UNIVERSITY OF CALIFORNIA
RIVERSIDE

Topics in Prediction for Spatially Varying Functional Data

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Applied Statistics

by

Nir Voloshin

December 2024

Dissertation Committee:

Dr. Yehua Li, Chairperson
Dr. Esra Kurum
Dr. Jun Li
Dr. Weixin Yao

The Dissertation of Nir Voloshin is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I am deeply grateful to my advisor, Dr. Yehua Li, for his guidance, advice, and patience throughout my research. This dissertation would not have been possible without your mentorship, and I sincerely appreciate the time and effort you have invested in me.

I would also like to thank my dissertation and oral exam committee members, Dr. Weixin Yao, Dr. Esra Kurum, and Dr. Jun Li, for their insightful questions, unwavering support, and valuable suggestions. To my teachers and mentors, I extend my heartfelt gratitude for inspiring me to pursue research in Statistics.

Lastly, I want to express my deepest appreciation to my friends and family for their steadfast support and unconditional love. To my friends, thank you for keeping me motivated, especially during the most challenging moments. To my family, your encouragement means the world to me, and I am forever indebted to you. In particular, I want to thank my dad for showing me the beauty of mathematics, which ultimately led me to the field of Statistics.

This dissertation is dedicated to my parents, Moshe and Anat. I wouldn't be where I am today without all your support throughout the years.

ABSTRACT OF THE DISSERTATION

Topics in Prediction for Spatially Varying Functional Data

by

Nir Voloshin

Doctor of Philosophy, Graduate Program in Applied Statistics
University of California, Riverside, December 2024
Dr. Yehua Li, Chairperson

The rise in technological developments in collecting data has allowed for variables to be continuously measured over time and space. This type of data can be characterized as multivariate spatially varying functional data. The field of Functional Data Analysis (FDA) consists of different types of models for functional data. Many existing functional regression models don't consider the spatial component which could be detrimental to prediction since there is the possibility of location-specific effects. The main emphasis of this dissertation is the development of a novel two-step procedure for prediction in Generalized Spatially Varying Functional Models (GSVFM).

Functional data are inherently infinite-dimensional because they represent continuous functions. Since the GSVFM suffers from the curse of dimensionality, functional models can not be estimated directly. To address both the infinite-dimensionality and the spatial varying components of the data, a novel two-step procedure is introduced. The first step of the procedure is to reduce the dimension of the GSVFM through the method of Functional Principal Components Analysis (FPCA). This reduces the GSVFM to a Generalized Spatial Varying Coefficient Model (GSVCM) which is the second step in the procedure. The GSVCM considers the spatial locations in the data. The proposed two-step procedure is able to capture location-specific effects that previous functional regression models can't.

This research is motivated by a crop-yield prediction application in agriculture. The agriculture data is collected at the county-level from five Midwest states, Kansas, Iowa, Illinois, Indiana and Missouri. For each county, we observe daily minimum and maximum temperature time series data. The temperature time series data can be viewed as functions, where the temperature is indexed by the day. Since the temperature data varies across the Midwest counties, this represents the multivariate spatially varying functional data. The precipitation, irrigated land and crop-yield are collected at the county level. The goal is to apply the GSVFM to predict the spatially varying crop yield through the scalar predictor variables and the multivariate spatially varying functional data. Existing functional models are used to compare performance with the GSVFM.

The dissertation consists of two projects that use the novel two-step procedure to estimate the GSVFM and the Spatially Varying Functional Quantile Model (SVFQM). The first project aims at predicting the conditional mean and the second project extends the GSVFM to the SVFQM that predicts the conditional quantile. This research addresses the current gap in functional models that do not consider the spatial component.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Functional Data Analysis

Functional Data Analysis (FDA) is a branch of Statistics that consists of methods and theory for functional data or otherwise known as curves or surfaces. Functional data are observed as functions that vary over a continuum such as time or space where each observation results from an underlying stochastic process. Intrinsically, functional data are infinite dimensional since they are defined over a continuum. While this can cause challenges for theory and computation, the high dimensionality is a rich source of information. Although functional data are treated as infinite dimension random functions, in practice the "functions" are discretely observed. Similarly to any type of multivariate data, the analysis of functional data can be performed using multivariate statistical methods by treating each discrete data point as a variable in the model. This leads to a model with potentially hundreds of predictor variables which can result in overfitting. In addition to overfitting, treating the data as discrete ignores the true underlying function for each observation. This leaves out a lot of information that can be gained from treating the data as functions.

