Measurement error in spatially varying coefficient models

by

Grace Vanderwyst

B.S., Kansas State University, 2024

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2026

Approved by:

Major Professor
Jingru Mu

Copyright

Abstract

Measurement error in explanatory variables can substantially distort regression analysis, affecting both estimation and prediction. While measurement error has been extensively studied in classical regression models, relatively little work has addressed its impact in spatially varying coefficient models.

In this paper, we investigate regression modeling with measurement error in spatial settings and develop estimation procedures for spatially varying coefficient models when covariates are contaminated by measurement error. Our approach integrates the simulation–extrapolation (SIMEX) method with spline-based spatial regression techniques, including bivariate spline triangulation and its penalized version, to correct for measurement error while accommodating spatial heterogeneity.

Through simulation studies on irregular spatial domains and a real data application, we evaluate the performance of the proposed methods in terms of coefficient estimation and prediction accuracy. The results illustrate the practical impact of measurement error on spatial regression and demonstrate the effectiveness of the proposed framework.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my sincere gratitude to my advisor, Dr. Jingru Mu, for her continued guidance and support throughout the completion of this report. I am truly honored to have had the opportunity to work with her and to learn so much from her during my graduate studies. She has greatly broadened my understanding of statistics by not only introducing me to spatially varying coefficient models and the simulation-extrapolation adjustment, but also by showing me how researchers develop and evaluate new statistical methods through theoretical work, simulation studies, and applications to real datasets.

Beyond this report, it has been a pleasure to have her as an instructor. I greatly appreciate the care she takes in explaining details in her teaching, as well as the time she devoted during office hours to answering my questions and strengthening my understanding. Now, as her teaching assistant, I have come to appreciate even more the time and care she puts into her courses each week. I am grateful for the example she has set, and I will carry those lessons with me as I begin the next chapter after graduation.

Dedication

I would like to dedicate this paper to my family: my mom, dad, brother, and grandmother. I am deeply grateful for the technical and emotional support you have provided throughout this process. Your encouragement and positivity have meant more to me than I can express, especially during the computer issues near the end of this report. Thank you for everything!

Chapter 1

Introduction

Spatially varying coefficient models were introduced in the spatial statistics literature in the 1990s and early 2000s. Since then, they have continued to be used across a wide range of fields, including environmental, economic, and medical applications (Alya et al., 2023; Li et al., 2021; Wu, 2020). However, in the presence of measurement error, the estimated coefficient surfaces in these models may exhibit attenuation bias, which means that the coefficient estimates are biased toward zero. To correct for this, a simulation-extrapolation algorithm, referred to as SIMEX, has been introduced for linear regression. However, to the best of our knowledge, SIMEX has not been widely implemented in spatially varying coefficient models. Therefore, this report investigates the application of the SIMEX adjustment to a few spatially varying coefficient models.

Following this chapter, Chapter 2 introduces two spatially varying coefficient models (SVCM): geographically weighted regression (GWR) and bivariate penalized spline triangulation (BPST). Because BPST can be fit either with or without a penalty term, it will be considered as two separate models: BST and BPST. For ease, Chapter 2 will start with the non-spatial linear regression model and following this, introduce spatial variation. Chapter 3 discusses measurement error in terms of variability and how the SIMEX adjustment helps correct for it to produce less biased coefficient estimates. Finally, the SVCMs will be implemented with and without the SIMEX adjustment in a simulation study (Chapter 4) and the real data application (Chapter 5). Chapter 4 introduces a known amount of measurement error into the data, while Chapter 5 implements these methods on an air quality dataset with an unknown amount of measurement error to act as a sensitivity tool. Finally, Chapter 6 concludes the report and makes suggestions for future studies.

Chapter 2

Literature Review

This chapter reviews regression methods for spatial data, beginning with constant non-spatial multiple linear regression (Section 2.1) and then proceeding to continuous spatial regression (Section 2.2). The discussion of continuous spatial regression focuses on geographically weighted regression, bivariate spline triangulation, and bivariate penalized spline triangulation (Sections 2.2.1 through 2.2.3).

2.1 Non-Spatial Regression

One common approach to modeling spatial data with regression is to assume spatial stationarity, meaning the predictor coefficients (the β parameters) do not vary across space. For simplicity, this report adopts stationarity as shorthand for spatial stationarity (as opposed to other forms of stationarity) unless explicitly noted. Under stationarity, the model estimates a single set of coefficients that remain constant throughout the study area (Brunsdon et al., 1998). A standard example of the stationary approach is multiple linear regression (MLR).

Following the notation of Brunsdon et al. (1998), the MLR model

$$y_i = \sum_{j=0}^{p} X_{ij}\beta_j + \epsilon_i, \qquad \epsilon_i \overset{iid}{\sim} N(0, \sigma^2) \tag{2.1}$$

expresses each response observation y_i as a linear combination of explanatory variables with corresponding coefficients and an error term ϵ_i. Here, $i = 1, \ldots, n$ indexes observations and $j = 0, \ldots, p$ indexes the explanatory variables. The case $j = 0$ corresponds to the intercept term, so $X_{i0} = 1$ for all i and β_0 represents the intercept. The error terms ϵ_i are assumed to be independent and identically distributed, with $\epsilon_i \sim N(0, \sigma^2)$. In matrix form,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I}_n), \tag{2.2}$$

the response $\mathbf{y}$, coefficients $\boldsymbol{\beta}$, and errors $\boldsymbol{\epsilon}$ are vectors, and the explanatory variable values are collected in the design matrix $\mathbf{X}$. Its first column is $\mathbf{1}$, corresponding to the intercept.

Given observed data $(\mathbf{y}, \mathbf{X})$, the unknown coefficient vector $\boldsymbol{\beta}$ is commonly estimated by ordinary least squares (OLS), which minimizes the sum of squared residuals:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \tag{2.3}$$

Here, $\hat{\boldsymbol{\beta}}$ denotes the OLS estimate of $\boldsymbol{\beta}$. In summation form, this minimization problem can be written as

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=0}^{p} X_{ij}\beta_j \right\}^2. \tag{2.4}$$

The fitted values are then given by $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$, or equivalently, $\hat{y}_i = \sum_{j=0}^{p} X_{ij}\hat{\beta}_j$ for $i = 1, \ldots, n$.

As discussed by Brunsdon et al. (1998), unless location is included explicitly as an explanatory variable, spatial position is not present in the model, and the coefficient vector $\boldsymbol{\beta}$ is estimated from $\mathbf{y}$ and $\mathbf{X}$ alone using OLS. This becomes problematic when stationarity is assumed in a study area that is actually non-stationary. In that case, the relationship between the explanatory variables and the response may vary across space, but the model constrains $\boldsymbol{\beta}$ to be constant across space. If this spatial variation is not modeled, then the residuals may violate the independent and identically distributed error assumptions.

2.2 Spatial Regression

Some basic ways to incorporate spatial variability into a regression model include indicator variables and zoning (Brunsdon et al., 1998). Indicator variables can represent broad location-based categories within a geographical study area, such as whether an observation is coastal or inland. Although this approach does not use each observation's exact coordinates to allow coefficients to vary continuously over space, it can still estimate differences associated with geographic features through additional regression coefficients (e.g., a coefficient associated with a coastal indicator, relative to an inland reference category). Alternatively, zoning groups observations into geographical units (e.g., U.S. Census geographical areas), and the data are aggregated within each unit using summary statistics. While indicator variables and zoning both incorporate spatial information into regression models, they do so through a discrete approach. Thus, the coefficients are constant within each category or zone and can change only at their boundaries, rather than varying continuously over space.

In contrast, non-stationary coefficient methods allow relationships between the response and explanatory variables to vary continuously as a function of geographic location using each observation's coordinates. These methods produce location-specific coefficient estimates, and therefore capture spatial variation on a finer scale. Because the regression coefficients are allowed to vary with location, the resulting regression model is now a spatially varying coefficient model (SVCM). The SVCM is expressed as

$$y_i = \sum_{j=0}^{p} X_{ij}\beta_j(\mathbf{s}_i) + \epsilon_i, \qquad \epsilon_i \overset{iid}{\sim} N(0, \sigma^2), \tag{2.5}$$

where $\mathbf{s}_i$ represents a vector of coordinates (e.g., latitude and longitude) for the ith observation, and $\beta_j(\mathbf{s}_i)$ denotes the coefficient for explanatory variable j at location $\mathbf{s}_i$. For notational convenience, $\boldsymbol{\beta}(\mathbf{s}_i)$ denotes the coefficient vector at location $\mathbf{s}_i$, so that $\boldsymbol{\beta}(\mathbf{s}_i) = (\beta_0(\mathbf{s}_i), \ldots, \beta_j(\mathbf{s}_i), \ldots, \beta_p(\mathbf{s}_i))'$. Estimation of the coefficient functions can then be expressed through least squares. A generic formulation in summation form is

$$\hat{\boldsymbol{\beta}}(\cdot) = \arg \min_{\boldsymbol{\beta}(\cdot)\in\mathcal{F}} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=0}^{p} X_{ij}\beta_j(\mathbf{s}_i) \right\}^2 . \tag{2.6}$$

The restriction $\boldsymbol{\beta}(\cdot) \in \mathcal{F}$ provides the structure needed to estimate the coefficient functions from the data. In spline-based approaches such as BST and BPST, $\mathcal{F}$ is specified through a spline space and, for BPST, an additional roughness penalty, leading to a global least squares or penalized least squares estimation problem. In contrast, GWR estimates $\hat{\boldsymbol{\beta}}(\mathbf{s}_i)$ by solving a local weighted least squares problem at each location $\mathbf{s}_i$, with weights determined by distances from nearby observations to $\mathbf{s}_i$. The following subsections therefore describe geographically weighted regression (GWR), bivariate spline triangulation (BST), and bivariate penalized spline triangulation (BPST).

2.2.1 Geographically Weighted Regression (GWR)

GWR is a local modeling approach for estimating the spatially varying coefficients in Equation 2.5. Rather than imposing a global functional form for $\boldsymbol{\beta}(\cdot)$, GWR calibrates a separate regression at each target location $\mathbf{s}_i$ using nearby observations more heavily than distant observations. This is accomplished by assigning distance-based kernel weights when estimating the local coefficients $\boldsymbol{\beta}(\mathbf{s}_i)$ (Brunsdon et al., 1998).

Formally, for each target location $\mathbf{s}_i$, GWR fits a weighted least squares regression in which observations closer to $\mathbf{s}_i$ receive larger weights than those that are farther away. Let α_{ik} denote the weight assigned to observation k when calibrating the local model at $\mathbf{s}_i$, where α_{ik} is a decreasing function of the distance between $\mathbf{s}_i$ and another observation location $\mathbf{s}_k$. The local coefficient estimate at $\mathbf{s}_i$ is then defined by the following weighted least squares problem, similar to Equation 2.6:

$$\hat{\boldsymbol{\beta}}(\mathbf{s}_i) = \arg \min_{\boldsymbol{\beta}(\mathbf{s}_i)\in\mathbb{R}^p} \sum_{k=1}^{n} \alpha_{ik} \left\{ y_k - \sum_{j=0}^{p} X_{kj}\beta_j(\mathbf{s}_i) \right\}^2 . \tag{2.7}$$

Thus, for each fixed target location $\mathbf{s}_i$, GWR estimates a single local coefficient vector $\hat{\boldsymbol{\beta}}(\mathbf{s}_i)$ using all observations, with nearby observations contributing more heavily than distant ones.

The weight α_{ik} is determined by a kernel function applied to the distance d_{ik} between locations $\mathbf{s}_i$ and $\mathbf{s}_k$. For example, $\alpha_{ik} = K(d_{ik}; h)$, where $K(\cdot)$ denotes the kernel function and h denotes the bandwidth parameter controlling the rate of spatial decay (Brunsdon et al., 1998). A common choice is the Gaussian kernel, which yields a gradual radial decay:

$$\alpha_{ik} = \exp\left(-\frac{d_{ik}^2}{2h^2}\right). \tag{2.8}$$

Brunsdon et al. (1998) note that the bandwidth h may be chosen in advance or selected by cross-validation. In leave-one-out cross-validation, candidate bandwidth values are evaluated by predicting each y_i from a local calibration at $\mathbf{s}_i$ with observation i omitted, denoted $\hat{y}_{-i}(h)$, and selecting the value of h that minimizes the resulting sum of squared prediction errors. The bandwidth h controls the spatial scale over which the coefficients are estimated: as h increases, more observations receive non-negligible weight, and the local estimates approach those from a global stationary regression. Conversely, as h decreases, the estimates become more localized and can vary more sharply over space, typically increasing the variance of the coefficient estimates and potentially prediction error.

Over the past fifteen years, GWR has been used to study spatially varying relationships in economic and environmental applications. Wu (2020) cites studies applying GWR to resource management problems in forestry, water resources, and atmospheric systems to account for spatial variation. More recently, researchers have also examined mixed geographically weighted regression (MGWR), which models some regression coefficients as spatially varying GWR terms and others as spatially constant linear regression terms. For example, Alya et al. (2023) compare linear regression, GWR, and MGWR in a study of poverty in Indonesia and report that MGWR outperformed both based on AIC (Akaike Information Criterion) and MSE (mean squared error). Taken together, Wu (2020) and Alya et al. (2023) suggest that GWR-based approaches can outperform multiple linear regression (MLR) when non-stationarity is expected.

Although GWR and spline-based approaches both allow regression coefficients to vary continuously over space, BST/BPST are particularly well-suited to irregular study areas, including domains with complex boundaries and interior holes. This is because BST/BPST represent the coefficient functions on a triangulation of the study area, providing a flexible geometric framework for modeling spatial variation on non-rectangular domains.

2.2.2 Bivariate Spline Triangulation (BST)

BST estimates the spatially varying coefficients in Equation 2.5 by representing each coefficient function $\beta_j(\cdot)$ in the vector $\boldsymbol{\beta}(\cdot)$ as a bivariate spline over a triangulation of the study area. Let $\hat{\boldsymbol{\beta}}(\cdot) = (\hat{\beta}_0(\cdot), \ldots, \hat{\beta}_p(\cdot))'$ denote the estimated coefficient function vector. Rather than calibrating a separate local regression at each target location, as in GWR, BST estimates the coefficient functions globally by minimizing the least squares criterion in Equation 2.6 over the bivariate spline space $\mathbb{S}_d^r(\triangle)$:

$$(\hat{\beta}_0(\cdot), \ldots, \hat{\beta}_p(\cdot)) = \arg \min_{\beta_0, \ldots, \beta_p \in \mathbb{S}_d^r(\triangle)} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=0}^{p} X_{ij} \beta_j(\cdot) \right\}^2 . \tag{2.9}$$

Therefore, unlike GWR, which estimates a local coefficient vector $\hat{\boldsymbol{\beta}}(\mathbf{s}_i)$ at each target location $\mathbf{s}_i$, BST estimates the coefficient functions globally over the triangulated study region. In BST, each $\beta_j(\cdot)$ is represented as a linear combination of spline basis functions defined on the triangulation, with the associated basis coefficients estimated by least squares (Mu et al., 2018; Zhou and Pan, 2014). Accordingly, the remainder of this subsection describes the triangulation $\triangle$ of the study area Ω, introduces the spline space $\mathbb{S}_d^r(\triangle)$, and presents the Bernstein basis representation used to compute $\hat{\beta}_j(\mathbf{s})$. Here, $\hat{\beta}_j(\cdot)$ denotes the estimated spatially varying coefficient function, and $\hat{\beta}_j(\mathbf{s})$ denotes its value at location $\mathbf{s}$.

