Computer Experiments with Both Quantitative and Qualitative Inputs

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Yulei Zhang, M.S.

Graduate Program in Department of Statistics

The Ohio State University

2014

Dissertation Committee:

Dr. William I. Notz, Advisor
Dr. Peter Craigmile
Dr. Matthew Pratola
Abstract

Physical experiments play an important role in agriculture, industry, and medical research. However, physical experiments can sometimes be difficult or even impossible to run. In these situations, computer experiments are becoming desirable surrogates for physical experiments. This dissertation considers designs and the predictive models for computer experiments with both quantitative and qualitative input variables.

The existing framework for building Gaussian stochastic process (GaSP) models with quantitative and qualitative inputs is to treat a given set of values of the qualitative inputs as determining a response surface in the qualitative inputs. A GaSP model is assumed for each of these response surfaces and the same covariance structure is used for each response surface. A cross-correlation parameter is introduced for each pair of sets of values of the qualitative variables in order to “capture” correlations between response surfaces. To guarantee that one has a legitimate overall covariance structure, certain conditions are imposed on the cross-correlation parameters. In the first part of this dissertation, we introduce two indicator-based GaSP models by transforming the qualitative inputs into quantitative variables and then use traditional correlation functions for quantitative inputs. We also show the equivalence properties between these new models and the existing model.

The second part of this dissertation is about the experimental designs with both quantitative and qualitative inputs. The special data structure requires that a “good”
design not only capture the cross-correlation information but also spread observations out over the entire quantitative inputs space. We propose two types of designs, the partial SLHD and partial CSLHD, which are modifications of existing designs in the literature, and compare their prediction accuracy with all the other existing designs for quantitative and qualitative. By examining several examples, we find that what constitutes a “good” design may vary from case to case. We summarize these findings with a “guideline” for selecting initial designs. Furthermore, when the initial design does not perform well, we also propose a sequential design algorithm to interpolate or extrapolate the target response levels in a GaSP model with mixed inputs.

Inspired by factor analysis, in the last part of this dissertation, we build a more general composite covariance structure by converting the GaSP model with several qualitative levels into a linear combination of independent stochastic processes with fewer constraints on the variance and correlation functions. Furthermore, this composite covariance structure can be extended to the case with multiple qualitative inputs. In these cases, we introduced the Kronecker product form of the composite covariance function, which can not only reduce the number of the parameters, but also capture the similarity between different qualitative inputs with some identical components. In addition, we propose an ANOVA decomposition form of the Gaussian processes, which imposes a factorial structure on the response outputs. Finally, we extend the sequential design algorithm to the composite GaSP model.
To my parents
Acknowledgments

It has been said that the age of 20-30 is the golden age for a person. It’s my pleasure to be admitted as a Ph.D. student of the Ohio State University and spend my golden age here. Throughout 5 years studying and working, I grew up from an aimless student to a person with clear career plan, and solid responsibility of work and life. The completeness of my Ph.D. study is impossible without the support from many faculty members, graduate students and my family.

First of all, I would like to express my sincere gratitude to professor William Notz, my Ph.D. thesis adviser, for his enthusiasm, patience and encouragement throughout my Ph.D. studies. Dr. Notz not only guided my research, but also taught me to think and behave as a professional statistician. I am impressed by his deep and broad knowledge and I am indebted to his patient guidance.

I would like to thank many professors in the Department of Statistics at the Ohio State University. I am grateful to Dr. Angela Dean, who led my way to the research of computer experiments in 2010 and guided me in the following 4 years. I thank Dr. Chris Holloman, who gave me the opportunity to work in the statistical consulting service (SCS) and guided me not only on the statistical methods but also on the consulting techniques. I also thank Dr. Elizabeth Stasny, the graduate student chair, for her warm-hearted encouragement and support. I would like to thank Drs. Peter Craigmile, Matthew Pratola, Angela Dean, Thomas Santner and Chris Hans for being
in my thesis defense committee and candidacy exam committee and providing me with insightful comments.

I would like to thank all the students joining in the department in 2009. During the times of exams and thesis research, we worked and cheered together. Thank you for the support. Thanks for the sweet and colorful memories!

Finally, the entire episode of my life in Columbus would not be possible without the unconditional support and love from my family in China. I am indebted to my parents, Zhang, Jie and Huangfu, Yun, for their everlasting love. This thesis is dedicated to them.
Vita

May, 1985 ........................................... Born - Changchun, China

2008 .............................................. B.S. Mathematics, Nankai University

2011 .............................................. M.S. Statistics,
       The Ohio State University

2010-present ................................. Graduate Teaching Associate,
       The Ohio State University

Fields of Study

Major Field: Statistics
Studies in: The design and analysis of computer experiments
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iv</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>v</td>
</tr>
<tr>
<td>Vita</td>
<td>vii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xiv</td>
</tr>
<tr>
<td>1. Introduction to Computer Experiments</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Physical Experiments</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Computer Experiments</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Gaussian Stochastic Process Model</td>
<td>2</td>
</tr>
<tr>
<td>1.3.1 Empirical Best Linear Unbiased Predictor</td>
<td>9</td>
</tr>
<tr>
<td>1.3.2 Hierarchical Predictive Distribution</td>
<td>13</td>
</tr>
<tr>
<td>1.4 Designs for Computer Experiments</td>
<td>15</td>
</tr>
<tr>
<td>1.5 Sequential Designs for Model Predictions</td>
<td>16</td>
</tr>
<tr>
<td>1.5.1 Sequential MSPE Design</td>
<td>17</td>
</tr>
<tr>
<td>1.5.2 Sequential IMSPE Design</td>
<td>17</td>
</tr>
<tr>
<td>1.5.3 Sequential Cross Validation Prediction Error (XVPE) Design</td>
<td>18</td>
</tr>
<tr>
<td>2. Introduction to Computer Experiments with Quantitative and Qualitative Inputs</td>
<td>20</td>
</tr>
<tr>
<td>2.1 Gaussian Process Models With Quantitative and Qualitative inputs</td>
<td>22</td>
</tr>
<tr>
<td>2.1.1 Cross-Correlation Structure Model</td>
<td>22</td>
</tr>
<tr>
<td>2.1.2 Hypersphere Parameterization Model</td>
<td>26</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>2.1.3</td>
<td>Hierarchical Quantitative-Qualitative Variable Model</td>
</tr>
<tr>
<td>2.1.4</td>
<td>ANOVA Kriging Model</td>
</tr>
<tr>
<td>2.2</td>
<td>Designs for Computer Experiments with Both Quantitative and Qualitative Inputs</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Nested Space-Filling Designs</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Nested Latin Hypercube Designs</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Sliced Latin Hypercube Designs</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Clustered Sliced Latin Hypercube Designs</td>
</tr>
<tr>
<td>3.</td>
<td>Indicator-Based Gaussian Stochastic Process Models for Computer Experiments with Both Quantitative and Qualitative Inputs</td>
</tr>
<tr>
<td>3.1</td>
<td>Pairwise-Indicator Gaussian Stochastic Process Model</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Motivation and Notation</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Pairwise-Indicator Data Transformation</td>
</tr>
<tr>
<td>3.1.3</td>
<td>The Equivalence Property</td>
</tr>
<tr>
<td>3.1.4</td>
<td>Remarks</td>
</tr>
<tr>
<td>3.2</td>
<td>Individual-Indicator Gaussian Stochastic Process Model</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Motivation</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Qualitative Indicator Vector Construction</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Individual Indicator Cross-Correlation Matrix</td>
</tr>
<tr>
<td>3.3</td>
<td>A Simple Application of the PIGaSP Model and IIGaSP Model on Commercial Software</td>
</tr>
<tr>
<td>3.4</td>
<td>The Effect on Prediction Accuracy from the Restrictions on the Range of the Cross-Correlation Parameters</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Motivating Example</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Simulation Comparisons with the Candidate Models</td>
</tr>
<tr>
<td>4.</td>
<td>Experimental Designs for Computer Experiments with Both Quantitative and Qualitative Inputs</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparison of SLHD and CSLHD</td>
</tr>
<tr>
<td>4.2</td>
<td>Partial SLHD and Partial CSLHD</td>
</tr>
<tr>
<td>4.3</td>
<td>The Effect of the Initial Designs on the Model Fitting</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Simple 2D Functions</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Branin Function</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Goldstein Price Function</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Branin and Goldstein Price Functions</td>
</tr>
<tr>
<td>4.3.5</td>
<td>Summary</td>
</tr>
<tr>
<td>4.4</td>
<td>Sequential ANOVA Kriging Algorithm for Computer Experiments with Quantitative and Qualitative Factors</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Definitions and Notations</td>
</tr>
</tbody>
</table>
Bibliography


List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Training Data of Example 3.3.1</td>
<td>60</td>
</tr>
<tr>
<td>3.2 Summary of the RMSEs in Example 3.3.1</td>
<td>62</td>
</tr>
<tr>
<td>3.3 Training data for Example 3.4.1</td>
<td>67</td>
</tr>
<tr>
<td>3.4 Comparison of the parameter estimates and RMSEs among three different restrictions on the range</td>
<td>68</td>
</tr>
<tr>
<td>3.5 Comparison of the parameter estimates and the RMSEs among different methods</td>
<td>71</td>
</tr>
<tr>
<td>5.1 Training data for Example 5.3.1</td>
<td>118</td>
</tr>
<tr>
<td>5.2 Comparison of the estimations and the RMSE</td>
<td>119</td>
</tr>
<tr>
<td>5.3 Two Steps Method comparison for Example 5.3.1</td>
<td>122</td>
</tr>
<tr>
<td>5.4 Two Steps Method comparison for Example 5.3.2</td>
<td>126</td>
</tr>
<tr>
<td>5.5 Average parameter estimations and RMSEs for general Kronecker product structure model of Example 5.6.1</td>
<td>135</td>
</tr>
<tr>
<td>5.6 Average parameter estimations and RMSEs for ANOVA decomposition model of Example 5.6.1</td>
<td>135</td>
</tr>
<tr>
<td>5.7 Input variables in the borehole example</td>
<td>140</td>
</tr>
<tr>
<td>5.8 Borehole example: three qualitative factors and their levels</td>
<td>140</td>
</tr>
</tbody>
</table>
5.9  Comparison of the RMSEs for borehole example . . . . . . . . . 143

5.10  Comparison of the sequential design and the RMSE . . . . . . . 146
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Effect of Varying the Power on the Sample Paths of a Gaussian Stochastic Process with Power Exponential Correlation Function. Four draws from a zero mean, unit variance with the exponential correlation (1.3.3) having fixed $\phi = 1.0$ with $\alpha = 2$ (solid lines), $\alpha = 0.75$ (dashed lines), and $\alpha = 0.20$ (dotted lines)</td>
<td>5</td>
</tr>
<tr>
<td>1.2 Effect of Varying the Scale Parameter on the Sample Paths of a Gaussian Stochastic Process with Power Exponential Correlation Function. Four draws from a zero mean, unit variance with the exponential correlation (1.3.3) having fixed $\alpha = 2.0$ with $\phi = 4$ (solid lines), $\phi = 16$ (dashed lines), and $\phi = 100$ (dotted lines)</td>
<td>6</td>
</tr>
<tr>
<td>2.1 An example of NSFD with $n_1 = 64$, $n_2 = 32$</td>
<td>37</td>
</tr>
<tr>
<td>2.2 An example of NLHD with three layers: $n_1 = 24$, $n_2 = 12$ and $n_3 = 6.$</td>
<td>39</td>
</tr>
<tr>
<td>2.3 An example of SLHD with $n=20$, $d=2$, $m=4$</td>
<td>42</td>
</tr>
<tr>
<td>2.4 An example of SLHD with $n=8$, $d=2$, $m=2$</td>
<td>44</td>
</tr>
<tr>
<td>2.5 An example of CSLHD with $n=20$, $d=2$, $m=4$</td>
<td>47</td>
</tr>
<tr>
<td>3.1 Response curves in Example 3.3.1</td>
<td>61</td>
</tr>
<tr>
<td>3.2 A comparison of the Example 3.3.1 with the true curves for the four models (Separate model, PIGaSP model, IIGaSP Model with $\Omega$ a diagonal matrix, and Hyperspherical Parameterization Model). The left top is for $Y(X,1)$, the right top is for $Y(X,2)$, the left bottom is for $Y(X,3)$, and the right bottom is for $Y(X,4)$</td>
<td>63</td>
</tr>
</tbody>
</table>
3.3 True Curves of Example 3.4.1 ........................................ 67

3.4 A comparison of the Example 3.4.1 with the true curves for the different ranges of the candidate models. The left-top panel is the plot for $Y(X, 1)$, the right-top one is the plot for $Y(X, 2)$, and the bottom one is the plot for $Y(X, 3)$ ........................................ 72

4.1 A comparison of SLHD and CSLHD with $n=20, d=2, m=4$ ........ 74

4.2 A partial SLHD with $n = 20, d = 1, m = 4, t_1 = 2$ ............... 77

4.3 A partial SLHD with $n = 40, d = 2, m = 4, t_1 = 3$: the top four panels are the designs sampled from a SLHD, the bottom left panel is the common design for each qualitative level, the bottom right panel is the partial SLHD including all the qualitative levels. ............... 79

4.4 A partial CSLHD with $n = 40, d = 2, m = 4, t_1 = 3$: the top four panels are the designs sampled from a CSLHD, the bottom left panel is the common design for each qualitative level, the bottom right panel is the partial CSLHD including all the qualitative levels. ............... 82

4.5 The true surfaces of the simple 2D functions ....................... 85

4.6 The boxplot comparison of the simple 2D functions RMSEs. The left 7 are for $t = 10$, the middle 7 are for $t = 20$, and the right 7 are for $t = 40$. ‘ST’ means the standard LHD, ‘SP’ means the $k$ LHD, ‘SA’ means the same LHD, ‘SL’ means the SLHD, ‘CS’ means the CSLHD, ‘PS’ means the partial SLHD, and ‘PC’ means the partial CSLHD . 86

4.7 The true surfaces of the Branin functions ......................... 87

4.8 The boxplot comparison of the Branin functions’ RMSEs. The left 7 are for $t = 10$, the middle 7 are for $t = 20$, and the right 7 are for $t = 40$. ‘ST’ means the standard LHD, ‘SP’ means the $k$ LHD, ‘SA’ means the same LHD, ‘SL’ means the SLHD, ‘CS’ means the CSLHD, ‘PS’ means the partial SLHD, and ‘PC’ means the partial CSLHD . 89

4.9 The true surfaces of the Goldstein-Price functions .................. 91
The boxplot comparison of the Goldstein-Price functions RMSEs. The left 7 are for \( t = 10 \), the middle 7 are for \( t = 20 \), and the right 7 are for \( t = 40 \). 'ST' means the standard LHD, 'SP' means the \( k \)LHD, 'SA' means the same LHD, 'SL' means the SLHD, 'CS' means the CSLHD, 'PS' means the partial SLHD, and 'PC' means the partial CSLHD.

The true surfaces of the Branin and Goldstein-Price functions.

The boxplot comparison of the Branin and Goldstein-Price functions RMSEs. The left 7 are for \( t = 10 \), the middle 7 are for \( t = 20 \), and the right 7 are for \( t = 40 \). 'ST' means the standard LHD, 'SP' means the \( k \)LHD, 'SA' means the same LHD, 'SL' means the SLHD, 'CS' means the CSLHD, 'PS' means the partial SLHD, and 'PC' means the partial CSLHD.

The true response surfaces in Example 4.4.1.

The design with the added points in Example 4.4.1.

Sequential ANOVA kriging predictor for the TR in Example 4.4.1.

Plot of the true response surfaces in Example 4.4.2.

The design with added points in Example 4.4.2.

Sequential ANOVA kriging predictor for the TR in Example 4.4.2.

The boxplot comparison of the HP model predictions and composite model predictions with the true curves.

Comparison of the True Curves of Example 5.3.1.

Comparison of the HP model predictions and composite model predictions with the true curves.

The Cholesky decomposition in the middle and the EFA decomposition on the right. Boxplot comparison (right panel) of the improvement of the two decomposition methods with the Cholesky decomposition on the left, the EFA decomposition on the right.

The boxplot comparison of the HP model predictions and composite model predictions with the true curves.
5.4 Boxplot comparison and the improvements of two steps methods using CSLHD designs for Example 5.3.1 with the Step One on the left, the Cholesky decomposition in the middle and the EFA decomposition on the right. Boxplot comparison (right panel) of the improvement of the two decomposition methods with the Cholesky decomposition on the left, the EFA decomposition on the right. 123

5.5 The True Curves of Example 5.3.2 125

5.6 Boxplot comparison (left panel) of the three methods using the same LHD designs each level for Example 5.3.2 with the Step One on the left, the Cholesky decomposition in the middle and the EFA decomposition on the right. Boxplot comparison (right panel) of the improvement of the two decomposition methods with the Cholesky decomposition on the left, the EFA decomposition on the right. 126

5.7 Boxplot comparison and the improvements of two steps methods using CSLHD designs for Example 5.3.2 with the Step One on the left, the Cholesky decomposition in the middle and the EFA decomposition on the right. Boxplot comparison (right panel) of the improvement of the two decomposition methods with the Cholesky decomposition on the left, the EFA decomposition on the right. 127

5.8 The True Curves of Example 5.4.1 133

5.9 Boxplot of the RMSEs of Example 5.6.1 for overall fit situation 136

5.10 Boxplot of the RMSEs of Example 5.6.1 for (1,1) individual fit situation 136

5.11 The True Surfaces of Example 5.6.2 138

5.12 Boxplot Comparison of RMSEs in Example 5.6.2: The left 5 boxplots are based on the Kronecker product decomposition method, the middle 5 boxplots are based on the ANOVA decomposition with the constraints, the right 5 boxplots are based on the regular ANOVA decomposition method. 139

5.13 CSLHD 146
5.14 Comparison of the Sequential Methods: the left top one is the MSPE criterion, the right top one is the adaptive arithmetic mean criterion, the left bottom one is the geometric mean criterion, the right bottom one is the harmonic mean criterion.
Chapter 1: Introduction to Computer Experiments

1.1 Physical Experiments

A physical experiment measures a stochastic response corresponding to the set of physical treatment input variables, which can provide insight into cause-and-effect by demonstrating what outcome occurs when a particular factor is manipulated (Dean and Voss 1999). In order to improve the experimental validity, statisticians develop three basic techniques: replication, blocking, and randomization. The first two help to increase precision in the experiment; the last one is used to decrease bias.

1.2 Computer Experiments

Physical experiments play an important role in agriculture, industry, and medical research. However, physical experiments can sometimes be difficult or even impossible to run. There can be difficulties conducting physical experiments because of impracticable, ethical, or economical issues. For example, physical experiments studying climate changes, the performance of prosthetic devices, and the failure depth of pockets in sheet metal are hard or impossible to conduct (See Santner, Williams and Notz (2003), Chapter 1 and Fang, Li and Sudjianto (2005), Chapter 1 for details.) In these
situations, computer experiments are becoming desirable surrogates for physical experiments. A computer experiment uses computer code to simulate a “response” of a complex physical process given any set of input variables. Instead of conducting a time-consuming, costly, or even impossible physical experiment, computer experiments can serve as a proxy for the physical process.

Compared to physical experiments, a computer experiment can produce either a deterministic answer or a stochastic response for a given set of input values. The most commonly used computer experiments are deterministic. Therefore, in this thesis, we assume that computer experiments are deterministic. The traditional techniques of randomization, blocking and replication are irrelevant in solving design and analysis problems associated with deterministic computer experiments since the replicates produce the same output value. Computer experiments can involve codes that require running times from minutes to days (Santner et al. 2003). Thus, statistical predictive models are typically needed to infer the responses at input points that have not been run.

**1.3 Gaussian Stochastic Process Model**

Gaussian Stochastic Processes (GaSPs) are popular for modeling outputs of computer experiments (Santner et al. 2003). Their flexibility, tractability, and interpolating features make them generally the prevalent models for the study of computer experiments (Santner et al. 2003, Fang et al. 2005).

A real-valued stochastic process \( \{ \epsilon(x) \} \), where \( x \) is defined on the region \( X \in \mathbb{R}^d \), is a Gaussian process if all the finite-distributions have a multivariate normal distribution. That is, for any choice of the distinct values \( x_1, \ldots, x_n \), the random
vector $\mathbf{\epsilon} = (\epsilon(x_1), \ldots, \epsilon(x_n))^T$ has a multivariate normal distribution with mean
vector $\mathbf{\mu} = E(\mathbf{\epsilon})$ and covariance matrix $\Sigma = \text{cov}(\mathbf{\epsilon}, \mathbf{\epsilon})$.

The *mean* and *covariance* functions of a Gaussian process are defined by

$$\mu(x) = E(\epsilon(x)), x \in \mathcal{X}$$

and

$$C(x_1, x_2) = \text{cov}(\epsilon(x_1), \epsilon(x_2)) x_1, x_2 \in \mathcal{X}$$

respectively.

Furthermore, $\{\epsilon(x)\}$ is said to be *stationary* if

1. $\mu(x) = E(\epsilon(x)) = \mu$ is independent of $x$, and
2. $C(x_t + x_h, x_t) = \text{cov}(\epsilon(x_t + x_h), \epsilon(x_t))$ is independent of $x_t$ for all $x_h$.

For stationary processes, the covariance function $C$ is usually expressed as a function on $\mathcal{X}$ instead of on $\mathcal{X} \times \mathcal{X}$. That is, we define $C(x_h) = C(x_t + x_h, x_t)$ as the covariance function of the stationary process. Accordingly, the correlation function can be defined by

$$R(x_h) = R(x_t + x_h, x_t) = \text{Corr}(\epsilon(x_t + x_h), \epsilon(x_t))$$

For future development, we first briefly review GaSP models with quantitative variables. We let $x = (x_1, \ldots, x_d)^T \in [0,1]^d$ (or for rectangular regions, $x$ can be scaled into $[0,1]^d$) denote an input having $d$ components and $y(x)$ denote the corresponding response value. The GaSP model views $y(\cdot)$ as a realization of the stochastic process

$$Y(x) = f^T(x)\beta + \epsilon(x) \quad (1.3.1)$$
where $f(x) = (f_1(x), \cdots, f_q(x))^T$ is a $q \times 1$ vector with the elements being real-valued functions of the inputs $x$, and $\beta = (\beta_1, \cdots, \beta_q)^T$ is a $q \times 1$ vector of unknown regression parameters. The process $\epsilon(\cdot)$ is a stationary Gaussian process with mean 0 and covariance between two responses $\epsilon(x_1)$ and $\epsilon(x_2)$

$$\text{Cov}(\epsilon(x_1), \epsilon(x_2)) = \sigma^2_{\epsilon} R(x_1 - x_2), \quad (1.3.2)$$

where $\sigma^2_{\epsilon}$ denotes the process variance and $R(x_1 - x_2)$ denotes the correlation function of $\epsilon(\cdot)$.

One popular choice of the correlation function is the power exponential correlation function; i.e.

$$R(x_1 - x_2) = \prod_{i=1}^{d} \exp\{-\phi_i |x_{i1} - x_{i2}|^{\alpha_i}\} \quad (1.3.3)$$

where $\phi_i \geq 0$ and $0 < \alpha_i \leq 2$ for $i = 1, \cdots, d$. The power parameters $\alpha_i$’s control the smoothness of the random function $Y(\cdot)$. For $0 < \alpha_i < 2$, the corresponding realizations of the process are continuous but not differentiable. This can be seen in the bottom two panels of Figure 1.1. If one knows that the physical process being modeled by the simulator is smooth in $x \in \mathcal{X}$, $\alpha_i = 2$ should be used. In particular, when all $\alpha_i$’s are fixed at 2, the correlation function is called the product Gaussian correlation function, and $Y(x)$ is infinitely differentiable (Santner et al. 2003). The draws in the top panel of Figure 1.1 shows the sample paths from a GaSP with Gaussian correlation function.

The scale parameters $\phi_i$’s measure the roughness of the response surface. As $\phi_i$ increases, the dependence between the response at two input points decreases but is never zero. Figure 1.2 shows the effect of varying the scale parameters on the sample paths of a GaSP with Gaussian correlation function. As shown in the figure, as the
scale parameter $\phi_i$ increases, the correlation for each fixed pair of inputs decreases. That means the process exhibits less dependence for “nearby” $x$.

Figure 1.1: Effect of Varying the Power on the Sample Paths of a Gaussian Stochastic Process with Power Exponential Correlation Function. Four draws from a zero mean, unit variance with the exponential correlation (1.3.3) having fixed $\phi = 1.0$ with $\alpha = 2$ (solid lines), $\alpha = 0.75$ (dashed lines), and $\alpha = 0.20$ (dotted lines).

The model parameters can be estimated by two methodologies. The first one is using the frequentist methodology, which includes maximum likelihood estimation
Figure 1.2: Effect of Varying the Scale Parameter on the Sample Paths of a Gaussian Stochastic Process with Power Exponential Correlation Function. Four draws from a zero mean, unit variance with the exponential correlation (1.3.3) having fixed $\alpha = 2.0$ with $\phi = 4$ (solid lines), $\phi = 16$ (dashed lines), and $\phi = 100$ (dotted lines)

(MLE), restricted maximum likelihood estimation, cross validated estimation, and posterior mode estimation. The second one is the Bayesian hierarchical methodology, which assumes a prior distribution on each of the model parameters and then derives the corresponding posterior distribution of the parameters given the data. Specifically, let $\theta = (\beta, \sigma^2, \phi)$ denote all the model parameters, $[\theta]$ denote the prior, and $y =$
\[(y(x_1), ..., y(x_n))^T\] denote the runs from the computer simulation. The posterior distribution of \(\theta\), \([\theta|y]\), is proportional to the product of the prior density, \([\theta]\), and the likelihood, \([y|\theta]\); i.e.,
\[\text{[\theta|y] } \propto \text{[\theta]} \times \text{[y|\theta]} .\] (1.3.4)

Furthermore, the posterior predictive distribution of \([Y(x_0)|y]\) can be derived from the posterior draw of the parameters. The details of the Bayesian methodology will be discussed in Subsection 1.3.2. In Chapter 3 and later, we will use the frequentist methodology to make inference about the model parameters and to predict unknown responses.

Given the \(n\) observations from the computer simulator, the goal is to predict the response \(Y(x_0)\) at a new data point \(x_0\). The best linear unbiased predictor (BLUP) provides the linear predictor \(\hat{Y}(x_0) = c^T(x_0)y\) that minimizes the mean squared prediction error (MSPE)
\[\text{MSPE}(\hat{Y}(x_0)) = E[(c^T(x_0)y - Y(x_0))^2],\] (1.3.5)
subject to the unbiasedness constraint \(E[c^T(x_0)y] = E[Y(x_0)]\).

The minimum MSPE predictor can be derived as \(E(Y(x_0)|y)\). In particular, when the correlation function \(R(\cdot)\) and the variance \(\sigma^2\) are known, we can derive the corresponding BLUP more easily.

Assume \(F\) is the \(n \times q\) matrix whose \(j^{th}\) row is \(f^T(x_j)\), and \(R\) is the \(n \times n\) matrix whose \((i,j)^{th}\) element is \(R(x_i - x_j)\). Furthermore, assume \(r_0 = (R(x_1 - x_0), \cdots, R(x_n - x_0))^T\) is the \(n \times 1\) vector of the correlations between outputs at the existing training data points \(y\), and \(Y(x_0)\). Then, the BLUP can be derived as
\[\hat{Y}_{UK}(x_0) = f^T(x_0)\hat{\beta} + r_0^T R^{-1}(y - F\hat{\beta})\] (1.3.6)
where \( \hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} y \) is the generalized least-squares estimate of \( \beta \).

In geostatistics, the BLUP in (1.3.6) is called the universal kriging (UK) estimate. Accordingly, the MSPE of the BLUP (1.3.6) is then given by

\[
\text{MSPE}(\hat{Y}_{UK}(x_0)) = \sigma^2 \left[ 1 - r_0^T R^{-1} r_0 + \left( f(x_0) - r_0^T R^{-1} F \right) \left( F^T R^{-1} F \right)^{-1} \left( f(x_0) - r_0^T R^{-1} F \right)^T \right].
\] (1.3.7)

In practice, \( f^T(x)\beta \) in (1.3.1) is usually simply assumed to be a constant mean term, \( \beta \), which implies that \( f(x) \equiv 1 \) and \( F \) is an all-one vector \( 1 \). Therefore the BLUP can be simplified as

\[
\hat{Y}_{OK}(x_0) = \hat{\beta} + r_0^T R^{-1} (y - 1\hat{\beta}),
\] (1.3.8)

where \( \hat{\beta} = (1^T R^{-1} 1)^{-1} 1^T R^{-1} y \). Then the corresponding MSPE is given by

\[
\text{MSPE}(\hat{Y}_{OK}(x_0)) = \sigma^2 \left[ 1 - r_0^T R^{-1} r_0 + \frac{(1 - r_0^T R^{-1} 1)^2}{1^T R^{-1} 1} \right].
\] (1.3.9)

In geostatistics, the BLUP in (1.3.8) is usually called the ordinary kriging (OK) estimator (Cressie 1993, Santner et al. 2003). In addition, if we assume the constant mean is known, the corresponding BLUP is called the simple kriging (SK) estimator (Cressie 1993, Santner et al. 2003). The BLUP and the corresponding MSPE are given by

\[
\hat{Y}_{SK}(x_0) = \beta + r_0^T R^{-1} (y - 1\beta);
\] (1.3.10)

\[
\text{MSPE}(\hat{Y}_{SK}(x_0)) = \sigma^2 \left[ 1 - r_0^T R^{-1} r_0 \right].
\] (1.3.11)

The BLUP can also be viewed as the conditional distribution of \( Y(x_0) \). Given the parameters \( \theta \), the random function \( Y(x_0) \) and the random vector \( y \) have a joint
multivariate normal distribution. That is,

\[
\begin{bmatrix}
  Y(x_0) \\
  y
\end{bmatrix} \sim MN \left( \begin{pmatrix}
  \hat{f}^T(x_0) \\
  F
\end{pmatrix} \beta, \sigma^2 \left( 1 + r_0^T R^{-1} r_0 \right) \right)
\] (1.3.12)

Thus, given the training data \( y \) and parameters \( \theta \), \( Y(x_0) \) has the conditional distribution

\[
Y(x_0)|y, \theta \sim N(f^T(x_0)\beta + r_0^T R^{-1}(y - F\beta), \sigma^2(1 - r_0^T R^{-1} r_0))
\] (1.3.13)

Notice that the conditional mean \( f^T(x_0)\beta + r_0^T R^{-1}(y - F\beta) \) is the same as the SK predictor of \( Y(x_0) \) in (1.3.10), while the conditional predictive variance \( \sigma^2(1 - r_0^T R^{-1} r_0) \) is the same as the MSPE of \( \hat{Y}_{SK}(x_0) \) in (1.3.11).

In practice, however, the parameter \( \theta = (\beta, \sigma^2, \phi) \) is usually unknown or only partially known. In these situations, the frequentist approach estimates the parameters and derives the corresponding empirical BLUP (EBLUP), while the Bayesian approach derives or simulates the posterior distribution of \( Y(x_0) \).

In Section 1.3.1, we discuss the EBLUP and derive the predictions of \( Y(x_0) \) as well as the predictive uncertainty. In Section 1.3.2, we derive the predictive distribution for the output of computer experiments under two hierarchical models.

### 1.3.1 Empirical Best Linear Unbiased Predictor

When \( \beta \) is unknown, we consider two cases based on the correlation functions.

1. If the correlation function \( R(\cdot) \) is known, the BLUP of \( y(x_0) \) is

\[
\hat{Y}(x_0) = f^T(x_0)\hat{\beta} + r_0^T R^{-1}(y - F\hat{\beta}),
\] (1.3.14)

where \( \hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} y \) is the generalized least squares estimator of \( \beta \). Both \( r_0 \) and \( R \) are specified when the correlation function \( R(\cdot) \) is known.
The estimate of the predictive uncertainty is given by

\[
\hat{\sigma}^2 \epsilon \left[ 1 - r_0^T R^{-1} r_0 + (f(x_0) - r_0^T R^{-1} F)(F^T R^{-1} F)^{-1}(f(x_0) - r_0^T R^{-1} F)^T \right],
\]

(1.3.15)

where \( \hat{\sigma}^2 = \frac{1}{n} (y - F \hat{\beta})^T R^{-1} (y - F \hat{\beta}) \) is the MLE of \( \sigma^2 \epsilon \).

2. When the correlation function \( R(\cdot) \) is unknown, we estimate the correlation parameter \( \phi \) and then derive the estimated correlation matrix \( \hat{R} \). Furthermore, by plugging the estimated correlation matrix \( \hat{R} \) in (1.3.14) and (1.3.15), we can obtain the EBLUP and corresponding predictive uncertainty of \( Y(x_0) \).