It is impossible to observe the true function because of the infinite domain. The data that's available to us is a collection of discrete data points collected at finite times or locations. This data can be sparse or dense which is important for the data to be treated correctly. To treat the data as functions, the underlying function is estimated by using the basis functions approach. There are many basis function methods such as polynomial splines, truncated-power splines, B-splines, and Fourier series. Each method has its own advantages depending on the application but some methods have similar performance. This gives a representation of the function that can be used in many types of statistical analyses. The use of basis functions results in a smooth function that is estimated from the discrete data. Since these smooth functions are continuous, a functional model can't be estimated directly and a dimension reduction method must be used. There are different approaches to dimension reduction but the most common is Functional Principal Components Analysis (FPCA) which is used to find the principal components that contribute the most variation. The general approach is to use the highest order principal components as predictor variables in a functional model. There are many different types of functional models that are extensions of multivariate models. Some of the most common models are the functional linear regression model, the generalized functional linear model and the functional additive model. Although these models have been shown to perform well for non-spatial functional data, they are not necessarily the best models for spatially varying functional data.

With the improvement of data collection technology, functional data has become more widely available and offers a lot of opportunities for gaining more insight from the data. Since there are many applications where we know that the sampled data result from some underlying function, it is advantageous to use FDA methods as opposed to classical mulitvariate statistical methods.

1.1.1 Examples of Functional Data

Weather data, such as daily temperature or daily precipitation throughout the year, is an example of a set of variables that change with respect to time and space (Wong et al., 2018). Because weather data has an impact on many aspects of our lives, it is of interest to study its patterns. It is unlikely to extract meaningful insights from the weather data without the spatial information, as the data is treated as if it all originates from the same location.

The frequency of car crashes on a highway changes with respect to location and can change over time (Kim and Wang, 2021). The frequency of car crashes can be treated as either spatial data or spatiotemporal data. With the inclusion of other variables, this data has multiple potential uses such as predicting the number of accidents at a specific location and time of day.

Stock prices fluctuate multiple times throughout the day. For very large companies, their shares can be traded up to a thousand times per second. The stock price data from each day is treated as a function (Kokoszka and Reimherr, 2017). This kind of data can be used to discover trends over a period of time and make predictions of the stock price at a given time.

An example of dense functional data is the concentration of nitrogen dioxide in a certain location (Horváth and Kokoszka, 2012). The concentration of nitrogen dioxide is measured at different times throughout the day. This type of data can be used to detect abnormal levels of nitrogen dioxide pollutants.

1.1.2 Representation of Functional Data

When working with functional models, the functions are assumed to be smooth, continuous functions. In practice, functional data are discretely observed where each discrete function x is defined as $x = \{x(t_1), x(t_2), \ldots, x(t_n)\}$ on a grid $(t_1, t_2 \ldots, t_n)$. The most popular method of converting discrete functions to smooth functions is through basis functions. The basis functions approach is to use a set of known functions ϕ_k that satisfy certain mathematical properties. These basis functions are mathematically independent from each other and are able to represent any function reasonably well by using a linear combination of a sufficiently large number K of the basis functions.

One of the most familiar basis functions is the set of monomials which are used to construct power series,

$$1, t, t^2, t^3, \ldots, t^k, \ldots$$

A more complicated set of basis functions that is very popular is the Fourier series system,

$$1, sin(\omega t), cos(\omega t), sin(2\omega t), cos(2\omega t), sin(3\omega t), cos(3\omega t), \ldots, sin(k\omega t), cos(k\omega t), \ldots,$$

where ω represents the period of the sine or cosine function. For a given point t, $x(t)$ can be represented by a linear expansion of a finite number of K basis functions $\{\phi_1, \phi_2, \ldots, \phi_K\}$ where

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t).$$

Letting $\boldsymbol{c} = (c_1, \ldots, c_k)'$ and $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_K)'$, $x(t)$ can be expressed in matrix notation as

$$x(t) = \boldsymbol{c}'\boldsymbol{\phi} = \boldsymbol{\phi}'\boldsymbol{c}.$$

Using basis functions allows us to approximate a function by K basis functions. The choice of basis functions should be chosen such that the functions have similar characteristics to the functions being estimated. The basis function approach is able to capture the rich source of information that functional data can provide.