Triangles are convenient for partitioning because any polygonal region can be decomposed into a finite set of triangles (Li et al., 2021). Since the basis functions are defined with respect to the triangulation of the study area, the choice of triangulation directly determines the bivariate spline model used to represent each coefficient function $\beta_j(\cdot)$ and, through this, its estimate $\hat{\beta}_j(\cdot)$. Although triangulations can be generated in different ways, the triangles in the triangulation are usually required to satisfy a common set of assumptions. First, the collection of triangles $\triangle = \{\tau_m\}_{m=1}^{K}$ must completely partition the study area Ω into K triangles:

$$\Omega = \bigcup_{m=1}^{K} \tau_m. \tag{2.10}$$

Second, each triangle τ_m is non-degenerate, meaning no triangle collapses to a line segment or vertex (Zhou and Pan, 2014). Third, for any pair of these triangles, the intersection is either empty, a common vertex, or a common edge (Li et al., 2021). And fourth, each observation's location $\mathbf{s}_i$ is assumed to lie within a triangle in the triangulation. Consequently, if $\mathbf{s}_i$ lies on a shared vertex or edge, it is counted more than once (Mu et al., 2018). Using these assumptions, the triangulation $\triangle$ can be generated using an algorithm or a restrictive condition. These algorithms include Delaunay and DistMesh (Mu et al., 2018). Alternatively, restrictive conditions can prioritize certain characteristics of the triangulation, such as requiring uniformity in the triangles' areas (Li et al., 2021).

Given the triangulation $\triangle$, $\beta_j(\cdot)$ for explanatory variable j is modeled as a piecewise bivariate polynomial of degree at most d, where d is a non-negative integer, with one polynomial piece defined on each triangle τ_m. The collection of the piecewise polynomial functions is represented by

$$\mathbb{S}_d(\triangle) = \left\{\beta_j(\cdot) : \beta_j|_{\tau_m} \in \mathbb{P}_d(\tau_m),\ m = 1, \ldots, K\right\}, \tag{2.11}$$

where $\beta_j|_{\tau_m}$ denotes the restriction of $\beta_j(\cdot)$ to τ_m, and $\mathbb{P}_d(\tau_m)$ is the space of bivariate polynomials of degree at most d on τ_m. Without continuity constraints across shared edges, $\mathbb{S}_d(\triangle)$ does not yet impose the continuity conditions required for a bivariate spline, and discontinuities may occur along triangle boundaries (Zhou and Pan, 2014).

To make the representation on a single triangle explicit, consider $\beta_j|_{\tau_m}$. On τ_m, the polynomial piece can be written using barycentric coordinates and Bernstein basis polynomials. Specifically, each observation location $\mathbf{s}_i \in \tau_m$ can be uniquely represented in barycentric coordinates as

$$\mathbf{s}_i = b_1\mathbf{v}_1 + b_2\mathbf{v}_2 + b_3\mathbf{v}_3, \qquad b_1 + b_2 + b_3 = 1, \qquad b_1, b_2, b_3 \geq 0, \tag{2.12}$$

where $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are the vertices of τ_m. The barycentric coordinate vector is $\mathbf{b}_i = (b_1, b_2, b_3)$, and this representation is unique under the constraints $b_1 + b_2 + b_3 = 1$ and $b_1, b_2, b_3 \geq 0$ (Zhou and Pan, 2014). These barycentric coordinates are then used in the Bernstein basis polynomials.

The degree-d Bernstein basis polynomials are defined in terms of the barycentric coordinates $\mathbf{b}_i = (b_1, b_2, b_3)$ for a given triangle τ_m. Specifically with

$$B_{d,j;qru}(\mathbf{s}_i) = \frac{d!}{q!r!u!} b_1^q b_2^r b_3^u, \qquad q + r + u = d, \qquad q, r, u \geq 0, \qquad \mathbf{s}_i \in \tau_m, \tag{2.13}$$

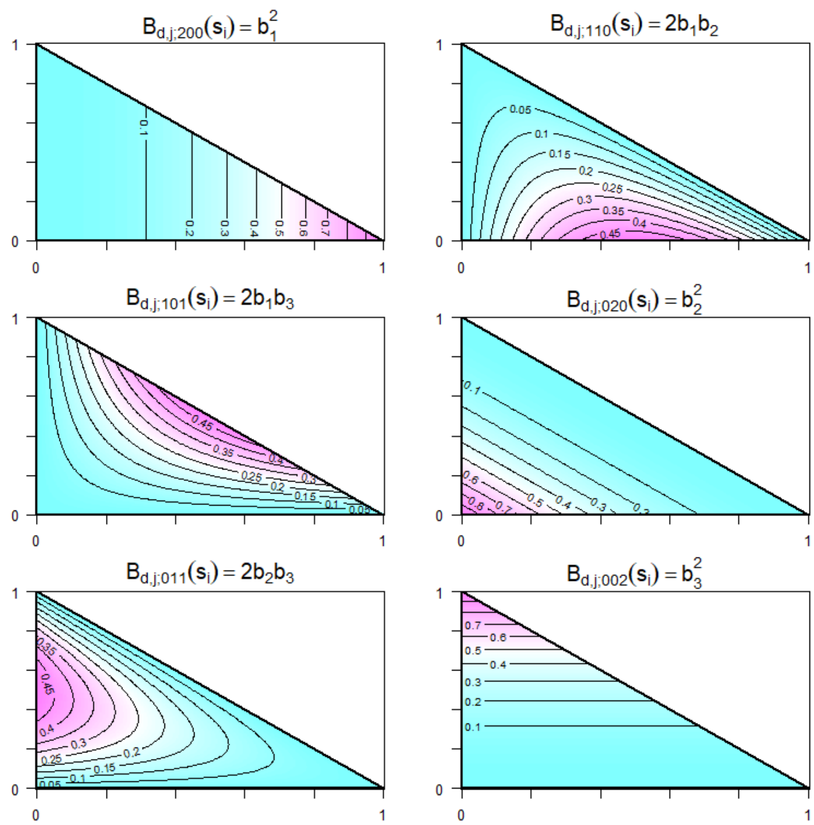
each basis polynomial $B_{d,j;qru}(\mathbf{s}_i)$ is indexed by nonnegative integers q, r, and u, which act as exponents on the barycentric coordinates and satisfy $q+r+u = d$ (Zhou and Pan, 2014). For $d = 2$, Equation 2.13 yields six nonnegative integer triples (q, r, u) satisfying $q + r + u = d$, shown on the right in Figure 2.1. Substituting these triples into Equation 2.13 gives the six degree-2 Bernstein basis polynomials on the triangle τ_m, shown on the left in Figure 2.1.

Using the degree-d Bernstein basis polynomials, the polynomial piece for $\beta_j(\mathbf{s}_i)$ on triangle τ_m can be written as

$$\beta_j(\mathbf{s}_i)|_{\tau_m} = \sum_{q+r+u=d} \gamma_{qru}^{(j,m)} B_{d,j;qru}(\mathbf{s}_i), \qquad \mathbf{s}_i \in \tau_m, \tag{2.14}$$

where each $\gamma_{qru}^{(j,m)}$ is a Bernstein coefficient for explanatory variable j on triangle τ_m (Zhou and Pan, 2014). Thus, a weighted sum of the degree-d Bernstein basis polynomials defines the polynomial piece, and hence the polynomial surface, over triangle τ_m. Evaluating this representation at observation location $\mathbf{s}_i$ yields the coefficient value $\beta_j(\mathbf{s}_i)$.

In matrix form, $\beta_j(\mathbf{s}_i)|_{\tau_m} = \mathbf{B}_{d,j;m}(\mathbf{s}_i)'\boldsymbol{\gamma}_{j,m}$, where $\mathbf{B}_{d,j;m}(\mathbf{s}_i)$ is the vector of Bernstein basis terms evaluated at $\mathbf{s}_i$ and $\boldsymbol{\gamma}_{j,m}$ is the corresponding coefficient vector. To express $\beta_j(\mathbf{s}_i)$



q	r	u
2	0	0
1	1	0
1	0	1
0	2	0
0	1	1
0	0	2

Figure 2.1: Quadratic Bernstein basis functions on a representative triangle for $d = 2$ on the left, with the corresponding index triples (q, r, u) satisfying $q + r + u = 2$ on the right.

over the entire triangulation $\triangle$, rather than on a single triangle τ_m, write $\beta_j(\mathbf{s}_i) = \mathbf{B}_{d,j}(\mathbf{s}_i)'\boldsymbol{\gamma}_j$, which is defined over $\triangle$ (Zhou and Pan, 2014). Substituting this form into Equation 2.6 gives the least squares problem

$$(\hat{\beta}_0(\cdot), \ldots, \hat{\beta}_p(\cdot)) = \arg \min_{\beta_0,\ldots,\beta_p \in \mathbb{S}_d(\triangle)} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=0}^{p} X_{ij} \mathbf{B}_{d,j}(\cdot)'\boldsymbol{\gamma}_j \right\}^2, \tag{2.15}$$

which is used to estimate the global coefficient functions over the triangulation (Mu et al., 2018).

With the piecewise-polynomial representation in place, potential discontinuities at triangle boundaries can be addressed by imposing continuity constraints across shared edges. By the definition of $\mathbb{S}_d(\triangle)$, no continuity is enforced across shared edges in $\triangle$, so $\beta_j(\cdot)$ may be discontinuous at triangle boundaries (Zhou and Pan, 2014). To obtain a surface that is at least continuous across all shared edges, smoothness is introduced through a nonnegative integer r. Let $C^r(\Omega)$ denote the collection of all functions over the study area Ω with continuous derivatives up to order r. Then the intersection of $C^r(\Omega)$ and $\mathbb{S}_d(\triangle)$ yields the smoother spline space $\mathbb{S}_d^r(\triangle)$. In particular, $\mathbb{S}_d^r(\triangle)$ restricts the piecewise polynomials so that $\beta_j(\cdot)$ is continuous across shared triangle edges when $r = 0$. For $r \geq 1$, the derivatives of $\beta_j(\cdot)$ up to order r must be continuous as well. Thus,

$$\mathbb{S}_d^r(\triangle) = C^r(\Omega) \cap \mathbb{S}_d(\triangle) = \{\beta_j(\cdot) \in C^r(\Omega) : \beta_j|_{\tau_m} \in \mathbb{P}_d(\tau_m), m = 1, \ldots, K\} \tag{2.16}$$

enforces smoothness by requiring $\beta_j(\cdot) \in C^r(\Omega)$ (Mu et al., 2018; Zhou and Pan, 2014). With the $C^r(\Omega)$ smoothness constraint, the polynomial pieces join smoothly across shared edges, forming a bivariate spline.

Smoothness across shared edges can be enforced by imposing linear constraints on the Bernstein coefficients, which leads to a constrained least squares problem. To enforce $C^r(\Omega)$ smoothness across shared edges, the coefficient vector $\boldsymbol{\gamma}_j$ is required to satisfy a set of linear constraints $\mathbf{H}_j\boldsymbol{\gamma}_j = 0$. The constraint matrix $\mathbf{H}_j$ is determined by the triangulation $\triangle$, the degree d, and the smoothness r (Mu et al., 2018; Zhou and Pan, 2014). To impose these constraints, a QR decomposition of $\mathbf{H}_j$ is used to construct a reparameterization of $\boldsymbol{\gamma}_j$. The transpose of the constraint matrix $\mathbf{H}_j'$ has the QR decomposition $\mathbf{H}_j' = (\mathbf{Q}_{1,j}\ \mathbf{Q}_{2,j})(\mathbf{R}_{j,1}\ 0)'$, where $(\mathbf{Q}_{1,j}\ \mathbf{Q}_{2,j})$ is an orthogonal matrix and $\mathbf{R}_{j,1}$ is an upper triangular matrix. From this, the reparameterization of $\boldsymbol{\gamma}_j$ is $\boldsymbol{\gamma}_j = \mathbf{Q}_{2,j}\boldsymbol{\theta}_j$ for some $\boldsymbol{\theta}_j$ (Zhou and Pan, 2014). Using this reparameterization, the constrained least squares problem becomes (Mu et al., 2018):

$$(\hat{\beta}_0(\cdot), \ldots, \hat{\beta}_p(\cdot)) = \arg \min_{\beta_0, \ldots, \beta_p \in \mathbb{S}_d^r(\triangle)} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=0}^{p} X_{ij}\mathbf{B}_{d,j}(\cdot)'\mathbf{Q}_{2,j}\boldsymbol{\theta}_j \right\}^2. \tag{2.17}$$

To simplify the computation of Equation 2.17, it can be assumed that the triangulation $\triangle$, degree d, and smoothness r are the same for all explanatory variables. Under this assumption, the same spline basis vector $\mathbf{B}_d(\mathbf{s})$ is used for all j, so $\mathbf{H}_j = \mathbf{H}$ and $\mathbf{Q}_{2,j} = \mathbf{Q}_2$

for every explanatory variable (Mu et al., 2018):

$$(\hat{\beta}_0(\cdot), \ldots, \hat{\beta}_p(\cdot)) = \arg \min_{\beta_0,\ldots,\beta_p \in \mathbb{S}_d^r(\triangle)} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=0}^{p} X_{ij} \mathbf{B}_d(\cdot)' \mathbf{Q}_2 \boldsymbol{\theta}_j \right\}^2 . \tag{2.18}$$

To estimate the vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_0', \boldsymbol{\theta}_1', \ldots, \boldsymbol{\theta}_p')'$, Mu et al. use the Kronecker product to solve the corresponding least squares problem for $\hat{\boldsymbol{\theta}}$ (Mu et al., 2018). From $\hat{\boldsymbol{\theta}}$, estimates of $\boldsymbol{\gamma}_j$ and $\beta_j(\mathbf{s}_i) = \mathbf{B}_d(\mathbf{s}_i)'\boldsymbol{\gamma}_j$ are obtained, denoted by $\hat{\boldsymbol{\gamma}}_j$ and $\hat{\beta}_j(\mathbf{s}_i)$, respectively (Mu et al., 2018).

BST is often used with an additional penalty term to control the smoothness of the fitted bivariate spline surfaces. This penalized version is referred to as bivariate penalized spline triangulation (BPST).