There are several ways to estimate the correlation parameter \( \phi \). We assume that the correlation function is parametric with \( R(\cdot) = R(\cdot|\phi) \) so that \( R = R(\phi) \) and \( r_0 = r_0(\phi) \)

**Maximum Likelihood EBLUPs** Under the normality assumption, the log likelihood is

\[
l(\beta, \sigma^2 \epsilon, \phi) = -\frac{1}{2} \left[ n \log \sigma^2 + \log(\det(R)) + (y - F \beta)^T R^{-1} (y - F \beta) / \sigma^2 \right]
\]

(1.3.16)

Given \( \phi \), the MLE of \( \beta \) is its generalized least squares estimate \( \hat{\beta}(\phi) = (F^T R^{-1} F)^{-1} F^T R^{-1} y \) and the MLE of \( \sigma^2 \) is \( \hat{\sigma}^2(\phi) = \frac{1}{n} (y - F \hat{\beta})^T R^{-1} (y - F \hat{\beta}) \).

Substituting these MLEs into equation (1.3.16), we obtain the maximum of (1.3.16) depends on \( \phi \) alone. Thus the MLE chooses \( \hat{\phi} \) to minimize

\[
n \log \hat{\sigma}^2(\phi) + \log(\det(R(\phi))).
\]

(1.3.17)
Restricted Maximum Likelihood EBLUPs The restricted maximum likelihood (REML) estimator of $\phi$ maximizes the likelihood of a maximal set of linearly independent combinations of the $y$ where each linear combination is orthogonal to $F\beta$, the mean vector of $y$. It can be shown that the REML estimator of $\sigma^2_\epsilon$ is

$$
\hat{\sigma}^2_\epsilon(\phi) = \frac{n}{n - q} \hat{\sigma}^2_\epsilon(\phi) = \frac{1}{n - q} \left( y - F\hat{\beta} \right)^T R^{-1} \left( y - F\hat{\beta} \right).
$$

(1.3.18)

The REML estimator of $\phi$ is the minimizer of

$$
(n - q) \log \hat{\sigma}^2_\epsilon(\phi) + \log(\det(R(\phi)))
$$

(1.3.19)

Cross-Validation EBLUPs Cross validation is another approach to estimate the $\phi$. Cross-validated predictions and corresponding prediction errors are computed at each of the training sites using a subset of the existing observations (Santner et al. 2003). The leave-one-out cross-validated estimator of $\phi$ minimizes the empirical mean squared prediction error

$$
XVPE(\phi) = \sum_{i=1}^{n} (\hat{Y}_{-i}(\phi) - y(x_i))^2,
$$

(1.3.20)

where $\hat{Y}_{-i}(\phi)$ ($i = 1, \cdots, n$) is the EBLUP in (1.3.14) based on all the data except $(x_i, y(x_i))$ when $\phi$ is the correlation parameter.

Posterior Mode EBLUPs The minimum MSPE predictor $E(Y(x_0)|y)$ can be written as

$$
E(Y(x_0)|y) = E(E(Y(x_0)|y, \phi)|y) = \int E(Y(x_0)|y, \phi)|\phi|y d\phi,
$$

(1.3.21)
where \( E(Y(x_0)|y, \phi) \) can be derived from
\[
E(Y(x_0)|y, \phi) = \int_{\beta, \sigma^2} [Y(x_0), \beta, \sigma^2|y, \phi] d\beta d\sigma^2 \tag{1.3.22}
\]
\[
= \int_{\beta, \sigma^2} [Y(x_0)|\beta, \sigma^2, \phi, y][\beta, \sigma^2|y, \phi] d\beta d\sigma^2.
\]

Even if we know \( E(Y(x_0)|y, \phi) \), \( [\phi|y] \) generally cannot be expressed in a closed form. The empirical Bayesian motivation is to assume that \( [\phi|y] \) is degenerate with a mass located at its mode \( \hat{\phi} \), and estimate (1.3.21) by \( E(Y(x_0)|y, \hat{\phi}) \).

If we choose the conjugate priors for \( [\beta, \sigma^2] \), then we can derive \( [\beta, \sigma^2|y, \phi] \). Furthermore, \( [\beta, \sigma^2|y, \phi] \) can be expressed as
\[
[\beta, \sigma^2|y, \phi] = \frac{[\beta, \sigma^2][y|\beta, \sigma^2, \phi]}{[y|\phi]} \tag{1.3.23}
\]

Therefore,
\[
[y|\phi] = \frac{[\beta, \sigma^2][y|\beta, \sigma^2, \phi]}{[\beta, \sigma^2|y, \phi]} \tag{1.3.24}
\]
and
\[
[\phi|y] \propto [y|\phi][\phi]. \tag{1.3.25}
\]

The posterior mode of \( [\phi|y] \) can be derived by
\[
\hat{\phi} = \text{argmax}_{\phi} [y|\phi][\phi].
\]

By choosing conjugate priors for \( [\beta, \sigma^2] \), we can express \( [\beta, \sigma^2|y, \phi] \) in a closed form. Thus, (1.3.25) will also be in a closed form. Furthermore, by plugging \( \hat{\phi} \) into \( E(Y(x_0)|y, \hat{\phi}) \), we can derive the empirical Bayesian Predictor:
\[
E(Y(x_0)|y, \hat{\phi}) = \int_{\beta, \sigma^2} [Y(x_0)|\beta, \sigma^2, \hat{\phi}, y][\beta, \sigma^2|y, \hat{\phi}] d\beta d\sigma^2. \tag{1.3.26}
\]
1.3.2 Hierarchical Predictive Distribution

A predictive distribution for the random variable \( Y(x_0) \) is meant to capture all the information about \( Y(x_0) \) that is contained in \( y \). The Bayesian predictive distribution is \([Y(x_0)|y]\), where one puts a prior, \([\theta]\), on the unknown parameters \( \theta \). One can numerically approximate \([Y(x_0)|y]\) by simulating \([\theta|y]\) and approximating the integral

\[
[Y(x_0)|y] \propto \int [Y(x_0)|y, \theta][\theta|y]d\theta
\]

(1.3.27)

by

\[
\sum_{i=1}^{m} [Y(x_0)|y, \theta^{(i)}]/m
\]

where \( m \) is the number of draws from \([\theta|y]\), and \( \theta^{(i)} \) denotes the \( i^{th} \) \( (i = 1, \ldots, m) \) draw from \([\theta|y]\). A general algorithm for drawing \([\theta|y]\) includes a Gibbs Sampling step for the posterior distribution of \([\sigma^2_{\epsilon}|\text{rest}]\) and \([\beta|\text{rest}]\), and a Metropolis-Hastings(MH) step for \([\phi|\text{rest}]\). The Law of Large Numbers guarantees that the approximation converges to the true value of \([Y(x_0)|y]\) as \( m \) tends to infinity.

Furthermore, we can also derive the corresponding minimum MSPE predictor of \( Y(x_0) , E(Y(x_0)|y) \). Intuitively, the predictor and the predictive uncertainty are given by

\[
E(Y(x_0)|y) = E(E(Y(x_0)|y, \theta)) \quad (1.3.28)
\]

\[
\text{Var}(Y(x_0)|y) = \text{Var}(E(Y(x_0)|y, \theta)) + E(\text{Var}(Y(x_0)|y, \theta)) \quad (1.3.29)
\]

where for each draw of \( \theta^{(i)} \), \( E(Y(x_0)|y, \theta^{(i)}) = f^T(x_0)\beta^{(i)} + r_0^{(i)^T}R^{(i)^{-1}}(y - F\beta^{(i)}) \) and \( \text{Var}(Y(x_0)|y, \theta^{(i)}) = \sigma^2_{\epsilon}(1 - r_0^{(i)^T}R^{(i)^{-1}}r_0^{(i)}) \) are the numerical result of the conditional mean and conditional variance in (1.3.13). The Strong Law of Large Numbers guarantees that the sample mean and sample variance of the sequence
of \(E(Y(x_0)|y, \theta^{(i)})\)'s converge to \(E(E(Y(x_0)|y, \theta))\) and \(\text{Var}(E(Y(x_0)|y, \theta))\), respectively, as \(m\) tends to infinity. In addition, the sample mean of the sequence of \(\text{Var}(Y(x_0)|y, \theta^{(i)})\)'s converges to \(E(\text{Var}(Y(x_0)|y, \theta))\). Thus, the predictor and the predictive uncertainty can be obtained from (1.3.28) and (1.3.29).

For some special cases, we can derive a closed form expression for the predictive distribution, \([Y(x_0)|y]\). We emphasize that in these cases the models of the predictive distribution assume that the correlation structure is known, in which case \(R\) and \(r_0\) are known. The model assumes that \(\beta\) is unknown and \(\sigma^2\) is known.

Suppose \((Y(x_0), y)\) follows a two-stage model with known \(\sigma^2\), which satisfies (1.3.12).

(a) If

\[
\beta \sim \text{MVN}(b_0, \tau^2 V_0) \tag{1.3.30}
\]

where \(b_0, V_0, \) and \(\tau^2\) are known, then \(Y(x_0)\) has the predictive distribution

\[
[Y(x_0)|y] \sim \text{Normal}(f^T(x_0)\mu_\beta + r_0^TR^{-1}(y - F\mu_\beta), \sigma_0^2), \tag{1.3.31}
\]

where

\[
\mu_\beta = \left( \frac{F^TR^{-1}F}{\sigma_\epsilon^2} + \frac{V_0^{-1}}{\tau^2} \right)^{-1} \left( \frac{F^TR^{-1}y}{\sigma_\epsilon^2} + \frac{V_0^{-1}b_0}{\tau^2} \right),
\]

and

\[
\sigma_0^2 = \sigma_\epsilon^2 \left\{1 - (f(x_0)^T, r_0^T) \begin{bmatrix} -\frac{\sigma_\epsilon^2}{\tau^2} V_0^{-1} & F^T \\ F & R \end{bmatrix}^{-1} \begin{bmatrix} f(x_0) \\ r_0 \end{bmatrix} \right\}.
\]

(b) If

\[
[\beta] \sim 1, \tag{1.3.32}
\]

then

\[
[Y(x_0)|y] \sim \text{Normal}(f^T(x_0)\hat{\beta} + r_0^TR^{-1}(y - F\hat{\beta}), \sigma_1^2), \tag{1.3.33}
\]
where

\[ \hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} y, \]

and

\[ \sigma_1^2 = \sigma^2 \left\{ 1 - (f(x_0)^T, r_0^T) \left[ \begin{array}{cc} 0 & F^T \\ F & R \end{array} \right]^{-1} \left( f(x_0) \right) \right\}. \]

### 1.4 Designs for Computer Experiments

Considering the situation in which simulations take a long time to run and hence can only be conducted for limited runs, the experimental design for a computer experiment needs to be chosen carefully. The design strategies for computer experiments differ from those for traditional physical experiments in two ways: (i) Due to the deterministic nature of the output, designs should not take more than one observation at any input; (ii) Designs should provide information about the response surface across the entire input space.

Space-filling designs, such as *Latin hypercube designs* (LHDs), uniform designs, distance-based designs, etc., have been widely used in computer experiments. Among these designs, LHDs have the property that when an \(n\)-point LHD is projected onto each input factor, each factor will have \(n\) uniformly-spaced levels. This property is the sense in which an LHD is considered space-filling.

McKay, Beckman and Conover (1979) introduced LHDs as a desirable way to select values of input variables for the purpose of estimating the mean of a function numerically. Let \( A = (a_{ij}) \) denote an \(n \times d\) matrix in which each column is independently generated by a permutation of \(\{1, \cdots, n\}\). An ordinary LHD \( D = (d_{ij}) \) of \(n\) runs and \(d\) inputs is then defined as

\[ d_{ij} = (a_{ij} - u_{ij})/n, \text{ for } i = 1, \cdots, n, j = 1, \cdots, d. \]  \hspace{1cm} (1.4.1)
where the $u_{ij}$ are independent $U[0,1)$ random variables, $d_{ij}$ is value of the $j^{th}$ input variable on the $i^{th}$ run. Therefore, there is exactly one point falling within one of the $n$ equal-spaced intervals $(0, 1/n], (1/n, 2/n], \cdots, ((n-1)/n, 1]$.

However, the general LHD without any restrictions may perform poorly in the sense of being space-filling in $[0,1]^d$. For example, when the permutation of each column in $A$ is $(1, 2, \cdots, n)^T$, the design points will lie on a line in the $d$ dimensional space, which does not seem to be space-filling. Therefore, a “good” LHD is generated by optimizing a criterion that could represent a desirable property of the design. Iman and Conover (1982), Owen (1994), and Tang (1998) introduced a criterion to find a desirable LHD by minimizing the correlations among input variables. Morris and Mitchell (1995) proposed the LHD-generating criterion by maximizing the minimum distance between points. Other approaches to obtain good LHDs are given by Tang (1993), Park (1994), Ye (1998), Ye, Li and Sudjianto (2000), and Jin, Chen and Sudjianto (2005).

1.5 Sequential Designs for Model Predictions

In this section, we introduce several sequential designs that can be implemented to give “good” model predictions over the entire design space. In practice, it is always possible that the model fit from the initial design might be unsatisfactory. In order to obtain a better model fit, it is necessary to take additional observations; i.e., to use a sequential design. In this section, we will summarize the existing sequential designs for model fit in computer experiments.
1.5.1 Sequential MSPE Design

We consider the GaSP model with the constant mean. A design can be implemented sequentially by selecting a new input point, $x_0$, with the largest MSPE in (1.3.9) based on the existing input points. In particular,

$$
\text{MSPE}(x_0) = \max_x \text{MSPE}(x) 
$$

$$
= \max_x \left( \sigma^2 \left[ 1 - \hat{r}(x)^T \hat{R}^{-1} \hat{r}(x) + \frac{(1 - \hat{r}(x)^T \hat{R}^{-1} 1)^2}{1^T \hat{R}^{-1} 1} \right] \right) \quad (1.5.1)
$$

where $\hat{r}(x)$ is the estimated correlation vector between the previous inputs ($x_1, \cdots, x_n$) and the new candidate input $x$ and $\hat{R}$ is the estimated correlation matrix from all previous data.

1.5.2 Sequential IMSPE Design

The maximum MSPE sequential design tends to select points on the boundaries of the input space (Sacks, Welch, Mitchell and Wynn 1989), which might lead to a poor fitted surface. Sacks et al. (1989) considered the integrated mean squared prediction error (IMSPE) criterion to select a set of new inputs $\chi_c$, that achieve

$$
\min_{\chi_c} \int_{\chi} \frac{\text{MSPE}_{all}(x)}{\sigma^2} \omega(x) dx 
$$

(1.5.2)

where $\chi$ is the input space, $\omega(\cdot)$ is a non-negative function satisfying $\int_{\chi} \omega(x) dx = 1$, which is typically a uniform weighting. The unknown parameters in (1.5.2) are estimated based on the previous data, while the correlation matrix $\hat{R}$ and the correlation vector $\hat{r}(x)$ are based on both the previous data and the new candidate input points. Differing from the maximum MSPE approach, the MSPE$_{all}$ in (1.5.2) for each set
of candidate input points is based not only on the existing points but also on the
new candidate input points. Therefore, the IMSPE selects a new input set $\chi_c$ that
minimizes the average MSPE of the existing points and the new data points.

The sequential IMSPE approach is found to lead to new input points that cluster
around the existing points (Sacks et al. 1989). Therefore, a modified IMSPE criterion
is proposed by Lam and Notz (2008). The modified criterion is to select $x_0$ that

$$\min_{x_0 \in \chi} \text{IMSPE}(x_0)/\min_{x_i \in \chi_P} (d(x_i, x_0))$$

(1.5.3)

where $\chi_P$ denotes all the observed input points. The distance penalty tends to push
the subsequent points away from the observed input points and hence avoids the
clustering issue.

1.5.3 Sequential Cross Validation Prediction Error (XVPE)
Design

As a popular approach for estimating model parameters, cross validation prediction error can be used as an alternative measure of the prediction error.

For a candidate point $x_0$, let $\hat{Y}(-j)(x_0)$ denote the EBLUP of $Y(x_0)$ based on all
the existing data except $\{x_j, y(x_j)\}, j = 1, \cdots, n$, while $\hat{Y}(x_0)$ denotes the EBLUP
of $Y(x_0)$ using all the data.

The arithmetic mean cross validation criterion proposed by Jin, Chen and Sud-
jianto (2002) is given by

$$XVPE_A(x_0) = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (\hat{Y}(-j)(x_0) - \hat{Y}(x_0))^2 \times \min_j (d(x_i, x_0))}$$

(1.5.4)

Instead of penalizing the distance from the new candidate point and the existing
points, Lam and Notz (2008) propose three cross-validation-based criteria to prevent
selecting points from clumping together. The geometric mean cross validate criterion uses the product term to penalize the small prediction error components.

\[
XVPE_G(x_0) = \sqrt[n]{\prod_{j=1}^{n}(\hat{Y}^{(-j)}(x_0) - \hat{Y}(x_0))^2}
\]  

(1.5.5)

The harmonic mean cross validation criterion enlarges the impact of the small prediction errors and diminishes the impact of the large ones

\[
XVPE_H(x_0) = \frac{n}{\sum_{j=1}^{n} \frac{1}{Y^{(-j)}(x_0) - \hat{Y}(x_0)}^2}
\]

(1.5.6)

The maximin cross validation criterion is based on the minimum cross validation prediction error \(XVPE_M(x_0)\)

\[
XVPE_M(x_0) = \min_{j}(\hat{Y}^{(-j)}(x_0) - \hat{Y}(x_0))^2.
\]

(1.5.7)

For each of these cross validation criteria, one can pick the new point, \(x_0\), that has the largest “mean” prediction error, in the sense of the corresponding cross validation criterion. The cross validation can be viewed as a semi-parametric measure of the prediction error which is an alternative to the MSPE. Unlike the maximum MSPE and IMSPE designs, given \(\hat{Y}^{(-j)}(x)\) and \(\hat{Y}(x)\), the cross validation methods do not make explicit use of the correlation matrix. Therefore, the cross validation methods can be used as semi-parametric methods of sequential designs in the sense of improving prediction accuracy.
Chapter 2: Introduction to Computer Experiments with Quantitative and Qualitative Inputs

The standard computer experiments in Chapter 1 require that all the input variables are quantitative. In many applications, however, many computer experiments involve both quantitative and qualitative input variables. For example, Rawlinson, Furman, Li, Wright and Bartel (2006) investigated the design of a knee prosthesis involving qualitative factors such as “prosthesis design” and “force pattern”. Another example including both quantitative and qualitative inputs was the air characteristics model discussed in Qian and Wu (2008). In the study of computational fluid-dynamics programs (Schmidt, Cruz and Iyengar 2005) used in the air characteristics model, the thermal dynamics data can involve qualitative factor such as “air diffuser unit location”, “hot air return vent location”, and “power unit type”. The goal of this chapter is to review the predictive models and experimental designs for computer experiments having both quantitative and qualitative inputs.

Several models have been proposed for Gaussian-process-based predictors with quantitative and qualitative inputs by statisticians. Qian, Wu and Wu (2008) introduced a general framework for building Gaussian process models with quantitative and qualitative inputs. Han, Santner, Notz and Bartel (2009) proposed a
Bayesian Gaussian process model, called the hierarchical quantitative-qualitative variable model (HQQV), using Markov chain Monte Carlo (MCMC) for computation. Zhou, Qian and Zhou (2011) considered the flexibility of the unrestricted correlation structure of qualitative factors in Qian et al. (2008), and proposed a hyperspherical parameterization for the cross-correlation parameters, which can not only simplify the computations, but also ensure the positive definite matrix with unit diagonal elements (abbreviated to PDUDE) property for the cross-correlation parameters. Han, Notz, Santner and Long (2013) further extended the HQQV model and described an ANOVA kriging method which can select “similar” qualitative levels to estimate the output at the target level when there exist distinctions among different levels.

Several designs related to the computer experiments involving quantitative and qualitative inputs are proposed. Qian, Tang and Wu (2009) introduced nested space-filling designs (NSFD) for a special qualitative variable – accuracy. They considered an experiment with two levels of accuracy (low accuracy and high accuracy), which can be viewed as a qualitative input with two levels. For more than 2 levels of accuracy (including 2 levels), Qian (2009) generalized the NSFD into nested Latin hypercube designs (NLHD) which can be used for conducting computer experiments with multiple levels of accuracy. Qian (2012) proposed sliced Latin hypercube designs (SLHD), which can be partitioned into slices of smaller Latin hypercube designs for general problems. Furthermore, Huang, Lin, Liu and Yang (2014) further developed SLHD by proposing clustered slice Latin hypercube designs (CSLHD), which are a type of SLHD with points clustered in the design region.
2.1 Gaussian Process Models With Quantitative and Qualitative inputs

2.1.1 Cross-Correlation Structure Model

The framework for building GaSP models with quantitative and qualitative inputs was first proposed by Qian et al. (2008). Suppose we have $K$ qualitative inputs, and the $j$th ($j = 1, \cdots, K$) qualitative input has $m_j$ levels. Let $w = (x^T, z^T)^T$, where $x = (x_1, \cdots, x_d)^T \in [0, 1]^d$ (if necessary, after scaling $x$ into $[0,1]^d$) denotes the quantitative inputs with $d$ components and $z = (z_1, \cdots, z_K)^T$ denotes the qualitative inputs, where $z_j$ is the level of the $j$th qualitative input, for $j = 1, \cdots, K$. We also let $y(w)$ be the corresponding response values. The GaSP model used by Qian et al. (2008) views $y(\cdot)$ as a realization from the stochastic process

$$Y(w) = f^T(w)\beta + \epsilon(w), \quad (2.1.1)$$

where $f(w) = (f_1(w), \cdots, f_q(w))^T$ is a $q \times 1$ pre-specified vector of regression functions with the elements being real-valued functions of inputs $w$, and $\beta = (\beta_1, \cdots, \beta_q)^T$ is a $q \times 1$ vector of unknown regression parameters. The process $\epsilon(\cdot)$ is a stationary Gaussian process with mean 0 and covariance between two responses $\epsilon(w_1)$ and $\epsilon(w_2)$

$$\text{Cov}(\epsilon(w_1), \epsilon(w_2)) = \sigma^2 \epsilon R(w_1, w_2), \quad (2.1.2)$$

where $\sigma^2 \epsilon$ denotes the process variance and $R(w_1, w_2)$ denotes the correlation function of $\epsilon(\cdot)$.

The correlation function $R(\cdot)$ is defined as follows:

$$R(w_1, w_2) = \prod_{j=1}^{K} \left[ \tau_{j, z_{j1}, z_{j2}} K \phi_j(x_1 - x_2) \right], \quad (2.1.3)$$
where $T_j = (\tau_{j,r,s})$ is an $m_j \times m_j$ PDUDE (positive definite matrix with unit diagonal entries), and $K_{\phi_j}(x_1 - x_2)$ is any valid correlation function for the quantitative inputs $x_1$ and $x_2$. As long as the PDUDE property holds for each $T_j, j = 1, \ldots, K$, $R(w_1, w_2)$ is a valid correlation function.

In particular, if $K_{\phi_j}(x_1 - x_2)$ has the form of the power exponential correlation function $\exp\{-\sum_{i=1}^d \phi_{ij}|x_{i1} - x_{i2}|^p\}$, then (2.1.3) becomes

$$R(w_1, w_2) = \prod_{j=1}^K \tau_{j,z_1,j_1} \tau_{j,z_2,j_2} \exp\{-\sum_{i=1}^d \phi_{i}|x_{i1} - x_{i2}|^p\}$$

(2.1.4)

The parameter $\tau_{j,z_1,j_2}$ measures the correlation between responses at any two input values that differ only on the qualitative factor $z_j$, and is also called cross-correlation parameter. The scale correlation parameters $\phi_1, \ldots, \phi_d$ measure the roughness of the response surface of the GP with the same levels of qualitative inputs. It should be noted that different qualitative levels share the same roughness parameter $\phi$. Similar to Chapter 1, when $p = 2$, the correlation function is called the product Gaussian correlation function. For the rest of this thesis, we will assume the product Gaussian correlation function, i.e. $p = 2$.

Suppose the data consist of $n$ different input values $W = (w_1, \ldots, w_n)^T$, and the corresponding responses, $y = (y(w_1), \ldots, y(w_n))^T$. Therefore, given the correlation parameters $\phi = (\phi_1, \ldots, \phi_d)^T$ and $T = \{T_1, \ldots, T_K\}$, the correlation matrix $R$ is determined by (2.1.4), and the $(i,j)^{th}$ entry is $R(w_i, w_j)$. Therefore, up to an additive constant, the log-likelihood of $y$ is

$$-\frac{1}{2} \left[ n \log \sigma^2 + \log |R| + \frac{(y - F\beta)^T R^{-1} (y - F\beta)}{\sigma^2} \right]$$

(2.1.5)

where $F = (f(w_1), \ldots, f(w_n))^T$ is an $n \times q$ matrix defined by the same pre-specified function vector $f(w)$ as in (2.1.1).
The estimates of the parameters are obtained through MLE. The estimates can be found by iterating between a **regression fitting** and a **correlation fitting**:

**Regression fitting:** Given \( \hat{\phi} \) and \( \hat{T} \), generate \( \hat{\beta} \) and \( \hat{\sigma}^2 \) by

\[
\hat{\beta} = (F^T[R(\hat{\phi}, \hat{T})]^{-1}F)^{-1}F^T[R(\hat{\phi}, \hat{T})]^{-1}y
\]

\[
\hat{\sigma}^2 = \frac{1}{n} (y - F\hat{\beta})^T R^{-1} (y - F\hat{\beta})
\]

**Correlation fitting:** Given \( \hat{\beta} \) and \( \hat{\sigma}^2 \), let \( u_i = (y_i - f^T(w_i)\hat{\beta})/\hat{\sigma} \), for \( i = 1, \cdots, n \). Then, fit the standardized GaSP model with the correlation matrix \( R \) and response vector \( u = (u_1, \cdots, u_n)^T \). The correlation fitting can be done by iterating between the following \( \phi \)-step and \( T \)-step:

**\( \phi \)-step** Given \( \hat{T} \),

\[
\hat{\phi} = \arg\min_{\phi} \left[ \text{tr}(uu^T R^{-1}) + \log |R| \right]
\]

subject to \( \phi_i \geq 0, i = 1, \cdots, d \).

**\( T \)-step** Given \( \hat{\phi} \),

\[
\hat{T} = \arg\min_{T} \left[ \text{tr}(uu^T R^{-1}) + \log |R| \right]
\]

subject to \( T_j \succ 0 \) (positive definite), and \( \text{diag}(T_j) = 1, j = 1, \cdots, K \).

Similar to the GaSP model in the general quantitative case, the fitted GaSP model can be used in predicting the response value, and the empirical BLUP of \( y \) at the point \( w_0 \) is given by

\[
\hat{y}(w_0) = f^T(w_0)\hat{\beta} + \hat{r}_0^T \hat{R}^{-1} (y - F\hat{\beta}) \tag{2.1.6}
\]

where \( \hat{\beta} \) is the estimate of \( \beta \), \( \hat{R} \) is the estimated correlation matrix of \( y \) and

\[
\hat{r}_0 = (\hat{R}(y(w_0), y(w_1)), \cdots, \hat{R}(y(w_0), y(w_n)))^T.
\]
The corresponding empirical predictive variance can be derived from the MSPE

\[ \hat{\sigma}_e^2 [1 - \hat{\mathbf{r}}_0^T \hat{\mathbf{R}}^{-1} \hat{\mathbf{r}}_0 + (\mathbf{f}(\mathbf{x}_0) - \hat{\mathbf{r}}_0^T \hat{\mathbf{R}}^{-1} \mathbf{F})(\mathbf{F}^T \hat{\mathbf{R}}^{-1} \mathbf{F})^{-1}(\mathbf{f}(\mathbf{x}_0) - \hat{\mathbf{r}}_0^T \hat{\mathbf{R}}^{-1} \mathbf{F})] \quad (2.1.7) \]

In practice, the cross-correlation matrix \( \mathbf{T} \) can be restricted to some specific cases to simplify computations:

**Exchangeable correlation:** Stein (1999) defines the exchangeable correlation function by assuming the same correlations for all the different pairs of \( z_{ji_1} \) and \( z_{ji_2}, j = 1, \cdots, K, i_1 \neq i_2, i_1, i_2 \in 1, \cdots, m_j \). Joseph and Delaney (2007) apply the exchangeable correlation structure in some physical experiments. The exchangeable correlation structure will simplify (2.1.4) into

\[ R(\mathbf{w}_1 - \mathbf{w}_2) = \exp \left\{ -\sum_{i=1}^d \phi_i |x_{i1} - x_{i2}|^p - \sum_{j=1}^K \theta_j I\{z_{j1} \neq z_{j2}\} \right\}, \quad (2.1.8) \]

where \( 0 < p \leq 2, \phi_i > 0 \) for \( i = 1, \cdots, d, \theta_j > 0 \) for \( j = 1, \cdots, K, \) and \( I\{z_{j1} \neq z_{j2}\} \) is the indicator function for the set of the qualitative levels \( \{z_{j1} \neq z_{j2}\} \).

**Multiplicative correlation:** When \( m_j \geq 4 \), McMillian, Sacks, Welch and Gao (1999) propose the multiplicative correlation to model the correlations of unordered categorical variables. The multiplicative correlation function in (2.1.4) becomes

\[ R(\mathbf{w}_1 - \mathbf{w}_2) = \exp \left\{ -\sum_{i=1}^d \phi_i |x_{i1} - x_{i2}|^p - \sum_{j=1}^K (\theta_{j1} + \theta_{j2}) I\{z_{j1} \neq z_{j2}\} \right\} \quad (2.1.9) \]

where \( 0 < p \leq 2, \phi_i > 0 \) for \( i = 1, \cdots, d \) and \( \theta_{jk} > 0 \) for \( k = 1, \cdots, m_j, j = 1, \cdots, K \). It can be noted that the number of cross-correlation parameters in (2.1.9) has been reduced from \( \sum_{j=1}^K m_j(m_j - 1)/2 \) in the general case to \( \sum_{j=1}^K m_j \).
2.1.2 Hypersphere Parameterization Model

The flexibility of the unrestrictive correlation structure for qualitative factors used in Qian et al. (2008) also brings computational complexity. Zhou et al. (2011) introduced a new parameterization using the hypersphere decomposition which not only inherited the flexibility of the correlation structure, but also essentially reduced the computational complexity.

The settings are the same as Section 2.1.1. To make sure (2.1.4) is a valid correlation function, the $m_j \times m_j$ matrix $T_j = (\tau_{j,r,s})$ must be a PDUDE (Qian et al. 2008). Thus, we can apply a Cholesky decomposition to $T_j$, which is given by

$$T_j = L_jL_j^T,$$  \hspace{1cm} (2.1.10)

where $L_j = (l_{r,s})$ is a lower triangular matrix with strictly positive diagonal entries.

Next, we can model the nonzero part of each row vector of $L_j$, $(l_{j,r,1}, \ldots, l_{j,r,r})$, as a point on the $r$-dimensional unit hypersphere described as follows. For $r = 1$, let $l_{j,1,1} = 1$ and for $r = 2, \ldots, m_j$, the spherical coordinate system is

$$\begin{align*}
  l_{r,1} &= \cos(\theta_{r,1}), \\
  l_{r,s} &= \sin(\theta_{r,1}) \cdots \sin(\theta_{r,s-1}) \cos(\theta_{r,s}), \quad \text{for} \quad s = 2, \ldots, r - 1, \\
  l_{r,r} &= \sin(\theta_{r,1}) \cdots \sin(\theta_{r,r-1}) \sin(\theta_{r,r-1}),
\end{align*}$$

where $\theta_{r,s} \in (0, \pi)$. The structure of the hypersphere coordinate implies that $T_j$ must have unit diagonal elements. Thus, the matrix $T_j$ generated by these $\theta_{r,s}$'s is always a PDUDE.

After obtaining the iterated MLEs of the parameters through the new hyperspherical parameterization, the BLUP of $y$ at any point can be obtained from the same formula as in (2.1.6). The hyperspherical parameterization has several appealing advantages. One is that it converts the complicated PDUDE constraint on $T_j$
into simple box constraints $\theta_{j,r,s} \in (0, \pi)$. The other is that the range of $\theta_{j,r,s}$ assures the flexible choice of the entries in $T_j$ and thus can detect different types of cross-correlations. Furthermore, the one-to-one transformation between the PDUDE matrix and the hypersphere parameter set implies the one-to-one correspondence between estimating the PDUDE matrix $T_j$ and the parameter set $\{\theta_{j,r,s}\}$. Thus, the hyperspherical parameterization of the cross-correlation parameters provide a fast and efficient way to fit models in computer experiments with both quantitative and qualitative input variables.

