Pairwise Meta-Modeling of Multivariate Output Computer Models Using Nonseparable Covariance Function

Yongxiang Li and Qiang Zhou
Department of Systems Engineering and Engineering Management
City University of Hong Kong
Hong Kong
(yongxili-c@my.cityu.edu.hk; q.zhou@cityu.edu.hk)

Gaussian process (GP) is a popular method for emulating deterministic computer simulation models. Its natural extension to computer models with multivariate outputs employs a multivariate Gaussian process (MGP) framework. Nevertheless, with significant increase in the number of design points and the number of model parameters, building an MGP model is a very challenging task. Under a general MGP model framework with nonseparable covariance functions, we propose an efficient meta-modeling approach featuring a pairwise model building scheme. The proposed method has excellent scalability even for a large number of output levels. Some properties of the proposed method have been investigated and its performance has been demonstrated through several numerical examples. Supplementary materials for this article are available online.

KEY WORDS: Computer experiment; Meta-models; Multivariate Gaussian process; Pairwise modeling; Pseudolikelihood.

1. INTRODUCTION

Many areas in science and engineering are confronted with the need for computer simulations to study complex real world systems or solve challenging design problems. Due to their increasing complexity, computer simulations for many real world problems may take exceedingly long time. Gaussian process-based emulators (Santner, Williams, and Notz 2003; Fang, Li, and Sudjianto 2005) have become a popular remedy to this issue. The standard statistical framework in emulating (deterministic) computer simulations deals only with univariate output models (Sacks, Schiller, and Welch 1989; Sacks et al. 1989; Currin et al. 1991). Much less effort has been spent on computer models with multivariate outputs, despite of their prevalence in many applications. In addition, computer models with both qualitative and quantitative factors, and ensemble analysis of multiple computer models (e.g., climate models based on various physical models, Tebaldi and Knutti 2007), can also be equivalently dealt with using the framework of multivariate output models. A simple approach to model the multivariate output is by separately constructing an individual GP (kriging) for each output. This method is sufficient when there is no correlation across different output levels. However, different output levels from computer simulation models in many real world applications can be correlated. Ignoring the cross-correlations among different output levels may cause substantial loss of efficiency, leading to poor parameter estimation and prediction accuracy.

Some attempts have been made on building meta-models with multiple outputs. Kennedy and O’Hagan (2000) used an autoregressive structure to jointly model several computer codes with different levels of accuracy, in the hope of improving prediction accuracy of the high-fidelity code with information from lower fidelity codes. Han et al. (2009) proposed a hierarchical Bayesian model to address the computer experiments that have both quantitative and qualitative input variables, called HQQV. In their model, different output levels are modeled by independent univariate GPs with similar parameters governed by a hierarchical Bayesian structure. By combining data from all levels, better parameter estimates can be obtained. Another work was by Qian, Wu, and Wu (2008) and further refined in Zhou, Qian, and Zhou (2011), where a general framework for building multivariate Gaussian process (MGP) models was proposed. The covariance functions in their framework are separable ones that share a common marginal covariance function at all output levels. This assumption simplifies the covariance structure and significantly reduces the number of model parameters. However, the separability assumption is not appropriate in some applications. Li et al. (2008) developed a test for separability of covariance functions in multivariate random fields, which sheds some light on what type of covariance functions may be chosen.

In this work, we deal with the situation where a nonseparable covariance structure is necessary to adequately model the outputs. Nonseparable covariance functions allow different covariance parameters for different output levels, which is much more flexible than separable ones. Some related works on nonseparable covariance functions have recently been done by MelKumyan and Ramos (2011) and Fricker, Oakley, and...
for modeling multivariate outputs. In practice, if every single output level of a computer code can be suitably modeled by a standard GP, then the collection of all its output levels may be handled by Model (1).

Nonseparable covariance functions are much more flexible than separable ones as they do not assume the same marginal covariance function for all outputs. There are mainly two ways to build valid nonseparable cross-covariance functions: coregionalization and kernel convolution.

The linear model of coregionalization (LMC) was proposed by Goulard and Voltz (1992). The key idea is to model the output processes \( \epsilon(\cdot) \) as linear combinations of a set of independent GPs \( \delta(\cdot) \):

\[
\epsilon(\cdot) = A\delta(\cdot),
\]

where commonly \( \delta(\cdot) \) is a vector of \( k \) independent GPs with zero mean and unit variance, and \( A \) is a \( k \times k \) full rank matrix, though theoretically \( \delta(\cdot) \) may consist of any number of independent GPs. The between-outputs covariance of \( \epsilon(\cdot) \) is hence \( \Sigma = AA^T \). There are several decomposition methods for \( A \). Gelfand et al. (2004) employed the Cholesky decomposition for computational convenience. This method assumes a hierarchical dependence of the outputs. Higdon et al. (2008) used spectral decomposition to solve the computer experiments with high-dimensional output where the matrix \( A \) was obtained via singular value decomposition (SVD) of the standardized simulation output matrix.

The nonseparable covariance functions by kernel convolution have been discussed in Ver Hoef et al. (1998), Higdon (2002), Majumdar and Gelfand (2007), and MelKumyan and Ramos (2011). In the similar vein, a more flexible cross-correlation structure was proposed by Fricker, Oakley, and Urban (2013).