2.2.3 Bivariate Penalized Spline Triangulation (BPST)

Building on BST, BPST adds a roughness penalty to the least squares criterion to control the smoothness of the fitted bivariate spline surface for each coefficient function $\beta_j(\cdot)$. This smoothness control allows each coefficient surface to capture large-scale spatial trends without overfitting local noise (Mu et al., 2018; Zhou and Pan, 2014).

BPST estimates the coefficient surfaces by minimizing a penalized residual sum of squares over the spline space $\mathbb{S}_d^r(\triangle)$, obtained by adding a penalty term $\lambda_j \, Pen\{\beta_j(\cdot)\}$ for each coefficient function:

$$(\hat{\beta}_0(\cdot), \ldots, \hat{\beta}_p(\cdot)) = \arg \min_{\beta_0,\ldots,\beta_p \in \mathbb{S}_d^r(\triangle)} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=0}^{p} X_{ij} \beta_j(\cdot) \right\}^2 + \sum_{j=0}^{p} \lambda_j \, Pen\{\beta_j(\cdot)\}. \tag{2.19}$$

Since a separate smoothness penalty is applied to each coefficient surface, the penalty term appears as a sum over j. The remainder of this subsection introduces λ_j, describes $Pen\{\beta_j(\cdot)\}$ and common penalty choices, and shows how representing each $\beta_j(\cdot)$ with a Bernstein spline basis (together with the C^r smoothness constraints) rewrites Equation 2.18 as a penalized least squares problem in terms of a finite set of spline basis coefficients under the common $(\triangle, d, r)$ assumption across j.

Within the penalty term $\lambda_j\, Pen\{\beta_j(\cdot)\}$, there are two components: a penalty parameter λ_j and a roughness penalty $Pen\{\beta_j(\cdot)\}$. The value of the penalty parameter λ_j controls the weight of the roughness penalty in the minimization. If λ_j is small, the contribution of the penalty term $\lambda_j\, Pen\{\beta_j(\cdot)\}$ will be small. As a result, the bivariate spline for $\beta_j(\mathbf{s})$ will favor fitting the observed data more closely rather than enforcing a smooth trend. Conversely, a larger λ_j will place more weight on smoothness, which will give a smoother estimated regression coefficient surface (Zhou and Pan, 2014).

The roughness penalty $Pen\{\beta_j(\cdot)\}$ measures the roughness of the coefficient surface and determines which types of surface features are penalized. This depends on the specific form of $Pen\{\beta_j(\cdot)\}$ that is selected. A common choice is

$$Pen_1\{\beta_j(\cdot)\} = \int_\Omega \left\{ \left(\frac{\partial^2 \beta_j}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 \beta_j}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 \beta_j}{\partial y^2}\right)^2 \right\} dx\, dy, \tag{2.20}$$

which is known as the thin-plate spline penalty (Mu et al., 2018; Zhou and Pan, 2014). The thin-plate spline penalty does not penalize linear functions. This is because for a linear surface (i.e., a plane), the second-order derivatives are identically zero, making the penalty evaluate to zero. Another choice is the squared Laplacian penalty

$$Pen_2\{\beta_j(\cdot)\} = \int_\Omega \left(\frac{\partial^2 \beta_j}{\partial x^2} + \frac{\partial^2 \beta_j}{\partial y^2}\right)^2 dx\, dy. \tag{2.21}$$

This option leaves a much larger class of surfaces unpenalized because curvature in the x and y directions may offset each other, resulting in the penalty evaluating to zero. The roughness penalty $Pen\{\beta_j(\cdot)\}$ is small when the coefficient surface is smooth and becomes larger as the surface becomes more variable, reflecting increased curvature (Zhou and Pan, 2014). It is also important to note that because the roughness penalty is invariant to rotation or translation of spatial coordinates, any coordinate system can be used to define these bivariate splines (Mu et al., 2018; Zhou and Pan, 2014).

To incorporate the penalty term $\lambda_j\, Pen\{\beta_j(\cdot)\}$ into the penalized least squares minimization for BPST, the roughness penalty is first expressed as a quadratic form in the spline coefficients (i.e., $\boldsymbol{\gamma}_j'\mathbf{P}_j\boldsymbol{\gamma}_j$). The minimization is then expressed sequentially under the Bernstein basis representation, the smoothness-constrained reparameterization, and the common $(\triangle, d, r)$ assumption across j.

First, under the Bernstein basis representation $\beta_j(\mathbf{s}) = \mathbf{B}_{d,j}(\mathbf{s})'\boldsymbol{\gamma}_j$, the roughness penalty can be written as $Pen\{\beta_j(\cdot)\} = Pen\{\mathbf{B}_{d,j}(\mathbf{s})'\boldsymbol{\gamma}_j\} = \boldsymbol{\gamma}_j'\mathbf{P}_j\boldsymbol{\gamma}_j$, where $\mathbf{P}_j$ is the penalty matrix associated with $\beta_j(\mathbf{s})$. Substituting this into the least squares minimization in Equation 2.15 yields the penalized objective

$$(\hat{\beta}_0(\cdot), \ldots, \hat{\beta}_p(\cdot)) = \arg \min_{\beta_0,\ldots,\beta_p \in \mathbb{S}_d(\triangle)} \sum_{i=1}^{n} \{y_i - \sum_{j=0}^{p} X_{ij}\mathbf{B}_{d,j}(\cdot)'\boldsymbol{\gamma}_j\}^2 + \sum_{j=0}^{p} \lambda_j\, \boldsymbol{\gamma}_j'\mathbf{P}_j\boldsymbol{\gamma}_j. \quad (2.22)$$

Next, under the Bernstein basis representation with the smoothness constraints $\beta_j(\mathbf{s}) = \mathbf{B}_{d,j}(\mathbf{s})'\mathbf{Q}_{2,j}\boldsymbol{\theta}_j$, the penalty can be written as $Pen\{\beta_j(\cdot)\} = \boldsymbol{\theta}_j'\mathbf{Q}_{2,j}'\mathbf{P}_j\mathbf{Q}_{2,j}\boldsymbol{\theta}_j$. In accordance with Equation 2.17, the penalty for the least squares minimization

$$\begin{aligned}(\hat{\beta}_0(\cdot), \ldots, \hat{\beta}_p(\cdot)) = \arg \min_{\beta_0,\ldots,\beta_p \in S_d^r(\triangle)} \Bigg[\sum_{i=1}^{n} \Bigg\{ y_i - \sum_{j=0}^{p} X_{ij}\, \mathbf{B}_{d,j}(\cdot)'\mathbf{Q}_{2,j}\boldsymbol{\theta}_j \Bigg\}^2 \\ + \sum_{j=0}^{p} \lambda_j\, \boldsymbol{\theta}_j'\mathbf{Q}_{2,j}'\mathbf{P}_j\mathbf{Q}_{2,j}\boldsymbol{\theta}_j \Bigg].\end{aligned} \quad (2.23)$$

will now enforce the smoothness constraints established in the BST subsection (Mu et al., 2018).

Finally, for simplicity, the assumption can be made that the triangulation $\triangle$, polynomial degree d, and smoothness r are the same for each explanatory variable j, so that $\mathbf{Q}_{2,j} = \mathbf{Q}_2$ and the BST least squares minimization reduces to Equation 2.18. Under this assumption, the penalty term simplifies to $Pen\{\mathbf{B}_d(\mathbf{s})'\mathbf{Q}_2\boldsymbol{\theta}_j\} = \boldsymbol{\theta}_j'\mathbf{Q}_2'\mathbf{P}\mathbf{Q}_2\boldsymbol{\theta}_j$ with the same penalty $\mathbf{P}$ given for each explanatory variable. This yields the penalized least squares problem (Mu et al., 2018):

$$
\begin{aligned}
(\hat{\beta}_0(\cdot), \ldots, \hat{\beta}_p(\cdot)) = \arg \min_{\beta_0, \ldots, \beta_p \in \mathbb{S}_d^r(\triangle)} \Bigg[& \sum_{i=1}^{n} \left\{ y_i - \sum_{j=0}^{p} X_{ij} \mathbf{B}_{d,j}(\cdot)' \mathbf{Q}_2 \boldsymbol{\theta}_j \right\}^2 \\
& + \sum_{j=0}^{p} \lambda \, \boldsymbol{\theta}_j' \mathbf{Q}_2' \mathbf{P} \mathbf{Q}_2 \boldsymbol{\theta}_j . \Bigg]
\end{aligned} \tag{2.24}
$$

From Equation 2.24, $\hat{\boldsymbol{\theta}}_j$ can be obtained using a Kronecker-product formulation. The corresponding Bernstein coefficient estimates are $\hat{\boldsymbol{\gamma}}_j = \mathbf{Q}_2 \hat{\boldsymbol{\theta}}_j$, and the fitted coefficient surfaces are $\hat{\beta}_j(\mathbf{s}_i) = \mathbf{B}_d(\mathbf{s}_i)' \hat{\boldsymbol{\gamma}}_j$.

In practice, the penalty parameter must be selected before the BPST estimator can be computed. The appropriate penalty parameter can be selected using a cross-validation method. With Equation 2.24, only one penalty parameter λ needs to be selected since a common smoothing parameter is used across coefficient surfaces (i.e., $\lambda_j = \lambda$ for all j) (Mu et al., 2018). Both Zhou and Pan (2014) and Mu et al. (2018) discuss generalized cross-validation, although Mu et al. (2018) found that multi-fold cross-validation performs just as well. The λ value that is selected from cross-validation can then be used to determine the estimates described in the previous paragraph.

In recent years, BPST has been extended within the statistics profession. For example, Kim et al. (2025) proposed bivariate penalized spline quantile regression, which estimates regression coefficients for a specified quantile of the response conditional on the predictors and spatial location, rather than for the mean response. While most of the literature focuses on methodological adaptations of BPST, there are occasional instances where BPST has been applied in non-statistical settings. In Yin et al. (2018), BPST-based models were used to identify which factors (such as slope steepness and rainfall erosivity) were most strongly associated with soil erosion. Even in work proposing new methods, authors often include applications as proof of concept. For example, Li et al. (2021) applied BPST to cross-sectional images of the human brain to identify factors associated with imaging patterns related to Alzheimer's disease. Overall, while BPST can be applied in many situations, it has not reached many fields outside of statistics yet.

Chapter 3

Measurement Error and SIMEX

This chapter introduces Simulation-Extrapolation (SIMEX) as a method for correcting measurement error in the non-spatial and spatial regression models presented in Chapter 2. Its implementation in Chapters 4 and 5 allow comparison of model performance with and without measurement error correction. Section 3.1 defines measurement error in explanatory variables. Following this, Section 3.2 presents the SIMEX procedure for MLR, and Section 3.3 presents the SIMEX procedures for BST and BPST.

3.1 Measurement Error in Explanatory Variables

Measurement error arises when information on explanatory variables is recorded with error. This problem can occur when the information is recorded inaccurately (lack of accuracy) or imprecisely (lack of precision). A lack of accuracy leads to biased explanatory variables, while a lack of precision leads to increased variability. This report focuses on the latter issue: lack of precision. To quantify and address this issue, it is important to first formulate measurement error.

Continuing the notation from Chapter 2, let X_{ij} denote the ith true (latent) value associated with explanatory variable j. Here, $i = 1, \ldots, n$ indexes observations and $j = 0, \ldots, p$ indexes the explanatory variables. The case $j = 0$ corresponds to the intercept term, so

$X_{i0} = 1$ for all i and the intercept column is assumed to be measured without error. Measurement error is represented by U_{ij}, so the observed value satisfies

$$W_{ij} = X_{ij} + U_{ij}. \tag{3.1}$$

For $j = 1, \ldots, p$, the measurement errors U_{ij} are assumed to have mean zero and variance σ_j^2 (Stefanski, 2000). For notational convenience, define $U_{i0} = 0$, and hence $W_{i0} = X_{i0} = 1$ and $\sigma_0^2 = 0$, for the intercept column. The variance σ_j^2 represents the magnitude of measurement error for explanatory variable j and is used to quantify lack of precision. Larger values of σ_j^2 indicate greater measurement error in the observed value W_{ij}.

If non-negligible measurement error is present in the explanatory variables, then proceeding with least squares in non-spatial multiple linear regression (MLR) without accounting for it will generally induce attenuation bias (Stefanski, 2000). Attenuation bias is the tendency for the regression coefficient parameter β_j to be biased toward zero and can be expressed as

$$\mathrm{E}[\hat{\beta}_j] = \frac{\beta_j}{1 + \frac{\sigma_j^2}{\sigma_{X_j}^2}}, \tag{3.2}$$

with $\mathrm{E}[\hat{\beta}_j]$ being the expected value of $\hat{\beta}_j$. The corresponding attenuation factor is $(1 + \sigma_j^2/\sigma_{X_j}^2)^{-1}$, where σ_j^2 denotes the variance of the measurement error for explanatory variable j and $\sigma_{X_j}^2$ denotes the variance of the true values for explanatory variable j. Therefore, Equation 3.2 shows that when measurement error is ignored, $\hat{\beta}_j$ is biased toward zero (Stefanski, 2000). For nonlinear spatially varying coefficient methods such as BST and BPST, ignoring measurement error in least squares estimation can also attenuate regression coefficient estimates. Although this attenuation is not summarized by a simple expression like Equation 3.2, it can still be addressed (Section 3.3). For example, Stefanski (2000) describes nonlinear regression settings (such as logistic regression) in which the consequences of explanatory variable measurement error cannot be easily determined.

Since this report includes spatial regression methods, it should be noted that in this work, measurement error is considered only in the explanatory variables. Each observation location $\mathbf{s}_i$ is assumed to be recorded both accurately and precisely.

3.2 SIMEX in MLR

One adjustment to least squares estimation that accounts for measurement error is the SIMEX algorithm. SIMEX (short for simulation and extrapolation) corrects bias in the parameter estimates by adding simulated measurement error and extrapolating back to the case of no measurement error (Cook and Stefanski., 1994).

The remaining sections of this chapter describe the overall SIMEX procedure, starting with its required inputs and the resulting output. This section focuses on applying SIMEX to the non-spatial MLR method.

The inputs required to run the SIMEX algorithm include the response vector $\mathbf{y}$ from the matrix form of MLR (Equation 2.2) and the matrix of observed explanatory variables $\mathbf{W}$. From Equation 3.1, the observed explanatory variables satisfy $\mathbf{W} = \mathbf{X} + \mathbf{U}$. Because measurement error is assumed, the latent values in the non-intercept columns of $\mathbf{X}$ are unobserved. SIMEX uses $\mathbf{W}$ in place of $\mathbf{X}$, with the intercept column fixed at $\mathbf{1}$ and the remaining columns given by the observed explanatory variables. In addition to $\mathbf{y}$ and $\mathbf{W}$, the remaining inputs include the assumed or estimated measurement-error variance matrix $\boldsymbol{\Sigma}_U$, the simulation levels λ, and the number of simulations B per λ level (Cook and Stefanski., 1994). The measurement error variance matrix is a diagonal matrix $\boldsymbol{\Sigma}_U = \text{diag}\left(0, \sigma_1^2, \ldots, \sigma_p^2\right)$, with each σ_j^2 element representing the measurement error variance for explanatory variable j. The leading zero reflects that the intercept column is observed without error. The simulation levels λ are used to identify how parameter estimates change as additional measurement error is introduced. Typically, the simulation levels λ are chosen as a set of 5–7 nonnegative values denoted Λ (e.g., $\Lambda = \{0, 0.5, 1.0, 1.5, 2.0\}$) (Cook and Stefanski., 1994). Finally, B denotes the number of Monte Carlo replicates drawn at each λ level, which are averaged to obtain an estimate at that simulation level (Cook and Stefanski., 1994).

