wu 2015). For example, computational fluid dynamic (CFD) models are popular for studying temperature management of data centers (Qian et al. 2008). Since running an expensive computer model under all input configurations is time-consuming, the Gaussian process emulator is often used as a proxy of the computer code for prediction, calibration, sensitivity analysis and uncertainty propagation purposes. See Sacks et al. (1989), Currin et al. (1991), Fang et al. (2005), and Santner et al. (2013) for references. The Gaussian process model consists

of a mean function and a covariance function. For computer models with only quantitative inputs, popular correlation functions include the Gaussian and Matérn. As reported in recent applications, computer models can contain both quantitative and qualitative inputs. See, for example, Qian et al. (2008), Han et al. (2009), Zhou et al. (2011), Zhang and Notz (2015) and Deng et al. (2017). To solve this issue, mixed-input Gaussian process emulators with both qualitative and quantitative inputs have been proposed by Qian et al. (2008) and Zhou et al. (2011) to incorporate a covariance model in the form of a product of the covariance matrix of the qualitative inputs and that of the quantitative inputs. Provided an accurate estimate of the covariance matrix for the qualitative inputs, the predictive performance of the mixed-input Gaussian process emulator is superior to that of an independent Gaussian process emulator for each categorical level combination. See Qian et al. (2008), Han et al. (2009), Zhou et al. (2011) and Zhang and Notz (2015). Qualitative inputs in computer experiments can also be modeled under some specific assumptions. For example, Gramacy and Taddy (2009) proposed treed Gaussian process, which can divide categorical input space and fit an independent Gaussian process for each partition. Roustant et al. (2018) developed group kernels to partition the categorical levels into different groups, and within each group the covariance between two levels is a constant. The covariance matrices for the qualitative factors in both Gramacy and Taddy (2009) and Roustant et al. (2018) are assumed to have some block structures. Zhang et al. (2019) developed a latent variable approach assuming the qualitative inputs can be mapped to some underlying numerical latent variables.

Different from the existing literature, we consider mixed-input Gaussian process emulation with a large number of categorical levels. There are two general situations that have a large number of categorical levels. The first situation comes from combining all possible level combinations of multiple qualitative inputs. Although each qualitative input may only contain a small number of levels, the total number of level combinations can be large. This situation can be handled by modeling the covariance matrix as a product of multiple smaller covariance matrices associated with each qualitative input. See Qian et al. (2008) and Zhou et al. (2011) for more details. The second situation, which is the focus of this work, is a single qualitative input with a large number of levels, as the multiple IO

operation modes we considered in our motivating example. In addition, a large number of qualitative alternatives appear in simulation optimization for discrete event simulation (Luo et al. 2015). From a methodological perspective, the major difference of these two situations lies in modeling and estimation of the Gaussian process covariance matrix. Unlike the first situation, the covariance matrix for a single qualitative input cannot be directly separated into several small covariance matrices in modeling. Thus, due to the large number of parameters, existing methods cannot work in estimating the covariance matrix of the second situation. In view of this difficulty, we propose a sparse covariance estimation approach for the mixed-input Gaussian process emulator to accommodate a large number of categorical levels. The proposed method uses the idea of a sparse estimation of the covariance matrix to shrink insignificant entries in the covariance matrix to zero. The method builds upon the majorization-minimization algorithm from Hunter and Li (2005) and the generalized gradient descent algorithm from Beck and Teboulle (2009) to estimate the covariance matrix under the L_1 regularization, and extends these algorithms to the new application of mixed-input Gaussian process emulation.

The remainder of the paper is organized as follows. Section 2 provides the modeling framework for computer experiments with both quantitative and qualitative inputs. Section 3 details the proposed approach. Section 5 applies the proposed method to the motivating example on studying performance variability in HPC systems. Section 4 gives additional numerical illustrations with synthetic datasets. Section 6 concludes with some discussion.

2 Gaussian process emulators with quantitative and qualitative inputs

We consider a computer model $y(\boldsymbol{x}, \boldsymbol{z})$ with quantitative inputs $\boldsymbol{x} = (x_1, \ldots, x_d)$ and qualitative inputs $\boldsymbol{z} = (z_1, \ldots, z_q)$. Let $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$ be all possible categorical level combinations of $\boldsymbol{z}$. For example, for the case of $q = 3$ with each qualitative input containing three levels, k would be $3^3 = 27$. The computer model output $y(\boldsymbol{x}, \boldsymbol{z})$ is assumed to be a multivariate

stochastic process with k components

$$y_1(\boldsymbol{x}), \ldots, y_k(\boldsymbol{x}), \tag{1}$$

where $y_i(\boldsymbol{x}) = y(\boldsymbol{x}, \boldsymbol{z}_i)$ for $i = 1, \ldots, k$. Following Qian et al. (2008) and Zhou et al. (2011), we model $y_i(\boldsymbol{x})$ as

$$y_i(\boldsymbol{x}) = \boldsymbol{f}_i(\boldsymbol{x})^\top \boldsymbol{\beta} + \varepsilon_i(\boldsymbol{x}), \text{ for } i = 1, \ldots, k, \tag{2}$$

where $\boldsymbol{f}_i(\boldsymbol{x})^\top \boldsymbol{\beta}$ is the mean function with known covariates

$$\boldsymbol{f}_i(\boldsymbol{x}) = \{f_{1,i}(\boldsymbol{x}), \ldots, f_{p,i}(\boldsymbol{x})\}^\top$$

and linear coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$. For example, one may express the model for $y_i(\boldsymbol{x})$ to be $y_i(\boldsymbol{x}) = \beta_i + \varepsilon_i(\boldsymbol{x})$, for $i = 1, \ldots, k$, where $\mathbf{f}_i(\boldsymbol{x})$ is a k-dimensional vector with the i-th entry loaded by one and the remaining entries filled by zeros. The second term $\varepsilon_i(\boldsymbol{x})$ in (2) is a zero-mean Gaussian process with a *product* covariance structure:

$$\text{cov}(\varepsilon_i(\boldsymbol{x}), \varepsilon_j(\boldsymbol{x}')) = \tau_{i,j}\phi(\boldsymbol{x}, \boldsymbol{x}'), \text{ for } i, j = 1, \ldots, k, \tag{3}$$

where $\tau_{i,j}$ is the covariance parameter quantifying the dependence between y_i and y_j in (1), and $\phi(\boldsymbol{x}, \boldsymbol{x}')$ in (3) is a correlation function modeling the dependence between the quantitative input values $\boldsymbol{x}$ and $\boldsymbol{x}'$. For the numerical examples in Section 4, we adopt the following correlation function from Sacks et al. (1989)

$$\phi(\boldsymbol{x}, \boldsymbol{x}') = \exp\Big(- \sum_{i=1}^{d} \theta_i |x_i - x_i'|^m \Big), \tag{4}$$

where $0 < m \leq 2$, and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$ are the correlation parameters.