1.2 Spatial Statistics

Spatial Statistics is a branch of Statistics that consists of methods and theory for data that are collected at different locations in space. More specifically, a variable is measured at different locations on a two-dimensional grid where, for example, the x and y axis represent the longitude and latitude, respectively. There are many different types of spatial data such as weather, population, and agriculture. Spatial data is an inherent part of our world since many variables change over space. Even though the value of a variable changes with respect to location, clearly there are some similarities between locations within some radius. As a result, there is some dependency among spatial data. Similar to the usual statistics framework, if this dependency is not taken into account, then the results will be inaccurate.

Spatial data can be viewed as a type of functional data since it represents a variable continuously changing over a continuum which is space in this case. The true underlying function in this case would be a bivariate function which is a surface over the two-dimensional spatial domain. Therefore, FDA methods can be applied to spatial data.

In some applications, there is data which vary over both time and space. Such data is considered spatiotemporal and can be viewed as resulting from an underlying function of both time and space (i.e. $f(t, \boldsymbol{s})$). This type of function is considered a random field which is a generalization of a stochastic process in the space $\mathbb{R}^n$ with $n = 3$ (Schabenberger and Gotway, 2005). This type of data provides a very rich source of information about different patterns but also has many challenges.

1.2.1 Foundations of Spatial Statistics

This section covers the foundations of spatial statistics and more detail can be found in Schabenberger and Gotway (2005). In spatial statistics, we define a collection of random variables over a spatial domain in d dimensions as a spatial process which is denoted as

$$\{Z(\boldsymbol{s}) : \boldsymbol{s} \in D \subset \mathbb{R}^d\},$$

where Z is the variable of interest. The spatial domain is of dimension $d \geq 2$ but generally we work with $d = 2$, the two-dimensional domain where $\boldsymbol{s} = (s_1, s_2)$ are the Cartesian coordinates. The variable $Z(\boldsymbol{s})$ can be observed everywhere within the spatial domain D.

A common occurrence in spatial data is spatial autocorrelation which refers to correlation between $Z(\boldsymbol{s}_i)$ and $Z(\boldsymbol{s}_j)$. To describe the autocorrelation function for a spatial process, consider a (weakly) stationary time series $Z(t_1), \ldots, Z(t_n)$ with $E\{Z(t_i)\} = 0$ and $Var\{Z(t_i)\} = \sigma^2, i = 1, \ldots, n$. The covariance function of the series between $Z(t_i)$ and $Z(t_j)$ is given by

$$Cov\{Z(t_i), Z(t_j)\} = E\{Z(t_i)Z(t_j)\} = C(t_j - t_i)$$

and the autocorrelation function is

$$R(t_j - t_i) = \frac{Cov\{Z(t_i), Z(t_j)\}}{\sqrt{Var\{Z(t_i)\}Var\{Z(t_j)\}}} = \frac{C(t_j - t_i)}{C(0)}.$$

In spatial data, autocorrelation decreases with increasing distance. The covariance function of a spatial process for spatial lag h is defined as

$$C(\boldsymbol{s}, \boldsymbol{h}) = Cov\{Z(\boldsymbol{s}), Z(\boldsymbol{s}+\boldsymbol{h})\} = E\big[\big\{Z(\boldsymbol{s}) - \mu(\boldsymbol{s})\big\}\big\{Z(\boldsymbol{s}+\boldsymbol{h}) - \mu(\boldsymbol{s}+\boldsymbol{h})\big\}\big]$$

and the spatial autocorrelation function is

$$R(\boldsymbol{s}, \boldsymbol{h}) = \frac{C(\boldsymbol{h})}{\sqrt{Var[Z(\boldsymbol{s})]Var[Z(\boldsymbol{s}+\boldsymbol{h})]}}.$$

An alternative to the autocorrelation function that originated in the geostatistical community is the semivariogram which is defined as

$$\begin{aligned}\gamma(\boldsymbol{s}, \boldsymbol{h}) &= \frac{1}{2}Var\{Z(\boldsymbol{s}) - Z(\boldsymbol{s}+\boldsymbol{h})\} \\ &= \frac{1}{2}\Big[Var\{Z(\boldsymbol{s})\} + Var\{Z(\boldsymbol{s}+\boldsymbol{h})\} - 2Cov\{Z(\boldsymbol{s}), Z(\boldsymbol{s}+\boldsymbol{h})\}\Big].\end{aligned}$$

Although the semivariogram has its benefits, in the rest of the discussion we will be using the covariance function $C(\boldsymbol{h})$. The modeling of the autocorrelation of spatial processes can occur in all directions which presents some challenges in modeling.