The output from the SIMEX algorithm is the coefficient vector

$$\hat{\boldsymbol{\beta}}_{\text{SIMEX}} = \left(\hat{\beta}_{0,\text{SIMEX}}, \hat{\beta}_{1,\text{SIMEX}}, \ldots, \hat{\beta}_{p,\text{SIMEX}}\right)', \tag{3.3}$$

where $\hat{\beta}_{j,\text{SIMEX}}$ ($j = 0, \ldots, p$) denotes the measurement error-adjusted estimate of β_j. Although the intercept has no measurement error ($\sigma_0^2 = 0$), $\hat{\beta}_{0,\text{SIMEX}}$ is included because the SIMEX-adjusted fit yields an updated intercept estimate alongside the measurement error-adjusted slope estimates. The remainder of this section provides the detailed SIMEX formulation for non-spatial MLR and outlines each step used to obtain $\hat{\boldsymbol{\beta}}_{\text{SIMEX}}$.

The SIMEX algorithm begins by using the inputs of $\mathbf{W}$, $\boldsymbol{\Sigma}_U$, and Λ to define $\mathbf{W}^{(\lambda)}$:

1. For each $\lambda \in \Lambda$, define $\mathbf{W}^{(\lambda)} = \mathbf{W} + \sqrt{\lambda}\, \mathbf{U}^*$.

For the first step, the matrix $\mathbf{W}^{(\lambda)}$ is constructed by adding simulated measurement error, $\sqrt{\lambda}\mathbf{U}^*$, to the measurement error-prone observed explanatory variable matrix $\mathbf{W}$. Specifically, each row of the error matrix $\mathbf{U}^*$ is generated independently from a distribution with mean zero and covariance $\boldsymbol{\Sigma}_U$. The magnitude of the added error is controlled by the simulation level λ through the scaling factor $\sqrt{\lambda}$ (Cook and Stefanski., 1994). Because the true explanatory variable matrix $\mathbf{X}$ is unobserved (Equation 3.1), this step introduces a controlled error-contamination process for assessing how coefficient estimates respond to increasing measurement error. Since the first diagonal element of $\boldsymbol{\Sigma}_U$ is zero, the intercept column of $\mathbf{W}^{(\lambda)}$ remains equal to $\mathbf{1}$ for all $\lambda \in \Lambda$.

2. For each $\lambda \in \Lambda$:

    a. For $b = 1, \ldots, B$:

        (i) Apply Step 1 with an independent draw $\mathbf{U}^{*(b)}$ to obtain:

$$\mathbf{W}^{(\lambda,b)} = \mathbf{W} + \sqrt{\lambda}\, \mathbf{U}^{*(b)}.$$

(ii) Fit the regression of $\mathbf{y}$ on $\mathbf{W}^{(\lambda,b)}$ to obtain:

$$\hat{\boldsymbol{\beta}}^{(\lambda,b)} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=0}^{p} \beta_j \, w_{ij}^{(\lambda,b)} \right\}^2,$$

where $w_{ij}^{(\lambda,b)}$ denotes the (i,j) entry of $\mathbf{W}^{(\lambda,b)}$.

b. After completing $b = 1, \ldots, B$, compute the Monte Carlo average estimates:

$$\hat{\boldsymbol{\beta}}(\lambda) = \frac{1}{B} \sum_{b=1}^{B} \hat{\boldsymbol{\beta}}^{(\lambda,b)}.$$

For the second step, the input B controls the number of Monte Carlo replications used to estimate the regression coefficients under simulated measurement error (Cook and Stefanski., 1994). In each replication b, a new error matrix $\mathbf{U}^{*(b)}$ is generated. Typically, the same set of draws $\{\mathbf{U}^{*(b)}\}_{b=1}^{B}$ is reused across all $\lambda \in \Lambda$, with the magnitude of the added error controlled by the factor $\sqrt{\lambda}$ (Cook and Stefanski., 1994). The resulting coefficient estimates obtained via least squares are then averaged over b within each λ to obtain more stable simulation-level estimates.

3. For the regression coefficients $j = 0, 1, \ldots, p$:

a. Fit an extrapolation function with λ as the input and the Monte Carlo average estimates as the output. For each $j = 0, 1, \ldots, p$, fit $g_j(\lambda)$ to $\hat{\beta}_j(\lambda)$.

b. Evaluate the fitted functions at $\lambda = -1$ to obtain the SIMEX estimates:

$$\hat{\beta}_{j,\text{SIMEX}} = g_j(-1), \quad j = 0, 1, \ldots, p.$$

c. Output:

$$\hat{\boldsymbol{\beta}}_{\text{SIMEX}} = \left(\hat{\beta}_{0,\text{SIMEX}}, \hat{\beta}_{1,\text{SIMEX}}, \ldots, \hat{\beta}_{p,\text{SIMEX}} \right)'.$$

For the third step, fitted extrapolation functions are evaluated at $\lambda = -1$ to obtain the SIMEX measurement error-adjusted coefficient estimates. Although different extrapolation forms could be used for the individual regression coefficients, it is common to apply a single form to all of them. Accordingly, the implementations in Chapters 4 and 5 use linear extrapolation for all the regression coefficients. Another frequent selection is quadratic extrapolation. These choices follow typical SIMEX practice in which low-order polynomial extrapolants are used to model the relationship between the estimates and λ (Cook and Stefanski., 1994). Figure 3.1 illustrates these extrapolation functions for a representative coefficient. Because the conditional measurement error variance satisfies $\mathrm{Var}(\mathbf{W}^{(\lambda)} \mid \mathbf{X}) = (1+\lambda)\mathbf{\Sigma}_U$, evaluating at $\lambda = -1$ corresponds to extrapolating to the limiting case of zero measurement-error variance (Cook and Stefanski., 1994). This addresses bias due to imprecise measurements in the explanatory variables.

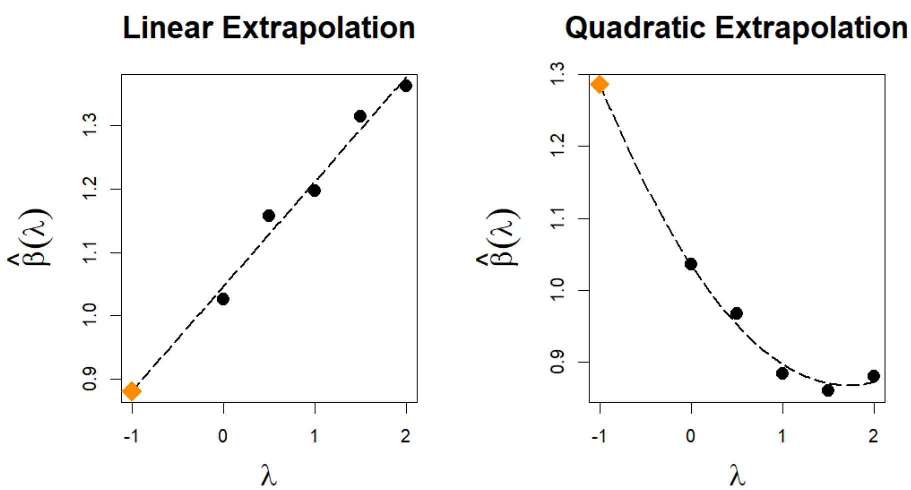


Figure 3.1: Illustrative linear and quadratic extrapolation examples for the grid $\Lambda = \{0, 0.5, 1.0, 1.5, 2.0\}$. Solid points represent example Monte Carlo average estimates $\hat{\beta}(\lambda)$, and dotted curves show the fitted extrapolation functions in λ. The SIMEX estimates (orange diamond markers) are obtained by evaluating the fitted functions at $\lambda = -1$.

As a whole, the SIMEX algorithm contains simulation in Steps 1 and 2 and extrapolation in Step 3. In general, increasing B and the size of the Λ set will result in more stable SIMEX estimates, at the cost of increased computational time proportional to $B \times |\Lambda|$.

3.3 SIMEX in BST and BPST

To adjust the BST and BPST methods in Chapter 2 for measurement error, SIMEX is applied as in Section 3.2, except that the target parameters are the transformed spline coefficient vectors $\{\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{j,\text{SIMEX}}\}_{j=0}^{p}$ associated with the triangulation $\triangle$, rather than the single regression coefficient vector $\hat{\boldsymbol{\beta}}_{\text{SIMEX}}$. Collecting these vectors across j yields the matrix $\widehat{\mathbf{Q}_2\boldsymbol{\Theta}}_{\text{SIMEX}} = \left[\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{0,\text{SIMEX}} \cdots \widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{p,\text{SIMEX}}\right]$. This choice matches the BST/BPST fitting step, which naturally yields the transformed coefficients $\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_j = \mathbf{Q}_2\hat{\boldsymbol{\theta}}_j$ that enter directly into the coefficient function evaluation. In the implementation, at each SIMEX replication the BST/BPST model is refit and the transformed coefficients $\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_j^{(\lambda,b)}$ are recorded, then averaged over B and extrapolated to obtain $\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{j,\text{SIMEX}}$ for $j = 0, 1, \ldots, p$. The complete SIMEX algorithms for BST and BPST are given in Appendix A.1 and Appendix A.2, respectively.

Given $\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{j,\text{SIMEX}}$, the SIMEX-adjusted coefficient functions are obtained under the reparameterized BST/BPST formulations (Equation 2.18 and Equation 2.24) as $\hat{\beta}_{j,\text{SIMEX}}(\mathbf{s}) = \mathbf{B}_d(\mathbf{s})'\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{j,\text{SIMEX}}$ for $j = 0, 1, \ldots, p$. Evaluating at an observed location $\mathbf{s}_i$ gives $\hat{\beta}_{j,\text{SIMEX}}(\mathbf{s}_i)$. As discussed in Section 3.1, this adjustment attributes measurement error to the explanatory variables only. Each location $\mathbf{s}_i$, and therefore the common basis vector evaluated at each location $\mathbf{B}_d(\mathbf{s}_i)$, is assumed to be observed without error.

SIMEX has been used in other spatial settings but does not appear to be widely applied to BST/BPST. For example, Alexeeff et al. (2016) applied SIMEX in a universal Kriging framework and considered spatially correlated measurement errors. More recent work has continued to focus on location-related error mechanisms. In this chapter, however, explanatory variable measurement error is assumed to be independent across locations. This assumption is made to focus on whether correcting explanatory variable measurement error improves BST/BPST. Under this assumption, the SIMEX implementation follows the same logic as in MLR, except that the extrapolated quantities are the BST/BPST spline coefficient estimates.

Chapter 4

Simulation Study

In this chapter, a horseshoe-domain simulation following Mu et al. (2018) is used to compare linear regression, GWR, BST, and BPST, introduced in Chapter 2. Because the measurement error setting includes only one explanatory variable, the linear regression model reduces to simple linear regression (SLR), allowing a clearer comparison of the methods with and without SIMEX. Measurement error with a specified variance is added to the simulated explanatory variable, and SLR, BST, and BPST are fitted using the contaminated variable, both with and without SIMEX. GWR is excluded from the SIMEX comparison. Performance is evaluated using computation time, mean squared error (MSE), and mean squared prediction error (MSPE). The study design is described in Section 4.1, and the results are presented in Section 4.2.

4.1 Simulation Study Design

The simulation study design consists of the data-generating process (Subsection 4.1.1) and the estimation and evaluation procedure (Subsection 4.1.2).

4.1.1 Data Generation Process

First, the data is simulated without measurement error, following the exact process used by Mu et al. (2018). The data is generated on a horseshoe ("sausage") domain. The domain is constructed using 361,301 evenly spaced points, with 901 unique values of the first coordinate s_1 (horizontal) and 401 unique values of the second coordinate s_2 (vertical). The true spatially varying coefficient model is

$$y_i = \sum_{j=0}^{1} X_{ij}\beta_j(\mathbf{s}_i) + \epsilon_i, \qquad \epsilon_i \overset{iid}{\sim} N(0, 0.5^2). \tag{4.1}$$

The intercept term is defined as $X_{i0} = 1$, with spatially varying coefficient surface $\beta_0(\mathbf{s})$ generated using the horseshoe domain test function of Wood (2003), which was modified from Ramsay (2002) (left panel of Figure 4.1). The model also includes a single explanatory variable $X_{i1} \sim \text{Unif}(0, 2)$ with coefficient surface $\beta_1(\mathbf{s})$. Letting $\mathbf{s} = (s_1, s_2)$, the true coefficient surface for X_{i1} is $\beta_1(\mathbf{s}) = 4\sin\big(0.05\pi(s_1^2 + s_2^2)\big)$ (right panel of Figure 4.1). Finally, the error term is generated from a normal distribution, $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$, with $\sigma = 0.5$.

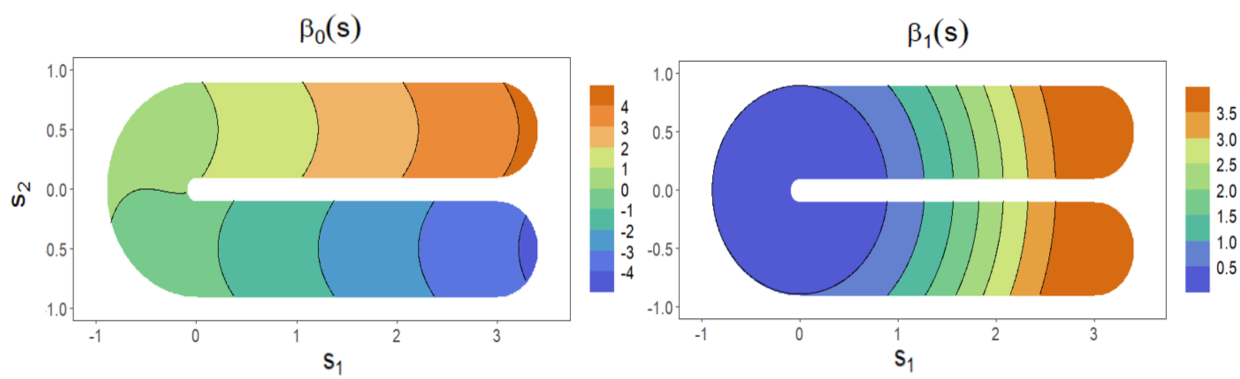


Figure 4.1: The simulation's true spatially varying coefficient surfaces $\beta_0(\mathbf{s})$ and $\beta_1(\mathbf{s})$ corresponding to the intercept term X_{i0} and explanatory variable X_{i1}, respectively.