Let $\boldsymbol{T}$ be the $k \times k$ covariance matrix of categorical levels with components $\{\tau_{ij}, i, j = 1, \ldots, k\}$ in (3). To guarantee that the multivariate stochastic process in (1) has a positive definite covariance structure, $\boldsymbol{T}$ is required to be a positive definite matrix. Different from the assumptions in Qian et al. (2008) and Zhou et al. (2011), we do not require the diagonal elements of $\boldsymbol{T}$ to be equal for more flexible modeling. The parameters to be estimated in

our model are $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ and $\boldsymbol{T}$. We adopt the maximum likelihood method to estimate these parameters and denote the estimators by $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{T}}$. Since $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\theta}}$ can be estimated using standard optimization procedures as in Qian et al. (2008) and Zhou et al. (2011), we focus on how to obtain $\hat{\boldsymbol{T}}$. This problem can be challenging for a large k. First, $\boldsymbol{T}$ contains $k \times (k-1)/2$ unknown parameters, which is large even with a moderate k. Second, $\hat{\boldsymbol{T}}$ needs to be positive definite. To meet this requirement, Qian et al. (2008) uses semi-definite programming and Zhou et al. (2011) adopts the hypersphere decomposition method. Unfortunately, these methods become inaccurate or infeasible when k is large. To overcome this difficulty, we propose a sparse covariance estimation approach in Section 3. The aim of the proposed approach is to shrink the insignificant entries of the covariance matrix to be zero, and hence reduce the number of parameters to be estimated.

3 A sparse covariance estimation approach for mixed-input Gaussian process

We develop a sparse covariance estimation approach to estimate the covariance matrix $\boldsymbol{T}$ for the categorical levels of computer model inputs. Our proposed approach is an extension of the sparse covariance estimation approach in Bien and Tibshirani (2011). The connection between our development and Bien and Tibshirani (2011) will be discussed in Remark 1.

3.1 Regularized likelihood function

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ denote the design points of the quantitative inputs $\boldsymbol{x}$ for the multivariate process model in (1). For the multivariate process in (1), $y_i(\boldsymbol{x}_j)$ may not be available for all $j = 1, \ldots, n$. Hence, we define a vector with all available outputs by $\boldsymbol{y}_i = \left\{y_i(\boldsymbol{x}_{j_1}), \ldots, y_i(\boldsymbol{x}_{j_{n_i}})\right\}^\top$, where $\{j_1, \ldots, j_{n_i}\} \subset \{1, \ldots, n\}$ are the indices of the associated design points. Combining all $\boldsymbol{y}_i$'s gives an output vector

$$\boldsymbol{y} = \left(\boldsymbol{y}_1^\top, \ldots, \boldsymbol{y}_k^\top\right)^\top \tag{5}$$

of length $N = \sum_{i=1}^{k} n_i$. We would like to point out that in many practical computer experiments it is unrealistic to assume that all the qualitative levels share the same design for quantitative inputs. Since computer experiments are expensive to run, the designs for quantitative inputs of each qualitative level are often chosen to be different to include more quantitative design points. This assumption was adopted in the development of designs for computer experiments with mixed inputs in Joseph et al. (2019) and and Deng et al. (2015).

Let $\boldsymbol{F}$ be an $N \times p$ matrix containing the linear covariates of $\boldsymbol{y}$. According to (2) and (3), the negative log-likelihood function of $\boldsymbol{y}$ with respect to $\boldsymbol{T}$ is proportional to

$$\log \left| \boldsymbol{A}^\top (\boldsymbol{T} \otimes \boldsymbol{\Phi}) \boldsymbol{A} \right| + (\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}})^\top \left[\boldsymbol{A}^\top (\boldsymbol{T} \otimes \boldsymbol{\Phi}) \boldsymbol{A} \right]^{-1} (\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}), \tag{6}$$

where $\otimes$ denotes the Kronecker product, $\boldsymbol{\Phi}$ is an $N \times N$ matrix based on a correlation function $\phi(\boldsymbol{x}, \boldsymbol{x}')$ and fixed correlation parameter $\hat{\boldsymbol{\theta}}$, $\boldsymbol{A}$ is an $(nk) \times N$ submatrix from an identity matrix of size nk associated with the output in (5), and $\hat{\boldsymbol{\beta}}$ is

$$\hat{\boldsymbol{\beta}} = \left\{ \boldsymbol{F}^\top \left[\boldsymbol{A}^\top (\boldsymbol{T} \otimes \boldsymbol{\Phi}) \boldsymbol{A} \right]^{-1} \boldsymbol{F} \right\}^{-1} \boldsymbol{F}^\top \left[\boldsymbol{A}^\top (\boldsymbol{T} \otimes \boldsymbol{\Phi}) \boldsymbol{A} \right]^{-1} \boldsymbol{y}. \tag{7}$$

Notice that $\boldsymbol{T} \otimes \boldsymbol{\Phi}$ represents the covariance matrix when the outputs observed from inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are available at all qualitative levels $i = 1, \ldots, k$. Therefore, $\boldsymbol{A}^\top (\boldsymbol{T} \otimes \boldsymbol{\Phi}) \boldsymbol{A}$ is the submatrix of $\boldsymbol{T} \otimes \boldsymbol{\Phi}$ representing the covariance matrix of the incomplete outputs in (5).

To capture the insignificant components in the covariance matrix, we add the L_1 penalty to the objective function in (6):

$$g(\boldsymbol{T}) = \log \left| \boldsymbol{A}^\top (\boldsymbol{T} \otimes \boldsymbol{\Phi}) \boldsymbol{A} \right| + (\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}})^\top \left[\boldsymbol{A}^\top (\boldsymbol{T} \otimes \Phi) \boldsymbol{A} \right]^{-1} (\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}) + \lambda \left\| \boldsymbol{P} \circ \boldsymbol{T} \right\|_1, \tag{8}$$

where $\boldsymbol{P}$ is an arbitrary $k \times k$ matrix with non-negative elements and $\circ$ denotes element-wise multiplication. One choice for $\boldsymbol{P}$ is a matrix with off-diagonal elements being ones. The L_1 penalty can shrink the insignificant entries in the covariance matrix to zero, and only keeping relatively significant entries.

Since (8) is neither concave or convex, it is challenging to solve. We develop an efficient

approach using the majorization-minimization algorithm from Hunter and Li (2005) and the generalized gradient descent algorithm from Beck and Teboulle (2009). These two algorithms are briefly reviewed in the Supplementary Material.