1.2.2 Random Fields

To introduce random fields, the definition of a stochastic process is defined in Definition 1.

Definition 1 *A stochastic process is a family or collection of random variables, the members of which can be identified or located (indexed) according to some metric.*

A random field is a stochastic process with an index set of dimension $d > 1$. Since a spatial process is a stochastic process with an index set of dimension $d = 2$, it is a random field. It is important to treat the spatial data as a random field because of the random mechanism that generates the data. Consider a random experiment ω where the outcome is $Z(\boldsymbol{s})$, the value of Z at location $\boldsymbol{s}$. A surface $Z(., \omega)$ is generated by a particular realization of ω. The key difference in this type of setting, is that if we collect n spatial observations, this does not translate to a sample of size n. Quite the contrary, these spatial observations represent a single realization of a random experiment, a sample size of one. It might seem that we can't make any progress with a sample size of one, but it is possible if the random field has certain stationary properties.

Definition 2 *A random field $\{Z(\boldsymbol{s}) : s \in D \subset \mathbb{R}^d\}$ is called a strict (or strong) stationary field if the spatial distribution is invariant under translation of coordinates*

$$\begin{aligned}&P\Big\{Z(\boldsymbol{s}_1) < z_1, Z(\boldsymbol{s}_2) < z_2, \ldots, Z(\boldsymbol{s}_k) < z_k\Big\}\\ &= P\Big\{Z(\boldsymbol{s}_1 + \boldsymbol{h}) < z_1, Z(\boldsymbol{s}_2 + \boldsymbol{h}) < z_2, \ldots, Z(\boldsymbol{s}_k + \boldsymbol{h}) < z_k\Big\},\end{aligned}$$

for all k and $\boldsymbol{h}$.

As the name suggests, this is the strongest type of stationarity. A weaker type of stationarity is second-order stationarity which is defined as in Definition 3.

Definition 3 *Second-order (or weak) stationarity of a random function implies that $E\{Z(\boldsymbol{s})\} = \mu$ and $Cov\{Z(\boldsymbol{s}), Z(\boldsymbol{s} + \boldsymbol{h})\} = C(\boldsymbol{h})$.*

A second-order random field is the spatial equivalent of a random sample in classical statistics. When the lag $\boldsymbol{h} = \mathbf{0}$, then the covariance function $C(\boldsymbol{h})$ is

$$Cov\{Z(\boldsymbol{s}), Z(\boldsymbol{s} + \mathbf{0})\} = Var\{Z(\boldsymbol{s}\} = C(\mathbf{0}),$$

which means that the variability for a second-order random field is constant across space. It can be shown that a strictly stationary random field is also a second-order stationary random field. To define different types of covariance functions for random fields, we first need to consider what conditions a *valid* covariance function must satisfy.

Definition 4 *For a covariance function $C(\boldsymbol{s}_i - \boldsymbol{s}_j)$ of a second-order stationary spatial random field in $\mathbb{R}^d$ to be valid, C must satisfy the positive-definiteness condition*

$$\sum_{i=1}^{k} \sum_{j=1}^{k} a_i a_j C(\boldsymbol{s}_i - \boldsymbol{s}_j) \geq 0, \tag{1.1}$$

for any set of locations and real numbers.

In some situations, a random field won't satisfy strict or second-order stationary but it's possible that the increments $Z(\boldsymbol{s}) - Z(\boldsymbol{s} + \boldsymbol{h})$ might be second-order stationary. A random field that satisfies this property is called intrinsic stationarity which is defined in Definition 5.

Definition 5 *The random field $\{Z(\boldsymbol{s}) : s \in D \subset \mathbb{R}^d\}$ is said to be intrinsically stationary if*

$$E\{Z(\boldsymbol{s})\} = \mu \textit{ and } \frac{1}{2} Var\{Z(\boldsymbol{s}) - Z(\boldsymbol{s} + \boldsymbol{h})\} = \gamma(\boldsymbol{h}),$$

where $\gamma(\boldsymbol{h})$ is the semivariogram of the random field.

