COMPUTER AND BIOLOGICAL EXPERIMENTS: MODELING, ESTIMATION, AND UNCERTAINTY QUANTIFICATION

A Dissertation
Presented to
The Academic Faculty

By

Li-Hsiang Lin

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial & Systems Engineering

Georgia Institute of Technology

August 2020

COMPUTER AND BIOLOGICAL EXPERIMENTS: MODELING, ESTIMATION, AND UNCERTAINTY QUANTIFICATION

Approved by:

Dr. V. Roshan Joseph, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. C. F. Jeff Wu, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Ying Hung
Department of Statistics
Rutgers, The State University of New Jersey

Dr. Xiaoming Huo
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Cheng Zhu
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: April 14, 2020

To my beloved family,

for their unconditional love and support.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all who have influenced, inspired, and supported my work in their various capacities. Without their support, this dissertation would not have been possible.

First, I would like to express my deep appreciation to my advisor, Professor C. F. Jeff Wu, for his continuous support of my Ph.D. studies and research. His enthusiasm and immense knowledge transformed me into a more mature thinker. An incredible scholar, he is also compassionate, and encourages his students to persevere during challenging times. His mentorship was pivotal in expanding my perspectives on research and insights in personal and professional development.

I am also extremely grateful of my co-advisor, Professor Roshan Joseph Vengazhiyil, for his guidance during my studies. With his patience, I learned to think critically to develop practical methodologies in statistics. I am very fortunate to acquire the necessary knowledge and skills of academic research from him.

I would also like to thank Professor Ying Hung for her support and advice for my research and career. She spent countless hours in discussion with me, proposing areas of improvements in my research. Furthermore, her valuable advice resulted in a smoother start of my academic journey. I owe a great debt of gratitude to her.

I would also like to extend my sincere appreciation to my committee members, Professor Xiaoming Huo and Professor Cheng Zhu, for their generous help and insightful suggestions on my doctoral studies and dissertation.

I am very thankful of my lab members, Dr. Chih-Li Sung, Dr. Simon Mak, Dr. Rui Tuo, Dr. Wenjia Wang, Dr. David Zhao, Arvind Krishna, Zhehui Chen, Shaowu Yuchi, and Chaofan Huang, for their company and enlightening conversations. I would also like to thank my friends, especially Muya Chang, Yi-Han Lu, Wendy Kohn, Ana Maria, Yen Kuo, Rui-Bao Wu, Namjoon Suh, and Hyojung Kim, for the fun and memorable time together. I

am extremely fortunate to have such wonderful people around me during my Ph.D. years.

Last, but by no means the least, my heartfelt appreciation goes to my family, especially my father, who continually encourages me to pursue my passion. Also my heartfelt thanks to Lauren, for her constant support, encouragement, and faith in me no matter the situation. This dissertation is dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Statistical experimental analysis is an indispensable tool in engineering, science, biomedicine, and technology innovation. There are generally two types of experiments: computer and physical experiments. Computer experiments are simulations using complex mathematical models and numerical tools, while physical experiments are actual experiments performed in a laboratory or observed in the field. Analyzing these experiments helps us understand real-world phenomena and motivates interesting statistical questions and challenges. This thesis presents new methodologies for applications in computer experiments and biomedical studies.

In Chapter 1, we propose a new method based on Gaussian processes (GPs) for analyzing computer experiments. GP is a popular choice for approximating a deterministic function in computer experiments. However, the role of transformation in GP modeling is not well understood. Here, we proposed using transformation in GP modeling to improve additivity. This involves finding a transformation of the response such that the deterministic function becomes an approximately additive function, which can then be easily estimated using an additive GP. We call this GP a Transformed Additive Gaussian (TAG) process. Furthermore, to capture possible interactions that are unaccounted for in the additive model, we proposed an extension of the TAG process called Transformed Approximately Additive Gaussian (TAAG) process. We develop efficient techniques for fitting a TAAG process. In fact, we show here that TAAG can be fitted to high-dimensional data much more efficiently than standard GP. Additionally, we show that compared with a standard GP, TAG produces better estimation, interpretation, visualization, and prediction. The proposed methods are implemented in the R package TAG.

In Chapter 2, we show that the concept of using transformation for improving the additivity of a target function is beneficial in big data modeling. After improving the additivity, the target function is easier to approximate and is expected to be well-approximated us-

ing fewer data points. This implies that we can use a subset of the big data to reduce the computational burden to approximate the target function well. Thus, using the technique of including a subset of big data, we proposed a new method to solve the problem of estimating a target function in large-scaled experiments. Several numerical comparisons show that our method outperforms proposed methods in recent literature for large-scaled computer experiments in terms of prediction accuracy and computational time.

In Chapter 3, motivated by a biological experiment, we propose a new method for quantifying uncertainty in biology studies. Uncertainty quantification attempts to appraise and quantify uncertainty in physical systems. However, in some physical systems, there lacks a method that can be further developed to quantify uncertainty. We were motivated by single-molecule experiments in the study of T cell signaling, where no models can be used for quantifying the features of the single-molecule experiments. To fix this problem, we developed a novel model, the varying coefficient frailty model, to quantify the uncertainty in the single-molecule experiments. The fitted varying coefficient model provides a rigorous quantification of an early and rapid impact on T cell signaling from the accumulation of bond lifetime, which can shed new light on the fundamental understanding of how T cells initiate immune responses. Theoretical properties of the estimators, including their unbiased properties near the boundary, are derived along with discussions on the asymptotic bias-variance trade-off. We can apply the model not only for single-molecule experiments, but also for survival analysis and reliability to explore time-varying effects from covariates with random effects.

In Chapter 4, we address the problem of identifying an optimal computer simulator for the observed physical experiments. In many applications, experimenters have several computer models with different scientific implications for a physical phenomenon. However, they may not know which computer model is the most optimal to describe the observed physics. An example from cell biology is that biologists have several biological models used for understanding cell adhesion between T lymphocytes and other cells, but they do

not know which biological model is most desirable for real lab data. To find the optimal model for such lab data, we propose a selection criterion based on leave-one-out cross-validation. We show that this criterion can be decomposed into a goodness-of-fit measure and a generalized degrees of freedom, capturing the complexity of the computer simulator. Asymptotic properties of the selected optimal simulator are discussed. Additionally, we show that the proposed procedure includes a conventional calibration method as special case. In the application of cell biology, an optimal simulator is selected, which gives new insight on the T cell recognition mechanism in the human immune system.

CHAPTER 1

TRANSFORMATION AND ADDITIVITY IN COMPUTER EXPERIMENTS

In this chapter, we discuss the problem of approximating a deterministic function using Gaussian Processes (GP). The role of transformation in GP modeling is not well understood. We argue that transformation of the response can be used for making the deterministic function approximately additive, which can then be easily estimated using an additive GP. We call such a GP a Transformed Additive Gaussian (TAG) process. To capture possible interactions which are unaccounted for in an additive model, we propose an extension of the TAG process called Transformed Approximately Additive Gaussian (TAAG) process. We develop efficient techniques for fitting a TAAG process. In fact, we show that it can be fitted to high-dimensional data much more efficiently than a standard GP. Furthermore, we show that the use of the TAAG process leads to better estimation, interpretation, visualization, and prediction. The proposed methods are implemented in the R package *TAG*.

1.1 Introduction

Transformation of the response is a common technique used in regression analysis, but not so much in the modeling of deterministic functions. There are many reasons for this. In regression analysis, transformations are used as a way to fix the violations in the statistical modeling assumptions such as constant variance or normality of the errors. Since there are no errors in a deterministic function, there does not seem to be any need for transformations! From a function approximation point of view, there also does not seem to be any advantage in transforming the response, and therefore transformations are rarely studied in the numerical analysis literature. To see this, suppose we are trying to approximate a function $y = f(\boldsymbol{x})$, $\boldsymbol{x} \in [0,1]^p$, using the data $\boldsymbol{y} = (y_1, \ldots, y_n)'$ observed over an experimental

design $\boldsymbol{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. We can obtain the function approximation $\hat{f}(\boldsymbol{x}|\boldsymbol{D}, \boldsymbol{y})$ directly using this data or $g^{-1}\{\widehat{g \circ f}(\boldsymbol{x}|\boldsymbol{D}, g(\boldsymbol{y}))\}$ using the transformed data, where $g(\cdot)$ denotes the transformation function, $g(\boldsymbol{y}) = (g(y_1), \ldots, g(y_n))'$, and $g \circ f(\cdot) = g\{f(\cdot)\}$. Although these two approximations can be quite different, they are asymptotically equivalent as long as the technique used for function approximation converges (see, for example, Theorem 14.5 of [1] for the conditions on point-wise convergence). Since the quality of a function approximation is assessed using its asymptotic convergence properties, the transformation does not seem to play any role in the mathematical analysis, and therefore it is ignored. Yet practitioners have found it useful to transform the response, but its usage seems to be sporadic with no proper guidelines. For example, a logarithmic transformation of the response is used for making the predictions nonnegative in the original scale, but many times at the cost of accuracy.

Gaussian process (GP) models, also known as kriging, are widely adopted for modeling deterministic functions ([2, 3]). Because of its probabilistic formulation, a case can be made for transforming the output. This approach is known by the name Trans-Gaussian kriging in spatial statistics ([4, 5]) and warped Gaussian process in machine learning ([6, 7]). However, the GP is used in modeling mainly due to its mathematical convenience and does not possess a strong justification as in the case of regression analysis. Thus, transforming the response to make its distribution look more Gaussian does seem questionable. Stationarity is another common assumption for GP modeling. However, since we observe only a single realization of the Gaussian process, assessing the validity of this assumption and achieving constancy of variance is not straightforward.

We propose transformation in GP modeling to improve additivity, that is, to find a transformation of the response so that the deterministic function becomes approximately additive in the variables. An additive function is easier to approximate, and therefore the approximation obtained using such a transformation is expected to perform better. To illustrate the idea, consider the function $f(\boldsymbol{x}) = \exp(x_1^2 + x_2^2)$. Clearly, by setting $g(y) = \log y$,

we can make this function additive. Many of the physical models such as those obtained using dimensional analysis are based on product rules ([8]), which can be made additive through a log-transform. But in general this will not work. Consider, for example, $f(\boldsymbol{x}) = 1/(x_1 + x_2 + 0.01x_1x_2)$. There is no simple transformation to make this function additive. However, by setting $g(y) = 1/y$, we can make this function approximately additive. Achieving even approximate additivity through transformation is beneficial because such a function can be well-approximated using fewer data points than what would be needed in the original untransformed scale.

Using additive models is not a new concept and has a long history in statistics (see, for example, [9]). An obvious disadvantage of additive models is that they cannot entertain higher-order interactions among the variables. [10] extended the additive modeling framework to include linear combinations of the variables, which has the ability to capture interactions. Different from previous works, we employ a GP model as the nonparametric smoother in the additive modeling framework. In this sense, our approach is closer to the additive GP models introduced by [11], but there are major differences. Their objective was to decompose the function into a sum of low-dimensional functions that include interactions, whereas our objective is to identify a transformation so that the function can be represented by a first-order low-dimensional function. The idea of transformation is also not new in additive models. [12] proposed additivity and variance stabilization (AVAS) algorithm in conjunction with additive models, but variance stabilization is not relevant to our problem because there is no error in deterministic computer experiments. Our approach is similar in spirit to the Alternating Conditional Expectation (ACE) method of [13], but differs in terms of the smoothing method used for the variables. Moreover, as we demonstrate in this paper, the use of GP models facilitates better uncertainty quantification of deterministic functions.

An argument against using transformations is that it makes the interpretation of the results difficult. This is certainly true in linear regression, where the parameter estimates

have a simple meaning which gets destroyed with transformations. However, this is not the case with GP modeling. Because of the nonlinear relationships, the results of GP models can only be interpreted/visualized by plotting the main effects obtained using a functional ANOVA decomposition. We will show that the interpretation of these main effects becomes better with transformations, and therefore what is considered a disadvantage in the regression setting becomes a blessing in GP modeling! One may also be skeptical about using additive models in deterministic computer experiments as it cannot provide interpolation unless the function is perfectly additive after transformation, which may be a rare case. We will argue in the paper that it is sufficient to achieve approximate additivity and that the additive model can be easily augmented to produce an interpolative model.

The article is organized as follows. In Section 2, we develop the main methodology for identifying transformations to make the function as additive as possible. Efficient estimation techniques for the unknown parameters in the model are developed in this section. In Section 3 we introduce approximately additive GP models which can achieve interpolation. Several advantages of the proposed method are discussed in Section 4. Its performance is tested using simulated and real examples in Section 5, and we conclude with some remarks in Section 6.

1.2 Transformed Additive Gaussian (TAG) Process

Our aim is to find a transformation for the response $g(y)$ so that the inverse transformed additive model

$$g^{-1}\{\mu + z_1(x_1) + \ldots + z_p(x_p)\}$$

is a good approximation to $y = f(\boldsymbol{x})$, where $\boldsymbol{x} = (x_1, \ldots, x_p)'$. Let

$$g(y) = \mu + z(\boldsymbol{x}) + \epsilon(\boldsymbol{x}), \tag{1.1}$$

where $z(\boldsymbol{x}) = z_1(x_1) + \ldots + z_p(x_p)$ and $\epsilon(\boldsymbol{x})$ is the approximation error. We will use a Bayesian framework to estimate the function by placing a GP prior on each $z_k(\cdot)$:

$$z_k(x_k) \overset{\text{ind.}}{\sim} GP(0, \tau_k^2 R_k(\cdot)),$$

for $k = 1, \ldots, p$, where τ_k^2 is the variance and $R_k(h) = Cor\{z_k(x), z_k(x+h)\}$ is the stationary correlation function. For the moment, we will assume $\epsilon(\boldsymbol{x}) \overset{\text{ind.}}{\sim} N(0, \sigma^2)$, which will be relaxed in the next section. Let $\tau^2 = \sum_{k=1}^p \tau_k^2$ and $\omega_k = \tau_k^2/\tau^2$. Then,

$$z(\boldsymbol{x}) \sim GP(0, \tau^2 R(\cdot)), \tag{1.2}$$

where

$$R(\boldsymbol{h}) = \sum_{k=1}^{p} \omega_k R_k(h_k) \tag{1.3}$$

with $\sum_{k=1}^p \omega_k = 1$. We call this model the *Transformed Additive Gaussian (TAG) process*.

Given the data $(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)$, we can obtain the posterior distribution of $z(\boldsymbol{x})$ as

$$z(\boldsymbol{x})|\boldsymbol{y} \sim N\left(\widehat{z}(\boldsymbol{x}), \tau^2\{1 + \delta - \boldsymbol{r}(\boldsymbol{x})'(\boldsymbol{R} + \delta \boldsymbol{I})^{-1}\boldsymbol{r}(\boldsymbol{x})\}\right), \tag{1.4}$$

where

$$\widehat{z}(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})'(\boldsymbol{R} + \delta \boldsymbol{I})^{-1}(g(\boldsymbol{y}) - \mu \mathbf{1}), \tag{1.5}$$

$\boldsymbol{r}(\boldsymbol{x})$ is the vector of correlations $(R(\boldsymbol{x} - \boldsymbol{x}_1), \ldots, R(\boldsymbol{x} - \boldsymbol{x}_n))'$, $\boldsymbol{R}$ is the correlation matrix with the ijth element $R(\boldsymbol{x}_i - \boldsymbol{x}_j)$, $\mathbf{1}$ is a vector of 1's having length n, and $\boldsymbol{I}$ is the $n \times n$ identity matrix. The variance ratio $\delta = \sigma^2/\tau^2$ is called a nugget term in GP models.

We need to specify/estimate the unknown hyper-parameters μ, τ^2, δ, and $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_p)'$ in order to use the posterior distribution. The correlation function can also have unknown parameters. A commonly used correlation function in computer experiments is the Gaus-

sian correlation function given by

$$R_k(h) = \exp(-h^2/s_k^2),$$

where s_k is an unknown length-scale parameter. Thus, we also need to estimate $\boldsymbol{s} = (s_1, \ldots, s_p)'$. Most importantly, we also need to estimate the unknown transformation function $g(\cdot)$. Here we use a parametric approach. A commonly used parametric transformation for nonnegative data ($y > 0$) is the Box-Cox transformation ([14]) given by

$$g_\lambda(y) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, \text{if } \lambda \neq 0 \\ \log y, \text{if } \lambda = 0 \end{cases}. \tag{1.6}$$

This transformation contains an unknown parameter λ. A two-parameter Box-Cox or Yeo-Johnson transformation ([15]) can be used if the data is not restricted to be nonnegative. In this paper we will focus on the foregoing one-parameter transformation, but the methods that we propose below are general and can be applied to more general cases. We now discuss the empirical Bayes estimation of all these unknown parameters: μ,τ^2, δ, $\boldsymbol{\omega}$, $\boldsymbol{s}$, and λ. Denote them by $\boldsymbol{\theta}$.

The marginal distribution of the transformed data is given by

$$g_\lambda(\boldsymbol{y})|\boldsymbol{\theta} \sim N\left(\mu\mathbf{1}, \tau^2(\boldsymbol{R} + \delta\boldsymbol{I})\right),$$

where $g_\lambda(\boldsymbol{y}) = (g_\lambda(y_1), \ldots, g_\lambda(y_n))'$. Under a noninformative prior $p(\boldsymbol{\theta}) \propto 1$, the posterior distribution of $\boldsymbol{\theta}$ is given by

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \propto \frac{1}{\tau^{2n}|\boldsymbol{R} + \delta\boldsymbol{I}|^{1/2}} \exp\{-\frac{1}{2\tau^2}[g_\lambda(\boldsymbol{y}) - \mu\mathbf{1}]'(\boldsymbol{R} + \delta\boldsymbol{I})^{-1}[g_\lambda(\boldsymbol{y}) - \mu\mathbf{1}]\} \prod_{i=1}^{n} y_i^{\lambda-1},$$

where the last term is due to the Jacobian of transformations. We can maximize $p(\boldsymbol{\theta}|\boldsymbol{y})$ to

obtain the posterior mode of $\boldsymbol{\theta}$. Its computation can be simplified as follows:

$$\begin{aligned}
\hat{\mu} &= \frac{\mathbf{1}'(\boldsymbol{R}+\delta\boldsymbol{I})^{-1}g_\lambda(\boldsymbol{y})}{\mathbf{1}'(\boldsymbol{R}+\delta\boldsymbol{I})^{-1}\mathbf{1}}, && (1.7)\\
\hat{\tau}^2 &= \frac{1}{n}(g_\lambda(\boldsymbol{y})-\hat{\mu}\mathbf{1})'(\boldsymbol{R}+\delta\boldsymbol{I})^{-1}(g_\lambda(\boldsymbol{y})-\hat{\mu}\mathbf{1}), && (1.8)\\
(\hat{\delta},\hat{\lambda},\hat{\boldsymbol{s}},\hat{\boldsymbol{\omega}}) &= \underset{\delta,\lambda,\boldsymbol{s},\boldsymbol{\omega}}{\arg\min}\, n\log\hat{\tau}^2+\log|\boldsymbol{R}+\delta\boldsymbol{I}|-2(\lambda-1)\sum_{i=1}^{n}\log y_i. && (1.9)
\end{aligned}$$

The last optimization is performed under the constraints $\delta > 0$, $\lambda \in [-2, 2]$, $\boldsymbol{s} > \mathbf{0}$, $\boldsymbol{\omega} \geq \mathbf{0}$, and $\sum_{k=1}^{p} \omega_k = 1$. We chose $[-2, 2]$ as the possible range of λ ([16], p. 134), but other ranges can also be used.

The foregoing nonlinear optimization in (1.9) is the most time-consuming step in fitting a TAG process. In a traditional GP model with Gaussian product correlation function, this optimization needs to be performed only in a p-dimensional space of the length-scale parameters, which is much easier than the $(2p + 2)$-dimensional optimization of the TAG process. Fortunately, we can speed up the estimation by taking advantage of the additive structure of the correlation function.

Let $\hat{\boldsymbol{c}} = (\boldsymbol{R}+\delta\boldsymbol{I})^{-1}(g_\lambda(\boldsymbol{y})-\mu\mathbf{1})$. Then, (1.5) can also be written as the sum of n basis functions:

$$\widehat{z}(\boldsymbol{x}) = \sum_{i=1}^{n} \hat{c}_i R(\boldsymbol{x}-\boldsymbol{x}_i).$$

Now because of the additive structure of the correlation function, we obtain

$$\begin{aligned}
\widehat{z}(\boldsymbol{x}) &= \sum_{i=1}^{n}\hat{c}_i\sum_{k=1}^{p}\omega_k R_k(x_k-x_{ik})\\
&= \sum_{k=1}^{p}\omega_k\sum_{i=1}^{n}\hat{c}_i R_k(x_k-x_{ik})\\
&= \sum_{k=1}^{p}\hat{z}_k(x_k). && (1.10)
\end{aligned}$$

This suggests that we can estimate the parameters by fitting an additive model using the

Algorithm 1 Estimation of the Transformed Additive Gaussian (TAG) process

procedure TAG($\{\boldsymbol{x}_i, y_i\}_{i=1}^n, \mathbf{D}$) ▷
    Obtain initial estimates $\boldsymbol{\omega}^{(0)}, \boldsymbol{s}^{(0)}, \delta^{(0)}, \lambda^{(0)}$ using Algorithm 10.
    Obtain $(\hat{\boldsymbol{\omega}}, \hat{\mathbf{s}}, \hat{\delta}, \hat{\lambda})$ from (1.9) using nonlinear optimization with $\hat{\mu}$ from (1.7) and $\hat{\tau}^2$ from (1.8).
    return $(\hat{\boldsymbol{\omega}}, \hat{\mathbf{s}}, \hat{\delta}, \hat{\lambda})$.
end procedure

efficient backfitting algorithm ([13]). The basis functions in the additive model should be chosen based on the correlation function. In our case, we should use a Gaussian basis function, which is not typical in additive model fitting where smoothing splines are commonly used. Therefore, to make use of the available software, we perform this estimation in two steps. First we fit an additive model using the *mgcv* package ([17]) in R for each value of $\lambda \in \{-2, -1.5, \ldots, 1.5, 2\}$ and choose the λ to minimize the generalized cross-validation error. This gives us $\hat{\lambda}$ and $\tilde{z}_k(x_k)$, where $\tilde{z}_k(x_k)$ is an approximation of $\hat{z}_k(x_k)$, for $k = 1, \ldots, p$. Now choose m equally spaced points in $[0, 1]$ given by $\boldsymbol{D} = \{0, 1/(m-1), \ldots, 1\}$. Let $\tilde{\boldsymbol{z}}_k$ be the predictions using $\tilde{z}_k(\cdot)$ at these m points. Then $\hat{\omega}_k \approx \widehat{var}(\tilde{\boldsymbol{z}}_k) / \sum_{k=1}^p \widehat{var}(\tilde{\boldsymbol{z}}_k)$. Furthermore, we can fit p one-dimensional GPs with the chosen correlation function to the datasets $\{\boldsymbol{D}, \tilde{\boldsymbol{z}}_k\}$ for $k = 1, \ldots, p$ using standard packages such as *DiceKriging* ([18]) in R. This gives estimates of $\boldsymbol{s}$. The details are described in the Appendix B. We can use these as initial estimates for the nonlinear optimization in (1.9). This considerably speeds up the optimization. The traditional GP fitting requires global optimization or many local optimizations with multiple starting values. Because in TAG we can quickly obtain good initial estimates, only a single local optimization is needed to obtain the empirical Bayes estimates, and therefore the fitting can be much faster than that of the standard GP fitting. The whole procedure is summarized in Algorithm 1. All of the algorithms in this article are implemented in the R package *TAG* ([19]) .

1.3 Transformed Approximately Additive Gaussian (TAAG) Process

Even with the best possible transformation, we may not be able to make the function additive and thus the approximation that we obtain using TAG can be unsatisfactory. In this section, we propose a simple extension of TAG to improve the approximation.

Remember that we took $\epsilon(\boldsymbol{x})$ in (1.1) to be independent and identically distributed as $N(0, \sigma^2)$. We only need to make this into a smooth GP to improve the approximation. So let $\epsilon(\boldsymbol{x}) \sim GP(0, \sigma^2 L(\cdot))$, where $L(\cdot)$ is a positive definite stationary correlation function. Thus the GP model becomes

$$g_\lambda(y) \sim GP(\mu, \nu^2\{(1-\eta)R(\cdot) + \eta L(\cdot)\}), \tag{1.11}$$

where $\nu^2 = \tau^2 + \sigma^2$, $\eta = \sigma^2/(\sigma^2 + \tau^2) \in [0, 1]$ and $R(\cdot)$ is as in (1.3). The resulting predictor is only approximately additive. Therefore we call this model the *Transformed Approximately Additive Gaussian (TAAG) process*. [20] proposed a closely related GP model, but the motivation behind TAAG process and its estimation techniques are completely different. TAAG process is also related to some of the other ideas proposed in the literature such as that of using a convex combination of GPs ([21]) and composite GPs ([22]).

For $L(\cdot)$, we can use the *product* Gaussian correlation function given by

$$L(\boldsymbol{h}) = \prod_{k=1}^{p} L_k(h_k; \gamma_k) = \prod_{k=1}^{p} \exp\left(-\frac{h_k^2}{\gamma_k^2}\right).$$

where γ_k is an unknown length-scale parameter. Thus, the whole set of unknown parameters in the TAAG model becomes $(\mu, \nu^2, \lambda, \boldsymbol{\omega}, \boldsymbol{s}, \boldsymbol{\gamma}, \eta)$. This is a lot of parameters to estimate and can be a difficult task. Moreover, there can be identifiability issues with the correlation parameters in $R(\cdot)$ and $L(\cdot)$. To overcome these issues, we propose to fix the TAG parameters $(\lambda, \boldsymbol{\omega}, \boldsymbol{s})$ at the estimates obtained earlier using Algorithm 1 and estimate

Algorithm 2 Estimation of the Transformed Approximately Additive Gaussian (TAAG) Process

procedure TAAG($\{\boldsymbol{x}_i, y_i\}_{i=1}^n$) ▷
    Obtain $\hat{\delta}$, $\hat{\lambda}$, $\hat{\boldsymbol{\omega}}$, and $\hat{\boldsymbol{s}}$ using Algorithm 1.
    Obtain $\hat{\boldsymbol{\gamma}}$ by fitting a standard GP model to the data $g_{\hat{\lambda}}(\boldsymbol{y})$.
    Obtain $\hat{\eta}$ from (1.12) using a one-dimensional optimization with $\hat{\mu}$ and $\hat{\nu}^2$ obtained from (1.13) and (1.14), respectively.
    return $\hat{\eta}$, $\hat{\boldsymbol{\gamma}}$, $\hat{\mu}$, and $\hat{\nu}^2$.
end procedure

only the remaining parameters.

To get further simplification, we estimate the length-scale parameters $\boldsymbol{\gamma}$ in $L(\cdot)$ by fitting a GP on $g_{\hat{\lambda}}(y)$. This has the added benefit that the TAAG process can reduce to a standard GP with no transformations when $\eta = 1$ and $\lambda = 1$. Thus, the standard GP becomes a special case of TAAG and therefore, one does not have to make a choice between TAAG process and a standard GP beforehand. With these simplifications, we only need to estimate μ, ν^2, and η. To encourage additive modeling, we use a beta prior on $\eta \sim Beta(\hat{\delta} + 1, 2)$, where $\hat{\delta}$ is obtained from Algorithm 1. The hyperparameters in the beta prior are chosen so that the mode of η is $\hat{\delta}/(1 + \hat{\delta})$, which would be its estimate if we were to use a TAG process. Thus, the empirical Bayes estimate of η can be obtained as

$$\hat{\eta} = \arg\min_{\eta} \log|(1-\eta)\widehat{\boldsymbol{R}} + \eta\boldsymbol{L}| + n\log\hat{\nu}^2 - 2\log\{\eta^{\hat{\delta}}(1-\eta)\}, \tag{1.12}$$

where

$$\hat{\mu} = \frac{\mathbf{1}'\{(1-\eta)\widehat{\boldsymbol{R}} + \eta\boldsymbol{L}\}^{-1} g_{\hat{\lambda}}(\boldsymbol{y})}{\mathbf{1}'\{(1-\eta)\widehat{\boldsymbol{R}} + \eta\boldsymbol{L}\}^{-1}\mathbf{1}}, \tag{1.13}$$

$$\hat{\nu}^2 = \frac{1}{n}(g_{\hat{\lambda}}(\boldsymbol{y}) - \hat{\mu}\mathbf{1})'\{(1-\eta)\widehat{\boldsymbol{R}} + \eta\boldsymbol{L}\}^{-1}(g_{\hat{\lambda}}(\boldsymbol{y}) - \hat{\mu}\mathbf{1}), \tag{1.14}$$

$\widehat{\mathbf{R}}$ is obtained by plugging $\hat{\boldsymbol{\omega}}$ and $\hat{\boldsymbol{s}}$ from Algorithm 1 in $\mathbf{R}$, and $\boldsymbol{L}$ is the $n \times n$ matrix with ijth element $L(\boldsymbol{x}_i - \boldsymbol{x}_j)$. The estimation procedure is shown in Algorithm 2.

The prediction and uncertainty quantification can be done as follows. The posterior

distribution of $g(f(\boldsymbol{x}))$ for given μ and ν^2 is given by

$$g \circ f(\boldsymbol{x})|\boldsymbol{y}, \mu, \nu^2 \sim N(\widehat{g \circ f}(\boldsymbol{x}), V(\boldsymbol{x})), \tag{1.15}$$

where

$$\begin{aligned}
\widehat{g \circ f}(\boldsymbol{x}) &= \mu + \{(1-\eta)\boldsymbol{r}(\boldsymbol{x}) + \eta \boldsymbol{l}(\boldsymbol{x})\}'\{(1-\eta)\boldsymbol{R} + \eta \boldsymbol{L}\}^{-1}(g(\boldsymbol{y}) - \mu \mathbf{1}), \\
V(\boldsymbol{x}) &= \nu^2 \left[1 - \{(1-\eta)\boldsymbol{r}(\boldsymbol{x}) + \eta \boldsymbol{l}(\boldsymbol{x})\}'\{(1-\eta)\boldsymbol{R} + \eta \boldsymbol{L}\}^{-1}\{(1-\eta)\boldsymbol{r}(\boldsymbol{x}) + \eta \boldsymbol{l}(\boldsymbol{x})\}\right],
\end{aligned}$$

where $\boldsymbol{l}(\boldsymbol{x})$ is an $n \times 1$ vector with jth element $L(\boldsymbol{x} - \boldsymbol{x}_j)$. We can either plug-in the estimates of μ and ν^2 or integrate them out ([3]), but for simplicity we will use the plug-in approach. From (1.15), we can obtain the probability density function of $f(\boldsymbol{x})|\boldsymbol{y}, \mu, \nu^2$ as

$$|\dot{g}(f(\boldsymbol{x}))| \frac{1}{\sqrt{2\pi V(\boldsymbol{x})}} \exp[-\{g \circ f(\boldsymbol{x}) - \widehat{g \circ f}(\boldsymbol{x})\}^2 / \{2V(\boldsymbol{x})\}],$$

where $\dot{g}(\cdot)$ is the derivative of $g(\cdot)$. In general, this can be a nonstandard distribution and computing its mean and variance may require numerical integration. [4] derives an approximate expression for the mean using Taylor series expansion. However, as [6] pointed out, it is much easier to use the median, which is given by

$$\tilde{f}(\boldsymbol{x}) = g^{-1}\left\{\widehat{g \circ f}(\boldsymbol{x})\right\}.$$

Similarly, we can obtain a 95% credible interval for the prediction as

$$\left[g^{-1}\left\{\widehat{g \circ f}(\boldsymbol{x}) - 1.96\sqrt{V(\boldsymbol{x})}\right\}, g^{-1}\left\{\widehat{g \circ f}(\boldsymbol{x}) + 1.96\sqrt{V(\boldsymbol{x})}\right\}\right].$$

Note that when using a Box-Cox transformation (1.6), we need constraints $y > 0$ and $\lambda g_\lambda(y) + 1 > 0$ to make sure that $g(\cdot)$ is one-to-one. Therefore we force the lower bound to be 0 if $\lambda\{\widehat{g \circ f}(\boldsymbol{x}) - 1.96\sqrt{V(\boldsymbol{x})}\} + 1 < 0$.

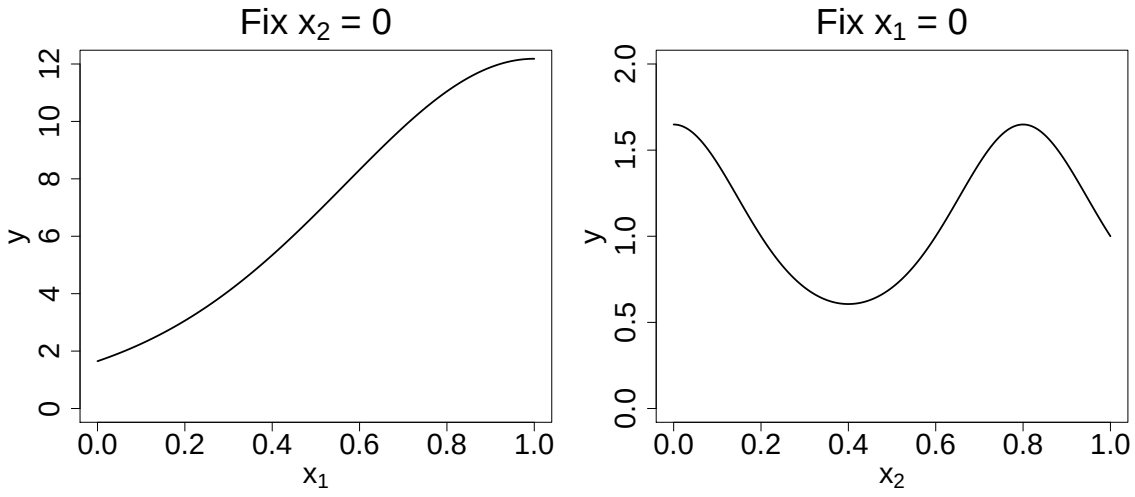

Figure 1.1: Marginal plots of the function in (1.16).