In this article, we adopt their proposed correlation function for more flexibility, which is given by

\[
R_{ij}(x, x') = \frac{\text{exp}\left\{-\frac{(x - x')^T}{2} (\Phi^{-1}/2 + \Phi^{-1}/2)^{-1} (x - x')\right\}}{|\Phi^{-1}/2 + \Phi^{-1}/2|^{1/4}},
\]

(2)

where \( \Phi = \text{diag}(\phi_{i1}, \phi_{i2}, \ldots, \phi_{ip}), \) and \( T_{i,j} \in [-1, 1] \) is the \( i \text{th}, j \text{th} \) element of a \( k \times k \) matrix \( T \) that is positive definite with unit diagonal elements (PDUDE). \( T_{i,j} \) denotes the cross-correlation factor between the \( i \text{th} \) and \( j \text{th} \) levels, with \( T_{i,i} = 1 \). A nice feature of (2) is that the \( i \text{th} \) marginal process \( \epsilon_i(x) \) has the most popular Gaussian correlation function \( \text{exp}\{-\frac{(x - x')^T}{2} \phi(x - x')\} \). Hence (2) can be viewed as a generalization of the Gaussian correlation function to an MGP framework. Parameters in the MGP model can be estimated by maximum likelihood method in a similar fashion as that in the standard univariate GP model, with the additional constraint that the \( T \) must be a PDUDE. In the rest of this article, we focus our discussion on the MGP model with the correlation function defined in (2).

3. THE PAIRWISE MODELING APPROACH

While the aforementioned MGP model is flexible and theoretically appealing, model building can be painfully prohibitive in practice. In modern scientific and engineering applications,
it is not rare for computer models to have a large number of output levels (e.g., Higdon et al. 2008; McFarland et al. 2008). For computer models with qualitative factors using the MGP framework, the number of output levels skyrockets easily since it is determined by the total level combinations of all the qualitative factors. For a moderately large problem, the number of parameters in the covariance function can easily reach hundreds or even thousands. In addition, more design points are usually required for multivariate outputs, resulting in a large-scale covariance matrix \( \mathbf{C} \). In building the meta-model, parameter estimation is conducted in a high-dimensional space, with inversion of a large-scale covariance matrix at each step. Hence, standard modeling approaches are plagued by exceedingly high computational load, numerical issues, and getting trapped into local maxima in the high-dimensional parameter space, rendering the modeling approach inapplicable.

To solve this issue, we propose an alternative modeling approach with excellent scalability, especially for high-computational load, numerical issues, and getting trapped into local maxima or even thousands. In addition, more design points are usually required for computer models with qualitative factors using the MGP output levels (e.g., Higdon et al. 2008; McFarland et al. 2008). It is not rare for computer models to have a large number of parameters contained in \( \mathbf{C} \) required for elements within each column of \( \mathbf{C} \). To achieve this, a special sequence must be followed, that is, constructing \( \mathbf{T} \) by column from the 1st column to the \((k-1)\)th column. There is no special order required for elements within each column of \( \mathbf{T} \), but we specify the order to be from top to bottom, without loss of generality. Taking the case \( k = 4 \), for example, \( \mathbf{T} \) can be expressed by \( \omega_{i,j} \)’s as

\[
\begin{aligned}
T_{1,1} &= f_{i,1}(u_{i,1}) = \cos \omega_{i,1}, \\
T_{1,2} &= f_{i,2}(u_{i,2}) = \cos \omega_{i,2} \cos \omega_{i,1} \\
&\quad + \sin \omega_{i,1} \cos \omega_{i,2} \sin \omega_{i,3}, \\
T_{1,3} &= f_{i,3}(u_{i,3}) = \cos \omega_{i,3} \cos \omega_{i,1} \\
&\quad + \sin \omega_{i,1} \cos \omega_{i,3} \cos \omega_{i,4}, \\
T_{1,4} &= f_{i,4}(u_{i,4}) = \cos \omega_{i,4} \cos \omega_{i,1} \\
&\quad + \sin \omega_{i,1} \cos \omega_{i,4} \cos \omega_{i,3} \sin \omega_{i,4} \sin \omega_{i,1} \sin \omega_{i,4} \sin \omega_{i,2} \sin \omega_{i,1} \sin \omega_{i,4}.
\end{aligned}
\]

It is easy to see that the matrix \( \mathbf{T} \) can be constructed following the sequence \( \{T_{2,1}, T_{3,1}, T_{4,1}, T_{2,2}, T_{3,2}, T_{4,2}, T_{2,3}, T_{3,3}, T_{4,3}\} \), each time by introducing only one more \( \omega_{i,j} \): (Step 1) The elements in the 1st column, \( T_{i,1} \)’s, can each be obtained based on the corresponding \( \omega_{i,1} \). In this step, \( \{T_{2,1}, T_{3,1}, T_{4,1}\} \) are calculated based on \( \{\omega_{2,1}, \omega_{3,1}, \omega_{4,1}\} \); (Step 2) \( T_{2,2} \) is calculated based on \( \{\omega_{2,2}, \omega_{3,2}, \omega_{4,2}\} \); (Step 3) \( T_{2,3} \) is calculated based on \( \{\omega_{2,3}, \omega_{3,3}\} \); (Step 4) \( T_{4,3} \) is calculated based on \( \{\omega_{4,3}, \omega_{4,1}\} \).