If a random field is second-order stationary, then it is also intrinsically stationary. As we saw in Section 1.2.1, the covariance between two points can depend on the *direction* within the spatial domain. For example, suppose that $Z(\boldsymbol{s}_1)$ is north of $Z(\boldsymbol{s}_2)$ and $Z(\boldsymbol{s}_3)$ is west from $Z(\boldsymbol{s}_1)$, where $Z(\boldsymbol{s}_2)$ and $Z(\boldsymbol{s}_3)$ are the same distance away from $Z(\boldsymbol{s}_1)$, respectively. The covariance between $Z(\boldsymbol{s}_1)$ and $Z(\boldsymbol{s}_2)$ could be different than the covariance between $Z(\boldsymbol{s}_1)$ and $Z(\boldsymbol{s}_3)$ which is what is meant by direction dependence. In some instances, the covariance between two spatial points does not depend on the direction but instead depends only on the absolute distance between the points. When this is the case, the random field posses a property called isotropy which can be defined as follows

Definition 6 *A covariance function $C(\boldsymbol{h})$ of a second-order stationary random field is isotropic if the covariance function depends only on the absolute distance between points.*

When we have a second-order stationary isotropic random field, the covariance function $C(\boldsymbol{h})$ can be written as follows

$$C(\boldsymbol{h}) = C^*(\|\boldsymbol{h}\|),$$

where $\|\boldsymbol{h}\|$ is the Euclidean norm of the lag vector

$$\|(\boldsymbol{s}+\boldsymbol{h}) - \boldsymbol{s})\| = \|\boldsymbol{h}\| = \sqrt{h_1^2 + h_2^2}.$$

A commonly used random field is the Gaussian random field which has many useful properties can be defined in Definition 7.

Definition 7 *A random field $\{Z(\boldsymbol{s}) : s \in D \subset \mathbb{R}^d\}$ is a Gaussian random field if the cumulative distribution function*

$$P\Big\{Z(\boldsymbol{s}_1) < z_1, Z(\boldsymbol{s}_2) < z_2, \ldots, Z(\boldsymbol{s}_k) < z_k\Big\}$$

is that of a k-variate Gaussian random variable for all k.

1.3 Functional Principal Components Analysis (FPCA)

Whenever we have a dataset with multiple variables, it is useful to reduce the dimensionality of the dataset for many reasons such as model overfitting and computation. Dimension reduction can also be very beneficial for exploratory data analysis since visualizing a dataset with multiple variables is often difficult. To perform dimension reduction, one of the most popular methods is Principal Components Analysis (PCA). PCA is an established method in the field of multivariate statistics where the basic idea is to find the most important modes of variation in the variables (Ramsay and Silverman, 2005). Functional Principal Components Analysis (FPCA) is an extension of PCA in $\mathbb{R}^n$ to functions in Hilbert spaces such as the L^2 space of square-integrable functions. To develop the method of FPCA, it is useful to discuss multivariate PCA to gain some intuition for FPCA.

1.3.1 Multivariate Principal Components Analysis

This section provides an overview of multivariate PCA. More detail can be found in Ramsay and Silverman (2005). Because of the high-dimensionaltity in multivariate statistics, it is common to look at linear combinations of variable values where

$$\xi_i = \sum_{j=1}^{p} \psi_j x_{ij},\ i = 1, \ldots, N,$$

where ψ_j is a weight applied to the observed values x_{ij} of the jth variables. Defining $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_p)^T$ and $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^T$ we get that

$$\xi_i = \boldsymbol{\psi}^T \boldsymbol{x}_i,\ i = 1, \ldots, N.$$

PCA can be defined in the following steps:

1. We want to solve for the weight vector $\boldsymbol{\psi}_1 = (\psi_{11}, \ldots, \psi_{p1})^T$ such that the linear combination values

   $$\xi_{i1} = \sum_j \psi_{j1} x_{ij} = \boldsymbol{\psi}_1' \boldsymbol{x}_i$$

   maximizes the mean square $N^{-1} \sum_i \xi_{i1}^2$ subject to the constraint

   $$\sum_j \psi_{j1}^2 = \|\boldsymbol{\psi}_1\|^2 = 1.$$

   This can be rewritten as a maximization problem

   $$\underset{\|\boldsymbol{\psi}_1\|^2=1}{\operatorname{argmax}} N^{-1} \sum_i \xi_{i1}^2. \tag{1.2}$$

2. For the second and subsequent steps, the goal is the same as in 1.2 but for the mth step, $\xi_{im} = \boldsymbol{\psi}_m' \boldsymbol{x}_i$ must satisfy $(m-1)$ additional constraints

   $$\sum_j \psi_{jk} \psi_{jm} = \boldsymbol{\psi}_k' \boldsymbol{\psi}_m = 0,\ k < m.$$