1.4 Some Advantages of TAAG

In this section we discuss the many advantages of TAAG using numerical examples.

1.4.1 New correlation function

Although our aim was not to develop a new correlation function, the one that came out of our modeling

$$(1-\eta)\sum_{k=1}^{p}\omega_k R_k(h_k;s_k)+\eta\prod_{k=1}^{p}L_k(h_k;\gamma_k)$$

is of independent interest. We will show that its parameters have better interpretability than those of the existing correlation functions in the literature. Consider a simple function

$$y=\exp\left\{2\sin(0.5\pi x_1)+0.5\cos(2.5\pi x_2)\right\} \tag{1.16}$$

where $\boldsymbol{x}\in[0,1]^2$. The marginal plots of the function are shown in Figure 1.1.

Suppose we generate data using a randomized Sobol' sequence ([23]) of $n=20$ points. For comparison, first we fit a standard GP using the commonly used Gaussian product correlation function $\exp(-\sum_{k=1}^{2}h_k^2/s_k^2)$. We used the R package *mlegp* ([24]) for obtaining

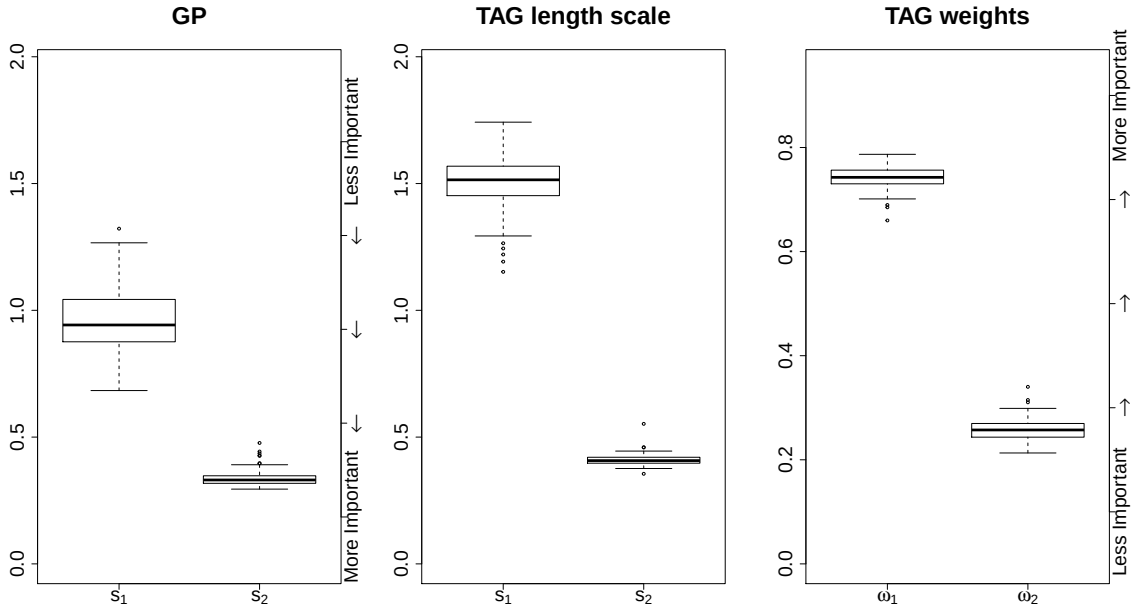

Figure 1.2: Parameter estimates for GP and TAG process for the function in (1.16).

the maximum likelihood estimates of the parameters. The left panel of Figure 1.2 shows boxplots of s_1 and s_2 from 100 repetitions of the randomized Sobol' sequence. Larger values of s_1 seem to suggest that x_1 is less important than x_2 ([25, 26, 27, 28, 29]; [3], Section 7.6; [30], p. 106). On the contrary, Figure 1.1 seems to imply that x_1 is more important than x_2 in terms of the contribution to response variability. This contradiction happened here because the function wiggles more with respect to x_2 resulting in a smaller value of s_2.

Now consider the new correlation function in (1.11). The middle panel of Figure 1.2 shows the boxplots of s_1 and s_2 in the TAG process estimated from the same 100 randomized Sobol' sequences. The right panel shows the boxplots of ω_1 and ω_2. In the new correlation function, s_i's can be used to understand how wiggly the function is, and ω_i's can be used to understand the importance of the variables. Since s_1 is larger than s_2, the function is expected to be less wiggly in x_1 than x_2, which agrees with Figure 1.1. Moreover, since ω_1 is more than ω_2, TAG process correctly identifies x_1 to be more important than x_2. Note that the interpretation of ω_i's as importance parameters is meaningful only

when the function is additive, that is, when $\eta \approx 0$. When η is large, the interpretation becomes approximate, and one should compute measures such as Sobol' sensitivity indices to understand the exact importance of variables.

There are other correlation functions proposed in the literature with more parameters such as the power exponential and Matérn correlation functions. However, the extra parameters in them only control the smoothness or roughness of the function. The function considered in this example is very smooth and infinitely differentiable in both the variables and therefore, those correlation functions cannot rectify the confounding issues between scale and importance. One possible approach to introduce importance parameters in a standard GP model is to add a mean function containing a linear combination of basis functions, but it is not clear what basis functions should be used ([31, 32]). On the other hand, TAG process can be viewed as providing these basis functions automatically through the additive correlation function, thereby avoiding the need to specify them through a mean function. This viewpoint also clarifies why we do not need to use a mean function in a TAG/TAAG process.

1.4.2 Prediction performance

TAAG is expected to perform well in the example function in (1.16) because it becomes perfectly additive under log-transformation. So consider a slightly modified version

$$\exp\{2\sin(0.5\pi x_1) + 0.5\cos(2.5\pi x_2)\} + 0.25\sin(\pi x_1)\cos(0.5\pi x_2), \tag{1.17}$$

which cannot be made additive through transformation. We will use this function to assess the prediction performance of the TAAG process.

As before, we generate data using a randomized Sobol' sequence of $n = 20$ points and fit TAG and TAAG processes. Predictions are made on 1,000 test points in $[0, 1]^2$, and the root mean squared prediction error (RMSPE) is computed. This is repeated 100 times

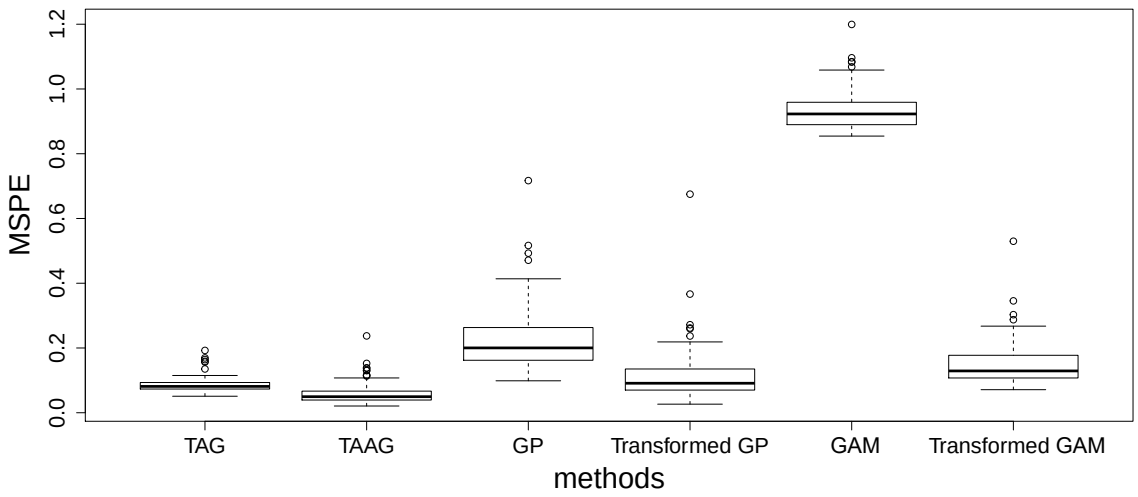

Figure 1.3: Prediction performance of TAG, TAAG, GP, transformed GP, AM, and transformed AM using the example function in (1.17).

by generating a new randomized Sobol' sequence each time. The resulting RMSPEs are shown as boxplots on the left side of Figure 1.3. We also fitted the commonly used GP with product Gaussian correlation function on the original data as well as the transformed data. Their RMSPEs are also shown in the same figure denoted as "GP" and "Transformed GP", respectively. As a further check, we also fitted an additive model on the original data and the transformed data ("AM" and "Transformed AM") using the R package *mgcv*. We can see that AM does not perform well, but surprisingly the transformed AM does well, even better than GP. This clearly shows the benefit of transformations. On the other hand, TAG improves over the Transformed AM and Transformed GP. TAAG performs better than TAG and seems to be the best among the six methods.

As mentioned before, uncertainty quantification is one of the main advantages of the TAG/TAAG processes. To assess their performance, we computed the interval score ([33]), which is defined as $(u-\ell)+(2/\alpha)(\ell-x)I\{x<\ell\}+(2/\alpha)(x-u)I\{x>u\}$ with $\alpha=95\%$. A smaller interval score indicates a better prediction interval. The interval scores for the 100 simulation cases are shown as boxplots in Figure 1.4. Clearly, TAAG is again the best among the six methods.

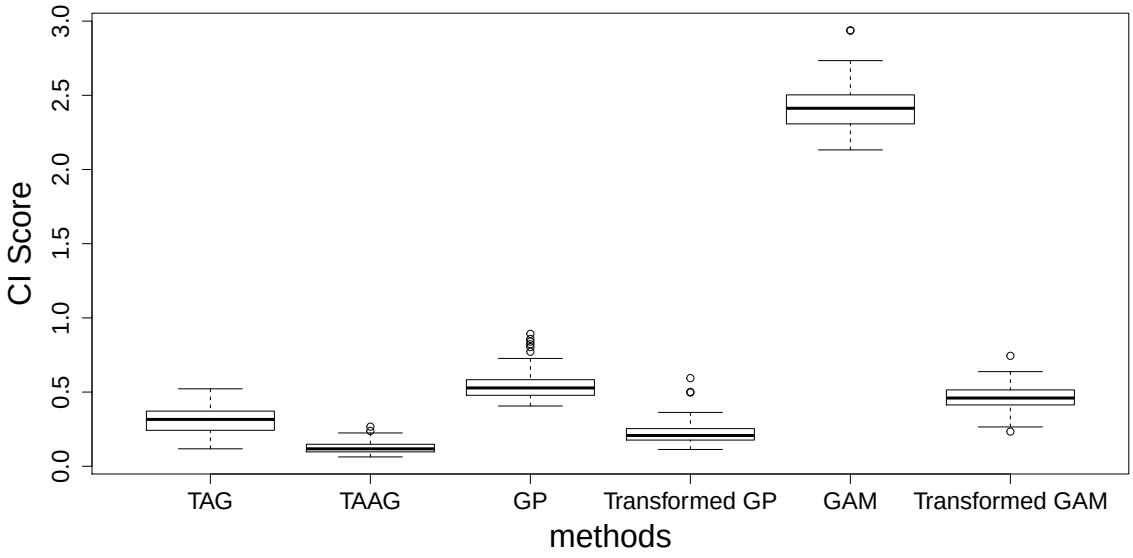

Figure 1.4: Interval scores (the smaller the better) of TAG, TAAG, GP, transformed GP, AM, and transformed AM using the example function in (1.17).

1.4.3 Interpretation and visualization

Another advantage of the TAAG process is that it enables better interpretation and visualization of the effects. As illustrated in Section 4.1, ω_i's can be used to quickly understand the importance of each variable, which represents the first-order Sobol' indices when the transformed response is perfectly additive ([34]). If $f(\boldsymbol{x})$ is not additive, η will be greater than 0 and its value can be used to understand the overall interaction effect. Moreover, the main effects of the variables in the transformed scale can be quickly visualized using

$$\hat{z}_k(x_k) = \omega_k \sum_{i=1}^{n} \hat{c}_i R_k(x_k - x_{ik}),$$

which does not require any extra computations (see (1.10)). On the other hand, one needs to use the computationally intensive functional ANOVA decomposition to get the main effects if we were to fit a standard GP. Of course, the main effects are meaningful only if there are no higher-order interactions. Because we use transformation to minimize the interaction effects, the main effects that we obtain using TAAG process are more trustworthy.

Table 1.1: The ω_i's and s_i's from the TAAG process for the Borehole function and the first-order Sobol' indices of the log-borehole function

Input variables	r_w	r	T_u	H_u	T_l	H_l	L	K_w
$\boldsymbol{\omega}$	0.878	0.002	0.002	0.038	0.002	0.038	0.033	0.009
$\boldsymbol{s}$	1.500	0.918	1.594	2.641	0.969	2.601	2.295	1.861
First-order Sobol' indices	0.889	0.000	0.000	0.036	0.000	0.035	0.032	0.008

We illustrate the foregoing advantages using the borehole function ([35]):

$$y = \frac{2\pi T_u (H_u - H_l)}{\log(\frac{r}{r_w}) \left[1 + \frac{2LT_u}{\log(\frac{r}{r_w}) r_w^2 K_w} + \frac{T_u}{T_l}\right]},$$

where the ranges for the eight variables are $r_w : (0.05, 0.15)$, $r = (100, 50000)$, $T_u = (63070, 115600)$, $H_u = (990, 1110)$, $T_l = (63.1, 116)$, $H_l = (700, 820)$, $L = (1120, 1680)$, and $K_w = (9855, 12045)$. Suppose we generate $n = 80$ data using the MaxPro design ([36]) and fit the TAAG process. It identified a log-transform for the response ($\hat{\lambda} = 0$). The ω_i's and s_i's from the fit are given in Table 1.1. The first-order Sobol' indices of the log-borehole function is also given in the same table. We can see that ω_i's are very close to the first-order Sobol' indices.

The centered main effects $\hat{\mathbf{z}}_k(x_k) - \bar{z}_k$, where $\bar{z}_k$ is the mean value of $\hat{\mathbf{z}}_k(\cdot)$, are shown in the panel (a) of Figure 1.5. The panels (b) and (c) of Figure 1.5 show the main effects computed directly from the borehole function with and without logarithmic transformation. We can see that the TAAG process approximates the main effects of the transformed response quite well. Moreover, $\hat{\eta} = 0.0392$ is very small, indicating that the interaction effects are negligibly small in the transformed scale. We can use the difference of Sobol's total index and first-order index to better understand the interaction effects. This is shown in Figure 1.6 for the original and transformed responses. We can see that the log-transformation has greatly helped in reducing the interaction effects. This clearly shows that the main effects plots of $\log y$ are much more meaningful to look at than those of y.

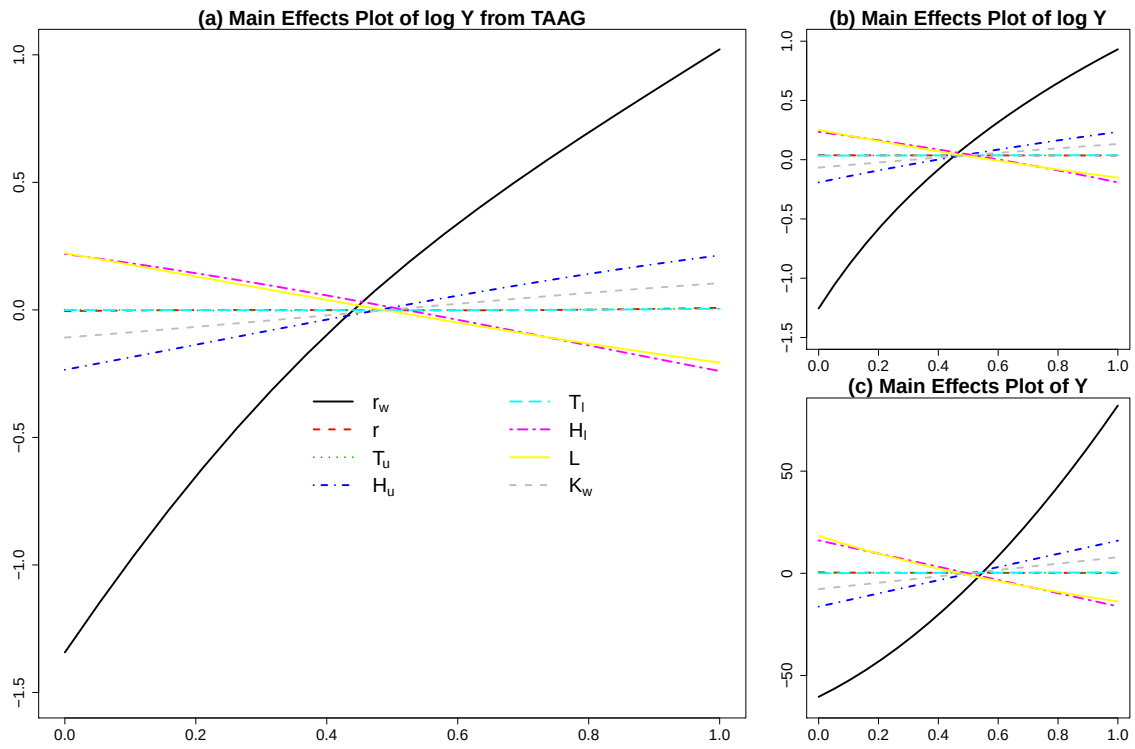

Figure 1.5: Borehole function example: (a) The main effects from TAAG with logarithmic transformation of the response, (b) the true main effects with logarithmic transformation of the response, and (c) the true main effects for the original (untransformed) response.

Note that we have centered the main effects $\hat{\mathbf{z}}_k(x_k) - \bar{z}_k$ before plotting, otherwise they will have different means and will be difficult to visualize. This indicates a possible identifiability issue present between $z_k(x_k)$ and μ. [37] chose a particular kernel that is orthogonal to the mean so that such identifiability issues can be avoided in fitting their Bayesian smoothing spline ANOVA model. [38] show how any given correlation function can be made orthogonal to the mean. Thus, using orthogonal GPs for $z_k(x_k)$'s can possibly avoid the identifiability issue that we have observed in TAG. However, this can unnecessarily complicate the modeling and estimation procedure. In our experience, the orthogonality is not needed if our aim is prediction. It becomes important only when we need to have a physical interpretation of μ. Thus, we will not use orthogonal GPs in our modeling and will use the foregoing simple fix of centering for visualizing the main effects.

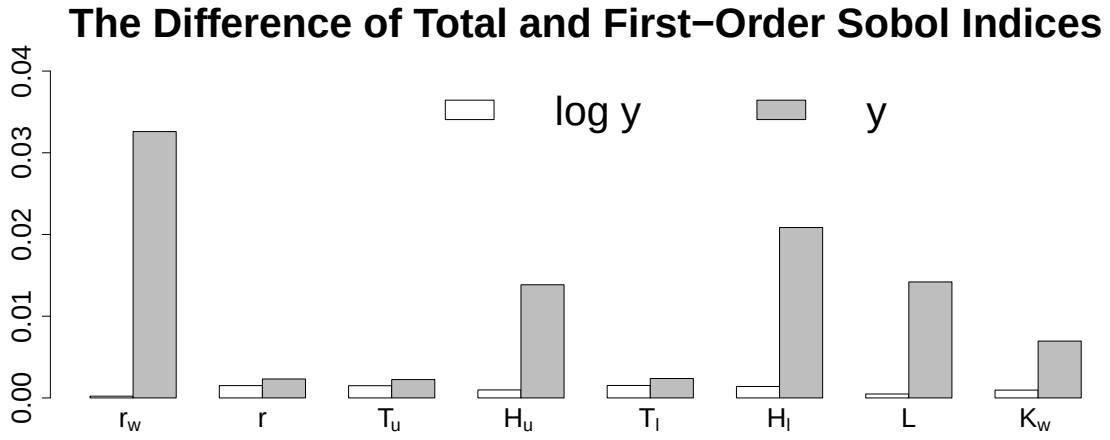

Figure 1.6: The difference of Sobol's total index and first-order index for the original and log-transformed responses in the borehole example. Large difference indicates large interaction effects.

1.4.4 High-dimensional data

Fitting GP models data with large n and p is always a challenging problem. This is because the likelihood function requires the inversion of the correlation matrix, whose computational complexity is $O(n^3)$. Moreover, thousands of evaluations of the likelihood function is needed to optimize it, especially in high dimensions. To understand the computational complexity with respect to the number of dimensions, first note that $O(n^2p)$ computations are needed to construct the correlation matrix. Consider a gradient-based optimization with a fixed number of iterations. Once the correlation matrix is inverted, the gradient of the likelihood can be calculated in $O(n^2p)$ ([30], pp.114). So the total computational cost is still $O(n^3 + n^2p)$. However, because the likelihood is likely to be multimodal, the number of initial points for optimization should be at least $O(p)$ to have a fair chance of finding the global optimum ([39]). Thus the computational complexity of optimizing the likelihood is at least $O(n^3p + n^2p^2)$. Since n should be increased at least proportional to p to get a meaningful approximation, the computational complexity with respect to p is at least $O(p^4)$, which can be quite heavy for large p.

Much of the recent research in GP modeling has focused on the large n problem, for example, using iterative kriging ([40]) and local GPs ([41]). However, we are not aware of

Algorithm 3 Fitting TAAG with high-dimensional data

1: **procedure** TAAG($\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, D) ▷
2: Obtain $\hat{\boldsymbol{\omega}}^{(0)}$, $\hat{\boldsymbol{s}}^{(0)}$, and $\hat{\lambda}^{(0)}$ using Algorithm 10.
3: Obtain $(\hat{\kappa}, \hat{\delta})$ by optimizing $n \log \hat{\tau}^2 + \log |\boldsymbol{R} + \delta \boldsymbol{I}|$ with $\hat{\tau}^2$ from (1.8), $\lambda = \lambda^{(0)}$, $\boldsymbol{\omega} = \boldsymbol{\omega}^{(0)}$, and $\boldsymbol{s} = \kappa \boldsymbol{s}^{(0)}$.
4: Obtain $(\hat{\eta}, \hat{\phi})$ by optimizing (1.12), where $\boldsymbol{\gamma} = \phi \boldsymbol{s}^{(0)}$.
5: **return** $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}^{(0)}$, $\hat{\boldsymbol{s}} = \hat{\kappa} \boldsymbol{s}^{(0)}$, $\hat{\lambda} = \lambda^{(0)}$, $\hat{\delta}$, $\hat{\eta}$, and $\hat{\boldsymbol{\gamma}} = \hat{\phi} \boldsymbol{s}^{(0)}$.
6: **end procedure**

any attempts to extend GP fitting to large p problems. The additive GP model framework introduced here offers a pathway to fit high-dimensional GP models efficiently. The key idea is that the additive structure of the model will allow us to fit p one-dimensional GPs instead of the one p-dimensional GP. These one-dimensional GPs can be fitted efficiently using only m points, where $m << n$ (see Step 4 of Algorithm 10).

The main time consuming step of Algorithm 1 is the $(2p + 2)$-dimensional optimization in (1.9). However, as we noted earlier, we have good initial estimates of $\boldsymbol{s}$ obtained by fitting p one-dimensional GPs to the additive functions estimated by the back fitting algorithm. So we let $\lambda = \lambda^{(0)}, \boldsymbol{\omega} = \boldsymbol{\omega}^{(0)}$, and $\boldsymbol{s} = \kappa \boldsymbol{s}^{(0)}$, where $\kappa \in (0, \infty)$ is an unknown parameter and $\boldsymbol{s}^{(0)}$ is obtained from Algorithm 10. Thus, the $(2p + 2)$-dimensional optimization reduces to a two-dimensional optimization, which is manageable. This considerably simplifies Algorithm 1. Similarly, in Algorithm 2, instead of obtaining $\hat{\boldsymbol{\gamma}}$ from a standard GP, we can use $\boldsymbol{\gamma} = \phi \boldsymbol{s}^{(0)}$, where $\phi \in (0, \infty)$, and then finding their estimates by optimizing (1.12). Of course, avoiding the optimization over the full $\boldsymbol{s}$ and $\boldsymbol{\gamma}$ can deteriorate the performance, but we found that little is lost by doing this. We summarize the procedure in Algorithm 3.

To illustrate the idea, consider the function

$$y = \prod_{i=1}^{p} \frac{|4x_i - 2| + a_i}{1 + a_i},$$

where $a_i = i/2, i = 1, 2, \cdots, p$, with $p = 10, 20, 30, \cdots, 100$ and $n = 10p$. The number of

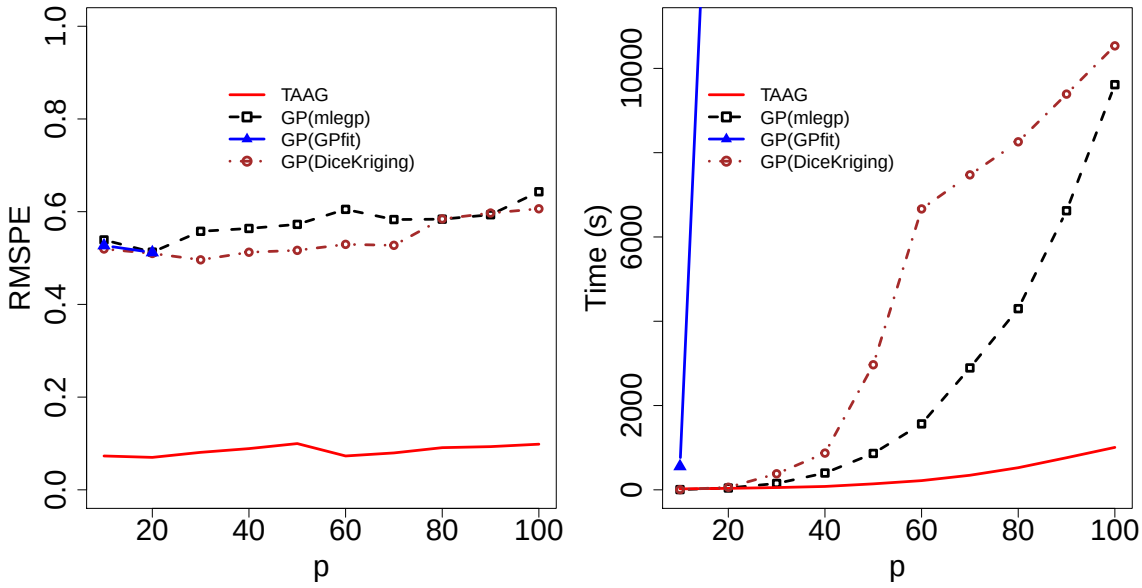

Figure 1.7: Computational time and root mean squared prediction errors of the TAAG process and GP.

testing points is $100p$. Both the training and testing designs are generated using randomized Sobol' sequence. Besides the simplifications mentioned in the previous paragraph, we use the R function *bam* in package *mgcv* in Algorithm 10, which is similar to *gam* except that the numerical methods are designed for large datasets. The left panel of Figure 1.7 shows the RMSPEs of GP and TAAG process and the right panel shows the total computational time for estimation and prediction of mean on a 2.6 GHz laptop. For fitting GP, we use the standard R packages *mlegp* ([24]) , *GPfit* ([39]), and *DiceKriging* ([18]). To make the comparisons fair, we set the number of initial points for optimization in *DiceKriging* to be $2p$, which is the default in *GPfit*. The time taken by GPfit for $p = 30$ is very high (56 hours), so we did not run it for $p > 30$. We can see that the RMSPEs of TAAG process are smaller than those from GP for all $p = 10, 20, \ldots, 100$ and the computational time saving increases with p. For example, in the 100-dimensional case, it takes about 2.7 hours for *mlegp* compared to only 20 minutes using the TAAG process.

1.5 More Examples

In this section, we compare TAAG with GP over a broad class of deterministic computer simulation examples. We took five functions from [42]: Franke function, OTL circuit function, piston simulation function, robot arm function, and wing weight function. Our experiment with these functions consists of $10p$ runs of simulations using maximum projection designs ([36]). We then fitted TAAG and a standard GP using *DiceKriging* ([18]). The predictions are compared using 1,000 Sobol' points obtained with scrambling. The root mean squared prediction errors (RMSPEs) over these 1,000 points are given in Table 1.2. We can see that TAAG uniformly performs better than GP, which is not surprising because GP is just a special case of TAAG.

Table 1.2 also includes the estimates of λ and η in TAAG. We can see that four out of five cases used a transformation of the response. Small $\hat{\eta}$ values indicate that the function becomes approximately additive after transformation. In fact, the piston simulation and wing weight functions become almost exactly additive after a log-transform, which is quite surprising. Not much improvement is achieved for the Franke and Robot Arm functions, but nothing is lost either. So in our opinion, there is no need to make a choice between TAAG and a standard GP, say for example, using cross validation methods. We can always use TAAG which has the added benefits of better interpretation and visualization.

We also compared TAAG and GP on two heat exchanger simulators (detailed and approximate) discussed in ([43]). The RMSPE computation over a 14-run validation dataset given in ([43]) is shown in the last two rows of Table 1.2. We observe that the prediction performance of TAAG is again better than that of GP with more gain observed for the approximate heat exchanger simulator.

Table 1.2: Summary of the results of the examples in Section 5.

		RMSPE		Estimates from TAAG	
Examples	dimension	GP	TAAG	$\hat{\lambda}$	$\hat{\eta}$
Franke	2	0.035	0.031	-0.5	.178
OTL Circuit	6	0.046	0.025	0.5	.021
Piston Simulation	7	0.012	0.0002	0	.0000
Robot Arm	8	0.035	0.028	1	0.131
Wing Weight	10	2.892	0.188	0	.0002
HE (Approximate)	4	4.436	2.089	0.5	.001
HE (Detailed)	4	2.217	1.937	1	.012

1.6 Conclusions

In this article, we have shown that using transformation on the response can be highly beneficial in GP modeling. It can make the deterministic function approximately additive, which can be efficiently approximated using simpler models such as additive models. By exploiting the underlying additive structure, we have developed efficient estimation techniques for fitting the transformed additive GP model. In fact, it can be fitted using a few one or two dimensional optimizations with initializations provided by the well-known back fitting algorithm. The estimation is so efficient that it can be applied to high-dimensional problems which otherwise would not have been possible with the standard GP models. The development has also led to a new correlation function with much more interpretable parameters than the commonly used correlation functions such as Gaussian or Matérn. The fitted models can be immediately visualized using main effects plots, which is another advantage of the proposed method. Moreover, the main effects plots are more meaningful in TAG/TAAG processes compared to the usual GP because of the minimization of the interaction effects.

Although we have focused on deterministic functions, the method can be extended to noisy data. Gaussian noise can be addressed by adding a nugget term in the TAAG process, but more work is needed for non-Gaussian data, which we leave as a topic for

future research. Another important direction for future research is regarding the method of transformation. Here we have used the one-parameter Box-Cox transformation, which worked well in the examples we have tried so far. However, we anticipate that, in more complex problems, a nonparametric transformation may perform better. The nonparametric transformation needs to be monotonic and easily invertible, which makes this extension nontrivial.

CHAPTER 2

TRANSFORMATION AND ADDITIVITY FOR MODELING BIG DATA

Although transformations are widely used in statistics, their usage for big data is overlooked in the literature. For approximating a function in large-scale computer experiments, we find that using transformations for improving the additivity of the function is beneficial. After improving the additivity, the target function is easier to approximate by an additive function and is expected to be well-approximated using few data points. Thus, we propose approximating the function by fitting a transformed additive model (TAM) to the subset of large-scale experiments. To capture interactions that are unaccounted for in an additive model, we propose another new method, namely, transformed approximately additive modeling (TAAM) technique. TAAM further improves the prediction performance of TAM without significantly increasing the computational cost. Several numerical comparisons show that TAAM outperforms proposed methods in recent studies for large-scale computer experiments in terms of prediction accuracy and computational time. Furthermore, the method is applied to the modeling problem in nonparametric multivariate regressions with big data.

2.1 Introduction

In science, engineering, and bio-medicine, computer experiments are becoming increasingly important in studying an extremely complex system. These systems are usually described by a complex mathematical model or a computer model implemented in large computer codes. To discover and understand the system, researchers may require a large-scale computer experiment obtained by evaluating at many input sites of the computer model. The experimental data are further used for constructing a statistical model called an emulator for the prediction and optimization of the complex system. However, most methods for

building emulators, such as Gaussian processes (GPs), suffer from computational problems as the number of data points becomes larger. To fix these problems, a new methodology for constructing emulators in large-scale computer experiments is proposed in this paper.

We propose using transformations for improving the additivity of the target function in large-scale experiments, that is, to find a transformation of the response so that the target function becomes approximately additive in its input variables. The concept of using transformations for improving additivity in approximating a deterministic function in computer experiments was proposed by [44] but not for large-scale experiments. In this paper, we further argue that the concept is beneficial for building emulators in large-scale computer experiments. This is because after improving the additivity, the target function is easier to approximate and is expected to be well-approximated using few data points. Thus, we can use a subset of big data to reduce the computational burden and build an emulator based on the subset to approximate the function well.

As the GP model is a popular method in constructing emulators, many subset-based techniques for Gaussian process, such as local GP ([45]), fixed-rank kriging ([46]), and sparse GP ([47, 48]), can be used for constructing emulators in large scale computer experiments. In these studies, the reason for using a subset only focuses on reducing the computational burden. For example, for the problem of fitting a GP to a sub-dataset with n data points instead of to a dataset with size N, the computational cost is reduced from $O(N^3)$ to $O(n^3)$. However, our method of using transformation for improving the additivity not only provides a statistical reason for applying the subset technique in modeling but also helps achieve a better approximation because an additive function can be approximated easily. A better approximation is important because one of the major goal of using emulators is to predict at input sites where no data points are evaluated.