3.2 The proposed estimation approach

We now discuss our proposed approach to minimize the objective function in (8). Some concave/convex properties are desired to meet the requirements of the majorization-minimization and generalized gradient descent algorithms. The properties are described as follows and verified in Appendix B.

- The function $\log\left|\boldsymbol{A}^\top(\boldsymbol{T}\otimes\boldsymbol{\Phi})\boldsymbol{A}\right|$ in (8) is concave with regard to $\boldsymbol{T}$.
- The function $\log\left|\boldsymbol{A}^\top(\boldsymbol{T}\otimes\boldsymbol{\Phi})\boldsymbol{A}\right|$ can be majorized by its first order Taylor expansion on an arbitrary $k\times k$ matrix $\boldsymbol{T}_0$ as follows

$$\log\left|\boldsymbol{A}^\top(\boldsymbol{T}\otimes\boldsymbol{\Phi})\boldsymbol{A}\right| \le \log\left|\boldsymbol{A}^\top(\boldsymbol{T}_0\otimes\boldsymbol{\Phi})\boldsymbol{A}\right| + \operatorname{tr}\left\{\boldsymbol{B}(\boldsymbol{T}_0)\left[(\boldsymbol{T}-\boldsymbol{T}_0)\otimes\boldsymbol{\Phi}\right]\right\}, \tag{9}$$

  where $\boldsymbol{B}(\boldsymbol{T}) = \boldsymbol{A}\left[\boldsymbol{A}^\top(\boldsymbol{T}\otimes\boldsymbol{\Phi})\boldsymbol{A}\right]^{-1}\boldsymbol{A}^\top$.
- The function $(\boldsymbol{y}-\boldsymbol{F}\hat{\boldsymbol{\beta}})^\top\left[\boldsymbol{A}^\top(\boldsymbol{T}\otimes\boldsymbol{\Phi})\boldsymbol{A}\right]^{-1}(\boldsymbol{y}-\boldsymbol{F}\hat{\boldsymbol{\beta}})$ in (8) is convex with regard to $\boldsymbol{T}$.

According to (9), the majorization function of $g(\boldsymbol{T})$ in (8) is

$$f_{\boldsymbol{T}_0}(\boldsymbol{T}) = \log\left|\boldsymbol{A}^\top(\boldsymbol{T}_0\otimes\boldsymbol{\Phi})\boldsymbol{A}\right| + \operatorname{tr}\left\{\boldsymbol{B}(\boldsymbol{T}_0)\left[(\boldsymbol{T}-\boldsymbol{T}_0)\otimes\boldsymbol{\Phi}\right]\right\}$$

$$+ (\boldsymbol{y}-\boldsymbol{F}\hat{\boldsymbol{\beta}})^\top\left[\boldsymbol{A}^\top(\boldsymbol{T}\otimes\boldsymbol{\Phi})\boldsymbol{A}\right]^{-1}(\boldsymbol{y}-\boldsymbol{F}\hat{\boldsymbol{\beta}}) + \lambda\|\boldsymbol{P}\circ\boldsymbol{T}\|_1. \tag{10}$$

Note that $g(\boldsymbol{T}) \le f_{\boldsymbol{T}_0}(\boldsymbol{T})$ for all $\boldsymbol{T}$, and $g(\boldsymbol{T}_0) = f_{\boldsymbol{T}_0}(\boldsymbol{T}_0)$. By using the majorization-minimization method, the optimization problem in (8) is reduced to minimizing $f_{\boldsymbol{T}_0}(\boldsymbol{T})$ iteratively. Each iteration solves the optimization problem

$$\operatorname{argmin}_{\boldsymbol{T}\succ 0} L(\boldsymbol{T}) + \lambda\|\boldsymbol{P}\circ\boldsymbol{T}\|_1, \tag{11}$$

where

$$L(\boldsymbol{T}) = \mathrm{tr}\left\{\boldsymbol{B}(\boldsymbol{T}_0)(\boldsymbol{T}\otimes\boldsymbol{\Phi})\right\} + (\boldsymbol{y}-\boldsymbol{F}\hat{\boldsymbol{\beta}})^\top\left[\boldsymbol{A}^\top(\boldsymbol{T}\otimes\boldsymbol{\Phi})\boldsymbol{A}\right]^{-1}(\boldsymbol{y}-\boldsymbol{F}\hat{\boldsymbol{\beta}}),$$

and the constraint $\boldsymbol{T}\succ 0$ guarantees that the solution $\hat{\boldsymbol{T}}$ is positive definite.

The first order derivative of $L(\boldsymbol{T})$ is $dL(\boldsymbol{T})/d\boldsymbol{T} = \tilde{\boldsymbol{B}}_0 - \tilde{\boldsymbol{Y}}^\top\boldsymbol{\Phi}\tilde{\boldsymbol{Y}}$. The (i,j)th entry of the $k\times k$ matrix $\tilde{\boldsymbol{B}}_0$ is $\mathrm{tr}(\{\boldsymbol{B}(\boldsymbol{T}_0)\}_{ij}\boldsymbol{\Phi})$, where $\{\boldsymbol{B}(\boldsymbol{T}_0)\}_{ij}$ is the (i,j)th block after dividing $\boldsymbol{B}(\boldsymbol{T}_0)$ into $k\times k$ blocks of an $n\times n$ submatrix. The (i,j)th entry of the $k\times n$ matrix $\tilde{\boldsymbol{Y}}$ is the $(i\times n+j-n)$th entry of the vector $\boldsymbol{A}\boldsymbol{H}(\boldsymbol{T})\boldsymbol{y}$ of length nk, where $\boldsymbol{H}(\boldsymbol{T})$ is defined as in the Supplementary Material. The generalized gradient descent algorithm is applied to (11). The tth step computes

$$\boldsymbol{T}_t = \mathrm{argmin}_{\boldsymbol{T}\succ 0}\left\{(2\delta_t)^{-1}\left\|\boldsymbol{T}-\boldsymbol{T}_{t-1}+\delta_t\frac{dL(\boldsymbol{T})}{d\boldsymbol{T}}\Big|_{\boldsymbol{T}=\boldsymbol{T}_{t-1}}\right\|^2 + \lambda\|\boldsymbol{P}\circ\boldsymbol{T}\|_1\right\}, \qquad (12)$$

where δ_t is the step size.