   The mth step in the PCA procedure can be rewritten as

   $$\underset{\substack{\|\boldsymbol{\psi}_1\|^2=1 \\ \boldsymbol{\xi}_k' \boldsymbol{\xi}_m = 0,\ \forall k < m}}{\operatorname{argmax}} \quad N^{-1} \sum_i \xi_{im}^2.$$

Maximizing the mean square in the first step is essential because we want to identify the strongest and most important mode of variation in the variables. The unit sum of squares constraint is necessary for the problem to be well-defined since without it, the mean squares of ξ_{im} can be made arbitrarily large. The $(m-1)$ additional constraints on the mth step of the procedure are necessary so that the mth weight vector $\boldsymbol{\psi}_m$ is orthogonal to the $(m-1)$ previous weight vectors. We want to impose orthogonality since at each step, we only want a new weight vector that is indicative of some new information. Since there is finite variation within a dataset, at each step, the variation will decrease. The most common situation is that the first few principal components will explain the majority of the variation in the data. Now the chosen principal components can be used to do some data exploration or as variables in our model.

The PCA procedure described above is one of the ways to define PCA but another common characterization of PCA that is used is in terms of the eigendecomposition of the covariance function or operator. To define this PCA procedure, let the $N \times p$ matrix $\boldsymbol{X}$ contain the values x_{ij} and the vector $\boldsymbol{\psi}$ the weights for a linear combination. Using $\boldsymbol{X}$ and $\boldsymbol{\psi}$, we can rewrite the mean square error criterion for finding the first principal component as

$$\max_{\boldsymbol{\psi}'\boldsymbol{\psi}=1} N^{-1}\boldsymbol{\psi}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\psi} \tag{1.3}$$

since the PC scores are defined as $\boldsymbol{\xi} = \boldsymbol{X}\boldsymbol{\psi}$. Letting $\boldsymbol{V} = N^{-1}\boldsymbol{X}'\boldsymbol{X}$ be the sample covariance matrix, equation (1.3) can now be written as

$$\max_{\boldsymbol{\psi}'\boldsymbol{\psi}=1} N^{-1}\boldsymbol{\psi}'\boldsymbol{V}\boldsymbol{\psi}.$$

It can be shown that the solution to this maximization problem is found by finding the eigenvector with the largest eigenvalue λ of the eigenequation

$$\boldsymbol{V}\boldsymbol{\psi} = \lambda\boldsymbol{\psi}.$$

This representation of PCA simplifies the problem since now our goal is to find the eigendecomposition of the sample covariance matrix $\boldsymbol{V}$. This representation will be used throughout the rest of this section.

1.3.2 Univariate Functional Principal Components Analysis

The following section extends the method of multivariate PCA described in section 1.3.1 to univariate functional data. More detail can be found in Ramsay and Silverman (2005). To begin the discussion of FPCA, we first need to define an inner product between two functions ψ and x where

$$\langle \psi, x \rangle = \int \psi(s)x(s)ds. \tag{1.4}$$

Using equation 1.4, the principal component scores are now defined as

$$\xi_i = \int \psi(s)x_i(s)ds.$$

In the first step of FPCA, the goal is to find the weight function $\psi_1(s)$ that maximizes

$$\max_{\int \psi_1(s)^2=1} \sum_i \xi_{i1}^2 = \max_{\int \psi_1(s)^2=1} N^{-1} \sum_i \left\{ \int \psi_1(s)x_i(s)ds \right\}^2.$$

Now to define FPCA in terms of the eigendecomposition of its sample covariance function, let $x_1, \ldots, x_N$ be a sample of a random function x and then we have

$$v(s,t) = N^{-1} \sum_{i=1}^{N} x_i(s)x_i(t).$$

Each of the eigenfunctions $\psi(s)$ satisfy the following equation

$$\int v(s,t)\psi(t)dt = \lambda\psi(s) \tag{1.5}$$

for an eigenvalue λ. The left side of the equation (1.5) is called an integral transform V of the weight function ψ defined by

$$V = \int v(.,t)\psi(t)dt.$$

In this context, V is called the covariance operator. Therefore, we can express the eigenequation as

$$V\psi = \lambda\psi, \tag{1.6}$$

where now ψ is an eigenfunction rather than an eigenvector. Because of the infinite dimensionality of equation 1.6, it is difficult to solve for the eigendecomposition of the covariance operator V directly. To reduce the dimension of the eigenequation, we can use a basis expansion for each functional observation x_i. Suppose that each function has basis expansion