A clear disadvantage of additive models is that they cannot entertain higher-order interactions among variables, leading to an unsatisfied prediction performance even with using big data. Such a disadvantage has been identified in the literature and is solved by extending

the additive modeling framework to include linear combinations of the variables ([10, 9]). Different from the previous literature, we employ a nonparametric smoother in the additive modeling framework to capture higher-order interactions. To further improve the prediction performance, a sequential procedure is proposed for identifying a data point whose estimated predicted performance is the worst and adding the data point into the sub-dataset for updating the model. This process can be performed in a sequential manner without expensive computational time.

The remainder of this article is organized as follows. Section 2 introduces the proposed models and the sequential updating technique. Section 3 discusses a method for constructing prediction confidence intervals of the proposed models. Section 4 provides several advantages of using the proposed method. Section 5 presents examples showcasing the capabilities of the method. Finally, Section 6 concludes the paper with a brief discussion.

2.2 Transformation and Additivity with Subset Techniques

Our aim is to use a sub-dataset $\mathbf{D}_{sub} \subset \mathbf{D}$ to construct an inverse *Transformed Additive Model* (TAM)

$$y(\mathbf{x}) = g^{-1}\{\mu + z_1(x_1) + \ldots + z_p(x_p)\}, \tag{2.1}$$

for approximating $f(\boldsymbol{x})$ well, where $\mathbf{x} = (x_1, \cdots, x_p)$. This implies that the sub-dataset $\mathbf{D}_{sub}$ must adequately represent and faithfully characterize the massive dataset $\mathbf{D}$, and such sub-dataset $\mathbf{D}_{sub}$ can be constructed by support points in [49]. Using $\mathbf{D}_{sub}$ to fit model (2.1), we need to estimate each component function $z_k(\cdot)$ and the transformation function $g(\cdot)$. If the transformation function $g(\cdot)$ is given, then the estimator of $z_k(\cdot)$ can be obtained through a penalized regression using $g(y(\mathbf{x}))$ as the responses and basis expansions on $z_k(\cdot)$ as inputs for $k = 1, \cdots, p$. This step can be conveniently implemented by using *mgcv* package [50]), so we focus on estimating $g(\cdot)$.

The goal of using transformations is to improve the additivity of $f(\mathbf{x})$. A convenient

way is to parametrize the unknown transformation function $g(\cdot)$ through the popular Box-Cox transformation (Box and Cox 1964) for nonnegative data (y ¿ 0):

$$g_\lambda(y) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, \text{if } \lambda \neq 0 \\ \log y, \text{if } \lambda = 0 \end{cases}, \tag{2.2}$$

whose transformation is controlled by an unknown parameter $\lambda \in R$. Then, we estimate λ by finding

$$\hat{\lambda} = \arg\min_{\lambda} \sum_{\{\mathbf{x}, y(\mathbf{x})\} \in \mathbf{D}_{sub}} \left[y(\mathbf{x}) - g_\lambda^{-1} \left\{ \hat{\mu}(\lambda) + \hat{z_1}(x_1; \lambda) + \ldots + \hat{z_p}(x_p; \lambda) \right\} \right]^2, \tag{2.3}$$

where the estimated mean $\hat{\mu}(\lambda)$ and component function $\hat{z_1}(\cdot; \lambda)$ in (2.3) are dependent on λ and obtained from $\mathbf{D}_{sub}$. The forgoing algorithm is summarized as Algorithm 4.

Algorithm 4 Transformed Additive Model (TAM)

1: **procedure** TAM($\mathbf{D}$) ▷
2: Obtain $\hat{\lambda}$ from (2.3). The $\hat{z}_k(x_k; \lambda)$ in (2.3) are obtained from the penalized least square method using $g_\lambda(y(\mathbf{x}))$ as the responses, where $(\mathbf{x}, y(\mathbf{x})) \in \mathbf{D}_{sub}$, and $\mathbf{D}_{sub}$ is the support point of $\mathbf{D}$.
3: **return** $\hat{\lambda}$, $\hat{\mu}$, $\hat{z}_k(x_k; \hat{\lambda})$, and $\mathbf{D}_{sub}$.
4: **end procedure**

Even with the best possible transformation, we cannot expect that the function becomes exactly additive in the transformed scale; thus, the prediction that we obtained using TAM can be unsatisfactory. To improve the prediction performance of TAM, we extend it to capture not only the main effects but also higher-order interaction effects in the transformed space. This step can be done by combing the additive function in (2.1) with a nonparametric function $z(x_1, \cdots, x_p)$, expressed as

$$y(\mathbf{x}) = g_\lambda^{-1} \left\{ \mu + (1 - \eta) \sum_{k=1}^{p} z_k(x_k) + \eta z(x_1, \cdots, x_p) \right\}, \tag{2.4}$$

where $\eta \in [0, 1]$. We call the extended model (2.4) as the transformed approximately ad-

ditive model (TAAM). To obtain an estimator of TAAM, because we expect that most of the variation is captured by the additive part in (2.4), a convenient way is to keep using $\hat{\lambda}$ $\hat{\mu}$, and $\hat{z}_k(x_k)$ obtained from Algorithm 1 as the estimators of λ, μ, and $z_k(x_k)$ in (2.4), where $k = 1, \cdots, p$. To estimate the function $z(x_1, \cdots, x_p)$ for capturing the interaction terms in the transformed scale, we suggest fitting a thin plate spline ([51]) to the data $g_\lambda(\mathbf{y}) \equiv (g_{\hat{\lambda}}(y_1), \cdots, g_{\hat{\lambda}}(y_n))'$, a general method for estimating a smooth function of multiple variables. Thus, the remaining unknown parameter in TAAM is only a one-dimensional parameter η. Its estimator can be obtained similar with obtaining $\hat{\lambda}$ in (2.3); that is,

$$\hat{\eta} = \underset{\eta \in [0,1]}{\arg\min} \sum_{\mathbf{x} \in \mathbf{D}_{sub}} \left(y(\mathbf{x}) - g_\lambda^{-1} \left\{ \hat{\mu} + (1-\eta) \sum_{k=1}^{p} \hat{z}_k(x_k) + \eta \hat{z}(x_1, \cdots, x_p) \right\} \right)^2 . \quad (2.5)$$

The prediction performance $|y(\mathbf{x}) - \hat{y}_{TAAM}(\mathbf{x})|^2$ for $(\mathbf{x}, y(\mathbf{x})) \in \mathbf{D} \backslash \mathbf{D}_{sub}$ helps identify the data point that can be used to further improve the TAAM (2.4), where $\hat{y}_{TAAM}(\mathbf{x})$ is the estimated prediction of function (2.4). Denote the data point whose value of the prediction performance $(y(\mathbf{x}) - \hat{y}_{TAAM}(\mathbf{x}))^2$ is the largest by $(\mathbf{x}^\star, y(\mathbf{x}^\star))$. We suggest adding the data point $(\mathbf{x}^\star, y(\mathbf{x}^\star))$ to D_{sub} :

$$\mathbf{D}^\star_{sub} = \mathbf{D}_{sub} \cup \{(\mathbf{x}^\star, y(\mathbf{x}^\star))\} \text{ with } (\mathbf{x}^\star, y(\mathbf{x}^\star)) = \underset{(\mathbf{x}, y(\mathbf{x})) \in \mathbf{D} \backslash \mathbf{D}_{sub}}{\arg\max} |g_{\hat{\lambda}}(y(\mathbf{x})) - \hat{g}_{\hat{\lambda}, \hat{\eta}}(y)|^2, \quad (2.6)$$

and use $\mathbf{D}^\star_{sub}$ to refit the TAAM. This step can be repeated until the average prediction performance $(y(\mathbf{x}) - \hat{y}_{TAAM}(\mathbf{x}))^2$ over $(\mathbf{x}, y(\mathbf{x})) \in \mathbf{D} \backslash \mathbf{D}^\star_{sub}$ is small enough. Further discussion about the stopping criterion is given in Section 4.4. We summarize the foregoing method for constructing TAAM as Algorithm 5.

2.3 Uncertainty Quantification

This section develops and discusses a method for quantifying prediction uncertainty from models TAM (2.1) and TAAM (2.4). The key idea is to connect the predictions from TAM

Algorithm 5 Transformed Approximately Additive Model (TAAM)

procedure TAAM($\mathbf{D}, \epsilon > 0$) ▷
    Set t = 1.
    Obtain $\hat{\lambda}^{(t)}$, $\hat{\mu}^{(t)}$, $\hat{z}_k^{(t)}(x_k)$, and $\mathbf{D}_{sub}^{(t)}$ from Algorithm 1.
    Obtain the estimator of η from (2.5), denoted by $\hat{\eta}^{(t)}$.
    Obtain $\hat{z}(x_1, \cdots, x_p)$ by fitting a thin plate spline to the data $g_{\hat{\lambda}}(\mathbf{y})$
    Update $\mathbf{D}^{(t)}$ by using (2.6) and set $t = t + 1$.
    Repeat steps 3 to 6 until the average prediction performance $(y(\mathbf{x}) - \hat{y}_{TAAM}(\mathbf{x}))^2$ over $(\mathbf{x}, y(\mathbf{x})) \in \mathbf{D} \backslash \mathbf{D}_{sub}^{((t))}$ is smaller than ϵ.
end procedure

and TAAM as the mean of a posterior distribution through a Bayesian interpretation. Thus, the posterior variance of the distribution can be used to construct a credible interval for quantifying the prediction uncertainty.

Suppose we want to quantify the prediction uncertainty at $\mathbf{x}_0$, and our goal is to construct a credible interval for $y(\mathbf{x}_0)$ from the posterior distribution $g(y(\mathbf{x}_0))|\mathbf{y}$, where $\mathbf{y}$ is a vector including responses from $\mathbf{D}_{sub}$. We use the TAM model in (2.1) to illustrate the method first. Recall that when fitting the TAM model, we apply a penalized least square method to $g(y(\mathbf{x}))$ with respect to $\mu + \sum_{i=1}^{p} z_i(x_i)$, and $z_k(\cdot)$ is represented using a basis expansion. Specifically, let $z_k(\cdot) = \sum_{j=1}^{b} h_{kj}(\cdot)\beta_{kj}$, where $\{\beta_{kj}\}_{j=1}^{b}$ are unknown coefficients, and $\{h_{kj}(\cdot)\}_{j=1}^{b}$ are basis functions defined using a sequence of b knots for $k = 1, \cdots, p$. This implies that the TAAM in (2.1) becomes

$$y(\mathbf{x}) = g_{\lambda}^{-1}\left\{\mu + \sum_{k=1}^{p}\sum_{j=1}^{b}\beta_{kj}h_{kj}(x_k)\right\}. \tag{2.7}$$

Additionally, when using the penalized least square, we need to specify the penalized parameter λ_k and smoothing matrix $\mathbf{S}_k$ for each component function for $k = 1, \cdots, p$. Then, the Bayesian interpretation given in the Appendix I gives the posterior distribution for

$\boldsymbol{\beta} = \left(\beta_{11}, \cdots, \beta_{1b_1}, \cdots, \beta_{p1}, \cdots, \beta_{pb_p}\right)$, which is

$$\boldsymbol{\beta}|\mathbf{y} \sim N\left(\left(\mathbf{H}^T\mathbf{H} + \sum_{k=1}^{p}\lambda_k\mathbf{S}_k\right)^{-1}\mathbf{H}^T\left\{g_\lambda(y(\mathbf{x})) - \mu\mathbf{1}\right\}, \hat{\sigma}^2\left(\mathbf{H}^T\mathbf{H} + \sum_{k=1}^{p}\lambda_k\mathbf{S}_k\right)^{-1}\right), \tag{2.8}$$

where $\hat{\sigma}^2$ is the residual sum of squares for the fitted model divided by $(n - tr(\sum_{k=1}^{p}\lambda_k\mathbf{S}_k))$, $tr(\cdot)$ is the trace function, $\mathbf{1}$ is a vector of 1's having length n, $\mathbf{H} = (\mathbf{H}_1 \ \cdots \ \mathbf{H}_p)$, and the ij-th element of $\mathbf{H}_k$ is $h_{kj}(x_{ki})$ for $k = 1, \cdots, p$. From (2.8), the posterior distribution $g_\lambda(y(\mathbf{x}_0))|\mathbf{y} = \mu + h^T(\mathbf{x}_0)\boldsymbol{\beta}|\mathbf{y}$ is

$$g_\lambda(y(\mathbf{x}_0))|\boldsymbol{y}, \mu, \eta, \lambda \sim N(\widehat{g_\lambda \circ y}(\boldsymbol{x}), V(\boldsymbol{x})), \tag{2.9}$$

where

$$\widehat{g_\lambda \circ y}(\boldsymbol{x}) = \mu + h^T(\mathbf{x}_0)\mathbf{S}^{-1}\mathbf{H}^T\left(\mathbf{I} + \mathbf{H}\mathbf{S}^{-1}\mathbf{H}^T\right)^{-1}\left\{g_\lambda(\mathbf{y}) - \mu\mathbf{1}\right\}, \tag{2.10}$$

$$V(\boldsymbol{x}) = h^T(\mathbf{x}_0)\mathbf{S}^{-1}h(\mathbf{x}_0) + h^T(\mathbf{x}_0)\mathbf{S}^{-1}\mathbf{H}(\mathbf{I} + \mathbf{H}\mathbf{S}^{-1}\mathbf{H}^T)^{-1}\mathbf{H}\mathbf{S}^{-1}h(\mathbf{x}_0), \tag{2.11}$$

$g_\lambda(\mathbf{y})$ is the transformed response vector, and $\mathbf{S} = \sum_{k=1}^{p}\lambda_k\mathbf{S}_k$. From (2.9), we can obtain the probability density function of $f(\boldsymbol{x})|\boldsymbol{y}$, and the median of the density is $\tilde{f}(\boldsymbol{x}) = g_\lambda^{-1}\left\{\widehat{g \circ f}(\boldsymbol{x})\right\}$ as [6] pointed out. This implies that a $(1-\alpha)100\%$ credible interval (CI) for the prediction is

$$\left[g_\lambda^{-1}\left\{\widehat{g_\lambda \circ f}(\boldsymbol{x}) - \phi_{\alpha/2}\sqrt{V(\boldsymbol{x})}\right\}, g_\lambda^{-1}\left\{\widehat{g_\lambda \circ f}(\boldsymbol{x}) + \phi_{\alpha/2}\sqrt{V(\boldsymbol{x})}\right\}\right], \tag{2.12}$$

where $\phi_{\alpha/2}$ is the critical value for the CI. Obtaining a CI from TAAM can be done straightforward by replacing the response $g_\lambda(\mathbf{y})$ in $\widehat{g_\lambda \circ f}(\boldsymbol{x})$ of (2.12) with $g_\lambda(\mathbf{y}) - \oplus_{k=1}^{p}\hat{\mathbf{z}}_k$, where $\oplus$ is the element-wise summation and $\mathbf{z}_k$ is a vector containing the function values of $\hat{z}_k(\cdot)$ from evaluating the TAM model in the inputs of the sub-dataset $\mathbf{D}_{sub}$. We summarize the method for quantifying prediction uncertainty as Algorithm 6.

Algorithm 6 UQ for TAM and TAAM

1: **procedure** UQTAM($\mathbf{D}, \boldsymbol{x}_0, \alpha$) ▷
2: Calculate (2.10) and (2.11).
3: Obtain the $(1-\alpha)100\%$ CI for the prediction from TAM by using (2.12).
4: Obtain the C.I. for the prediction from TAAM by replacing the response $g_\lambda(\mathbf{y})$ in $\widehat{g_\lambda \circ y}(\boldsymbol{x}_0)$ of (2.12) with $g_\lambda(\mathbf{y}) - \oplus_{k=1}^{p} \hat{\mathbf{z}}_k$.
5: **end procedure**

Note that when using Algorithm 3 with a one-parameter Box-Cox transformation (2.2), the lower bound needs to be set at 0 if $\lambda\{\widehat{g \circ f}(\boldsymbol{x}) - 1.96\sqrt{V(\boldsymbol{x})}\} + 1 < 0$ to guarantee that $g_\lambda(\cdot)$ is one-to-one. Moreover, although we demonstrate Algorithms 1 to 3 by using a one-parameter transformation, the algorithms can be applied to more general transformations, such as the Yeo-Johnson transformation (Yeo and Johnson 2000), relaxing the $y > 0$ constraint in the Box-Cox transformation.

2.4 Advantages

In this section, we discuss the advantages of using TAAM methods.

2.4.1 Prediction Performance, Computational Time, and Uncertainty Quantification

This subsection provides a numerical example to demonstrate the prediction performance, computational time, and the ability of uncertainty quantification using the proposed method. The example is a three-dimensional function considered in the classic beam bending problem ([52]). The detailed function form of the bending function is given in Table B.2. We also compare TAAM with the local Gaussian process (LaGP) and Multi-Resolution Functional ANOVA (MRFA), which are recently proposed methods for modeling large scale computer experiments proposed by [41] and [53]. The methods can be implemented through R packages LaGP ([45]) and MRFA ([54]). All the numerical results were obtained using R ([55]) on a 2.6 GHz laptop.

The experimental designs are generated from Sobol' sequences with scrambling independently ([34]). The sample sizes for training the models are 10^k for $k = 3, 4, 5, \cdots, 9.$,

and the testing dataset has a sample size of 10,000. Then, we fit models TAAM (Algorithm 2), LaGP, and MRFA to the training data and compare their prediction performance in terms of root-mean-squared-prediction-error (MSPE) and their computational time. The computational time of laGP and MRFA for the cases $n = 10^k$ and $k > 6$ is too high compared with the time of running TAAM, so we do not run them. The results are summarized in Table 2.1. From the table, we observe that the TAAM substantially outperforms the other two methods in terms of the computational time, emulator accuracy, and scalability.

	MSPE($\times 10^{-11}$)			Time (Sec)		
Sample Size (N)	TAAM	MRFA	LaGP	TAAM	MRFA	LaGP
1,000	1.4	280	3300	3.84	13.41	102.52
10,000	1.8	140	2300	10.38	71.31	106.47
100,000	1.3	14	4200	26.33	2385.67	141.93
1,000,000	1.1	X	X	54.04	X	X
10,000,000	1.0	X	X	233.92	X	X
100,000,000	1.7	X	X	643.36	X	X

Table 2.1: Prediction performance and computational time from TAAM, MRFA, and LaGP in the bending function examples.

We further compare the ability for quantifying prediction uncertainty using the methods TAAM and LaGP. (However, We do not compare using MRFA because its R package does not provide an option for obtaining prediction variance). To assess their abilities, we computed the prediction interval score ([33]), which is defined as $(u - \ell) + (2/\alpha)(\ell - x)I\{x < \ell\} + (2/\alpha)(x - u)I\{x > u\}$ with $\alpha = 95\%$. A smaller interval score indicates a better prediction interval. The interval scores are summarized in Table 2.2. From the table, we observe that TAAG is better than LaGP.

	Interval Scores ($\times 10^{-4}$)	
Sample Size (N)	TAAM	LaGP
1,000	**4.227**	89.042
10,000	**4.086**	9.025
100,000	**4.127**	4.316

Table 2.2: Interval scores (smaller is better) from TAAM and LaGP in the bending function examples.

2.4.2 Computational Complexity

We discuss the computational complexity of using TAM and TAAM methods and compare them with other methods in this subsection. The complexity of TAM is $O(b^3)$, where b is the number of knots in (2.7). The number b can be as large as the size of the sub-dataset used in TAM. In this case, if the size the sub-dataset is n, then the computational complexity of TAM is $O(n^3)$. This complexity is the same as that of many GP-based methods incorporated with a subset-based method used for fixing the computational problem of fitting a GP model, which needs $O(N^3)$ and is quite expensive when the sample size N is large ([56]). Thus, TAM can reduce much computational time compared with a standard GP. For obtaining the complexity of the TAAM method, because the sub-dataset used in TAAM is expanded from size n to $n+m$, the computational complexity is $O((n+m)^3)$. Although TAAM increases the computational complexity of TAM, its prediction accuracy is higher because it uses more data points to construct the model. We will provide some methods for further reducing the computational time of TAAM in the next subsection. We summarize the forgoing discussion in Figure 2.1.

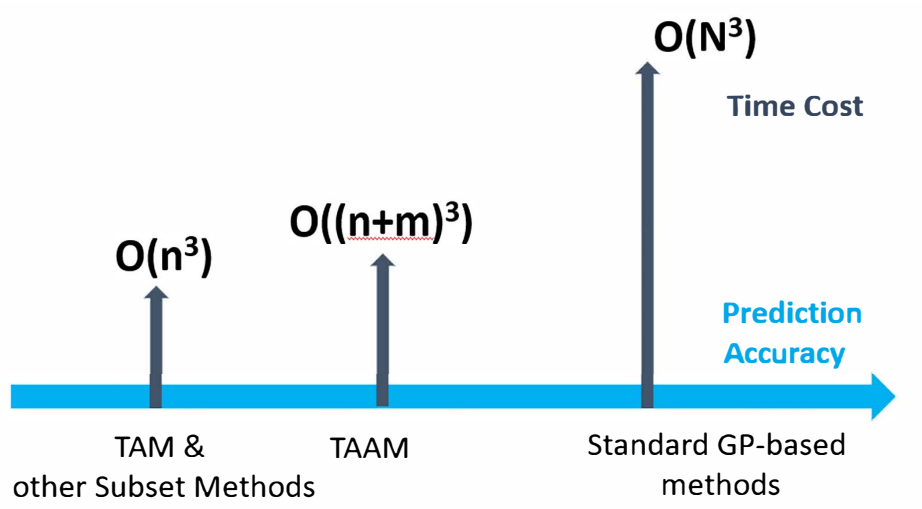

Figure 2.1: Comparison of Complexity of TAM and TAAM

2.4.3 Parallel Computing

In this subsection, we propose a method to increase the speed of running TAM and TAAM. When we examine the details of the forgoing algorithms, we found that the most time-consuming part is the sequential updating step in Algorithm 2. This step can be sped up by a method based on utilizing the space-filling property ([57]) of the data points in the sub-dataset, inherited from the support points. The data points can be used to partition the domain of the data into multiple sub-regions by assigning the lines that are equidistant to any two of the support points as the boundary lines of the sub-regions. Figure 2.2 demonstrates an example with 20 sub-regions partitioned by 20 data points.

Algorithm 7 Parallel Computing in Fitting TAAM

1: **procedure** TAAM($\mathbf{D}, \epsilon > 0$) ▷
2: Set t = 1.
3: Obtain $\hat{\lambda}^{(t)}$, $\hat{\mu}^{(t)}$, $\hat{z}_k^{(t)}(x_k)$, and $\mathbf{D}_{sub}^{(t)}$ from Algorithm 1.
4: Obtain the estimator of η from (2.5), denoted by $\hat{\eta}^{(t)}$.
5: Obtain $\hat{z}(x_1, \cdots, x_p)$ by fitting a thin plate spline to the data $g_{\hat{\lambda}}(\mathbf{y})$
6: Use $\mathbf{D}^{(1)}$ to split the input domain into multiple sub-regions, and, for each sub-region, pick one data point in $\mathbf{D} \backslash \mathbf{D}^{(t)}$ whose prediction error is the worse.
7: Update $\mathbf{D}^{(t)}$ by adding the picked data points in step 6 to $\mathbf{D}^{(t)}$ and set $t = t + 1$.
8: Repeat steps 3 to 6 until the average prediction performance $(y(\mathbf{x}) - \hat{y}_{TAAM}(\mathbf{x}))^2$ over $(\mathbf{x}, y(\mathbf{x})) \in \mathbf{D} \backslash \mathbf{D}_{sub}^{(t)}$ is smaller than ϵ.
9: **end procedure**

In each sub-region, we pick one data point $\in \mathbf{D} \backslash \mathbf{D}_{sub}$ whose prediction error is the worse. These data points can be added into the sub-dataset, and then the extended sub-dataset is used for updating the TAAM. Thus, instead of updating the TAAM one point at a time, we can update the TAAM parallelly through data points from multiple sub-regions of the domain. The foregoing method can be repeated until the prediction performance of TAAM is satisfied, is summarized in Algorithm 7. The parallel procedure can even be sped-up by running on a computer processor designed for distributed optimizations ([58]). For illustration, we implement the parallel computing method on fitting a TAAM to the bending function example in Section 4.1 with a sample size $N = 2000$, a sub-sample size

$n = 100$, and the size of the extra points $m = 500$. The extra points are selected sequentially (Algorithm 5) and parallelly (Algorithm 7). The computational time and MSPEs are summarized in Table 2.3. From the table, we observe that the parallel method greatly improves the computational time a lot and possess competitive predictability compared with the sequential method.

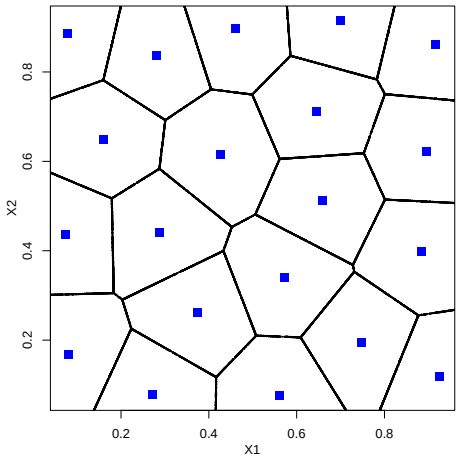

Figure 2.2: Domain partition of the data into different sub-region for applying parallel computing to the sequential updating step in Algorithm 2.

	m	Computational Time	MSPE
Sequential	500	135.42	4.862×10^{-12}
Parallel	500	2.13	5.572×10^{-11}

Table 2.3: Computational time and the MSPEs sequential computing (Algorithm 5) and parallel computing (Algorithm 7).

2.4.4 Stopping Criterion

Although many subset-based methods for fixing the modeling problem on a large scale dataset have been proposed, how to determine the subset size is still unclear in the literature. The subset method we used can be used for determining the (sub-)sample size of building TAM and the sequential step in TAAM. Because the sub-dataset is composed of data points from the original dataset $\mathbf{D}$, we propose viewing the data points in the sub-dataset as a training dataset and the data points not in the sub-dataset but in $\mathbf{D}$ as a testing dataset. Thus, the testing dataset can be further used to access the information of the prediction performance from TAM and TAAM, such as calculating the MSPEs over the testing dataset. The MSPE values can be used to further decide the sample sizes used in TAM and TAAM.

To demonstrate the idea, we record the prediction errors of the bending function exam-

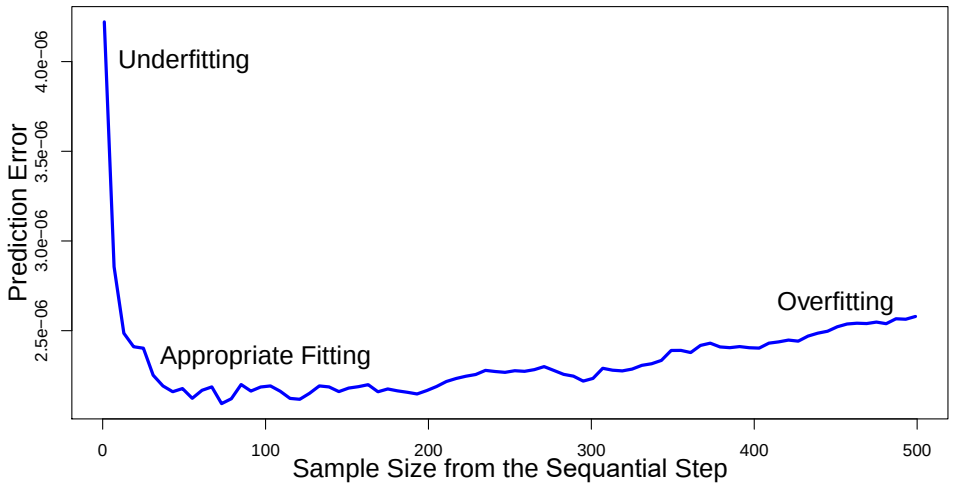

Figure 2.3: Accessing the prediction errors of various sample sizes when fitting a TAAM.

ple with sample size $10,000$ in Section 4.1 when the size in the sequential step of Algorithm 5 increases from 0 to 500. From Figure 2.3, we observe that implementing the sequential step in TAAM with a size from 0 to 25 may cause the overfitting problem and a size from 400 to 500 may have an overfitting problem. Thus, a size of 40 is a better choice in this example.

2.5 Numerical Comparisons

This section demonstrates the prediction performance of the proposed method with more numerical examples. These examples include several deterministic datasets from four common computer models and three real datasets as summarized in Table B.2. The software and computer are the same as we described in Section 4.1, and in the following section, we also compare the proposed with LaGP and MRFA methods.

2.5.1 More Examples from Computer Experiments

In this subsection, we present three more example functions for the TAAM method in comparison with MRFA and laGP: Ackley function, Schwefel function, and borehole function. Their function forms and input ranges are given in Table B.2, and further details can be

found in [42] The training data sizes are 1,000 and 10,000, and the testing data size is 10,000. All of these data are generated from the Sobol' sequences with scrambling independently ([59]).

The comparison results are shown in Tables 2.4 and 2.5. Compared with LaGP, although the computational time of TAAM may be larger in some cases, the prediction performance of TAAM is much better. Taking the Ackley function with N = 10,000 as an example, the computational time for TAAM is 124.35, higher than 13.6 from laGP with 20 neighborhood points, but the prediction error for TAAM is 0.341, around 10 times smaller than that for laGP with 20 neighborhood points. For MRFA, the computational time of TAAM is smaller, and the prediction performance of TAAM is competitive.

		Time (Sec)			
Functions	Sample Size (N)	TAAM	MRFA	LaGP (10d)	LaGP (25d)
Ackley (d = 2)	1,000	26.67	47.03	12.17	46.55
	10,000	124.35	140.8	13.6	56.55
Schwefel	1,000	32.77	44.27	11.33	54.72
	10,00	47.98	158.48	14.15	54.8
Borehole	1,000	34.38	118.88	113.43	499.67
	10,000	156.14	586.68	118.2	631.5
Power Plant	9568	158.54	9309.02	7.77	32.53
NASA	1503	68.4	$>>$ 158.54	1.48	6.05
CASP	40730	1003.63	2546.45	87.5	451.38

Table 2.4: Recorded computational time of the examples in Section 5.2

		MSPE			
Datasets	Sample Size (N)	TAAM	MRFA	LaGP (10d)	LaGP (25d)
Ackley	1,000	4.49×10^{-1}	6.58×10^{-1}	1.07×10^{1}	1.65×10^{0}
	10,000	3.41×10^{-1}	3.22×10^{-1}	3.02×10^{0}	2.81×10^{0}
Schwefel	1,000	8.39×10^{-8}	5.87×10^{-7}	1.52×10^{-1}	1.25×10^{-1}
	10,00	4.70×10^{-8}	2.95×10^{-7}	1.40×10^{-1}	1.17×10^{-1}
Borehole	1,000	5.33×10^{-2}	3.02×10^{-1}	1.67×10^{1}	3.47×10^{0}
	10,000	6.20×10^{-3}	7.30×10^{-3}	2.45×10^{-1}	4.20×10^{0}
CCPP	9568	4.438	20.836	6.095	5.1124
NASA	1503	4.523	6.636	6.514	6.342
PTS	40730	5.045799	9.517	6.586	5.977

Table 2.5: Recorded MSPE of the examples in Sections 4.2 and 4.3.

2.5.2 Applications on Noisy Data

Although the motivations of the proposed method are commonly derived for deterministic data in computer experiments and spatial statistics, the method can be extended to fit noise data by adding a nugget term ([60]) in the covariance function in (2.8). Data with noise are more common in real-world applications. In this subsection, we demonstrate the prediction performance of the proposed method using three large-scale noisy datasets. These noisy datasets are from the UCI Machine Learning Repository ([61]). The first example contains 9,568 data points collected from a combined-cycle Power Plant (CCPP) ([62]). The input variables consist of hourly average ambient variables, such as temperature, ambient Pressure, relative humidity, and exhaust vacuum, and these input variables are used to predict the net hourly electrical energy output of the plant. The second example is a NASA dataset, including 1,503 data points obtained from a series of aerodynamic and acoustic tests of two- and three-dimensional airfoil blade sections conducted in an anechoic wind tunnel ([63]). The input variables include frequency, angle of attack, chord length, free-stream velocity, and suction-side displacement thickness, which are used to predict scaled sound pressure. The third example is from materials science about measuring the physicochemical properties of protein tertiary structure (PTS). This dataset contains 40,730 data points and include nine input variables. More information is provided in Appendix II.

Because MRFA and LaGP also can be extended to fit noisy data, we also compare TAAM with the two methods. For the comparison of their predictability, as these datasets do not provide any testing dataset, we randomly select 500 data points from NASA, 2,000 points from the CCPP datasets, and 5,000 pints from the PTS datasets as the testing datasets. The prediction accuracy in terms of RMSPEs and execution time obtained from the three methods are given in the last two rows of Tables 2.4 and 2.5. The prediction accuracy of the TAAM is better than that of MRFA. The reason is that MRFA usually performs better when the important input variables are sparse when some group structures exist among the input variables ([64]). However, such structures do not exist in the three datasets. In addi-

tion, the computational time of TAAM is expected to outperform MRFA because TAAM is based on a subset technique, but MRFA is based on the whole dataset. Thus, considering the results presented in the two previous sections, the TAAM substantially outperforms the other methods in terms of computational time, emulator accuracy, and scalability in the deterministic data and noisy data.