2.1.3 Hierarchical Quantitative-Qualitative Variable Model

Han et al. (2009) proposed a new Bayesian Hierarchical Quantitative-Qualitative Variable (HQQV) Model based on the assumption that the outputs corresponding to different levels of a qualitative input are drawn from Gaussian stochastic processes with “similar” correlation structures and magnitudes of variations. Instead of the cross-correlation parameter $\tau_{j,r,s}$, the similarities between different levels are captured by the prior distributions on the parameters. Han et al. (2009) first introduced the HQQV model with an arbitrary number of quantitative inputs and one qualitative input. They then generalized the HQQV model to handle multiple qualitative inputs. To compare with the other methods, we will introduce the general case directly.

Suppose we have $K$ qualitative inputs and the $j^{th}$ input has $m_j$ levels for all $j \in \{1, \cdots, K\}$. We use similar notation as before but use superscripts $(j)$ to distinguish the $j^{th}$ qualitative input from the others. We assume the qualitative inputs are $z = (z^{(1)}, \cdots, z^{(K)})^T$, where $z^{(j)} \in \{1, \cdots, m_j\}$ is the input for $j^{th}$ qualitative input
of expressing priors, we use \( \rho \) (parameters. Thus, we assume the following parameters:

\[ Y \sim \text{independent Gaussian processes having 0 mean and variances} \] 

\[ \epsilon = \begin{pmatrix} \sigma_1, \ldots, \sigma_m \end{pmatrix} \] 

\[ \text{and} \quad \rho = \begin{pmatrix} \rho_{11}, \ldots, \rho_{mm} \end{pmatrix} \] 

for \( j = 1, \ldots, K \).

Given the parameters \((\beta^{(j)}, \sigma^{2(j)}, \rho^{(j)})\), \( j = 1, \ldots, K \), the model is

\[ Y(x, z) | (\beta, \sigma^2, \rho) \sim \sum_{j=1}^{K} Y_j(z^{(j)}, x) | (\beta^{(j)}, \sigma^{2(j)}, \rho^{(j)}) \] (2.1.12)

where \( Y_1(\cdot, \cdot), \ldots, Y_K(\cdot, \cdot) \) are \( K \) independent Gaussian processes. Individually, we have

\[ Y_j(x, z^{(j)}) | (\beta^{(j)}, \sigma^{2(j)}, \rho^{(j)}) \sim \beta^{(j)}_{z(j)} + \epsilon_{j,z}(x) \] (2.1.13)

with \( \beta^{(j)}_{z(j)} \) being an unknown constant, \( \epsilon_{j,1}(\cdot), \ldots, \epsilon_{j,m_j}(\cdot) \) being \( m_j \) independent Gaussian processes having 0 mean and variances \( \sigma_1^{2(j)}, \ldots, \sigma_{m_j}^{2(j)} \). The covariance between \( \epsilon_{j,z}(x_a) \) and \( \epsilon_{j,z}(x_b) \) is given by \( \text{Cov}(\epsilon_{j,z}(x_a), \epsilon_{j,z}(x_b)) = \sigma_{z}^{2(j)} R(x_a - x_b | \rho^{(k)}_z) \), and \( R(\cdot | \rho^{(j)}_z) \) is the product Gaussian correlation function. Thus, the \( Y(x, z) \) is a GaSP with mean \( \sum_{j=1}^{K} \beta^{(j)}_{z(j)} \) and covariance between \( Y(x_a, z_a) \) and \( Y(x_b, z_b) \)

\[ C(w_a, w_b) = \text{Cov}(Y(x_a, z_a), Y(x_b, z_b)) = \sum_{j=1}^{K} \sigma_{z}^{2(j)} R(x_a - x_b | \rho_{z(a)}^{(j)}) I(z_a^{(j)} - z_b^{(j)}) \] (2.1.14)

where \( I(z_a^{(j)} - z_b^{(j)}) \) is the indicator of \( z_a^{(j)} - z_b^{(j)} = 0 \). We assume the priors on parameters are as follows, for each \( j = 1, \ldots, K \) and each \( z = 1, \ldots, m_j \):

\[ \beta^{(j)}_{z} \propto 1; \]
\[ \sigma^2_z (j) \sim \text{Inverse Gamma}(5, 0.2 \times K) \]

\[ \rho_{zi}^{(j)} \sim \text{Beta}(\alpha_i^{(j)}, \gamma_i^{(j)}), \text{ for } i = 1, \ldots, d \]

Our goal is to find a predictor \( \hat{y}^{HQV}(x_0, z_0) \) of an unknown response given the response vector \( y_n = (y(x_1, z_1), \ldots, y(x_n, z_n)) \). The model first simulates from the posterior distribution of the model parameters using the Metropolis-Hastings algorithm, and then uses the draws from the simulations to approximate the minimum MSPE predictor \( \hat{y}^{HQV}(x_0, z_0) = E(Y(x_0, z_0)|y_n) \). Specifically, for all \( j \in \{1, \ldots, K\} \), \( z \in \{1, \ldots, m_j\} \) and \( i \in \{1, \ldots, d\} \), the initial values of \( \beta_z^{(j)}, \sigma^2_z \) and \( \rho_{zi}^{(j)} \) are taken to be 0, 1/\( K \), and the median \( \{0.005, \max_{1 \leq z \leq m_j} \{\rho_{zi}^{(j)}\}, 0.995\} \). Given the parameters \( (\beta, \sigma^2, \rho) \) and the training data \( y \), the conditional mean and the conditional variance of \( Y(x_0, z_0) \) can be derived as

\[
E(Y(x_0, z_0)|y, \beta, \sigma^2, \rho) = \sum_{j=1}^{K} \beta_z^{(j)} + \Sigma_{0n} \Sigma_{nn}^{-1} \left( y - 1 \sum_{j=1}^{K} \beta_z^{(j)} \right),
\]

and

\[
\text{Var}(Y(x_0, z_0)|y, \beta, \sigma^2, \rho) = \Sigma_{00} - \Sigma_{0n} \Sigma_{nn}^{-1} \Sigma_{0n},
\]

where \( \Sigma_{00} = \sigma^2_z \) is the variance of the qualitative combination level \( z_0 \), \( \Sigma_{nn} \) is the \( n \times n \) covariance matrix of \( y \) defined by (2.1.14), where the \( (i, j)^{th} \) entry is \( C(w_i, w_j) \), and \( \Sigma_{0n} \) is an \( n \times 1 \) covariance vector between the training data \( y \) and the new response \( Y(x_0, z_0) \), where the \( i^{th} \) entry is \( C(w_i, w_0) \).

Therefore, the predictor \( \hat{y}^{HQV}(x_0, z_0) \) and a measure of the predictive uncertainty can be derived from

\[
E(Y(x_0, z_0)|y_n) = E(E(Y(x_0, z_0)|y, \beta, \sigma^2, \rho)),
\]

29
and
\[
\text{Var}(Y(x_0, z_0)|y_n) = \text{Var}(E(Y(x_0, z_0)|y_n, \beta, \sigma^2, \rho)) + E(\text{Var}(Y(x_0, z_0)|y_n, \beta, \sigma^2, \rho)).
\]

For each draw of \((\beta, \sigma^2, \rho)\), \(\hat{y}(w_0)\) and \(\hat{\sigma}^2(\hat{y}(w_0))\) are the numerical result of \(E(Y(x_0, z_0)|y_n, \beta, \sigma^2, \rho)\) and \(\text{Var}(Y(x_0, z_0)|y_n, \beta, \sigma^2, \rho)\) as in (2.1.15) and (2.1.16), respectively. The strong law of large numbers guarantees that the combinations of the sample means and the sample variance of the sampled sequences \(E(Y(x_0, z_0)|y_n, \beta, \sigma^2, \rho)\) and \(\text{Var}(Y(x_0, z_0)|y_n, \beta, \sigma^2, \rho)\) converge to \(E(Y(x_0, z_0)|y_n)\) and \(\text{Var}(Y(x_0, z_0)|y_n)\) almost everywhere as in (2.1.17) and (2.1.18) as the number of simulations tends to infinity.

2.1.4 ANOVA Kriging Model

The GaSP models considered so far depend on the assumption that the outputs corresponding to different levels of the qualitative inputs have “similar” structures in the sense of correlation functions. Han et al. (2013) improved the previous HQQV models and introduced the ANOVA kriging model. The advantage of the ANOVA kriging model is that prediction at one level only uses data from those levels with similar structure.

Let \(\hat{y}^{HQQV}(x_0, z_0)\) denote the HQQV predictor in Han et al. (2009) and use the same notation as Section 2.1.3. The HQQV ANOVA kriging (HAK) predictor is generated by using the HQQV predictor.

We first consider the simple case that there is one qualitative input variable with \(m\) levels. Let \(z \in \{1, \ldots, m\}\) denote the qualitative levels, and we are interested in predicting the output variable at level \(z = z_0\). The training data associated with
level $z_0$ are denoted by $x_{1}^{(z_0)}, \ldots, x_{n_0}^{(z_0)}$. The idea of the HAK predictor is to only use the most “similar” set of levels to predict the unknown level $z_0$. The construction of HAK predictor includes two steps:

Step 1 is the validation step. Denote the HQQV predictor for the training data at level $z_0$ by $(\hat{y}^{HQV}(x_{1}^{(z_0)}, z_0), \ldots, \hat{y}^{HQV}(x_{n_0}^{(z_0)}, z_0))$. For all $z \neq z_0$, augment the training data at level $z$ by predicting $(x_{1}^{(z_0)}, z), \ldots, (x_{n_0}^{(z_0)}, z)$ and merging these predictions $(\hat{y}^{HQV}(x_{1}^{(z_0)}, z), \ldots, \hat{y}^{HQV}(x_{n_0}^{(z_0)}, z))$ with the training data set. We can construct all $2^{m-1}$ subsets of the $m - 1$ levels $1, \ldots, z_0 - 1, z_0 + 1, \ldots, m$.

Let $\hat{y}^{HQV}(x_{i}^{(z_0)}, z_0)$ be the HQQV predictor based on data $\{y(x_1^{(z_0)}, z_0), \ldots, y(x_0^{(z_0)}, z_0), y(x_{i+1}^{(z_0)}, z_0), \ldots, y(x_{n_0}^{(z_0)}, z_0)\}$. Let $\hat{y}^{HQV}(x_{i}; z)$ be the HQQV predictor based on the data $\{y(x_1^{(z_0)}, z), \ldots, y(x_{i-1}^{(z_0)}, z), y(x_{i+1}^{(z_0)}, z), \ldots, y(x_{n_0}^{(z_0)}, z)\}$ for $z = z_1^c, \ldots, z_{c}^{c}$ and the same posterior draws of the parameters used by $\hat{y}^{HQV}(x_{i}^{(z_0)}, z_0)$.

For any given subset of the $m - 1$ levels $1, \ldots, z_0 - 1, z_0 + 1, \ldots, m$, denoted by $z_1^c, \ldots, z_{c}^{c}$, we compute the validation term as follows

$$
\sum_{i=1}^{n_0} \left\{ \frac{2}{n_{e+1}} B_{-i}(x_{i}^{(z_0)}, z_0) D_{-i}(x_{i}^{(z_0)}, z_0) - \frac{1}{(n_{e+1})^2} D_{-i}^2(x_{i}^{(z_0)}, z_0) \right\} = \sum_{i=1}^{n_0} B_{-i}(x_{i}^{(z_0)}, z_0)^2 - \sum_{i=1}^{n_0} \left( \frac{1}{n_{e+1}} D_{-i}(x_{i}^{(z_0)}, z_0) - B_{-i}(x_{i}^{(z_0)}, z_0) \right)^2, \quad (2.1.19)
$$

where

$$
B_{-i}(x_{i}^{(z_0)}, z_0) = \hat{y}^{HQV}(x_{i}^{(z_0)}, z_0) - y(x_{i}^{(z_0)}, z_0),
$$

and

$$
D_{-i}(x_{i}^{(z_0)}, z_0) = \sum_{z \in \{z_1^c, \ldots, z_{c}^{c}\}} \left[ \hat{y}^{HQV}(x_{i}, z) - y(x_{i}^{(z_0)}, z) \right].
$$

Intuitively, $B_{-i}(x_{i}^{(z_0)}, z_0)$ measures the cross-validation bias at level $z_0$, while $D_{-i}(x_{i}^{(z_0)}, z_0)$ measures the total cross-validation bias at the candidate subset levels other than $z_0$. 31
When the response surfaces of the candidate levels are similar to the target response surface at level \( z_0 \), \( D_i(x_i^{(z_0)}, z_0) \) should be approximately equal to \( P_c \ast B_i(x_i^{(z_0)}, z_0) \). The maximum of (2.1.19) implies that the response surfaces of the \( P_c \) candidate subset levels have the most similar validation bias as the surface of the unknown level \( z_0 \). Thus, after we calculate each possible subset, we can find the set of levels with the largest value of (2.1.19). We denote the corresponding set of levels by \( z_1^*, \ldots, z_p^* \).

Step 2 is the prediction step. We define \( \hat{A}^*(x) \) as

\[
\hat{A}^*(x) = \frac{1}{1 + P} \left[ \hat{y}^{HQV}(x, z_0) + \sum_{z \in \{z_1^*, \ldots, z_P^*\}} \hat{y}^{HQV}(x, z) \right],
\]

which is an overall average predictor for all the levels in \( \{z_1^*, \ldots, z_P^*\} \). Notice that \( \hat{y}^{HQV}(x, z_0) \) and \( \hat{y}^{HQV}(x, z) \) are the HQQV predictors using the training data at all levels. Thus, we can obtain \( \{\hat{A}^*(x_1^{(z_0)}), \ldots, \hat{A}^*(x_{n_{z_0}}^{(z_0)}), \hat{A}^*(x_0)\} \). By subtracting these \( \{\hat{A}^*(x_1^{(z_0)}), \ldots, \hat{A}^*(x_{n_{z_0}}^{(z_0)})\} \) values from the training data, we obtain the deviation data set \( y^{(z_0)} \) as

\[
y^*(x_i^{(z_0)}, z_0) = y(x_i^{(z_0)}, z_0) - \hat{A}^*(x_i^{(z_0)}),
\]

for all \( i = 1, \ldots, n_{z_0} \). Correspondingly, the HQV predictor based on the deviation data set \( y^{(z_0)} \) can be obtained, denoted by \( \hat{y}^{*HQV}(x_0, z_0) \). Then, the HAK predictor \( \hat{y}^{HAK}(x_0, z_0) \) is the sum of \( \hat{A}^*(x_0) \) and \( \hat{y}^{*HQV}(x_0, z_0) \),

\[
\hat{y}^{HAK}(x_0, z_0) = \hat{y}^{*HQV}(x_0, z_0) + \hat{A}^*(x_0).
\]

We can extend the HAK predictor to the case when there are \( K \) qualitative input factors. The idea is similar to the general ANOVA model. We need to first determine which main effects/interaction effects are present by using expert knowledge.
or by conducting a sensitivity analysis. Then, the HAK predictor is the sum of the estimated effects and the prediction of the corresponding deviation.

Suppose \( J \) main / interaction effects are assumed to be present. For the \( j \)th present qualitative factor \((j = 1, \cdots, J^*)\), the optimized set is denoted by \( z_{j,1}, \cdots, z_{j,P_j} \), and the corresponding optimized set of the remaining levels can be derived from (2.1.19), denoted by \( z_{j,1}^C, \cdots, z_{j,P_j}^C \). Generally, assume the \( j \)th main (or interaction) effect is corresponding to the first \( J^* \) qualitative inputs, we use \( z_j = z_j^{(0)} = (z_{j,1}^{(0)}, \cdots, z_{j,J^*}^{(0)}) \) to denote the \( j \)th main (or interaction) effect with the level \( z_j^{(0)} \). In addition, we denote all the corresponding optimized set of the remaining levels by \( z_{j,1}^C, \cdots, z_{j,P_j}^C \).

\( \hat{A}_{z_j = z_j^{(0)}}(x) = \frac{1}{P_j} \left[ \sum_{z_j^C \in \{z_{j,1}^C, \cdots, z_{j,P_j}^C\}} \hat{y}^{HQV}(x, z_j^{(0)}, z_j^C) \right] \).

For example, suppose we have \( J = 3 \) qualitative inputs \((z_1, z_2, z_3)\), and each input’s optimized set has two levels. Then, the main effect of \((z_1 = 1)\) is given by

\[
\hat{A}_{z_1 = 1}(x) = \frac{1}{4} \left[ \hat{y}^{HQV}(x, 1, 1, 1) + \hat{y}^{HQV}(x, 1, 1, 2) + \hat{y}^{HQV}(x, 1, 2, 1) + \hat{y}^{HQV}(x, 1, 2, 2) \right].
\]

The interaction effect of \((z_1 = 1, z_2 = 2)\) is given by

\[
\hat{A}_{z_1 = 1, z_2 = 2}(x) = \frac{1}{2} \left[ \hat{y}^{HQV}(x, 1, 2, 1) + \hat{y}^{HQV}(x, 1, 2, 2) \right].
\]

Furthermore, we define the overall average effect \( \hat{A}(x) \) as

\[
\hat{A}(x) = \frac{1}{\prod_{j=1}^J P_j} \left[ \sum_{j=1}^J \sum_{z_j \in \{z_{j,1}, \cdots, z_{j,P_j}\}} \sum_{z_j^C \in \{z_{j,1}^C, \cdots, z_{j,P_j}^C\}} \hat{y}^{HQV}(x, z_j, z_j^C) \right]
\]

(2.1.22)
In addition, we define \( \hat{y}^\text{HQV}_{(J+1)}(x_0, z_0, 1, \ldots, z_0, K) \) as the HQQV prediction of the deviation data,

\[
y^*_{(J+1)}(x_0, z_0, 1, \ldots, z_0, K) = y(x_0, z_0, 1, \ldots, z_0, K) - \left( \hat{A}^*(x_0) + \sum_{j=1}^{J} \hat{A}^*_C(x_0) \right).
\]

Then, the prediction of \( y(x_0, z(1) = z_0, 1, \ldots, z(K) = z_0, K) \) is

\[
y^{HA}_{(1)}(x_0, z(1) = z_0, 1, \ldots, z(K) = z_0, K) = \hat{A}^*(x_0) + \sum_{j=1}^{J} \hat{A}^*_C(x_0) + \hat{y}^\text{HQV}_{(J+1)}(x_0, z_0, 1, \ldots, z_0, K).
\]

(2.1.23)

2.2 Designs for Computer Experiments with Both Quantitative and Qualitative Inputs

Experimental designs for computer experiments only involving quantitative inputs have been well explored in the literature. However, the literature on designs for computer experiments with both quantitative and qualitative inputs is limited. Here we review this literature.

2.2.1 Nested Space-Filling Designs

Computer experiments with different levels of accuracy occur because in some cases complex computer programs can be run at different levels of sophistication with computational time increasing substantially as level of accuracy increases. The different levels of accuracy can be viewed as a qualitative input. For this situation, Qian et al. (2009) construct nested space-filling designs based on Galois fields and orthogonal arrays.

Qian et al. (2009) consider the situation where there are 2 levels of accuracy: a low-accuracy experiment (LE) and a high-accuracy experiment (HE), where HE is more accurate but more expensive than LE. The set of design points for LE and HE are
denoted by $D_t$ and $D_h$, respectively. Due to budget limitations, the number of points in $D_h$ must be smaller than the number of points in $D_t$. Qian et al. (2009) assume that the HE should be run only at points at which the LE has been run. Therefore, $D_h$ is nested within $D_t$, i.e., $D_h \subset D_t$. We denote a symmetric orthogonal array (OA) of size $n$, $m$ constraints, $s$ levels, and strength $t \leq 2$ by $OA(n, m, s, t)$, which is an $n \times m$ matrix with entries from a set of $s$ levels, such that for every $n \times t$ submatrix, the $s^t$ level combinations occur equally often. For every prime $p$ and every integer $u \geq 1$, there exists a Galois field $GF(p^u)$ of order $p^u$. The additive group $GF(p^u)$ is cyclic. The multiplicative group $GF(p^u)/\{0\}$ is cyclic, which allows easy calculations under multiplication. Furthermore, let $s_1 = p^{u_1}$ and $s_2 = p^{u_2}$ be powers of the same prime $p$ with integer $u_1 > u_2 \geq 1$. We assume $p_1(x)$ is the irreducible polynomial associated with $GF(s_1)$, and $p_2(x)$ is the irreducible polynomial associated with $GF(s_2)$. In addition, $\varphi$ is used to denote the modulus projection from $GF(s_1)$ to $GF(s_2)$, such that for any $f(x) \in GF(s_1)$,

$$\varphi(f(x)) = f(x) \mod p_2(x). \quad (2.2.1)$$

Suppose $A_1$ is an $OA(n_1, m_1, s_1, t_1)$, and $A_2$ is a subarray of $A_1$ with size $n_2$. For the array $A_1$ with entries from $GF(s_1)$, let $\varphi(A_1)$ denote the array obtained from $A_1$ after the levels of its entries are collapsed according to $\varphi$. We further assume that $\varphi$ collapses the $s_1$ levels of $A_1$ into $s_2$ levels. In addition, we suppose that $A_2$ becomes an $OA(n_2, m_2, s_2, t_2)$ after collapsing according to $\varphi$. Under these conditions, $(A_1, A_2)$ is said to be a nested orthogonal array (NOA), denoted by $NOA(A_1, A_2, \varphi)$. Note that $A_1$ contains $\varphi(A_2)$.

Based on the Rao-Hamming construction, Qian et al. (2009) derive a family of $NOA(A_1, A_2, \varphi)$, such that
1. \( A_1 \) is an OA\((n_1, m_2, s_1, 2)\), where \( n_1 = s_1^k \), \( m_2 = (s_2^k - 1)/(s_2 - 1) \) and \( k \geq 2 \) is an integer;

2. Provided that \( 2u_2 \leq u_1 + 1 \), \( A_2 \) is a subarray of \( A_1 \), and \( \varphi(A_2) \) is an OA\((n_2, m_2, s_2, 2)\) with \( n_2 = s_2^k \).

Therefore, one can use the array \( A_1 \) to generate an OA-based Latin hypercube design \( D_l \) and denote the subset of \( D_l \) corresponding to \( A_2 \) by \( D_h \). Then, \( D_l \) and \( D_h \) provide two space-filling designs with \( D_h \) nested within \( D_l \), where both \( D_l \) and \( D_h \) achieve stratification in two dimensions. Notice the use of nested designs make sense if it is important to observe the high-accuracy experiment only at inputs at which we observe the low-accuracy experiment.

Figure 2.1 displays a two-dimensional example of a NSFD. The low accuracy design has 64 input points, while the high accuracy design has 32 input points nested in the low accuracy design. Both of these designs are two-dimensional space-filling designs.

### 2.2.2 Nested Latin Hypercube Designs

The NSFD in Qian et al. (2009) only exists for certain parameter values according to the Rao-Hamming construction. Although the pair of nested designs achieve two-dimensional stratification, the smaller design cannot achieve stratification when the input points are projected onto any one-dimensional subspace (See Figure 2.1). Qian (2009) extended the applications and properties of NSFD and proposed a new type of nested design, called a *nested Latin hypercube design* (NLHD).

A two-layer NLHD is to define a special Latin hypercube design that contains a smaller Latin hypercube design as a subset, where the whole set is the first layer,
and the smaller Latin hypercube design is the second layer. Suppose $m, t \geq 1$ are integers and $n = tm$. A nested permutation denoted by $\pi$ is a special permutation on $\mathbb{Z}_n = \{1, \cdots, n\}$, such that the set generated by the first $m$ elements, $\{\lceil \pi(1)/t \rceil, \cdots, \lceil \pi(m)/t \rceil\}$ is a permutation on $\mathbb{Z}_m$. Furthermore, we denote the first $m$ elements of $\pi$ by $\tau = \{\tau(1), \cdots, \tau(m)\}$ and the remaining $n - m$ elements by $\rho = \{\rho(1), \cdots, \rho(n - m)\}$. Then, the generation of a nested permutation $\pi$ with 2 layers includes three steps:
**Step 1:** Draw a random permutation \( \pi_0 = \{\pi_0(1), \cdots, \pi_0(m)\} \) on \( \mathbb{Z}_m \).

**Step 2:** For \( i = 1, \cdots, m \), draw \( \tau(i) \) from the discrete uniform distribution with support \( \{(\pi_0(i) - 1)t + 1, (\pi_0(i) - 1)t + 2, \cdots, \pi_0(i)t\} \).

**Step 3:** Draw a random permutation \( \rho \) on \( \mathbb{Z}_n/\tau \).

Then, by independently generating \( d \) nested permutations \( \pi \), one for each column, we obtain a nested Latin hypercube matrix \( A \). Accordingly, we use \( A \) and \( U[0,1) \) random variables to generate a NLHD \( D \) with two layers.

For the generalization of NLHD to more than 2 layers, we assume an integer \( u \geq 3 \), and \( n = \prod_{i=1}^{u} s_i \) with integer \( s_i \geq 1 \). Define \( m_i = \prod_{j=1}^{i} s_j \), \( t_i = n/m_i \) with \( m_u = n \) and \( m_0 = 0 \). We can generate a nested permutation \( \pi \) with \( u \) layers from the following algorithm, for \( i = 1, \cdots, u \),

**Step 1:** Draw a random permutation \( \pi_i = \{\pi_i(1), \cdots, \pi_i(m_i - m_{i-1})\} \) on \( \mathbb{Z}_{m_i}/C_i \), where \( C_1 \) is the empty set, and \( C_i \) is \( \{[\pi(1)/t_i], \cdots, [\pi(m_{i-1})/t_i]\} \) for \( i \geq 2 \).

**Step 2:** For \( j = m_{i-1} + 1, \cdots, m_i \), draw \( \pi(j) \) from the discrete uniform distribution with support \( \{(\pi_i(j-m_{i-1}) - 1)t_i + 1, (\pi_i(j-m_{i-1}) - 1)t_i + 2, \cdots, \pi_i(j-m_{i-1})t_i\} \).

Similar to the 2-layer case, we can generate a nested Latin hypercube design \( D \) with \( u \) layers using the \( u \)-layer nested permutations, together with \( U[0,1) \) random variables. Therefore, the design \( D \) has \( u \) layers, where the \( i^{th} \) layer includes the first \( m_{u+1-i} \) runs. Due to one-dimensional projection properties, there is exactly one point from the \( i^{th} \) layer that falls within each of the \( m_{u+1-i} \) evenly spaced intervals \( (0, 1/m_{u+1-i}], (1/m_{u+1-i}, 2/m_{u+1-i}], \cdots, ((m_{u+1-i} - 1)/m_{u+1-i}, 1] \).
Figure 2.2 shows a NLHD with three layers. The first layer (low accuracy layer) has 24 input points. The second layer has 12 input points nested in the first layer. The third layer (high accuracy layer) has 6 input points nested in the second layer.

Figure 2.2: An example of NLHD with three layers: \( n_1 = 24 \), \( n_2 = 12 \) and \( n_3 = 6 \).

### 2.2.3 Sliced Latin Hypercube Designs

For computer experiments involving both qualitative and quantitative inputs, Qian (2012) proposed a new type of space-filling design: the *sliced Latin hypercube design* (SLHD), which is a special LHD that can be divided into slices of smaller
Latin hypercube designs. Thus, the SLHD can provide a design for each level of the qualitative inputs that is itself a smaller Latin hypercube design. If the levels of the qualitative variable have negligible effect on the output, and we ignore the qualitative variable, the design is still a LHD for the quantitative inputs only.

We consider design construction for $d$ quantitative input variables, each taking values in $(0, 1]$. Given a matrix $A$ having $(i, j)^{th}$ element $a_{ij}$, let $\lceil A \rceil$ denote the matrix whose $(i, j)^{th}$ element is $\lceil a_{ij} \rceil$, where $\lceil a_{ij} \rceil$ is the smallest integer $\geq a_{ij}$. We assume there are $m$ qualitative levels with $t$ design points for each level and $n = mt$.

For the set $Z_n = \{1, \cdots, n\}$, we define a permutation matrix $PM(t, m)$ on $Z_n$ to be a $t \times m$ matrix in which each element of $Z_n$ appears exactly once. In addition, if $A$ is a $PM(t, m)$, and each column of $\lceil A/m \rceil$ forms a permutation on $Z_t$, we call $A$ a $t$ by $m$ sliced permutation matrix, denoted by $SPM(t, m)$. For example, suppose $n = 6, t = 2, m = 3$, we can permute the set $Z_6$ into the matrices $A_1 = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$ and $A_2 = \begin{pmatrix} 1 & 5 & 2 \\ 4 & 3 & 6 \end{pmatrix}$. The permutation to obtain the matrix $A_1$ is a $PM(2,3)$, but not a $SPM(2,3)$, while the permutation to obtain $A_2$ is both a $PM(2,3)$ and a $SPM(2,3)$.

More specifically, we can divide the elements of $Z_n$ into $m$ blocks, $b_1, \cdots, b_t$, where

$$b_i = \{a \in Z_n | \lceil a/t \rceil = i \}, i = 1, \cdots, t.$$ 

Then we can generate an $SPM(t, m)$ in two steps.

**Step 1:** For $i = 1, \cdots, t$, let the $i^{th}$ row of a $t \times m$ matrix $H$ be the random permutation on the set $b_i$ with permutations carried out independently from one row to another.
Step 2: For $j = 1, \cdots, m$, randomly shuffle the entries in the $j^{th}$ column of $H$ with the permutation carried out independently from one row to another.

Thus, we generate an $SPM(t, m) H$ from this algorithm.

We can generate $d$ independent $SPM(t, m)$’s, $H_1, \cdots, H_d$ from the above algorithm. For $c = 1, \cdots, m$, we can define an $m \times d$ matrix $A^{(c)}$ by letting its $j^{th}$ column be the $c^{th}$ column of $H_j$, for $j = 1, \cdots, d$. Then, we can obtain a matrix $A$ by combining $A^{(1)}, \cdots, A^{(m)}$, row by row, denoted by

$$A = \bigcup_{c=1}^{m} A^{(c)}$$

We say, $A$ is an $n$ by $d$ sliced Latin hypercube array with $m$ slices, such that each column of $A$ is a permutation on $Z_n$, and, for $c = 1, \cdots, m$, $\lfloor A^{(c)}/m \rfloor$ is a smaller Latin hypercube of $t$ runs with each column being a permutation on $Z_t$. Furthermore, we can generate the design matrix $D$ through $A$ and $U[0, 1)$ random variables as in (1.4.1). The array $D$ is called a sliced Latin hypercube design (SLHD) and has the following two properties:

(i) The design $D$ is a Latin hypercube design with $n$ levels;

(ii) The subdesign $D^{(c)}$’s, corresponding to $A^{(c)}(c = 1, \cdots, m)$ form a partition of $D$ and each of them is a LHD with $t$ levels.

Figure 2.3 is a 20 by 2 SLHD with 4 slices. The designs for different qualitative levels are displayed by different colors. For each qualitative level, the subdesign is a 5 by 2 LHD. The whole design is a 20 by 2 LHD.
2.2.4 Clustered Sliced Latin Hypercube Designs

SLHDs have appealing space-filling properties for computer models involving both quantitative and qualitative inputs. However, as Huang et al. (2014) point out, these space-filling properties do not guarantee that the correlations among different responses can be accurately captured. By this, Huang et al. (2014) mean that for any pair of qualitative levels, the values of the quantitative inputs in the design may not be (nearby) identical, which Huang et al. (2014) claim may cause difficulties for
the estimated correlation functions to capture the true cross correlations. Figure 2.4 shows the potential prediction issues of an 8 by 2 SLHD with 2 slices. The two curves have a similar sinusoidal pattern which suggests a relatively large cross-correlation value. However, the input points for the red and blue curves are not (nearly) identical. The resulting data would not indicate the similarity of the two curves. The different pattern of the input points for these two curves would lead to a smaller value of the cross-correlation value than the similarity might suggest and hence produce poorer predictions because one would not borrow information from one curve for predicting the other.

To avoid this problem, Huang et al. (2014) proposed a new type of design, called a clustered-sliced Latin hypercube design (CSLHD). CSLHD is a special SLHD, but it is able to better capture the correlations among different responses.

We use similar notation as in Section 2.2.3. That is, we assume the design region is $[0, 1)^d$, the number of qualitative levels is $m$ with $t$ observations on each level, and $n = mt$. The construction of a CSLHD includes four steps:

**Step 1:** For $i = 1, \cdots, t$, let $g_i = ((i - 1)m + 1, (i - 1)m + 2, \cdots, im)^T$.

**Step 2:** Let $\pi = (\pi(1), \cdots, \pi(t))^T$ be a permutation of $Z_t = \{1, \cdots, t\}$. For $i = 1, \cdots, t$, let $\tau_i$ be a random permutation on $g_{\pi(i)}$, and $h_k = (\tau_1(k), \cdots, \tau_t(k))^T$, for $k = 1, \cdots, m$. Define a $n \times 1$ column vector $h$ to be $h = (h_1^T, \cdots, h_m^T)^T$.