$$x_i(t) = \sum_{k=1}^{K} c_{ik}\phi_k(t),$$

where K represents the number of basis functions. To write this more compactly, define the vector-valued function $\boldsymbol{x}$ to have components $x_1, \ldots, x_N$ and the vector-valued function $\boldsymbol{\phi}$ to have components $\phi_1, \ldots, \phi_K$. Then the simultaneous expansion of all N curves is

$$\boldsymbol{x} = \boldsymbol{C}\boldsymbol{\phi}, \tag{1.7}$$

where the coefficient matrix $\boldsymbol{C}$ is $N \times K$. Using equation (1.7), the covariance function can be written as

$$\begin{aligned} v(s,t) &= N^{-1} \sum_{i=1}^{N} x_i(s) x_i(t) \\ &= N^{-1} \boldsymbol{\phi}(s)' \boldsymbol{C}(s)' \boldsymbol{C}(t) \boldsymbol{\phi}(t). \end{aligned}$$

For any choice of bases, we define the order K symmetric matrix $\boldsymbol{W}$

$$w_{k_1,k_2} = \int \phi_{k_1} \phi_{k_2} \text{ or } \boldsymbol{W} = \int \boldsymbol{\phi}' \boldsymbol{\phi}.$$

Our knowledge of $\boldsymbol{W}$ depends on the choice of bases. If, for example, we were to use the orthonormal Fourier series, $\boldsymbol{W} = \boldsymbol{I}$, the order K identity matrix. Suppose that the eigenfunction ψ has the expansion

$$\psi(s) = \sum_{k=1}^{K} b_k \phi_k(s) = \boldsymbol{\phi}(s)' \boldsymbol{b}.$$

The left hand side of 1.5 now becomes

$$\int v(s,t) \psi(t) dt = \boldsymbol{\phi}(s)' N^{-1} \boldsymbol{C}' \boldsymbol{C} \boldsymbol{W} \boldsymbol{b}.$$

Therefore, the eigenequation can be expressed as

$$\boldsymbol{\phi}(s)' N^{-1} \boldsymbol{C}' \boldsymbol{C} \boldsymbol{W} \boldsymbol{b} = \lambda \boldsymbol{\phi}(s)' \boldsymbol{b}. \tag{1.8}$$

Since 1.8 must hold for all s, this implies the matrix equation

$$N^{-1} \boldsymbol{C}' \boldsymbol{C} \boldsymbol{W} \boldsymbol{b} = \lambda \boldsymbol{b}. \tag{1.9}$$

It can be shown that $\|\psi\|^2 \Rightarrow \boldsymbol{b}' \boldsymbol{W} \boldsymbol{b} = 1$ and that two functions ψ_1 and ψ_2 are orthogonal if and only if $\boldsymbol{b}_1' \boldsymbol{W} \boldsymbol{b}_2 = 0$. To solve for the principal components, define $\boldsymbol{u} = \boldsymbol{W}^{1/2} \boldsymbol{b}$ so that now the eigenequation in (1.9) becomes

$$N^{-1}\boldsymbol{W}^{1/2}\boldsymbol{C}'\boldsymbol{C}\boldsymbol{W}^{1/2}\boldsymbol{u} = \lambda\boldsymbol{u}.$$

Then, we solve for the eigendecomposition of the matrix $\boldsymbol{A} = N^{-1}\boldsymbol{W}^{1/2}\boldsymbol{C}'\boldsymbol{C}\boldsymbol{W}^{1/2}$ and get eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K$ and eigenvalues $\lambda_1, \ldots, \lambda_K$. Using the eigenvectors, we can solve for the basis coefficient vectors $\boldsymbol{b}_i = \boldsymbol{W}^{-1/2}\boldsymbol{u}_i$ and then the eigenfunctions $\psi_1, \ldots, \psi_K$ are defined as

$$\psi_l(s) = \boldsymbol{\phi}(s)'\boldsymbol{b}_l = \boldsymbol{\phi}(s)'\boldsymbol{W}^{-1/2}\boldsymbol{u}_l. \tag{1.10}$$