### 3.2 The Proposed Method

#### 3.2.1 The Pairwise Modeling Approach. Assume the design \( \mathbf{X} \) for the MGP model has \( n \) training points with \( n_i \) points at the \( i \)th level denoted by \( \mathbf{X}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}]^T \), \( i = 1, \ldots, k \), where \( \sum_{i=1}^k n_i = n \). Denote the corresponding simulation outputs by \( \mathbf{y} = [y_1^T, y_2^T, \ldots, y_k^T]^T \), where \( y_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,n_i}]^T \). Note we do not impose the constraint that \( \mathbf{X}_i = \mathbf{X}_j \) for \( i \neq j \), though it is possible in practice. The full likelihood function of the MGP model is then given by

\[
L(\sigma, \mathbf{F}, \Phi; \mathbf{y}) = (2\pi)^{-n/2} |\mathbf{C}|^{-1/2} \exp \left[ -\frac{(\mathbf{y} - \mathbf{F}\mathbf{F}^T)}{2} \right] \times \mathbf{C}^{-1},
\]

where \( \mathbf{F} = \text{diag}(\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_k) \) is an \( n \times q \) block-diagonal matrix with the \( i \)th block given by \( n_i \times q_i \) matrix \( \mathbf{F}_i = [f_i(x_{i,1}), \ldots, f_i(x_{i,n_i})]^T \), \( \mathbf{C} \) is the covariance matrix containing blocks \( \mathbf{C}_{i,j} \), the covariance matrix between \( i \)th and \( j \)th output levels. Denote the correlation matrix by \( \mathbf{R} \), whose \((i\text{th},j\text{th})\) block is \( \mathbf{R}_{i,j} = \mathbf{C}_{i,j} / \sigma_i \sigma_j, i, j = 1, \ldots, k \).

The idea of composite likelihood, or more generally pseudolikelihood (Besag 1975; Lindsay 1988; Lindsay et al. 2011; Varin et al. 2011) is to simplify the full likelihood function.
by multiplying the marginal or conditional likelihoods of some subsets. Based on the idea, the composite likelihood version of (5) with marginal bivariate GP models is

\[
L_{pw}(\sigma, \beta, \Phi, T; y) = \prod_{1 \leq i < j \leq k} \exp \left[ -\frac{1}{2} \frac{(y^{(i,j)} - F^{(i,j)}(\theta^{(i,j)}))}{(C^{(i,j)})^{-1}(y^{(i,j)} - F^{(i,j)}(\beta^{(i,j)}))} \right],
\]

\[
(2\pi)^{n_{i,j}/2} |C^{(i,j)}|^{1/2},
\]

where \(y^{(i,j)} = [y_i^T, y_j^T]^T\), \(\beta^{(i,j)} = [\beta_i, \beta_j]^T\), \(F^{(i,j)} = diag(F_i, F_j)\) and

\[
C^{(i,j)} = \left[ \begin{array}{cc} C_{i,i} & C_{i,j} \\ C_{j,i} & C_{j,j} \end{array} \right] = \left[ \begin{array}{cc} \sigma_i \sigma_i R_{i,i} & \sigma_i \sigma_j R_{i,j} \\ \sigma_j \sigma_j R_{j,i} & \sigma_j \sigma_j R_{j,j} \end{array} \right]
\]

\[
= \left[ \begin{array}{cc} \sigma_i \sigma_i T_{i,i} K_{i,i} & \sigma_i \sigma_j T_{i,j} K_{i,j} \\ \sigma_j \sigma_i T_{j,i} K_{j,i} & \sigma_j \sigma_j T_{j,j} K_{j,j} \end{array} \right],
\]

where \(K_{i,j} = R_{i,j}/T_{i,j}\) according to (2).

It is clear that both (5) and (6) contain the same set of model parameters. To reduce the dimensionality, which is the major cause of difficulties in estimating the MGP meta-model, \((k - 1)/2\) pairwise submodels based on marginal bivariate GPs are individually built and estimated. The HD is introduced to reparameterize the \(T\) matrix to guarantee the validity of the overall covariance structure. Based on the sequential construction procedure of HD described in Section 3.1, the pairwise submodels need to be built according to the specified sequence.

The building of pairwise submodels is carried out in two steps. First, the \((k - 1)\) pairwise submodels based on the 1st level and the \(i\)th levels are built separately and their parameters are estimated by maximizing the likelihood

\[
L_{i,1}(\theta_i, \phi_i, \omega_{i,1}; y_i, y_j) = \prod_{1 \leq i < j \leq k} \exp \left[ -\frac{1}{2} \frac{(y^{(i,j)} - F^{(i,j)}(\theta^{(i,j)}))}{(C^{(i,j)})^{-1}(y^{(i,j)} - F^{(i,j)}(\beta^{(i,j)}))} \right],
\]

\[
(2\pi)^{n_{i,j}/2} |C^{(i,j)}|^{1/2},
\]

where \(\theta_i = \{\sigma_i, \beta_i, \Phi_i\}, i = 2, \ldots, k\). Denote the estimated parameters by \(\hat{\theta}_i, \hat{\beta}_i, \hat{\Phi}_i, \hat{\omega}_{i,1}\).

In the second step, subsequent submodels in the specified sequence can be built one by one conditional on those already obtained. Assume now we are at the step of building the pairwise submodel for the \(i\)th and \(j\)th output levels, the cross-correlation parameter is \(T_{i,j} = f_{i,j}(u_{i,j}) = f_{i,j}(u_{i,j}, \omega_{i,j}), 1 < j < i \leq k\). At this step, all the \(\omega_{i,j}\)’s contained in \(u_{i,j}\) should have already been obtained in previous submodels, denote them by \(\hat{\omega}_{i,j}\). Therefore, the only free parameter in determining \(T_{i,j}\) is \(\omega_{i,j}\), which can be estimated by maximizing the pairwise likelihood \(L_{i,j}\) conditional on \(\hat{\omega}_{i,j}\):

\[
L_{i,j}(\theta_i, \phi_i, \omega_{i,j}; y_i, y_j) = \prod_{1 \leq i < j \leq k} \exp \left[ -\frac{1}{2} \frac{(y^{(i,j)} - F^{(i,j)}(\theta^{(i,j)}))}{(C^{(i,j)})^{-1}(y^{(i,j)} - F^{(i,j)}(\beta^{(i,j)}))} \right],
\]

\[
(2\pi)^{n_{i,j}/2} |C^{(i,j)}|^{1/2},
\]

For any pairwise marginal likelihood defined in (7) or (8), the maximum likelihood estimation can be carried out following the procedure given in Appendix A of the online supplementary material. The above procedure is repeated until all pairwise submodels are built. Denote the model parameters estimated from the \((i, j)\th pair by \(\hat{\theta}_i, \hat{\beta}_j, \hat{\phi}_i, \hat{\Phi}_i, \hat{\omega}_{i,j}\). Parameters with multiple estimates are averaged at the end to produce a single estimate. The procedure of the proposed approach is summarized as below.