In our implementation, the outer loop uses the majorization-minimization algorithm, and the inner loop uses the generalized gradient decent algorithm. As in Bien and Tibshirani (2011), the constraint $\boldsymbol{T}\succ 0$ of (12) is tightened to $\boldsymbol{T}\succeq\delta I_k$ for some $\delta>0$, which can be computed as the smallest eigenvalue of $\boldsymbol{T}_{t-1}$. Then, we follow the alternating direction method of multipliers in Appendix 3 of Bien and Tibshirani (2011) to solve the optimization problem in (12). The step size in (12) is selected using the backtracking procedure (Wright and Nocedal 1999). In our numerical examples, a single outer or inner loop of the algorithm converges within 100 iterations.

Remark 1. *There is a connection between our problem setting and that in Bien and Tibshirani (2011). Our problem setting is equivalent to the problem in Bien and Tibshirani (2011) when there is no quantitative input and only one qualitative input. Without quantitative inputs, (2) becomes*

$$y_i = \beta_i + \varepsilon_i, \text{ for } i = 1,\ldots,k,$$

and the covariance matrix of this model is reduced to $\boldsymbol{T}$. Therefore, the covariance estimation approach in Bien and Tibshirani (2011) can be applied to this case directly by providing

a sample covariance matrix.

If quantitative inputs are available, our problem can be converted to the sparse covariance estimation problem under a special case that all n design points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ of the quantitative inputs are available for all categorical levels. Under this special case, the matrix $\boldsymbol{A}$ is reduced to an identity matrix of size $n \times k$. And the key terms in (8) are simplified to

$$\log \left| \boldsymbol{A}^\top (\boldsymbol{T} \otimes \boldsymbol{\Phi}) \boldsymbol{A} \right| = k \log |\boldsymbol{\Phi}| + n \log |\boldsymbol{T}|$$

and

$$(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}})^\top \left[\boldsymbol{A}^\top (\boldsymbol{T} \otimes \boldsymbol{\Phi}) \boldsymbol{A} \right]^{-1} (\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}}) = \operatorname{tr}(\boldsymbol{T}^{-1} \boldsymbol{U} \boldsymbol{\Phi}^{-1} \boldsymbol{U}^\top),$$

where the (i, j)th element of the $n \times k$ matrix $\boldsymbol{U}$ is $y_i(\boldsymbol{x}_j) - \boldsymbol{f}_i(\boldsymbol{x}_j)^\top \hat{\boldsymbol{\beta}}$. Thus, the objective function in (8) becomes

$$\log |\boldsymbol{T}| + \operatorname{tr}(\boldsymbol{T}^{-1} n^{-1} \boldsymbol{U}^\top \boldsymbol{\Phi}^{-1} \boldsymbol{U}) + \lambda \|\boldsymbol{P} \circ \boldsymbol{T}\|_1. \tag{13}$$

By treating $n^{-1} \boldsymbol{U}^\top \boldsymbol{\Phi}^{-1} \boldsymbol{U}$ as a sample covariance matrix of the qualitative inputs, this problem now reduces to the sparse covariance estimation problem in Bien and Tibshirani (2011). However, the algorithms developed for the simplified situation in Bien and Tibshirani (2011) is not directly available as a solution to our problem setting. Replacing an identity matrix by the matrix $\boldsymbol{A}$ increased the complexity of the development and implementation of our algorithms. In particular, the properties in Section 3.2 need to be verified under this new problem setting (given in Appendix B), which leads to a substantial modification in our estimation procedure.

Remark 2. *The penalty parameter λ can be chosen by a cross-validation procedure. In our numerical examples, we conduct two-fold cross-validation to choose the penalty parameter λ from 0.001 to 5.*

4 Simulation Study

In our numerical study, we compare the following methods for the prediction performance of Gaussian process emulators with qualitative and quantitative inputs:

1. The proposed method with P in (8) is a symmetric matrix with diagonal entries 0, and off-diagonal entries 1. The penalty parameter λ is chosen by two-fold cross-validation as stated in Remark 2. This method is denoted by **RC**.

2. The individual Kriging method in Zhou et al. (2011) that fits independent Gaussian process emulator for each categorical level combination. This method is denoted by **IC**.

3. The unconstrained correlation function for the qualitative inputs in Zhou et al. (2011). This method is denoted by **UC**.

4. The exchangeable correlation function in Qian et al. (2008) and Zhou et al. (2011).

For the EC approach compared in the case study, we consider two different variations of this approach: (1) the level combinations of multiple qualitative inputs are combined as a single qualitative input, denoted by **EC**; (2) separable exchangeable correlation functions for each qualitative input, denoted by **ECsep**. Notice that **ECsep** is the same as **EC** if there is only one categorical factor such as case in Section 5.

We provide a numerical study based on a synthetic test function to further demonstrate the effectiveness of the proposed approach. Consider the borehole function (Morris et al. 1993) that models the water flow rate through a borehole. As in Fang et al. (2005), the response is first transformed by logarithm:

$$y = \log\left\{\frac{2\pi x_1 x_5}{\log(x_6/x_2)\left[1+\frac{2x_3x_1}{\log(x_6/x_2)x_2^2x_7}+\frac{x_1}{x_4}\right]}\right\}, \tag{14}$$

where the ranges and units of the inputs $x_1 \sim x_7$ are given in Table 1. Additional information about the inputs can be found in Morris et al. (1993). As in Zhou et al. (2011), we treat inputs x_5, x_6, and x_7 as the qualitative inputs. Table 2 specifies the number of levels for each qualitative input and the number of categorical level combinations for four different cases. The categorical levels of inputs x_5, x_6, and x_7 are chosen as equally spaced points within their ranges specified in Table 1. For example, the three levels for x_5 are: 290, 340, and 390.

Table 1: Ranges of the inputs $x_1 \sim x_7$ in the Borehole function

Variable	Range	Unit	Variable	Range	Unit
x_1	63070-115600	m^2/yr	x_5	290-390	M
x_2	0.05-0.15	M	x_6	100-50000	M
x_3	1120-1680	M	x_7	9855-12045	m/yr
x_4	63.1-116	m^2/yr			

Table 2: The number of levels of the three qualitative inputs in the Borehole function

Case	x_5	x_6	x_7	k
1	3	3	3	27
2	2	2	10	40
3	2	2	15	60
4	2	2	20	80

The datasets are generated as follows. A Latin hypercube design (McKay et al. 1979) with eight runs and four columns is generated for the quantitative inputs x_1–x_4. For each categorical level, we randomly choose four of the eight runs as the input points for the training dataset. In this way, the input point within each category are not necessarily the same. The outputs of these input points within each category are calculated using (14). By combining the samples from all k categorical levels, the sample sizes of the training datasets are 4×27, 4×40, 4×60 and 4×80 in the four cases, respectively. To evaluate the predictive performances of different approaches, we generate test datasets with a Latin hypercube design containing 100 runs as the input points of each categorical level.