2.6 Conclusions

This article shows that using transformations on the response of a p-dimensional target function can be highly beneficial in big data modeling. Two modeling techniques are proposed here. Their advantages include reducing the computational time for estimating the target function and providing satisfying prediction performance. Several numerical comparisons show that the methods outperform many recently proposed methods for big data in the fields of computer experiments and nonparametric regression. As many challenges in statistical modeling are arising in the era of big data, the proposed methods may shed light on modeling more generally statistical problems, such as classification problems and online learning problems. These problems will be challenging but interesting future research topics.

CHAPTER 3
VARYING COEFFICIENT FRAILTY MODELS WITH APPLICATIONS IN SINGLE MOLECULAR EXPERIMENTS

Motivated by an analysis of single molecular experiments in the study of T cell signaling, a new model called local linear varying coefficient frailty model is proposed in this chpater. Frailty models have been extensively studied but extensions to non-constant coefficients are limited to spline-based methods which tend to produce estimation bias near the boundary. To address this problem, we introduce a local polynomial kernel smoothing technique with a modified EM algorithm to estimate the unknown parameters. Theoretical properties of the estimators, including their unbiased property near the boundary, are derived along with discussions on the asymptotic bias-variance trade-off. The finite sample performance is examined by simulation studies, and comparisons with existing spline-based approaches are conducted to show the potential advantages of the proposed approach. The proposed method is implemented for the analysis of T cell signaling. The fitted varying coefficient model provides a rigorous quantification of an early and rapid impact on T cell signaling from the accumulation of bond lifetime, which can shed new light on the fundamental understanding of how T cells initiate immune responses.

3.1 Introduction

This paper is motivated by an analysis of single molecular experiments with the goal of understanding how the interactions between T cells and antigen-presenting cells initiate immune responses. T cell uses the T cell receptor (TCR) to recognize antigen in the form of peptide-major histocompatibility complex (pMHC) on antigen-presenting cells. Recognition is signified by a cascade of intracellular signaling events, including a transient rise of intracellular calcium (Ca^{2+}), and ultimately resulting in developmental decisions or ef-

fector functions [65]. Therefore, an important step to understand the recognition process is to study how the TCR-pMHC interactions trigger the rise of intracellular calcium.

By conducting a series of single molecular experiments called force-clamp assay [66], [67] showed that the T cell signaling is induced by the accumulation of TCR-pHMC bond lifetimes in repeated cell adhesion. This discovery provides a critical initial understanding of the immune system, but a sophisticated model that can quantify the underlying mechanism is absent. Development of such a model is not straightforward because of two features, which are associated with this study and commonly shared by many other applications. First, the experiments are performed by multiple replicates to account for the heterogeneity across cells. Therefore, a model that can borrow strength across different replicates and take into account the cell-to-cell variability is needed. Second, according to [67], it is observed that an early and rapid accumulation of bond lifetimes appears to be more likely to trigger T cell signaling. This indicates a time-varying effect from the bond lifetime accumulation and the effects near the left boundary which associated with the early accumulation are of main interest.

To the best of our knowledge, a modeling framework that takes into account the aforementioned features has not yet been systematically developed in the literature. A Cox model with random effects, often called a frailty model, is widely used in survival analysis with repeated measurements [68, 69, 70, 71, 72, 73, 74, 75]. We can predict the triggering probability and take into account the heterogeneity among cells using the frailty model, but most of the existing works on frailty models rely on the assumption of constant regression coefficients, which is not valid according to the second feature. Although some discussions on the extensions of varying coefficient models are available [76, 77, 78, 79], there are no theoretical justifications for estimation and inference, and they are mostly spline-based approaches which tend to produce estimation bias near the boundary [80]. On the other hand, there are extensive discussions on Cox models with varying coefficients [81, 82, 83], which can capture the dynamic impact of the bond lifetime accumulation over time but fail

to account for the cell-cell variability. Therefore, a new type of frailty model that can quantify the time-varying effects from covariates, especially with an accurate estimation on the boundary, and also account for the heterogeneity among experimental units is called for.

A local linear varying coefficient frailty model is proposed in this paper which allows the regression coefficients in the frailty model to change over time. The generalization to varying coefficients provides flexibility in model fitting but also posts some challenges in estimation and inference. A local linear smoothing technique [84] is introduced with a modified EM algorithm to estimate the unknown parameters, including the varying coefficients and the variance components. Asymptotic properties of the estimators, especially near the boundary, are developed and discussions on the bias-variance trade-off are provided. These results are particularly useful to support the estimated effect from the early accumulation of bond lifetime.

Beyond the current application, the proposed varying coefficient frailty models can be broadly applied in survival analysis and reliability analysis to model time-varying effects from covariates. For example, the proposed method can be implemented in [85] to estimate the time-varying impact from the smoking status to the risk of lung cancer. The proposed method can also be applied to the Framingham heart study [68] to explore dynamic impacts from risk factors on cardiovascular disease over time.

The remainder of the paper is organized as follows. The detailed experimental settings and the varying coefficient frailty models are introduced in Section 2. The modified EM approach and the local kernel smoothing algorithm are proposed in Section 3. Theoretical properties of the proposed varying coefficient estimators, including the asymptotic behavior of the estimators in the interior and near the boundary, are discussed in Section 4. Numerical comparisons and the finite sample performance of the proposed method are demonstrated by simulation studies in Section 5. The force-clamp assay is revisited and analyzed by the proposed model in Section 6. A summary and concluding remarks are given in Section 7. The additional regular conditions used in the proof of the theoretical

properties are given in Appendix A.

3.2 Statistical Analysis of T Cell Signaling

We first introduce the experimental settings of T cell signaling experiments in Section 2.1. Some preliminary analysis of the data is demonstrated. Then the varying coefficient frailty model is introduced in Section 2.2.

3.2.1 Experimental settings and the Data

The TCR−pMHC bond lifetimes are measured by a single molecular experiment called force−clamp assay [86, 66]. The force-clamp assay is illustrated by Figure 3.1 (A and B) and can be described as follows. On the left-hand side of Figure 3.1A, a biomembrane force probe (BFP) uses a micropipette-aspirated human red blood cell (RBC) with a probe bead attached to its apex as a force transducer. The probe bead was coated with pMHC to serve as a surrogate antigen-presenting cell (Figure 3.1B, left). A micropipette-aspirated T cell (Figure 3.1A, right) obtained from transgenic mice was driven to briefly (0.1s) contact the probe bead to prompt bond formation. Via T cell retraction, a tensile force on the TCR-pMHC bond was ramped (at 1000 pNs) to and clamped at a preset level until bond dissociation. A force-clamp cycle is demonstrated in Fig 3.1C and the bond lifetime is measured as the force-clamp period indicated in red in Fig 3.1C. To evaluate how the TCR-pMHC interaction relates to T cell signaling, a fluorescence optical path is added to BFP to simultaneously measure bond lifetime and calcium flux. For each T cell, the force-clamp cycle (i.e., Fig 3.1C) is repeated for 10 minutes; concurrently, intracellular Ca^{2+} is observed using fura-2 ratiometric imaging [67].

Calcium signals are generally classified into two types: type α and type β [67]. Examples for the two types of signals, with colored images of the molecules, are given in Figure 3.2. The calcium level is measured based on a ratiometric indicator. Red color indicates higher calcium levels while blue color indicates lower calcium levels. Type α signals are

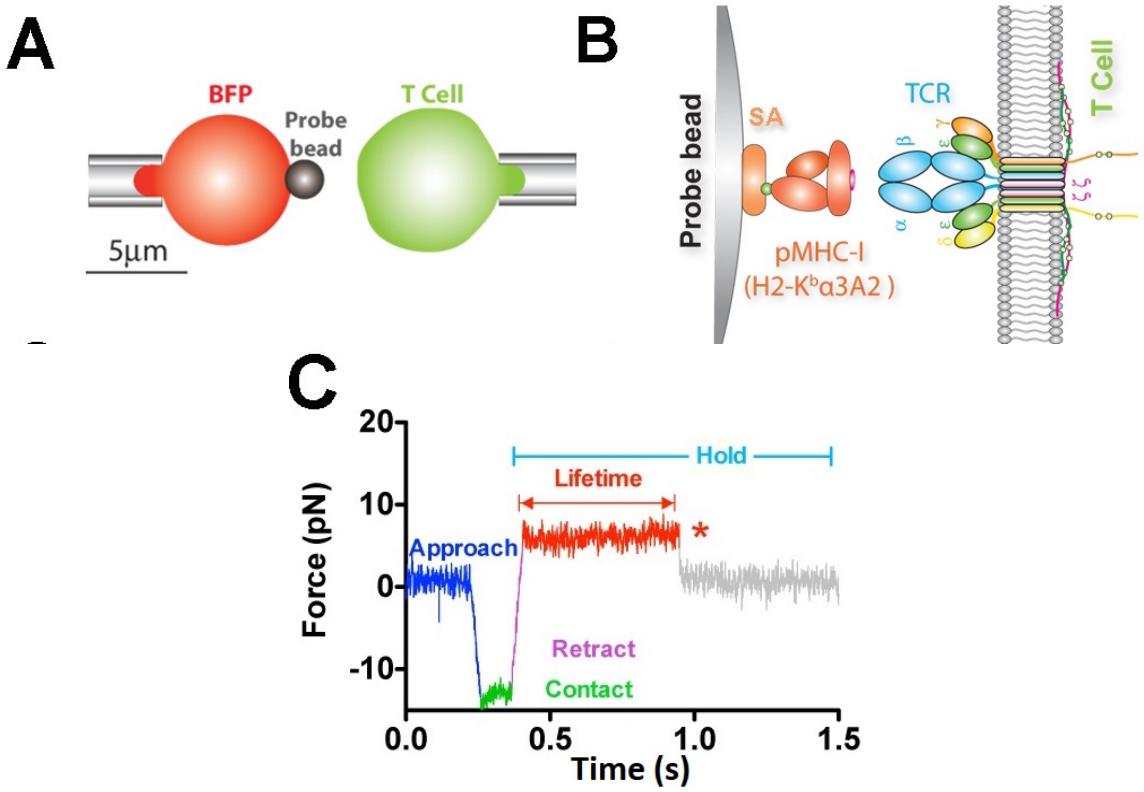

Figure 3.1: A and B: Illustration of a force-clamp assay. C: The bond lifetime is measured in one force-clamp cycle.

also called *triggered events* because there exists a rapid increase of the Ca^{2+} level, such as the increase from 70th second to 78th second shown in Figure 3.2A. Based on [67], the calcium signals are defined as type α if the Ca^{2+} curve contains a rapid increase of Ca^{2+} levels to more than 150% of the initial baseline. The time to a triggered event is defined as the time duration required for the calcium level to reach the 150% increase. In contrast, type β signals are called *non-triggered events* because the calcium levels are not triggered within the experimental period, which is known as right censored in statistical jargon, and therefore the colors remain almost unchanged over time as shown in Figure 3.2B.

For each T cell in repeated force-clamp assays, it produces a pair of curves over time as shown in the two examples in Figure 3.3. In each example, the calcium levels are plotted as red dots over time with their values referred to the axis on the left-hand side, while the corresponding cumulative bond lifetimes are plotted as black dashed line with their values

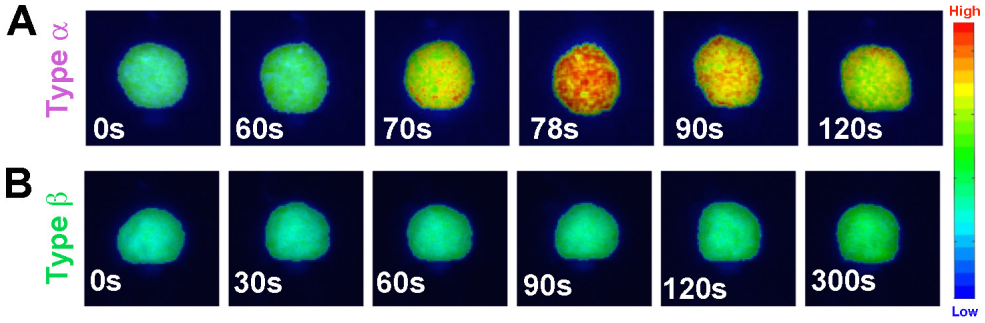

Figure 3.2: Illustration of the two types of calcium signals

referred to the axis on the right-hand side. According to [67], it appears that an early and rapid accumulation of bond lifetime is more likely to trigger the calcium level and lead to a type α signal. For example, compared with the non-triggered event in Figure 3.3B, the triggered event in Figure 3.3A shows a much faster accumulation of the bond lifetime, especially in the first fifty seconds.

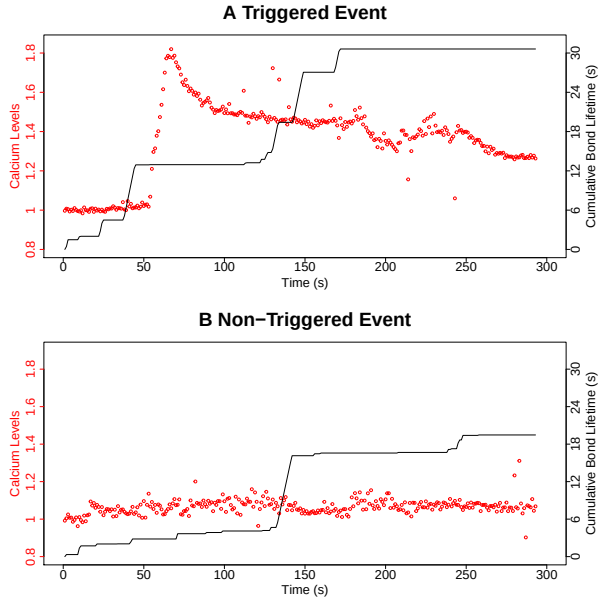

Figure 3.3: A and B show examples for the triggering and non-triggering event, respectively. The calcium signals (left axis) are plotted in red points and the cumulative bond lifetime (right axis) are plotted in solid lines.

Due to the inherently stochastic nature of single molecular interactions, repeated force-clamp cycles are conducted for multiple T cells as replicates to account for the heterogene-

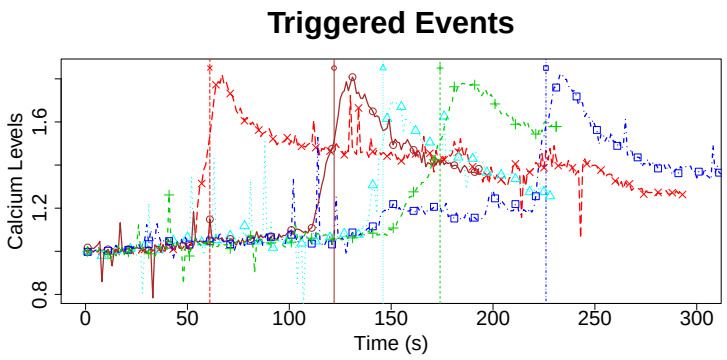

Figure 3.4: Five randomly selected examples of triggered events. The vertical lines are the triggering time points for each of the events.

ity in the cells. Figure 3.4 demonstrates five examples of the triggered events with their triggered points marked by vertical dashed lines and the time to triggered event is the time duration from zero to the corresponding dashed lines. These examples show that, although monoclonal TCR is used to reduce population heterogeneity, individual T cells still behave differently because of the cell-to-cell variability.

3.2.2 Varying coefficient frailty models

Understanding the mechanism of calcium triggering and the cause for different triggering time is crucial to the fundamental understanding of the immune system. Apart from cell-to-cell variability, [67] believe that a key driving force is the accumulation of bond lifetime in repeated adhesion contacts. Therefore, the goal here is to relate the time passes, before a triggering event, to the cumulative bond lifetime.

To address this problem, we first consider a frailty model, an extension of the Cox proportional hazard model to account for unobserved heterogeneity [69, 71, 72, 74, 75]. Define a hazard function by

$$\lambda(t; X_i(t)) = \lim_{\Delta t \to 0} P(t \leq T \leq t + \Delta t | T \geq t, X(t)),$$

where T is the *time to a triggered event* and $X(t)$ is the cumulative bond lifetime at time

point t. This hazard function describes the conditional probability of the calcium signal being triggered at time t, given no triggering before time t. Suppose the repeated force clamp assays are conducted for n replicates of T cells, denote T_i as the time to a triggered event for the i-th replicate, a frailty model can be written as

$$\lambda_i(t; X_i(t), a_i) = \lambda_0(t) \exp\{\beta X_i(t) + a_i\}, \tag{3.1}$$

where a_i is a random effect taking into account the heterogeneity among n replicates of T cells, $\lambda_i(t)$ is the hazard function for the i-th T cell replicate, and $\lambda_0(t)$ is an unspecified baseline function representing the triggering probability when the cumulative bond lifetime is 0 at time t. The frailty model in equation (3.1) can account for the cell-to-cell variability while modeling the effect from the cumulative bond lifetime on the triggering probability. However, the effect is assumed to be an unknown constant β and therefore cannot distinguish the difference between an early bond lifetime accumulation and a late one. Such an effect appears to be non-constant and capturing the varying effect on the left boundary is particularly important because it is observed that an early accumulation of bond lifetime seems to be more effective in triggering the calcium.

To address the aforementioned problems and rigorously quantify the time-varying effect from the bond lifetime accumulation, we propose a new model called local linear varying coefficient frailty (LLVCF) model which can be written as follows:

$$\lambda_i(t; \mathbf{X}_i(t), \mathbf{Z}_i, \mathbf{a}_i) = \lambda_0(t) \exp\{\mathbf{X}_i^T(t)\boldsymbol{\beta}(t) + \mathbf{Z}_i^T \mathbf{a}_i\}, \tag{3.2}$$

where $\lambda_i(t) = \lim_{\Delta t \to 0} P(t \leq T_i \leq t + \Delta t | T_i \geq t, \mathbf{X}_i(t), \mathbf{Z}_i, \mathbf{a}_i)$ is the hazard function of the i-th replicate, $\mathbf{a}_i$ is a vector of q random effects associated with covariates $\mathbf{Z}_i^T$, and $\mathbf{X}_i^T(t)$ is a p-dimensional covariates available at time t with the unknown time-varying coefficients $\boldsymbol{\beta}(t) = (\beta_1(t), \ldots, \beta_p(t))$. The random effects $\mathbf{a}_i$ are assumed to be independent multivariate normal $N(0, \boldsymbol{\Sigma}_a)$. In the current study, only one random effect, $a_i \sim N(0, \sigma_a^2)$,

is incorporated to account for the cell-to-cell variability and to facilitate borrowing strength across different replicates of T cells. Furthermore, there is one covariate involved in the study, thus $p = 1$ and model (3.2) can be simplified as

$$\lambda_i(t; X_i(t), a_i) = \lambda_0(t) \exp\{\beta_1(t) X_i(t) + a_i\}. \tag{3.3}$$

The main interest lies in estimating the time-varying coefficient $\beta_1(t)$ which quantifies the *dynamic* impact from the cumulative bond lifetime.

For the estimation of the varying coefficient $\boldsymbol{\beta}(t)$, we consider a kernel smoothing technique called local linear regression. Compared with the spline-based methods, local linear regression has the boundary bias correction property [87, 80]. This is desirable for the current analysis because the earlier impact (i.e., left boundary effect) from bond lifetime accumulation is of significant interest. Local linear regression is widely used for nonparametric estimation [84, 81, 82, 83, 88]. The idea is to approximate $\boldsymbol{\beta}(t)$ by a first-order Taylor expansion. Denote the approximation of the j-th varying coefficient at time point t_0 by

$$\beta_j(t) = \gamma_{1j}(t_0) + \gamma_{2j}(t_0)(t - t_0), \tag{3.4}$$

where $j = 1, \ldots, p$, and $t_0 \in (0, \tau)$, and τ the maximum of the experimental time. Therefore, the estimator of $\gamma_{1j}(t_0)$ is a local linear estimator for the varying coefficient function $\beta(\cdot)$ at time t_0. Similarly, the estimator of the local slope $\gamma_{2j}(t_0)$ is an estimator of $\beta_j'(\cdot)$ at time t_0.

Note that the proposed LLVCF model in (3.2) has two popular models as its special cases. If $\boldsymbol{\beta}(t) = \boldsymbol{\beta}$ and $\boldsymbol{\Sigma}_a = 0$ in (3.2), the LLVCF model is equivalent to the well-known Cox model [89]. On the other hand, without the random effect $\mathbf{a}_i$ in (3.2), the LLVCF model can be written as the varying coefficient Cox model [81, 82, 83].

3.3 Estimation

Although estimation procedures have been developed for the frailty model [68, 71, 73, 74], they are mainly designed for constant coefficients and therefore cannot be directly applied to estimate the varying coefficients in LLVCF. We propose a new estimation procedure based on a local polynomial kernel smoothing technique [90, 84] and a modified expectation-maximization (EM) algorithm [91, 92].

We start with some notation. For the i-th replicate, denote the censoring time of the experiment by C_i and define $Y_i = \min(T_i, C_i)$. An indicator variable, d_i, is defined by $d_i = 1$ if a triggered event is observed before the end of the experiment (i.e., $Y_i = T_i$) and $d_i = 0$ if it is right-censored (i.e., $Y_i = C_i$). Note that C_is are chosen to be larger than the potential event triggering time, and they are assumed to be independent of T_is [67]. For the ith replicate, the data are denoted by $(Y_i, \mathbf{X}_i(t), \mathbf{Z}_i, d_i)$, where $i = 1, \ldots, n$ and $t = 1, 2, \cdots, \tau$. For simplicity, derivations herein assume that no tie is observed, i.e., that no multiple triggering events occur at the same time. Extensions to address issues with ties are discussed in Section 7. Based on the standard derivations in survival analysis with right censored data, the log-likelihood function can be written as $\sum_{i=1}^{n} \{d_i \log \lambda_i(Y_i) + \log S_i(Y_i)\}$, where $S_i(Y_i) \equiv P(T_i \geq Y_i) = \exp(-\int_0^{Y_i} \lambda_i(t)dt)$. Up to a constant difference, the log-likelihood given $\mathbf{a}_i$, for $i = 1, \cdots$, and n, can be written as

$$\ell^{\star} \equiv \sum_{i=1}^{n} d_i \left[\log(\lambda_0(Y_i)) + \mathbf{X}_i^T(Y_i)\boldsymbol{\beta}(Y_i)\right] - \int_0^{Y_i} \lambda(s) \exp(\mathbf{X}_i^T(s)\boldsymbol{\beta}(s) + \mathbf{Z}_i^T \mathbf{a}_i)ds. \quad (3.5)$$

Denote the data by $\mathbf{D} = \{(Y_i, \mathbf{X}_i(t), \mathbf{Z}_i, d_i)\}_{i=1}^{n}$ and the unknown functions and parameters by $\boldsymbol{\theta} = (\lambda_0, \boldsymbol{\beta}, \boldsymbol{\Sigma}_a)$. The parameters $\boldsymbol{\theta}$ can be estimated by performing the following two steps iteratively.

E-step: Let $\hat{\boldsymbol{\theta}}^{(m)} = (\hat{\lambda}_0^{(m)}, \hat{\boldsymbol{\beta}}^{(m)}, \hat{\boldsymbol{\Sigma}}_a^{(m)})$ be the estimated functions and parameters in

the m-th iteration. The conditional expectation of (3.5) can be written as:

$$E[\ell^\star|\mathbf{D}, \hat{\boldsymbol{\theta}}^{(m)}] = Q_1(\boldsymbol{\beta}(t), \lambda_0(t)) + Q_2(\boldsymbol{\Sigma}_a),$$

where

$$\begin{aligned} Q_1(\boldsymbol{\beta}(t), \lambda_0(t)) &= \sum_{i=1}^{n} \Bigg\{ d_i \left(\log(\lambda_0(Y_i)) + \mathbf{X}_i^T(Y_i)\boldsymbol{\beta}(Y_i) \right) - \int_0^{Y_i} \lambda_0(s) \exp(\mathbf{X}_i^T(s)\boldsymbol{\beta}(s) + \\ &\qquad \log E[\exp(\mathbf{Z}_i^T \mathbf{a}_i)|\mathbf{D}, \hat{\boldsymbol{\theta}}^{(m)}]) ds \Bigg\} \end{aligned} \tag{3.6}$$

and

$$Q_2(\boldsymbol{\Sigma}_a) = -\frac{1}{2} \sum_{i=1}^{n} \left\{ \log |\boldsymbol{\Sigma}_a| + E\left[\mathbf{a}_i^T \boldsymbol{\Sigma}_a^{-1} \mathbf{a}_i | \mathbf{D}, \hat{\boldsymbol{\theta}}^{(m)} \right] \right\}. \tag{3.7}$$

Note that the E-step involves the conditional expectations for functions of the random effects, which is not observable. To compute these expectations, Gauss-Legendre quadrature approximation [70] and MCMC methods with a log-concave prior on the random effects [72] can be used. Detailed formulas and algorithms are given in the supplemental material. The estimation procedure can be easily extended to other distributional assumptions for the random effects.

M-step: The estimation of $\hat{\boldsymbol{\theta}}^{(m+1)}$ in the $(m+1)$-th iteration consists of three elements, $\hat{\lambda}_0^{(m+1)}(\cdot)$, $\hat{\boldsymbol{\beta}}^{(m+1)}(\cdot)$, and $\hat{\boldsymbol{\Sigma}}_a^{(m+1)}$. First, the baseline hazard $\lambda_0(t)$ is estimated by the non-parametric maximum likelihood estimator (NPMLE) [93]. Assuming that $R(t) = \{j : Y_j \geq t\}$, the NPMLE for $\lambda_0(t)$ is

$$\hat{\lambda}_0^{(m+1)}(t) = \left[\sum_{j \in R(t)} \exp(\mathbf{X}_j^T(t)\boldsymbol{\beta}(t) + \log E[\exp(\mathbf{Z}_i^T \mathbf{a}_i)|\mathbf{D}, \hat{\boldsymbol{\theta}}^{(m)}]) \right]^{-1} \tag{3.8}$$

if $t = Y_i$ for all i such that $d_i = 1$; otherwise, $\hat{\lambda}_0^{(m+1)}(t) = 0$.

Define $\boldsymbol{\gamma}(t) = (\boldsymbol{\gamma}_1(t), \boldsymbol{\gamma}_2(t))$, where $\boldsymbol{\gamma}_1(t) = (\gamma_{11}(t), \cdots, \gamma_{1p}(t))$ and $\boldsymbol{\gamma}_2(t) = (\gamma_{21}(t), \cdots, \gamma_{2p}(t))$.

Plugging in the first-order approximation (3.4) of $\boldsymbol{\beta}(t)$ at time t to the profile likelihood

$$\sum_{i=1}^{n} d_i \left\{ \mathbf{X}_i^T(Y_i)\boldsymbol{\beta}(Y_i) - \log \left[\sum_{j \in R(Y_i)} \exp(\mathbf{X}_j^T(Y_i)\boldsymbol{\beta}(Y_i) + \log E[\exp(\mathbf{Z}_i^T \mathbf{a}_i) | \mathbf{D}, \hat{\boldsymbol{\theta}}] \right] \right\},$$

which is obtained by substituting $\hat{\lambda}_0^{(m+1)}(t)$ into (3.6), we can estimate $\boldsymbol{\gamma}(t)$ by a kernel smoothing method. That is, to maximize the local partial likelihood constructed by kernel smoothing techniques as follows [94, 84]:

$$\begin{aligned} \hat{\boldsymbol{\gamma}}^{(m+1)}(t) &= \arg\max_{\boldsymbol{\gamma}(t)} \sum_{i=1}^{n} d_i K_h(Y_i - t) \left\{ \tilde{\mathbf{X}}_i^T(Y_i, Y_i - t)\boldsymbol{\gamma}(t) - \right. \\ &\quad \left. \log \left[\sum_{j \in R(Y_i)} \exp(\tilde{\mathbf{X}}_j^T(Y_i, Y_i - t)\boldsymbol{\gamma}(t) + \Delta_i) \right] \right\}, \end{aligned} \tag{3.9}$$

where $\tilde{\mathbf{X}}_j(Y_i, Y_i - t) = (\mathbf{X}_j^T(Y_i), \mathbf{X}_j^T(Y_i) \cdot (Y_i - t))^T$, $\Delta_j = \log E[\exp(\mathbf{Z}_j^T \mathbf{a}_j) | \mathbf{D}, \hat{\lambda}^{(m+1)}, \hat{\boldsymbol{\gamma}}^{(m)}, \boldsymbol{\Sigma}_a^{(m)}]$, $K_h(s) = (1/h)K(s/h)$, $K(\cdot)$ is a kernel function, and h is the bandwidth representing the size of the local neighborhood. In this paper, we use the Epanechnikov kernel, $K(s) = (3/4)(1 - s^2)$ for $s \in [-1, 1]$, which is a common choice in kernel methods because of its optimal properties given in [90] and [84]. From a practical point of view, many of the common kernels, such as Epanechnikov and Gaussian, tend to produce similar estimators, so the choice of kernel is not usually critical in practice [95]. Note that the local linear smoothing technique (3.4) and (3.9) for estimating $\boldsymbol{\beta}(t)$ can be easily applied for other basis function approaches, such as penalized splines [96].

The last element in the M-step is to estimate the variance components $\boldsymbol{\Sigma}_a$ by maximizing (3.7), and the estimator can be written as

$$\hat{\boldsymbol{\Sigma}}_a^{(m+1)} = \frac{1}{n} \sum_{i=1}^{n} E[\mathbf{a}_i \mathbf{a}_i^T | \mathbf{D}, \hat{\lambda}^{(m+1)}, \hat{\boldsymbol{\gamma}}^{(m+1)}, \hat{\boldsymbol{\Sigma}}_a^{(m)}].$$

The iterative procedure is terminated if the log-likelihood increment is smaller than a predetermined value. Denote the estimates by $\hat{\lambda}_0(t)$, $\hat{\boldsymbol{\gamma}}(t)$ and $\hat{\boldsymbol{\Sigma}}_a$. Due to the parametric formulation of NPMLE and local linear approximation, the convergence of the proposed EM procedure is guaranteed according to existing results in the EM literature [91, 92]. In practice, the initial settings of the EM algorithm can be $\mathbf{a} = 0$ and $\boldsymbol{\beta}(t)$ estimated from a Cox varying coefficient model, such as [81] and [82]. The procedure is summarized by the following algorithm.

Algorithm 8 An extedned EM algorithm for the varying coefficient frailty model

1: **procedure**

2: Given data $\{(Y_i, \mathbf{X}_i(t_j), \mathbf{Z}_i, d_i) : i = 1, \cdots, n \text{ and } j = 1, \cdots, N\}$ and initial estimates $\hat{\boldsymbol{\beta}}^{(m)}(t)$ for $\boldsymbol{\beta}(t)$ and $\hat{\boldsymbol{\Sigma}}_a^{(m)}$ for $\boldsymbol{\Sigma}_a$, with $m = 0$.

3: [E-step]: Derive $Q_1(\boldsymbol{\beta}(t), \lambda_0(t))$ from (3.6) and $Q_2(\boldsymbol{\Sigma}_a)$ from (3.7) based on data $\mathbf{D}$ and

current estimators.

4: [M-step]: $\hat{\boldsymbol{\beta}}^{(m+1)}(t) \leftarrow \hat{\boldsymbol{\gamma}}_1(t)$, where $\hat{\boldsymbol{\gamma}}_1(t)$ is from (3.9) and $\hat{\boldsymbol{\Sigma}}_a^{(m+1)} \leftarrow \arg\max_{\boldsymbol{\Sigma}} Q_2(\boldsymbol{\Sigma})$.

Then, set m as $m + 1$.

5: Repeat the E-step and M-step until convergence.

6: **return** $\hat{\boldsymbol{\beta}}^{(m)}$ and $\hat{\boldsymbol{\Sigma}}_a^{(m)}$.

7: **end procedure**

3.4 Asymptotic Theorem

Theoretical properties of the proposed varying coefficient estimators are developed in this section and the bias-and-variance trade-off in bandwidth selection is discussed. Based on these results, the bias correction property near the boundary is derived from [87, 84]. The additional regularity conditions are given in Appendix A, and detailed proofs of the results

are provided in the supplemental material.

We first derive theoretical properties, including asymptotic normality and an explicit expression for the asymptotic bias and variance, for the local linear estimator $\hat{\boldsymbol{\gamma}}_1(t)$. The result is summarized in the following theorem.

Theorem 3.4.1. *Under the Conditions A1 - A6 in Appendix A, when $t \in [h, \tau - h]$, we have:*

$$\sqrt{nh}\left\{(\hat{\boldsymbol{\gamma}}_1(t) - \boldsymbol{\gamma}_{01}(t) - \frac{h^2}{2}\boldsymbol{\gamma}''_{01}(t)\int_{-1}^{1} s^2K^2(s)ds\right\} \xrightarrow{D} N_p(0, \mathbf{I}^{-1}(t)\int_{-1}^{1} K^2(s)ds),$$

as $n \to \infty$, where $\mathbf{I}(t)$ is the Fisher information matrix of $\hat{\boldsymbol{\gamma}}_1(t)$ which can be written as

$$\mathbf{I}(t) = E\left\{\lambda_1(t)P(Y_1 \geq t)X(t)X(t)^T\right\} - \frac{E\left\{\lambda_1(t)P(Y_1 \geq t)\mathbf{X}(t)\right\}E\left\{\lambda_1(t)P(Y_1 \geq t)\mathbf{X}(t)\right\}^T}{E\left\{\lambda_1(t)P(Y_1 \geq t)\right\}}. \tag{3.10}$$

Based on Theorem 3.4.1, it appears that the leading bias term for $\hat{\boldsymbol{\gamma}}_1(t)$ is in the order of h^2, while the asymptotic variance is in the order of $(nh)^{-1}$ because $\mathbf{V}(t) = (nh)^{-1}\mathbf{I}^{-1}(t)\int_{-1}^{1} K^2(s)ds$. These two orders indicate a bias-and-variance trade-off in the selection of h. For example, a larger bandwidth leads to an estimator with smaller variance but larger bias. Determining the optimal bandwidth is important in practice. Some discussions regarding bandwidth selection are given in Section 6.