1. Build the \((k - 1)\) pairwise submodels between the 1st level and the \(i\)th level, \(i = 2, 3, \ldots, k\). Estimated model parameters \(\{\hat{\theta}_i, \hat{\beta}_i, \hat{\phi}_i, \hat{\omega}_i\}\) are obtained by maximizing (7) for each submodel.

2. Assume all submodels involving the first \(j - 1\) (\(2 \leq j \leq k - 1\)) output levels have been built. Then, submodels between the \(i\)th level and the \(j\)th level can be built one by one, for \(i = j + 1, j + 2, \ldots, k\). Estimated model parameters \(\{\hat{\theta}_i, \hat{\beta}_i, \hat{\phi}_i, \hat{\omega}_i\}\) are obtained by maximizing (8) for each submodel.

3. Repeat the preceding step, increasing \(j\) by 1 each time until \(j = k - 1\).

Take the following averages for multiple parameter estimates for \(i = 1, 2, \ldots, k\):

\[
\hat{\theta}_i = \frac{1}{k - 1} \sum_{t \neq i} \hat{\theta}_i, \quad \hat{\beta}_i = \frac{1}{k - 1} \sum_{t \neq i} \hat{\beta}_i, \quad \hat{\phi}_i = \frac{1}{k - 1} \sum_{t \neq i} \hat{\phi}_i.
\]

Finally, calculate \(\hat{T}_i, j = f_{i,j}(\hat{u}_{i,j})\). Note that \(\hat{T}_i, j\) and the averaged estimates, \(\hat{\theta}_i, \hat{\beta}_i, \hat{\phi}_i, \hat{\omega}_i\), do not form an optimal parameter set for the likelihood function \(L_{i,j}\) anymore. To alleviate this, we can reestimate each \(T_{i,j}\) conditional on the values of \(\hat{\theta}_i, \hat{\beta}_i, \hat{\phi}_i, \hat{\omega}_i\), following the same sequential pairwise estimation procedure specified above. For example, we reestimate \(\omega_{i,1}\) (hence \(T_{i,1}\)) by maximizing the conditional likelihood \(L_{i,1}(\omega_{i,1} | \hat{\theta}_i, \hat{\beta}_i, \hat{\phi}_i, \hat{\omega}_i; y_i, y_1)\) as in (7). Our experience with extensive numerical simulations has indicated that adding the reestimation step for \(\hat{T}_i, j\) can produce consistently better results than that without this step.

In Fieuws and Verbeke (2006), a similar pairwise modeling approach in a different context is implemented without the HD reparameterization and the reestimation step. We shall refer to their method as the FV method in the remaining part of this article. Since all the pairwise submodels are built independently in FV, the estimated matrix \(\hat{T}\), denoted by \(\hat{T}^{FV} = \{\hat{T}_{i,j}^{FV}\}\), may not be positive definite, leading to an invalid covariance function. As can be seen later in the numerical examples, the probability of producing invalid covariance function can be quite high in some cases.

In our proposed approach, some submodels are built conditionally on some upstream submodels in the sequence specified. This leads to the concern of potential error propagation, that is, inaccurately estimated upstream parameters may adversely affect the estimate of some downstream parameters. This mostly affects the parameters in the cross-correlation matrix \(T\). We provide some insights into this issue by introducing two propositions in Appendix B of the online supplementary material, each followed by a short discussion. This issue will also be examined in the numerical examples.

### 3.2.2 Model Prediction

With the parameters obtained from the proposed approach, the response at any untried location
can be predicted. The empirical best linear unbiased predictor (EBLUP) of $y$ at the new input point $x_0$ is

$$y(x_0) = f(x_0)^T \hat{\beta} + \hat{G}(x_0)^T \hat{C}^{-1} (y - \hat{F} \hat{\beta}),$$

where $\hat{C}$ is the covariance matrix of $y$ with estimated parameters and

$$\hat{G}(x_0) = \begin{bmatrix} \hat{C}_{1,1}(x_0, X_1) & \hat{C}_{1,2}(x_0, X_2) & \cdots & \hat{C}_{1, n}(x_0, X_n) \\ \hat{C}_{2,1}(x_0, X_1) & \hat{C}_{2,2}(x_0, X_2) & \cdots & \hat{C}_{2, n}(x_0, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C}_{n,1}(x_0, X_1) & \hat{C}_{n,2}(x_0, X_2) & \cdots & \hat{C}_{n, n}(x_0, X_n) \end{bmatrix},$$

where $\hat{C}_{i, j}(x_0, X_j) = [\hat{C}_{i, j}(x_0, x_{j,1}), \hat{C}_{i, j}(x_0, x_{j,2}), \ldots, \hat{C}_{i, j}(x_0, x_{j,n})]^T$.