Using the training datasets, Gaussian process emulators are fitted by RC, IC, UC, EC, and ECsep, and the correlation function in (4) with $m = 1$ for the quantitative inputs. Using the test datasets, we compute the mean and standard deviation of the mean squared errors (MSE) over all k categorical levels. Different values of the tuning parameter λ are used for RC. The results are given in Table 3. As an example, Figure 1 presents the correlation matrix estimates for $k = 27$, where (a), (b) and (c) provide the estimates from UC, RC with $\lambda = 0.01$, and RC with $\lambda = 1$, respectively. We observe that the estimated correlation matrix from UC is much denser than that from RC.

We comment on the results in Table 3. For $k = 27$, the best result is generated by UC. The proposed method shows larger advantages than UC when k is large. As k increases, the performance of IC becomes more and more competitive. The inferior

performances of EC and ECsep show that the exchangeable correlation structure does not fit the data generated from the borehole function. Also, the result indicates that inaccurate estimates of covariance entries in $\boldsymbol{T}$ (i.e., UC, EC, ECsep) may not necessarily improve the prediction performance for the Gaussian process with mixed inputs. As mentioned in Remark 2, the penalty parameter λ is chosen from two-fold cross-validation with a candidate set containing 19 different values, and the computation time includes the process of implementing with different values of λ's. If the value of λ is pre-specified, the computing time is comparable with UC. The code of this paper is provided via Codeocean https://codeocean.com/capsule/4782881/tree.

Table 3: Results of the example in Section 4.1. mean: average MSE over all k categorical levels. sd: standard deviation of the MSEs over all k categorical levels. time: computational time in minutes. The computing time of RC includes a cross-validation procedure for choosing the penalty parameter λ.

Case	k		RC (λ)	IC	UC	EC	ECsep
1	27	mean	0.0241 ($\lambda = 0.001$)	0.0667	0.0363	0.0521	0.0520
		sd	0.0048	0.0779	0.0689	0.0004	0.0005
		time	0.8909	0.0647	0.0028	0.0014	0.0018
2	40	mean	0.0243 ($\lambda = 0.001$)	0.0675	0.0317	0.0519	0.0518
		sd	0.0040	0.0691	0.0565	0.0002	0.0003
		time	2.5200	0.0995	0.0032	0.0022	0.0026
3	60	mean	0.0240($\lambda = 0.004$)	0.0727	0.0286	0.0518	0.0517
		sd	0.0035	0.0737	0.0464	0.0002	0.0002
		time	10.562	0.1355	0.0062	0.0047	0.0050
4	80	mean	0.0237($\lambda = 0.001$)	0.0711	0.0270	0.0517	0.0968
		sd	0.0027	0.0702	0.0402	0.0001	0.0002
		time	25.954	0.2048	0.0115	0.0151	0.0146

5 Application: an example of performance variability in HPC systems

We consider an example for studying high performance computing (HPC) systems. Cameron et al. (2019) studied the input/output (IO) performance variability of HPC systems as regards different system configurations, in which the IOzone benchmark (Norcott 2019) is used to measure IO performance under various configurations. The output variable in the

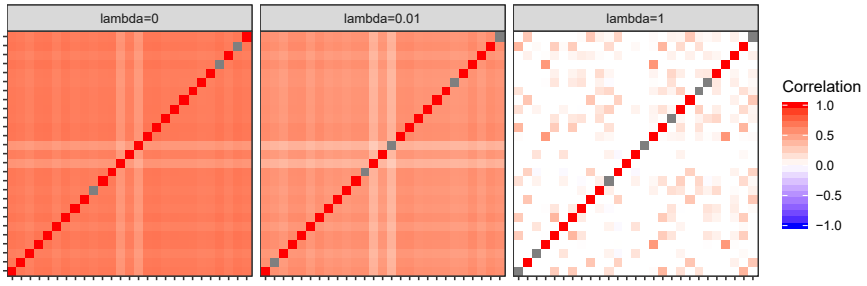

Figure 1: The correlation estimators of the borehole example with $k = 27$: (a) UC; (b) RC with $\lambda = 0.01$; (c) RC with $\lambda = 1$.

HPC dataset is the IO throughput variability measure. The input of the HPC dataset contains both quantitative and qualitative variables. We use a subset of the data in Cameron et al. (2019), which consists of two quantitative variables, the CPU frequency and the number of threads, and one categorical variable, the IO operation mode. The CPU frequency has 15 distinct values, which are $1.2, 1.4, 1.5, \ldots, 3.0$, and the number of threads has nine distinct values which are $1, 2, 4, 8, \ldots, 256$. The IO operation mode has 13 levels, which are Initial write, Fwrite, Random write, Pwrite, Rewrite, Randomread, Mixed workload, Stride read, Reverse read, Re-read, Fread, Pread, and Read. A detailed description for different modes are in Norcott (2019), and a brief description of the IO operation modes is provided in Table 4. In total, we have $13 \times 15 \times 9 = 1755$ system configurations.

Table 4: IO Operation Modes in IOZone for the HPC Example

Mode	Description
Initial write	An initial test and measures the performance of writing a new file
Rewrite	Measures the performance of writing an already existing file
Read	Measures the performance of reading an existing file.
Re-read	Measures the performance of reading a file which is read recently
Reverse Read	Measures the performance of reading a file backwards.
Stride read	Measures the performance of reading a file with a stride behavior
Random read	Measures the performance of reading a file with accesses being made to random locations within the file
Mixed workload	Measures the performance of reading and writing a file with accesses being made to random locations within the file
Random write	Measures the performance of writing a file with accesses being made to random locations within the file.
Fwrite	Measures the performance of writing a file using fwrite() function.
Fread	Measures the performance of reading a file using fread() function
Pwrite	Measures the performance of writing a file using pwrite() function.
Pread	Measures the performance of writing a file using pread() function.

We randomly select eight quantitative input points for each qualitative level. Then the size of the training dataset is 8×13, and the remaining data points are included in the testing dataset to compute MSEs. Table 5 provides the prediction results of different approaches. Table 5 indicates that RC with a cross-validation selected penalty parameter gives the smallest MSE value among all alternative methods.

Table 5: Results of the example in Section 5. mean: average MSE over 13 categorical levels; sd: standard deviation over 13 categorical levels; time: computing time in minutes. The computing time of RC includes the cross-validation procedure for choosing the penalty parameter λ.