Next, measurement error is introduced into the explanatory variable by defining the observed quantity $W_{i1}^{(r)}$ for repetition r as $W_{i1}^{(r)} = X_{i1} + U_{i1}^{(r)}$, where $U_{i1}^{(r)} \overset{iid}{\sim} N(0, \sigma_U^2)$ with $\sigma_U = 0.8$. The intercept is assumed to be observed without error (Section 3.1), so $W_{i0} = X_{i0} = 1$. To reduce simulation randomness, the measurement-error-contaminated explanatory variable is generated four times and then averaged, giving:

$$\bar{W}_{i1} = \frac{1}{4}\sum_{r=1}^{4} W_{i1}^{(r)} = X_{i1} + \frac{1}{4}\sum_{r=1}^{4} U_{i1}^{(r)}. \tag{4.2}$$

All methods are fit using $\bar{W}_{i1}$ as the observed explanatory variable, while X_{i1} is treated as the true latent measurement error-free explanatory variable. Since $\bar{U}_{i1}$ is the average of four independent error terms, it follows that $\bar{U}_{i1} \sim N\left(0, \frac{\sigma_U^2}{4}\right)$. Therefore, the measurement error in $\bar{W}_{i1}$ has standard deviation $\sigma_1 = \frac{\sigma_U}{2} = 0.4$.

4.1.2 Estimation and Evaluation Procedure

For estimation, 100 Monte Carlo samples are drawn from the simulated population data of 361,301 points for each of the three sample sizes, $n = 500$, $n = 1000$, and $n = 2000$, to fit the models described in Chapter 2. The SLR, GWR, BST, and BPST models are fitted without SIMEX. To account for measurement error, the SLR, BST, and BPST models are also fitted with SIMEX. GWR is not fitted with SIMEX because applying SIMEX to GWR would require additional methodological choices, such as how bandwidth selection should be handled across simulation levels. By contrast, for BST and BPST the triangulation and spline basis are fixed in advance, so SIMEX is implemented by re-estimating the model coefficients on that fixed structure. This makes the SIMEX implementation for BST and BPST more comparable to the standard regression-based SIMEX framework considered in this report.

Regardless of the model, the SIMEX algorithm inputs include the measurement error variance matrix $\mathbf{\Sigma}_U = \text{diag}\,(0,\ 0.4^2)$, the simulation level set $\Lambda = \{0, 0.25, 0.5, 1\}$, the number of Monte Carlo replicates $B = 20$, and the linear extrapolation function. Although SIMEX typically uses 5–7 simulation levels (Chapter 3), the four values in the Λ set and the choice of replicates $B = 20$ were adopted as a practical compromise between adequate extrapolation and computational burden. A linear extrapolation function is used for numerical stability, since higher-order extrapolation can produce erratic or excessively large extrapolated estimates.

In addition to the common SIMEX settings, model specifications are required for GWR, BST, and BPST. For GWR, the bandwidth parameter h is selected using the `spgwr` package in R. For BST and BPST, the polynomial degree $d = 2$ and the smoothness $r = 1$. The triangulation is constructed using the Delaunay algorithm through the `Triangulation` package. This produces a triangulation with 68 triangles and 58 vertices. A plot of the triangulation used for BST and BPST is located in Figure 4.2. For BPST, the penalty parameter is selected by cross-validation using the `BPST` package over a grid of 9 equally spaced values on the log scale from 10^{-2} to 10^{2}. With the exception of the triangulation, the BST and BPST specifications follow Mu et al. (2018).

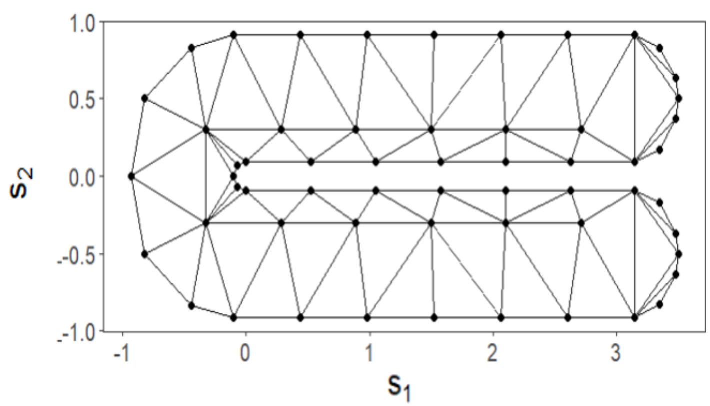


Figure 4.2: The Delaunay triangulation used for BST and BPST. It consists of 68 triangles and 58 vertices.

The models are fit using a combination of custom code and R packages. Specifically, the SLR model is fit using the `lm()` function, while the GWR model is fit using the `spgwr` package. The BST and BPST models are implemented using custom code together with the `BPST` package. The SIMEX adjustments are also implemented through custom code. In each case, the custom code is designed to follow the methodology described in Chapters 2 and 3. It should also be noted that while BST is usually completely unpenalized, a very small ridge constant, 10^{-6}, was added to the coefficient estimation step to improve numerical stability in samples where the estimating matrix was nearly singular. This adjustment was used only for numerical stabilization and was not intended as a smoothing penalty.

For evaluation, the estimated regression coefficients, $\hat{\beta}_j$ or $\hat{\beta}_j(\cdot)$, and the response values, $\hat{y}_i$, are compared across models. Since the data is generated in the simulation study, the true coefficient values are known, which allows direct assessment of coefficient estimation performance. In real data applications, the true coefficients are typically unknown. The evaluation criteria include mean squared error (MSE), mean squared prediction error (MSPE), and computation time. MSE is defined as the average squared error over the sampled simulation data of size n:

$$\mathrm{MSE}(\hat{\beta}_j) = \frac{1}{n}\sum_{i=1}^{n}\left\{\hat{\beta}_j(\mathbf{s}_i) - \beta_j(\mathbf{s}_i)\right\}^2, \qquad j = 0, 1 \tag{4.3}$$

$$\mathrm{MSE}(\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}\left\{\hat{y}_i - y_i\right\}^2. \tag{4.4}$$

For SLR, the regression coefficient for each explanatory variable is constant over space, so $\hat{\beta}_j$ is used in place of $\hat{\beta}_j(\mathbf{s}_i)$ in Equation 4.3. Similarly, MSPE is defined as the average squared error over all observations in the simulated population:

$$\mathrm{MSPE}(\hat{\beta}_j) = \frac{1}{N}\sum_{i=1}^{N}\left\{\hat{\beta}_j(\mathbf{s}_i) - \beta_j(\mathbf{s}_i)\right\}^2, \qquad j = 0, 1 \tag{4.5}$$

$$\mathrm{MSPE}(\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}\left\{\hat{y}_i - y_i\right\}^2. \tag{4.6}$$

Like Equation 4.5, for SLR, $\hat{\beta}_j$ is used in place of $\hat{\beta}_j(\mathbf{s}_i)$. For each model and sample size, MSE and MSPE are first computed within each Monte Carlo replication and then averaged over the 100 replications to obtain the values reported in Section 4.2. Computational efficiency is also assessed by averaging the recorded computation times over the Monte Carlo replications.

4.2 Simulation Study Results

This section presents the simulation results for the methods without the SIMEX adjustment (Subsection 4.2.1) and the methods with the SIMEX adjustment (Subsection 4.2.2).

4.2.1 Non-SIMEX Results

The non-SIMEX results suggest patterns in coefficient surface recovery, response prediction, and computation time. For this simulation study on the irregular horseshoe domain, BPST is the strongest method overall, followed closely by GWR.

For the coefficient surfaces, BPST performs the best at each sample size according to MSE and MSPE, followed by GWR. Table 4.1 shows that BPST attains the smallest MSE and MSPE values for $\beta_0(\mathbf{s})$ and $\beta_1(\mathbf{s})$. This indicates that BPST provides the most accurate recovery of the true coefficient surfaces among the methods considered and therefore, performs best over the full horseshoe domain. GWR performs similarly, though not as well, likely because it is less well suited than BPST to the irregular domain. The main departure from the results of Mu et al. (2018), which were obtained using the true explanatory variable, is that BST does not outperform GWR. Although BST is also designed for irregular domains, its lack of penalization may make it more sensitive to local fluctuations in the sampled data under measurement error, resulting in rougher and less stable coefficient surface estimates. It is also important to note that the triangulation used here differs from that of Mu et al. (2018), which may also contribute to the difference in BST performance. As expected, SLR performs considerably worse than the spatially varying coefficient models because it estimates a single constant coefficient over a domain where the true coefficient surfaces vary spatially. Overall, these results are consistent with the idea that, in the presence of measurement error, the penalty in BPST helps stabilize coefficient estimation and contributes to more accurate recovery of the coefficient surfaces than the unpenalized BST fit.

For a visual comparison of the coefficient surface estimates, Figure 4.3 displays the estimated coefficient surfaces averaged over the 100 Monte Carlo replications at $n = 2000$. In the figure, SLR shows constant estimates for both the intercept $\hat{\beta}_0(\mathbf{s})$ and explanatory variable coefficient $\hat{\beta}_1(\mathbf{s})$. Therefore, SLR does not capture the spatial variation in the true surfaces. Among GWR, BST, and BPST, the main spatial features are recovered in both coefficient surfaces. BST and BPST also appear to better preserve the true contour shape.

Sample Size	Method	$\beta_0(\mathbf{s})$		$\beta_1(\mathbf{s})$		y		Time (s)
		MSE	MSPE	MSE	MSPE	MSE	MSPE	
$n = 500$	SLR	5.958	5.955	2.348	2.330	8.926	8.877	0.003
	GWR	0.646	0.653	0.599	0.597	0.628	0.916	1.263
	BST	0.795	0.973	0.708	0.821	0.641	1.124	0.470
	BPST	0.568	0.566	0.540	0.537	0.724	0.837	0.604
$n = 1000$	SLR	5.920	5.932	2.332	2.327	8.855	8.866	0.001
	GWR	0.579	0.584	0.552	0.555	0.662	0.848	4.931
	BST	0.609	0.659	0.578	0.598	0.708	0.903	0.623
	BPST	0.539	0.540	0.521	0.521	0.745	0.808	0.667
$n = 2000$	SLR	5.929	5.930	2.338	2.339	8.816	8.855	0.002
	GWR	0.549	0.550	0.533	0.535	0.699	0.812	15.312
	BST	0.565	0.574	0.547	0.546	0.737	0.845	1.376
	BPST	0.535	0.536	0.516	0.518	0.754	0.792	1.018

Table 4.1: Monte Carlo simulation results for the non-SIMEX estimators SLR, GWR, BST, and BPST for sample sizes $n = 500$, 1000, and 2000. Entries give the average MSE and MSPE for $\beta_0(\mathbf{s})$, $\beta_1(\mathbf{s})$, and y, along with the average computation time in seconds. These averages are calculated over 100 Monte Carlo replications.

However, none of these methods fully reproduce the more extreme low and high regions.

BPST also performs best in response prediction, showing the strongest generalization among the four models based on the relationship between MSE and MSPE. In the y column of Table 4.1, both GWR and BST attain smaller MSE values than BPST, likely because they capture more local variation in the sampled data. However, over the full domain, BPST attains the smallest MSPE among the four models. SLR again performs worst by a substantial margin. Therefore, when emphasis is placed on MSPE as a measure of predictive ability over the full domain rather than only at the sampled points, BPST performs best. GWR remains a close competitor.

Although the main focus of this simulation study is on coefficient recovery and response prediction, computation time (in seconds) provides an additional comparison of the practical cost of each method. Looking at Table 4.1, SLR is essentially instantaneous at all sample sizes. Among the spatially varying coefficient models, the runtime increases with sample size for GWR, BST, and BPST, but the increase is much steeper for GWR. BST and BPST remain considerably faster than GWR across all sample sizes, especially for $n = 1000$ and

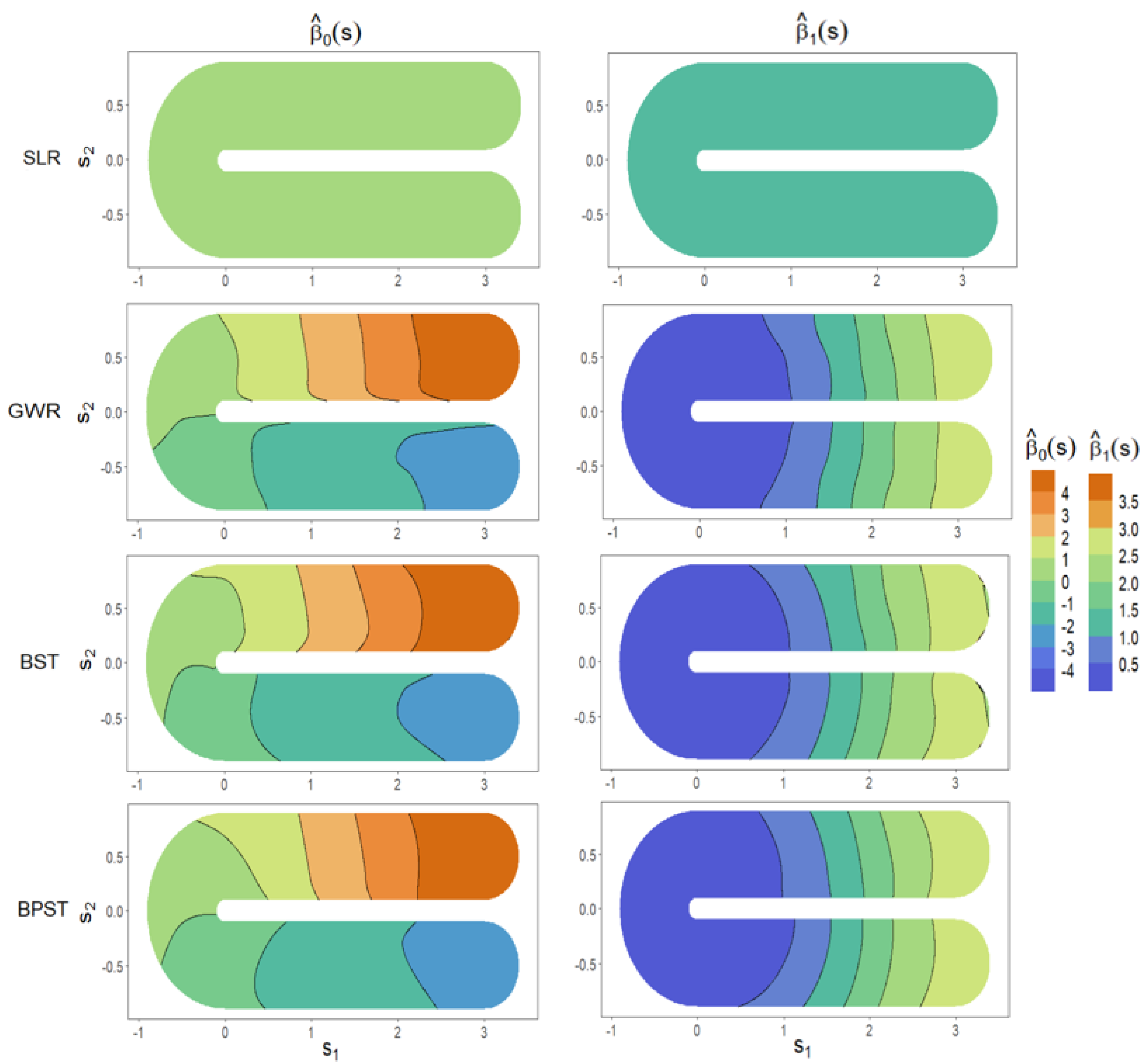


Figure 4.3: Estimated coefficient surfaces averaged over the Monte Carlo replications for the non-SIMEX methods SLR, GWR, BST, and BPST on the horseshoe domain with sample size $n = 2000$. The left column shows $\hat{\beta}_0(\mathbf{s})$ and the right column shows $\hat{\beta}_1(\mathbf{s})$. Common color scales, matching those used for the true coefficient surfaces in Figure 4.1, are used within each column to facilitate direct comparison.