In most GP-based models, a variance estimator of the model prediction at $x_0$ can often be provided with plug-in parameter estimates. In our case, the formula has been complicated by the pairwise estimation approach, particularly the estimation of $\hat{\beta}$. Hence, we suggest using the bootstrap method for estimating the variance following the method proposed by Den Hertog et al. (2006). Though originally proposed for univariate GP model, the algorithm can be easily extended to our case.

4. NUMERICAL EXAMPLES

This section gives some numerical examples to demonstrate the performance of the proposed method. The proposed method in Section 3.2.1 is denoted as PW. Four other reference methods are introduced for comparison: (1) the individual Kriging method (IK), where a standard univariate GP model is applied to each output; (2) the full multivariate approach (MV) based on maximizing the full likelihood in (5), with $T$ being parameterized by the HD to ensure its validity; (3) The HQQV method by Han et al. (2009) introduced in Section 1; (4) The LMC method with spectral decomposition for the between-outputs covariance in Higdon et al. (2008) (with maximum likelihood estimates). Note the autoregressive model in Kennedy and O’Hagan (2000) is a special case of the LMC model.

In all models, a constant term for the regression function is employed, which is the standard practice in GP models (Kleijnen 2009). All computations are done with MATLAB 2013 in a 64-bit Windows Server 2012 environment. The model building time reported is the time cost based on a single core CPU at 2.6 GHz. Since HQQV uses Markov chain Monte Carlo (MCMC) to generate posterior sample for prediction,
its running time heavily depends on the chain size. In general, there is no standard rule for choosing the chain length and it varies with applications. To achieve a fair comparison on the running time of HQQV, a chain length of 20,000 is used. The reason is that, in the examples, a chain length of 10,000 yields discernible worse prediction results, while that of 30,000 produces no substantial improvements but with a much longer running time.

4.1 Example 1: A Simulated MGP

In this example, data are obtained from repeatedly simulating an MGP with specified parameters. The advantage of simulating from an MGP is that the underlying true parameters are known, allowing us to assess not only the accuracy of the model prediction but also the accuracy of parameter estimation.

**Example 1.1.** The MGP is defined in one-dimensional space [0, 8] and has four levels of outputs. The true values of $\sigma$, $\beta$, $\Phi$, and $T$ are listed in second column of Table 1. A typical random realization from this MGP is illustrated in Figure 1.

The training points are generated by the sliced Latin hypercube design (SLHD) in Qian (2012), with 20 points at each level. The testing points are 50 evenly spaced points at each level: [0, 8/49, 16/49, …, 8]. The simulation is repeated for 5000 times, each time the SLHD and response values are regenerated. The accuracy of parameter estimation is presented in Table 1, and model prediction accuracy is given in Figure 2(a). It can be seen that parameters can be properly estimated by our method, and its prediction performance is close to that of the full multivariate approach.

**Example 1.2.** Here, we rerun Example 1.1 using smaller cross-correlation parameters to investigate the method’s performance when outputs are less correlated. All other settings remain the same as in the Example 1.1. The true values of parameters and simulation results are also given in Table 1, and the predictive accuracy is shown in Figure 2(b). Similar conclusions as those in Example 1.1 can be made, except that the prediction accuracy of MV and PW is closer to IK in this case, which is expected as an integrated modeling approach will not gain much advantage if cross-correlations are not strong enough.

It is also noted that the proposed method depends on the ordering of output levels, which is often arbitrarily set. To give some numerical insight for the sensitivity to the ordering, the above example has been repeated for 24 times, each with a distinct ordering. The prediction results based on all orderings are given in Figure 3. Clearly, the ordering of output levels does not play an important role in this example.
Table 2. Accuracy of cross-correlation parameter estimation in Example 1.3

| Parameter | True value | RMSE<sub>AV</sub> | RMSE<sub>PW</sub> | RMSE<sub>FV</sub> |
|-----------|------------|-------------------|-------------------|-------------------|
| $T_{2,1}$ | -0.9422    | 0.0294            | 0.0319            | 0.0323            |
| $T_{3,1}$ | 0.9394     | 0.0308            | 0.0330            | 0.0331            |
| $T_{4,1}$ | 0.9394     | 0.0303            | 0.0325            | 0.0328            |
| $T_{3,2}$ | -0.9771    | 0.0118            | 0.0130            | 0.0124            |
| $T_{4,2}$ | -0.9771    | 0.0120            | 0.0133            | 0.0126            |
| $T_{4,3}$ | 0.9218     | 0.0379            | 0.0394            | 0.0412            |

Figure 4. (a) Boxplots of predictive RMSEs; (b) boxplots of model building time.

Figure 5. Boxplots of predictive accuracy and model building time in Example 2.
Table 3. Three qualitative factors and their levels in the borehole example

| Qualitative factor | Level 1 | Level 2 | Level 3 |
|--------------------|---------|---------|---------|
| $H_l$              | 700     | 760     | 820     |
| $r$                | 10,000  | 20,000  | 30,000  |
| $K_w$              | 10,000  | 11,000  | 12,000  |

We have obtained similar conclusions from other examples as well. This suggests that, although ordering does have some impact on the results, in general, no special care seems to be needed.

Example 1.3. Now we further reevaluate Example 1.1 by increasing the magnitude of each $T_{r,s}$ with all other parameters unchanged. The new parameters and results are listed in Table 2. Comparing to that in the Example 1.1, the modified $T$ matrix is closer to singularity, which leads to 34.33% of $\hat{T}^{FV}$ obtained from the FV method being nonpositive definite. Due to the Proposition 1 in the supplementary material, potential error propagation may exist in the PW method when $\hat{T}^{FV}$ is nonpositive definite. The results show that this phenomenon has negligible impact in this example. Comparing Table 2 with Table 1, it is also interesting to see that the RMSEs of $T_{i,j}$'s are significantly smaller for the modified $T$ matrix. This agrees with the results from Proposition 2 in the supplementary material.