	RC(λ)	IC	UC	EC
mean	0.0905 ($\lambda = 0.12$)	0.3299	0.1108	0.1023
sd	0.0967	0.2826	0.0672	0.0781
time	152.800	0.0083	0.0019	0.0011

The correlation matrix given by the proposed method RC are depicted in Figure 2. We discuss some interesting findings. In general, we observe moderate to strong correlations among most pairs of performance variability of IO operation modes, which could be caused by the fact that all operations are carried out for the same file size and record size, and are likely be affected by the cache. The write operations (i.e., Initial write, Fwrite, Random write, Pwrite, Rewrite) have the strongest correlations as shown in Figure 2. The Initial write and Fwrite have the largest correlation because essentially both operations need to write a new file. The read operations (e.g., Stride read, Reverse read, etc.) display correlations around 0.5. The Mixed workload is a mix of read and write operations and shows moderate correlation (i.e., around 0.5) to all other modes. Interestingly, read and write operations tend to be weakly correlated (i.e., around 0.3) because those two types of operations are quite different.

Existing literature, e.g., Cameron et al. (2019), assumes that the performance variabilities of different operation modes are uncorrelated and thus separate analyses for the data collected from each operation mode are often used. From a new perspective, our work studies the correlations among different IO operation modes. Our results reveal that there exist correlations and the degree of correlations vary as well. These results shed new light on the modeling and analysis of performance variability data.

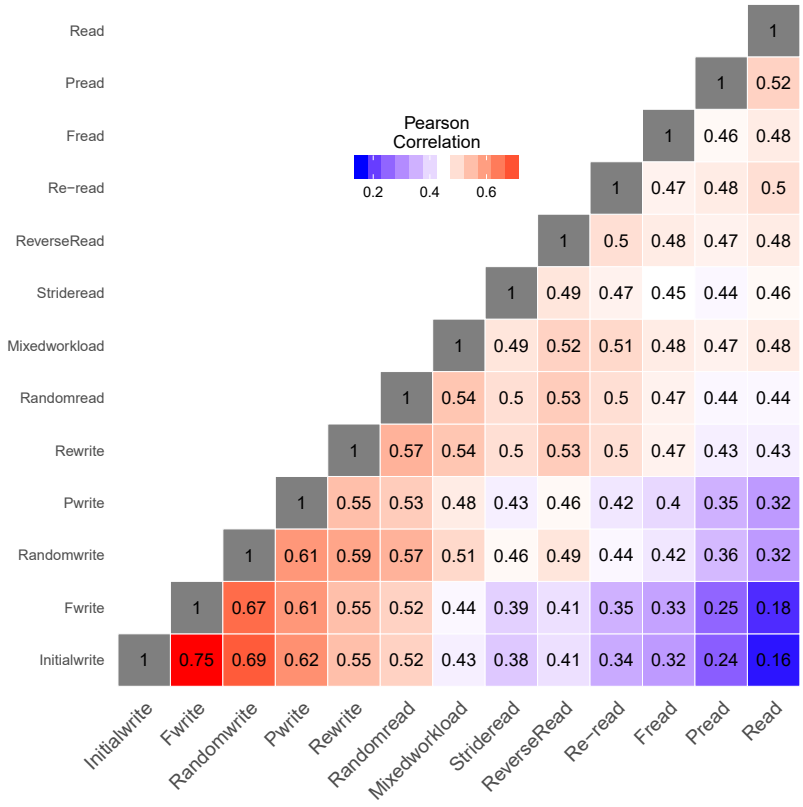

Figure 2: The correlation matrix from RC ($\lambda = 0.12$) in Section 5

6 Discussion

We have proposed a new method for emulating computer models with both quantitative and categorical inputs. Through simulation study and a real application on studying performance variability in HPC systems, this method is shown to be effective in mixed-input Gaussian process emulator when the number of categorical levels is large. When there are multiple categorical inputs, the covariance matrix of these categorical inputs from the proposed method is assumed to be completely unstructured, whereas Qian et al. (2008) and Zhou et al. (2011) both use a fully separable covariance structure for different categorical inputs. The proposed approach relaxes the separable assumptions. However, a drawback of the unstructured covariance matrix is the need to estimate a large number of unknown parameters.

Here are some possible directions for future research. First, our current approach requires that there are observations available for all possible qualitative level combinations,

which may not be always feasible. Extending our method to more general situations needs further research. For example, one may develop a composite approach by combining the covariance matrix from our approach and the separable covariance in Qian et al. (2008) and Zhou et al. (2011) together to impute the covariance parameters for unobserved qualitative levels. Second, our adopted product correlation function was successfully used in Qian et al. (2008) and Zhou et al. (2011). A possible extension is to accommodate non-separable covariance models, e.g., Fricker et al. (2013). The non-separable covariance structure will create new challenges in parameter estimation as it will involve many more parameters than the separate case. Third, designs for computer experiments with qualitative and quantitative inputs have been studied by Qian and Wu (2009), Qian (2012), and Deng et al. (2013), among many others. But the link between design and modeling is still not clear. One might be interested in creating new designs that suitable for our proposed modeling framework. Slicing is still important but another key is to best overlap points for the different level combinations of the qualitative factors. It is possible to develop a new sequential design strategy for computer experiments with mixed inputs by building an interface between design and modeling. This strategy can benefit simulation optimization applications with a large number of alternatives (Luo et al. 2015). In addition, it would be valuable to evaluate the covariance matrix estimated under the L_2 penalty, as well as the precision matrix estimated under different penalty options.

Acknowledgements

The authors would like to thank the Editor, Associate Editor and two reviewers for their comments and suggestions that have led to improvement of this work. Chien is supported by NSF Grant DMS 1564376. The work by Xu and Hong was partially supported by funds from NSF under Grant CNS-1565314 and CNS-1838271 to Virginia Tech.

About the authors

Qiong Zhang is an Assistant Professor in Statistics at Clemson University, Clemson, SC.

Peter Chien is a Professor in Statistics at the University of Wisconsin-Madison.

Qing Liu is an Associate Professor in Marketing at the University of Wisconsin-Madison, WI.

Li Xu is a fourth year PhD student in the department of statistics at Virginia Tech.

Yili Hong is an Associate Professor of Statistics at Virginia Tech. His research interests include machine learning and engineering applications, reliability analysis, and spatial statistics.

References

Beck, A. and Teboulle, M. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems." *SIAM Journal on Imaging Sciences*, 2(1):183–202 (2009).

Bien, J. and Tibshirani, R. J. "Sparse estimation of a covariance matrix." *Biometrika*, 98:807–820 (2011).