**Step 3:** Repeat Step 2 $d$ times independently; each time an $n \times 1$ column vector $h$ is generated and combine those $d$ column vectors, column by column, to obtain an $n \times d$ array $A$. 

Step 4: The design matrix $D$ in $(0,1)^d$ is derived from

$$D = \frac{A - J_{n,d}/2}{n},$$

where $J_{n,d}$ is an $n \times d$ matrix of ones. For $k = 1, \ldots, m$, define $D^{(k)}$ to be the $k^{th}$ slice of $D$ consisting of rows $(k-1)t + 1, \ldots, kt$ of $D$.

Then, the design matrix $D$ is a \textit{clustered-sliced Latin hypercube design} (CSLHD), denoted by $CSLHD(n,d,m)$. The general properties for CSLHD include:
(i) The design $D$ is a LHD with $n$ levels;

(ii) The subdesigns $D^{(k)}$ form a partition of $D$ and each of them is a LHD with $t$ levels.

(iii) For $i = 1, \ldots, t$, let $H^{(i)}$ be the $i^{th}$ subarray of $D$ consisting of all the $i^{th}$ rows of $D^{(1)}, \ldots, D^{(m)}$, and let $M_i$ be the maximum distance for $H^{(i)}$. Then,

$$M = \max_{1 \leq i \leq t} M_i \leq (1 - m^{-1}) \sqrt{d/t}.$$  

The first two properties guarantee that the CSLHD is a SLHD, while the last property ensures the clustered structure, which is claimed to be a key property for capturing the cross correlations among different qualitative levels.

One possible modification of $D$ is that we can generate the design matrix $D$ through $A$ and $U[0, 1)$ random variables as in (1.4.1). Then, the upper bound of $M$ in property (iii) is adjusted to simply $\sqrt{d/t}$. In addition to the clustered structure, the one-dimensional projection of the design matrix $D$ also has the uniform distribution within each subinterval $(i/t, (i + 1)/t], (i = 0, \ldots, t - 1)$.

Furthermore, like the general LHD, the CSLHD is not unique. Huang et al. (2014) also provide a way to find an “optimal” CSLHD based on the centered $L_2$-discrepancy criterion (Hickernell 1998). For a design $D = (d_{ij})$ with $n$ runs and $d$ factors in $[0, 1)^d$, the centered $L_2$-discrepancy is defined by

$$CL_2(D) = \left(\frac{13}{12}\right)^d - \frac{2}{n} \sum_{k=1}^{n} \prod_{l=1}^{d} \left\{ 1 + \frac{1}{2} |d_{kl} - 0.5| - \frac{1}{2} |d_{kl} - 0.5|^2 \right\}$$

$$+ \frac{1}{n^2} \sum_{k=1}^{n} \sum_{j=1}^{n} \prod_{i=1}^{d} \left\{ 1 + \frac{1}{2} |d_{ki} - 0.5| + \frac{1}{2} |d_{ji} - 0.5| - \frac{1}{2} |d_{ki} - d_{ji}| \right\} \quad (2.2.3)$$
Let $\mathcal{D}$ be the class of CSLHDs with $n$ runs, $d$ column and $m$ slices. Then the optimized CSLHD is to find a CSLHD $D^* \in \mathcal{D}$ such that,

$$CL_2(D^*) = \min_{D \in \mathcal{D}} CL_2(D). \quad (2.2.4)$$

Therefore, based on the centered $L_2$-discrepancy and using a column-exchange algorithm approach, one can generate an optimized CSLHD among all the CSLHDs, denoted by $OCSLHD(n, d, m)$.

Figure 2.5 displays a 20 by 2 CSLHD with 4 slices. The designs for different qualitative levels are displayed by different colors. As a special type of SLHD, the subdesign is a 5 by 2 LHD for each qualitative level. The whole design is a 20 by 2 LHD. In addition, for each input point, there exist nearby input points with the other three qualitative levels.

In the construction of the CSLHD, an assumption is being made, namely, the values of the qualitative inputs for any pair of the qualitative levels should be almost identical. The authors claim that this will allow one to better “capture” cross-correlation and hence produce better predictions. However, these authors do not substantiate this claim and we will revisit this in Chapter 4.
Figure 2.5: An example of CSLHD with n=20, d=2, m=4
Chapter 3: Indicator-Based Gaussian Stochastic Process Models for Computer Experiments with Both Quantitative and Qualitative Inputs

The GaSP models in Section 2.1 require statistical algorithms to fit. The choice of the initial parameter values in algorithms that search for the optimum values as well as their ranges can generate predictions with different accuracies. In addition, the complicated correlation structure including both quantitative and qualitative parameters can cause computational difficulties. The correlation matrix \( R \) may become singular or nearly singular for some choices of the correlation parameters. In these cases, it may be difficult to numerically determine the maximum likelihood estimate (MLE) of the parameters. Qian et al. (2008) and Zhou et al. (2011) also provided iterated algorithms to solve such computational difficulties arising from their models for qualitative and quantitative inputs. Their algorithms aim to find the MLE values of the cross-correlation parameters and the quantitative correlation parameters iteratively. Therefore, their algorithms improve the computation time as well as the efficiency of estimating the MLE values.

Alternatively, one can transform the qualitative input structure into a quantitative input structure. In this way, one can fit a GaSP model for the transformed data inputs using existing algorithms for models with quantitative inputs only. In this chapter, we
will discuss two indicator-based GaSP models that transform the qualitative inputs to quantitative inputs. Furthermore, we will compare and discuss the effect of cross-correlation parameters ranges on the prediction accuracies for both the indicator-based models and the candidate models.

3.1 Pairwise-Indicator Gaussian Stochastic Process Model

3.1.1 Motivation and Notation

We follow the notation in Chapter 2. \( X = (x_1, \cdots, x_n)^T \) are the quantitative inputs and \( Z = (z_1, \cdots, z_n)^T \) are the qualitative inputs, where \( x_u = (x_{u1}, \cdots, x_{ud})^T \) is the \( u \)th row vector of quantitative inputs, and \( z_u = (z_{u1}, \cdots, z_{uK})^T \) is the \( u \)th row vector of qualitative inputs for \( u = 1, \cdots, n \). Thus the design matrix can be written as \( W = (X, Z) \). For \( j = 1, \cdots, K \), assume the \( j \)th qualitative variable has \( m_j \) levels, denoted by \( 1, \cdots, m_j \). Let \( m = \prod_{j=1}^{K} m_j \). Then, the \( K \)-dimensional qualitative input variables could be viewed as one qualitative input variable with \( m \) levels, denoted by \( 1, \cdots, m \). For \( u = 1, \cdots, n \), the inputs can be rewritten as \( w_u = (x_u^T, c_u^*)^T \), where \( c_u^* \) represents the level-combination of \( z_u \), which is among \( 1, \cdots, m \).

The corresponding valid correlation structure for the GaSP model with the qualitative inputs replaced by a single qualitative variable \((c_u^*, u = 1, \cdots, m)\) can be derived from (2.1.4). The correlation between \( \epsilon(w_u) \) and \( \epsilon(w_v) \) (Qian et al. (2008)) is given by

\[
\text{corr}(\epsilon(w_u), \epsilon(w_v)) = \tau_{c_u^*, c_v^*} \prod_{i=1}^{d} \exp\{-\phi_i |x_{ui} - x_{vi}|^2\},
\]

(3.1.1)

where \( \tau_{c_u^*, c_v^*} \) represents the cross-correlation between level-combinations \( c_u^* \) and \( c_v^* \), \( u, v = 1, \cdots, m \). A sufficient condition on \( \tau_{c_u^*, c_v^*} \) so that (3.1.1) is a valid correlation structure is the \( m \times m \) matrix \( T = (\tau_{c_u^*, c_v^*}) \) must be a PDUDE. The purpose of
This section is to construct equivalent correlation structures which can simplify the computational complexity of the cross-correlation matrices.

The correlation structure in (3.1.1) involves the estimation of the roughness parameters \( \phi = (\phi_1, \cdots, \phi_d) \) and the \( m \times m \) cross-correlation parameter matrix \( T = (\tau_{c_u^*, c_v^*}) \). An intuitive motivation is to develop a way to represent the qualitative inputs such that the correlation structure is the same as the correlation structure for the GaSP model only involving quantitative inputs. That is, we aim to replace \( c_u^* \) with \( (a_{ij}(c_u^*)) \), and to rewrite the correlation structure as

\[
\text{corr}(\epsilon(w_u), \epsilon(w_v)) = \prod_{i=1}^{d} \exp\left\{ -\phi_i |x_{ui} - x_{vi}|^2 \right\} \prod_{j=1}^{m-1} \prod_{k=j}^{m-1} \exp\left\{ -\psi_{jk} |a_{jk}(c_u^*) - a_{jk}(c_v^*)|^2 \right\},
\]

(3.1.2)

To ensure (3.1.2) is a valid representation of the correlation structure, we need to prove the equivalence between (3.1.1) and (3.1.2). Based on the above motivation, a new type of GaSP model in the presence of qualitative variables called the Pairwise-Indicator Gaussian Stochastic Process (PIGaSP) Model is proposed.

### 3.1.2 Pairwise-Indicator Data Transformation

For each \( w_u = (x_u^T, c_u^T)^T, u = 1, \cdots, n \), define \( I_j(c_u^*) = \begin{cases} 1 & \text{if } c_u^* = j, \\ 0 & \text{otherwise} \end{cases} \) for \( j = 1, \cdots, m \). And define \( a_{jk}(c_u^*) \) as

\[
a_{jk}(c_u^*) = \begin{cases} I_j(c_u^*) + I_k(c_u^*) & \text{for } j \neq k, \\ I_j(c_u^*) & \text{for } j = k. \end{cases}
\]

Thus, we can rewrite \( w_u \) as \( w_u = (x_u^T, a_u^T)^T \), where \( a_u = (a_{11}(c_u^*), \cdots, a_{1,m-1}(c_u^*), a_{22}(c_u^*), \cdots, a_{2,m-1}(c_u^*), \cdots, a_{m-1,m-1}(c_u^*)) \). Note that

\[
a_{jk}(c_u^*) = \begin{cases} 1 & \text{for } j = c_u^*, k \neq c_u^*, \\ 1 & \text{for } j \neq c_u^*, k = c_u^*, \\ 1 & \text{for } j = k = c_u^*, \\ 0 & \text{otherwise.} \end{cases}
\]

The correlation structure of \( w_u = (x_u^T, a_u^T)^T \) can be written as in (3.1.2).
3.1.3 The Equivalence Property

Since $\tau_{c_u^*,c_v^*} \equiv 1, \forall c_u^*$, there are $(m - 1)m/2$ unknown parameters in the cross-correlation parameter matrix $T = (\tau_{c_u^*,c_v^*})$ with $\tau_{c_u^*,c_v^*} = \tau_{c_v^*,c_u^*}$. Thus, by requiring $a_{jk}(c_u^*) = a_{kj}(c_v^*)$ and $\psi_{jk} = \psi_{kj}$, the roughness parameters of the qualitative inputs for the reconstructed data have the same number of unknown parameters as in $T$. To prove the equivalence between (3.1.1) and (3.1.2), we need to derive a non-singular transformation between the cross-correlation parameter matrix $(T = (\tau_{c_u^*,c_v^*}))$ in (3.1.1) and the roughness parameters of the qualitative inputs $(\psi_{jk}, j = 1, \cdots, m - 1, k = j, \cdots, m - 1)$ in (3.1.2). We now do this for the case where all $\tau_{c_u^*,c_v^*} > 0$.

For $c_u^* \neq c_v^*; c_u^*, c_v^* < m$, we can take the logarithm of (3.1.1) and (3.1.2) and derive the expression for $\tau_{c_u^*,c_v^*}$ as

$$- \log \tau_{c_u^*,c_v^*} = \sum_{j,k=1}^{m-1} \psi_{jk}[a_{jk}(c_u^*) - a_{jk}(c_v^*)]^2$$

$$= \sum_{k=1,k \neq c_u^*,c_v^*}^{m-1} \psi_{c_u^*k} + \sum_{j=1,j \neq c_u^*,c_v^*}^{m-1} \psi_{jc_v^*} + \sum_{k=1,k \neq c_u^*,c_v^*}^{m-1} \psi_{c_v^*k}$$

$$+ \sum_{j=1,j \neq c_u^*,c_v^*}^{m-1} \psi_{jc_v^*} + \psi_{c_u^*c_v^*} + \psi_{c_v^*c_u^*}$$

$$= \left( \sum_{k=1,k \neq c_u^*}^{m-1} \psi_{c_u^*k} - \psi_{c_v^*c_v^*} \right) + \left( \sum_{j=1,j \neq c_u^*}^{m-1} \psi_{jc_v^*} - \psi_{c_u^*c_u^*} \right)$$

$$+ \left( \sum_{k=1,k \neq c_v^*}^{m-1} \psi_{c_v^*k} - \psi_{c_u^*c_v^*} \right) + \left( \sum_{j=1,j \neq c_v^*}^{m-1} \psi_{jc_v^*} - \psi_{c_u^*c_v^*} \right) + \psi_{c_u^*c_v^*} + \psi_{c_v^*c_u^*}$$

$$= 2 \sum_{k=1,k \neq c_u^*}^{m-1} \psi_{c_u^*k} + 2 \sum_{j=1,j \neq c_v^*}^{m-1} \psi_{jc_v^*} - 4\psi_{c_u^*c_v^*} + \psi_{c_u^*c_u^*} + \psi_{c_v^*c_v^*}.$$  \hspace{1cm} (3.1.3)
Note that $a_{jk}(m) = 0, \forall j, k$. Thus for $c_u^* = m, c_v^* < m$ and $c_v^* = m, c_u^* < m$, we can derive the expression for $\tau_{m,c_u^*}$ and $\tau_{c_v^*,m}$ as

$$-\log \tau_{m,c_u^*} = \sum_{j,k=1}^{m-1} \psi_{jk} |a_{jk}(m) - a_{jk}(c_u^*)|^2 = \sum_{j,k=1}^{m-1} \psi_{jk} a_{jk}(c_u^*)^2$$

$$= \sum_{k=1,k\neq c_v^*}^{m-1} \psi_{c_v^* k} + \sum_{j=1,j\neq c_v^*}^{m-1} \psi_{j c_v^*} + \psi_{c_v^* c_v^*}$$

$$= 2 \sum_{k=1,k\neq c_v^*}^{m-1} \psi_{c_v^* k} + \psi_{c_v^* c_v^*}; \quad (3.1.4)$$

$$-\log \tau_{c_v^*,m} = 2 \sum_{j=1,j\neq c_u^*}^{m-1} \psi_{j c_u^*} + \psi_{c_u^* c_u^*}. \quad (3.1.5)$$

By subtracting (3.1.4) and (3.1.5) from (3.1.3), we can derive the expression for $\psi_{c_v^* c_u^*}$ for $c_u^* \neq c_v^*, c_u^*, c_v^* < m$ as

$$\psi_{c_v^* c_u^*} = \frac{1}{4} \left( \log \tau_{c_v^* c_u^*} - \log \tau_{mc_v^*} - \log \tau_{c_v^* m} \right). \quad (3.1.6)$$

Then by plugging (3.1.6) into (3.1.5), we can derive the expression for $\psi_{c_v^* c_u^*}$ for $c_u^* = 1, \cdots, m-1$ as

$$\psi_{c_v^* c_u^*} = -\log \tau_{c_v^* m} - 2 \sum_{j=1,j\neq c_u^*}^{m-1} \psi_{j c_u^*}$$

$$= -\log \tau_{c_v^* m} + \frac{1}{2} \sum_{j=1,j\neq c_u^*}^{m-1} \left( \log \tau_{j c_u^*} - \log \tau_{mc_u^*} - \log \tau_{jm} \right). \quad (3.1.7)$$

Therefore, there exists a non-singular transformation between the cross-correlation parameter matrix in (3.1.1), assuming all $\tau_{c_v^*,c_u^*}$'s are restricted to be greater than zero, and the roughness parameters of the qualitative inputs in (3.1.2). That means, instead of using the cross-correlation parameters, we can reconstruct the input data with both quantitative and qualitative variables into the form $w_u = (x_u^T, a_u^T)^T$ and use the regular GaSP model as in Section 1.3 to get an equivalent model for the
response variables. Note, a valid correlation function of (3.1.2) requires $\psi_{jk} \geq 0$ for $j = 1, \ldots, m - 1$ and $k = 1, \ldots, m - 1$.

### 3.1.4 Remarks

For the purpose of simplifying numerical computation, several choices of $\tau_{c_i,c_j}$ in the literature have been proposed. Based on the equivalence property between positive $\tau_{c_i,c_j}$ and $\psi_{jk}$, we can derive the expression for the $\psi_{jk}$ in each circumstance.

1. **Exchangeable Correlation** Stein (1999) imposed $\tau_{c^*_j,c^*_k} = c \ (0 < c < 1)$ for $j \neq k$, which has been applied by Joseph and Delaney (2007) to model some physical experiments. Thus, in our case, the $\psi_{c^*_j,c^*_k}$'s become

   $$
   \psi_{c^*_j,c^*_k} = \frac{1}{4} (\log c - \log c - \log c) = -\frac{1}{4} \log c;
   $$

   $$
   \psi_{c^*_j,c^*_j} = - \log c - 2 \sum_{j=1, j \neq c^*_u}^{m-1} \frac{1}{4} \log c = \frac{1}{2} (m - 4) \log c.
   $$

2. **Multiplicative Correlation** McMillian et al. (1999) imposed $\tau_{c^*_j,c^*_k} = \exp\{-\lambda_j + \lambda_k\} I(j \neq k) \ (\lambda_j, \lambda_k > 0)$ for $m \geq 4$. Accordingly, we can show that

   $$
   \psi_{c^*_j,c^*_k} = \frac{1}{4} (-\lambda_j - \lambda_k + \lambda_m + \lambda_k + \lambda_m + \lambda_j) = \frac{1}{2} \lambda_m;
   $$

   $$
   \psi_{c^*_j,c^*_j} = \lambda_j + \lambda_m - 2 \sum_{j=1, j \neq c^*_u}^{m-1} \frac{1}{2} \lambda_m = \lambda_j - (m - 3) \lambda_m.
   $$

3. When the number of qualitative variables is large, the number of parameters to be estimated is also large. As in (2.1.4), Qian et al. (2008) proposed a product form of the correlation functions which can simplify computations,

   $$
   corr(\epsilon(w_u), \epsilon(w_v)) = \left[ \prod_{j=1}^{K} z^{(j)}_{x_{u_j},x_{v_j}} \right] \left[ \prod_{i=1}^{d} \exp\{-\phi_i|x_{ui} - x_{vi}|^2\} \right],
   $$

53
Similarly, we can transform the qualitative inputs into the indicator inputs for each superscript and define the corresponding parameters for each superscript.

For each \( w_u = (x_u^T, z_u^T)^T, u = 1, \ldots, n \), define the indicator for the \( l \)th qualitative input as \( I_{j}^{(l)}(z_u) = I(z_{lj} = z_{lu}) \), for \( j = 1, \ldots, m_j - 1 \), and \( a_{jk}^{(l)}(z_u) \) as

\[
a_{jk}^{(l)}(z_u) = \begin{cases} 
I_j^{(l)}(z_u) + I_k^{(l)}(z_u) & \text{for } j \neq k; \\
I_j^{(l)}(z_u) & \text{for } j = k.
\end{cases}
\]

Thus, the pairwise-indicator correlation structure can be written as

\[
\text{corr}(\epsilon(w_u), \epsilon(w_v)) = \prod_{i=1}^{d} \exp\{-\theta_i |x_{ui} - x_{vi}|^2\} \exp\{-\psi_{jk}^{(l)} |a_{jk}^{(l)}(z_u) - a_{jk}^{(l)}(z_v)|^2\}.
\]

### 3.2 Individual-Indicator Gaussian Stochastic Process Model

#### 3.2.1 Motivation

Kennedy and O’Hagan (2000) discuss a general form for the Gaussian correlation structure. That is, for any two quantitative inputs \( x_1 \) and \( x_2 \), the Gaussian correlation function can be written as

\[
R(x_1 - x_2) = \exp\{- (x_1 - x_2)^T \Omega (x_1 - x_2) \} \quad (3.2.1)
\]

where \( \Omega \) is an unknown symmetric positive-definite matrix. In particular, the Gaussian correlation function in (1.3.3) has the form \( \Omega = \text{diag}(\phi_1, \ldots, \phi_d) \).

Motivated by the generalization of the Gaussian correlation structure in (3.2.1) and the hyperspherical parameterization in Section 2.1.2, we propose a new form for the GaSP model for computer experiments with both quantitative and qualitative inputs, called the Individual-indicator Gaussian Stochastic Process (IIGaSP) model,
which is also based on representing the qualitative inputs as quantitative. Similarly, with this representation, data can be fit directly through the standard models only involving quantitative inputs.

3.2.2 Qualitative Indicator Vector Construction

For each $w_u = (x_u^T, c_u^*)^T$, $u = 1, \cdots, n$, we define an $(m-1)$-dimensional indicator vector $q_u$ as

$$q_u = \begin{cases} e_i & \text{for } c_u^* = i < m; \\ 0 & \text{for } c_u^* = m, \end{cases}$$

(3.2.2)

where $e_i, i = 1, \cdots, m-1$ is a set of $(m-1)$-dimensional basis vectors, and $(e_1, \cdots, e_{m-1})$ forms the $(m-1)$-dimensional identity matrix. Then, we can rewrite $w_u$ as

$$w_u = (x_u^T, q_u^T)^T,$$

for $u = 1, \cdots, m$.

3.2.3 Individual Indicator Cross-Correlation Matrix

From this new representation of the inputs, we can write the general Gaussian correlation function between two sets of inputs $w_u = (x_u^T, q_u^T)^T$ and $w_v = (x_v^T, q_v^T)^T$ ($u \neq v$) as

$$\text{corr}(\epsilon(w_u), \epsilon(w_v)) = \prod_{i=1}^d \exp\{-\phi_i |x_{ui} - x_{vi}|^2\} \exp\{- (q_u - q_v)^T \Omega (q_u - q_v)\}$$

(3.2.3)

where $\Omega = (\omega_{uv})$ is an unknown symmetric positive-definite matrix corresponding to the roughness parameters for the reorganized qualitative inputs.

It is easy to show the equivalence between the cross-correlation parameters in (3.1.1) and the roughness parameter matrix in (3.2.3) provided the $\tau_{c_u^*, c_v^*}$’s are restricted to be positive. Comparing these two correlation functions, we have

$$\tau_{c_u^*, c_v^*} = \exp\{- (q_u - q_v)^T \Omega (q_u - q_v)\}$$

(3.2.4)
By taking the logarithm of (3.2.4) and doing some matrix multiplications, we can show

\[- \log \tau_{cc^*} = \omega_{uu} + \omega_{vv} - 2\omega_{uv}, \quad \text{for } c_u^* \neq c_v^* < m; \]  
\[- \log \tau_{cc^*,m} = \omega_{uu}, \quad \text{for } c_u^* < m. \]  

Thus, we can obtain the expressions for \(\omega_{uv}\) from (3.2.5) and (3.2.6), which are

\[\omega_{uv} = \frac{1}{2}(\log \tau_{cc^*} - \log \tau_{cc^*,m} - \log \tau_{cc^*,m}), \quad \text{for } c_u^* \neq c_v^* < m; \]  
\[\omega_{uu} = -\log \tau_{cc^*,m}, \quad \text{for } c_u^* < m. \]  

Therefore, there exists a non-singular transformation between the cross-correlation parameters in (3.1.1) and the indicator’s roughness parameter matrix in (3.2.3) in the setting where the \(\tau_{cc^*,c_v}\)’s are restricted to be positive.

Note, if we impose a diagonal structure on \(\Omega\), i.e. \(\Omega = \text{diag}(\omega_1, \cdots, \omega_{m-1})\), the correlation function becomes

\[\text{corr}(\epsilon(w_u), \epsilon(w_v)) = \prod_{i=1}^{d} \exp\{-\phi_i |x_{ui} - x_{vi}|^2\} \prod_{j=1}^{m-1} \exp\{-\omega_j |q_{uj} - q_{vj}|^2\} \]  

From equation (3.2.7) and (3.2.8), the diagonal structure on \(\Omega\) imposes a special multiplicative structure on \(\tau_{cc^*,c_v}\), such that

\[\tau_{cc^*} = \tau_{cc^*,m} \tau_{mc^*} = \exp\{- (\omega_{uu} + \omega_{vv})\}, \quad \text{for } c_u^* \neq c_v^* < m; \]  
\[\tau_{cc^*,m} = \exp\{-\omega_{uu}\} \neq 0, \quad \text{for } c_u^* < m. \]  

Assuming this special structure on \(\tau_{cc^*,c_v}\) holds, formula (3.2.9) implies that we can fit the GaSP model with only quantitative inputs for the \((d + m - 1)\)-dimensional reorganized data \(w_u = (x_u^T, q_u^T)^T\). Instead of complicated algorithms, the reorganized
data can be easily fit with standard GaSP model packages or software, like mPerk, JMP, DACE, etc.

Note that for the first two special cases of $\tau_{c^*c^*}$ mentioned in Section 3.1.4, $\omega_{uv}$ can also be simplified:

1. **Exchangeable Correlation:** If we impose $\tau_{c^*c^*} = c (0 < c < 1)$ for $j \neq k$, we have
   \[
   \begin{align*}
   \omega_{uv} &= -\frac{1}{2} \log c, & \text{for } u \neq v; \\
   \omega_{uu} &= -\log c, & \text{for } u = 1, \ldots, m - 1.
   \end{align*}
   \]

2. **Multiplicative Correlation:** If we let $\tau_{c^*c^*} = \exp\{-(\lambda_u + \lambda_v)I(u \neq v)\}(\lambda_u, \lambda_v > 0)$, we can show that
   \[
   \begin{align*}
   \omega_{uv} &= \lambda_m, & \text{for } u \neq v; \\
   \omega_{uu} &= \lambda_u + \lambda_m, & \text{for } u = 1, \ldots, m - 1.
   \end{align*}
   \]

After transforming the qualitative inputs, if we fit the model with (3.2.9), standard GaSP model in (3.2.9) might lead to poor prediction accuracy. One can use the more general symmetric positive-definite structure to improve the prediction accuracy. Considering the potential computational complexity caused by the number of parameters in the unconstrained $\Omega$, the hyperspherical coordinate representation in Section 2.1.2 provides a way to reduce the computational issues without loss of flexibility on the choice of the parameters.

The constraint on $\Omega$ is that it needs to be a symmetric positive-definite matrix. Thus, we can apply a Cholesky decomposition to $\Omega$. That is,

\[
\Omega = LL^T, \quad \text{(3.2.10)}
\]
where $L = (l_{r,s})$ is a $(m - 1) \times (m - 1)$ dimensional lower triangular matrix with strictly positive diagonal entries.

Next, similar to the hyperspherical parameterization in Section 2.1.2, we can model each row vector $(l_{r,1}, \ldots, l_{r,r})(r = 1, \ldots, m - 1)$ as a point on the $r$-dimensional unit hypersphere described as follows. For $r = 1$, let $l_{1,1} = t_1(t_1 > 0)$ and for $r = 2, \ldots, (m - 1)$, the hyperspherical coordinate system is

\[
\begin{align*}
    l_{r,1} &= t_r \cos(\theta_{r,1}), \\
    l_{r,s} &= t_r \sin(\theta_{r,1}) \cdots \sin(\theta_{r,s-1}) \cos(\theta_{r,s}), \quad \text{for} \ s = 2, \ldots, r - 1, \\
    l_{r,r} &= t_r \sin(\theta_{r,1}) \cdots \sin(\theta_{r,r-1}) \sin(\theta_{r,r-1})
\end{align*}
\]

(3.2.11)

where $\theta_{r,s} \in (0, \pi)$, $t_r > 0$. The structure of the hyperspherical coordinate system implies that there are $(m - 1) + \sum_{r=2}^{m-1} (r - 1) = \frac{m(m-1)}{2}$ unknown parameters in $L$, which is the same number of unknown parameters in $\Omega$. Also, equation (3.2.10) guarantees that $\Omega$ generated by these $\theta_{r,s}$’s is always a symmetric positive-definite matrix.

For illustration, consider the case with a 4-level qualitative input, i.e., $m = 4$, in which case the matrix $\Omega$ is a $3 \times 3$ symmetric positive definite matrix

\[
\Omega = \begin{bmatrix}
    \omega_{11} & \omega_{12} & \omega_{13} \\
    \omega_{12} & \omega_{22} & \omega_{23} \\
    \omega_{13} & \omega_{23} & \omega_{33}
\end{bmatrix},
\]

(3.2.12)

and is decomposed as

\[
\Omega = LL^T = \begin{bmatrix}
t_1 & 0 & 0 \\
l_{21} & l_{22} & 0 \\
l_{31} & l_{32} & l_{33}
\end{bmatrix} \begin{bmatrix}
t_1 & l_{21} & l_{31} \\
l_{21} & l_{22} & l_{32} \\
l_{31} & l_{32} & l_{33}
\end{bmatrix}.
\]

(3.2.13)

Next, we can transform $(l_{21}, l_{22})$ into a 2D hyperspherical coordinate system

\[
\begin{align*}
    l_{21} &= t_2 \cos(\theta_{21}), \\
    l_{22} &= t_2 \sin(\theta_{21})
\end{align*}
\]

(3.2.14)

and $(l_{31}, l_{32}, l_{33})$ into a 3D hyperspherical coordinate system

\[
\begin{align*}
    l_{31} &= t_3 \cos(\theta_{31}), \\
    l_{32} &= t_3 \sin(\theta_{31}) \cos(\theta_{32}), \\
    l_{33} &= t_3 \sin(\theta_{31}) \sin(\theta_{32})
\end{align*}
\]

(3.2.15)
where we can derive a non-singular transformation between $\Omega$ and $(t_r, \theta_{r,s})$ from the following equation

$$
\begin{align*}
\omega_{11} &= t_1^2; \\
\omega_{22} &= t_2^2; \\
\omega_{33} &= t_3^2; \\
\omega_{21} &= \omega_{12} = t_1 t_2 \cos(\theta_{21}); \\
\omega_{31} &= \omega_{13} = t_1 t_3 \cos(\theta_{31}); \\
\omega_{23} &= \omega_{32} = t_2 t_3 (\cos(\theta_{21}) \cos(\theta_{31}) + \sin(\theta_{21}) \sin(\theta_{31}) \cos(\theta_{32})).
\end{align*}
$$

(3.2.16)

Therefore, the matrix $\Omega$ generated by $(t_r, \theta_{r,s})$ satisfies the symmetric positive definite property automatically. By using the hyperspherical coordinate system, computational complexity in estimating $\Omega$ under the constraint that it be positive definite is reduced. Ultimately, we can fit a general GaSP model from the representation $w_u = (x_u^T, q_u^T)^T$ and the correlation structure of (3.2.3) to obtain predictions for the response variables. To distinguish the hyperspherical structure of $\Omega$ matrix from the general structure, we call the model with the hyperspherical structure in (3.2.10) and (3.2.11) the *Hypersphere Indicator Gaussian Stochastic Process* (HIGaSP) model. Note that IIGaSP model and HIGaSP model are equivalent, the names only indicate the different representation systems on $\Omega$.

### 3.3 A Simple Application of the PIGaSP Model and IIGaSP Model on Commercial Software

The PIGaSP model and the IIGaSP model provide methods for representing the qualitative input variables as quantitative through the use of indicator variables, which transforms the complicated situation with both quantitative and qualitative inputs into the regular situation with only quantitative inputs. Some commercial software, such as JMP, allows one to fit a stationary GaSP model with the Gaussian correlation function for quantitative inputs. Therefore, we can fit a stationary GaSP
model with both quantitative and qualitative inputs using commercial software if we adopt the representations of the PIGaSP and IIGaSP models. Although there are some constraints (we will discuss the constraints later) when using the commercial software, it is convenient to apply the software directly. As long as the response patterns among the responses with different qualitative are very similar, we can achieve a good fit from commercial software.

We will use JMP to illustrate the process of fitting the two models in Section 3.1 and Section 3.2.

**Example 3.3.1.** Consider an experiment involving one qualitative variable with four levels, (1, 2, 3, 4), and one quantitative variable $x$. The simulated response curves are taken from the class $y(x) = a_1 \exp(a_2 x) + a_3 (x - a_4)^2$ with $x \in [0, 1)$. The coefficients values of $(a_1, a_2, a_3, a_4)$ for these four levels are set to be $(1, 3, 5, 2), (2, 3, 2, 1), (3, 1, 10, 3), (1, 4, 5, 4)$, respectively.