Confidence intervals can be constructed based on the results of Theorem 3.4.1 as follows. First, a consistent estimator of the variance of $\hat{\boldsymbol{\gamma}}_1(t)$, denoted as $\hat{\mathbf{V}}(t)$, can be obtained by the upper left $p \times p$ matrix of $(nh)^{-1}[-Q''_n(\hat{\boldsymbol{\gamma}}(t))/n]^{-1}\int_{-1}^{1} K^2(s)ds$, where $Q''_n(\hat{\boldsymbol{\gamma}}(t))$ is the second derivative of (3.9). Then, we can construct the $100(1-\alpha)\%$ confidence interval for the j-th component of $\boldsymbol{\gamma}_1(t)$ by $\{\hat{\boldsymbol{\gamma}}_{1j}(t) \pm z_{\alpha/2}\hat{\mathbf{V}}_{jj}(t)\}$, where $\hat{\mathbf{V}}_{jj}(t)$ is the j-th diagonal element of $\hat{\mathbf{V}}(t)$ and $z_{\alpha/2}$ is the $100(1-\alpha/2)$ percentile of the standard Normal distribution.

The asymptotic results in Theorem 1 are developed for the interior points $t \in [h, \tau - h]$.

In the next theorem, asymptotic estimation performance near the boundary is discussed. Without loss of generality, we consider the left boundary $t = ch, 0 < c \leq 1$ and a similar result can be obtained for the right boundary point $t = \tau - ch$. When t goes to 0 from the right side of 0, it is denoted by $t \to 0+$.

Theorem 3.4.2. *2 Assume that* $\lim_{t\to 0+} \lambda_0(t) > 0$, $\lim_{t\to 0+} P(Y \geq t|\mathbf{X}(u)) > 0$, $\mathbf{Z}(u) > 0$, *and* $\boldsymbol{\beta}''(t)$ *is right continuous at time 0. Then, under conditions A1 - A6, we have*

$$\sqrt{nh}\left\{\hat{\boldsymbol{\gamma}}_1(ch) - \boldsymbol{\gamma}_{01}(ch) - \frac{h^2}{2}\boldsymbol{\gamma}''(0+)\int_{-c}^{1} s^2 K(s)ds\right\} \to N(0, I^{-1}(0+)\int_{-c}^{1} K^2(s)ds).$$

The theorem shows that the asymptotic estimation bias and variance are in the order of h^2 and $(nh)^{-1}$ near the boundary, which are the same as the estimator in the interior point. This implies that the proposed local linear varying coefficient estimators enjoy the bias correction property near boundaries [87, 84].

For the variance components $\hat{\boldsymbol{\Sigma}}_a$ of the random effects in model (3.2), we have the following asymptotic properties.

Theorem 3.4.3. *3 Given the conditions A1 - A7 in the Appendix A, if* $nh^4 \to 0$ *as* $n \to \infty$, *we have*

(a) $\hat{\boldsymbol{\Sigma}}_a \xrightarrow{P} \boldsymbol{\Sigma}_a$.

(b) $\sqrt{n}(vec(\hat{\boldsymbol{\Sigma}}_a) - vec(\boldsymbol{\Sigma}_a)) \xrightarrow{D} N(0, U(\boldsymbol{\Sigma}_a))$, *where* $vec(\boldsymbol{\Sigma}_a)$ *converts* $\boldsymbol{\Sigma}_a$ *into a column vector and* $U(\boldsymbol{\Sigma}_a) = E(\mathbf{a} \otimes \mathbf{a}^T \otimes \mathbf{a} \otimes \mathbf{a}^T) - vec(\boldsymbol{\Sigma}_a)vec(\boldsymbol{\Sigma}_a)^T$.

Theorem 3 implies that if the order of bandwidth h can be written as $n^{-\alpha}$ with $1/4 < \alpha < 1$, then the $\sqrt{n}$-rate of convergence and asymptotic normality of $\hat{\boldsymbol{\Sigma}}_a$ hold.

3.5 Simulation Study

3.5.1 Comparison with spline-based varying coefficients frailty models

The existing varying coefficient frailty models are mainly developed by using spline-based approaches [76, 77, 78, 79]. Therefore we need to compare the proposed method with the spline-based methods in terms of their estimation accuracy, especially the performance near the boundary. The data is generated according to the hazard function in (3.3) with the varying coefficient specified by $\beta(t) = (t-1)^2/4$ and $X(t) = |W|(1 + (t+1)^2/2)$, where $W \sim N(3, 1)$. The baseline function is specified by $\lambda_0(t) = 0.5$ and the variance component for the random effect a is $\sigma_a^2 = 0.5$. The censoring times C are randomly generated from Uniform$(1.5, 2)$ leading to a 20% censoring rate. Two sample sizes $n = 100$ and 200 are considered. For each sample size, numerical comparisons are evaluated based on 200 replications.

The spline-based estimators are obtained by modifying the M-step using the cubic B-spline basis and the nature cubic spline basis ([79]). The result of cubic B-spline is demonstrated by the red dashed line and the nature cubic B-spline is demonstrated by the brown long-dashed line in Figure 3.5 with 9 knots selected at (0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2). The true function is indicated by the dotted line. Further comparisons with the spline-based methods with 5, 7, and 11 knots are summarized in the supplemental material. Based on these results, it appears that the local linear estimator, denoted by the blue curve, outperforms the spline-based estimators especially near the boundary.

3.5.2 Finite sample performance

Based on the same simulation setting in Section 5.1, we further examine the finite sample performance of the proposed estimators. Figure 3.6 demonstrates the average performance of $\hat{\gamma}_{11}(t)$ and their 95% confidence intervals with bandwidth $= 0.25, 0.5$, and 0.75. The average performance of $\hat{\gamma}_{11}(t)$ is denoted by the red dotted lines with crosses for $n =$

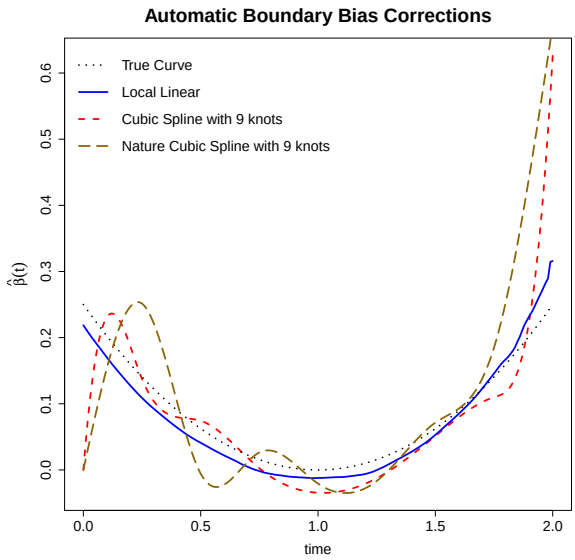

Figure 3.5: Comparison of the varying coefficient estimator from the local linear method (blue solid line) with bandwidth 0.5 and the spline-based methods using the cubic spline (red dashed line) and the nature cubic splines (brown long-dashed line) with knots at (0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2). The true curve is the black dotted line.

100 and denoted by blue dashed lines with circles for $n = 200$. The 95% confidence interval of $\hat{\gamma}_{11}(t)$ is marked by red dashed lines for $n = 100$ and blue dotted lines for $n = 200$. The black lines are the true $\beta(t)$. For a fixed bandwidth, it is clear that the estimation bias and uncertainty are reduced with the increase of sample size. It also appears in both sample sizes that, with the increase of bandwidth, the confidence interval becomes narrower while the estimation bias increases. This observation is consistent with the bias-and-variance trade-off in bandwidth selection shown in Theorems 1 and 2. The estimation uncertainty is larger in the boundary, especially for small bandwidth, which is common in kernel smoothing methods.

The estimated variance of the random effect, denoted by $\hat{\sigma}_a^2$, are reported in Table 3.1. In general, the variance components tend to be underestimated, especially for smaller sample size. This is commonly observed in conventional random effect estimation due to the underestimation of the uncertainty in estimating $(\hat{\boldsymbol{\gamma}}_{11}(t), \hat{\lambda}_0(t), \hat{\sigma}_a^2)$ [72]. For both sample sizes, the smallest bandwidth $h = 0.25$ appears to have the best estimation of

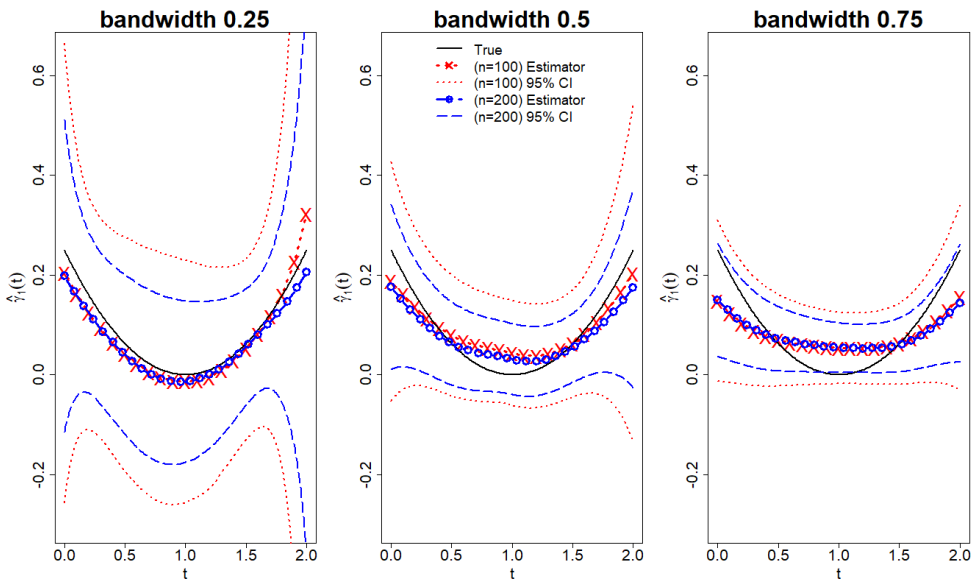

Figure 3.6: Estimated varying coefficients in Example 1 with $h = 0.25$, $h = 0.5$, and $h = 0.75$. The black lines are the true $\beta(t)$. The average performance of $\hat{\gamma}_{11}(t)$, which is equivalent to $\hat{\beta}(\cdot)$ evaluated at t, is denoted by the red dotted lines with crosses for $n = 100$ and blue dashed lines with circles for $n = 200$. The 95% confidence interval of $\hat{\gamma}_{11}(t)$ is marked by red dotted lines for $n = 100$ and blue dashed lines for $n = 200$.

σ_a^2. This observation is consistent with the theoretical results in Theorem 3(a) because, given a finite sample, a smaller bandwidth leads to a faster convergence of $nh^4 \to 0$ and therefore a better estimation consistency of σ^2. The estimation of baseline hazard is evaluated by the root mean square error (RMSE) of the estimated cumulative hazard functions, $\{\sum_{\{i:d_i=1\}}(\hat{\Lambda}_0(t_i) - \Lambda_0(t_i))^2\}^{1/2}$, where $\hat{\Lambda}_0(t_i) = \sum_{t \le t_i} \hat{\lambda}_0(t)$ from (3.8) with $\beta(t) = \hat{\gamma}_{11}(t)$, and $\Lambda_0(t_i) = \int_0^t \lambda_0(t)dt = 0.5t$ in this example. Their average performance with standard deviations are summarized in Table 3.1. The results indicate that the estimation accuracy for baseline hazard is improved as the sample size increases.

3.5.3 An example with two varying coefficients

In this example, we focus on a more challenging setting with two varying coefficients and a non-constant baseline hazard function. The data are generated from equation (3.2) with $p = 2$, $q = 1$, the random effect following $a \sim N(0, 0.25)$, the varying coefficient specified

Table 3.1: The average performance of the estimated variance components and the root mean squared error of the estimated cumulative hazard functions in Example 1. Their standard deviations are given in the parenthesis .

		$h = 0.25$	$h = 0.5$	$h = 0.75$
n=100	$\hat{\sigma}_a^2$	0.469 (0.164)	0.450 (0.156)	0.429 (0.132)
	RMSE $\hat{\Lambda}_0$	0.183 (0.201)	0.151 (0.146)	0.140 (0.145)
n=200	$\hat{\sigma}_a^2$	0.480 (0.106)	0.469 (0.098)	0.447 (0.075)
	RMSE $\hat{\Lambda}_0$	0.143 (0.110)	0.119 (0.067)	0.123 (0.063)

as

$$\boldsymbol{\beta}(t) = (\exp(-(t-0.5)^2)/2, 1/2 + (t-1)^2/2)^T,$$

and $\mathbf{X}(t) = (U_1(3/2 + (t+1)^2/2), (U_2/4)I_{\{t\leq 1\}}(t) + (U_3/2)I_{\{t>1\}}(t))^T$, where U_1, U_2, U_3 are independent variables following Uniform(0,1). The baseline hazard function is assumed to be $\lambda_0(t) = (1/2)t^2$, which is a function of time. The censoring times C are randomly generated from Uniform(1.5,2) leading to a 16.8% censoring rate. Similar to the previous example, we also consider two sample sizes $n = 250$ and $n = 500$. For each sample size, numerical performance is evaluated based on 100 replications.

Based on the proposed estimation procedure, the varying coefficients are estimated and the average performance of $\hat{\gamma}_{11}(t)$ and $\hat{\gamma}_{12}(t)$ are plotted in red dotted lines with crosses for $n = 250$ and blue dashed lines with circles for $n = 500$ in Figure 3.7. Their 95% confidence intervals are given in dotted lines for n=250 and dashed lines for n = 500. In Figure 3.7, a similar bias-and-variance trade-off in bandwidth selection is observed for both sample sizes. The estimated variance components are reported in Table 3.2 and the RMSEs of the estimated cumulative hazard functions are summarized, where the true cumulative hazard function is $(1/6t^3)$. The smallest bandwidth again provides the best estimation of σ_a^2 and a similar improvement on the baseline hazard estimation is observed by the increase of sample size.

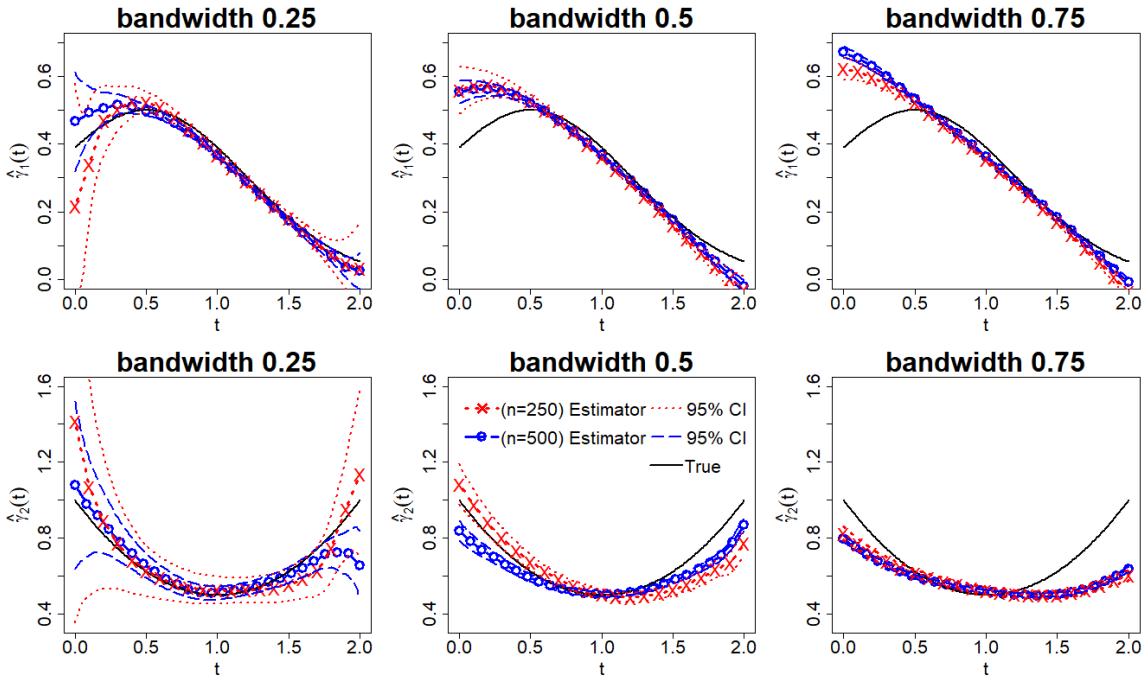

Figure 3.7: Estimated varying coefficients in Example 2 with $h = 0.25$, $h = 0.5$, and $h = 0.75$. The black lines are the true varying coefficient functions. The average performance of $\hat{\gamma}_{11}(t)$ and $\hat{\gamma}_{12}(t)$, which are equivalent to $\hat{\beta}_1(t)$ and $\hat{\beta}_2(t)$, are denoted by the red dotted lines with crosses for $n = 250$ and blue dashed lines with circles for $n = 500$. The 95% confidence interval of $\hat{\gamma}_{11}(t)$ is marked by red dotted lines for $n = 250$ and blue dashed lines for $n = 500$.

3.6 Revisiting the T Cell Signaling Experiment

We return to analyze the T cell signaling experiment in Section 2 by using the varying-coefficient frailty model. In this experiment, the cumulative bond lifetime is assumed to be $X(t)$. The experiments are performed based on 51 replicates, and there are 22 triggering events observed. Each replicates are observed over 600 seconds, i.e., $t = 0, 1, \cdots, 599$. Therefore, the fitted model can be described by equation (3.3) with $p = 1$, $n = 51$, and $\tau = 599$.

Before fitting model (3.3), an important practical issue is the selection of an optimal bandwidth. We implement a selection procedure based on cross-validation, which is widely used in nonparametric regression. First, randomly split the data into M subsets, denoted by $D_1, \cdots, D_M$. Leaving out D_m as the testing data, we can estimate parameters based on

Table 3.2: The average performance of the estimated variance components and the root mean squared error of the estimated cumulative hazard functions in Example 2. Their standard deviations are given in the parenthesis.

		$h = 0.25$	$h = 0.5$	$h = 0.75$
n=250	$\hat{\sigma}_a^2$	0.234 (0.107)	0.223 (0.102)	0.218 (0.098)
	RMSE $\hat{\Lambda}_0$	0.124 (0.113)	0.112 (0.96)	0.105 (0.88)
n=500	$\hat{\sigma}_a^2$	0.245 (0.093)	0.231 (0.085)	0.222 (0.072)
	RMSE $\hat{\Lambda}_0$	0.097 (0.85)	0.073 (0.061)	0.065 (0.057)

$M-1$ subsets and calculate the prediction error (PE) by the negative partial log-likelihood [82] denoted by

$$PE_m(h) = -\sum_{i \in D_m} d_i \left\{ \mathbf{X}_i^T \hat{\boldsymbol{\gamma}}_1(Y_i) + E[\mathbf{Z}_j^T \mathbf{a}_j] - \log \left[\sum_{j \in R(Y_i)} \exp(\mathbf{X}_j^T \hat{\boldsymbol{\gamma}}_1(Y_i) + \log E[\exp(\mathbf{Z}_j^T \mathbf{a}_j])) \right] \right\}, \tag{3.11}$$

where $m = 1, \cdots, M$. Note that the expectation is a conditional expectation conditioned on the $M-1$ subsets, and the estimator derived from the $M-1$ subsets. Therefore, the average prediction error is calculated by $PE(h) = (1/M) \sum_{m=1}^{M} PE_m(h)$ and the optimal bandwidth can be selected by minimizing $PE(h)$ among all candidates of h.

In the current example, a 10-fold cross-validation with PE defined by (3.11) is used to identify the optimal bandwidth h from the range between 40 to 100 seconds. The PE values for different bandwidth are given in Figure 3.8. By minimizing PE, the optimal bandwidth is selected to be 60 seconds, and the resulting estimation of varying coefficient is given in Figure 3.9. The red line represents the estimated varying coefficient $\hat{\gamma}_{11}$ and the dashed line represents its 95% confidence interval. It appears that the cumulative bond lifetime has a significantly positive impact on T cell signaling in the first 60 seconds, which agrees with the observations in Liu et al. (2014) and supports the conjecture made by biologists. Furthermore, the proposed method provides a mathematical quantification of the dynamic impact over time. More specifically, the estimation result shows a rapidly increasing impact from bond lifetime accumulation in the first 17 seconds, the impact reaches the peak

with estimated coefficient $\hat{\gamma}_{11}(17) = 0.2$. Then, the impact slowly decreases and remains relatively stable after 100 seconds. The estimated variance component is $\hat{\sigma}_a^2 = 0.353$. Its 95% confidence interval (0.171,0.536) provides clear evidence of the heterogeneity among experimental subjects. According to (3.8), the estimated maximum baseline hazard function is 0.013, which implies the maximum of $\lim_{\Delta t \to 0} P(t \leq T \leq t + \Delta t | X(t) = 0)$ is 0.002. This result indicates that without any impact from bond lifetime accumulation, the probability of T cell signaling is small.

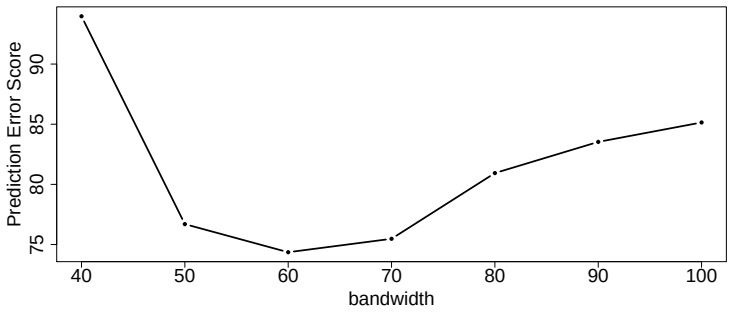

Figure 3.8: Optimal bandwidth selection based on the cross validation prediction errors in T cell signaling experiments

We further use a graphical tool based on the Cox-Snell residuals to assess the goodness-of-fit of the proposed LLVCF model. The residuals are defined by $\{\hat{\Lambda}(t_i)\}_{i=1}^n$, where $\hat{\Lambda}(\cdot)$ is the fitted cumulative hazard function. According to [97], if the model assumption is valid, then the residuals should have the same cumulative hazard rate as $\exp(1)$, which is the red dashed line passing through the origin with slope one in Figure 3.10. As shown in Figure 3.10, the cumulative hazard rate of the residuals (the black solid line) appears to be close to the red dashed line and the 95% confidence interval (the black dotted line) covers the red line. This provides a graphical support for the fitted model.

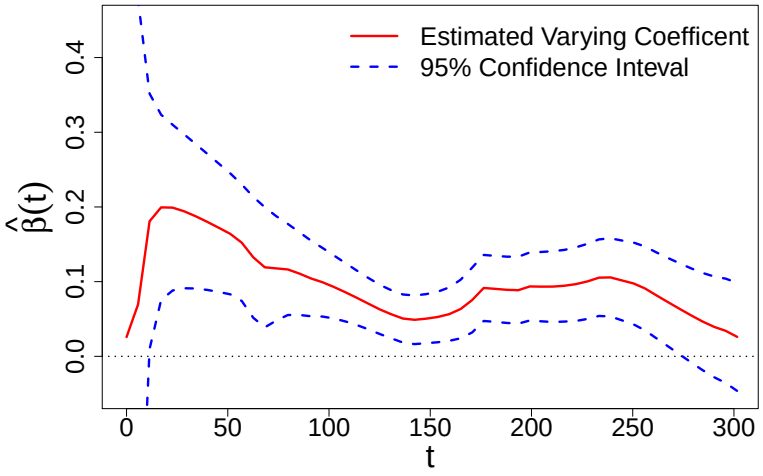

Figure 3.9: The estimated varying coefficient with $h = 0.1$ in the T cell signaling experiment. The red line represents $\hat{\gamma}_{11}(t)$ and the black dash line represents the 95% confidence interval. The horizontal dotted line represents zero effect.

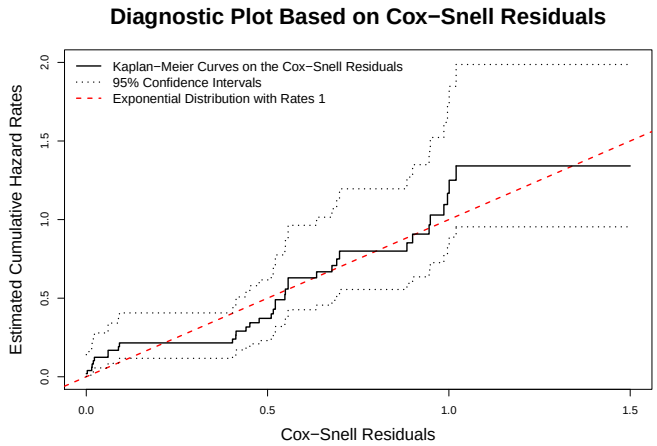

Figure 3.10: The Cox-Snell residual plot for the LLVCF model

3.7 Conclusions

Motivated by the analysis of T cell signaling, we introduce a new model called local linear varying coefficient frailty model. Estimation procedures and the asymptotic properties of the estimators including the bias correction property near the boundary are developed for

statistical inference. The bias-and-variance trade-off in bandwidth selection is discussed. Our numerical studies show that the local linear method outperforms the spline-based methods, especially near the boundary, and confirm the theoretical properties in finite-sample performance. The application of the proposed method to real data in T cell signaling experiments reveals important insights to the understanding of the immune system.

The proposed work lays a foundation for the generalization of conventional frailty models to incorporate varying coefficients. It can be extended to situations in which ties occur. It can be addressed by modifying the partial likelihood in (3.9) with approximation techniques introduced by [89], [98], [99], and [100]. Developments on estimation procedures and theoretical properties along this direction deserve further attention. Additionally, based on the connections between the Cox model and a Cured model ([101]), another research topic may be the extension of using the LLVCF model to estimate cured rates in clinical trials.

CHAPTER 4

OPTIMAL SIMULATOR SELECTION

Computer simulators are widely used for the study of complex systems. They serve as efficient alternatives to physical experiments and can provide scientific insights that may not be obtainable in physical experiments. In many applications, there are multiple simulators available with different scientific implications, and the goal is to identify an optimal simulator that better captures the underlying mechanism of the observed physical experiments. This issue is of significant scientific interest in different fields but there is no such procedure in the computer experiment literature. To address the problem, we propose a selection criterion based on leave-one-out cross-validation. It is shown that this criterion can be decomposed into a goodness-of-fit measure and a generalized degrees of freedom capturing the complexity of the simulator. Asymptotic properties of the selected optimal simulator are discussed. Additionally, it is shown that the proposed procedure includes a conventional calibration method as a special case. The finite sample performance of the proposed procedure is demonstrated through numerical examples. In the application of cell biology, an optimal simulator is selected which can shed light on the T cell recognition mechanism in the human immune system.

4.1 Introduction

There are generally two types of experiments for the studies of complex systems: physical and computer experiments. Physical experiments refer to actual experiments performed in a laboratory or observed in the field. They are often time-consuming, expensive, and/or infeasible to conduct. Therefore, an efficient alternative is to conduct computer experiments that refer to simulations using complex mathematical models and numerical tools. A conventional assumption in computer experiments is that there is only one simulator available,

and the goal is to build a model by incorporating the information from the simulator and the physical experiments. Detailed discussions can be found in [102].

There has been a growing interest in optimal simulator selection in many scientific applications where multiple simulators are available to explain the underlying phenomenon. These simulators often have different scientific implications, and scientists are interested in identifying an optimal simulator that better captures the underlying mechanism in the observed physical experiments. For example, among different queuing models, it is important to identify the best simulator for a particular medical service in a hospital [103, 104]. Geologists want to know which global weather model can be best used for predicting the weather of a local region [105]. Biologists need to select some differential equations to represent the growth (or decline) of a biological population [106]. However, the issue of optimal simulator selection has been overlooked in the statistical literature, and there is no systematic procedure for obtaining an optimal simulator.

The goal in optimal simulator selection is different from the variable selection problems in computer experiments [107, 108], where the focus is to identify significant variables by using one computer simulator. It is also different from studies of multi-fidelity simulations where multiple simulations are developed based on the same physical law but with different approximation accuracy, and the objective is to incorporate information efficiently from all the computer simulators [109, 110].

To identify an optimal simulator, we propose a new criterion based on leave-one-out cross-validation. It is shown that this criterion can be decomposed into a measure of goodness-of-fit for physical experiments and a generalized degrees of freedom capturing the complexity of the simulator due to calibration. Asymptotic properties of the selected optimal simulator are discussed. It is also shown that the proposed criterion includes the conventional L_2-norm calibration criterion [111] as a special case when there is only one simulator available.

4.2 Cross Validation for Optimal Simulator Selection

Assume that there are n observations available from physical experiments denoted by $\mathbf{D} \equiv \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i$ is a p-dimensional input. For notational simplicity, we first assume the outputs y_i's are continuous, and

$$y_i = \xi(\mathbf{x}_i) + \epsilon_i, \tag{4.1}$$

where $\xi(\mathbf{x}_i)$ is known as the *true process* in computer experiment literature, and ϵ_i are identically distributed random variables with zero mean and finite variance [111]. The true process ξ can be estimated by nonparametric regression methods, such as kernel ridge regression [112] and Gaussian process [102], and the estimator is denoted by $\hat{\xi}(\cdot)$. Apart from physical experiments, there are K candidate computer simulators $f_k(\mathbf{x}, \boldsymbol{\theta}_k)$, where $k = 1, \cdots, K$, and $\boldsymbol{\theta}_k$ is a set of unknown parameters called calibration parameters, associated with the kth simulator (Section 8 of [102]). The calibration parameters in each $\boldsymbol{\theta}_k$ can be different. For studies of complex systems, it is often infeasible to perform simulations for all experimental settings of interest due to the computational cost or complexity. Instead, the computer simulator is replaced by a statistical model called *emulator*. Therefore, given the computer experiment data $\mathbf{D}_k^s$ for a given k, it is assumed that an *emulator* denoted by $\hat{f}_k(\mathbf{x}; \boldsymbol{\theta}_k)$ is constructed as a surrogate for prediction, inference, and uncertainty quantification. Various methods in surrogate modeling are applicable here, including Gaussian process models [102] and spline-based models [113]. By incorporating the information from the physical experiments data $\mathbf{D}$, the estimated calibration parameter $\hat{\boldsymbol{\theta}}_k(\mathbf{D})$ is obtained by minimizing the discrepancy between the simulator and the data $\mathbf{D}$ [111].

Given the outputs from the K simulators and the observations from physical experiments, our goal is to identify the true simulator $f_0(\mathbf{x}; \boldsymbol{\theta}_0)$, which is defined by

$$f_0 = \min\{f_k : ||f_k(\mathbf{x}; \boldsymbol{\theta}_k) - \xi(\mathbf{x})||_{L_2}\}, \tag{4.2}$$

where $\boldsymbol{\theta}_0$ is the true calibration parameter, and $||\cdot||_{L_2}$ is the L_2 norm. Because solving (4.2) is computationally complicated and demanding, we propose a leave-one-out cross-validation (LOOCV) criterion to select an optimal simulator, $\hat{f}_T$, as follows. Define a LOOCV error by

$$\widehat{Err}_k = \frac{1}{n}\sum_{i=1}^{n} \widehat{Err}_{k,(i)}, \tag{4.3}$$

where $\mathbf{D}_{(-i)} = \mathbf{D}\backslash\{(\mathbf{x}_i, y_i)\}$, $\widehat{Err}_{k,(i)} \equiv Q\left\{\hat{\xi}(\mathbf{x}_i), \hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}_{(-i)}))\right\}$, $\hat{\boldsymbol{\theta}}_k(\mathbf{D}_{(-i)})$ is the estimated calibration parameters, and $i = 1, \cdots, n$. The function $Q(\hat{\xi}(\mathbf{x}_i), \hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D})))$ is used to quantify the prediction error at $\mathbf{x}_i$. It can be written as

$$Q(\hat{\xi}(\mathbf{x}_i), \hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}))) = q(\hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}))) + \dot{q}(\hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D})))(\hat{\xi}(\mathbf{x}_i) - \hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}))), \tag{4.4}$$

where $q(\cdot)$ is a convex function, and $\dot{q}(\cdot)$ is its derivative. A common choice for $q(\cdot)$ is $q(\mathbf{x}) = f_k(\mathbf{x})(1 - f_k(\mathbf{x}))$, which leads to the squared error loss. Other proper scoring rules can also be applied to $q(\cdot)$ [33].

Based on (4.3), the optimal simulator $\hat{f}_T$ can be obtained by

$$T \equiv \underset{k=1,\cdots,K}{\arg\min}\, \widehat{Err}_k. \tag{4.5}$$

The procedure is summarized in Algorithm 1. This procedure can also be generalized to non-Gaussian outputs. Take the binary output as an example, the same procedure follows by replacing the true process by $\xi(\mathbf{x}) = P(y(\mathbf{x}) = 1)$. A demonstration for applying Algorithm 1 to binary output is given in Section 5.