Example 1.4. Now we compare the methods using a further modified MGP defined in the two-dimensional input space $X \in [0, 3]^2$ with $k = 15$ levels of outputs. This MGP contains 135 parameters in its correlation function, in addition to process variances and regression parameters. We set $\beta_i = 0$, $\sigma_i = 2 + U_i, \Phi_i = \text{diag}(2 + U_i, 2 + U_i)$, for $i = 1, 2, \ldots, k$, and $T_{i,j} = 0.85 + 0.05U_{T_{i,j}}$ for $1 \leq j < i \leq k$, where $U_i, U_{i,1}, U_{i,2}, U_{i,3}$, and $U_{T_{i,j}}$'s are obtained by independently sampling from the uniform distribution on $[0, 1]$ in each iteration. It is possible that $T$ is negative definite due to the random generation mechanism in some iterations; it is simply regenerated until a valid one is found. The training data points are generated by SLHD with 100 points at each level, and the testing data contain 400 grid points as $[0, 3/19, 6/19, \ldots, 3]^2$. Due to the long model building time of the MV method, only 100 iterations have been conducted.

The boxplots of predictive RMSEs as well as model building time are shown in Figure 4. It is clear that the PW method reaches a similar level of prediction accuracy as the MV method, while significantly reducing the model building time. Note that it takes about 10 hours to build each MV model for this example, which has only two-dimensional input and a moderate number of output levels. This shows the excellent scalability of the PW method, as well as its capability in delivering similar prediction performance as the full multivariate approach.
4.2 Example 2: Deterministic Quadratic Curves

In this section, we adopt the quadratic function example from Han et al. (2009) with some modifications. The multivariate output model is simulated by a set of similar quadratic functions. Following Han et al. (2009), denote a set of $k$ quadratic curves by $y_j(x) = H_{1,j}x^2 + H_{2,j}x + H_{3,j}$, for $j = 1, 2, \ldots, k$, where $x \in [0, 1]$. The coefficient $H_{i,j}$ is the $(i, j)$ element of a $3 \times k$ matrix $H_k$. Three levels of output are used in the original example, but here we extend them to 20 and 50 levels as a high-dimensional example. Corresponding to the $j$th output level, the $j$th column of $H_k$ ($k = 20$ or 50) is defined with $H_{1,j} = a$, $H_{2,j} = -2ab$, and $H_{3,j} = c$, where $a = -5 - 3U_{j,1}$, $b = (2 + U_{j,2})/5$, $c = -1 + 2U_{j,3}$, and $U_{j,1}$, $U_{j,2}$, $U_{j,3}$ are independently generated from the uniform random distribution on $[0, 1]$. The training points are from an SLHD with five points at each level, and the testing points are 15 evenly spaced points at each level defined as $[0, 1/14, 2/14, \ldots, 1]$. The two experiments with respectively 20 and 50 output levels are both repeated for 100 iterations, each time a new SLHD and a new $H_k$ matrix are randomly generated.

The boxplots of predictive RMSE and model fitting time for both cases are shown in Figure 5. The example with 50 output levels is very computationally demanding. A single model building process for the LMC method costs more than 1 day on average, followed by the MV and HQQV methods using about 10 hr. But only about half an hour is required by the PW method. The time saving is not so dramatic in the 20-level example, but the PW method still uses the least amount of modeling time.

In terms of prediction accuracy, the PW and MV methods are the best, while HQQV and LMC are similar and relatively inferior. But all four integrated modeling methods outperform IK significantly, once again demonstrating the benefit of jointly modeling correlated outputs. It is also observed that the appealing features of the PW method may gradually disappear as the number of output levels decreases. The MV method may be well-qualified to handle problems at smaller scales. Therefore, we only recommend the PW method for moderate to large-scale problems which cannot be handled well by other methods.

4.3 Example 3: The Borehole Example

The borehole function is a popular testing example in the computer experiments literature. It models the flow rate of water through a borehole. Here, we use the following slightly modified version of the borehole function:

$$y = 2\pi (H_u - H_l) \left\{ \frac{\log{(r/r_w)}}{10,000 T_u} + \frac{2L}{\log{(r/r_w)} r_w^2 K_w} + \frac{100}{T_l} \right\}.$$

The detailed descriptions and ranges for each variable can be found in Morris, Mitchell, and Ylvisaker (1993). To produce multiple output levels, three variables are chosen as qualitative

Maximizing the complex full likelihood function in a search space of 1275 dimensions is certainly a daunting task for any search algorithm. The process is also plagued by numerical issues in inverting the $250 \times 250$ covariance matrix at each step of searching.

It is also observed that the appealing features of the PW method may gradually disappear as the number of output levels decreases. The MV method may be well-qualified to handle problems at smaller scales. Therefore, we only recommend the PW method for moderate to large-scale problems which cannot be handled well by other methods.
Thermal management of the IC chips is essential to ensuring computers, cell phones, and other modern electronic products. The goal of this experiment is to investigate the cooling effects of various designs of the heat sink. Figure 7 shows the schematic layout of the circuit board. There are several peripheral chips surrounding a main chip on the board. A disk-stack heat sink is mounted to the main chip.

The results are given in Figure 6. It can be seen that the proposed pairwise method gives consistently better and more stable prediction results than other methods, but using the least amount of time. The reason is that the full MGP models fail to explore the likelihood properly in high-dimensional parameter space and hence give unsatisfactory parameter estimation. It is also seen that as the number of training points reduces to eight, the PW method does not gain much advantage over the MV in terms of both prediction accuracy and running time. This suggests that the PW may not be preferable if the sample size is small, when very limited data points are available in building each bivariate model.