The training input data (Table 3.1) are generated by using a CSLHD with $n = 16$, $d = 1$, and $m = 4$ as described in Section 2.2.4. Figure 3.1 shows the observations and the true response curves. The inputs for evaluating the fit of the predictors are 100 equally-spaced points in $(0, 1] (\{0.01, 0.02, \ldots, 1\})$.

<table>
<thead>
<tr>
<th>$z$</th>
<th>$0.5134$</th>
<th>$0.4254$</th>
<th>$0.1592$</th>
<th>$0.9304$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z=1$</td>
<td>0.5657</td>
<td>0.4809</td>
<td>0.2292</td>
<td>0.7894</td>
</tr>
<tr>
<td>$z=3$</td>
<td>0.7003</td>
<td>0.2898</td>
<td>0.0947</td>
<td>0.8700</td>
</tr>
<tr>
<td>$z=4$</td>
<td>0.6461</td>
<td>0.3435</td>
<td>0.0402</td>
<td>0.9996</td>
</tr>
</tbody>
</table>

We consider four methods to fit the model:
Method 1: Independently fit separate stationary GaSP model for each qualitative level.

Method 2: Apply the transformation method in Section 3.1 to the data, fit a GaSP model for the transformed data $w = (x, a^T)^T$.

Method 3: Apply the transformation method in Section 3.2 to the data, fit a GaSP model for the transformed data $w = (x, q^T)^T$. Note that the hyperspherical structure for the cross-correlation matrix $\Omega$ can not be imposed in JMP. However, we can assume the diagonal structure for the cross-correlation matrix $\Omega$ and carry out the predictions from JMP.
Method 4: Due to different correlation structures, JMP can not fit the model in Qian et al. (2008) and Zhou et al. (2011) directly. As a competing candidate model, we use MATLAB to fit the hypersphere parameterization (HP) model in Section 2.1.2.

The root-mean-squared error (RMSE) of a predictor $\hat{y}(\cdot)$ at level $k$ over $N$ test points $x_1^*, \cdots, x_N^*$ is

$$\text{RMSE}_k = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y(x_i^*, k) - \hat{y}(x_i^*, k))^2},$$

which is a measure of the prediction accuracy for qualitative level $k$. The overall RMSE can also be computed for all the curves by

$$\text{RMSE} = \sqrt{\frac{1}{mN} \sum_{k=1}^{m} \sum_{i=1}^{N} (y(x_i^*, k) - \hat{y}(x_i^*, k))^2},$$

which is a measure of the overall prediction accuracy for all the qualitative levels.

For each model, the RMSEs are calculated for each level of the qualitative variable as well as overall. The summary of the RMSEs of the above four models are listed in Table 3.2. Figure 3.2 displays the comparison of these four models with the true functions.

<table>
<thead>
<tr>
<th>Qualitative Level</th>
<th>Separate Model</th>
<th>PI GaSP Model</th>
<th>II GaSP Model with $\Omega$ a diagonal matrix</th>
<th>HP Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{RMSE}_1$</td>
<td>0.9911</td>
<td>0.1223</td>
<td>0.1775</td>
<td>0.1420</td>
</tr>
<tr>
<td>$\text{RMSE}_2$</td>
<td>2.5773</td>
<td>1.4976</td>
<td>0.7519</td>
<td>0.7893</td>
</tr>
<tr>
<td>$\text{RMSE}_3$</td>
<td>0.0047</td>
<td>0.0991</td>
<td>0.1025</td>
<td>0.1472</td>
</tr>
<tr>
<td>$\text{RMSE}_4$</td>
<td>3.4618</td>
<td>1.6520</td>
<td>1.5473</td>
<td>1.8048</td>
</tr>
<tr>
<td>Overall $\text{RMSE}$</td>
<td>2.2141</td>
<td>1.1177</td>
<td>0.8663</td>
<td>0.9902</td>
</tr>
</tbody>
</table>
Figure 3.2: A comparison of the Example 3.3.1 with the true curves for the four models (Separate model, PIGaSP model, IIGaSP Model with $\Omega$ a diagonal matrix, and Hyperspherical Parameterization Model). The left top is for $Y(X, 1)$, the right top is for $Y(X, 2)$, the left bottom is for $Y(X, 3)$, and the right bottom is for $Y(X, 4)$.

The results indicate that the separate-model method performs well at level $z = 3$ of the qualitative input. This may be because at level 3 the curve is very simple and only a few observations are needed for a good fit. However, in general, the performance of the separate-model method is not as accurate as the other methods due to the lack of information.

Among the other three methods, we might expect HP model to do best because $-1 < \tau < 1$, whereas the other models constrain $0 < \tau < 1$. PIGaSP model might
be expected to outperform IIGaSP with $\Omega$ diagonal because it uses more parameters. However, IIGaSP with $\Omega$ diagonal is the least computationally complex, and hence it may give better results, because due to this numerical simplicity, it is more effective numerically at finding optimum parameter estimates.

For the HP model, we use MATLAB to carry out the fitting procedure. The results vary for different bounds on the quantitative correlation parameter $\phi$ in HP model. The results shown above set 1.5 as the upper bound of $\phi$. A larger value of $\phi$’s upper bound might lead to a worse prediction accuracy, sometimes even worse than the separate-model if the likelihood for $\phi$ becomes flat as $\phi$ increases. In contrast, these two indicator-based methods use the default setting of the parameter boundaries in JMP, and perform well on the prediction accuracy for all the qualitative levels. Among these three models, the PIGaSP model performs the best for $z = 1, 3$, while the IIGaSP model does the best for $z = 2, 4$ and for the overall fitting. The performance of the HP model is fairly good, in between the two indicator-based models for $z = 1, 2$ and the overall fitting. It performs the worst for $z = 3, 4$.

The under-performance of the HP model might be caused by the bounds placed on $\phi$. Also, the similar trends among these four qualitative levels suggest positive cross-correlations which weaken the advantage of allowing $-1 < \phi < 1$ in the HP model. Compared to the IIGaSP model, the PIGaSP model has more parameters and one might expect better performance. However, the complex structure of the PIGaSP model might also cause the computational problems. For qualitative inputs with a large number of levels, the representation of the qualitative inputs in PIGaSP model leads to a large number of parameters, which might lead to problems identifying the MLEs and hence poor prediction accuracy. Therefore, the IIGaSP model with $\Omega$
diagonal provides a simpler but numerically stable form of these two indicator-based models.

Note that the correlation structure must be the traditional Gaussian correlation structure when one wants to fit a GaSP model in JMP. Thus, we can not apply the hyperspherical structure of the Gaussian correlation structure directly in JMP. However, we can modify the codes for Gaussian correlation functions in some software, such as MATLAB, to fit the HIGaSP model in Section 3.2, in which case fitting the model is more efficient computationally although there are more parameters to estimate. Furthermore, for some complicated simulation examples, because of the inability to fully constrain the parameter space, JMP may not yield a good results in the sense of prediction accuracy. In these situations, one can use software, like MATLAB, which allows one to specify the initial values of the parameters or impose constraints on the parameters to narrow the region over which one searches for MLEs.

3.4 The Effect on Prediction Accuracy from the Restrictions on the Range of the Cross-Correlation Parameters

The condition that the $T_i$’s be PDUDE doesn’t constrain the $\tau_{s,t}$’s to be non-negative. Intuitively, if $\tau_{s,t} < 0$, this corresponds to a case where curves $r$ and $s$ are “negatively correlated”. This assumes $\tau_{s,t}$ really represents this “negative correlation”. When fitting two curves that appear to be negatively correlated, intuitively one might expect that allowing the cross-correlation parameters to be negative will improve predictions. However, this has not been explored empirically. We will explore this further with a simulation example.
Note that, for \( \mathbf{w}_1 = (x, z_1), \mathbf{w}_2 = (x, z_2) \),

\[
R(\mathbf{w}_1 - \mathbf{w}_2) = \tau_{z_1, z_2}
\]

and for \( \mathbf{w}_1 = (x, z), \mathbf{w}_2 = (x, z) \),

\[
R(\mathbf{w}_1 - \mathbf{w}_2) = \exp\left\{ -\sum_{i=1}^{d} \phi_i |x_{i1} - x_{i2}|^p \right\}
\]

### 3.4.1 Motivating Example

In this subsection, we introduce an example to investigate the effect of constraints on the cross-correlation parameters.

**Example 3.4.1.** Motivated by Zhou et al. (2011)’s cosine function example, we create an example with one curve that would be considered highly negatively correlated with the others.

\[
y = \begin{cases} 
-3 \cos(3\pi x) & z_1 = 1 \\
2 \cos(3\pi x) & z_1 = 2 \\
-2 \cos(3\pi x) & z_1 = 3
\end{cases}
\]

Figure 3.3 shows the curves corresponding to the three different qualitative levels. We would expect that \( \tau_{13} \) will be positive and close to 1 and \( \tau_{12} \) and \( \tau_{23} \) will be negative and close to -1, if the \( \tau_{ij} \) actually measure cross-correlation.

We generate the training data set using an \( n \times d \) CSLHD (Huang et al. (2014)) with \( m \) slices as described in Section 2.2.4. We let \( n = 18, d = 1, \) and \( m = 3 \). The training data are listed in Table 3.3.

Furthermore, we let the inputs for computing MSPE of the predictor be 20 equally spaced points taken from \([0, 1] \) \((0.075, 0.125, \cdots, 0.975)\).

Three restrictions on the range of the cross-correlation parameters are given by:
Figure 3.3: True Curves of Example 3.4.1

Table 3.3: **Training data for Example 3.4.1**

<table>
<thead>
<tr>
<th>$z$</th>
<th>0.1732</th>
<th>0.4699</th>
<th>0.6708</th>
<th>0.0457</th>
<th>0.6049</th>
<th>0.8349</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z=1$</td>
<td>0.2407</td>
<td>0.4172</td>
<td>0.7487</td>
<td>0.1558</td>
<td>0.6638</td>
<td>0.9785</td>
</tr>
<tr>
<td>$z=2$</td>
<td>0.2969</td>
<td>0.3482</td>
<td>0.8075</td>
<td>0.0879</td>
<td>0.5370</td>
<td>0.9079</td>
</tr>
</tbody>
</table>

**Full range:** The ranges of $\tau_{12}, \tau_{13}, \tau_{23}$ are all from (-1,1).

**Positive range:** The ranges of $\tau_{12}, \tau_{13}, \tau_{23}$ are all from [0,1).

**Restrictive range:** The ranges of $\tau_{12}, \tau_{13}, \tau_{23}$ are $(-1,0], [0,1)$ and $(-1,0]$.  

67
We use MATLAB to carry out the original algorithm for the cross-correlation (CC) GaSP model in Section 2.1.1. We fit the model directly from the cross-correlation matrix $T$ without any transformation or constraint on $T$. Table 3.4 shows the comparison of the parameter estimates as well as the RMSEs among these different restrictions.

Table 3.4: Comparison of the parameter estimates and RMSEs among three different restrictions on the range

<table>
<thead>
<tr>
<th>Methods</th>
<th>$\tau_{12}$</th>
<th>$\tau_{13}$</th>
<th>$\tau_{23}$</th>
<th>$\phi_1$</th>
<th>$RMSE_1$</th>
<th>$RMSE_2$</th>
<th>$RMSE_3$</th>
<th>$RMSE$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CC_{Full}$</td>
<td>-0.9077</td>
<td>0.6564</td>
<td>-0.2340</td>
<td>15.3433</td>
<td>0.0433</td>
<td>0.2259</td>
<td>0.3201</td>
<td>0.1690</td>
</tr>
<tr>
<td>$CC_{Positive}$</td>
<td>0.0351</td>
<td>0.9371</td>
<td>0.4182</td>
<td>9.9341</td>
<td>1.7306</td>
<td>1.3037</td>
<td>0.9645</td>
<td>1.3653</td>
</tr>
<tr>
<td>$CC_{Restricted}$</td>
<td>-0.3710</td>
<td>0.8987</td>
<td>-0.8889</td>
<td>11.6435</td>
<td>0.1986</td>
<td>0.0447</td>
<td>0.0593</td>
<td>0.1178</td>
</tr>
</tbody>
</table>

As discussed in Zhou et al. (2011), the unrestrictive correlation structure of the qualitative inputs also introduces computational complexity. The PDUDE property of the cross-correlation matrix can easily produce poor estimates of the MLEs and hence lead to bad predictions. The results of each scenario in Table 3.4 are derived from the best prediction results of several runs. Table 3.4 clearly indicates that the restriction that $\tau_{12}$ and $\tau_{23}$ be positive affects the predictions severely. When we allow $\tau_{12}$ and $\tau_{13}$ to be negative, the full range and restrictive range of $\tau$’s can perform much better than the positive range on the prediction accuracy. In practice, however, one does not have any prior knowledge on the range of $\tau$’s. The full range constraint on $\tau$’s should provide a decent prediction accuracy in general. This example supports the assumption that the $\tau_{ij}$ measure cross-correlation in some sense.
3.4.2 Simulation Comparisons with the Candidate Models

Section 3.4.1 illustrates the effect of the cross-correlation parameters’ ranges on the prediction accuracy. We have discussed several other models (exchangeable correlation model, multiplicative correlation model, the general model using the hyperspherical parameterization, PIGaSP model and HIGaSP model) that can be used to obtain predictors. However, due to the nature of the transformations, some of these only allow positive cross-correlation parameters. In this section, we will compare and illustrate the effect of the cross-correlation ranges among several different methodologies in terms of prediction accuracy.

We assume an experiment with $K$-dimensional qualitative inputs. For each existing frequentist GaSP model, the corresponding cross-correlation ranges can be derived as follows:

**Exchangeable Correlation** For $k = 1, \cdots, K$, $\tau_{ij}^{(k)} = \exp\{-\theta_k I\{z_i^{(k)} \neq z_j^{(k)}\}\}$ is positive. Therefore, the ranges of the cross-correlation parameters are all positive for the exchangeable correlation model.

**Multiplicative Correlation** For $k = 1, \cdots, K$, $\tau_{ij}^{(k)} = \exp\{- (\theta_i^{(k)} + \theta_j^{(k)}) I\{z_i^{(k)} \neq z_j^{(k)}\}\}$ is positive. Therefore, the ranges of the cross-correlation parameters are all positive for the multiplicative correlation model.

**Hypersphere Correlation** From (2.1.11), for $k = 1, \cdots, K$, $t_r^{(k)}$ could be negative when $\theta_{r,s}^{(k)} \in (\pi/2, \pi)$ ($r = 2, \cdots, m_k$, and $s = 1, \cdots, r - 1$). Therefore, one can derive negative values for the off-diagonal element in $T_k$ when the ranges of $\theta$’s are all from $(0, \pi)$. Accordingly, by constraining all the ranges of $\theta$’s on $(0, \pi/2)$, one can restrict to all-positive cross-correlation parameters.
**Pairwise-Indicator Correlation** For each qualitative input, the exponential transformation from $\psi$ to $\tau$ guarantees that the $\tau$'s are positive.

**Hypersphere-Indicator Correlation** From (3.2.4), exponential transformation from the positive definite matrix $\Omega$ to $\tau$ also guarantees the $\tau$'s are positive.

In Example 3.4.1, we investigated the effect of the cross-correlation parameter range on the prediction accuracy. In this section, we will compare the results in Table 3.4 with other candidate models in terms of prediction accuracy. The candidate models include the original cross-correlation based GaSP model (CC), hypersphere parameterization model (HP), the Pairwise-Indicator GaSP model (PI), the Hypersphere-Indicator GaSP model (HI), exchangeable cross-correlation model (EX) and multiplicative cross-correlation model (MU). We use maximum likelihood to estimate parameters in the models and the EBLUP for prediction. For all the candidate models, we compute the corresponding estimated $\tau$'s to allow for comparisons. In addition, to assess the effect of the cross-correlation parameter range, we also enforce certain different restrictions on the candidate models. For the HP model, if the bounds for all the $\theta$'s are set to be $(0, \pi/2]$, the corresponding $\tau$'s are positive. In contrast, if the bounds for all the $\theta$'s are between 0 and $\pi$, the corresponding $\tau$'s can be estimated either positive or negative. For all the exponential based transformation models, the range of $\tau$'s are automatically positive.

Table 3.5 shows that the comparison of the estimated $\tau$’s, $\phi_1$, individual RMSEs, overall RMSEs and the corresponding -log-likelihood for all the scenarios. Figure 3.4 compares the predictions of each candidate models with the true curves.
Table 3.5: Comparison of the parameter estimates and the RMSEs among different methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>$\hat{\tau}_{12}$</th>
<th>$\hat{\tau}_{13}$</th>
<th>$\hat{\tau}_{23}$</th>
<th>$\phi_1$</th>
<th>RMSE$_1$</th>
<th>RMSE$_2$</th>
<th>RMSE$_3$</th>
<th>RMSE</th>
<th>-log-like</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC$_{Full}$</td>
<td>-0.9077</td>
<td>0.6564</td>
<td>-0.2340</td>
<td>15.3433</td>
<td>0.0433</td>
<td>0.2259</td>
<td>0.3201</td>
<td>0.1690</td>
<td>-584.6237</td>
</tr>
<tr>
<td>CC$_{Positive}$</td>
<td>0.0351</td>
<td>0.9371</td>
<td>0.4182</td>
<td>9.9341</td>
<td>1.7306</td>
<td>1.3037</td>
<td>0.9645</td>
<td>1.3653</td>
<td>-413.9012</td>
</tr>
<tr>
<td>CC$_{Restricted}$</td>
<td>-0.3710</td>
<td>0.8987</td>
<td>-0.8889</td>
<td>11.6435</td>
<td>0.1986</td>
<td>0.0447</td>
<td>0.0593</td>
<td>0.1178</td>
<td>-557.7099</td>
</tr>
<tr>
<td>HP$_{Full}$</td>
<td>-0.7089</td>
<td>0.7420</td>
<td>-0.6926</td>
<td>20.0000</td>
<td>0.2879</td>
<td>0.0869</td>
<td>0.1183</td>
<td>0.1866</td>
<td>-6.5070</td>
</tr>
<tr>
<td>HP$_{Positive}$</td>
<td>0.0000</td>
<td>0.8650</td>
<td>0.0729</td>
<td>20.0000</td>
<td>0.3699</td>
<td>0.1651</td>
<td>0.2086</td>
<td>0.2631</td>
<td>-0.0529</td>
</tr>
<tr>
<td>PI$_{Positive}$</td>
<td>0.0000</td>
<td>0.7408</td>
<td>0.0000</td>
<td>16.7794</td>
<td>0.3149</td>
<td>0.1150</td>
<td>0.1641</td>
<td>0.1972</td>
<td>1.1318</td>
</tr>
<tr>
<td>HI$_{Positive}$</td>
<td>0.0000</td>
<td>0.8624</td>
<td>0.0000</td>
<td>19.7716</td>
<td>0.4059</td>
<td>0.1604</td>
<td>0.2271</td>
<td>0.2840</td>
<td>0.1698</td>
</tr>
<tr>
<td>EX$_{Positive}$</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>20.0000</td>
<td>0.4579</td>
<td>0.1584</td>
<td>0.0716</td>
<td>0.2800</td>
<td>11.9190</td>
</tr>
<tr>
<td>MU$_{Positive}$</td>
<td>0.0000</td>
<td>0.9905</td>
<td>0.0000</td>
<td>20.0000</td>
<td>0.5585</td>
<td>0.1679</td>
<td>0.3855</td>
<td>0.3640</td>
<td>-5.0631</td>
</tr>
</tbody>
</table>

The exchangeable cross-correlation model uses the same parameter value to capture the relationship between levels even if there exist positive and negative relationships among levels. In this example, this restriction results in all the cross-correlations estimated to be zero. After allowing the full range of the cross correlation parameters for both the CC method and the HP method, the prediction accuracies are improved dramatically for the 2$^{nd}$ and 3$^{rd}$ curves. This result confirms our settings and assumptions that Curve 2 and Curve 3 are negatively correlated, and that they can “borrow” the negative information of each other and improve the prediction accuracy from the negative cross-correlation.

Although allowing negative cross-correlation ranges can help to capture the negative relationship between curves, the simplified computations in other candidate models lead to a decent fit for positive cross-correlation ranges. Instead of estimating negative cross-correlation, $HP_{Positive}$, $PI_{Positive}$, $HI_{Positive}$, and $MU_{Positive}$ estimate $\tau_{12}$ and $\tau_{23}$ as small, which implies the model will predict level 2 separately from the other two levels. The two indicator-based models (PI and HI) not only avoid computational complexity but also lead to a good prediction accuracy. Therefore,
Figure 3.4: A comparison of the Example 3.4.1 with the true curves for the different ranges of the candidate models. The left-top panel is the plot for $Y(X, 1)$, the right-top one is the plot for $Y(X, 2)$, and the bottom one is the plot for $Y(X, 3)$.

The two indicator-based models improve the prediction accuracy among levels even though there exist negative relationships among levels.
Chapter 4: Experimental Designs for Computer Experiments with Both Quantitative and Qualitative Inputs

In Section 2.2, we introduced four types of experimental designs for computer experiments with both quantitative and qualitative input variables. Among these four types of designs, NSFD and NLHD can only be applied to the special setting in which the qualitative variable indicates the accuracy of the code, while SLHD and CSLHD can be applied to the general case involving both qualitative and quantitative input variables.

In this chapter, we will first compare the difference between the SLHD and CSLHD in an illustrative example and propose a simplified algorithm of constructing the CSLHD. Secondly, we will propose two new types of experimental designs for computer experiments with mixed input variables. Then, we will use examples to compare the effect of different types of initial designs on the prediction accuracy for computer experiments with both quantitative and qualitative input variables. Finally, we will develop a frequentist-based sequential ANOVA algorithm for the design of computer experiments with both quantitative and qualitative input variables.
4.1 Comparison of SLHD and CSLHD

Huang et al. (2014) proposed the CSLHD and demonstrated the difference between the SLHD and CSLHD. As an illustrative example, Figure 4.1 shows the comparison between a SLHD and CSLHD with \( n = 20, d = 2, \) and \( m = 4 \). The points with the same color are points within the same slice, and hence have the same level of qualitative input. It is worth noting that the points in SLHD are spread out over the whole design region without any clustering of points. In contrast, the CSLHD clusters the points from different slices, which is thought to allow one to better capture the correlations among different qualitative levels.

![Figure 4.1: A comparison of SLHD and CSLHD with n=20, d=2, m=4](image_url)

For easy application of CSLHD, we can simplify the algorithm for constructing a CSLHD based on regular LHDs which can be generated by many software packages, such as R, Matlab, etc. Similar to the notation in Section 2.2.4, we let \( t = n/m \), which is 5 in Figure 4.1. The simplified algorithm is as follows:
i Generate a $t \times d$ regular LHD, denoted by $D_0 = (d_{ij}^0), i = 1, \cdots , t, j = 1, \cdots , d$.

ii Evenly divide the design region $[0, 1)^d$ into $t^d$ subregions $R_1, \cdots , R_{t^d}$. The $R_j$’s are the cubic regions obtained by dividing the interval $[0, 1)$ for each coordinate into $t$ equally spaced intervals and then taking Cartesian products. For $i = 1, \cdots , t, j = 1, \cdots , d$, if $d_{ij}^0 \in R_c$, generate an $m \times d$ LHD within region $R_c$, denoted by $D_i = (d_{kj}^{(i)}), i = 1, \cdots , t, k = 1, \cdots , m, j = 1, \cdots , d$.

iii Combine the $k^{th}$ row of each $D_i$, row by row, to obtain a $t \times d$ design matrix $D^{(k)} = (d_{kj}^{(i)}), i = 1, \cdots , t, j = 1, \cdots , d$, with $k$ fixed from $1, \cdots , m$. $D^{(k)}$ is the $k^{th}$ slice of the CSLHD.

This simplified algorithm allows one to use software for generating LHDs to achieve the goal of clustering and slicing at the same time. In addition, the built-in uniform random variable adjustment for generating LHDs in MATLAB guarantees the randomness of the points within each slice as well as the points across different slices. This randomness allows the CSLHD to sample everywhere on the input space, not just the middle point of each cubic subregion.

Furthermore, in order to obtain the OCSLHD, one can generate a large number of CSLHDs using this algorithm, and apply the centered $L_2$-discrepancy criterion to these CSLHDs. The CSLHD with the minimum value of the centered $L_2$-discrepancy criterion is the OCSLHD. An alternative criterion for choosing the optimal design is the maximin criterion (Morris and Mitchell (1995)). One can choose a design among those generated for which the minimum distance between points for each qualitative level greater than some threshold. The maximin criterion can guarantee the design
points for each qualitative level are far away from each other and hence guarantee a space-filling property.

4.2 Partial SLHD and Partial CSLHD

SLHD and CSLHD have intuitively appealing properties for experimental designs involving both quantitative and qualitative inputs. Not only is the overall design a LHD, but also the design for each qualitative level is a LHD. Moreover, as mentioned in Huang et al. (2014), for CSLHDs, the clustering guarantees that we observe each of the response surfaces determined by each level of the qualitative variable at similar points. This is thought to help one better capture the cross-correlation and hence produce better predictions. One question is whether such clustering does indeed produce better predictions. Is it best to observe each response surface at essentially the same points (as CSLHDs do), is it better to observe each response surface at very different points (as SLHDs do), or some compromise? To explore this further, we introduce two new types of designs: partial SLHD and partial CSLHD.

Suppose we wish to generate an experimental design on \([0,1)^d\) with \(n\) design points and \(m\) levels qualitative inputs. Assume \(n\) is a multiple of \(m\), for each level of qualitative input, we wish to observe \(t = n/m\) values of the quantitative inputs. For each qualitative input, we further require that \(t_1\) of these \(t\) values of the quantitative inputs are identical. In other words, we wish to observe \(t_1\) identical points on the response surface determined by different values of the qualitative inputs. We want the remaining \(t_2\) values for each qualitative input to differ. In particular, these will correspond to different slices of an SLHD. Figure 4.2 shows a simple example of such a design with \(n = 20, d = 1, m = 4,\) and \(t_1 = 2\).
To generate such a design, which we call a partial SLHD, do the following:

1. Generate a SLHD $D_S$ of size $n$ with $d$ quantitative inputs and $m$ levels of qualitative inputs.

2. Generate a LHD $D_C$ of size $t$ with $d$ quantitative inputs.

3. Randomly select $t_1$ input points from the design $D_C$, and denote these by $D_{C_{\text{common}}}$.
4 Evenly divide the design region $[0,1]^d$ into $t^d$ subregions $R_1, \cdots, R_{t^d}$ as we did for our simple algorithm for generating a CSLHD. For each level of the qualitative input in $D_S$, exclude the input points in the same subregions as the points in $D_{Common}$ randomly. Sample $t_2$ input points from the remaining points on $D_S$ for this level of the qualitative input. Denote these $t_2$ points by $D_{Sliced,l}$, where $l$ is the level of the qualitative variable.

5 For each qualitative level $l$, combine $D_{Sliced,l}$ with $D_{Common}$. The union of all these is the partial SLHD, denoted $D_{partialS}$, for all the qualitative inputs.

6 Step 1-5 provide a way to generate a partial SLHD. To generate an optimal partial SLHD, we can repeat 1-5 until the stopping rule is met. An appropriate stopping rule could be choosing to stop when the minimum distances among quantitative inputs are all greater than $1/t$ and the minimum distance among $D_{Common}$ is greater than $1/t_1$.

The partial SLHD has the following properties:

1 For each qualitative level $l$, $D_{Sliced,l}$ is a size $t_2$ subdesign of a LHD with $t$ levels, $D_{Common}$ is a size $t_1$ subdesign of a LHD with $t$ levels. For all $l$, $D_{Sliced,l}$ and $D_{Common}$ share no common subregion $R_i, i = 1, \cdots, t^d$.

2 The set of quantitative input points in the union of all $D_{Sliced,l}$ is a size $t_2 \times m$ subdesign of a LHD with $n$ levels.

As an example, Figure 4.3 displays the procedure for generating a partial SLHD with $n = 40, d = 2, m = 4$ and $t_1 = 3$. The maximin selection from step 6 guarantees the space-filling property of each qualitative level.
Figure 4.3: A partial SLHD with \( n = 40, d = 2, m = 4, t_1 = 3 \): the top four panels are the designs sampled from a SLHD, the bottom left panel is the common design for each qualitative level, the bottom right panel is the partial SLHD including all the qualitative levels.
Similarly, an algorithm for generating a design with a mixture of common points and clustered points, which we call partial CSLHD, is as follows:

(i) Generate a $t \times d$ regular LHD, denoted by $D_0 = (d_{ij}^0), i = 1, \cdots, t, j = 1, \cdots, d$.

(ii) Randomly select $t_1$ input points from $D_0$ and denote these by $D_{Common}$. The remaining input points in $D_0$ are denoted by $D_R = (d_{ij}^R), i = 1, \cdots, t_2, j = 1, \cdots, d$.

(iii) Evenly divide the design region $[0,1)^d$ into $t^d$ subregions $R_1, \cdots, R_{t^d}$. For $i = 1, \cdots, t_2, j = 1, \cdots, d$, if $d_{ij}^R \in R_c$, generate an $m \times d$ LHD within region $R_c$, denoted by $D_i = (d_{kj}^{(i)}), i = 1, \cdots, t$.

(iv) As in the algorithm for constructing a CSLHD, combine the $k^{th}$ row of each $D_i$, row by row, to obtain a $t_2 \times d$ design matrix $D^{(k)}, k = 1, \cdots, m$. Thus, $D^{(k)}$ is the $k^{th}$ slice of the CSLHD.

(v) The design for the $k^{th}$ qualitative input ($k = 1, \cdots, m$) is $D^{(k)} \cup D_{Common}$, which we denote by $D^{(k)}_{PartialC}$. The union of the $D^{(k)}_{PartialC}$ is our partial CSLHD, denoted by $D_{PartialC}$.

(vi) To derive an optimal partial CSLHD, repeat (i)-(v) until some stopping rule is met. An appropriate stopping rule could be choosing to stop when the minimum distances among $D_0$ are all greater than $1/t$ and the minimum distance among $D_{Common}$ is greater than $1/t_1$.

Besides the properties of the regular partial SLHD, the partial CSLHD has two more properties:

(i) For each qualitative level $k$ ($k = 1, \cdots, m$), $D^{(k)}_{PartialC}$ is a LHD with $t$ levels.
(ii) For $i = 1, \cdots, t$, let $H^{(i)}_{\text{PartialC}}$ be the $i^{th}$ subarray of $D_{\text{PartialC}}$ consisting of all the $i^{th}$ rows of $D^{(1)}_{\text{PartialC}}, \cdots, D^{(m)}_{\text{PartialC}}$, and let $M_i$ be the maximum inter-point distance for $H^{(i)}_{\text{PartialC}}$. Then, $M = \max_{1 \leq i \leq t} M_i \leq (1 - m^{-1}) \sqrt{d/t}$.

Figure 4.4 shows the procedure of generating a partial CSLHD with $n = 40$, $d = 2$, $m = 4$ and $t_1 = 3$. It is clearly that the design points for each qualitative level is also an LHD and hence guarantees the space-filling property.

4.3 The Effect of the Initial Designs on the Model Fitting

Section 2.2 introduced four types of experimental designs that can be used for computer experiments with both quantitative and qualitative inputs. We will focus on the experimental designs involving the general qualitative inputs. In this section, we will use simulation examples to illustrate the effect of the initial designs on the model fit. There are two properties of the initial design that could affect the prediction accuracy: one is the total number of data points (sample size), the other is the type of design. Therefore, we vary the sample size and design type in our simulated experiments. In these simulations, the number of runs is the same for each qualitative level, reflecting a lack of prior information about the need to sample some qualitative levels more heavily than others. Following our previous notation, we let $t$ be the number of LHD runs per qualitative level and $m$ be the number of qualitative levels (or combinations of levels). The total number of points is $n = tm$. We consider $t = 10, 20, 40$, and the experimental design for each training sample size is constructed in seven different ways.
Figure 4.4: A partial CSLHD with $n = 40, d = 2, m = 4, t_1 = 3$: the top four panels are the designs sampled from a CSLHD, the bottom left panel is the common design for each qualitative level, the bottom right panel is the partial CSLHD including all the qualitative levels.
**Standard Latin Hypercube** In this approach, one Latin hypercube design of size $n = tm$ is generated over the quantitative sample space. It is then randomly split into $m$ groups of $t$ runs, and each group is assigned a qualitative level.

**k LHD** In this approach, a separate Latin hypercube design is generated for each given qualitative level. That is, we generate $m$ independent Latin hypercube designs, each of size $t$ and corresponding to one qualitative level.

**same LHD** In this approach, a single Latin hypercube design is generated and this same LHD is used for each given qualitative level.