4.3 Theoretical Properties

In the following lemma, we show that (4.3) can be decomposed into the goodness-of-fit of the emulator and a quantity GD_k measuring the complexity of the emulator due to calibration. Therefore, minimizing (4.3) implies a minimization of not only the discrepancy

Algorithm 9 The algorithm for simulator selection

procedure LOOCV($\mathbf{D} \equiv \{(\mathbf{x}_i, y_i\}_{i=1}^n), \{\mathbf{D}_k^s : k = 1, \cdots, K\}$)
  Estimate the true process $\hat{\xi}(\mathbf{x}_i)$
  for each k in $1, 2, \cdots, K$ **do**
    for each i in $1, 2, \cdots, n$ **do**
      Use $\mathbf{D}_k^s$ to build an emulator $\hat{f}_k(\mathbf{x}; \boldsymbol{\theta}_k)$ where $\boldsymbol{\theta}_k$ is the calibration parameter.
      Obtain the estimated calibration parameter $\hat{\boldsymbol{\theta}}_k(\mathbf{D}_{(-i)})$.
      Calculate $\widehat{Err}_{k,(i)} = Q(\hat{\xi}(\mathbf{x}_i), \hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}_{(-i)})))$
    end for
    Obtain $T \equiv \arg\min_{k=1,\cdots,K} \widehat{Err}_k$, where $\widehat{Err}_k = \frac{1}{n}\sum_{i=1}^n \widehat{Err}_{k,(i)}$.
  end forreturn The optimal simulator $\hat{f}_T$.
end procedure

between physical and computer experiments but also the simulator complexity. The detailed proofs can be found in the Appendix.

Lemma 4.3.1.

$$E[\widehat{Err}_k] = e\bar{r}r_k + GD_k, \tag{4.6}$$

where

$$e\bar{r}r_k = \frac{1}{n}\sum_{i=1}^{n} Q(y_i, \hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}_{(-i)})))) \tag{4.7}$$

and

$$GD_k = \frac{1}{n}\sum_{i=1}^{n} E[\left\{ \dot{q}\left[\hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}_{(-i)}))\right](y_i - \hat{\xi}(\mathbf{x}_i))\right\}. \tag{4.8}$$

The quantity in (4.7) determines how well the emulator fits the physical observations and GD_k in (4.8) captures the complexity of the emulator due to calibration. The quantity GD_k is referred as the generalized degrees of freedom for the kth emulator which is analogous to optimism in [114] and the generalized degrees of freedom for linear model in [115]. Based on Lemma 4.3.1, GD_k can be estimated by $\widehat{GD}_k = \widehat{Err}_k - e\bar{r}r_k$. In the special case where the kth emulator is not associated with any calibration parameter, we have $GD_k = \frac{1}{n}\sum_{i=1}^n \dot{q}\left[\hat{f}_k(\mathbf{x}_i)\right] E[\left\{(y_i - \hat{\xi}(\mathbf{x}_i))\right\} = 0$ and $\widehat{GD}_k = 0$.

For the selected optimal simulator, the estimated prediction error and complexity are

denoted by $\widehat{Err}_T$ and $\widehat{GD}_T = \widehat{Err}_T - e\bar{r}r_T$. Given the true simulator $f_0(\mathbf{x}, \boldsymbol{\theta}_0)$ in (4.2), we denote its prediction error by $Err_0 = \frac{1}{n}\sum_{i=1}^{n} E\left\{Q(Y_i, f_0(\mathbf{x}_i; \boldsymbol{\theta}_0))\right\}$ and complexity by $GD_0 = \frac{1}{n}\sum_{i=1}^{n} E[\Big\{\dot{q}\left[f_0(\mathbf{x}_i; \boldsymbol{\theta}_0)\right](y_i - \hat{\xi}(\mathbf{x}_i))\Big\}$. In the following theorem, it is shown that the estimated prediction error $\widehat{Err}_T$ and the estimated model complexity $\widehat{GD}_T$ are asymptotically equivalent to those calculated based on the true simulator.

Theorem 4.3.2. *Suppose $\hat{\boldsymbol{\theta}}_T$ and $\hat{f}_T$ as consistent estimators of $\boldsymbol{\theta}_0$ and $\hat{f}_0$. Then as $n \to \infty$, we have*

(i) $\widehat{Err}_T - Err_0 \to 0$ *in probability, and*

(ii) $\widehat{GD}_T - GD_0 \to 0$ *in probability.*

The proposed selection procedure can also be applied to the conventional calibration problem with $K = 1$. The following result shows that the estimated calibration parameters and the resulting discrepancy based on the proposed leave-one-out procedure converge asymptotically to those obtained by the conventional L_2 calibration [111].

Theorem 4.3.3. *Assume that $K = 1$, $Q(\cdot)$ is the squared loss, and $\mathbf{x}$ follows a uniform distribution on $[0, 1]^p$. Denote $Err_{1,(i)}(\boldsymbol{\theta}_1) = Q\Big\{\hat{\xi}(\mathbf{x}_i), \hat{f}_1(\mathbf{x}_i; \boldsymbol{\theta}_1(\mathbf{D}_{(-i)}))\Big\}$. As $n \to \infty$, we have*

$$\arg\min_{\boldsymbol{\theta}_1} \frac{1}{n}\sum_{i=1}^{n} Err_{1,(i)}(\boldsymbol{\theta}_1) \to \arg\min_{\boldsymbol{\theta}_1} ||\hat{\xi}(\mathbf{x}) - \hat{f}_1(\mathbf{x}; \boldsymbol{\theta}_1))||_{L_2} \tag{4.9}$$

in probability, where $||\cdot||_{L_2}$ is the L_2 discrepancy between $\hat{\xi}(\mathbf{x})$ and $\hat{f}_1(\mathbf{x}; \boldsymbol{\theta}_1)$, and $\widehat{Err}_1$ converges in probability to the minimum L_2 discrepancy.

4.4 Numerical Studies

In this section, the true process $\xi(\mathbf{x})$ is estimated by the kernel ridge regression; that is, the estimator is the minimizer of the following loss function

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \xi(\mathbf{x_i}))^2 + \lambda||f||^2_{\mathcal{N}_\Phi}, \tag{4.10}$$

where $\lambda > 0$ is a penalized parameter, $||\cdot||_{\mathcal{N}_\Phi}$ is the norm of the reproducing kernel Hilbert space $\mathcal{N}_\Phi$ generated by a kernel function $\Phi(\cdot)$. We consider a Matèrn kernel $\Phi(\cdot)$ with roughness coefficient 2.5, i.e., $\Phi(h) = (1/[\Gamma(\nu)2^{\nu-1}])(2\sqrt{\nu}h^2)^\nu \mathcal{K}_\nu(2\sqrt{\nu}\phi h^2)$ with $\nu = 2.5$, where $\Gamma(\cdot)$ is the gamma function, $\mathcal{K}_\nu(\cdot)$ is the Bessel function, and ϕ is the range parameter. The penalized parameter λ in (4.10) and the range parameter ϕ are chosen by 10-fold cross-validation. Emulators are constructed by the Gaussian process (GP) models

$$f(\mathbf{x}) \sim GP(\mu, R_\phi((\mathbf{x}', \boldsymbol{\theta}'), (\mathbf{x}, \boldsymbol{\theta}))), \tag{4.11}$$

where μ is the unknown mean and $R_\phi((\mathbf{x}', \boldsymbol{\theta}'), (\mathbf{x}, \boldsymbol{\theta})) = \exp\left[-\phi\{(\mathbf{x}' - \mathbf{x}) + (\boldsymbol{\theta}' - \boldsymbol{\theta})\}\right]$. The calibration parameters are estimated by the Bayesian calibration procedure [116, 117].

4.4.1 Example 1: the Branin function

Two simulators are constructed by the Branin function with two different sets of calibration parameters:

$$\begin{aligned} f_1(x_1, x_2) &= \left(x_2 - bx_1^2 + 5.3x_1 - r\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10, \\ f_2(x_1, x_2) &= \left(x_2 - bx_1^2 + cx_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10, \end{aligned}$$

where simulator $f_1(x_1, x_2)$ contains the calibration parameters (b, r), simulator $f_2(x_1, x_2)$ contains the calibration parameters (b, c), $b \in [0, 2]$, $r \in [5, 7]$, and $c \in [4, 6]$. For both simulators, computer experiments are conducted by using a 60-run maximum projection design [36]. Simulator $f_2(x_1, x_2)$ is also used as the true process to generate physical experiments by $y = f_2(x_1, x_2) + \epsilon$, where $\epsilon \sim N(0, 4)$, with a 30-run design constructed by a Sobol sequence.

The true process is estimated by minimizing loss function (4.10) with the tuning parameter 10^5 and the range parameter 0.4, selected by 10-fold cross-validation. Based on

(4.3) and Lemma 4.3.1, the leave-one-out cross-validation scores for the two simulators are reported in Table 4.1 with the estimated generalized degrees of freedom. By using the proposed criterion, the selected optimal simulator is $T = 2$, which agrees with the numerical settings. Furthermore, the estimated generalized degrees of freedom for the two simulators are similar, which implies a similar complexity for the two simulators. This observation also agrees with the numerical settings in which equal number of calibration parameters are associated with the simulators.

Table 4.1: The leave-one-out cross-validation scores and the estimated generalized degrees of freedom for the two simulators in Example 1.

k	$\widehat{Err}_k$	$\widehat{GD}_k$
1	13.715	3.423
2	9.072	3.429

4.4.2 Example 2: multi-fidelity simulators

The proposed procedure is demonstrated by using two simulators introduced by [118] for the study of multi-fidelity simulations. Define the low-fidelity and high-fidelity simulators, f_1 and f_2, by

$$\begin{aligned} f_1(x_1, x_2; t_s, t_\ell) &= \left(1 - \exp\left(\frac{1}{-2x_2}\right)\right) \frac{1000 t_s x_1^3 + 1900 x_1^2 + 2092 x_1 + 60}{1000 t_\ell x_1^3 + 500 x_1^2 + 4 x_1 + 20}, \\ f_2(x_1, x_2; t_s, t_h) &= f_1(x_1, x_2; t_s, t_\ell = 0.1) + 5 \exp(-t_s) \frac{x_1^{t_h}}{100 x_2^{2+t_h} + 1}, \end{aligned}$$

where there are three calibration parameters: t_ℓ for the low-fidelity simulator, t_h for the high-fidelity simulator, and a shared calibration parameter t_s for both simulators, whose values are set to be $t_\ell = 0.1, t_h = 0.3$, and $t_s = 0.2$. The physical experiments are generated by

$$y(x_1, x_2) = f_2(x_1, x_2; t_s = 0.2, t_h = 0.3) + \frac{10 x_1^2 + 4 x_2^2}{50 x_1 x_2 + 10} + \epsilon$$

with $\epsilon \sim N(0, 0.25)$. These functions are shown in Figure 4.2. The goal here is to identify an optimal simulator based on the observed physical data. This is different from the conventional goal in the study of multi-fidelity simulations.

A 40-run maximum projection design is used for the two simulators, and the physical experiments are performed based on a 30-run Sobol' points. The true process is estimated by minimizing (4.10) with the range parameter ϕ in the Matèrn correlation function set to be 0.7. For the two simulators, the estimated leave-one-out cross-validation error and generalized degrees of freedom are reported in Table 4.2. The high-fidelity simulator f_2 has a much smaller cross-validation error and therefore is chosen as the optimal simulator. It provides a slightly higher model complexity as compared with the low-fidelity simulator according to the values of $\widehat{GD}_k$.

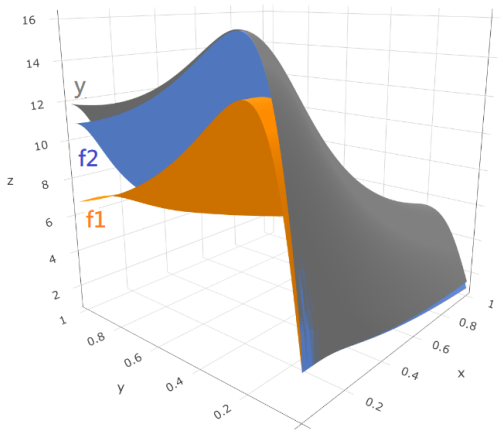

Figure 4.1: The response surfaces for the three functions in Example 2.

Table 4.2: The leave-one-out cross-validation scores and the estimated generalized degrees of freedom for the two simulators in Example 2.

k	$\widehat{Err}_k$	$\widehat{GD}_k$
1	3.149	5.03
2	0.283	5.21

4.4.3 Example 3: the study of simulator complexity

To demonstrate the performance of the generalized degrees of freedom with respect to different complexity in simulators, we consider two simulators with different numbers of calibration parameters: $f_1(x) = \beta_1 x + \beta_2 x^2 + \beta_3 x^3$ and $f_2(x) = \gamma_1 x$, where β_1, β_2, β_3, and γ_1 are the calibration parameters. Physical experiments are generated from $y(x) =$

$x + 2x^2 + 3x^3 + 0.1\sin(20x) + \epsilon$, where $\epsilon \sim N(0, 0.25)$. These functions are illustrated in Figure 4.2(a). A 100-run maximin design [119] is used to generate computer experiments based on the two simulators, and a 61-run maximin design is implemented for physical experiments.

Based on 100 replicates, the average of $\widehat{GD}_1$ is 2.737 with standard deviation 0.661, and the average of $\widehat{GD}_2$ is 1.371 with standard deviation 0.120. These results are summarized by a boxplot in Figure 4.2(b). The estimated generalized degrees of freedom for the first simulator almost doubles the size of the second one, which reflects the complexity associated with the first simulator due to a larger number of calibration parameters and a higher-order polynomial.

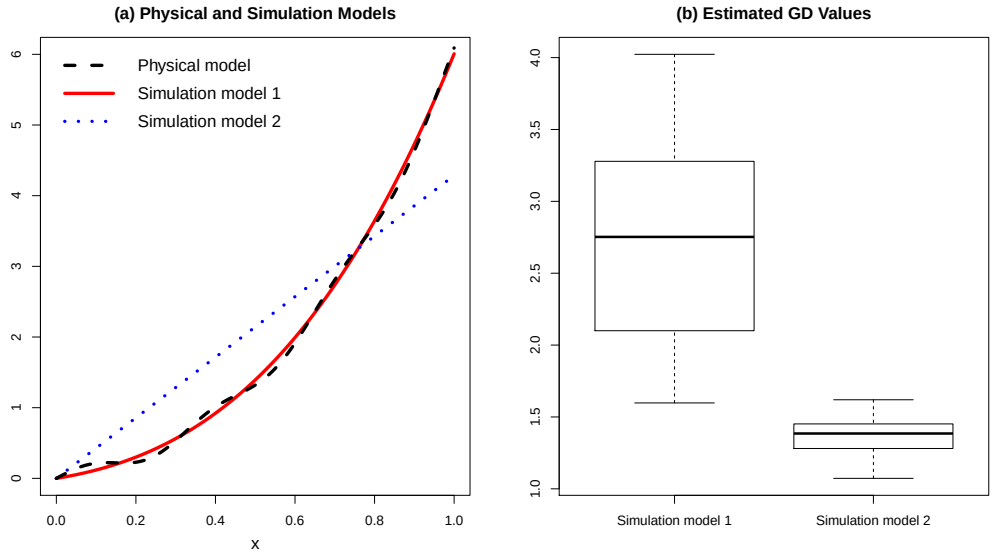

Figure 4.2: (a) The physical model and two simulators. (b) The estimates of the generalized degree of freedoms for the two simulators.

4.5 Optimal Simulator for T-cell Signaling

It has long been known that the adaptive immune system defends the organism against diseases by recognition of pathogens by the T cell. T cell receptor (TCR) is the primary molecule on the T cell in detecting foreign antigens which are present in major histocompatibility complex (pMHC) molecule expressed by infected cells. However, much is still

unknown regarding the underlying antigen recognition mechanism.

To understand the recognition mechanism through the TCR-pMHC interactions, biologists develop micropipette adhesion frequency assays which are physical experiments performed in a laboratory. Although micropipette assays allow accurate measurements, they are time-consuming and often involve complicated experimental manipulation. Furthermore, some variables of interest cannot be studied in the lab due to technical complexity in experimental settings. As a result, a cost-effective approach is to illuminate the unknown recognition mechanism through computer simulations. Based on the idea of the kinetic proofreading model, two simulators are developed under two different recognition mechanisms: one is the conformation-change mechanism (denoted by CC in Figure 4.3(A)), and the other is the receptor-pulling mechanism (denoted by RP in 4.3(B)). The two mechanisms associate with two different ways of TCR-pMHC interactions, either the molecules have conformational change due to the binding or involve force due to the pulling of the TCR-pMHC bond [120]. Biologists are interested in understanding which mechanism is behind the recognition process, but it cannot be directly detected by physical experiments. Therefore, the goal of this study is to identify the optimal mechanism based on the observed experimental data from the laboratory.

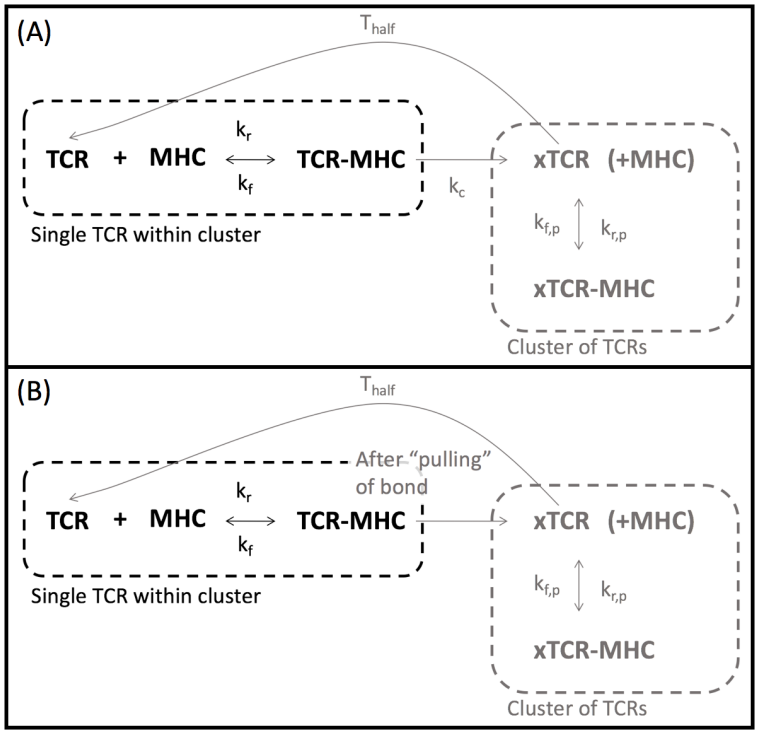

Figure 4.3: Two simulators capturing two biological mechanisms

Two control variables, the contact time x_{ct} and waiting time x_{wt}, are involved both in the lab experiments as well as the two simulators. Denote $\mathbf{x} = (x_{wt}, x_{ct})$. Four calibration parameters, denoted by x_{Kf}, x_{Kr}, $x_{Kr,p}$, and x_{Kc}, are involved in CC mechanism, while only the first three of them are involved in RP mechanism. The descriptions for the variables are given in Table 4.3, and further detail can be found in [121]. The two mechanisms are simulated by the Gillespie algorithm [122], which is a stochastic simulation algorithm. The experimental outputs are binary, indicating a TCR-pHMC binding or not. A 60-run OA-based Latin hypercube design (Tang, 1993) is implemented for the two simulators, and each design consists of 20 replicates to capture the cell-cell variability. Therefore, the sample size of the computer experiment is 1200 for each mechanism. For the physical experiments, the sample size is $n = 272$ and the settings of x_{ct} and x_{wt} are randomly chosen from the sample space $[0.25, 5] \times [1, 6]$.

Given the binary binding outcomes $y(\mathbf{x})$ observed in the laboratory, the true process is

Table 4.3: The range and description of Input variables in the T cell adhesion frequency assay experiments. (Note: s represents second)

Type of variables		Physical Experiments	Simulators		Description	Range
			CC	RP		
Control variables	x_{wt}	✓	✓	✓	waiting time in between contacts (s)	$[1, 6]$
	x_{ct}	✓	✓	✓	cell-cell contact time (s)	$[0.25, 5]$
Calibration Parameters	x_{Kc}		✓		kinetic proofreading rate for activation of cluster (1/s)	$[0.1, 100]$
	x_{Kf}		✓	✓	on-rate enhancement of inactive TCRs ($\mu m^2/s$)	$[10^{-8}, 10^{-10}]$
	x_{Kr}		✓	✓	off-rate enhancement of inactive TCRs ($1/s$)	$[0.1, 10]$
	$x_{r,p}$		✓	✓	off-rate enhancement of activated TCRs ($1/s$)	$[0.01, 100]$

defined as the binding probability, $\xi(\mathbf{x}) = P(y(\mathbf{x}) = 1)$, and estimated by a kernel logistic regression

$$logit\{\hat{\xi}(\mathbf{x})\} = \hat{\beta}_0 + \sum_{i=1}^{n} \hat{\beta}_i \Phi(\mathbf{x}_i, \mathbf{x}), \tag{4.12}$$

where $logit\{\cdot\}$ is the logistic link function, $\{\hat{\beta}_i\}_{i=1}^n$ are the estimated coefficients, and $\Phi(\mathbf{x}', \mathbf{x})$ is the Matérn kernel. Define $p(\mathbf{x}; \boldsymbol{\theta}) = P(f(\mathbf{x}; \boldsymbol{\theta}) = 1)$, and its emulators are constructed by the generalized Gaussian Process models [123]

$$logit\{p(\mathbf{x}; \boldsymbol{\theta})\} \sim GP(\mu, R_\phi((\mathbf{x}', \boldsymbol{\theta}'), (\mathbf{x}, \boldsymbol{\theta}))). \tag{4.13}$$

The calibration parameters are estimated by minimizing the L_2 discrepancy proposed by [121].

The leave-one-out cross-validation errors for the two simulators are summarized in Table 4.4 along with the estimated generalized degrees of freedom. The optimal simulator is the CC mechanism because its LOOCV is smaller, while its complexity is slightly higher than that for the RP mechanism. From a biological perspective, the selection of the CC mechanism indicates that the molecules have conformational changes due to the TCR-pMHC binding. Analyzing the CC mechanism using all the data, we have $\hat{\mu} = -1.112$, $\hat{\phi} = 0.311$, and $\hat{\boldsymbol{\theta}} = (1.559, 8.560 \times 10^{-7}, 1.443, 1.594)$. By plugging in the estimated calibration parameters, the simulated binding probability according to the CC mechanism (red dashed lines) in Figure 4.4 as a function of the two control variables, waiting time and contact time. It appears that the selected optimal emulator, CC mechanism, can reasonably

Table 4.4: The leave-one-out cross-validation error and the estimated degrees of freedom for the two simulators.

Simulator	LOOCV	Generalized degree of freedom
CC mechanism	0.097	5.376
RP mechanism	0.130	4.973

capture the trend observed in the lab experiments.

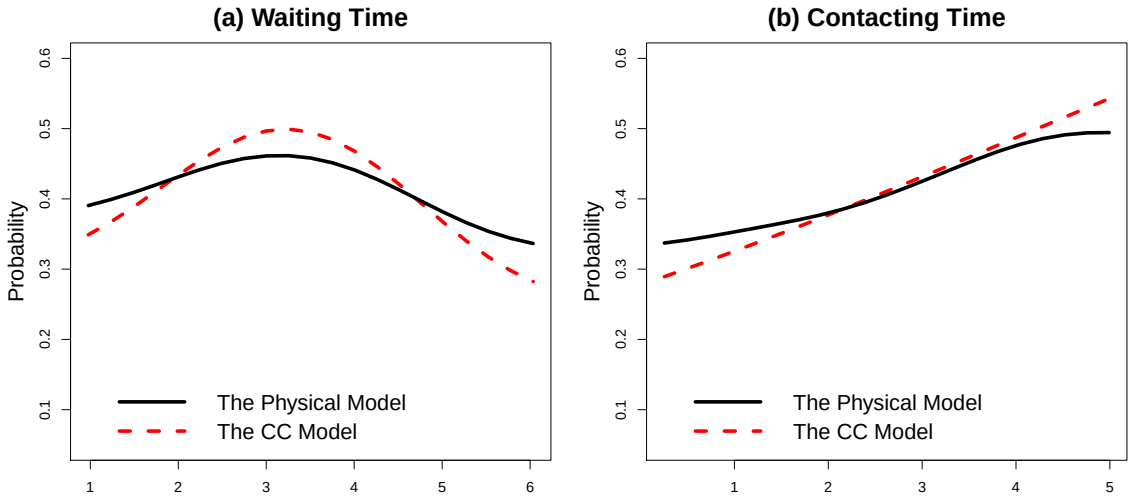

Figure 4.4: The fitted adhesion models from the phsical experiment and from the computer experiment of the CC model for two control variables: waiting time and contacting time.

4.6 Conclusions

In many applications, identifying an optimal simulator for the observed physical experiments can provide scientific insights that are not available from lab experiments. There is, however, no systematic statistical method to tackle this problem. We propose a new criterion based on the idea of leave-one-out cross-validation. Theoretical properties of the selection method based on the criterion and the estimated optimal simulator are discussed. It is also shown that asymptotically the proposed approach includes the L_2 calibration method as a special case. Simulation studies are conducted to demonstrate the performance of the proposed method. By applying the proposed method, the selected optimal T-cell signaling simulator reveals conformational changes in molecules due to the binding, which may shed

new light on the antigen recognition mechanism in human immune system.

Appendices

APPENDIX A

SUPPLEMENTAL MATERIAL FOR CHAPTER 1

A.1 The Initial Algorithm for the TAG Process Model

This subsection provides the details of obtaining initial estimates of the unknown parameters in the TAG process in Chapter 1. We use the *gam* or *bam* function in *mgcv* ([17]) to fit the additive model and use *DiceKriging* ([18]) to fit the one-dimensional GPs.

Algorithm 10 Initialization

procedure INITIAL($\{\boldsymbol{x}_i, y_i\}_{i=1}^n, \mathbf{D}$) ▷

Fit an additive model on $g_\lambda(\mathbf{y})$ for each $\lambda \in \{-2, -1.5, \ldots, 1.5, 2\}$ and then choose the λ to minimize the generalized cross-validation error. This gives $\hat{\lambda}^{(0)}$, the fitted additive model $g_{\hat{\lambda}^{(0)}}(\mathbf{y})$, and $\tilde{z}_k(x_k)$ for $k = 1, \ldots, p$.

$\delta^{(0)} = 1/R^2 - 1$, where R^2 is the fraction of the response variance explained by $g_{\hat{\lambda}^{(0)}}(\mathbf{y})$.

Obtain predictions of each component $\tilde{z}_i(x_i)$ at $\mathbf{D} = \{0, 1/(m-1), \ldots, 1\}$ with $m = 31$. Denote it as $\tilde{z}_i(\mathbf{D})$.

$\omega_i^{(0)} = \text{var}\left\{\tilde{z}_i(\mathbf{D})\right\} / \sum_{i=1}^p \text{var}\left\{\tilde{z}_i(\mathbf{D})\right\}$.

for i from 1 to p **do**

Obtain $s_i^{(0)}$ by fitting a GP on $\{\mathbf{D}, \tilde{z}_i(\mathbf{D})\}$.

end for

return $\boldsymbol{\omega}^{(0)}, \mathbf{s}^{(0)}, \lambda^{(0)}$, and $\delta^{(0)}$.

end procedure

A.2 The Details of the Computer Experiments Functions

In this subsection, we provide more details of the example functions and datasets used in section 1.5. The first 5 example functions can be found in [42] and the last two datasets are from [43].

1. The robot arm function describes the end position of a robot arm with 4 segements:

$$y = (u^2 + v^2)^{1/2}, u = \sum_{i=1}^{4} L_i \cos(\sum_{j=1}^{i} \theta_j), \text{ and } v = \sum_{i=1}^{4} L_i \sin(\sum_{j=1}^{i} \theta_j),$$

where the eight inputs are the segements $L_i \in [0, 1]$ and angles $\theta_i \in [0, 2\pi]$ for $i = 1, 2, 3$, and 4.

2. The OTL circuit function models an output transformerless push-pull circuit:

$$y = \frac{(V_{b1} + 0.74)\beta(R_{c2} + 9)}{\beta(R_{c2} + 9) + R_f} + \frac{11.35R_f}{\beta(R_{c2} + 9) + R_f} + \frac{0.74R_f\beta(R_{c2} + 9)}{\{\beta(R_{c2} + 9) + R_f\}R_{c1}} \text{ and } V_{b1} = \frac{12R_{b2}}{R_{b1} + R_{b2}},$$

where the six inputs with their ranges are $R_{b1} \in [50, 150], R_{b2} \in [25, 70], R_f \in [0.5, 3], R_{c1} \in [1.2, 2.5], R_{c2} \in [0.25, 1.2]$, and $\beta \in [50, 300]$.

3. The piston simulation function describes a piston moving within a cylinder:

$$y = 2\pi\sqrt{\frac{M}{k + S^2\frac{P_0V_0}{T_0}\frac{T_a}{V^2}}}, V = \frac{S}{2k}\left(\sqrt{A^2 + 4k\frac{P_0V_0}{V_0}T_a} - A\right), \text{ and } A = P_0S + 19.62M - \frac{kV_0}{S},$$

where the ranges of the seven variables are $M \in [30, 60], S \in [0.005, 0.02], V_0 \in [0.002, 0.01], k \in [1000, 5000], P_0 \in [90000, 110000], T_a \in [290, 296]$, and $T_0 \in [340, 360]$.

4. Wing weight function models a light aircraft wing:

$$y = 0.036S_w^{0.758}W_{fw}^{0.0035}\left(\frac{A}{\cos^2(\Lambda)}\right)^{0.6} q^{0.006}\lambda^{0.04}\left(\frac{100t_c}{\cos(\Lambda)}\right)^{-0.3}(N_zW_{dg})^{0.49} + S_wW_p,$$

where the ten input variables and their usual input ranges are $S_w \in [150, 200], W_{fw} \in [220, 300], A \in [6, 10], \Lambda \in [-10, 10], q \in [16, 45], \lambda \in [0.5, 1], t_c \in [0.08, 0.18], N_z \in [2.5, 6], W_{dg} \in [1700, 2500]$, and $W_p \in [0.025, 0.08]$.

5. The franke function describes a surface with two peaks of different heights and a

smaller dip:

$$y = \frac{3}{4}\exp(-\frac{(9x_1-2)^2}{4} - \frac{(9x_2-2)^2}{4}) + \frac{3}{4}\exp(-\frac{(9x_1+1)^2}{49} - \frac{(9x_2+1)^2}{10}) + \frac{1}{2}\exp(-\frac{(9x_1-7)^2}{4} - \frac{(9x_2-3)^2}{4}) - \frac{1}{5}\exp(-(9x_1-4)^2 - (9x_2-7)^2),$$

where the two inputs x_1 and x_2 are in $[0, 1]$.

6. The approximated HE dataset is used to design a heat exchanger to maximize the total rate of a steady state heat transfer. The dataset contains 64 simulations with 4 input variables including the mass flow rate of entry air $\dot{m} \in (.00055, .001)$, the temperature of entry air $T_{\text{in}} \in (270, 303.15)$, the temperature of the heat source $T_{\text{wall}} \in (202.4, 360)$ and the solid material thermal conductivity $k \in (330, 400)$. This dataset also includes 14 runs of simulations as a testing dataset. These datasets are from [43].

7. The detailed HE dataset is generated with the same goal as the approximated HE dataset but by a more expensive simulation dataset. The dataset contains 22 runs of simulations with the same 4 input variables as the approximated HE dataset.

APPENDIX B

SUPPLEMENTAL MATERIAL FOR CHAPTER 2

This supplemental material provides the details for obtaining the posterior distribution (2.8) and (2.9) used in Chapter 2. Given the sub-dataset $\mathbf{D}_{sub} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, the coefficient estimates of all the unknown coefficients $(\mu, \beta_{11}, \cdots, \beta_{pb_p}) \equiv \boldsymbol{\beta}$ in model (2.1) can be obtained by the penalized least square method (reference) if smoothing matrices $\mathbf{S}_k$ with smoothing parameters λ_k for $k = 1, \cdots, p$ are given. That is, by minimizing

$$\left[g(\mathbf{y}) - \mu\mathbf{1} - \sum_{k=1}^{p} \mathbf{X}_k\boldsymbol{\beta}_k\right]^T \left[g(\mathbf{y}) - \mu\mathbf{1} - \sum_{k=1}^{p} \mathbf{X}_k\boldsymbol{\beta}_k\right] + \sum_{k=1}^{p} \lambda_k\boldsymbol{\beta}_k^T\mathbf{S}_k\boldsymbol{\beta}_k, \tag{B.1}$$

where $g(\mathbf{y}) = [g(y_1)\cdots g(y_n)]^T$, the ij-th element of $\mathbf{X}_k$ is $h_{kj}(x_{ki})$, and the j-th element of $\boldsymbol{\beta}_k$ is β_{kj}. We have

$$\hat{\boldsymbol{\beta}} = \left[\mathbf{X}^T\mathbf{X} + \sum_{k=1}^{p} \lambda_k\mathbf{S}_k\right]^{-1} \mathbf{X}^T\left(g(\mathbf{y}) - \mu\mathbf{1}\right), \tag{B.2}$$

where $\mathbf{X} = (\mathbf{1}\ \mathbf{X}_1 \cdots \mathbf{X}_p)$, $\mathbf{1}$ is a vector of 1's having length n. The estimator $\hat{\boldsymbol{\beta}}$ in (B.2) is equivalent to the posterior mean of a Bayesian model, which can be expressed as $g(\mathbf{y})|\boldsymbol{\beta} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2)$ and $\boldsymbol{\beta}$ is an exponential prior proportional to $\exp(-\sum_{k=1}^{p} \lambda_k\boldsymbol{\beta}_k^T\mathbf{S}_k\boldsymbol{\beta}_k/\sigma^2)$. Thus, the posterior distribution of $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta}|\mathbf{y} \sim N\left(\left(\mathbf{H}^T\mathbf{H} + \sum_{k=1}^{p} \lambda_k\mathbf{S}_k\right)^{-1} \mathbf{H}^T\left\{g_\lambda(y(\mathbf{x})) - \mu\mathbf{1}\right\}, \hat{\sigma}^2\left(\mathbf{H}^T\mathbf{H} + \sum_{k=1}^{p} \lambda_k\mathbf{S}_k\right)^{-1}\right),$$

which is (1.15).