The HQQV method is much more time-consuming than other methods as it uses MCMC to draw the posterior samples. Even so, it yields unsatisfactory results in this example. In fact, the underlying mechanism of borrowing information across levels in HQQV is different from that of others, which are through the cross-covariance of an MGP. It assumes that the univariate GPs are independent of each other, but with similar parameters. By capturing the output similarity at the parameter level of GPs, the HQQV is able to handle data that look seemingly different at all levels but with similar smoothness and dynamic behaviors. However, methods based on an MGP framework such as ours, are less likely to provide substantial improvements over IK if the univariate GPs are independent of each other. It is hard to conclude which mechanism is better in general as it will depend on the actual application, but the HQQV framework apparently is not suitable for this example.

### 4.4 A Heat Sink Computer Experiment

In this section, the proposed method is applied to a computer experiment for a heat sink design. IC chips are widely used in computers, cell phones, and other modern electronic products. Thermal management of the IC chips is essential to ensuring their normal functioning without overheating. In some applications, heat sinks are required for cooling down the chips, for example, CPUs and graphical processors in desktop computers. In this example, a simulation model based on finite element analysis is used to demonstrate the design of a disk-stack heat sink. This model is built and simulated in COMSOL Multiphysics. Details of this model can be found in COMSOL (2015). The goal of this experiment is to investigate the cooling effects of various designs of the heat sink.

In this example, a simulation model based on finite element analysis is used to demonstrate the design of a disk-stack heat sink. Details of this model can be found in COMSOL (2015). The goal of this experiment is to investigate the cooling effects of various designs of the heat sink.

| Number of disks | With copper bottom layer | Without copper bottom layer |
|-----------------|---------------------------|-----------------------------|
|                 | 5  | 10  | 15  | 5   | 10  | 15  |
| Aluminum        | 50.56 | 46.27 | 44.80 | 50.21 | 46.46 | 44.85 |
| Copper          | 48.96 | 44.37 | 42.56 | 48.94 | 44.25 | 42.55 |
| Aluminum Alloys | 50.38 | 46.14 | 45.19 | 50.40 | 46.39 | 45.09 |
| 6061            | 49.83 | 45.48 | 44.71 | 49.82 | 45.75 | 44.45 |
| Aluminum Alloys | 6063 | 49.83 | 45.48 | 44.71 | 49.82 | 45.75 | 44.45 |

With copper bottom layer
Without copper bottom layer

#### Table 4. The lowest temperature reached for each combination of qualitative factors
put levels for validation, while the others are used for building the meta-models. The above procedure is repeated for 100 iterations and results are given in Figure 8. It should be noted that, since fitting an MV meta-model is very time-consuming, the model parameters in the covariance function are only estimated once based on the entire data points. Then, these parameters are treated as known in the 100 iterations. It is evident from Figure 8 that the PW method performs similarly as the MV method, with a small fraction of the modeling time. Both multivariate approaches are much better than the IK method.

A leave-one-out cross-validation procedure is also carried out for this example. Since parameter estimates are essentially the same based on either all 720 sample points or the 719 training points in each cross-validation fold, we estimate the model parameters based on all sample points and do not reestimate them in each fold. The RMSE obtained is 0.08, based on deviations between predicted values and corresponding true values at these 720 points. Comparing with the range of the response variable (40–70), the prediction accuracy is very good. To further investigate the uncertainty of the predicted value in each fold, we employ the bootstrap variance estimation method in Den Hertog et al. (2006) as previously discussed. A bootstrap sample size of 6000 is used to estimate the variation and provide a 95% confidence interval for each predicted value in one fold. Of all these 720 intervals centered at the predicted values, 94.2% of them contain the true values. Since it is not possible to show all the 720 intervals here, we randomly select a point at each of the 24 level combinations and provide the prediction intervals against their respective true values in Figure 9.

The meta-model can then be used to find the optimal design of the heat sink. The lowest possible temperatures reached by adding the heat sink are listed in Table 4. It can be seen that the best and worst two designs have a difference of about 8°C, and the impact of adding a copper bottom layer is negligible to the temperature of the main chip. The results suggest that the optimal design of the heat sink should use copper as its material with 15 disks, and the lowest temperature is achieved with \((x_1, x_2, x_3, x_4) = (18.09, 39.58, 1.06, 3.31)\) mm. A verification simulation is conducted with these factor values and the result is 42.66°C, very close to the meta-model predicted value in Table 4.

5. SUMMARY AND DISCUSSION

MGP models with nonseparable covariance functions offer a general solution to building meta-models for computer models with multivariate outputs, computer models with both quantitative and qualitative input factors, and ensemble of computer models. However, conventional modeling approaches face significant challenges in building the model due to the high-dimensional parameter space and large-scale covariance matrix. In this article, we have proposed a pairwise modeling approach to solve this problem. The proposed method takes advantage of the sequential construction property of the HD. The high-dimensional MGP model is broken down into a series of bivariate GP models. Each bivariate GP model is built with a much smaller number of parameters and a small-scale covariance matrix. The method achieves excellent scalability, particularly for high-dimensional output. Several numerical examples have demonstrated that the proposed method can deliver similar prediction performance as the full multivariate approach, with only a small fraction of its computational need. In cases where the output dimension is high, the proposed method may even outperform the full multivariate approach.