**SLHD** This approach is based on Section 2.2.3. We generate a sliced Latin hypercube design with $m$ slices, where each slice of $t$ runs corresponds to one qualitative level.

**CSLHD** This approach is based on Section 2.2.4. We generate a clustered sliced Latin hypercube design with $m$ slices, where each slice of $t$ runs corresponds to one qualitative level.

**Partial Sliced Latin Hypercube** We generate a sliced Latin hypercube design with $m$ slices, where each slice of $t$ runs corresponds to one qualitative level. In our examples, the size of the common design is set to be $t_1 = \lceil t/4 \rceil$.

**Partial Clustered Sliced Latin Hypercube** We generate a partial clustered sliced Latin hypercube design with $m$ slices, where each slice of $t$ runs corresponds to one qualitative level. In our examples, the size of the common design is set to be $t_1 = \lceil t/4 \rceil$. 
We will consider the situation in which there are two quantitative inputs and one qualitative input. We use the hypersphere parameterization method in Section 2.1.2 ((2.1.10) and (2.1.11) by Zhou et al. (2011)) to fit the GaSP model. For a “test” sample of $51 \times 51$ equally spaced points of the sample space $[0, 1]^2$, the overall RMSE is calculated to assess the prediction accuracy. For each selected sample size and experimental design method, we repeat the procedure of data generation, modeling fitting, and assessment of prediction accuracy 50 times. For each repetition of the data generation, a different initial design will be generated and hence will produce different prediction accuracy.

Because the designs may perform differently in the sense of the prediction accuracy, for different true response surfaces, we generate several examples to compare the performance of these types of designs.

### 4.3.1 Simple 2D Functions

**Example 4.3.1.** We generate four similar 2D functions from a combination of quadratic terms and the sinusoidal terms. The input space is $[0, 1]^2$. The true surfaces are given by

$$
y = \begin{cases} 
x_1^2 + 2 \sin(2\pi x_1 x_2) + x_2^2 & z_1 = 1 \\
x_1^2 + 10 \sin(2\pi x_1 x_2) + x_2^2 & z_1 = 2 \\
10x_1^2 + 2 \sin(2\pi x_1 x_2) + x_2^2 & z_1 = 3 \\
x_1^2 + 2 \sin(2\pi x_1 x_2) + 12x_2^2 & z_1 = 4 
\end{cases}
$$

Figure 4.5 shows the true surfaces. The four surfaces have similar quadratic and sinusoidal patterns, but varying in the scales.
To give an idea of the cross-correlations, for one of the runs from partial CSLHD with sample size $t=40$, the estimated cross-correlation matrix is:

$$
\begin{pmatrix}
1.0000 & 0.1889 & -0.0578 & -0.0961 \\
0.1889 & 1.0000 & 0.0659 & 0.0151 \\
-0.0578 & 0.0659 & 1.0000 & 0.0793 \\
-0.0961 & 0.0151 & 0.0793 & 1.0000
\end{pmatrix}
$$

The estimated values of the $\hat{\tau}_{ij}$'s are all relatively small, which represent the relatively small similarity between surfaces. Figure 4.6 compares the RMSEs of the seven design methods with three different sample sizes. When the sample size is small ($t=10$), the CSLHD method, the same LHD method and the partial CSLHD method
perform well among these methods. All three emphasize having the same or similar values of the quantitative inputs for all 4 cases. They also perform well for \( t = 20 \). The standard LHD method performs poorly for all the three different sample sizes. The differences between the design methods are trivial for large sample size \( (t = 40) \).

Figure 4.6: The boxplot comparison of the simple 2D functions RMSEs. The left 7 are for \( t = 10 \), the middle 7 are for \( t = 20 \), and the right 7 are for \( t = 40 \). ‘ST’ means the standard LHD, ‘SP’ means the \( k \) LHD, ‘SA’ means the same LHD, ‘SL’ means the SLHD, ‘CS’ means the CSLHD, ‘PS’ means the partial SLHD, and ‘PC’ means the partial CSLHD

4.3.2 Branin Function

Example 4.3.2. Branin Function In this example, we consider the Branin function (Törn and Zilinskas (1989)), one of the most popular test functions in the computer experiments literature. We use the Branin function as the true surface of the 1\textsuperscript{st} qualitative level and modify it to create the surfaces for the other 3 qualitative levels.
The true functions of the four level surfaces are given by

\[ y = \begin{cases} 
(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos(x_1) + 10 & z_1 = 1 \\
(x_2 - \frac{5.1}{4\pi^2} x_1^2 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos(x_1) + 10 & z_1 = 2 \\
(x_2 + \frac{5}{\pi} x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos(x_1) + 10 & z_1 = 3 \\
(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 70 & z_1 = 4 
\end{cases} \]

The original input range of the Branin function is taken from \([-5, 10] \times [0, 15]\). We rescale the original range into the unit range \([0, 1]^2\) and display the true surfaces in Figure 4.7.
To give an idea of the cross-correlations, for one of the runs from partial CSLHD with sample size \( t = 40 \), the estimated cross-correlation matrix is:

\[
\begin{pmatrix}
1.0000 & -0.0324 & 0.2295 & 0.2906 \\
-0.0324 & 1.0000 & -0.0516 & -0.1770 \\
0.2295 & -0.0516 & 1.0000 & 0.3354 \\
0.2906 & -0.1770 & 0.3354 & 1.0000
\end{pmatrix}
\]

The estimated values of the \( \hat{\tau}_{ij} \)'s are not large, but several are of moderate size, which means there exist certain similarities between surfaces. The second surface seems to be “different” from the others (small and negative cross-correlations). Compared to the other surfaces, the 3\(^{rd}\) surface has a larger response range. Therefore, a “good” design for this example should help to distinguish the 3\(^{rd}\) surface from others.

In addition, without the same or clustered design points for the different qualitative levels, the sinusoidal pattern in Branin function should make it difficult to capture the cross-correlation information, if, as some authors have suggested, duplicate or near duplicate points are needed to do this. Figure 4.8 compares the overall RMSEs of the testing sample for the different choice of samples size and experimental design methods. When \( t = 10 \), with the help of the same design points or nearby design points among different qualitative levels, the CSLHD method and the partial CSLHD method successfully distinguish the 3\(^{rd}\) surface from the others (the cross-correlation matrix estimate is close to the case when \( t = 40 \)) and hence produce better prediction accuracies with the partial CSLHD performing best. The partial CSLHD performs well for all sample sizes, but as sample size increases, differences between methods become small (presumably with a sample size of 40, any reasonably space filling design for a given value of the qualitative variable, produces good fit). Interestingly, the k-LHD and the SLHD-based methods perform well and outperform the CSLHD for \( t = 20 \).
Figure 4.8: The boxplot comparison of the Branin functions’ RMSEs. The left 7 are for \( t = 10 \), the middle 7 are for \( t = 20 \), and the right 7 are for \( t = 40 \). ‘ST’ means the standard LHD, ‘SP’ means the \( k \) LHD, ‘SA’ means the same LHD, ‘SL’ means the SLHD, ‘CS’ means the CSLHD, ‘PS’ means the partial SLHD, and ‘PC’ means the partial CSLHD.

### 4.3.3 Goldstein Price Function

**Example 4.3.3.** The Goldstein-Price function (Törn and Zilinskas (1989)) also has two quantitative inputs. We let the 1\(^{st}\) level of the qualitative input be the Goldstein-Price function, and create the other three levels by making some adjustments to the 1\(^{st}\) level. The true functions for the four qualitative levels are
\[ y(x_1, x_2, 1) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \]
\[ \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \]
\[ y(x_1, x_2, 2) = [1 + (x_1 + x_2 + 1)^2(10 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \]
\[ \times [50 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \]
\[ y(x_1, x_2, 3) = [1 + (x_1 + x_2 + 1)^2(5 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \]
\[ \times [70 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \]
\[ y(x_1, x_2, 4) = [1 + (x_1 + x_2 + 1)^2(1 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \]
\[ \times [100 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \]

The original input range of the Goldstein Price function is taken from \([-2, 2] \times [-2, 2]\). We rescale the original range into the unit range \([0, 1]^2\) and display the true surfaces in Figure 4.9.

To give an idea of the cross-correlations, for one of the runs from partial CSLHD with sample size \(t = 40\), the estimated cross-correlation matrix is:

\[
\begin{pmatrix}
1.0000 & 0.5217 & 0.2463 & 0.0621 \\
0.5217 & 1.0000 & 0.7205 & 0.5807 \\
0.2463 & 0.7205 & 1.0000 & 0.8693 \\
0.0621 & 0.5807 & 0.8693 & 1.0000
\end{pmatrix}
\]

All the four surfaces have the similar shape, only varying in scales. The cross-correlation matrix estimate for \(t = 40\) demonstrate this fact with relatively large values of \(\hat{\tau}_{12}, \hat{\tau}_{23}, \hat{\tau}_{24}\) and \(\hat{\tau}_{34}\). We expect designs lacking in duplicate or clustered points to do a reasonable job of capturing the cross-correlation in this example because of the surfaces appear very similar except at two corners. Designs that do not use all the same (or nearly the same) points for each surface may perform well because they
allow one to “borrow” information from other curves about unobserved regions, but at the same time allow us to estimate cross-correlations reasonably well. Indeed, as shown in Figure 4.10, the $k$-LHD method and the partial SLHD method perform slightly better than the others for the small sample size ($t = 10$) with the partial SLHD performing best. Note that for $t = 20$, the CSLHD performs the worst. As sample size increases, the distinction among different methods becomes trivial. The scale for the original Goldstein Price function is on the order of $10^6$, which explains the large scale of the RMSEs ($10^5$).
Figure 4.10: The boxplot comparison of the Goldstein-Price functions RMSEs. The left 7 are for $t = 10$, the middle 7 are for $t = 20$, and the right 7 are for $t = 40$. ‘ST’ means the standard LHD, ‘SP’ means the $k$ LHD, ‘SA’ means the same LHD, ‘SL’ means the SLHD, ‘CS’ means the CSLHD, ‘PS’ means the partial SLHD, and ‘PC’ means the partial CSLHD.

4.3.4 Branin and Goldstein Price Functions

Example 4.3.4. In this example, we generate two similar surfaces from the Branin function and two similar surfaces from the Goldstein Price function. Due to the large scale of the Goldstein Price function, the two functions generated from the Goldstein Price function are rescaled by $\frac{1}{10^4}$ to match the scale for the Branin function. The
true functions for the four surfaces are given by

\[
y(x_1, x_2, 1) = \frac{1}{10^4}[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]
\times[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]
\]

\[
y(x_1, x_2, 2) = \frac{1}{10^4}[1 + (x_1 + x_2 + 1)^2(18 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]
\times[50 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]
\]

\[
y(x_1, x_2, 3) = \left( x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10(1 - \frac{1}{8\pi}) \cos(x_1) + 10
\]

\[
y(x_1, x_2, 4) = \left( x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{4}{\pi}x_1 - 6 \right)^2 + 10(1 - \frac{1}{8\pi}) \cos(x_1) + 10
\]

The original input range of the Branin function is \([-5, 10] \times [0, 15]\), and the original input range of the Goldstein Price function is \([-2, 2] \times [-2, 2]\). We rescale these ranges into the unit range \([0, 1]^2\) and display the true surfaces in Figure 4.11.

To give an idea of the cross-correlations, for one of the runs from partial CSLHD with sample size \(t = 40\), the estimated cross-correlation matrix is:

\[
\begin{pmatrix}
1.0000 & 0.9179 & -0.1379 & 0.0542 \\
0.9179 & 1.0000 & -0.1455 & 0.0766 \\
-0.1379 & -0.1455 & 1.0000 & 0.7049 \\
0.0542 & 0.0766 & 0.7049 & 1.0000
\end{pmatrix}
\]

The similarity between the two Branin surfaces (\(\hat{\tau}_{34} = 0.7049\)) and the two Goldstein-Price surfaces (\(\hat{\tau}_{12} = 0.9179\)) and differences between the Branin and Goldstein-Price surfaces (\(\hat{\tau}_{13} = -0.1379, \hat{\tau}_{14} = 0.0542, \hat{\tau}_{23} = -0.1455, \hat{\tau}_{24} = 0.0766\)), would seem to require a “good” design not only to spread among the whole sample space but also to recognize the correct “similar” surfaces. Figure 4.12 shows the comparison of the RMSEs for different methods. The k-LHD method is the best for each sample size. The SLHD, the standard LHD, and the partial SLHD methods generally are the next best across sample sizes. When the sample size is small (\(t = 10\), the
differences between methods are trivial. When sample size is large ($t = 20, 40$), the k-LHD, the standard LHD, the SLHD, and the partial SLHD methods stand out among all the design methods. The large sample size forces these four designs to produce somewhat clustered design points among different surfaces. In addition, considered as a whole, these three designs are more spread out over the sample space for the quantitative inputs and hence (presumably by borrowing information from the curve that is similar) produce better predictions for each surfaces. In this case, where not
all curves are similar, a high degree of “cluster” or use of the same points does not appear to help.

Figure 4.12: The boxplot comparison of the Branin and Goldstein-Price functions RMSEs. The left 7 are for \( t = 10 \), the middle 7 are for \( t = 20 \), and the right 7 are for \( t = 40 \). ‘ST’ means the standard LHD, ‘SP’ means the \( k \)-LHD, ‘SA’ means the same LHD, ‘SL’ means the SLHD, ‘CS’ means the CSLHD, ‘PS’ means the partial SLHD, and ‘PC’ means the partial CSLHD

4.3.5 Summary

From the above examples, it can be noticed that the results vary from example to example. We summarize our findings as follows:

1 Not surprisingly, larger sample size does help the model to predict more accurately, regardless of design.

2 In contrast to the argument about the advantage of the clustering and duplication, clustering and duplication do not always produce the best design.

3 For the flat surfaces with similar patterns among all the qualitative levels, the \( k \)-LHD, the SLHD and the partial SLHD can produce better predictions.
4 For the rough surfaces with small or medium cross-correlation values, the CSLHD and partial CSLHD can produce better predictions.

5 When there exist similarities and differences among surfaces with different qualitative levels, the performance for different methods are similar for a small sample size. As the sample size increases, the performance of the k-LHD, the SLHD and the partial SLHD outperform other methods.

6 Without knowing the patterns or similarities of the surfaces, it’s not clear which design is the best. However, one can start with a partial SLHD or a partial CSLHD, which performs well in most of our examples. Furthermore, a set of a sequential design points can be added if the initial design doesn’t produce a good fit.

4.4 Sequential ANOVA Kriging Algorithm for Computer Experiments with Quantitative and Qualitative Factors

Considering the limited number of runs possible in many computer experiments, a small size initial design for a complicated underlying response surface might lead to poor approximations of the output. In order to improve the prediction, design points can be added sequentially. Several such strategies for computer experiments only involving quantitative inputs have been proposed in the literature. These depend on the goal of the computer experiment, and include sequential criterion-based optimal designs, sequential designs for model fit of response surfaces, and sequential designs for global optimization. Especially for the purpose of global optimization of the response surface, the use of expected improvement criteria has gained much attention. Jones, Schonlau and Welch (1998) first developed an expected improvement criterion for
global optimization of the response surface. Ranjan, Bingham and Michailidis (2008) extended the use of expected improvement to estimate the contour of a computer model. Lam and Notz (2008) modified the expected improvement criterion to achieve a good global fit of the model.

However, in the literature on computer experiments with both quantitative and qualitative factors, there is little on sequential design. The sequential design strategies described above do not involve the qualitative input and hence can not be applied directly to computer experiments with mixed factors. The goal of this chapter is to predict a target response (TR) at a certain level of qualitative factors based on the remaining ones. We call the responses at the other levels of the qualitative factors auxiliary responses (ARs). Considering that the TR might only be sampled over a certain limited region with limited runs, our prediction will include both interpolation and extrapolation of the TR. Motivated by the ANOVA kriging model (Han et al. 2013), “useless” ARs should be identified and not be used to predict the TR. Thus, the idea of the algorithm is to select and sample points on the best AR that could help to predict the TR sequentially.

4.4.1 Definitions and Notations

In this section, we assume the ARs all have been observed over the quantitative input region $R = [0,1]^d$, while the TR has been observed only over the limited quantitative input region $R_{inter} = [a,1]^d (0 < a < 1)$. Thus, the predictions within the region $R_{inter}$ are interpolations, and the predictions within the region $R_{extra} = [0,1]^d/[a,1]^d$ are extrapolations. We use the same notation as in Chapter 3. For multiple qualitative factors with $m_1, \cdots, m_K$ levels of each factor, we assume $m =$
\[ \prod_{j=1}^{K} m_j \]. Furthermore, we denote the qualitative levels for the ARs by \( 1, \ldots, m-1 \), and the one for the TR by \( m \).

For the initial designs for each level of the qualitative inputs for the ARs, we first take \( m-1 \) independent \( LHD(t, d) \)s on the whole region \( R \). Then, because of the limited region for the TR, we take a restricted \( LHD(t_0, d) \) on the region \( R_{\text{inter}} \), where \( t_0 \leq t \).

Suppose \( X = \{ x_{j}^{(0)} | j = 1, \ldots, N \} \) is the set of quantitative inputs used as test data, with the first \( N_0 \) in the region \( R_{\text{inter}} \) and the rest in the region \( R_{\text{extra}} \). The BLUP for each AR and TR at any \( x_{j}^{(0)} \) in the test set based on the initial design, is \( \hat{y}(x_{j}^{(0)}, i), \ldots, \hat{y}(x_{j}^{(0)}, m-1), \hat{y}(x_{j}^{(0)}, m) \), where the BLUPs for \( j = 1, \ldots, N_0 \) correspond to interpolations and the BLUPs for \( j = N_0 + 1, \ldots, N \) correspond to extrapolations.

Motivated by the average predictor in ANOVA kriging (Section 2.1.4; Han et al. (2013)), for \( i = 1, \ldots, m-1 \), we define the average interpolation-effect predictor for each AR by

\[
\hat{A}^{(i)}_{\text{inter}}(x) = \frac{1}{2} \{ \hat{y}(x, i) - E(\hat{y}(x, i)I\{R_{\text{inter}}\}) + \hat{y}(x, m) - E(\hat{y}(x, m)I\{R_{\text{inter}}\}) \}, x \in R_{\text{inter}},
\]

where \( I\{R_{\text{inter}}\} \) is the indicator function of the region \( R_{\text{inter}} \).

Thus, we can derive the interpolation sum of squares for each AR, defined by

\[
S^{(i)}_{\text{inter}} = \sum_{j=1}^{N_0} \{ (\hat{y}(x_{j}^{(0)}, i) - E(\hat{y}(x_{j}^{(0)}, i)I\{R_{\text{inter}}\}) - \hat{A}^{(i)}_{\text{inter}}(x_{j}^{(0)}))^2 + (\hat{y}(x_{j}^{(0)}, m) - E(\hat{y}(x_{j}^{(0)}, m)I\{R_{\text{inter}}\}) - \hat{A}^{(m)}_{\text{inter}}(x_{j}^{(0)}))^2 \} \]

(4.4.2)
Notice that in (4.4.1), when the $i^{th}$ AR is similar to the TR over $\mathbf{x} \in \mathbf{R}_{\text{inter}}$, $\hat{A}_{\text{inter}}^{(i)}(\mathbf{x}_j^{(0)})$ is close to $\hat{y}(\mathbf{x}_j^{(0)}, m) - E(\hat{y}(\mathbf{x}_j^{(0)}, m)I\{\mathbf{R}_{\text{inter}}\})$. Furthermore, both $\hat{A}_{\text{inter}}^{(i)}(\mathbf{x}_j^{(0)})$ and $\hat{y}(\mathbf{x}_j^{(0)}, i) - E(\hat{y}(\mathbf{x}_j^{(0)}, i)I\{\mathbf{R}_{\text{inter}}\})$ are close to the “error” term in TR, $\hat{y}(\mathbf{x}_j^{(0)}, m) - E(\hat{y}(\mathbf{x}_j^{(0)}, m)I\{\mathbf{R}_{\text{inter}}\})$, and hence $S_{\text{inter}}^{(i)}$ is small.

Among all the $S_{\text{inter}}^{(i)}$’s, we find the minimum, denoted by $S_{\text{inter}}^{(*)}$. We take the $i^{*th}$ AR as the most suitable AR for predicting the TR over $\mathbf{R}_{\text{inter}}$. Then, we can apply ANOVA main effect kriging on the $i^{*th}$ AR and the TR.

Similarly, for $i = 1 \cdots, m - 1$, we can also define the average extrapolation-effect predictor for the $i^{th}$ AR

$$\hat{A}_{\text{extra}}^{(i)}(\mathbf{x}) = \frac{1}{2}\{\hat{y}(\mathbf{x}, i) - E(\hat{y}(\mathbf{x}, i)I\{\mathbf{R}_{\text{extra}}\}) + \hat{y}(\mathbf{x}, m) - E(\hat{y}(\mathbf{x}, m)I\{\mathbf{R}_{\text{extra}}\})\}, \mathbf{x} \in \mathbf{R}_{\text{extra}}$$

(4.4.3)

and the extrapolation sum of squares for the $i^{th}$ AR

$$S_{\text{extra}}^{(i)} = \sum_{j=N_{0}+1}^{N} \{(\hat{y}(\mathbf{x}_j^{(0)}, i) - E(\hat{y}(\mathbf{x}_j^{(0)}, i)I\{\mathbf{R}_{\text{extra}}\}) - A_{\text{extra}}^{(i)}(\mathbf{x}_j^{(0)}))^2 + (\hat{y}(\mathbf{x}_j^{(0)}, m) - E(\hat{y}(\mathbf{x}_j^{(0)}, m)I\{\mathbf{R}_{\text{extra}}\}) - A_{\text{extra}}^{(m)}(\mathbf{x}_j^{(0)}))^2\}$$

(4.4.4)

Additionally, we define the extrapolation weight $k_i, i = 1, \cdots, m - 1$ by

$$k_i = \frac{\frac{1}{S_{\text{extra}}^{(i)}}}{\sum_{j=1}^{m-1} \frac{1}{S_{\text{extra}}^{(j)}}}, i = 1, \cdots, m - 1$$

(4.4.5)

$k_i$ is a measure of extrapolation strength for each AR. The idea is to use weighted combinations of all the ARs to extrapolate the TR.

Suppose $\mathbf{x}_j^{(i)}$ is the $j^{th}$ quantitative point in the current design for the response at $i^{th}$ level, where $j = 1, \cdots, t_i, i = 1, \cdots, m$. We also define the minimax correlation
for all the responses at each level by

$$r_0^{(i)} = \min_{x^{(0)} \in R} \max_{1 \leq j \leq t_i} \text{corr}(x_j^{(i)}, x^{(0)}), i = 1, \cdots, m - 1$$

$$x_{j^*}^{(i)} = \arg\min_{x^{(0)} \in R} \max_{1 \leq j \leq t_i} \text{corr}(x_j^{(i)}, x^{(0)}), i = 1, \cdots, m - 1$$

$$r_0^{(m)} = \min_{x^{(0)} \in R_{\text{inter}}} \max_{1 \leq j \leq t_m} \text{corr}(x_j^{(m)}, x^{(0)})$$

$$x_{j^*}^{(m)} = \arg\min_{x^{(0)} \in R_{\text{inter}}} \max_{1 \leq j \leq t_m} \text{corr}(x_j^{(m)}, x^{(0)})$$  (4.4.6)

Let $k = \arg\min_{1 \leq i \leq m} r_0^{(i)}$ be the level of the qualitative inputs with the minimum minimax correlation. $x_{j^*}^{(i)}$ is the point in $R$ least correlated with those points at which we have observed $y$ on curve $i$. Then $x_{j^*}^{(k)}$ is overall the point in $R$ least correlated with all the observations so far. Hence $x_{j^*}^{(k)}$ is the point for which our current data is least informative.

4.4.2 Sequential ANOVA Kriging Algorithm for Computer Experiments with Quantitative and Qualitative Inputs

Suppose we wish to add $K$ additional points to the initial design sequentially. From the initial design, we find the $i^{th}$ AR ($i^*$ may change after adding certain points) that is the most suitable AR to help predict the TR over $R_{\text{inter}}$. Then, we perform the following algorithm.

**Step 1:** Calculate the current minimum minimax correlation from the formula (4.4.6) and the corresponding qualitative level $k$. Add a point into the training data at $w^* = (x_{j^*}^{(k)}, k)^T$. Obtain the corresponding response value $y(x_{j^*}^{(k)}, k)$.

**Step 2:** Fit the BLUP for the new updated AR or TR.

**Step 3:** Check the stopping rule, if one exists. If additional points are needed, repeat from Step 1 onwards with the next point.
Step 4: Once one stops adding points, calculate the interpolation sum of squares for each AR and identify the minimum one $S_{\text{inter}}^{(i^*)}$. Observe the average interpolation effect function $\hat{y}_{\text{inter}}^A(x) = \frac{1}{2}(\hat{y}(x, i^*) + \hat{y}(x, m))$ and the deviation data set $y_{\text{inter}}^*(x, m) = y(x, m) - \hat{y}_{\text{inter}}^A(x)$. Fit the BLUP for the deviation data set $y_{\text{inter}}^*(x, m)$ and derive the predictor $\hat{y}_{\text{inter}}^*(x, m)$. Then, the ANOVA kriging interpolation predictor for the TR is

$$\hat{y}_{\text{inter}}^{AK}(x, m) = \hat{y}_{\text{inter}}^*(x, m) + \hat{y}_{\text{inter}}^A(x) \quad (4.4.7)$$

Step 5: Calculate the extrapolation sum of squares for each AR and derive the extrapolation weights $k_i, i = 1, \ldots, m - 1$. Let $\hat{y}_{\text{extra}}(x, m)$ be the current extrapolation predictor. Obtain the weighted average extrapolation effect function $\hat{y}_{\text{extra}}^A(x) = \frac{1}{2}(\sum_{i=1}^{m-1} k_i \hat{y}(x, i) + \hat{y}_{\text{extra}}(x, m))$ and the deviation data set $y_{\text{extra}}^*(x, m) = y(x, m) - \hat{y}_{\text{extra}}^A(x)$. Fit the BLUP for the deviation data set $y_{\text{extra}}^*(x, m)$ and derive the predictor $\hat{y}_{\text{extra}}^*(x, m)$. Then, the ANOVA kriging extrapolation predictor for the TR is

$$\hat{y}_{\text{extra}}^{AK}(x, m) = \hat{y}_{\text{extra}}^*(x, m) + \hat{y}_{\text{extra}}^A(x) \quad (4.4.8)$$

One possible stopping rule for the algorithm is to stop when $\min_{1 \leq i \leq m} r^{(i)}_0 \geq r^*$, where $r^*$ is a tolerance value for the minimax correlation. Another is to stop after $N^*$ points are added, where $N^*$ is the maximum number of observations that can be afforded.
4.4.3 Remarks and Notes

The sequential ANOVA kriging algorithm uses the minimum minimax correlation criterion to select and add points to the design. There are alternative ways of selecting the new points for the design.

**Sensitivity criterion:** When there are multiple quantitative inputs, sensitivity analysis is available for each AR and the TR. We can derive the main effect sensitivity of each AR and the TR, and calculate the sum of the absolute differences of the main effect sensitivities between each AR and the TR. The AR with the minimum value, denoted by $k^*$, implies the similarity between the AR and the TR. Thus, we can derive the $r_0^{(k^*)}$, $r_0^{(m)}$, $x_{j^*}^{(k^*)}$ and $x_{j^*}^{(m)}$ from the formula (4.4.6). Presumably, $x_{j^*}^{(k^*)}$ and $x_{j^*}^{(m)}$ are the two points from similar curves with least information. Repeatedly add one new point to the $k^{th}$ AR at $x_{j^*}^{(k^*)}$ and another new point to the TR at $x_{j^*}^{(m)}$.

**Absolute difference criterion:** Repeatedly add one point to each AR and TR. After adding one point to each response, we calculate the absolute difference between the new BLUP and the previous BLUP. The place with the maximum absolute difference is where we need to add the next point. Note that in this case, we need to add $m$ points at each iteration. That is because the maximum absolute difference will be zero if we don’t add a new point to that curve.

Considering the requirement of repeatedly adding points to the TR, the two criteria can be applied when there is no limitation on the number of points on the TR.
4.4.4 Simulation Examples

This section presents two examples to illustrate the sequential ANOVA kriging Algorithm for computer experiments with quantitative and qualitative inputs.

Example 4.4.1. Consider an experiment with one quantitative variables, \( x \in [0, 1] \), and one qualitative factor \( z \) with four levels. The true response functions of this experiment are

Response 1: \( y(x, 1) = x + 3x^2 \);

Response 2: \( y(x, 2) = x^2 + \cos\left(\frac{\pi}{2} x\right) \);

Response 3: \( y(x, 3) = 2x^2 + \sin(\pi x) \);

Response 4: \( y(x, 4) = 2x^2 + 0.5\cos(\pi x) \).

The first three are the ARs and the last one is the TR. The initial design for each AR is independently chosen to be a LHD with \( n = 5, d = 1 \) from the interval \([0, 1]\). The initial design for the TR is chosen to be a LHD with \( n = 3, d = 1 \) from the interval \([0.2, 1]\). Thus, the predictions for the interval \([0.2, 1]\) are interpolations, while the predictions for the interval \([0, 0.2]\) are extrapolations. The plot of the true responses with the initial designs is displayed in Figure 4.13. As we can seen in the plot, the TR (the light blue one) shares more similarity with the second response (the green one) compared with the others.

After sequentially adding 20 points to the model (see Figure 4.14 for the design), the minimum minimax correlation is 0.9858. The ANOVA kriging model selects the second AR as the interpolation response. The extrapolation weights of each ARs are 0.0716, 0.5286, and 0.3998, respectively. Notice that the algorithm adds more points
The reason is probably because curve 1 is close to the TR (curve 4) in the interpolation region, and curve 2 is close to the TR in the extrapolation region of the TR.

The testing data are 101 equally-spaced points in $[0,1]$, denoted by $x_1^*, x_2^*, \cdots, x_N^*$. Assume there are $n_0$ observations for the TR after 20 steps, denoted by $x_1, x_2, \cdots, x_{n_0}$. In order to assess the fit of the model, the cross validated prediction error (XVPE) is calculated by the formula

$$XVPE = \sqrt{\frac{1}{n_0} \sum_{i=1}^{n_0} (\hat{y}_{inter}^{AK(i)}(x_i, m) - \hat{y}_{inter}^{AK}(x_i, m))^2}$$  \hspace{1cm} (4.4.9)
where $\hat{y}_{\text{inter}}^{AK(i)}(x, m)$ is the ANOVA kriging predictor by deleting the observation $x_i$. The XVPE for this example is 0.2404, which is a reasonable value. Furthermore, in order to assess the predictor for interpolation and extrapolation, the RMSEs are derived for both situations from the formula:

$$RMSE = \sqrt{\frac{1}{N_{\text{pred}}} \sum_{i=1}^{N_{\text{pred}}} (y(x_i^*, m) - \hat{y}(x_i^*, m))^2},$$

where the RMSE for interpolation is calculated by the testing data from $[0.2, 1]$, and the RMSE for extrapolation is calculated by the testing data from $[0,0.2)$. In this example, the RMSE for interpolation is 0.0050, while the RMSE for extrapolation is
0.0280. As a visual illustration, Figure 4.15 shows the interpolation fit (the blue one) and the extrapolation fit (the green one) as well as the true function (the red one).

![Figure 4.15: Sequential ANOVA kriging predictor for the TR in Example 4.4.1](image)

**Example 4.4.2.** Consider an experiment with two quantitative variables, $\mathbf{x} = (x_1, x_2) \in [0, 1]^2$, and one qualitative factor $z$ with four levels. The true response surface functions of this experiment are

**Surface 1:** $y(\mathbf{x}, 1) = 2 \cos(3.5\pi(x_1 + x_2))$;

**Surface 2:** $y(\mathbf{x}, 2) = -2 \cos(-4\pi(x_1 + x_2)) + 3.5$;
Surface 3: \( y(x, 3) = 20(x_1 + x_2 - 1)^2 + 0.1 \sin(2\pi(x_1 + x_2)) + 3; \)

Surface 4: \( y(x, 4) = 30(x_1 + x_2 - 1)^2 + 0.1 \sin(2\pi(x_1 + x_2)) + 8. \)

The first three surfaces are the ARs and the last one is the TR. The initial design for each AR is independently chosen to be a LHD with \( n = 20, d = 2 \) from the region 
\([0, 1]^2\). The initial design for the TR is chosen to be a LHD with \( n = 12, d = 2 \) from the subregion 
\([0.2, 1]^2\). Thus, the predictions for the region \([0.2, 1]^2\) are interpolations, while the predictions for the region \([0, 1]^2/[0.2, 1]^2\) are extrapolations. The plot of the true response surfaces is displayed in Figure 4.16. As we can seen in the plot, the TR surface (the top one) shares more similarity with the third response surface (the second top one). And we expect a good predictor can use more information of the third response surface to predictor the TR surface for both interpolation and extrapolation.