B.1 The Details of the Computer Experiments Functions

In this subsection, we provide more details of the example functions and datasets used in section 2.5.

Name	p	Information	Input Ranges
Bending Function	2	A computer model with function form $\frac{4}{10^9}\frac{L^3}{bh^3}$	$L \in [10, 20]$ $h \in [0.1, 0.2]$ $b \in [1, 2]$
Ackley Function	2	A computer model with function form $-20e^{\left(-\frac{2}{10}\sqrt{\frac{1}{2}\sum_{i=1}^{2} x_i^2}\right)} - e^{\left(\frac{1}{2}\sum_{i=1}^{2}\cos 2\pi x_i\right)} + 20 + e$	$x_i \in [-32/768, 32.768]$ for $i = 1, 2$
Schwefel Function	2	A computer model with function form $837.9658 - \sum_{i=1}^{2} x_i \sin\left(\sqrt{\lvert x\rvert_i}\right)$	$x_i \in [-500, 500]$ for $i = 1, 2$
Borehole Function	8	A computer model with function form $\frac{2\pi T_u(H_u - H_\ell)}{ln(\frac{r}{r_w})\left(1+\frac{2LT_u}{ln(\frac{r}{r_w})r_w^2 K_w}+\frac{T_u}{T_l}\right)}$	$r_w \in [0.05, 0.15]$ $r \in [100, 50000]$ $T_u \in [63070, 115600]$ $H_u \in [990, 1110]$ $T_l \in [63.1, 116]$ $H_l \in [700, 820]$ $L \in [1120, 1680]$ $K_w \in [9855, 12045]$

Table B.1: The numerical examples used in Section 1.5

Name	p	Information	Input Ranges
CCPP	4	The dataset contains 9568 data points collected from a Combined Cycle Power Plant. Features consist of hourly average ambient variables Temperature (T), Ambient Pressure (AP), Relative Humidity (RH), and Exhaust Vacuum (V) to predict the net hourly electrical energy output (EP) of the plant.	$T \in [1.81, 37.11]$ $AP \in [992.89, 1033.30]$ $RH \in [25.56, 81.56]$ $EP \in [420.26, 495.76]$
NASA	5	The NASA data set comprises different size of airfoils at various wind tunnel speeds and angles of attack. Features consist of Frequency (F), Angle of attack (AA), Chord length (CL), Free-stream velocity (FSV), and Suction side displacement thickness (DT).	$F \in [200, 20000]$ $AA \in [0, 22.2]$ $CL \in [0.0254, 0.3048]$ $FSV \in [31.7, 71.3]$ $DT \in [0.0004, 0.0584]$
PTS	9	This is a data set of Physicochemical Properties of Protein Tertiary Structure, whose 9 material properties are recorded and denoted by $F_1, \cdots, F_9$.	$F_1 \in [2392.05, 40034.90]$ $F_2 \in [403.5, 15312.0]$ $F_3 \in [0.093, 0.578]$ $F_4 \in [10.310, 369.317]$ $F_5 \in [319490.2, 5472011.4]$ $F_6 \in [31.9704, 598.4080]$ $F_7 \in [0, 105948.2]$ $F_8 \in [0, 350]$ $F_9 \in [15.229, 55.309]$

Table B.2: The numerical examples used in Section 2.5

APPENDIX C

SUPPLEMENTAL MATERIAL FOR CHAPTER 3

This supplemental material for Chapter 3 includes the conditions assumed in the theorems, the detailed proofs of the theoretical results, some technical details on implementing the E-step of the proposed algorithm, and more simulations about the comparison of the local linear varying coefficient frailty (LLVCF) method and a spline-based method.

C.1 Conditions for the Theorems

A1. The kernel function $K(s)$ is a bounded and symmetric density function with compact support $s \in [-1, 1]$.

A2. The function $\boldsymbol{\beta}(\cdot)$ has continuous and bounded second-order derivatives for $t \in (0, \tau)$.

A3. The sequence $h \to 0$ and $nh \to \infty$ as $n \to \infty$, and nh^5 is bounded.

A4. The hazard function $\lambda(t|\mathbf{X}(t), \mathbf{Z}(u)) < \infty$ for $t \in (0, \tau)$ is continuous, $\int_0^\tau \lambda_0(u)du < \infty$, and $P(Y \geq u|\mathbf{X}(u), \mathbf{Z}(u)) > 0$

A5. $|\mathbf{X}(u)|$ is bounded.

A6. $\mathbf{I}(t)$ from (11) is nonsingular $t \in [0, \tau]$.

A7. For any $\epsilon > 0$, $\sum_{i=1}^n E(||\mathbf{g}_i||_2^2 [I(\mathbf{g}_i) > \epsilon]) \to 0$, where $\mathbf{g}_i = vec(E[\mathbf{a}_i \mathbf{a}_i^T | \mathbf{D}, \hat{\theta}])$ for i in $1, \cdots, n$.

Conditions A1 to A6 are used for proving the asymptotical normality of $\boldsymbol{\gamma}_1$. To prove the consistency and asymptotic normality of $\hat{\Sigma}_a$, we need an extra condition A7.

C.2 Proofs of the Theorems

We first introduce the general notations used in the following content of this section.

Notation

Let $\mathbf{B}$ be a block diagonal matrix diag$\{\mathbf{I}_p, h\mathbf{I}_p\}$, where $\mathbf{I}_p$ is a $p \times p$ identity matrix, $\boldsymbol{\alpha} = \mathbf{B}(\hat{\boldsymbol{\gamma}}(t) - \boldsymbol{\gamma}_0(t))$, where $\boldsymbol{\gamma}_0(t)$ is the true values , $\tilde{\mathbf{U}}_i(u, u-t) = \mathbf{B}^{-1}\tilde{\mathbf{X}}_i(u, u-t)$, and $p(u|\mathbf{X}(u)) = P(Y \geq t|\mathbf{X}(t))$. Let

$$S_{n,j}(\boldsymbol{\alpha}, u) = n^{-1}\sum_{i=1}^{n} Y_i(u)\exp(\tilde{\mathbf{X}}_i^T(u, u-t)\boldsymbol{\gamma}_0(t) + \tilde{\mathbf{U}}_i^T(u, u-t)\boldsymbol{\alpha} + \Delta_i)\tilde{\mathbf{U}}^{\otimes j}(u, u-t) \tag{C.1}$$

for $j = 0, 1, 2$, where $\Delta_i = \log E[\exp(\mathbf{Z}_i^T\mathbf{a})|\mathbf{D}, \hat{\boldsymbol{\lambda}}(\cdot), \hat{\boldsymbol{\gamma}}_1(\cdot), \hat{\boldsymbol{\Sigma}}_a]$ and for a vector $\mathbf{v}$, $\mathbf{v}^{\otimes 0} = 1, \mathbf{v}^{\otimes 1} = \mathbf{v}$, and $\mathbf{v}^{\otimes 2} = \mathbf{v}\mathbf{v}^T$, and

$$S_{n,j}^{\star}(u) = n^{-1}\sum_{i=1}^{n} Y_i(u)\exp(\tilde{\mathbf{X}}_i^T(u, u-t)\boldsymbol{\gamma}_0(t) + \Delta_i)\tilde{\mathbf{U}}_i^{\otimes j}(u, u-t). \tag{C.2}$$

We also define the asymptotic version of (C.1) and (C.2); that is, for $j = 0, 1$, and 2, we have

$$S_j(\boldsymbol{\alpha}, u) = E[p(u|\mathbf{X}(u))\exp(\tilde{\mathbf{X}}^T(u, u-t)\boldsymbol{\gamma}_0(t) + \tilde{\mathbf{U}}^T(u, u-t)\boldsymbol{\alpha} + Z^T\mathbf{a})\tilde{\mathbf{U}}^{\otimes j}(u, u-t)],$$

and

$$S_j^{\star}(u) = E[p(u|\mathbf{X}(u))\exp(\tilde{\mathbf{X}}^T(u, u-t)\boldsymbol{\gamma}_0(t) + Z^T\mathbf{a})\tilde{\mathbf{U}}^{\otimes j}(u, u-t)].$$

Note that $S_{n,0}$, $S_{n,0}^{\star}$, and S_0 are scalars, $S_{n,1}$ and $S_{n,1}^{\star}$ are $2p$-dimensional vectors, and $S_{n,2}$ and $S_{n,2}^{\star}$ are $2p \times 2p$ matrices.

Before presenting the proofs, we need the following lemma [83].

Theorem C.2.1. *Let W_n be $n^{-1}\sum_{i=1}^{n} Y_i(t)g(t_i, (t_i - t_0)/h, Z_i(t))K_h(t_i - t_0)$, where $g(\cdot, \cdot, \cdot)$ is a continuous function and $E(g(T, u, Z(t))|T = t_0)$ is continuous at the point t_0 in the*

interior point. If $h \to 0$ *such that* $nh/\log(n) \to \infty$, *then*

$$\sup_{0 \le t \le \tau} |W_n(t) - W(t)| \xrightarrow{P} 0, \tag{C.3}$$

where $W(t) = \int_{-1}^{1} Y(t)E(g(t_0, u, Z(t))|t = t_0)K(u)duf(t_0)$ *and* $f(\cdot)$ *is the density function of* T.

C.2.1 Proof of Theorem 3.4.1

We first rewrite the likelihood function (9) in the main paper by using counting process as follows:

$$\sum_{i=1}^{n} \int_0^{\tau} K_h(u-t) \Bigg[\tilde{\mathbf{X}}_i^T(u, u-t)\boldsymbol{\gamma}(t) - \log \left\{ \sum_{i=1}^{n} Y_i(s) \exp(\tilde{\mathbf{X}}_j^T(u, u-t)\boldsymbol{\gamma}(t) + \Delta_i) \right\} \Bigg] dN_i(u), \tag{C.4}$$

where $N_i(t) = I(Y_i \le t, d_i = 1)$, $Y_i(t) = I(Y_i \ge t)$, and $\Delta_i = \log E[\exp(\mathbf{Z}_i^T \mathbf{a}_i)|\mathbf{D}, \hat{\boldsymbol{\gamma}}(t)]$. To further facilitate technical arguments, we reparameterize the local likelihood function expressed in (C.4) via $\boldsymbol{\alpha} = \mathbf{B}(\boldsymbol{\gamma}(t) - \boldsymbol{\gamma}_0(t))$, where $\mathbf{B}$ = diag$(\mathbf{I}_{p\times p}, h\mathbf{I}_{p\times p})$ and $\boldsymbol{\gamma}_0(t)$ is the true values. For simplicity, we use $\boldsymbol{\gamma}_0$ instead of $\boldsymbol{\gamma}_0(t)$ in the following content. Then, (C.4) is re-expressed as

$$\begin{aligned} l_n(\boldsymbol{\alpha}, \tau) &= \int_0^{\tau} K_h(u-t) n^{-1} \sum_{i=1}^{n} \left[\tilde{\mathbf{X}}_i^T(u, u-t)\boldsymbol{\gamma} + \tilde{\mathbf{U}}_i^T(u, u-t)\boldsymbol{\alpha} \right] dN_i(u) \\ &\quad - \int_0^{\tau} K_h(u-t) \log\{nS_{n,0}(\boldsymbol{\alpha}, u)\} d\bar{N}(u), \end{aligned}$$

where $\bar{N}(u) = n^{-1}\sum_{i=1}^{n} N_i(u)$. Through direct calculations, we observe

$$l_n(\boldsymbol{\alpha},\tau)-l_n(\mathbf{0},\tau) = \int_0^\tau K_h(u-t)n^{-1}\sum_{i=1}^{n}\tilde{\mathbf{U}}_i^T(u,u-t)\boldsymbol{\alpha}dN_i(u)-\int_0^\tau K_h(u-t)\log\{\frac{S_{n,0}(\boldsymbol{\alpha},u)}{S_{n,0}(\mathbf{0},u)}\}d\bar{N}(u). \tag{C.5}$$

To apply the martingale theory, the process needs to be associated with the statistical information accruing during the time [0,τ], namely, the filtration $\mathcal{F}_{n\tau} = \sigma\{\mathbf{X}_i(u), N_i(u), Y_i(u), i = 1,\cdots,n, 0 \le u \le \tau\}$. Thus, under the independent censoring scheme,

$$M_i(t) \equiv N_i(t) - \int_0^t Y_i(u)\lambda(u|\mathbf{X}_i)du \tag{C.6}$$

is an $\mathcal{F}_{n\tau}$-martingale. Substituting (C.6) into (C.5) gives

$$l_n(\boldsymbol{\alpha},\tau) - l_n(\mathbf{0},\tau) = G_n(\boldsymbol{\alpha}) + J_n(\boldsymbol{\alpha},\tau), \tag{C.7}$$

where

$$G_n(\boldsymbol{\alpha},\tau) = \int_0^\tau K_h(u-t)\left[S_{n,1}^\star(u)^T\boldsymbol{\alpha} - \log\left\{\frac{S_{n,0}(\boldsymbol{\alpha},u)}{S_{n,0}(\mathbf{0},u)}\right\}S_{n,0}^\star(u)\right]\lambda_0(u)du \tag{C.8}$$

and

$$J_n(\boldsymbol{\alpha},\tau) = n^{-1}\sum_{i=1}^{n}\int_0^\tau K_h(u-t)\left[\tilde{\mathbf{U}}_i^T(u,u-t)\boldsymbol{\alpha} - \log\left\{\frac{S_{n,0}(\boldsymbol{\alpha},u)}{S_{n,0}(\mathbf{0},u)}\right\}\right]\lambda_0(u)dM_i(u) \tag{C.9}$$

For proving the asymptotic normality of $\hat{\boldsymbol{\gamma}}_1(t)$, we consider $\boldsymbol{\alpha} = (nh)^{-1/2}\mathbf{B}(\boldsymbol{\gamma}(t) - \boldsymbol{\gamma}_0(t))$. Then, from (C.7), we have

$$l_n((nh)^{-1/2}\boldsymbol{\alpha},\tau) - l_n(\mathbf{0},\tau) = G_n((nh)^{-1/2}\boldsymbol{\alpha},\tau) + J_n((nh)^{-1/2}\boldsymbol{\alpha},\tau), \tag{C.10}$$

where $G_n(\cdot,\tau)$ and $J_n(\cdot,\tau)$ are defined in (C.8) and (C.9), respectively.

We first reexpress (C.10) to have a quadratic form in $\boldsymbol{\alpha}$. Through using second order

Taylor's expansion on $\log\{S_{n,0}((nh)^{-1/2}\boldsymbol{\alpha}, u)/S_{n,0}(\mathbf{0}, u)\}$, the common term in $G_n((nh)^{-1/2}\boldsymbol{\alpha}, \tau)$ and $J_n((nh)^{-1/2}\boldsymbol{\alpha}, \tau)$ at $\boldsymbol{\alpha} = \mathbf{0}$, we have

$$\log\left\{\frac{S_{n,0}((nh)^{-1/2}\boldsymbol{\alpha}, u)}{S_{n,0}(\mathbf{0}, u)}\right\} = \frac{S_{n,1}(\mathbf{0}, u)}{(nh)^{1/2}S_{n,0}(\mathbf{0}, u)}\boldsymbol{\alpha} + \frac{1}{2}(nh)^{-1}\boldsymbol{\alpha}^T\left[\frac{S_{n,2}(\mathbf{0}, u)}{S_{n,0}(\mathbf{0}, u)} - \frac{S_{n,1}(\mathbf{0}, u)^{\otimes 2}}{S_{n,0}^2(\mathbf{0}, u)}\right]\boldsymbol{\alpha} + o_p((nh)^{-1}). \tag{C.11}$$

By Lemma 1 on, (C.11) becomes

$$\log\left\{\frac{S_{n,0}((nh)^{-1/2}\boldsymbol{\alpha}, u)}{S_{n,0}(\mathbf{0}, u)}\right\} = \frac{S_1(\mathbf{0}, u)^T}{(nh)^{1/2}S_0(\mathbf{0}, u)}\boldsymbol{\alpha} + \frac{1}{2}(nh)^{-1}\boldsymbol{\alpha}^T\left[\frac{S_2(\mathbf{0}, u)}{S_0(\mathbf{0}, u)} - \frac{S_1(\mathbf{0}, u)^{\otimes 2}}{S_0^2(\mathbf{0}, u)}\right]\boldsymbol{\alpha} + o_p((nh)^{-1}). \tag{C.12}$$

After subsbstituting (C.12) into (C.8) with applying Lemma 1 again to $S_{n,0}^\star(u)$ and $S_{n,1}^\star(u)$ in (C.8), $G_n(\cdot, \tau)$ becomes

$$G_n((nh)^{-1/2}\boldsymbol{\alpha}, \tau) = (nh)^{-1/2}G_{n,1}(\tau)^T\boldsymbol{\alpha} - \frac{1}{2}(nh)^{-1}\boldsymbol{\alpha}^T F_{n,1}(\tau)\boldsymbol{\alpha} + o_p((nh)^{-1}),$$

where $G_{n,1}(\tau) = \int_0^\tau K_h(u-t)\left[S_1^\star(u) - (S_1(\mathbf{0}, u)/S_0(\mathbf{0}, u))S_0^\star(u)\right]\lambda_0(u)du$ and

$$F_{n,1}(\tau) = \int_0^\tau K_h(u-t)\left[\frac{S_2(\mathbf{0}, u)}{S_0(\mathbf{0}, u)} - \frac{S_1(\mathbf{0}, u)^{\otimes 2}}{S_0^2(\mathbf{0}, u)}\right]S_0^\star(u)\lambda_0(u)du. \tag{C.13}$$

Applying Lemma 1 to (C.13), we derive

$$F_{n,1}(\tau) = \boldsymbol{I}(t) \otimes \boldsymbol{\Omega} + o_p(1), \tag{C.14}$$

where $\boldsymbol{I}(t)$ is the the Fisher information matrix of $\hat{\gamma}_1(t)$ defined in Theorem 4.1, and Ω is a 2×2 matrix with its (i, j)-th elements being $\int u^{i+j-2}K(u)du$. Using (C.14),

$G_n((nh)^{-1/2}\boldsymbol{\alpha},\tau)$ can be written as

$$G_n((nh)^{-1/2}\boldsymbol{\alpha},\tau) = (nh)^{-1/2}G_{n,1}(\tau)^T\boldsymbol{\alpha} - \frac{1}{2}(nh)^{-1}\boldsymbol{\alpha}^T(\boldsymbol{I}(t)\otimes\Omega)\boldsymbol{\alpha} + o_p((nh)^{-1}). \quad \text{(C.15)}$$

Similarly, for $J_n((nh)^{-1/2}\boldsymbol{\alpha},\tau)$, we also substitute (C.12) into (C.9) and derive

$$J_n((nh)^{-1/2}\boldsymbol{\alpha},\tau) = (nh)^{-1/2}J_{n,1}(\tau)^T\boldsymbol{\alpha} - \frac{1}{2}(nh)^{-1}\boldsymbol{\alpha}^T F_{n,2}(\tau)\boldsymbol{\alpha} + o_p((nh)^{-1}),$$

where

$$J_{n,1}(\tau) = n^{-1}\sum_{i=1}^{n}\int_0^\tau K_h(u-t)\left[\tilde{\mathbf{U}}_i(u,u-t) - \frac{S_{n,1}(\mathbf{0},u)}{S_{n,0}(\mathbf{0},u)}\right]dM_i(u)$$

and

$$F_{n,2}(\tau) = \int_0^\tau K_h(u-t)\left[\frac{S_2(\mathbf{0},u)}{S_0(\mathbf{0},u)} - \frac{S_1(\mathbf{0},u)^{\otimes 2}}{S_0^2(\mathbf{0},u)}\right]d\bar{M}(u)$$

with $\bar{M}(u) = \frac{1}{n}\sum_{i=1}^n M_i(u)$. Through some direct calculation, we have $F_{n,2}(\tau) = O_p(\gamma_n)$ and derive

$$J_n((nh)^{-1/2}\boldsymbol{\alpha},\tau) = (nh)^{-1/2}J_{n,1}(\tau)^T\boldsymbol{\alpha} + o_p((nh)^{-1}). \quad \text{(C.16)}$$

By combing the results in (C.16) and (C.15), we obtain

$$l_n((nh)^{-1/2}\boldsymbol{\alpha},\tau) - l_n(\mathbf{0},\tau) = [G_n(\cdot,\tau)+J_n(\cdot,\tau)]^T(nh)^{-1/2}\boldsymbol{\alpha} - \frac{1}{2}(nh)^{-1}\boldsymbol{\alpha}^T\boldsymbol{I}(t)\otimes\boldsymbol{\Omega}\boldsymbol{\alpha} + o_p((nh)^{-1}). \quad \text{(C.17)}$$

Now, let $\hat{\boldsymbol{\alpha}}$ be the maximizer of $l_n((nh)^{-1/2}\boldsymbol{\alpha},\tau)$ with respect to $\boldsymbol{\alpha}$. By the quadratic approximation Lemma [84, p.210] on (C.17), we obtain

$$\hat{\boldsymbol{\alpha}} = (nh)^{-1/2}\boldsymbol{\alpha}^T(\boldsymbol{I}(t)\otimes\Omega)^{-1}\left[G_{n,1}(\tau) + J_{n,1}(\tau)\right] + o_p(1). \quad \text{(C.18)}$$

Since $(\mathbf{\Sigma}(t) \otimes \Omega)^{-1} = (\mathbf{\Sigma}^{-1}(t) \otimes \Omega^{-1})$, the first p component of (C.18) yields

$$(nh)^{-1/2}(\hat{\boldsymbol{\gamma}}_1 - \boldsymbol{\gamma}_{01}) = (nh)^{-1/2}\mathbf{\Sigma}^{-1}(t)[G^{\star}(n,1)(\tau) + J^{\star}_{n,1}(\tau)] + o_p(1), \tag{C.19}$$

where

$$G^{\star}_{n,1}(\tau) = n^{-1}\sum_{i=1}^{n}\int_0^{\tau} K_h(u-t)\left[\mathbf{A}(u) - \tilde{S}_1(\mathbf{0},u)\lambda_0(u)\frac{S^{\star}_0(u)}{S_0(\mathbf{0},u)}\right]du \tag{C.20}$$

with $\tilde{S}_1(u) = E\left\{p(u|\mathbf{X}(u))\exp(\tilde{\mathbf{X}}(u,u-t)^T\boldsymbol{\gamma} + \Delta_i)\mathbf{X}(u)\right\}$, $\mathbf{A}(u) = E\left\{p(u|\mathbf{X}(u))\lambda(u)\mathbf{X}(t)\right\}$, and $\lambda(u)$ is the hazard function, and

$$J^{\star}_{n,1}(\tau) = \int_0^{\tau} K_h(u-t)n^{-1}\sum_{i=1}^{n}\left[\mathbf{X}_i(u) - \frac{\tilde{S}_{n,1}(u)}{S_{n,0}(\mathbf{0},u)}\right]dM_i(u) \tag{C.21}$$

with $\tilde{S}_{n,1}(u) = n^{-1}\sum_{i=1}^{n} Y_i(u)\exp(\tilde{\mathbf{X}}(u,u-t)^T\boldsymbol{\beta} + \Delta_i)\mathbf{X}(u)$.

The bias of $\hat{\boldsymbol{\gamma}}_1$ can be derived from (C.20). That is, by applying Taylor's expansion on $\mathbf{A}(u) - \tilde{S}_1(u)\lambda_0(u)\frac{S^{\star}_0(u)}{S_0(\mathbf{0},u)}$ in (C.20) around t and some direct calculations, we have

$$\mathbf{A}(u) - \tilde{S}_1(u)\lambda_0(u)\frac{S^{\star}_0(u)}{S_0(\mathbf{0},u)} = \frac{1}{2}(u-t)^2\mathbf{I}(u)\hat{\boldsymbol{\gamma}}_1{}''(\tau) + o_p(h^2). \tag{C.22}$$

Then, plug (C.22) into of (C.19), (C.19) can be rewritten as

$$\sqrt{nh}\left[\hat{\boldsymbol{\gamma}}_1 - \boldsymbol{\gamma}_{01} - \frac{h^2}{2}\mu_2\boldsymbol{\gamma}_{01}{}''(t)\right] = \mathbf{I}^{-1}(t)(nh)^{-1/2}J^{\star}_{n,1}(\tau) + o_p(1). \tag{C.23}$$

The process $\tilde{J}_n(u) \equiv (nh)^{1/2}J^{\star}_{n,1}(u)$ is a (local) square integrable martingale with its predictable variation process expressed as

$$\langle\tilde{J}_n(u), \tilde{J}_n(u)\rangle(\tau) = \frac{h}{n}\sum_{i=1}^{n}\int_0^{\tau} K^2(u-t)\left[\mathbf{X}_i(u) - \frac{\tilde{S}_{n,1}(u)}{S_{n,0}(\mathbf{0},u)}\right]^{\otimes 2} Y_i(u)\exp(\mathbf{X}_i^T\boldsymbol{\beta}+\Delta_i)\lambda_0(u)du. \tag{C.24}$$

By applying Lemma 1 to (C.24), we derive

$$\langle \tilde{J}_n(u), \tilde{J}_n(u) \rangle(\tau) = \int K^2(u) du \mathbf{I}(t) + o_p(1). \tag{C.25}$$

With (C.25) and a proof similar to that of [124], the Linderberg condition for the process $\tilde{J}_n(u)$ holds. So, by the martingale central limit theorem, we establish

$$(nh)^{1/2} J^{\star}_{n,1}(u) \xrightarrow{D} N(0, \int_{-1}^{1} K^2(s) ds \mathbf{I}(t)), \tag{C.26}$$

in distribution for $t \in [0, \tau]$. Therefore, with (C.26), (C.23) implies

$$\sqrt{nh} \left[\hat{\boldsymbol{\gamma}}_1 - \boldsymbol{\gamma}_{01} - \frac{h^2}{2} \mu_2 \boldsymbol{\gamma}''_{01}(t) \right] \xrightarrow{D} N(0, \boldsymbol{I}^{-1}(t) \int_{-1}^{1} K^2(s) ds),$$

which completes the proof of Theorem 4.1.

C.2.2 Proof of Theorem 3.4.2

To prove the asymptotic behavior of the estimator near the boundary points, we need some modifications on the proof of the asymptotic behavior of the estimator in the interior points (Theorem 4.1). These modifications are similar to those for the ordinary regression setting [87, 84]. We first need to modify the Lemma 1 for the left boundary point $t = ch$, where $c \in (0, 1]$:

Theorem C.2.2. *2 Let W_n be $n^{-1} \sum_{i=1}^{n} Y_i(t) g(t_i, (t_i - t_0)/h, Z_i(t)) K_h(t_i - t_0)$, where $g(\cdot, \cdot, \cdot)$ is a continuous function and $E(g(T, u, Z(t))|T = 0)$ is right continuous at the point 0. If $h \to 0$ such that $nh/\log(n) \to \infty$, then*

$$\sup_{0 \le t \le \tau} |W_n(t) - W(t)| \xrightarrow{P} 0, \tag{C.27}$$

where $W(t) = \int_{-c}^{1} Y(t)E(g(0,u,Z(t))|t=0)K(u)duf(0)$ *and* $f(\cdot)$ *is the density function of* T.

Using Lemma 2 and following the procedure to derive (C.23) in Theorem 4.1, we have

$$\sqrt{nh}\left[\hat{\boldsymbol{\gamma}}_1 - \boldsymbol{\gamma}_{01} - \frac{h^2}{2}\boldsymbol{\gamma}_{01}{}''(0+)\int_{-c}^{1} s^2K(s)ds\right] = \mathbf{I}^{-1}(0+)(nh)^{-1/2}J^{\star}_{n,1}(\tau) + o_p(1), \quad \text{(C.28)}$$

where $J^{\star}_{n,1}(\tau) = \int_0^{\tau} K_h(u-0)n^{-1}\sum_{i=1}^{n}\left[\mathbf{X}_i(u) - \frac{\tilde{S}_{n,1}(u)}{S_{n,0}(\mathbf{0},u)}\right]dM_i(u)$. Then, by applying lemma 2 to the process $\tilde{J}_n(u) \equiv (nh)^{1/2}J^{\star}_{n,1}(u)$ in (C.28), we derive

$$\langle \tilde{J}_n(u), \tilde{J}_n(u)\rangle(\tau) = \int_{-c}^{1} K^2(s)ds\mathbf{I}(0+) + o_p(1). \quad \text{(C.29)}$$

With (C.29), by the martingale central limit theorem, we establish

$$(nh)^{1/2}J^{\star}_{n,1}(u) \xrightarrow{D} N(0, \int_{-c}^{1} K^2(s)ds\mathbf{I}(t)), \quad \text{(C.30)}$$

in distribution . Therefore, (C.30) with (C.28) implies

$$\sqrt{nh}\left[\hat{\boldsymbol{\gamma}}_1 - \boldsymbol{\gamma}_{01} - \frac{h^2}{2}\boldsymbol{\gamma}''_{01}(0+)\int_{-c}^{1} s^2K(s)ds\right] \xrightarrow{D} N(0, \boldsymbol{I}^{-1}(0+)\int_{-c}^{1} K^2(s)ds),$$

which completes the proof of Theorem 4.2.

C.2.3 Proof of Theorem 3.4.3

We first consider the case when $\boldsymbol{\beta}(\cdot)$ is known. Then, $Q_2(\boldsymbol{\Sigma}_a)$ is a likelihood function of a multivariate normal likelihood with unknown covariance matrix $\boldsymbol{\Sigma}_a$, and $\hat{\boldsymbol{\Sigma}}_a$ is a maximum likelihood estimator from the likelihood; thus, $\hat{\boldsymbol{\Sigma}}_a$ in this case possesses the consistency and asymptotically normality as maximum likelihood estimators. To express the details conveniently, we rearrange $\hat{\boldsymbol{\Sigma}}_a$ as a column vector $vec(\hat{\boldsymbol{\Sigma}}_a)$ and $\boldsymbol{\Sigma}_a$ as $vec(\boldsymbol{\Sigma}_a)$. Then,

through the law of large number, we have the consistency property of $\hat{\mathbf{\Sigma}}$; that is,

$$vec(\hat{\mathbf{\Sigma}}) \xrightarrow{P} vec(\mathbf{\Sigma}) \tag{C.31}$$

as $n \to \infty$. To prove the aymptotical normality of $\hat{\mathbf{\Sigma}}$, note that for $i = 1, \cdots, n$ $E[\mathbf{a}_i \mathbf{a}^T]$ conditioned on a censored event or a triggered event may follow a different distribution, so $E[\mathbf{a}_i \mathbf{a}^T]$ are independent variables but not following an identical distribution. In this case, under condition A.7, we can apply the multivariate Lindeberg central limit theorem [125] and establish

$$\sqrt{n}\{vec(\hat{\mathbf{\Sigma}}_a) - vec(\mathbf{\Sigma}_a)\} \xrightarrow{D} N(0, U(\mathbf{\Sigma}_a)), \tag{C.32}$$

where $U(\mathbf{\Sigma}_a) = E(aa^T \otimes aa^T) - (vec(\mathbf{\Sigma}_a))(vec\mathbf{\Sigma}_a)^T$. Now, we consider the case of unknown $\boldsymbol{\beta}(t)$. The estimator of $\mathbf{\Sigma}_a$ is denoted by $\hat{\mathbf{\Sigma}}_a(\boldsymbol{\beta})$ when $\boldsymbol{\beta}(\cdot)$ is known and denoted by $\hat{\mathbf{\Sigma}}_a(\hat{\boldsymbol{\beta}})$ when $\boldsymbol{\beta}$ is estimated by $\hat{\boldsymbol{\beta}}$. Based on Theorem 4.1, $\hat{\boldsymbol{\beta}}$ has a bias of order h^2, and therefore we have $\hat{\mathbf{\Sigma}}(\hat{\boldsymbol{\beta}}) - \hat{\mathbf{\Sigma}}(\boldsymbol{\beta}) = o_p(h^2)$. This implies $\sqrt{n}\{vec(\hat{\mathbf{\Sigma}}_a(\hat{\boldsymbol{\beta}})) - vec(\mathbf{\Sigma}_a)\} = \sqrt{n}\{vec(\hat{\mathbf{\Sigma}}_a(\boldsymbol{\beta})) - vec(\mathbf{\Sigma}_a) + o_p(h^2)\} = \sqrt{n}\{vec(\hat{\mathbf{\Sigma}}_a(\boldsymbol{\beta})) - vec(\mathbf{\Sigma}_a)\} + o_p(\sqrt{n}h^2)$. Under the condition $nh^4 \to 0, o_p(\sqrt{n}h^2)$ goes to 0. Thus, when $\boldsymbol{\beta}(t)$ is estimated, the asymptotic normality (C.32) remains valid. The asymptotic normality implies $vec(\hat{\mathbf{\Sigma}}_a(\hat{\boldsymbol{\beta}})) - vec(\mathbf{\Sigma}_a) = O_p(1/\sqrt{n})$, which goes to 0 as $n \to \infty$. Therefore, the consistency of $\hat{\mathbf{\Sigma}}(\hat{\boldsymbol{\beta}})$ is established.