$n = 2000$. This indicates that, for larger sample sizes, BPST offers more computational efficiency over GWR.

In summary, the results from this subsection suggest that BPST performs the best among the methods considered for this horseshoe-domain setting with measurement error.

4.2.2 SIMEX Results

The SIMEX results indicate that applying the SIMEX adjustment can improve the performance of the SLR, BST, and BPST methods. These improvements can be seen in the coefficient surface estimates, response predictions, and computation times.

For each method, the SIMEX coefficient surface estimates have smaller MSE and MSPE values for $\hat{\beta}_0(\mathbf{s})$ and $\hat{\beta}_1(\mathbf{s})$ than their non-SIMEX counterparts, except for BST when $n = 500$. In this case, the SIMEX BST method has a lower MSE but higher MSPE which shows it is more susceptible to local variation in the sampled data (see Table 4.2). It is also interesting to note that with the SIMEX adjustment, BST outperforms GWR for $n = 1000$ and $n = 2000$.

Visually, these coefficient surfaces recover more of the original surface than the non-SIMEX methods presented in Subsection 4.2.1. Although not all of the color regions are recovered in the surfaces shown in Figure 4.4, they capture more of the extreme low and high values than the corresponding surfaces in Figure 4.3.

For each sample size in Table 4.2, most SIMEX-adjusted models yield larger MSE and MSPE values than the corresponding non-SIMEX models. Although this may appear to indicate poorer performance, these results should be interpreted in the context of measurement error correction. Because SIMEX is designed specifically to adjust for measurement error, smaller MSE and MSPE values are not necessarily more desirable in this setting. Instead, the smaller values may reflect a closer fit to the contaminated explanatory variable instead of more accurate recovery of the latent, error-free explanatory variable. Therefore, although the MSE and MSPE values for y are generally larger under the SIMEX-adjusted models, the coefficient estimate results suggest that the SIMEX-adjusted methods more accurately recover the true coefficient surfaces. From this perspective, these findings should be regarded

Sample Size	Method	$\beta_0(\mathbf{s})$		$\beta_1(\mathbf{s})$		y		Time (s)
		MSE	MSPE	MSE	MSPE	MSE	MSPE	
$n = 500$	SLR	5.800	5.791	2.188	2.176	8.945	8.903	0.140
	GWR	—	—	—	—	—	—	—
	BST	0.669	1.063	0.584	0.902	0.698	1.482	39.242
	BPST	0.234	0.236	0.205	0.207	0.722	0.921	46.757
$n = 1000$	SLR	5.760	5.773	2.173	2.168	8.873	8.888	0.155
	GWR	—	—	—	—	—	—	—
	BST	0.351	0.436	0.299	0.341	0.765	1.048	50.300
	BPST	0.193	0.194	0.173	0.174	0.797	0.879	58.952
$n = 2000$	SLR	5.758	5.762	2.167	2.169	8.883	8.871	0.223
	GWR	—	—	—	—	—	—	—
	BST	0.231	0.240	0.221	0.216	0.796	0.939	110.515
	BPST	0.170	0.172	0.155	0.156	0.810	0.860	89.115

Table 4.2: Monte Carlo simulation results for the SIMEX estimators with SLR, BST, and BPST for sample sizes $n = 500$, 1000, and 2000. Entries give the average MSE and MSPE for $\beta_0(\mathbf{s})$, $\beta_1(\mathbf{s})$, and y, along with the average computation time in seconds. These averages are calculated over 100 Monte Carlo replications.

as encouraging.

Although computation time is not emphasized to the same extent as the other metrics in Table 4.2, it is substantially longer for the SIMEX-adjusted models. For SLR, the computation time remains negligible, although it increases considerably under SIMEX. For BST, the computation time rises to approximately 1–2 minutes, compared with at most about 1 second without SIMEX. A similar pattern is observed for BPST, with an average computation time of approximately 1.5 minutes when $n = 2000$. Additionally, at the largest sample size, BST requires more time to fit than BPST. This may suggest an inverse relationship between BST and BPST computation times at larger sample sizes. However, additional simulations at larger sample sizes would be needed to investigate this further.

The results of this simulation study corroborate the methods discussed in Chapters 2 and 3. Although incorporating the SIMEX adjustment increases computation time, the simulation results indicate that it more effectively recovers the true coefficient surfaces in the presence of measurement error.

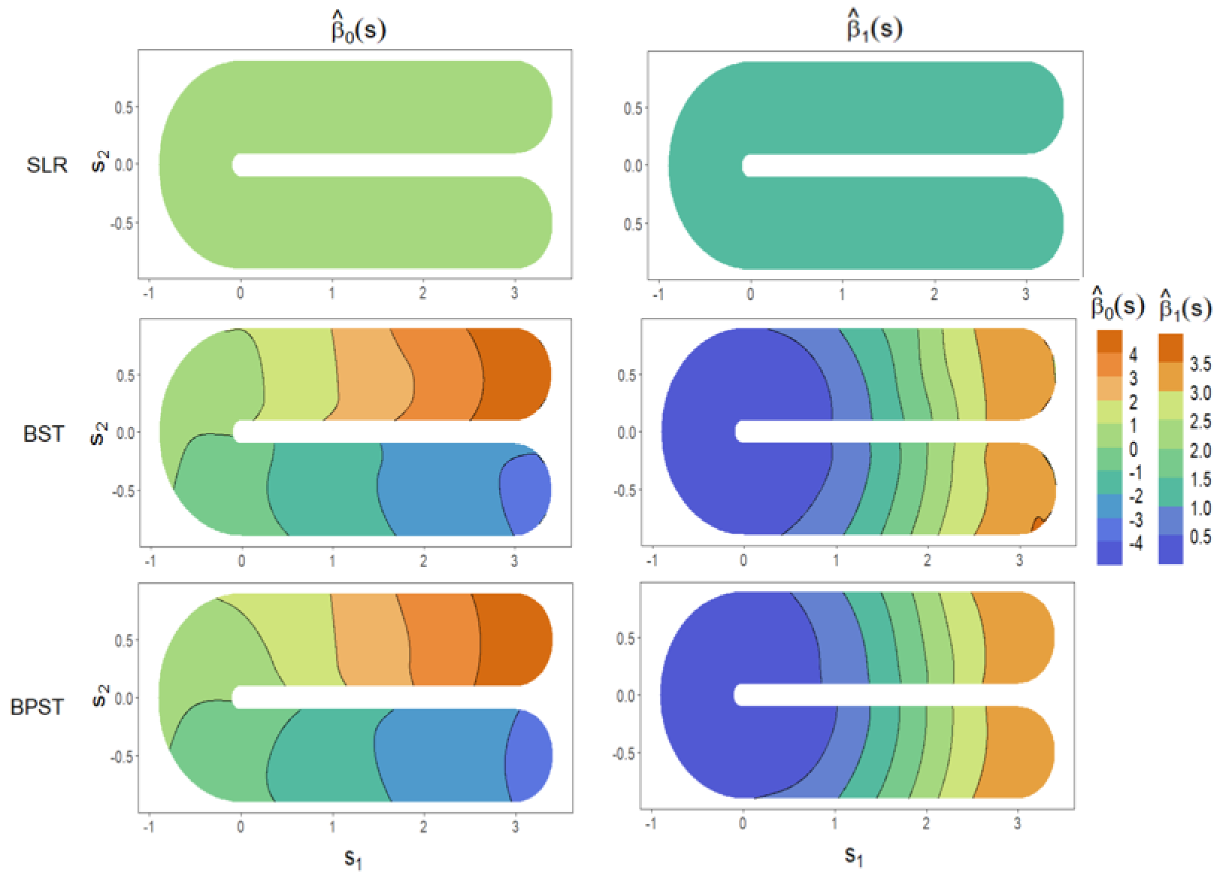


Figure 4.4: Estimated coefficient surfaces averaged over the Monte Carlo replications for the SIMEX methods SLR, BST, and BPST on the horseshoe domain with sample size $n = 2000$. The left column shows $\hat{\beta}_0(\mathbf{s})$ and the right column shows $\hat{\beta}_1(\mathbf{s})$. Common color scales, matching those used for the true coefficient surfaces in Figure 4.1, are used within each column to facilitate direct comparison.

Chapter 5

Real Data Analysis

This chapter applies SLR, GWR, and BPST to an observed NOAA air quality dataset over the contiguous United States. For comparability, only one explanatory variable is used. As in Chapter 4, GWR is not fitted with SIMEX, and BST is omitted in favor of the more stable BPST approach. Because the measurement error magnitude is unknown, SIMEX is applied under several assumed error standard deviations to assess sensitivity. Section 5.1 describes the data and estimation procedure, and Section 5.2 presents the results.

5.1 Real Data Description and Analysis Procedure

The real data analysis includes a description of the NOAA air quality dataset (Subsection 5.1.1) and the estimation and evaluation procedure (Subsection 5.1.2).

5.1.1 Real Data Description

This real data application is based on the dataset considered in (Mu et al., 2018). The response variable is the daily average $\mathrm{PM}_{2.5}$ concentration in μg per m^3 during winter 2011 at locations across the contiguous United States. In contrast to Mu et al. (2018), who considered multiple meteorological explanatory variables, the present analysis uses only wind speed in meters per second, denoted by WS. This simplification maintains consistency with the simulation study in Chapter 4, which also used a single explanatory variable, and provides a more straightforward setting for assessing the potential contribution of measurement error.

For context, $PM_{2.5}$ refers to fine particulate matter with diameter less than or equal to 2.5 microns. Due to their small size, these particles are easily inhaled and can penetrate deeply into the respiratory system. Elevated $PM_{2.5}$ concentrations have been associated with serious adverse health outcomes, including respiratory disease, cardiovascular disease, and increased mortality (Zhai et al., 2022). Previous studies have identified wind speed as an important meteorological factor related to $PM_{2.5}$ levels in multiple countries (Liu, 2021; Zhai et al., 2022). In particular, these studies report a negative association between wind speed and $PM_{2.5}$ concentrations, supporting the use of wind speed as the explanatory variable in this application.

The original dataset contains $1,249$ locations across the contiguous United States. Although some daily $PM_{2.5}$ observations are missing, aggregating the data over the winter season, defined as December, January, and February, addresses this issue for the present analysis. When constructing the triangulation for BPST, two observations were found to lie outside the triangulated domain. To ensure comparability across methods, these observations were removed, leaving a final dataset of $1,247$ locations. These locations and the ones that were removed can be seen in Figure 5.1.

In the simulation study (Chapter 4), the explanatory variable had a known measurement error standard deviation. In the NOAA air quality dataset, however, the standard deviation of the measurement error for wind speed (WS) is unknown. Therefore, in this applied setting, the SIMEX adjustment is implemented under a range of assumed measurement error standard deviations to examine whether the fitted results are sensitive to potential measurement error in the explanatory variable. In this case, the SIMEX adjustment is applied to the SLR and BPST models.

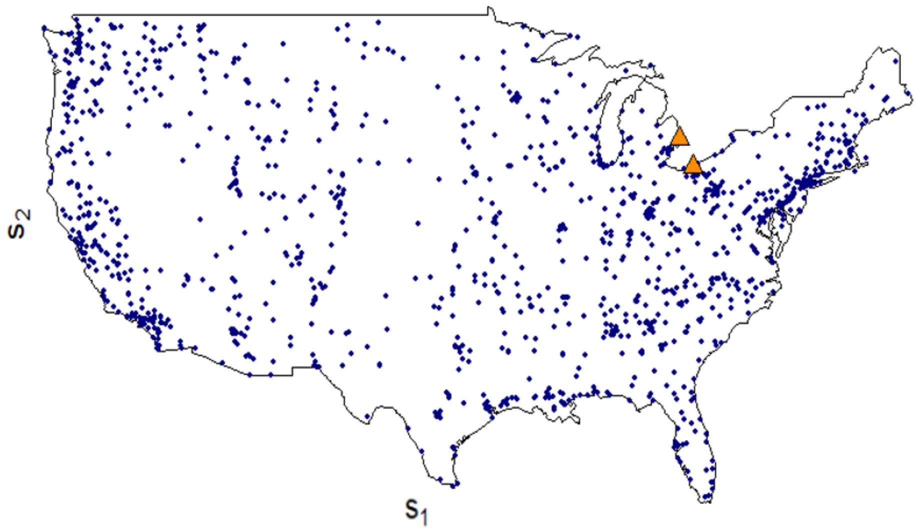


Figure 5.1: The $PM_{2.5}$ monitoring locations across the contiguous United States. The blue points indicate the 1,247 retained locations used in the analysis, while the orange triangles indicate the two observations removed because they lie outside the triangulation configured for BPST.

5.1.2 Estimation and Evaluation Procedure

For estimation, the SLR, GWR, and BPST models are fit without SIMEX, and SLR and BPST are fit with SIMEX to investigate potential measurement error.

The GWR and BPST models are used to fit the spatially varying coefficient model relating wind speed (WS) to $PM_{2.5}$:

$$\mathrm{PM}_{2.5,i} = \beta_0(\mathbf{s}_i) + \beta_1(\mathbf{s}_i)\mathrm{WS}_i + \epsilon_i. \tag{5.1}$$

Here, $\beta_0(\mathbf{s}_i)$ denotes the spatially varying intercept at location $\mathbf{s}_i$, $\beta_1(\mathbf{s}_i)$ denotes the spatially varying coefficient for wind speed WS_i, and ϵ_i is the random error term.

The SLR model is also fit as a baseline for comparison:

$$\mathrm{PM}_{2.5,i} = \beta_0 + \beta_1\mathrm{WS}_i + \epsilon_i. \tag{5.2}$$

This SLR model contains non-spatial, constant coefficients β_0 and β_1. In contrast to GWR and BPST, this model assumes that the relationship between wind speed and $PM_{2.5}$ is constant across all locations.