After sequentially adding 30 points to the model (see Figure 4.17 for the design), the minimum minimax correlation is 0.5506. The ANOVA kriging model selects the third AR as the interpolation response. And the extrapolation weights of each ARs are 0.0358, 0.0520, and 0.9122, respectively. That means, the ANOVA kriging predictor mainly depends on the third AR, which is the same as our expectations. The testing data are taken at 51 \( \times \) 51 equally-spaced points in \([0, 1]^2\), denoted by \( x_1^*, x_2^*, \ldots, x_N^* \) with the first \( N_0 = 41^2 \) points in the region \([0.2, 1]^2\). In order to assess the fit of the model, the XVPE is calculated from formula (4.4.9). The XVPE for this example is 4.4390, which is a reasonable value given the range of the response surfaces. Furthermore, we can also derive the RMSE for interpolation and extrapolation. In this example, the RMSE for interpolation is 0.1847, while the RMSE for extrapolation
Figure 4.16: Plot of the true response surfaces in Example 4.4.2

is 0.7394. As a visual illustration, Figure 4.18 shows the interpolation fit and the extrapolation fit as well as the true surface.
Figure 4.17: The design with added points in Example 4.4.2
Figure 4.18: Sequential ANOVA kriging predictor for the TR in Example 4.4.2
Chapter 5: A General Composite Covariance Structure for GaSP Model with Mixed Inputs

Consider the simple case with only one qualitative input \( z \) with \( m \) levels, where \( z \in \{1, \cdots, m\} \). Recall that when there exists only one qualitative input, the Gaussian correlation structure for GaSP with mixed inputs (Qian et al. (2008)) is given by

\[
R(w_1 - w_2) = \tau_{z_1, z_2} \exp\left\{ - \sum_{i=1}^{d} \phi_i |x_{i1} - x_{i2}|^2 \right\} \tag{5.0.1}
\]

The parameter \( \tau_{z_1, z_2} \), which is also called cross-correlation parameter, measures the correlation between responses at any two input values that differ only on the qualitative factor \( z \). The scale correlation parameters \( \phi_1, \cdots, \phi_d \) measure the roughness of the response surfaces of the GaSP model with a fixed level of qualitative inputs.

Qian et al. (2008) also provided a general method for constructing a valid correlation function for \( \epsilon(w) \) in (5.0.1). This was accomplished by representing \( \epsilon(w) \) as a linear combination of independent mean 0 stochastic processes with the same variance and correlation function, and provided a characterization of the matrix \( T = (\tau_{r,s}) \) that allowed them to prove the PDUDE property.

In order to define the correlation function of \( \epsilon(w) = \epsilon(x, j), j = 1, \cdots, m \), Qian et al. (2008) let \( \epsilon^*(x) = (\epsilon(x, 1), \cdots, \epsilon(x, m))^T \) be a mean-0 \( m \)-variate process. Assume

\[
\epsilon^*(x) = AZ(x), \tag{5.0.2}
\]
where \( A = (a_1, \cdots, a_m)^T \) is an \( m \times m \) nonsingular matrix with unit row vectors (i.e. \( a_u^T a_u = 1 \), for \( u = 1, \cdots, m \)), and \( Z(x) = (Z_1(x), \cdots, Z_m(x))^T \) consists of \( m \) independent mean 0 Gaussian stochastic processes, \( Z_1(x), \cdots, Z_m(x) \), with the same variance \( \sigma^2 \) and correlation functions, \( R_\phi(\cdot), j = 1, \cdots, m \). Thus,

\[
\text{Corr}(Z(x_s), Z(x_t)) = R_\phi(x_s - x_t)I_m
\]

Then, we can obtain the correlation function for \( \epsilon(w) \) as

\[
\text{Corr}(\epsilon(w_s), \epsilon(w_t)) = \text{Corr}(a_s^T Z(x_s), a_t^T Z(x_t)) = a_s^T a_t R_\phi(x_s - x_t) \tag{5.0.3}
\]

Therefore, we can derive the cross-correlation parameter in (5.0.1) (Qian et al. (2008)) as \( \tau_{s,t} = a_s^T a_t \). The cross-correlation matrix can be expressed as

\[
T = AA^T. \tag{5.0.4}
\]

(5.0.1) implies for different levels of the qualitative factors, the roughness parameters and the variances are the same. We now relax this restriction. A more general construction of the correlation function is to convert \( \epsilon(w) \) into a linear combination of independent stochastic processes with “fewer” constraints on the variance and correlation functions.

5.1 Construction of the Composite Covariance Functions For Gaussian Processes With One Qualititative Input

We now relax the restrictions on \( Z_i(x), i = 1, \cdots, m \) and propose a new composite covariance structure. In order to define the composite covariance function of \( \epsilon(w) = \epsilon(x, j), j = 1, \cdots, m \), let \( \epsilon^*(x) = (\epsilon(x, 1), \cdots, \epsilon(x, m))^T \) be a mean-0 \( m \)-variate process. Analogous to the correlation structure construction in Qian et al.
(2008), we assume
\[ \epsilon^*(x) = AZ(x), \] (5.1.1)

where \( A = (a_1, \cdots, a_m)^T \) is an \( m \times m \) nonsingular matrix and \( Z(x) = (Z_1(x), \cdots, Z_m(x))^T \) consists of \( m \) independent mean 0 Gaussian stochastic processes, \( Z_1(x), \cdots, Z_m(x) \), with the same variance \( \sigma^2 \) but different correlation functions, \( R_{\phi_j}(\cdot), j = 1, \cdots, m \). Thus,
\[
\text{Corr}(Z(x_s), Z(x_t)) = \text{diag}(R_{\phi_1}(x_s - x_t), \cdots, R_{\phi_m}(x_s - x_t))
\]

Then, we can derive the covariance function for \( \epsilon(w) \) as
\[
\text{Cov}(\epsilon(w_s), \epsilon(w_t)) = \text{Cov}(a_s^T Z(x_s), a_t^T Z(x_t)) = a_s^T \text{diag}(R_{\phi_1}(x_s - x_t), \cdots, R_{\phi_m}(x_s - x_t))a_t \sigma^2 \]

(5.1.2)

We call the GaSP model with the covariance structure in (5.1.2) the Composite Gaussian Stochastic Process (CGaSP) model. Note that when all the correlation functions are the same \((R_{\phi_1}(x_s - x_t) = \cdots = R_{\phi_m}(x_s - x_t))\) and \( A \) has the unit row vector, we can derive the cross-correlation parameter in (5.0.1) and hence derive the correlation matrix in (5.0.4). Therefore, the covariance structure in (5.1.2) is an extension to the correlation structure in (5.0.1).

5.2 A Two-Step Procedure on the Global Fit of GaSP Model with One Qualitative Input

The general composite covariance structure in (5.1.2) involves more cross-correlation parameters in \( A \) and roughness parameters in the \( \phi \)'s. This creates computational complexity and numerical issues when estimating all the parameters in \( A \) and the
\(\phi\)'s. We can simplify the procedure with an additional step from the Hypersphere Parameterization model in Subsection 2.1.2.

The motivation is from the Cholesky decomposition of the cross-correlation matrix \(T\) in (2.1.10). The Cholesky decomposition of the estimated cross-correlation matrix \(T_j\) can be used as the nonsingular matrix \(A\) in (5.1.1). However, the decomposition of \(T\) in (5.0.4) is not unique. As one of the many decompositions for \(T\), the lower triangular matrix \(L\) from the Cholesky decomposition assumes the progressive complexity from the 1st qualitative level to the last one. The representation of level \(i\) is only in terms of previous levels. The “progressive complexity” imposed by the Cholesky decomposition may not be the most plausible choice.

On the other hand, exploratory factor analysis (EFA) provides methods for decomposing the cross-correlation matrix. The goal of factor analysis is to explain the joint variation in a set of random variables as arising from a smaller set of underlying “factors” (independent, unobserved, randomness). One starting point for EFA is the correlation matrix for this set of random variables. One hopes that this correlation matrix can be approximated by one of lower rank, determined by the underlying factors, matrix. The details of EFA can be found in Harman (1976).

Although EFA is usually used in reducing the dimension of the variables in multivariate analysis, it can also be applied directly to the correlation matrix \(T\) in (5.0.4). In particular, the factor loading \(L\) derived from EFA on correlation matrix has two properties.

1. \(T = LL^T\)

2. \(\Sigma = L^T * L\), where \(\Sigma\) is a diagonal matrix consisting of the eigenvalues of \(T\).
Therefore, we can find the factor loading $L$ using the singular value decomposition. Formally, the singular value decomposition of the symmetric $m \times m$ matrix $T$ is a factorization of the form $T = U \Sigma U^T$, where $U$ is a $m \times m$ orthogonal matrix ($UU^T = U^TU = I$, where $I$ is the identity matrix), $\Sigma$ is an $m \times m$ diagonal matrix with nonnegative real numbers on the diagonal. The diagonal entries of $\Sigma$ are the singular values of $T$. The $m$ columns of $U$ are called the left-singular vectors of $T$. Then, $U^T T U = U^T L * L^T U = \Sigma$, which implies that the different columns of $U^T L$ are orthogonal to each other. It follows that $U^T L$ can be expressed as the square root of the diagonal matrix $\Sigma$. Finally, the factor loading matrix for the cross-correlation matrix $T$ can be derived from

$$L = U^T \sqrt{\Sigma}. \quad (5.2.1)$$

For example, assume $T$ is a $3 \times 3$ cross-correlation matrix as follows:

$$T = \begin{bmatrix} 1 & -0.95 & 0.81 \\ -0.95 & 1 & -0.93 \\ 0.81 & -0.93 & 1 \end{bmatrix}.$$  

The cross-correlation matrix $T$ would arise if the curve with qualitative level 1 and qualitative level 3 are very similar while the curve with qualitative level 2 the reflection of those for qualitative levels 1 and 3 (See Figure 5.5 for an example of such a situation). It can also be shown that the matrix $T$ is a PDUDE matrix. The factor loading in (5.2.1) is given by

$$L = \begin{bmatrix} -0.9537 & -0.2955 & 0.0562 \\ 0.9951 & 0.0219 & 0.0968 \\ -0.9461 & 0.3208 & 0.0452 \end{bmatrix}.$$  

The eigenvalues of the cross-correlation matrix are given by $(2.7947, 0.1907, 0.0146)$, which implies that 93.16% of the variability can be explained by the $1^{st}$ factor, 6.36% can be additionally explained by the $2^{nd}$ factor and only 0.49% can be additionally
explained by the 3rd factor. Furthermore, for the main factor (factor 1, corresponding to the 1st column), the 1st and 3rd rows of the loading matrix \( L \) have similar values, while the 2nd row has essentially the opposite value of the other two. The results from the loading matrix and the eigenvalues are consistent with our comments about the cross-correlation matrix \( T \). Therefore, the loading matrix \( L \) is an appropriate candidate for the decomposition matrix of \( T \) and hence can be used in the covariance structure in (5.1.2).

Consequently, a simplified two-step procedure of predicting the response curves includes two steps:

**Step One:** Apply the hypersphere parameterization model in Subsection 2.1.2 and estimate the corresponding cross-correlation matrix \( T \).

**Step Two:** Apply factor analysis on \( T \) and set \( A \) to be the factor loading \( L \) from (5.2.1). Note that one could apply orthogonal rotations to \( L \) in the hopes of obtaining a “simpler” structure of \( A \) (easier to interpret \( A \)). Fit the GaSP model again with the covariance structure in (5.1.2).

This process can help one determine if there exists a relatively simple underlying structure for the effect of the qualitative levels (i.e. they are determined by a few underlying factors). If there is a simple structure, the number of parameters is reduced. Therefore, the two-step procedure can not only capture the complexity difference among different factor levels, but also simplify the computation difficulties in estimating the matrix \( A \).
5.3 Simulation Examples with One Qualitative and One Quantitative Input

In this Section, we will use two simulation examples to illustrate the advantages of the CGaSP model in the global fit of GaSP model with one qualitative and one quantitative input.

5.3.1 A Motivation Example from Huang et al. (2014)

The motivating example is from Huang et al. (2014). An experiment involving one qualitative input of three levels, 1, 2, 3, and one quantitative input \( x \) is considered. The simulated response curves are as follows:

\[
\begin{align*}
    y = \begin{cases} 
    \exp(2.5x) + 3(x - 0.5)^2 & z_1 = 1 \\
    \exp(2.5x) - 2 & z_1 = 2 \\
    2\cos(8\pi x) & z_1 = 3
    \end{cases}
\end{align*}
\]

The three true response curves are displayed in Figure 5.1. It is clear that the curves with levels 1 and 2 are similar to each other, but the curve with level 3 is distinct from the other two. The expectation is that there should be two underlying factors, one captures the trend in levels 1 and 2, the other captures the “wiggles” in level 3.

We apply the composite model with both \( A \) and \( \phi \)’s unknown and compare the prediction accuracy with the hypersphere parameterization model ((2.1.10) and (2.1.11) by Zhou et al. (2011)). Since the curve with level 3 is more “sophisticated” than the other two curves, the training data (Table 5.1) consists of 17 points with 5 points for the curves with level 1 and 2 and 7 points for the curve with level 3. The training data for curves with level 1 and 2 are generated by using a CSLHD with \( n = 10, d = 1, \) and \( m = 2 \) as described in Subsection 2.2.4, while the training data
for curve with level 3 is generated by a separate LHD with \( n = 7, \ d = 1 \). The test-set data are 50 equally-spaced points in \((0, 1) (0.01, 0.03, \cdots, 0.99)\).

\[
\begin{array}{cccccc}
  \text{z=1} & 0.4840 & 0.6043 & 0.1848 & 0.9846 & 0.3364 \\
  \text{z=2} & 0.5921 & 0.7979 & 0.0130 & 0.8704 & 0.2335 \\
  \text{z=3} & 0.8165 & 0.3516 & 0.1411 & 0.6918 & 0.2355 & 0.4610 & 0.9543 \\
\end{array}
\]
We consider two methods to fit the model:

*Method 1:* Fit the data with the hypersphere parameterization (HP) model in Subsection 2.1.2.

*Method 2:* Fit the data with the composite covariance structure (CGaSP) model in Section 5.1 with both $A$ and $\phi$’s unknown.

Parameter estimation for both models is by maximum likelihood. We use MATLAB to carry out the two methods. Figure 5.2 displays the comparison of the true curves with the predictions from the two methods. Table 5.2 shows the comparison of the estimated parameters as well as the RMSEs. The CGaSP model has more parameters, so one would expect better fit. Because of the different structures, the composite covariance model estimates three $\phi_i$’s. In addition, the estimated $A$ matrix in the composite covariance model is given by

$$
A = \begin{pmatrix}
6.2249 & -2.4843 & -0.0084 \\
7.1651 & -1.6946 & -0.1919 \\
0.1619 & 1.1658 & -0.1928
\end{pmatrix}
$$

<table>
<thead>
<tr>
<th>Method</th>
<th>$\hat{\phi}_1$</th>
<th>$\hat{\phi}_2$</th>
<th>$\hat{\phi}_3$</th>
<th>$RMSE_1$</th>
<th>$RMSE_2$</th>
<th>$RMSE_3$</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGaSP</td>
<td>0.9221</td>
<td>0</td>
<td>80.0000</td>
<td>0.0450</td>
<td>0.3439</td>
<td>1.5339</td>
<td>0.9080</td>
</tr>
<tr>
<td>Method</td>
<td>$\hat{\tau}_{12}$</td>
<td>$\hat{\tau}_{13}$</td>
<td>$\hat{\tau}_{23}$</td>
<td>$\phi_1$</td>
<td>$RMSE_1$</td>
<td>$RMSE_2$</td>
<td>$RMSE_3$</td>
</tr>
<tr>
<td>HP</td>
<td>0.3687</td>
<td>-0.4107</td>
<td>0.6961</td>
<td>39.5175</td>
<td>1.8739</td>
<td>1.0589</td>
<td>1.7420</td>
</tr>
</tbody>
</table>

The estimates of the $A$ matrix relates the underlying processes $Z(x)$ in (5.1.1) to the 3 curves. The estimates of $\hat{\phi}_i$’s for the CGaSP method imply that the underlying processes $Z(x) = (Z_1(x), Z_2(x), Z_3(x))$ in (5.1.1) has the covariance structure corresponding to three different roughness parameters.
Figure 5.2: Comparison of the HP model predictions and composite model predictions with the true curves

From Table 5.2, the roughness of the curve with level 3 affects the overall estimates of $\phi$ for hypersphere parameterization method. In contrast, the composite covariance model captures the different degrees of roughness. The three different estimates of the roughness parameters ($\hat{\phi}_1, \hat{\phi}_2, \hat{\phi}_3$) indicate three different underlying Gaussian stochastic processes in CGaSP model. By allowing different $\phi_i$'s for each level, the CGaSP method is more appropriate in this example because we would expect a
different roughness parameter for curve 3. Therefore, not surprisingly, the CGaSP model improves the prediction accuracy dramatically, which can also be seen in Figure 5.2.

In order to investigate the improvement of the two-step procedure described in Section 5.2 on the global fit of GaSP model with mixed inputs, we apply the two-step procedure to this example and compare the results with the results in HP model. Two decomposition methods for the second step are considered. One is the EFA decomposition, the other is the Cholesky decomposition. We record the results for the first step (HP model), and compare the results with the second step of the two decomposition methods.

The training data consists of 24 points with 8 points for each curve. We used two methods for selecting the training data:

1. For each level of $z_1$, a training sample is obtained by using the same LHD on $[0, 1]$ of eight runs for $x_1$.

2. The training sample is obtained by a CSLHD with $n = 24$, $d = 1$, and $m = 3$.

The test-set sample consists of 20 equally spaced points of the interval $[0, 1]$ (0, 1/19, 2/19, · · · , 1). The RMSEs are calculated for each qualitative level as well as for the overall fit. This procedure of data generation, modeling fitting, and assessment of prediction accuracy was repeated 100 times. For each time of repetition, a new design will be generated and hence produce a new prediction accuracy. Figure 5.3 compares the RMSEs of the test-set sample for the same initial design, while Figure 5.4 shows the comparison for the CSLHD. The average of the parameter estimates for $\phi_i$’s as well as the RMSEs are shown in Table 5.3.
Table 5.3: Two Steps Method comparison for Example 5.3.1

<table>
<thead>
<tr>
<th>Method</th>
<th>$\hat{\phi}_1$</th>
<th>$\hat{\phi}_2$</th>
<th>$\hat{\phi}_3$</th>
<th>$RMSE_1$</th>
<th>$RMSE_2$</th>
<th>$RMSE_3$</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>One$^{Same}$</td>
<td>44.8186</td>
<td>44.8186</td>
<td>44.8186</td>
<td>0.7600</td>
<td>0.6502</td>
<td>1.6777</td>
<td>1.1673</td>
</tr>
<tr>
<td>Two$^{Same}$</td>
<td>16.2786</td>
<td>39.2011</td>
<td>37.0021</td>
<td>0.3548</td>
<td>0.2469</td>
<td>1.7937</td>
<td>1.0841</td>
</tr>
<tr>
<td>Two$^{CHO}$</td>
<td>38.9431</td>
<td>36.6926</td>
<td>44.3211</td>
<td>0.7020</td>
<td>0.5574</td>
<td>1.6954</td>
<td>1.1526</td>
</tr>
<tr>
<td>One$^{CSLHD}$</td>
<td>65.6042</td>
<td>65.6042</td>
<td>65.6042</td>
<td>0.9404</td>
<td>0.6765</td>
<td>1.3861</td>
<td>1.0816</td>
</tr>
<tr>
<td>Two$^{CSLHD}$</td>
<td>10.6777</td>
<td>54.2542</td>
<td>59.5539</td>
<td>0.5333</td>
<td>0.5622</td>
<td>1.5991</td>
<td>1.0791</td>
</tr>
<tr>
<td>Two$^{CHO}$</td>
<td>8.2097</td>
<td>17.1817</td>
<td>65.9353</td>
<td>0.7636</td>
<td>0.9777</td>
<td>1.3700</td>
<td>1.1098</td>
</tr>
</tbody>
</table>

Figure 5.3: Boxplot comparison (left panel) of the three methods using the same LHD designs each level for Example 5.3.1 with the Step One on the left, the Cholesky decomposition in the middle and the EFA decomposition on the right. Boxplot comparison (right panel) of the improvement of the two decomposition methods with the Cholesky decomposition on the left, the EFA decomposition on the right.

When the training data are generated from the same LHD for each qualitative level, 83 out of 100 overall RMSEs for the two-step procedure are smaller than the RMSEs of the $1^{st}$ step (HP model) for the EFA decomposition. 55 out of 100 overall RMSEs for the two-step procedure are improved from the $1^{st}$ step for the Cholesky decomposition.
Figure 5.4: Boxplot comparison and the improvements of two steps methods using CSLHD designs for Example 5.3.1 with the Step One on the left, the Cholesky decomposition in the middle and the EFA decomposition on the right. Boxplot comparison (right panel) of the improvement of the two decomposition methods with the Cholesky decomposition on the left, the EFA decomposition on the right.

When the training data are generated from a CSLHD, 52 out of 100 overall RMSEs for the two-step procedure are smaller than the RMSEs of the 1\textsuperscript{st} step for the EFA decomposition. 44 out of 100 overall RMSEs for the two-step procedure are improved from the 1\textsuperscript{st} step for the Cholesky decomposition.

For the 1\textsuperscript{st} step (HP model), the CSLHD provides significant improvement over the same design. Therefore, given one uses a CSLHD design, differences in methods between the two steps are relatively small. The RMSEs shown in Table 5.3 verify the advantage of the two-step procedure with EFA decomposition for both types of designs. The Cholesky decomposition, however, fails to improve the global fit of the model for about half of the iterations in these two types of design cases. Therefore, the two-step procedure with EFA decomposition not only simplifies computation, but also
provides improved prediction accuracy for the GaSP model with both quantitative and qualitative input variables.

5.3.2 A Simulation Example from Zhou et al. (2011)

Another simulation example is from Zhou et al. (2011). This example also involves one qualitative input of three levels, 1, 2, 3, and one quantitative input $x$. The simulated response curves are given by

Example 5.3.2.

$$y = \begin{cases} 
\cos(6.8\pi x/2) & z_1 = 1 \\
-\cos(7\pi x/2) & z_1 = 2 \\
\cos(7.2\pi x/2) & z_1 = 3 
\end{cases}$$

The three true response curves are displayed in Figure 5.5. It is clear that the curves with level 1 and 3 are similar to each other, but the curve with level 3 is the “opposite” the other two.

We use training data consisting of 24 points with 8 points for each curve and carry out two methods for selecting the training data:

1. For each level of $z_1$, a training sample is obtained by using the same LHD on $[0, 1]$ of eight runs for $x_1$.

2. The training sample is obtained by a CSLHD with $n = 24$, $d = 1$, and $m = 3$.

The test-set sample is the same as in Example 5.3.2 $(0, 1/19, 2/19, \cdots , 1)$. The RMSEs are calculated for each qualitative level as well as for the overall fit. This procedure of data generation, modeling fitting, and assessment of prediction accuracy was repeated 100 times. For each time of repetition, a new design will be generated and hence produce a new prediction accuracy. Figure 5.6 compares the RMSEs of the testing sample for the same design, while Figure 5.7 shows the comparison for the
CSLHD The average of the parameter estimates for \( \phi_i \)'s as well as the RMSEs are shown in Table 5.4.

When the training data are generated from the same LHD for each qualitative level, 97 out of 100 overall RMSEs for the two-step procedure are smaller than the RMSEs of the 1st step (HP model) for the EFA decomposition. 42 out of 100 overall RMSEs for two-step procedure are improved from the 1st step (HP model) for the Cholesky decomposition.
Table 5.4: Two Steps Method comparison for Example 5.3.2

<table>
<thead>
<tr>
<th>Method</th>
<th>$\hat{\phi}_1$</th>
<th>$\hat{\phi}_2$</th>
<th>$\hat{\phi}_3$</th>
<th>$RMSE_1$</th>
<th>$RMSE_2$</th>
<th>$RMSE_3$</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$One_{Same}$</td>
<td>10.0000</td>
<td>10.0000</td>
<td>10.0000</td>
<td>0.0189</td>
<td>0.0216</td>
<td>0.0300</td>
<td>0.0241</td>
</tr>
<tr>
<td>$Two_{Same}EFA$</td>
<td>9.9996</td>
<td>8.9218</td>
<td>5.3438</td>
<td>0.0184</td>
<td>0.0229</td>
<td>0.0288</td>
<td>0.0239</td>
</tr>
<tr>
<td>$Two_{Same}CHO$</td>
<td>6.7456</td>
<td>6.8170</td>
<td>10.0000</td>
<td>0.0175</td>
<td>0.0239</td>
<td>0.0301</td>
<td>0.0244</td>
</tr>
<tr>
<td>$One_{CSLHD}$</td>
<td>10.0000</td>
<td>10.0000</td>
<td>10.0000</td>
<td>0.0212</td>
<td>0.0238</td>
<td>0.0325</td>
<td>0.0264</td>
</tr>
<tr>
<td>$Two_{EFA}CSLHD$</td>
<td>9.9999</td>
<td>8.9000</td>
<td>5.3462</td>
<td>0.0206</td>
<td>0.0252</td>
<td>0.0312</td>
<td>0.0261</td>
</tr>
<tr>
<td>$Two_{CHO}CSLHD$</td>
<td>6.7329</td>
<td>6.8117</td>
<td>10.0000</td>
<td>0.0191</td>
<td>0.0259</td>
<td>0.0326</td>
<td>0.0265</td>
</tr>
</tbody>
</table>

Figure 5.6: Boxplot comparison (left panel) of the three methods using the same LHD designs each level for Example 5.3.2 with the Step One on the left, the Cholesky decomposition in the middle and the EFA decomposition on the right. Boxplot comparison (right panel) of the improvement of the two decomposition methods with the Cholesky decomposition on the left, the EFA decomposition on the right.

When the training data are generated from a CSLHD, 96 out of 100 overall RMSEs for the two-step procedure are smaller than the RMSEs of the 1st step (HP model) for the EFA decomposition. 49 out of 100 overall RMSEs for two-step procedure are improved from the 1st step for the Cholesky decomposition.

The overall performance of the same design is slightly better than that of the CSLHD for this example. The similar or opposite trends among curves imply that the
Figure 5.7: Boxplot comparison and the improvements of two steps methods using CSLHD designs for Example 5.3.2 with the Step One on the left, the Cholesky decomposition in the middle and the EFA decomposition on the right. Boxplot comparison (right panel) of the improvement of the two decomposition methods with the Cholesky decomposition on the left, the EFA decomposition on the right.

cross-correlations are either close to 1 or -1 in this example. Therefore, the same initial design may help to better capture the cross-correlation values and hence produce a slightly better result. The improvement from the hypersphere parameterization model to the two-step procedure with EFA decomposition is tiny but mostly positive in this example. The RMSEs shown in Table 5.3 indicates the advantage of the two-step procedure with EFA decomposition over the Cholesky decomposition for both types of designs. Therefore, Example 5.3.2 also demonstrates that the two-step procedure with EFA decomposition is “best” for prediction in GaSP model with both quantitative and qualitative input variables.
5.4 Construction of the Composite Correlation Functions for Multiple Qualitative Inputs

A simple method for dealing with \( K \) qualitative inputs is to regard the combinations of the qualitative input as distinct levels of a new single qualitative input. For this method, there will be \( \prod_{j=1}^{K} m_j \) different qualitative levels, where \( m_j \) is the number levels of qualitative input \( j \).

However, as mentioned in Han et al. (2009), this method not only involves a large number of unknown parameters, but also ignores the factorial structure.

One way to address this issue is to use the product form of the cross-correlation parameters proposed by Qian et al. (2008). In this case, the number of cross-correlation parameters can be reduced from \( \left( \prod_{j=1}^{K} m_j \right)^2 \) to \( \sum_{j=1}^{K} \left( m_j^2 \right) \).

The correlation structure with the product cross-correlation form of Qian et al. (2008)’s model is given by

\[
R(\mathbf{w}_1 - \mathbf{w}_2) = \left[ \prod_{j=1}^{K} \tau_{j,z_{i1},z_{j2}} \right] \exp\left\{ - \sum_{i=1}^{d} \phi_i |x_{i1} - x_{i2}|^2 \right\}
\]

(5.4.1)

We define the overall cross-correlation matrix \( T \) as follows,

\[
T = (\tau_{z_i,z_j})
\]

(5.4.2)

where the order of \( z_i \) is defined by the lexicographic ordering.

For example, if \( K = 2, m_1 = m_2 = 2 \), \( T \) is defined as

\[
T = \begin{pmatrix}
\tau_{(1,1),(1,1)} & \tau_{(1,1),(1,2)} & \tau_{(1,1),(2,1)} & \tau_{(1,1),(2,2)} \\
\tau_{(1,2),(1,1)} & \tau_{(1,2),(1,2)} & \tau_{(1,2),(2,1)} & \tau_{(1,2),(2,2)} \\
\tau_{(2,1),(1,1)} & \tau_{(2,1),(1,2)} & \tau_{(2,1),(2,1)} & \tau_{(2,1),(2,2)} \\
\tau_{(2,2),(1,1)} & \tau_{(2,2),(1,2)} & \tau_{(2,2),(2,1)} & \tau_{(2,2),(2,2)} \\
\end{pmatrix}
\]

128
The relationship between the overall cross-correlation matrix $\mathbf{T}$ and the local cross-correlation matrix $\mathbf{T}_j$ can be written as

$$\mathbf{T} = \bigotimes_{j=1}^{K} \mathbf{T}_j. \quad (5.4.3)$$

We call the cross-correlation structure in (5.4.1) the Kronecker product structure.

We can derive a corresponding composite covariance structure for this multiple qualitative inputs case. For each qualitative input $z_j, j = 1, \ldots, K$, let $\mathbf{A}_j = (\mathbf{a}^{(j)}_1, \ldots, \mathbf{a}^{(j)}_{m_j}), j = 1, \ldots, K$ be a series of nonsingular matrices. We assume that $\mathbf{Z}^{(j)}(\mathbf{x}) = (Z^{(j)}_1(\mathbf{x}), \ldots, Z^{(j)}_{m_j}(\mathbf{x})), j = 1, \ldots, K$, where $Z^{(j)}_1(\mathbf{x}), \ldots, Z^{(j)}_{m_j}(\mathbf{x})$ are independent mean 0 Gaussian stochastic processes with the same variance $\sigma^2$ but different correlation functions $R_{\phi_i}^{(j)}(\cdot), i = 1, \ldots, m_j, j = 1, \ldots, K$. Therefore,

$$\text{Corr} (\mathbf{Z}^{(j)}(\mathbf{x}_s), \mathbf{Z}^{(j)}(\mathbf{x}_t)) = \text{diag}(R_{\phi_1}^{(j)}(\mathbf{x}_s - \mathbf{x}_t), \ldots, R_{\phi_{m_j}}^{(j)}(\mathbf{x}_s - \mathbf{x}_t))$$

Furthermore, the covariance function is

$$\text{Cov}(\epsilon(\mathbf{w}_s), \epsilon(\mathbf{w}_t)) = \prod_{j=1}^{K} \text{Cov}(\mathbf{a}^{(j)}_s^T \mathbf{Z}^{(j)}(\mathbf{x}_s), \mathbf{a}^{(j)}_t^T \mathbf{Z}^{(j)}(\mathbf{x}_t))$$

$$= \prod_{j=1}^{K} \mathbf{a}^{(j)}_s^T \text{diag}(R_{\phi_1}^{(j)}(\mathbf{x}_s - \mathbf{x}_t), \ldots, R_{\phi_{m_j}}^{(j)}(\mathbf{x}_s - \mathbf{x}_t)) \mathbf{a}^{(j)}_t \sigma^2 \quad (5.4.4)$$

Inspired by the Kronecker product structure of the cross-correlation matrix $\mathbf{T}$ in (5.4.1), we can define $\mathbf{A}$ as

$$\mathbf{A} = \bigotimes_{j=1}^{K} \mathbf{A}_j \quad (5.4.5)$$

As in the one qualitative input case, when the correlation functions for each qualitative input are the same ($R_{\phi_1}^{(j)}(\mathbf{x}_s - \mathbf{x}_t) = \cdots = R_{\phi_{m_j}}^{(j)}(\mathbf{x}_s - \mathbf{x}_t), j = 1, \ldots, K$) and $\mathbf{A}_j$ has unit row vectors, we can express the cross-correlation parameters in Qian et al.
(2008) as \( \tau_{j,s,t} = a_s^{(j)T} a_t^{(j)} \). Accordingly, \( T_j = A_j A_j^T, j = 1, \cdots, K \) and \( T = AA^T \). Therefore, the covariance function in (5.4.4) is an extension to the covariance structure in (5.4.1). We call the model with the covariance structure (5.4.4) the \emph{general Kronecker product decomposition Gaussian Stochastic Process} model (KPGaSP).