C.3 Details for Computing the E-Step of the Extended EM Algorithm

This section provides some computation details of the E-step of the proposed algorithm in the main paper. In general, the computation of the conditional expectation can be expressed as $E[h(\mathbf{a}_i)] = \int h(\mathbf{a}_i) p(a_i|\mathbf{D}, \hat{\boldsymbol{\theta}})$, where $p(a_i|\mathbf{D}, \hat{\boldsymbol{\theta}})$ is the conditional density of a_i given data $\mathbf{D}$ and $\hat{\boldsymbol{\theta}} = (\hat{\lambda}_0, \hat{\boldsymbol{\beta}}, \hat{\mathbf{\Sigma}}_a)$. For example, $h(\mathbf{a}_i) = \exp(\mathbf{Z}_i^T \mathbf{a}_i)$ in (9) of the main paper. To derive these conditional expectations for our application, we may use numerical integration for the formula given below to derive these conditional expectations. This formula can be

considered in two cases: $d_i = 1$ and $d_i = 0$, for $i = 1, \cdots, n$. For clarity, we denote $p(a_i|\mathbf{D}, \hat{\boldsymbol{\theta}})$ as $p(a_i|d_i)$. When the observed data is censored ($d_i = 0$),

$$p(a_i|d_i = 0) = \frac{P(a_i, T_i > Y_i)}{P(T_i > Y_i)} = \frac{P(T_i > Y_i|a_i)P(a_i)}{P(T_i > Y_i)} = \frac{S(t_i|a_i)P(a_i)}{\int S(Y_i|a_i)P(a_i)da_i},$$

where $S(Y_i|a_i)$ is $\exp(-\int_0^{Y_i} \lambda(t|a_i)dt)$, and $\lambda(t|a_i)$ is $\lambda(t)$ from equation (3) in the main paper with giving X_i and a_i. When the observed data is a triggered time ($d_i = 1$),

$$\begin{aligned} p(a_i|d_i = 1) &= \frac{P(a_i, T_i = Y_i)}{P(T_i = Y_i)} = \frac{P(T_i = Y_i|a_i)P(a_i)}{\int P(T_i = Y_i|a_i)P(a_i)da_i} \\ &= \frac{f(t_i|a_i)P(a_i)}{\int f(t_i|a_i)P(a_i)da_i} = \frac{S(t_i|a_i)\exp(a_i)P(a_i)}{\int S(t_i|a_i)\exp(a_i)P(a_i)da_i} \end{aligned}$$

The extension of the two formulas to higher dimensions is straightforward.

C.4 More Comparisons Between the Proposed Method with a Spline Based Method

In this section, we provide more simulations for comparing the estimation accuracy of the local linear varying coefficient frailty method and that of spline-based varying coefficient frailty methods using the cubic spline basis and the nature cubic spline basis. The way to simulate data is the same setting described in Section 5.1 of the paper. After deriving the data, we implement the spline-based method with three selected knots: (a) (0, 0.5, 1, 1.5, 2), (b) (0, 0.5, 0.75, 1, 1.25, 1.5, 2), and (c) (0, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 1.9, 2), and the local linear based method with bandwidth 0.5. The resulting plots are in Figure C.1. From the figure, we observed that the local linear estimator (blue solid curve) outperforms the spline-based estimators using the cubic spline (red dashed line) and the nature cubic splines (brown long-dashed line estimators, especially near the boundary, as we conclude in Section 5.1 of the main paper.

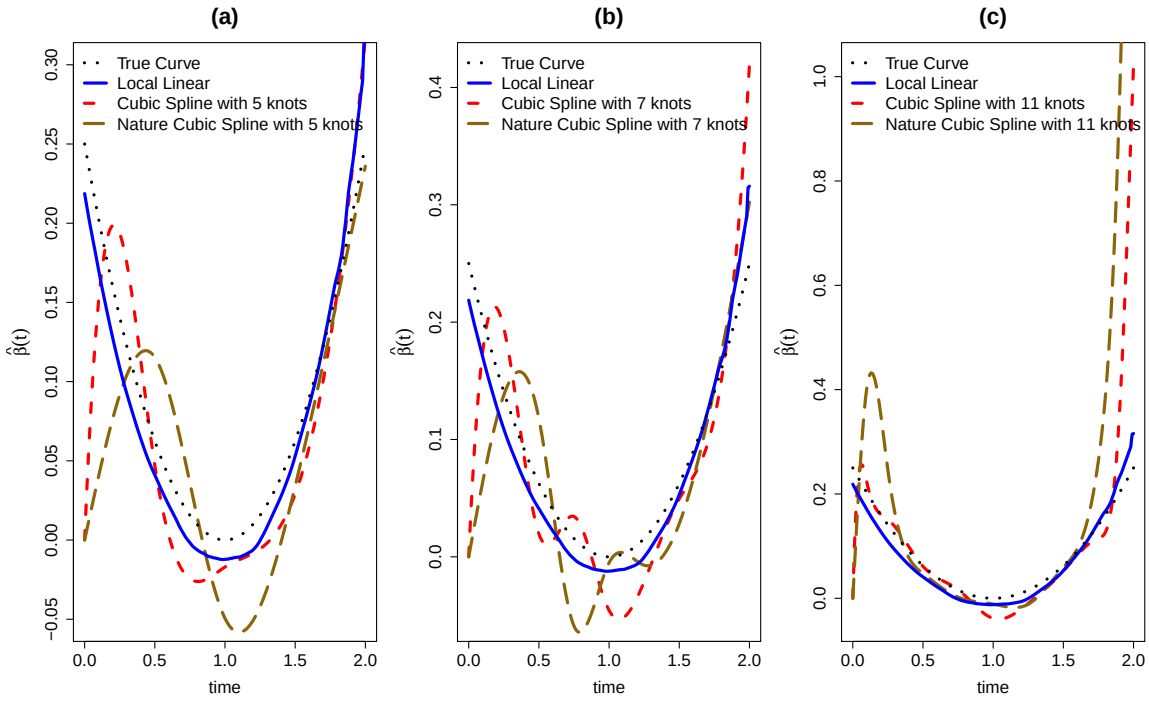

Figure C.1: More comparisons of the varying coefficient estimator from the proposed local linear method (blue solid line) with bandwidth 0.5 and the Spline based method (red dashed line) with knots at (a) (0, 0.5, 1, 1.5, 2), (b) (0, 0.5, 0.75, 1, 1.25, 1.5, 2), and (c) (0, 0.1, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 1.9, 2). The true curve is the black dotted line.

APPENDIX D

SUPPLEMENTAL MATERIAL FOR CHAPTER 4

The appendix provides the detailed proof of the lemma and theorems given in Section 4.3.

D.1 Proofs of the Main Theorems

D.1.1 Proof of Lemma 4.3.1

By the definition of $\widehat{Err}_k$ in (4.3) and $e\bar{r}r_k$ in (4.7), we have

$$E[\widehat{Err}_k - e\bar{r}r_k] = \frac{1}{n}\sum_{i=1}^{n} E\left[Q\left\{\hat{\xi}(\mathbf{x}_i), \hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}_{(-i)}))\right\} - Q\left\{y_i, \hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}_{(-i)}))\right\}\right], \tag{D.1}$$

and from (4.4), we have $Q(\hat{\xi}(\mathbf{x}_i), \hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}))) = q(\hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}))) + \dot{q}(\hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D})))(\hat{\xi}(\mathbf{x}_i) - \hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D})))$. Based on (3.11) of Theorem 1 in [114], it implies

$$Q\left\{\hat{\xi}(\mathbf{x}_i), \hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}_{(-i)}))\right\} - Q\left\{y_i, \hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}_{(-i)}))\right\} = \dot{q}(\hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}_{(-i)})))(y_i - \hat{\xi}(\mathbf{x}_i)). \tag{D.2}$$

Combining (D.1) and (D.2), we have

$$E[\widehat{Err}_k - e\bar{r}r_k] \;=\; \frac{1}{n}\sum_{i=1}^{n} E\left[\dot{q}(\hat{f}_k(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_k(\mathbf{D}_{(-i)})))(y_i - \hat{\xi}(\mathbf{x}_i))\right],$$

which is the generalized degree of freedom GD_k in (4.8).

D.1.2 Proof of Theorem 4.4.2

For any consistent estimator $\hat{\boldsymbol{\theta}}_T$ (Theorem 1 of [111] and Theorem 3.1 of [121]), we have

$$f_0(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_T(\mathbf{D}_{(-i)})) - f_0(\mathbf{x}_i; \boldsymbol{\theta}) \to 0 \tag{D.3}$$

in probability. For any consistent estimator $\hat{f}_T$ (Theorem 2.1 in [126] and Corollary 1.3 in [127]), it follows that

$$\hat{f}_T(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_T(\mathbf{D}_{(-i)})) - f_0(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_T(\mathbf{D}_{(-i)})) \to 0 \tag{D.4}$$

in probability. Under (D.3) and (D.4),

$$\begin{aligned} \hat{f}_T(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_T(\mathbf{D}_{(-i)})) - f_0(\mathbf{x}_i; \boldsymbol{\theta}) &= \hat{f}_T(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_T(\mathbf{D}_{(-i)})) - f_0(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_T(\mathbf{D}_{(-i)})) + \\ &\quad f_0(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_T(\mathbf{D}_{(-i)})) - f_0(\mathbf{x}_i; \boldsymbol{\theta}) \\ &\to 0 \end{aligned}$$

in probability. This implies

$$\widehat{Err}_T - Err_0 = \frac{1}{n}\sum_{i=1}^{n}\left[Q\left\{\hat{\xi}(\mathbf{x}_i), \hat{f}_T(\mathbf{x}_i; \hat{\boldsymbol{\theta}}_T)\right\} - Q\left\{\hat{\xi}(\mathbf{x}_i), f_0(\mathbf{x}_i; \boldsymbol{\theta})\right\}\right] \to 0 \tag{D.5}$$

in probability. This proves part (i) of Theorem 3.2. From (D.5), we have $\widehat{GD}_T - GD_0 = \widehat{Err}_T - Err_0 \to 0$ in probability, which proves part (ii) of Theorem 3.2.

D.1.3 Proof of Theorems 4.4.3

Suppose $\mathbf{x}_1, \cdots, \mathbf{x}_n$ are realizations of a random variable $\mathbf{X}$ whose probability density function is denoted by $p_{\mathbf{X}}(\mathbf{x})$. Then, we have

$$\frac{1}{n}\sum_{i=1}^{n} Q\left\{\hat{\xi}(\mathbf{x}_i), \hat{f}_1(\mathbf{x}_i; \boldsymbol{\theta})\right\} \to E\left[Q\left\{\hat{\xi}(\mathbf{X}), \hat{f}_1(\mathbf{X}; \boldsymbol{\theta})\right\})\right] = \int Q\left\{\hat{\xi}(\mathbf{x}), \hat{f}_1(\mathbf{x}; \boldsymbol{\theta})\right\} p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}, \tag{D.6}$$

in probability by the law of large numbers. Following (D.3), we have

$$\frac{1}{n}\sum_{i=1}^{n} Q\left\{\hat{\xi}(\mathbf{x}_i), \hat{f}_1(\mathbf{x}_i; \boldsymbol{\theta}(\mathbf{D}_{(-i)}))\right\} - \frac{1}{n}\sum_{i=1}^{n} Q\left\{\hat{\xi}(\mathbf{x}_i), \hat{f}_1(\mathbf{x}_i; \boldsymbol{\theta})\right\} \to 0 \tag{D.7}$$

in probability. From (D.6) and (D.7),

$$\frac{1}{n}\sum_{i=1}^{n} Q\left\{\hat{\xi}(\mathbf{x}_i), \hat{f}_1(\mathbf{x}_i; \boldsymbol{\theta}(\mathbf{D}_{(-i)}))\right\} \to \int Q\left\{\hat{\xi}(\mathbf{x}), \hat{f}_1(\mathbf{x}; \boldsymbol{\theta})\right\} p_X(\mathbf{x}) d\mathbf{x} \tag{D.8}$$

in probability. When $p_X(\mathbf{x})$ follows a p-dimensional uniform distribution on $[0,1]^p$ and Q is chosen as square loss $Q(h_1, h_2) = (h_2 - h_1)^2$,

$$\arg\min_{\boldsymbol{\theta}} \int_{\mathbf{X}} Q\left\{\hat{\xi}(\mathbf{x}), \hat{f}_1(\mathbf{x}; \boldsymbol{\theta})\right\} p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} = ||\hat{\xi}(\mathbf{x}) - \hat{f}_1(\mathbf{x}; \boldsymbol{\theta}_1)||_{L_2}, \tag{D.9}$$

which is the criterion used for the L_2-norm calibration [111]. Combining (D.8) and (D.9), we have $\widehat{Err}_1 \to ||\hat{\xi}(\mathbf{x}) - \hat{f}_1(\mathbf{x}; \boldsymbol{\theta}_1)||_{L_2}$ in probability.

REFERENCES

[1] G. E. Fasshauer, *Meshfree Approximation Methods with MATLAB,* New Jersey: World Scientific, 2007.

[2] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, "Design and Analysis of Computer Experiments," *Statistical Science*, vol. 4, pp. 409–423, 1989.

[3] T. J. Santner, B. J. Williams, and W. I. Notz, *The Design and Analysis of Computer Experiments (2nd Edition),* New York: Springer, 2018.

[4] N. Cressie, *Statistics for Spatial Data,* New York: Wiley, 1993.

[5] V. De Oliveira, B. Kedem, and D. A. Short, "Bayesian Prediction of Transformed gaussian Random Fields," *Journal of the American Statistical Association*, vol. 92, pp. 1422–1433, 1997.

[6] E. Snelson, Z. Ghahramani, and C. E. Rasmussen, "Warped Gaussian Processes," *Advances in Neural Information Processing Systems*, pp. 337–344, 2004.

[7] M. Lázaro-Gredilla, "Bayesian Warped Gaussian Processes," *Advances in Neural Information Processing Systems*, pp. 1619–1627, 2012.

[8] A. A. Sonin, *The Physical Basis of Dimensional Analysis,* Massachusetts: Department of Mechanical Engineering, MIT, 2001.

[9] T. J. Hastie and R. J. Tibshirani, *Generalized Additive Models,* New York: CRC press, 1990.

[10] J. H. Friedman and W. Stuetzle, "Projection Pursuit Regression," *Journal of the American Statistical Association*, vol. 76, pp. 817–823, 1981.

[11] D. K. Duvenaud, H. Nickisch, and C. E. Rasmussen, "Additive Gaussian Processes," *Advances in Neural Information Processing Systems*, pp. 226–234, 2011.

[12] R. Tibshirani, "Estimating Transformations for Regression via Additivity and Variance Stabilization," *Journal of the American Statistical Association*, vol. 83, pp. 394–405, 1988.

[13] L. Breiman and J. H. Friedman, "Estimating Optimal Transformations for Multiple Regression and Correlation," *Journal of the American statistical Association*, vol. 80, pp. 580–598, 1985.

[14] G. E. Box and D. R. Cox, "An Analysis of Transformations," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 26, pp. 211–252, 1964.

[15] I.-K. Yeo and R. A. Johnson, "A New Family of Power Transformations to Improve Normality or Symmetry," *Biometrika*, vol. 87, pp. 954–959, 2000.

[16] C. F. J. Wu and M. S. Hamada, *Experiments: Planning, Analysis, and Optimization (2nd Edition),* New York: Wiley, 2009.

[17] S. N. Wood, *Generalized Additive Models: An Introduction with R,* New York: CRC press, 2017.

[18] O. Roustant, D. Ginsbourger, and Y. Deville, "DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization," *Journal of Statistical Software*, vol. 51, pp. 1–55, 2012.

[19] L.-H. Lin and V. R. Joseph, "tag: transformed additive guassian process", R package version 0.1.0, 2019.

[20] T. A. Plate, "Accuracy versus Interpretability in Flexible Modeling: Implementing a Tradeoff Using Gaussian Process Models," *Behaviormetrika*, vol. 26, pp. 29–50, 1999.

[21] O. Harari and D. M. Steinberg, "Convex Combination of Gaussian Processes for Bayesian Analysis of Deterministic Computer Experiments," *Technometrics*, vol. 56, pp. 443–454, 2014.

[22] S. Ba and V. R. Joseph, "Composite Gaussian Process Models for Emulating Expensive Functions," *The Annals of Applied Statistics*, vol. 6, pp. 1838–1860, 2012.

[23] D. Christophe and S. Petr, "randtoolbox: generating and testing random numbers", R package version 1.17.1, 2018.

[24] G. M. Dancik and K. S. Dorman, "mlegp: Statistical Analysis for Computer Models of Biological Systems Using R," *Bioinformatics*, vol. 24, pp. 1966–1967, 2008.

[25] C. K. Williams and C. E. Rasmussen, "Gaussian Processes for Regression," *Advances in Neural Information Processing Systems*, pp. 226–234, 1996.

[26] C. Linkletter, D. Bingham, N. Hengartner, D. Higdon, and K. Q. Ye, "Variable Selection for Gaussian Process Models in Computer Experiments," *Technometrics*, vol. 48, pp. 478–490, 2006.

[27] T. Savitsky, M. Vannucci, and N. Sha, "Variable Selection for Nonparametric Gaussian Process Priors: Models and Computational Strategies," *Statistical Science: A Review Journal of the Institute of Mathematical Statistics*, vol. 26, p. 130, 2011.

[28] G Yi, J. Q. Shi, and T Choi, "Penalized Gaussian Process Regression and Classification for High-Dimensional Nonlinear Data," *Biometrics*, vol. 67, pp. 1285–1294, 2011.

[29] B. J. Reich, E. Kalendra, C. B. Storlie, H. D. Bondell, and M. Fuentes, "Variable Selection for High Dimensional Bayesian Density Estimation: Application to Human Exposure Simulation," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 61, pp. 47–66, 2012.

[30] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning,* Massachusetts: The MIT Press, 2006.

[31] V. R. Joseph, Y. Hung, and A. Sudjianto, "Blind Kriging: A New Method for Developing Metamodels," *Journal of Mechanical Design*, vol. 130, p. 031 102, 2008.

[32] Y. Hung, "Penalized Blind Kriging in Computer Experiments," *Statistica Sinica*, vol. 21, pp. 1171–1190, 2011.

[33] T. Gneiting and A. E. Raftery, "Strictly Proper Scoring Rules, Prediction, and Estimation," *Journal of the American Statistical Association*, vol. 102, pp. 359–378, 2007.

[34] I. M. Sobol', "On Sensitivity Estimation for Nonlinear Mathematical Models," *Matematicheskoe Modelirovanie*, vol. 2, pp. 112–118, 1990.

[35] M. D. Morris, T. J. Mitchell, and D. Ylvisaker, "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction," *Technometrics*, vol. 35, pp. 243–255, 1993.

[36] V. R. Joseph, E. Gul, and S. Ba, "Maximum Projection Designs for Computer Experiments," *Biometrika*, vol. 102, pp. 371–380, 2015.

[37] B. J. Reich, C. B. Storlie, and H. D. Bondell, "Variable Selection in Bayesian Smoothing Spline ANOVA Models: Application to Deterministic Computer Codes," *Technometrics*, vol. 51, pp. 110–120, 2009.

[38] M. Plumlee and V. R. Joseph, "Orthogonal Gaussian Process Models," *Statistica Sinica*, vol. 28, pp. 601–619, 2018.

[39] B. MacDonald, P. Ranjan, and H. Chipman, "GPfit: An R Package for Fitting a Gaussian Process Model to Deterministic Simulator Outputs," *Journal of Statistical Software*, vol. 64, pp. 1–23, 2015.

[40] B. Haaland and P. Z. Qian, "Accurate Emulators for Large-Scale Computer Experiments," *The Annals of Statistics*, vol. 39, pp. 2974–3002, 2011.

[41] R. B. Gramacy and D. W. Apley, "Local Gaussian Process Approximation for Large Computer Experiments," *Journal of Computational and Graphical Statistics*, vol. 24, pp. 561–578, 2015.

[42] S. Surjanovic and D. Bingham, *Virtual Library of Simulation Experiments: Test Functions and Datasets,* from http://www.sfu.ca/ ssurjano, Retrieved June 3, 2019, 2019.

[43] Z. Qian, C. C. Seepersad, V. R. Joseph, J. K. Allen, and C. J. Wu, "Building Surrogate Models Based on Detailed and Approximate Simulations," *Journal of Mechanical Design*, vol. 128, pp. 668–677, 2006.

[44] L.-H. Lin and R. V. Joseph, "Transformation and additivity in gaussian processes," *Technometrics*, vol. to appear, DOI, pp. 10.1080/00401706.2019.1665592, 2019.

[45] R. B. Gramacy, "laGP: Large-Scale Spatial Modeling via Local Approximate Gaussian Processes in R," *Journal of Statistical Software*, vol. 72, pp. 1–46, 2016.

[46] N. Cressie and G. Johannesson, "Fixed rank kriging for very large spatial data sets," *Journal of the Royal Statistical Society: Series B*, vol. 70, pp. 209–226, 2008.

[47] A. J. Smola and P. L. Bartlett, "Sparse greedy gaussian process regression," in *Advances in neural information processing systems*, 2001, pp. 619–625.

[48] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," in *Advances in neural information processing systems*, 2006, pp. 1257–1264.

[49] S. Mak and R. Joseph, "Support points," *The Annals of Statistics*, vol. 46, pp. 2562–2592, 2018.

[50] S. Wood, *Generalized Additive Models: An Introduction with R (2nd Edition)*. New York: CRC press, 2017.

[51] J. Duchon, "Splines minimizing rotation-invariant semi-norms in sobolev spaces," in *Constructive theory of functions of several variables*, Springer, 1977, pp. 85–100.

[52] M. Plumlee and D. W. Apley, "Lifted brownian kriging models," *Technometrics*, vol. 59, no. 2, pp. 165–177, 2017.

[53] C.-L. Sung, W. Wang, M. Plumlee, and B. Haaland, "Multiresolution functional anova for large-scale, many-input computer experiments," *Journal of the American Statistical Association*, pp. 1–23, 2019.

[54] C.-L. Sung, *Mrfa: Fitting and predicting large-scale nonlinear regression problems using multi-resolution functional anova (mrfa) approach*, R package version 0.4.0, 2019, pp. 10.1080/01621459.2019.1595630.

[55] R Core Team, *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, 2020.

[56] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When gaussian process meets big data: A review of scalable gps," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[57] V. R. Joseph, "Space-filling designs for computer experiments: A review," *Quality Engineering*, vol. 28, no. 1, pp. 28–35, 2016.

[58] M. Chang, L.-H. Lin, J. Romberg, and A. Raychowdhury, "Optimo: A 65-nm 279-gops/w 16-b programmable spatial-array processor with on-chip network for solving distributed optimizations via the alternating direction method of multipliers," *IEEE Journal of Solid-State Circuits*, 2019.

[59] D. Christophe and S. Petr, *Randtoolbox: Generating and testing random numbers*, R package version 1.30.0, 2019.

[60] R. B. Gramacy and H. K. Lee, "Cases for the nugget in modeling computer experiments," *Statistics and Computing*, vol. 22, no. 3, pp. 713–722, 2012.

[61] A. Asuncion and D. Newman, *University of California Irvine (UCI) Machine Learning Repository*. https://archive.ics.uci.edu/ml/index.php, 2020.

[62] P. Tüfekci, "Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods," *International Journal of Electrical Power & Energy Systems*, vol. 60, pp. 126–140, 2014.

[63] T. F. Brooks, D. S. Pope, and M. A. Marcolini, "Airfoil self-noise and prediction," 1989.

[64] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[65] J. E. Smith-Garvin, G. A. Koretzky, and M. S. Jordan, "T cell activation," *Annual Review of Immunology*, vol. 27, pp. 591–619, 2009.

[66] B. T. Marshall, M. Long, J. W. Piper, T. Yago, R. P. McEver, and C. Zhu, "Direct observation of catch bonds involving cell-adhesion molecules," *Nature*, vol. 423, pp. 190–193, 2003.

[67] B. Liu, W. Chen, B. D. Evavold, and C. Zhu, "Accumulation of dynamic catch bonds between tcr and agonist peptide-mhc triggers t cell signaling," *Cell*, vol. 157, pp. 357–368, 2014.

[68] J. P. Klein, "Semiparametric estimation of random effects using the cox model based on the em algorithm," *Biometrics*, vol. 48, pp. 795–806, 1992.

[69] S. A. Murphy, "Consistency in a proportional hazards model incorporating a random effect," *The Annals of Statistics*, vol. 22, pp. 712–731, 1994.

[70] X. Xue and R. Brookmeyer, "Bivariate frailty model for the analysis of multivariate survival time," *Lifetime Data Analysis*, vol. 2, pp. 277–289, 1996.

[71] D. J. Sargent, "A general framework for random effects survival analysis in the cox proportional hazards setting," *Biometrics*, vol. 54, pp. 1486–1497, 1998.

[72] F. Vaida and R. Xu, "Proportional hazards model with random effects," *Statistics in Medicine*, vol. 19, pp. 3309–3324, 2000.

[73] R. Ma, D. Krewski, and R. T. Burnett, "Random effects Cox models: A Poisson modelling approach," *Biometrika*, vol. 90, pp. 157–169, 2003.

[74] D. Zeng, D. Lin, and X. Lin, "Semiparametric transformation models with random effects for clustered failure time data," *Statistica Sinica*, vol. 18, pp. 355–377, 2008.

[75] A. Gamst, M. Donohue, and R. Xu, "Asymptotic properties and empirical evaluation of the npmle in the proportional hazards mixed-effects model," *Statistica Sinica*, vol. 19, pp. 997–1011, 2009.

[76] Y. Mazroui, A. Mauguen, S. Mathoulin-Pélissier, G. MacGrogan, V. Brouste, and V. Rondeau, "Time-varying coefficients in a multivariate frailty model: Application to breast cancer recurrences of several types and death," *Lifetime Data Analysis*, vol. 22, pp. 191–215, 2016.

[77] Z. Zhang, L. Song, X. Wang, and M. Amin, "Estimation of multivariate frailty models with varying coefficients," *Communications in Statistics-Theory and Methods*, pp. 1–12, 2018.

[78] Z. Yu, L. Liu, D. M. Bravata, L. S. Williams, and R. S. Tepper, "A semiparametric recurrent events model with time-varying coefficients," *Statistics in Medicine*, vol. 32, pp. 1016–1026, 2013.

[79] Z. Yu, L. Liu, D. M. Bravata, and L. S. Williams, "Joint model of recurrent events and a terminal event with time-varying coefficients," *Biometrical Journal*, vol. 56, pp. 183–197, 2014.

[80] Y. Li and D. Ruppert, "On the asymptotics of penalized splines," *Biometrika*, vol. 95, no. 2, pp. 415–436, 2008.

[81] Z. Cai and Y. Sun, "Local linear estimation for time-dependent coefficients in cox's regression models," *Scandinavian Journal of Statistics*, vol. 30, pp. 93–111, 2003.

[82] L. Tian, D. Zucker, and L. Wei, "On the cox model with time-varying regression coefficients," *Journal of the American Statistical Association*, vol. 100, pp. 172–183, 2005.

[83] J. Fan, H. Lin, and Y. Zhou, "Local partial-likelihood estimation for lifetime data," *The Annals of Statistics*, vol. 34, pp. 290–325, 2006.

[84] J. Fan and I. Gijbels, *Local Polynomial Modelling and Its Applications,* Chapman and Hall, 1996.

[85] L. D. Fisher and D. Y. Lin, "Time-dependent covariates in the cox proportional-hazards regression model," *Annual Review of Public Health*, vol. 20, pp. 145–157, 1999.

[86] W. Chen, J. Lou, and C. Zhu, "Forcing switch from short-to intermediate-and long-lived states of the αA domain generates LFA-1/ICAM-1 catch bonds," *Journal of Biological Chemistry*, vol. 285, pp. 35 967–35 978, 2010.

[87] J. Fan and I. Gijbels, "Variable bandwidth and local linear regression smoothers," *The Annals of Statistics*, pp. 2008–2036, 1992.

[88] J. Cai, J. Fan, J. Jiang, and H. Zhou, "Partially linear hazard regression with varying coefficients for multivariate survival data," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, pp. 141–158, 2008.

[89] D. R. Cox, "Regression models and life-tables (with discussions)," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 34, pp. 187–220, 1972.

[90] M. P. Wand and M. C. Jones, *Kernel Smoothing,* Chapman and Hall/CRC, 1994.

[91] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B*, vol. 39, pp. 1–38, 1977.

[92] C. F. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of Statistics*, pp. 95–103, 1983.

[93] S. Johansen, "An extension of Cox's regression model," *International Statistical Review*, vol. 51, pp. 165–174, 1983.

[94] R. Tibshirani and T. Hastie, "Local likelihood estimation," *Journal of the American Statistical Association*, vol. 82, pp. 559–567, 1987.

[95] A. C. Davison and N. I. Ramesh, "Local likelihood smoothing of sample extremes," *Journal of the Royal Statistical Society: Series B*, vol. 62, pp. 191–208, 2000.

[96] R. J. Gray, "Flexible methods for analyzing survival data using splines, with applications to breast cancer prognosis," *Journal of the American Statistical Association*, vol. 87, pp. 942–951, 1992.

[97] D. R. Cox and E. J. Snell, "A general definition of residuals," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 30, pp. 248–265, 1968.

[98] N. Breslow, "Covariance analysis of censored survival data," *Biometrics*, vol. 30, pp. 89–99, 1974.

[99] B. Efron, "The efficiency of Cox's likelihood function for censored data," *Journal of the American Statistical Association*, vol. 72, pp. 557–565, 1977.

[100] J. D. Kalbfleisch and R. L. Prentice, *The Statistical Analysis of Failure Time Data*. John Wiley & Sons, 2011, vol. 360.

[101] L.-H. Lin and L.-S. Huang, "Connections between cure rates and survival probabilities in proportional hazards models," *Stat*, vol. 8, e255, 2019.

[102] T. J. Santner, B. J. Williams, W. Notz, and B. J. Williams, *The Design and Analysis of Computer Experiments, Second Edition*. New York: Springer, 2018.

[103] S. Fomundam and J. W. Herrmann, "A survey of queuing theory applications in healthcare," *Institute for Systems Research Technical Reports*, vol. School of Engineering, University of Maryland, 2007.

[104] C Lakshmi and S. A. Iyer, "Application of queueing theory in health care: A literature review," *Operations Research for Health Care*, vol. 2, pp. 25–39, 2013.

[105] L. F. Richardson, *Weather Prediction by Numerical Process.* Cambridge: Cambridge University Press, 2007.

[106] F. Brauer and C. Castillo-Chavez, *Mathematical Models in Population Biology and Epidemiology.* New York: Springer, 2012.

[107] L. S. Bastos and A. O'Hagan, "Diagnostics for gaussian process emulators," *Technometrics*, vol. 51, pp. 425–438, 2009.

[108] A. M. Overstall and D. C. Woods, "Multivariate emulation of computer simulators: Model selection and diagnostics with application to a humanitarian relief model," *Journal of the Royal Statistical Society: Series C*, vol. 65, pp. 483–505, 2016.

[109] M. C. Kennedy and A. O'Hagan, "Predicting the output from a complex computer code when fast approximations are available," *Biometrika*, vol. 87, pp. 1–13, 2000.

[110] R. Tuo, C. F. J. Wu, and D. Yu, "Surrogate modeling of computer experiments with different mesh densities," *Technometrics*, vol. 56, pp. 372–380, 2014.

[111] R. Tuo and C. F. J. Wu, "Efficient calibration for imperfect computer models," *Annals of Statistics*, vol. 43, pp. 2331–2352, 2015.

[112] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge: Cambridge University Press, 2004.

[113] G. Wahba, *Spline Models for Observational Data.* Philadelphia: SIAM, 1990.

[114] B. Efron, "How biased is the apparent error rate of a prediction rule?" *Journal of the American Statistical Association*, vol. 81, pp. 461–470, 1986.

[115] J. Ye, "On measuring and correcting the effects of data mining and model selection," *Journal of the American Statistical Association*, vol. 93, pp. 120–131, 1998.

[116] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B*, vol. 63, pp. 425–464, 2001.

[117] D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne, "Combining field data and computer simulations for calibration and prediction," *SIAM Journal on Scientific Computing*, vol. 26, pp. 448–466, 2004.

[118] J. Goh, D. Bingham, J. P. Holloway, M. J. Grosskopf, C. C. Kuranz, and E. Rutter, "Prediction and computer model calibration using outputs from multifidelity simulators," *Technometrics*, vol. 55, pp. 501–512, 2013.

[119] M. E. Johnson, L. M. Moore, and D. Ylvisaker, "Minimax and maximin distance designs," *Journal of Statistical Planning and Inference*, vol. 26, pp. 131–148, 1990.

[120] W. R. Rittase, *Combined Experimental and Modeling Studies Reveal New Mechanisms in T Cell Antigen Recognition (Ph.D. Thesis).* Atlanta: Georgia Institute of Technology, 2018.

[121] C.-L. Sung, Y. Hung, W. Rittase, C. Zhu, and C. F. J. Wu, "Calibration for computer experiments with binary responses and application to cell adhesion study," *Journal of the American Statistical Association*, vol. DOI: 10.1080/01621459.2019.1699419, 2020.

[122] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *Journal of Computational Physics*, vol. 22, pp. 403–434, 1976.

[123] C.-L. Sung, Y. Hung, W. Rittase, C. Zhu, and C. F. J. Wu, "A generalized gaussian process model for computer experiments with binary time series," *Journal of the American Statistical Association*, vol. DOI: 10.1080/01621459.2019.1604361, 2019.

[124] P. K. Andersen and R. D. Gill, "Cox's regression model for counting processes: A large sample study," *The Annals of Statistics*, vol. 10, pp. 1100–1120, 1982.

[125] A. W. Van der Vaart, *Asymptotic Statistics,* Cambridge University Press, 2000.

[126] S. J. Yakowitz and F Szidarovszky, "A comparison of kriging with nonparametric regression methods," *Journal of Multivariate Analysis*, vol. 16, pp. 21–53, 1985.

[127] D. D. Cox, "Multivariate smoothing spline functions," *SIAM Journal on Numerical Analysis*, vol. 21, pp. 789–813, 1984.