The BST model is not included in this application because BPST is its penalized counterpart and provides a more stable alternative in practice. Since the two models have the same underlying structure aside from the penalty term, only BPST is retained for the real data analysis.

For the SLR and BPST models, the SIMEX algorithm requires specification of the measurement error variance matrix $\mathbf{\Sigma}_U = \operatorname{diag}(0,\ \sigma_1^2)$. Since the exact measurement error standard deviation for wind speed is unknown, assumed error levels of 5%, 10%, 20%, and 30% of the sample standard deviation of wind speed were considered. In the dataset, the sample standard deviation of wind speed is approximately 0.817, rounded to three decimal places. Accordingly, the assumed values of σ_1 were approximately 0.041, 0.082, 0.163, and 0.245, corresponding to 5%, 10%, 20%, and 30% of the sample standard deviation of wind speed, respectively. These values represent relatively small measurement error scenarios. Given that the data was sourced from a professionally maintained monitoring system, larger error levels were assumed to be less plausible.

The remaining SIMEX inputs were the simulation level set $\Lambda = \{0, 0.25, 0.5, 1\}$, the number of Monte Carlo replicates $B = 40$, and a linear extrapolation function. The simulation level set was chosen to represent a sequence of increasing measurement error levels while keeping the number of simulation points manageable. The number of Monte Carlo replicates was set to $B = 40$, which was more feasible in this application because fewer runs were required than in the simulation study.

The model specifications for GWR remain the same as in the simulation study in Chapter 4. The BPST model uses polynomial degree $d = 3$, smoothness parameter $r = 1$, and a grid of 11 equally spaced penalty parameter values on the $\log_2$ scale, ranging from 2^0 to 2^{10}. The triangulation was generated using the MATLAB `DistMesh` package and consists of 177 triangles and 120 vertices. Figure 5.2 displays this triangulation over the contiguous United States, and Figure 5.3 shows the observation locations within the triangulation. Otherwise, the same packages and customized code are used for SLR, GWR, BPST, and the SIMEX adjustments.

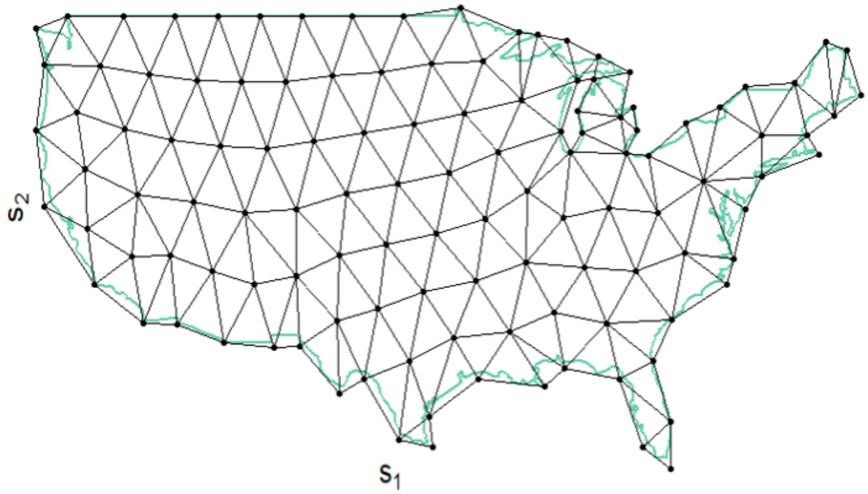


Figure 5.2: The DistMesh triangulation used for BPST over the contiguous United States. It consists of 177 triangles and 120 vertices.

For evaluation, MSE and MSPE are assessed only for the response variable $PM_{2.5}$. The corresponding MSE and MSPE for the regression coefficients cannot be evaluated in this application because the true coefficients are unknown, as is typically the case in observed-data settings. Consequently, the evaluation procedure used in this chapter differs from that used in the simulation study and is described below.

As in the simulation study, the MSE is calculated as the average squared error. In this application, however, it is computed over the entire dataset of size $N = 1{,}247$ to summarize the overall in-sample fit. Formally, the MSE is calculated as

$$\mathrm{MSE}(\widehat{\mathrm{PM}}_{2.5}) = \frac{1}{N}\sum_{i=1}^{N}\left(\widehat{\mathrm{PM}}_{2.5,i} - \mathrm{PM}_{2.5,i}\right)^2. \tag{5.3}$$

Unlike in the simulation study, the MSPE is not computed from a separate prediction set generated from a known model. Instead, it is obtained using 10-fold cross-validation. In each fold, the dataset is partitioned into 10 approximately equal subsets, with 9 subsets used for model fitting and the remaining validation subset used for prediction. Because the total number of observations is not evenly divisible by 10, the fold sizes differ slightly. The predicted values from all 10 validation folds are then combined, and the MSPE is computed

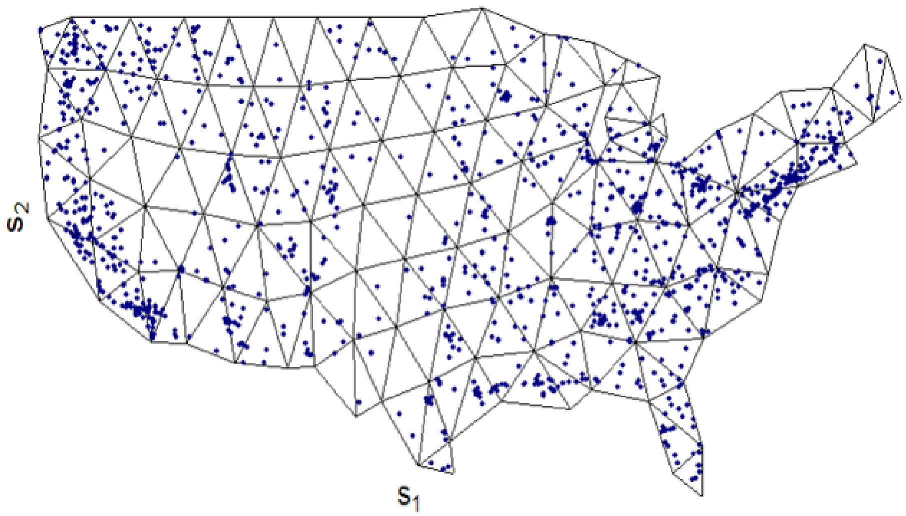


Figure 5.3: The triangulation for BPST with the monitoring locations overlaid.

over the full set of held-out observations. The MSPE based on 10-fold cross-validation is calculated as

$$\mathrm{MSPE}(\widehat{\mathrm{PM}}_{2.5}) = \frac{1}{N}\sum_{k=1}^{10}\sum_{i=1}^{n_k}\left(\widehat{\mathrm{PM}}_{2.5,ik} - \mathrm{PM}_{2.5,ik}\right)^2, \tag{5.4}$$

where n_k is the size of the kth validation subset, and $\widehat{\mathrm{PM}}_{2.5,ik}$ denotes the predicted value for observation i in fold k obtained from the model fit excluding the kth validation subset.

With the estimation and evaluation procedure defined, Section 5.2 presents the corresponding real data analysis results.

5.2 Real Data Analysis Results

This section presents the real data analysis results for the methods without the SIMEX adjustment (Subsection 5.2.1) and with the SIMEX adjustment under different assumed measurement error standard deviations (Subsection 5.2.2), and discusses the implications of these results for the potential presence of measurement error.

5.2.1 Non-SIMEX Results

The real data analysis results for the non-SIMEX methods generally follow the same pattern observed in the simulation study (Chapter 4). In terms of MSPE, BPST outperforms both GWR and SLR.

Model	$PM_{2.5}$	
	MSE	MSPE
SLR	21.138	21.186
GWR	10.218	14.346
BPST	10.243	13.774

Table 5.1: Real data analysis results for the non-SIMEX estimators with SLR, GWR, and BPST for prediction of $PM_{2.5}$ across the contiguous United States using an intercept and the wind speed as an explanatory variable. Entries give the MSE and MSPE.

As shown in Table 5.1, for prediction of $PM_{2.5}$ across the contiguous United States, BPST shows the strongest generalization performance. Although its MSE is slightly higher than that of GWR, its MSPE is lower than that of both other models. As expected from the simulation study, the SLR model produced substantially larger values of both MSE and MSPE than the spatially varying coefficient models. Unlike in the simulation study, BPST did not outperform GWR in terms of MSE. However, it did achieve a noticeably lower MSPE. This suggests that GWR fit the data more locally than BPST and, as a result, was less able to capture the broader trend in unseen observations.

The difference in performance between the GWR and BPST models is especially apparent in their estimated coefficient surfaces. As shown in Figure 5.4, the GWR model produces coefficient surfaces with substantially wider ranges than those of BPST. In particular, the GWR intercept surface appears to vary over a very large range, and the wind speed coefficient also appears to take relatively extreme values. Given that the observed $PM_{2.5}$ concentrations in the dataset range from 0.555 to 39.777 μg per m^3, for a total range of approximately 39.2 μg per m^3, these coefficient estimates appear implausibly large. This suggests that the GWR model may be producing unstable local estimates. In fact, during bandwidth selection for the GWR model in the cross validation, the `spgwr` package also produced a warning indicating that `NA` or `NaN` values occurred in the optimization routine and were replaced by a large positive value. This suggests that the GWR fitting procedure encountered numerical instability for some candidate bandwidths, possibly due to sparse monitoring locations or weak local information in some regions. In contrast, the BPST method yields much more plausible spatially varying coefficients, with both the intercept and wind speed surfaces varying over a more moderate range. In terms of these coefficient surfaces, BPST outperformed GWR.

As a whole, the model that should be used for investigating the relationship between wind speed and $PM_{2.5}$ is the BPST model.

5.2.2 SIMEX Results and Implications

The results discussed in this subsection indicate that a small amount of measurement error, if present, does not have a substantial impact on model performance. Therefore, in an applied setting, the BPST model without SIMEX adjustment appears to be the more appropriate choice.

ME % (ME Std. Dev. [m/s])	Model	$PM_{2.5}$	
		MSE	MSPE
5% (0.041)	SLR	21.138	21.186
	GWR	–	–
	BPST	10.236	13.771
10% (0.082)	SLR	21.138	21.192
	GWR	–	–
	BPST	10.216	13.609
20% (0.163)	SLR	21.138	21.194
	GWR	–	–
	BPST	10.192	13.631
30% (0.245)	SLR	21.138	21.196
	GWR	–	–
	BPST	10.179	13.647

Table 5.2: Real data analysis results for the SIMEX estimators with SLR and BPST for prediction of $PM_{2.5}$ across the contiguous United States using an intercept and wind speed as an explanatory variable. The measurement error (ME) level is reported as a percentage of the standard deviation of wind speed, with the corresponding measurement error standard deviation (in m/s) given in parentheses. Entries give the MSE and MSPE.

From Table 5.2, the performance of both SLR and BPST is fairly stable across the range of assumed measurement error levels. For SLR, the MSE remains unchanged to three decimal places, while the MSPE increases only slightly as the measurement error level increases. For BPST, both MSE and MSPE change only slightly across the assumed measurement error levels, and the resulting values remain very close to those from the original non-SIMEX fit in Table 5.1. In particular, the BPST MSE ranges from 10.179 to 10.236, compared with 10.243 for the original fit, while the BPST MSPE ranges from 13.609 to 13.771, compared with 13.774 for the original fit. Overall, these results suggest that small to somewhat moderate measurement error in wind speed does not substantially affect model performance in this application.

This is further demonstrated by the estimated coefficient surfaces. Comparing the fitted surfaces at the 10% measurement error level with those from the original BPST fit in Figure 5.5, the overall shapes remain very similar, although the ranges of the coefficient estimates are slightly wider. This suggests that a small amount of measurement error has little practical effect on the estimated coefficients. The 10% measurement error level is shown because it represents a modest and practically relevant scenario. Although larger measurement error levels were also considered, the 10% case illustrates that the fitted coefficient surfaces remain broadly similar to those from the original BPST fit. At the higher measurement error levels, the estimated coefficient surfaces preserved similar spatial patterns, but the coefficient ranges became progressively wider.

From these results, the BPST model without SIMEX adjustment appears to be adequate for this application, since incorporating small amounts of measurement error produces only minimal changes in both predictive performance and the estimated coefficient surfaces.

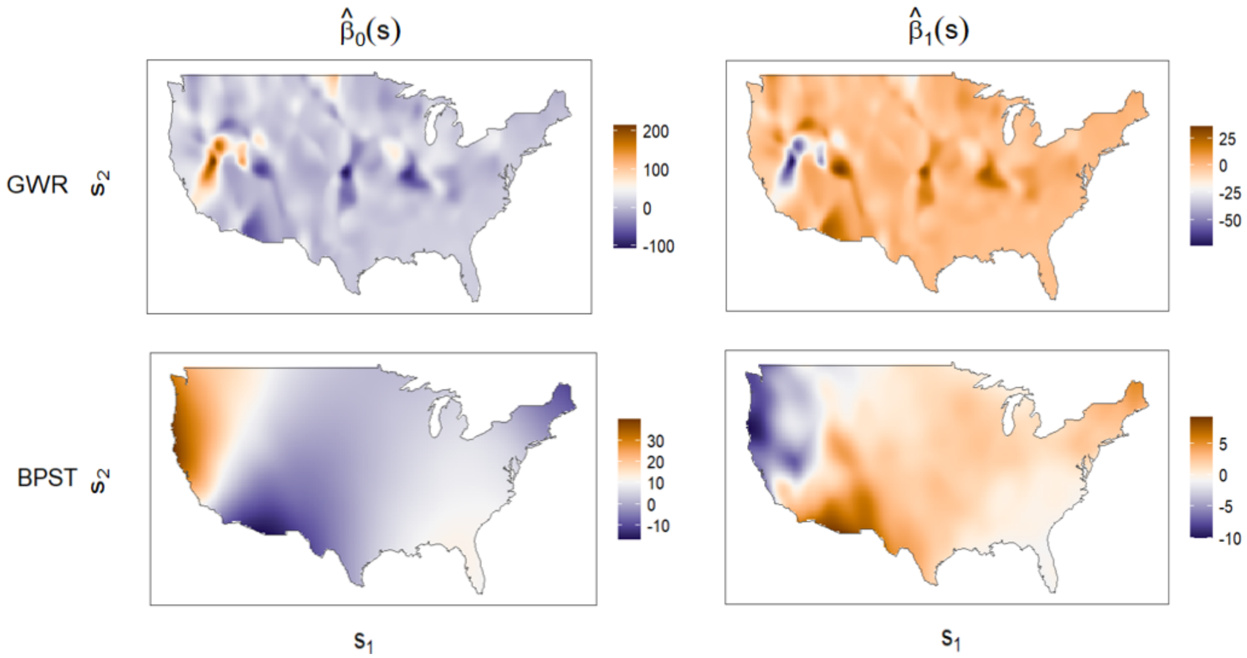


Figure 5.4: Estimated coefficient surfaces for the GWR and BPST methods. The left column displays $\hat{\beta}_0(\mathbf{s})$ for the intercept, and the right column displays $\hat{\beta}_1(\mathbf{s})$ for wind speed. The color scales in each panel are determined by the range of the estimated coefficient values for each individual surface.