A reasonable variation of our method is to use the model parameters estimated from the pairwise approach to be initial values of the full likelihood approach, and further improve them by maximizing the full likelihood function. In our numerical examples, we have tried this technique but found no clear differences, and the modeling time of MV is not substantially reduced. However, improvements may be obtained in some applications by carefully tuning the optimization algorithm in MV. Users can certainly consider this approach if the additional model building time can be tolerated. Another concern is the ordering of output levels, which is often arbitrarily chosen. Although results from the proposed method indeed depend on the ordering, our extensive experience with the numerical examples suggests the model performance in general is not very sensitive to this issue, much like what is shown in Figure 3.

Finally, it should be pointed out that we do not advocate the use of our method in all scenarios, and the users should implement the method at their discretion. In cases when the sample size is small and/or when the number of output levels is small, the PW method can be inferior to the MV method due to loss of efficiency. The good news is that when the sample size and the number of output levels are small, the problem can generally be well handled by the MV already. The PW method is most useful when the scale of the problem is beyond the capability of a full likelihood approach.

SUPPLEMENTARY MATERIALS

Online supplementary materials of this article include appendices, technical proofs, and Matlab code used for the examples.

ACKNOWLEDGMENTS

The authors thank the editor, the associate editor, and two referees for their helpful comments that have led to improvements in the article. This work is supported by the National Natural Science Foundation of China (No. 11301441) and the Hong Kong Research Grants Council (ECS No. 138213).

[Received August 2014. Revised June 2015.]

REFERENCES

Besag, J. (1975), “Statistical Analysis of Non-lattice Data,” The Statistician, 24, 179–195. [485]

COMSOL (2015), “Disk-Stack Heat Sink (Model ID: 3576),” COMSOL Application Gallery. Available at http://www.comsol.com/models. [492]

Currin, 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. [483]

Den Hertog, D., Kleijnen, J. P., and Siem, A. Y. D. (2006), “The Correct Kriging Variance Estimated by Bootstrapping,” Journal of the Operational Research Society, 57, 400–409. [487,493]

Fang, K. T., Li, R., and Sudjianto, A. (2005), Design and Modeling for Computer Experiments, London: Chapman & Hall. [483]

Fieuws, S., and Verbeke, G. (2006), “Pairwise Fitting of Mixed Models for the Joint Modeling of Multivariate Longitudinal Profiles,” Biometrics, 62, 424–431. [484,485,486]

Fricker, T. E., Oakley, J. E., and Urban, N. M. (2013), “Multivariate Gaussian Process Emulators With Nonseparable Covariance Structures,” Technometrics, 55, 47–56. [484]

TECHNOMETRICS, NOVEMBER 2016, VOL. 58, NO. 4
Gelfand, A. E., Schmidt, A. M., Banerjee, S., and Sirmans, C. F. (2004), “Non-stationary Multivariate Process Modelling Through Spatially Varying Coregionalization,” Test, 13, 1–50. [484]
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. [483,487,491]
Higdon, D. (2002), “Space and Space-Time Modeling Using Process Convolutions,” in Quantitative Methods for Current Environmental Issues, eds. C. W. Anderson, V. Barnett, P. C. Chatwin, and A. El-Shaarawi, New York: Springer, pp. 37–54. [484]
Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008), “Computer Model Calibration Using High-Dimensional Output,” Journal of American Statistical Association, 103, 570–583. [484,487]
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. [483,487]
Kleijnen, J. P. (2009), “Kriging Metamodeling in Simulation: A Review,” European Journal of Operational Research, 192, 707–716. [487]
Lindsay, B. G. (1988), “Composite Likelihood Method,” Contemporary Mathematics, 80, 221–239. [485]
Lindsay, B. G., Yi, G. Y., and Sun, J. (2011). “Issues and Strategies in the Selection of Composite Likelihoods,” Statistica Sinica, 21, 71–105. [485]
Majumdar, A., and Gelfand, A. (2007), “Spatial Modeling for Multivariate Environmental Data Using Convolved Covariance Functions,” Mathematical Geology, 39, 225–245. [484]
McFarland, J., Mahadevan, S., Romero, V., and Swiler, L. (2008), “Calibration and Uncertainty Analysis for Computer Simulations With Multivariate Output,” AIAA Journal, 46, 1253–1265. [485]
Melkumyan, A., and Ramos, F. (2011), “Multi-Kernel Gaussian Processes,” in ICAL Proceedings-International Joint Conference on Artificial Intelligence, Vol. 22, pp. 1408–1413. [483,484]
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. [491]
Qian, P. Z. G. (2012), “Sliced Latin Hypercube Designs,” Journal of the American Statistical Association, 107, 393–399. [488]
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, 383–396. [483]
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. [485]
Sacks, J., Schiller, S. B., and Welch, W. J. (1989), “Designs for Computer Experiments,” Technometrics, 31, 41–47. [483]
Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), “Design and Analysis of Computer Experiments,” Statistical Science, 4, 409–423. [483]
Santner, T. J., Williams, B. J., and Notz, W. I. (2003), The Design and Analysis of Computer Experiments, New York: Springer. [483,484]
Tebaldi, C., and Knutti, R. (2007), “The Use of the Multi-Model Ensemble in Probabilistic Climate Projections,” Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 365, 2053–2075. [483]
Varin, C., Reid, N., and Firth, D. (2011), “An Overview of Composite Likelihood Methods,” Statistica Sinica, 21, 5–42. [485]
Ver Hoef, J. M., and Barry, R. P. (1998), “Constructing and Fitting Models for Cokriging and Multivariable Spatial Prediction,” Journal of Statistical Planning and Inference, 69, 275–294. [484]
Zhou, Q., Qian, P. Z. G., and Zhou, S. (2011), “A Simple Approach to Emulation for Computer Models With Qualitative and Quantitative Factors,” Technometrics, 53, 266–273. [483,485,492]