Cameron, K. W., Anwar, A., Cheng, Y., Xu, L., Li, B., Ananth, U., Bernard, J., Jearls, C., Lux, T., Hong, Y., Watson, L. T., and Butt, A. R. "MOANA: Modeling and Analyzing I/O Variability in Parallel System Experimental Design." *IEEE Transactions on Parallel and Distributed Systems*, 1–1 (2019).

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. "Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments." *Journal of the American Statistical Association*, 86(416):953–963 (1991).

Deng, X., Hung, Y., and Lin, C. D. "Design for computer experiments with qualitative and quantitative factors." *Statistica Sinica*, 1567–1581 (2013).

—. "Design for computer experiments with qualitative and quantitative factors." *Statistica Sinica*, 1567–1581 (2015).

Deng, X., Lin, C. D., Liu, K.-W., and Rowe, R. "Additive Gaussian Process for Computer Models With Qualitative and Quantitative Factors." *Technometrics* (2017).

Fang, K.-T., Li, R., and Sudjianto, A. *Design and modeling for computer experiments*. CRC Press (2005).

Fricker, T. E., Oakley, J. E., and Urban, N. M. "Multivariate Gaussian process emulators with nonseparable covariance structures." *Technometrics*, 55(1):47–56 (2013).

Gramacy, R. B. and Taddy, M. "Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an R package for treed Gaussian process models." *University of Cambridge Statistical Laboratory Tech. Rep* (2009).

Han, G., Santner, T. J., Notz, W. I., and Bartel, D. L. "Prediction for computer experiments having quantitative and qualitative input variables." *Technometrics*, 51(3):278–288 (2009).

Hunter, D. R. and Li, R. "Variable selection using MM algorithms." *Annals of statistics*, 33(4):1617–1642 (2005).

Joseph, V. R., Gul, E., and Ba, S. "Designing computer experiments with multiple types of factors: The MaxPro approach." *Journal of Quality Technology*, 1–12 (2019).

Luo, J., Hong, L. J., Nelson, B. L., and Wu, Y. "Fully sequential procedures for large-scale ranking-and-selection problems in parallel computing environments." *Operations Research*, 63(5):1177–1194 (2015).

McKay, M. D., Conover, W. J., and Beckman, R. J. "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code." *Technometrics*, 21:239–245 (1979).

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

Norcott, W. D. "IOzone Filesystem Benchmark." http://www.iozone.org/ (2019).

Qian, P. Z. G. "Sliced Latin hypercube designs." *Journal of the American Statistical Association*, 107(497):393–399 (2012).

Qian, P. Z. G. and Wu, C. F. J. "Sliced space-filling designs." *Biometrika*, 96(4):945–956 (2009).

Qian, P. Z. G., Wu, H., and Wu, C. F. J. "Gaussian process models for computer experiments with qualitative and quantitative factors." *Technometrics*, 50(3):383–396 (2008).

Roustant, O., Padonou, E., Deville, Y., Clément, A., Perrin, G., Giorla, J., and Wynn, H. "Group kernels for Gaussian process metamodels with categorical inputs." *arXiv preprint arXiv:1802.02368* (2018).

Sacks, J., Schiller, S. B., and Welch, W. J. "Designs for computer experiments." *Technometrics*, 31(1):41–47 (1989).

Santner, T. J., Williams, B. J., and Notz, W. I. *The design and analysis of computer experiments*. Springer Science & Business Media (2013).

Wright, S. J. and Nocedal, J. *Numerical optimization*, volume 2. Springer New York (1999).

Wu, C. F. J. "Post-fisherian experimentation: from physical to virtual." *Journal of the American Statistical Association*, 110(510):612–620 (2015).

Zhang, Y. and Notz, W. I. "Computer experiments with qualitative and quantitative variables: a review and reexamination." *Quality Engineering*, 27(1):2–13 (2015).

Zhang, Y., Tao, S., Chen, W., and Apley, D. W. "A latent variable approach to Gaussian process modeling with qualitative and quantitative factors." *Technometrics*, 1–12 (2019).

Zhou, Q., Qian, P. Z. G., and Zhou, S. "A simple approach to emulation for computer models with qualitative and quantitative factors." *Technometrics*, 53(3):266–273 (2011).

Appendix A Majorization-minimization and generalized gradient descent algorithms

We introduce two useful algorithms: majorization-minimization and generalized gradient descent. The majorization-minimization method Hunter and Li (2005) minimizes a function $g(x)$ using its majorization function $f_{x_0}(x)$ given as $g(x) \leq f_{x_0}(x)$ for all x and $g(x_0) = f_{x_0}(x_0)$. Starting with an appropriate initial solution x_0, this method searches for the minimum of $g(x)$ by repeatedly solving $x_t = \text{argmin}_x f_{x_{t-1}}(x)$.

The generalized gradient descent algorithm Beck and Teboulle (2009) solves the optimization problem

$$\min \{L(x) + p(x) \mid x \in \mathcal{C}\}, \tag{15}$$

where $L(x)$ is a differentiable convex function, $p(x)$ is a non-differentiable convex function, and $\mathcal{C}$ is the feasible region. Given an initial solution $x_0 \in \mathcal{C}$, the algorithm approximates $L(x)$ by the following quadratic function

$$L(x_0) + (x - x_0)^\top \frac{dL(x)}{dx}\Big|_{x=x_0} + (2\delta)^{-1}\|x - x_0\|^2 \propto (2\delta)^{-1}\left\|x - \left(x_0 - \delta \frac{dL(x)}{dx}\Big|_{x=x_0}\right)\right\|^2,$$

where δ is the step size. Starting from x_0, the algorithm iteratively performs the following calculation:

$$x_t = \text{argmin}_{x\in\mathcal{C}} \left\{(2\delta)^{-1}\left\|x - \left(x_{t-1} - \delta \frac{dL(x)}{dx}\Big|_{x=x_{t-1}}\right)\right\|^2 + p(x)\right\} \tag{16}$$

until convergence.

Appendix B Verification of concave/convex conditions

Let $f(\boldsymbol{T}) = \log|\boldsymbol{A}^\top(\boldsymbol{T} \otimes \boldsymbol{\Phi})\boldsymbol{A}|$. The first and second order differential forms of $f(\boldsymbol{T})$ are

$$df(\boldsymbol{T}) = \text{tr}\{\boldsymbol{B}(\boldsymbol{T})(d\boldsymbol{T} \otimes \boldsymbol{\Phi})\} \tag{17}$$

and

$$d^2 f(\boldsymbol{T}) = -\text{tr}\{\boldsymbol{B}(\boldsymbol{T})(d\boldsymbol{T} \otimes \boldsymbol{\Phi})\}^2, \tag{18}$$

which shows that $f(\boldsymbol{T})$ is concave. By expressing the first order Taylor expansion of $f(\boldsymbol{T})$, we also show that (9) is the majorized function of $f(\boldsymbol{T})$.