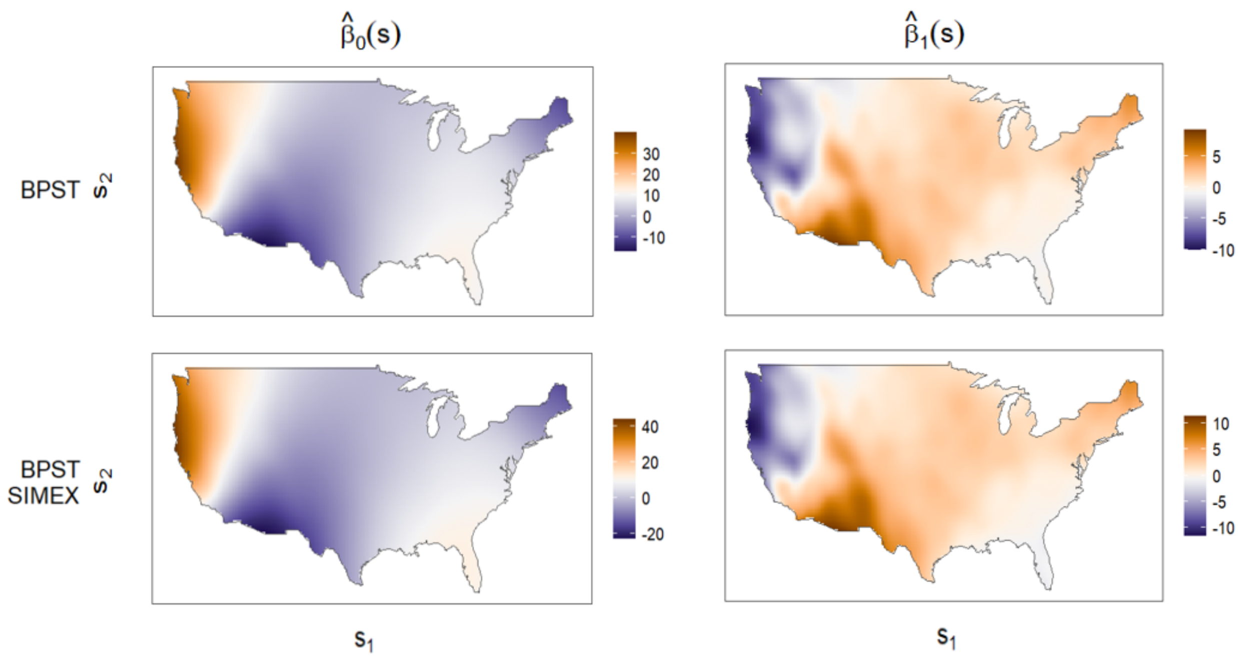


Figure 5.5: Estimated coefficient surfaces for the original BPST fit and the BPST fit with SIMEX adjustment for $PM_{2.5}$ across the contiguous United States. The top row corresponds to the original BPST fit, while the bottom row corresponds to the BPST fit with SIMEX adjustment at the 10% measurement error level. The left column displays $\hat{\beta}_0(\mathbf{s})$ for the intercept, and the right column displays $\hat{\beta}_1(\mathbf{s})$ for the wind speed. The color scales in each panel are determined by the range of the estimated coefficient values for each individual surface.

Chapter 6

Conclusion

In this report, prediction performance and coefficient surface estimation were compared for the non-spatial simple linear regression model (SLR) and the spatially varying coefficient models geographically weighted regression (GWR), bivariate spline triangulation (BST), and bivariate penalized spline triangulation (BPST), with and without SIMEX adjustment. The simulation results in Chapter 4, which incorporated measurement error on an irregular horseshoe domain, indicate that BPST is better suited to an irregular domain than GWR. They also show that, without penalization, GWR can outperform BST. With SIMEX adjustment, the models performed better than their non-SIMEX counterparts, suggesting that accounting for measurement error is beneficial when it is present. The real data analysis in Chapter 5 indicated that model fitting for the NOAA dataset across the contiguous United States is not greatly affected by accounting for a small amount of measurement error. BPST continued to outperform GWR on this irregular domain, indicating that it is the more appropriate spatially varying coefficient model for this application. In future studies, datasets with larger assumed amounts of measurement error could be used with the BST/BPST SIMEX adjustment to assess whether the resulting changes are more substantial. Additionally, the models considered here included only a single explanatory variable. Future work could examine the BST/BPST SIMEX adjustments in settings with multiple explanatory variables, including cases in which multivariate measurement error is present or the measurement error structure differs across observations in a way that is not spatially driven. Overall, this report suggests that incorporating the SIMEX adjustment into BST and BPST can improve performance when measurement error is present. When the presence of measurement error is unknown on an irregular spatial domain, the SIMEX adjustment in the BST and BPST models can be used as a sensitivity analysis tool to assess whether the fitted results change substantially under assumed levels of error.

Bibliography

Alexeeff, S. E., Carroll, R. J., and Coull, B. (2016). Spatial measurement error and correction by spatial simex in linear regression models when using predicted air pollution exposures. *Biostatistics*, 17(2):377–389. https://doi.org/10.1093/biostatistics/kxv048.

Alya, N. A., Almaulidiyah, Q., Farouk, B. R., Rantini, D., Ramadan, A., and Othman, F. (2023). Comparison of geographically weighted regression (gwr) and mixed geographically weighted regression (mgwr) models on the poverty levels in central java in 2023. *IAENG International Journal of Applied Mathematics*, 54(12):2746–2757. https://www.proquest.com/scholarly-journals/comparison-geographically-weighted-regression-gwr/docview/3158199886/se-2.

Brunsdon, C., Fotheringham, S., and Charlton, M. (1998). Geographically weighted regression-modelling spatial non-stationarity. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 47(3):431–443. http://www.jstor.org/stable/2988625.

Cook, J. R. and Stefanski., L. A. (1994). Simulation-extrapolation estimation in parametric measurement error models. *Journal of the American Statistical Association*, 89(428):1314–1328. https://doi.org/10.2307/2290994.

Kim, M., Wang, L., and Wang, H. J. (2025). Estimation and inference of quantile spatially varying coefficient models over complicated domains. *Journal of the American Statistical Association*, 120(551):1853–1867. https://doi.org/10.1080/01621459.2025.2480867.

Li, X., Wang, L., Wang, H. J., and the Alzheimer's Disease Neuroimaging Initiative (2021). Sparse learning and structure identification for ultrahigh-dimensional image-on-scalar regression. *Journal of the American Statistical Association*, 116(536):1994–2008. https://doi.org/10.1080/01621459.2020.1753523.

Liu, Y. (2021). Relationships of wind speed and precipitable water vapor with regional pm2.5 based on wrf-chem model. *Natural Resource Modeling*, 34(2):e12306. https://doi.org/10.1111/nrm.12306.

Mu, J., Wang, G., and Wang, L. (2018). Estimation and inference in spatially varying coefficient models. *Environmetrics*, 29(1):e2485. https://doi.org/10.1002/env.2485.

Ramsay, T. (2002). Spline smoothing over difficult regions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 64(2):307–319. https://doi.org/10.1111/1467-9868.00339.

Stefanski, L. A. (2000). Measurement error models. *Journal of the American Statistical Association*, 95(452):1353–1358. https://doi.org/10.2307/2669787.

Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 65(1):95–114. https://doi.org/10.1111/1467-9868.00374.

Wu, D. (2020). Spatially and temporally varying relationships between ecological footprint and influencing factors in china's provinces using geographically weighted regression (gwr). *Journal of Cleaner Production*, 261:121089. https://doi.org/10.1016/j.jclepro.2020.121089.

Yin, S., Zhu, Z., Wang, L., Liu, B., Xie, Y., Wang, G., and Li, Y. (2018). Regional soil erosion assessment based on a sample survey and geostatistics. *Hydrology and Earth System Sciences*, 22(3):1695–1712. https://hess.copernicus.org/articles/22/1695/2018/.

Zhai, K., Al-Wahshi, F., Mahmoud, L., Sebah, M., and Yousef, M. S. (2022). Data mining indicates an association between ambient pm2.5 levels and wind speed in an urban environment (education city, doha, qatar). *QScience Connect*, 2022(1). https://doi.org/10.5339/connect.2022.2.

Zhou, L. and Pan, H. (2014). Smoothing noisy data for irregular regions using penalized bivariate splines on triangulations. *Computational Statis-*

tics, 29(1-2):263–281. https://www.proquest.com/scholarly-journals/smoothing-noisy-data-irregular-regions-using/docview/1498272148/se-2.

Appendix A

SIMEX Algorithms for BST and BPST

A.1 BST SIMEX Algorithm

1. For each $\lambda \in \Lambda$, define

$$\mathbf{W}^{(\lambda)} = \mathbf{W} + \sqrt{\lambda}\,\mathbf{U}^{*}.$$

2. For each $\lambda \in \Lambda$:

   a. For $b = 1, \ldots, B$:

      (i) Apply Step 1 with an independent draw $\mathbf{U}^{*(b)}$ to obtain

$$\mathbf{W}^{(\lambda,b)} = \mathbf{W} + \sqrt{\lambda}\,\mathbf{U}^{*(b)}.$$

      (ii) Fit the BST model using $\mathbf{y}$ and $\mathbf{W}^{(\lambda,b)}$ to obtain $\{\hat{\boldsymbol{\theta}}_j^{(\lambda,b)}\}_{j=0}^{p}$, where

$$\{\hat{\boldsymbol{\theta}}_j^{(\lambda,b)}\}_{j=0}^{p} = \arg\min_{\{\boldsymbol{\theta}_j\}_{j=0}^{p}} \sum_{i=1}^{n} \left\{ y_i - \sum_{j=0}^{p} W_{ij}^{(\lambda,b)}\, \mathbf{B}_d(\mathbf{s}_i)'\mathbf{Q}_2\boldsymbol{\theta}_j \right\}^2,$$

      where $W_{ij}^{(\lambda,b)}$ denotes the (i,j) entry of $\mathbf{W}^{(\lambda,b)}$. Record the transformed coefficients

$$\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_j^{(\lambda,b)} = \mathbf{Q}_2\,\hat{\boldsymbol{\theta}}_j^{(\lambda,b)}, \qquad j = 0, 1, \ldots, p.$$

b. After completing $b = 1, \ldots, B$, compute the Monte Carlo averages

$$\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_j(\lambda) = \frac{1}{B}\sum_{b=1}^{B} \widehat{\mathbf{Q}_2\boldsymbol{\theta}}_j^{(\lambda,b)}, \qquad j = 0, 1, \ldots, p.$$

3. For each $j = 0, 1, \ldots, p$:

a. Fit an extrapolation function with λ as the input and the Monte Carlo average estimates as the output. Specifically, fit $g_j(\lambda)$ to $\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_j(\lambda)$ by applying the extrapolation componentwise to the entries of $\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_j(\lambda)$.

b. Evaluate the fitted functions at $\lambda = -1$ to obtain the SIMEX estimates:

$$\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{j,\mathrm{SIMEX}} = g_j(-1), \qquad j = 0, 1, \ldots, p.$$

c. Output:

$$\widehat{\mathbf{Q}_2\boldsymbol{\Theta}}_{\mathrm{SIMEX}} = \left[\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{0,\mathrm{SIMEX}}\ \widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{1,\mathrm{SIMEX}}\ \cdots\ \widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{p,\mathrm{SIMEX}}\right].$$

A.2 BPST SIMEX Algorithm

Note: The symbol λ denotes the SIMEX simulation level throughout this appendix. To avoid notational conflict with the BPST smoothing parameter, the BPST penalty parameter is written as λ^{pen} here.

1. For each $\lambda \in \Lambda$, define

$$\mathbf{W}^{(\lambda)} = \mathbf{W} + \sqrt{\lambda}\,\mathbf{U}^*.$$

2. For each $\lambda \in \Lambda$:

a. For $b = 1, \ldots, B$:

(i) Apply Step 1 with an independent draw $\mathbf{U}^{*(b)}$ to obtain

$$\mathbf{W}^{(\lambda,b)} = \mathbf{W} + \sqrt{\lambda}\,\mathbf{U}^{*(b)}.$$

(ii) Fit the BPST model using $\mathbf{y}$ and $\mathbf{W}^{(\lambda,b)}$ to obtain $\{\hat{\boldsymbol{\theta}}_j^{(\lambda,b)}\}_{j=0}^p$, where

$$\{\hat{\boldsymbol{\theta}}_j^{(\lambda,b)}\}_{j=0}^p = \arg\min_{\{\boldsymbol{\theta}_j\}_{j=0}^p} \left[\sum_{i=1}^n \left\{ y_i - \sum_{j=0}^p W_{ij}^{(\lambda,b)}\, \mathbf{B}_d(\mathbf{s}_i)' \mathbf{Q}_2 \boldsymbol{\theta}_j \right\}^2 + \sum_{j=0}^p \lambda^{\text{pen}}\, \boldsymbol{\theta}_j' \mathbf{Q}_2' \mathbf{P} \mathbf{Q}_2 \boldsymbol{\theta}_j \right],$$

where $W_{ij}^{(\lambda,b)}$ denotes the (i,j) entry of $\mathbf{W}^{(\lambda,b)}$. Record the transformed coefficients

$$\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_j^{(\lambda,b)} = \mathbf{Q}_2\, \hat{\boldsymbol{\theta}}_j^{(\lambda,b)}, \qquad j = 0, 1, \ldots, p.$$

b. After completing $b = 1, \ldots, B$, compute the Monte Carlo averages

$$\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_j(\lambda) = \frac{1}{B} \sum_{b=1}^B \widehat{\mathbf{Q}_2\boldsymbol{\theta}}_j^{(\lambda,b)}, \qquad j = 0, 1, \ldots, p.$$

3. For each $j = 0, 1, \ldots, p$:

a. Fit an extrapolation function with λ as the input and the Monte Carlo average estimates as the output. Specifically, fit $g_j(\lambda)$ to $\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_j(\lambda)$ by applying the extrapolation componentwise to the entries of $\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_j(\lambda)$.

b. Evaluate the fitted functions at $\lambda = -1$ to obtain the SIMEX estimates:

$$\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{j,\text{SIMEX}} = g_j(-1), \qquad j = 0, 1, \ldots, p.$$

c. Output:

$$\widehat{\mathbf{Q}_2\boldsymbol{\Theta}}_{\text{SIMEX}} = \left[\widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{0,\text{SIMEX}} \ \widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{1,\text{SIMEX}} \ \cdots \ \widehat{\mathbf{Q}_2\boldsymbol{\theta}}_{p,\text{SIMEX}}\right].$$