Furthermore, one can also apply the simplified two-step procedure in the multiple qualitative case. The algorithm also involves two steps:

**Step One:** Apply the hypersphere parameterization model in Subsection 2.1.2 and obtain the corresponding cross-correlation matrices \( T_1, \cdots, T_K \).

**Step Two:** Apply factor analysis to each \( T_j, (j = 1, \cdots, K) \) and set \( A_j, (j = 1, \cdots, K) \) to be the corresponding factor loading from (5.2.1). Fit the GaSP model again with the covariance structure in (5.4.4).

The reduced number of parameters in the simplified two-step procedure reduces the computation time dramatically. When \( K \geq 3 \), the number of unknown parameters in the matrices \( \{A_j, j = 1, \cdots, K\} \) can be large, which not only adds computational difficulty but also lowers the prediction accuracy. The simplified two-step procedure is recommended for such situations to reduce the computational time without the cost of prediction accuracy.

### 5.5 Special ANOVA decomposition of the Gaussian Processes Model with Multiple Qualitative Inputs

The covariance structure in Section 5.1 provides an intuitive extension of (5.0.1) in general form. The Gaussian stochastic processes \( \epsilon^*(x) \) can be viewed as a linear combination of independent stochastic processes as in (5.1.1). However, for the
multiplicative covariance structure in the Kronecker decomposition form, we can not
derive a corresponding linear combination form for $\epsilon^*(x)$ as in (5.1.1).

Inspired by analysis of variance, we can impose a factorial structure on the $Y(x, z)$
by placing constraints on the form of $A$ in (5.1.1). Based on different constraints,
we can derive different parameterizations of an ANOVA decomposition of Gaussian
stochastic processes.

### 5.5.1 General ANOVA Decomposition with No Constraint

Suppose $\epsilon(w) = \epsilon(x, z)$ has $K$ qualitative inputs $(t_1, \cdots, t_K)$. Following the
previous notation, let $Z_i(j)(x), i = 1, \cdots, m_j, j = 1, \cdots, K$ be independent mean 0
Gaussian stochastic processes with the same variance $\sigma^2$, but different correlation
functions $R_{\phi_i(j)}(\cdot), i = 1, \cdots, m_j, j = 1, \cdots, K$. If we view $Z_i(j)$ as the $i^{th}$ level main
effect of the $j^{th}$ qualitative input of the Gaussian process $\epsilon(x, z)$, we can decompose
$\epsilon(x, z)$ as

$$\epsilon(x, z) = \epsilon(x, z_1, \cdots, z_K) = \sum_{j=1}^{K} a^{(j)} Z^{(j)}(x)$$  \hspace{1cm} (5.5.1)

The covariance function is then

$$\text{Cov}(\epsilon(w_s), \epsilon(w_t)) = \left(\sum_{j=1}^{K} a^{(j)} Z^{(j)}(x) \right)^2 \sigma^2$$  \hspace{1cm} (5.5.2)

### 5.5.2 ANOVA Decomposition with constraints

One can impose a more classical factorial structure on $\epsilon(x, z)$ also. In this case,
we need to include an overall mean effect $Z_\mu(x)$ to capture the overall trend of the
processes. Moreover, one can also include a small "error" effect $Z_\epsilon(x)$ to allow small
fluctuations between different qualitative levels. We can then decompose $\epsilon(x, z)$ as

$$\epsilon(x, z) = Y(x, z_1, \cdots, z_K) = \mu Z_\mu(x) + \sum_{j=1}^{K} a^{(j)} Z^{(j)}(x) + \epsilon Z_\epsilon(x)$$  \hspace{1cm} (5.5.3)
The covariance function is

\[
\text{Cov}(\epsilon(w_s), \epsilon(w_t)) = (\mu^2 R(u)(x_s-x_t) + \sum_{j=1}^{K} a^{(j)} z_{sj}^2 \phi^{(j)}(x_s-x_t) + \epsilon^2 R(x_s-x_t)) \sigma^2
\]

(5.5.4)

Inspired by constraints placed on parameters in ANOVA, we can use two commonly used constraints to estimate the parameters in (5.5.4).

1. Impose \( \sum_{i=1}^{m_j} a^{(j)} = 0, j = 1, \ldots, K \)

2. Impose \( a^{(j)} = 0, j = 1, \ldots, K \).

Implementing the 1\(^{st}\) constraint requires complicated calculations, especially for multi-level situations. The 2\(^{nd}\) constraint is a somewhat simpler choice for an ANOVA decomposition, and we use this constraint. Furthermore, since \( \mu \) and \( \epsilon \) are non-identifiable, we usually ignore the “error” effect in practice.

Note that in (5.5.1) there are \( \sum_{j=1}^{K} m_j a^{(j)} \)'s. In (5.5.4) with the constraints, there are \( 2 + \sum_{j=1}^{K} (m_j - 1) \) parameters on \( a \).

5.6 Simulation Examples for the GaSP Model Involving Multiple Qualitative Inputs

5.6.1 An Illustration Example with One Quantitative and Two Qualitative Inputs in Han et al. (2013)

We consider a simulation example with one quantitative input and two qualitative inputs in Han et al. (2013). Both qualitative inputs have two levels and the quantitative input is in \([0, 1]\). The true curves are generated from the following equations:

**Example 5.6.1.**

\[
y = \begin{cases} 
0.3x + 0.1 \sin(2.5\pi x) + 0.5(x - 0.5)^2 & z_1 = 1, z_2 = 1 \\
0.1 + 0.3x + 0.1 \sin(2.5\pi x) + 2.5(x - 0.5)^4 - 0.4x^5 & z_1 = 1, z_2 = 2 \\
0.2 + 0.3x + 0.1 \sin(2\pi x) + 0.5(x - 0.5)^2 & z_1 = 2, z_2 = 1 \\
0.3 + 0.3x + 0.1 \sin(2\pi x) + 2.5(x - 0.5)^4 - 0.4x^5 & z_1 = 2, z_2 = 2
\end{cases}
\]

132
Figure 5.8 shows the true curves. Notice that the curve at level \((z_1, z_2) = (1, 1)\) shares the sine curve trend \(0.1 \sin(2.5\pi x)\) with the curve at \((z_1, z_2) = (2, 1)\), while the curve at level \((z_1, z_2) = (1, 2)\) shares the sine curve trend \(0.1 \sin(2\pi x)\) with the curve at \((z_1, z_2) = (2, 2)\). The curve at level \((z_1, z_2) = (1, 1)\) also shares the quadratic trend \(0.5(x - 0.5)^2\) with the curve at \((z_1, z_2) = (1, 2)\), while the curve at level \((z_1, z_2) = (2, 1)\) shares the fourth power trend \(2.5(x - 0.5)^4\) with the curve at \((z_1, z_2) = (2, 2)\). Therefore, a good predictor should be able to capture the sine trend for the 1st qualitative input and the polynomial trend for the 2nd qualitative input.
We consider two situations:

1. We investigate the prediction accuracy at combination \((z_1, z_2) = (1, 1)\), where only 4 data points are sampled. At each of the other three \((z_1, z_2)\) combinations, 10 data points are acquired. For each qualitative combination level, the input points are equally spaced over \([0, 1]\) by generating LHDs. The goal is to see the effect of “borrowing” information from the other three curves on prediction accuracy for the first curve.

2. We investigate the overall prediction accuracy at all combination levels. A CSLHD is generated with \(n = 20, d = 1,\) and \(m = 4\), so that we observe 5 points on each curve.

Sections 5.3 and 5.4 provide two methodologies to predict the observations with multiple qualitative inputs. We will examine the two situations with both methods: the general Kronecker product structure model, and the ANOVA decomposition model with the constraints \(a_2^{(j)} = 0, j = 1, 2\). To avoid computational problems that might lead to (near) singularity of the correlation matrices, the bounds for the entries of the nonsingular matrix \(A_j\) are all set to be 1, while the upper bounds for all the \(\phi\)’s are set to be 10. To be consistent with the results in Han et al. (2013), the test-set sample is taken to be the 101 equally spaced points of the interval \([0, 1]\) \((0, 0.01, 0.02, \cdots, 1)\). The RMSEs are calculated for each qualitative level as well as for the overall fit. For each situation and each method, the procedure of data generation, modeling fitting, and assessment of prediction accuracy was repeated 100 times. For each repetition, a new design will be generated and hence produce a new prediction accuracy.
Table 5.5 shows the average (overall 100 repetitions) of the estimated $\phi_{ij}$’s as well as the RMSEs for the general Kronecker product structure model for the two data generation situations. Table 5.6 shows the average of the estimated $\phi_i^j$’s as well as the RMSEs for the ANOVA decomposition model of the two data generation situations. The bold values are what we are interested in (overall RMSE for situation 2 and $RMSE_1$ for situation 1).

Table 5.5: Average parameter estimations and RMSEs for general Kronecker product structure model of Example 5.6.1

<table>
<thead>
<tr>
<th>Situation</th>
<th>$\hat{\phi}_{11}$</th>
<th>$\hat{\phi}_{12}$</th>
<th>$\hat{\phi}_{21}$</th>
<th>$\hat{\phi}_{22}$</th>
<th>$RMSE_{\hat{\phi}_{11}}$</th>
<th>$RMSE_{\hat{\phi}_{12}}$</th>
<th>$RMSE_{\hat{\phi}_{21}}$</th>
<th>$RMSE_{\hat{\phi}_{22}}$</th>
<th>$RMSE$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>1.9994</td>
<td>2.0330</td>
<td>2.8746</td>
<td>2.9627</td>
<td>0.0123</td>
<td>0.0112</td>
<td>0.0148</td>
<td>0.0110</td>
<td>0.0136</td>
</tr>
<tr>
<td>(1,1) Fit</td>
<td>1.8019</td>
<td>1.5417</td>
<td>1.9618</td>
<td>1.8452</td>
<td>0.0204</td>
<td>0.0004</td>
<td>0.0001</td>
<td>0.0004</td>
<td>0.0102</td>
</tr>
</tbody>
</table>

Table 5.6: Average parameter estimations and RMSEs for ANOVA decomposition model of Example 5.6.1

<table>
<thead>
<tr>
<th>Situation</th>
<th>$\hat{\phi}_\mu$</th>
<th>$\hat{\phi}_1^{(1)}$</th>
<th>$\hat{\phi}_1^{(2)}$</th>
<th>$RMSE_{\hat{\phi}_1^{(1)}}$</th>
<th>$RMSE_{\hat{\phi}_1^{(2)}}$</th>
<th>$RMSE_{\hat{\phi}_2}$</th>
<th>$RMSE_{\hat{\phi}_2}$</th>
<th>$RMSE_{\hat{\phi}_2}$</th>
<th>$RMSE_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>1.9441</td>
<td>6.4953</td>
<td>6.0844</td>
<td>0.0122</td>
<td>0.0113</td>
<td>0.0092</td>
<td>0.0092</td>
<td>0.0120</td>
<td></td>
</tr>
<tr>
<td>(1,1) Fit</td>
<td>1.6660</td>
<td>5.3314</td>
<td>4.4842</td>
<td>0.0016</td>
<td>0.0008</td>
<td>0.0010</td>
<td>0.0010</td>
<td>0.0011</td>
<td></td>
</tr>
</tbody>
</table>

For the overall fit setting, Figure 5.9 displays the boxplot comparisons of RMSEs for each curve as well as for overall fit between the two models. For the individual fit setting (Fit of (1, 1)), Figure 5.10 also displays the boxplot comparisons of RMSEs between the two models.

In Han et al. (2013), only one fixed initial design is considered. For that specified initial design, the corresponding RMSEs of the $Y(X, 1, 1)$ predictor from the HQQV, HAK, and HAK with interaction models are 0.0384, 0.0237, and 0.0182, respectively.

From Table 5.6 as well as the box plots in Figure 5.10, it is clearly that the ANOVA decomposition model improves dramatically over Han et al. (2013)’s methods. The
Figure 5.9: Boxplot of the RMSEs of Example 5.6.1 for overall fit situation

Figure 5.10: Boxplot of the RMSEs of Example 5.6.1 for (1,1) individual fit situation

average RMSE for curve (1,1) for the general Kronecker product model (0.0204) is very close to the best one (HAK with interaction) from Han et al. (2013)’s example and performs better than the other two (HQQV and HAK). Furthermore, in this example,
the ANOVA decomposition model outperforms the general Kronecker product model not only in the individual fit setting but also in the overall fit setting.

5.6.2 A Simulation Example with two Quantitative and two Qualitative Inputs

Consider a simulation example with two quantitative and two qualitative inputs. Suppose that both qualitative inputs have two levels and the quantitative inputs are in $[0, 1]^2$. The true surfaces are from the following equations:

\[
\begin{align*}
    y &= \begin{cases} 
        \sin(2\pi x_1) + 2(x_2 - 2)^2 & z_1 = 1, z_2 = 1 \\
        \sin(2\pi x_1) + 3(x_2 - 3)^2 & z_1 = 1, z_2 = 2 \\
        \cos(3\pi x_1) + 2(x_2 - 2)^2 & z_1 = 2, z_2 = 1 \\
        \cos(3\pi x_1) + 3(x_2 - 3)^2 & z_1 = 2, z_2 = 2
    \end{cases}
\end{align*}
\]

Figure 5.11 displays the true surfaces. The $1^{st}$ qualitative input $z_1$ corresponds to the sine term of the $1^{st}$ quantitative input $x_1$, and the $2^{nd}$ qualitative input $z_2$ corresponds to the quadratic term of the $2^{nd}$ quantitative input $x_2$. Based on this additive structure, we expect the ANOVA decomposition model should perform better than the general Kronecker product model.

We consider 3 different methods: the general Kronecker product method, the ANOVA decomposition method with constraints $a_{2}^{(j)} = 0, j = 1, 2$, and the ANOVA decomposition method with no constraint. The training sample is taken from a CSLHD with $n = 40, d = 2$, and $m = 4$. The test-set sample is taken $51 \times 51$ equally spaced points of the sample space $[0, 1]^2$. The RMSEs are calculated for each combined qualitative level and overall. For each method, the procedure of data generation, modeling fitting, and assessment of prediction accuracy was repeated 50 times. For each time of repetition, a new design will be generated and hence produce
Figure 5.11: The True Surfaces of Example 5.6.2

a new prediction accuracy. Figure 5.12 shows the comparison of the RMSEs for each method and each combined level.

The regular ANOVA decomposition method outperforms the other two methods as we expected. The failure to account for some levels’ effect in the ANOVA decomposition model with constraints may be the reason to cause its poor performance. The general Kronecker product method requires the same number of parameters $\phi_i^j$, but requires additional parameters for the $A$ matrix. The additional redundant parameters might be the reason for its poor performance.
5.6.3 The Borehole Example

Morris, Mitchell and Ylvisaker (1993) introduced the borehole function to illustrate various methods in computer experiments. The borehole model formulation is based on assumptions of no groundwater gradient, steady-state flow from the upper aquifer into the borehole and from the borehole into the lower aquifer. The response variable $y$ of the borehole function is the flow rate of water through a borehole in $m^3/yr$. Table 5.7 presents the eight input variables of the borehole function.

The response $y$ of the function is defined as

$$y = \frac{2\pi T_u (H_u - H_i)}{\left(\log(r/r_w) \left[1 + \frac{2LT_u}{\log(r/r_w)r_w K_w} + \frac{T_u}{T_i}\right]\right)}$$  \hspace{1cm} (5.6.1)

Zhou et al. (2011) used the borehole function to create an example with four quantitative input variables ($x_1, x_2, x_3, x_4$) and three qualitative input variables ($z_1, z_2, z_3$).
Table 5.7: **Input variables in the borehole example**

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

The three qualitative factors correspond to replacing four of the original quantitative variables with qualitative variables representing fixed levels of these quantitative variables and are listed in Table 5.8. Note that $H_u - H_l$ is treated as a single factor due to identifiability issues because $y$ is a function of $H_u$ and $H_l$ only through $H_u - H_l$.

The remaining four input variables in Table 5.7 are treated as the quantitative inputs. That is, $x_1 = r_w$, $x_2 = T_u$, $x_3 = T_l$, $x_4 = L$.

Table 5.8: **Borehole example: three qualitative factors and their levels**

<table>
<thead>
<tr>
<th>Level</th>
<th>$z_1 = r$</th>
<th>$z_2 = H_u - H_l$</th>
<th>$z_3 = K_w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10,000</td>
<td>200</td>
<td>10,000</td>
</tr>
<tr>
<td>2</td>
<td>20,000</td>
<td>300</td>
<td>11,000</td>
</tr>
<tr>
<td>3</td>
<td>30,000</td>
<td>400</td>
<td>12,000</td>
</tr>
</tbody>
</table>

In Zhou et al. (2011)'s example, a training sample of 216 runs is obtained by generating $3^3 = 27$ separate 8-run LHDs for the quantitative variable, one for each of the 27 level combinations of the qualitative variables. Data from the training sample
were used by Zhou et al. (2011) to fit prediction models based on the exchangeable correlation, the multiplicative correlation, and the hyperspherical parameterization correlation methods. The accuracy of the prediction model for each of the three methods was assessed on a test-set sample of 405 runs in which the test-set is obtained by generating $3^3 = 27$ separate 15-run LHDs for the quantitative variable, one for each of the 27 level combinations of the qualitative variables. They repeated the procedure of data generation, modeling fitting, and model assessment 100 times for each of the three methods. For each time of repetition, a new training data and a new test-set data were generated and hence produced a new prediction accuracy. They compared the RMSEs for the test-set data in boxplots. The hyperspherical parameterization correlation method stands out as best in comparison to the others.

For our methods, because of the complexity of the decomposition structures, the running times are much longer than the Hyperspherical parameterization models. Hence, we only consider one set of training data and test-set data, and compare the results with the hyperspherical parameterization model’s results for purposes of comparisons with Zhou et al. (2011). We consider five candidate models:

1. **Hyperspherical parameterization model**: We apply the hyperspherical parameterization model with the Kronecker product structure in (5.4.1). We need to estimate $4 \phi_i$’s for the quantitative inputs and $9 \theta_j^{(j)}$ for the qualitative inputs.

2. **The general Kronecker product decomposition model**: We apply the general Kronecker product decomposition model in (5.4.4). The unknown parameters include 27 parameters in the $A_j$ matrices and 36 $\phi_j$’s.
3. **The simplified two-step procedure model:** We treat the factor loadings of the $T_j$ matrix in Method 1 above as the known $A_j$ and apply the Kronecker product decomposition method in (5.4.4). The unknown parameters include 36 $\phi_j$’s.

4. **The main effect ANOVA decomposition model:** We apply the main effect ANOVA decomposition model in (5.5.1). The unknown parameters include 9 $a_i^{(j)}$’s ($i = 1, \cdots, 3, j = 1, \cdots, 3$) and 36 $\phi^{(k)}_{ij}$’s ($i = 1, \cdots, 3, j = 1, \cdots, 4, k = 1, \cdots, 3$).

5. **The main effect ANOVA decomposition model with constraints:** We impose $a_3^{(j)} = 0$ and include an overall effect $\mu$ on the ANOVA decomposition model in (5.5.4). The unknown parameters include the overall effect $\mu$ and 4 corresponding $\phi_{ij}$’s ($j = 1, \cdots, 4$), 6 $a_i^{(j)}$’s ($i = 1, \cdots, 2, j = 1, \cdots, 3$) and 24 $\phi^{(k)}_{ij}$’s ($i = 1, \cdots, 2, j = 1, \cdots, 4, k = 1, \cdots, 3$).

The RMSEs of each combined qualitative level and the overall RMSE can be calculated for the test-set data. Table 5.9 shows the comparison of the RMSEs among these 5 models.

The hyperspherical parameterization (HP) method doesn’t perform well for the borehole example. The HP method has far fewer parameters than the others. This may explain the poorer fit but faster run time for HP method. Furthermore, the fast running time for the HP model provides a starting point for the $A_j$ matrices used in the simplified two-step (TS) method (method 3). Based on the factor decomposition of the $T_j, j = 1, 2$ matrices from HP model, we can apply the second step of the simplified two-step (TS) method. The second step not only shortens the running time
of the regular Kronecker product (KP) decomposition model, but also improves the prediction accuracy dramatically. The TS method outperforms the other methods in most of the combined levels and the overall fit. The main effect ANOVA decomposition model (ANOVA) performs similar to the regular KP decomposition model and both of them outperform the main effect ANOVA decomposition model with constraints (ANOVA$_C$). The ANOVA$_C$ model treats the last level of each qualitative input as the reference level, which assumes the simplicity of the last level. However, the complicated structure of the borehole function makes the last level nontrivial. Therefore, the constraint is not necessary and imposes unrealistic restrictions on the parameter, which explains why the performance of the regular ANOVA is better than the ANOVA$_C$.

143
5.7 Sequential Design Algorithm for the Composite Gaussian Stochastic Process Models with Mixed inputs

In order to improve prediction accuracy, we propose a sequential design algorithm to obtain a good global fit of the model. With qualitative inputs, we need to determine both the qualitative levels and the quantitative values of a new design point. Cross validation is a popular method for model estimation and validation. We modify the cross validation prediction error criteria in Lam and Notz (2008) (Section 1.5.3) and introduce a measure of the prediction error which can be used as a design criterion to select additional input points for both the quantitative and qualitative inputs.

Consider the simple case with only one qualitative input $z$. We assume the existing data have $m$ qualitative levels, each level has $n_z$ ($z = 1, \ldots, m$) observations. Suppose $(x, z)$ is a new candidate point. Let $\hat{Y}^{(-j_z)}(x, z)$ denote the EBLUP from the general composite covariance model (Section 5.1) based on all the data except \{(x_{j_z}, z), y(x_{j_z}, z)\}, ($z = 1, \ldots, m$, $j_z = 1, \ldots, n_z$), and $\hat{Y}(x, z)$ denote the EBLUP from the same model using all the data.

**Arithmetic Mean** For the new candidate point $(x, z)$, we compute the adaptive arithmetic mean cross validation error:

$$XVPE_A(x, z) = \sqrt{\frac{1}{n_z} \sum_{j_z=1}^{n_z} (\hat{Y}^{(-j_z)}(x, z) - \hat{Y}(x, z))^2 \times \min_j (d(x_i, x_0))}$$ (5.7.1)

**Geometric Mean** The geometric mean cross validation error for computer experiments with both quantitative and qualitative inputs is given by

$$XVPE_G(x, z) = \sqrt[n_z]{\prod_{j_z=1}^{n_z} (\hat{Y}^{(-j_z)}(x, z) - \hat{Y}(x, z))^2}$$ (5.7.2)
**Harmonic Mean** The *harmonic mean* cross validation error with both quantitative and qualitative inputs is given by

$$XVPE_{H}(\mathbf{x}, z) = \frac{n}{\sum_{j=1}^{n_z} \frac{1}{\hat{Y}(-jz)(\mathbf{x}, z) - Y(\mathbf{x}, z))^2}}$$ (5.7.3)

For each of those validation criteria, one can pick as the next design point, \((\mathbf{x}_0, z_0)\), on the surface of qualitative level \(z_0\), that has the largest “mean” cross validation error.

Another strategy to select the new point is based on the MSPE. One can generate a grid of points on the sample space and for each compute EBLUP and MSPE. The point with the maximum MSPE is the one with the largest prediction uncertainty. Therefore, one can add the point with the largest MSPE as the next design point.

**5.7.1 An Example of the Sequential Algorithm for the Composite GaSP model**

**Example 5.7.1.** (Example 5.3.1, continued) In Example 5.3.1, we generated an initial design of 17 points and applied the composite GaSP model to these 17 points. In this example, we will use the same initial design of Example 5.3.1 (Table 5.1) and apply the sequential algorithms to add points. Figure 5.13 displays the curves with the initial designs. The first two curves are flat, while the last sine curve is rough. From Figure 5.13, one would expect that most (or all) points should be added to curve 3 in order to “capture” the sinusoidal variation accurately.

We apply four sequential algorithms to the existing data: the MSPE criterion, the adaptive arithmetic mean criterion, the geometric mean criterion and the harmonic mean criterion. After sequentially adding 5 points, the corresponding parameter estimates and the RMSEs are listed in Table 5.10. The first row shows the results for
Figure 5.13: CSLHD

the composite GaSP model based only on the initial points. Figure 5.14 displays the predictions of each sequential criterion as well as the added points.

Table 5.10: Comparison of the sequential design and the RMSE

<table>
<thead>
<tr>
<th>Method</th>
<th>$\phi_1$</th>
<th>$\phi_2$</th>
<th>$\phi_3$</th>
<th>$RMSE_1$</th>
<th>$RMSE_2$</th>
<th>$RMSE_3$</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Composite</td>
<td>0.9221</td>
<td>0</td>
<td>80.0000</td>
<td>0.0450</td>
<td>0.3439</td>
<td>1.5339</td>
<td>0.9080</td>
</tr>
<tr>
<td>MSPE</td>
<td>0.0060</td>
<td>0.9652</td>
<td>63.5266</td>
<td>0.1894</td>
<td>0.7346</td>
<td>0.8168</td>
<td>0.6436</td>
</tr>
<tr>
<td>adaptiveXV</td>
<td>0.0139</td>
<td>0.8499</td>
<td>133.8788</td>
<td>0.0885</td>
<td>0.1032</td>
<td>0.3549</td>
<td>0.2194</td>
</tr>
<tr>
<td>geometricXV</td>
<td>0.6936</td>
<td>0.9301</td>
<td>114.4166</td>
<td>0.0608</td>
<td>0.0346</td>
<td>1.3625</td>
<td>0.7877</td>
</tr>
<tr>
<td>harmonicXV</td>
<td>7.8415</td>
<td>128.3549</td>
<td>22.3125</td>
<td>0.8657</td>
<td>0.2152</td>
<td>1.5706</td>
<td>1.0428</td>
</tr>
</tbody>
</table>
Figure 5.14: Comparison of the Sequential Methods: the left top one is the MSPE criterion, the right top one is the adaptive arithmetic mean criterion, the left bottom one is the geometric mean criterion, the right bottom one is the harmonic mean criterion.

Table 5.10 indicates that the adaptive arithmetic mean criterion outperforms other criteria. The harmonic mean criterion fails to improve the prediction accuracy. The results for harmonic mean criterion are worse than the original prediction results. The reason can be seen in Figure 5.14. The harmonic mean criterion selects points that all clump together, which provide no information about the response at other value of the
quantitative inputs and can cause near singularity problems for the covariance matrix. The adaptive arithmetic mean criterion, on the other hand, selects points spread out on the 3\textsuperscript{rd} curve, which is the one we need more information about. Therefore, the adaptive arithmetic mean criterion improves the prediction accuracy dramatically only based on 5 additional points. The MSPE criterion improves the fit of the 3\textsuperscript{rd} curve a bit, but the extra information on the 3\textsuperscript{rd} curve seems to affect the fit of the predictions on the other two curves, which leads to only a small overall improvement. The geometric criterion also clumps points. No points are added between 0 and 0.1 on the 3\textsuperscript{rd} curve, which leads to poor fit between 0 and 0.1 on the 3\textsuperscript{rd} curve.
Chapter 6: Concluding Remarks and Future Directions

In this dissertation, we have discussed computer experiments with both quantitative and qualitative inputs. In this chapter, we summarize some conclusions and list some future topics.

6.1 For Models

In Chapter 3, we introduced two indicator-based GaSP models in which we represent the qualitative inputs as transformed quantitative inputs when the cross-correlation parameter $\tau_{ij}$’s are restricted to be positive. The two new methods can be easily implemented in commercial software, like JMP. We further discussed the effect of the restrictions on the range of the cross-correlation parameters on the prediction accuracy.

In Chapter 5, we extended the Qian et al. (2008) regular cross-correlation GaSP model to a general composite covariance structure GaSP model. The new model allows one to capture potential roughness differences between some qualitative levels. To simplify the computational complexity of the composite GaSP model, we also provided a two-step procedure which can efficiently improve the prediction accuracy of the general hypersphere parameterization model (Zhou et al. (2011)).
For a given problem, which model provides the “best” prediction accuracy varies from case to case. We summarize some general conclusions that can be made from the simulation results in Chapter 3 and Chapter 5.

First, for the examples with simple response surface structures, the two indicator-based methods can produce decent and quick predictions in JMP. The shortcoming is that the corresponding cross-correlation parameters are restricted to positive values, which excludes the case where two curves are “negatively correlated”. However, with an appropriate setting of the parameter bounds, one can obtain a good fit for “negatively correlated” curves in MATLAB, by estimating the corresponding cross-correlation parameters as zero.

Second, the hypersphere parameterization model is recommended for the case that the roughness parameters for different qualitative levels are similar. When there are potentially different roughness for different qualitative levels, the composite GaSP model is recommended. One can either fit the composite model directly or fit the model with the simplified two-step procedure.

Third, when there are multiple qualitative inputs, Qian et al. (2008) proposed a Kronecker product structure for the cross-correlation matrix which reduces the number of cross-correlation parameters and hence reduces the computational complexity. Accordingly, the Kronecker product structure can also be applied to the composite GaSP model. We also derive the two-step procedure with the Kronecker product structure. For computer experiments with a large number of combined qualitative levels, the two-step Kronecker product decomposition model can produce a good fit efficiently (see borehole example in Section 5.6.3). Furthermore, when the effect of the qualitative input can be decomposed into the main effect of each qualitative input,
we introduced the ANOVA decomposition GaSP model. The ANOVA decomposition model can reduce the total number of parameters in the Kronecker product GaSP model dramatically for the qualitative inputs with multiple levels and hence reduce the computational complexity and running time.

6.2 For experimental designs

6.2.1 Initial Designs

In Chapter 4, we proposed two types of designs for computer experiments with both quantitative and qualitative inputs. One argument is that identical or clustered values of the quantitative inputs for different qualitative levels can help to better capture the correlations among different qualitative levels hence experimental designs should have this property. However, there does not appear to be any theoretical or empirical justifications for this argument. Good designs should depend on the true response surfaces as well as the sample sizes. Generally speaking, there are two features of designs: one is how spread-out points are over the whole quantitative inputs space, the other is duplication and clustering of the quantitative inputs for the different qualitative levels. We explore this in Chapter 4 and now summarize general conclusions that can be made from the simulation studies in Chapter 4.

For flat surfaces with similar shapes for all the qualitative levels, spread-out designs (like k-LHD, SLHD, partial SLHD) appear to produce better prediction.

For rough surfaces with small or medium similarities in shape among all the qualitative levels, designs with identical or clustered points (CSLHD, partial CSLHD) appear to produce better predictions.
For surfaces with a mixture of degrees of roughness and similarity of shapes, the performance for these two types of designs are similar for a small sample sizes. As sample sizes increase, spread-out designs appear to perform better.

6.2.2 Sequential Designs

When the initial design leads to poor predictions, design points can be added sequentially to improve the predictions. In this dissertation, we considered two situations in which sequential designs can be used.

One situation is to predict the target response (TR) on restrictive input regions. This situation might arise if for one of the qualitative levels (the target response), certain regions of the input space for the quantitative variables are “difficult” to simulate. The idea of sequential ANOVA Kriging is to select the closest auxiliary response (AR) for interpolation and extrapolation predictions on the TR and to add design points sequentially at points (values of the quantitative inputs) and level where we have the “least” information, perhaps as reflected in large prediction variances.

The other situation is to predict all the levels of the responses. Our sequential algorithm is based on the composite GaSP model. The criterion for adding a new design point is based on the cross validation error or the MSPE. The new point can be added to the response level and point with the largest “mean” cross validation error or MSPE.

6.3 Future topics

We now describe four future directions one could take regarding computer experiments with both quantitative and qualitative inputs.
1. Additional examples with more quantitative inputs and more qualitative inputs (especially when the qualitative inputs have some factorial structure) can be explored for each of the predictive models as well as the experimental designs. An important challenge is the computational issues.

2. One can study other criteria (than CV or MSPE) for sequential designs. In this dissertation, we considered designs to perform well if the overall fit is good. But it might be interesting to explore problems where the good is optimization. One question might be to determine which level of the qualitative input is “best”.

3. When there are multiple qualitative inputs, one can explore a more sophisticated ANOVA decomposition of the composite GaSP model, such as inclusion of interactions.

4. Another future research direction can be a more thorough study of designs, including the proper trade-off between duplication of points and speeding out points. In this dissertation, we set a quarter of the quantitative design points among different qualitative levels to be the same in both partial SLHD and partial CSLHD methods. One can explore the effect on prediction accuracies from different portion of duplicates.
Bibliography