Let $h(\boldsymbol{T}) = (\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}})^\top \{\boldsymbol{A}^\top(\boldsymbol{T} \otimes \boldsymbol{\Phi})\boldsymbol{A}\}^{-1}(\boldsymbol{y} - \boldsymbol{F}\hat{\boldsymbol{\beta}})$ and $\boldsymbol{H}(\boldsymbol{T}) = \{\boldsymbol{A}^\top(\boldsymbol{T} \otimes \boldsymbol{\Phi})\boldsymbol{A}\}^{-1} - \{\boldsymbol{A}^\top(\boldsymbol{T} \otimes \boldsymbol{\Phi})\boldsymbol{A}\}^{-1}\boldsymbol{F}(\boldsymbol{F}^\top\{\boldsymbol{A}^\top(\boldsymbol{T} \otimes \boldsymbol{\Phi})\boldsymbol{A}\}^{-1}\boldsymbol{F})^{-1}\boldsymbol{F}^\top\{\boldsymbol{A}^\top(\boldsymbol{T} \otimes \boldsymbol{\Phi})\boldsymbol{A}\}^{-1}$. Then express the target function as

$$h(\boldsymbol{T}) = \boldsymbol{y}^\top \boldsymbol{H}(\boldsymbol{T})\boldsymbol{y}.$$

The first and second order differential forms of $h(\boldsymbol{T})$ are

$$dh(\boldsymbol{T}) = -\boldsymbol{y}^\top \boldsymbol{H}(\boldsymbol{T})\boldsymbol{A}^\top(d\boldsymbol{T} \otimes \boldsymbol{\Phi})\boldsymbol{A}\boldsymbol{H}(\boldsymbol{T})\boldsymbol{y}$$

and

$$d^2 h(\boldsymbol{T}) = \boldsymbol{y}^\top \boldsymbol{H}(\boldsymbol{T})\boldsymbol{A}^\top(d\boldsymbol{T} \otimes \boldsymbol{\Phi})\boldsymbol{A}\boldsymbol{H}(\boldsymbol{T})\boldsymbol{A}^\top(d\boldsymbol{T} \otimes \boldsymbol{\Phi})\boldsymbol{A}\boldsymbol{H}(\boldsymbol{T})\boldsymbol{y}.$$

Thus, $h(\boldsymbol{T})$ is convex.

Appendix C Additional Numerical Results

Assume that the design for the quantitative inputs $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ are the same for all the qualitative levels. Consider predicting at $\boldsymbol{x}_0$ for the jth qualitative level. Assume $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ are both known. Then the predicted value for $y_j(\boldsymbol{x}_0)$ is

$$\hat{y}_j(\boldsymbol{x}_0) = \boldsymbol{f}_j(\boldsymbol{x}_0)^\top \boldsymbol{\beta} + (\boldsymbol{T}_{\cdot j} \otimes \boldsymbol{\phi}(\boldsymbol{x}_0))^\top (\boldsymbol{T} \otimes \boldsymbol{\Phi})^{-1}(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}),$$

where $\boldsymbol{\phi}(\boldsymbol{x}_0) = (\phi(\boldsymbol{x}_0, \boldsymbol{x}_1), \ldots, \phi(\boldsymbol{x}_0, \boldsymbol{x}_n))^\top$, and $\boldsymbol{T}_{\cdot j}$ is the j-th column of $\boldsymbol{T}$. According to the property of the Kronecker product, $(\boldsymbol{T}_{\cdot j} \otimes \boldsymbol{\phi}(\boldsymbol{x}_0))^\top (\boldsymbol{T} \otimes \boldsymbol{\Phi})^{-1}(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}) = \text{vec}\left\{\boldsymbol{\phi}(\boldsymbol{x}_0)^\top \boldsymbol{\Phi}^{-1}\boldsymbol{U}\boldsymbol{T}^{-1}\boldsymbol{T}_{\cdot j}\right\}$, where $\boldsymbol{U}$ is the same as defined in (13), and vec$(\cdot)$ denotes the vectorization of a matrix. Notice that $\boldsymbol{T}^{-1}\boldsymbol{T}_{\cdot j}$ is a vector with jth entry of 1 and other

entries of 0. Then $\boldsymbol{U}\boldsymbol{T}^{-1}\boldsymbol{T}_{\cdot j}$ becomes the j-th column of $\boldsymbol{U}$, which is

$$(y_j(\boldsymbol{x}_1) - \boldsymbol{f}_j(\boldsymbol{x}_1)^\top\boldsymbol{\beta}, \ldots, y_j(\boldsymbol{x}_n) - \boldsymbol{f}_j(\boldsymbol{x}_n)^\top\boldsymbol{\beta})^\top \triangleq \boldsymbol{y}_j - \boldsymbol{F}_j\boldsymbol{\beta}.$$

Therefore, $\hat{y}_j(\boldsymbol{x}_0)$ can be alternatively expressed by

$$\hat{y}_j(\boldsymbol{x}_0) = \boldsymbol{f}_j(\boldsymbol{x}_0)^\top\boldsymbol{\beta} + \boldsymbol{\phi}(\boldsymbol{x}_0)^\top\boldsymbol{\Phi}^{-1}(\boldsymbol{y}_j - \boldsymbol{F}_j\boldsymbol{\beta}),$$

which is the same as the predictor built based on the data from the jth categorical level. Therefore, if we have the same design for the quantitative inputs over all the qualitative levels, the data points from the different levels are not contributing to the prediction. This is the reason why designs for quantitative inputs are often different across the different qualitative levels. As such, the entire dataset contributes to the estimation of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. Hence, the difference between the different approaches are very small if the quantitative input design is the same for all the qualitative levels. We use Case 1 in Table 1 to demonstrate this phenomenon in Table 6. In this experiment, the design of the quantitative inputs is the same for all qualitative levels with size n of 4 or 8. The results indicate that the performance of all the approaches is essentially the same up to five digits.

Table 6: Additional results of case 1 in Section 4.1. mean: average MSE over all k categorical levels. sd: standard deviation of the MSEs over all k categorical levels. time: computational time in minutes. The computing time of RC includes a cross-validation procedure for choosing the penalty parameter λ.

		RC	IC	UC	EC	ECsep
$n = 4$	mean	0.10987	0.10987	0.10987	0.10987	0.10987
	sd	0.00007	0.00007	0.00008	0.00007	0.00007
$n = 8$	mean	0.02148	0.02148	0.02148	0.02148	0.02148
	sd	0.00001	0.00001	0.00001	0.00001	0.00001