Since the eigenfunction in 1.10 can't be estimated for all values of s, $\psi_l(s)$ is discretized by n time points. Then the matrix of estimated eigenfunctions $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_K$ can be written as

$$\boldsymbol{\Psi} = \begin{bmatrix} \psi_{11}, & \psi_{21}, & \ldots, & \psi_{K1} \\ \psi_{12}, & \psi_{22}, & \ldots, & \psi_{K2} \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{1n}, & \psi_{2n}, & \ldots, & \psi_{Kn} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\psi}_1, & \boldsymbol{\psi}_2, & \ldots, & \boldsymbol{\psi}_K \end{bmatrix},$$

where ψ_{lt} for $t = 1, \ldots, n$, represents the value of the l'th eigenfunction at a given point t and each discrete eigenfunction $\boldsymbol{\psi_l}$ for n time points is defined as

$$\boldsymbol{\psi}_l = \begin{bmatrix} \psi_{11}, & \psi_{12}, & \ldots, & \psi_{1n} \end{bmatrix}' = \begin{bmatrix} \boldsymbol{\phi}(s_1)'\boldsymbol{b}_l, & \boldsymbol{\phi}(s_2)'\boldsymbol{b}_l, & \ldots, & \boldsymbol{\phi}(s_n)'\boldsymbol{b}_l \end{bmatrix}'.$$

Defining the basis functions matrix $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}(s_1), & \boldsymbol{\phi}(s_2), & \ldots, & \boldsymbol{\phi}(s_n) \end{bmatrix}'$ and the basis coefficient vector $\boldsymbol{b}_l = \begin{bmatrix} b_{l1}, & b_{l2}, & \ldots, & b_{lK} \end{bmatrix}'$ we get

$$\boldsymbol{\psi}_l = \boldsymbol{\Phi}\boldsymbol{b}_l.$$

Now we have that the matrix $\boldsymbol{\Psi}$ can be expressed as

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\psi}_1, & \boldsymbol{\psi}_2, & \ldots, & \boldsymbol{\psi}_K \end{bmatrix} = \boldsymbol{\Phi}\boldsymbol{B}.$$

Using the estimates of the eigenfunctions, we can calculate the principal component scores for each functional observation x_i. In the case where we use orthonormal basis functions and the matrix $\boldsymbol{W} = \boldsymbol{I}$, the lth principal component score of the ith observation can be calculated as follows

$$\xi_{il} = \int \psi_l(t)x_i(t)dt \approx \sum_{k=1}^{K} c_{ik}b_{lk} = \tilde{\xi}_{il}.$$

Then for the ith observation x_i, the vector of principal component scores is

$$\tilde{\boldsymbol{\xi}}_i = \begin{bmatrix} \tilde{\xi}_{i1}, & \tilde{\xi}_{i2}, & \ldots, & \tilde{\xi}_{iK} \end{bmatrix}', \; i = 1, \ldots, N.$$

1.3.3 Multivariate Functional Principal Components Analysis

This section extends univariate FPCA described in Section 1.3.2 to multivariate FPCA. More detail can be found in Ramsay and Silverman (2005). In Section 1.3.2, we focused on FPCA for only one variable of interest. In many applications, we are often interested in studying two or more variables. More specifically, it is of interest to look at the simultaneous variation of both variables when they vary jointly. In this case, it is better to perform multivariate FPCA as opposed to univariate FPCA on each of the variables.

Suppose we observe two random samples of random functions, $x_1, x_2, \ldots, x_N$ and $y_1, y_2, \ldots, y_N$. Define v_{xx} to be the covariance function of x_i, v_{yy} that of y_i and v_{xy} to be the cross-covariance function where $v_{yx}(s,t) = v_{xy}(s,t)'$. The vector $\boldsymbol{\psi} = (\psi^x, \psi^y)$ of weight functions is a principal component where ψ^x denotes the variation in x and ψ^y that in y.

To define multivariate FPCA, we first need to define an inner product between these kinds of functions. The inner product for two functions is defined as

$$\langle \boldsymbol{\psi}_1, \boldsymbol{\psi}_2 \rangle = \int \psi_1^x \psi_2^x + \int \psi_1^y \psi_2^y.$$

To define the principal component scores, we define the random functions to have the same structure as the principal components where $\boldsymbol{F}_i = (x_i, y_i)$ and then ξ_{il} is defined as

$$\xi_{il} = \langle \boldsymbol{\psi}_l, \boldsymbol{F}_i \rangle = \int \psi_l^x x_i + \int \psi_l^y y_i.$$

The covariance operator V is now defined as

$$V = \begin{bmatrix} v_{xx} & v_{xy} \\ v_{yx} & v_{yy} \end{bmatrix}.$$

Then the eigenequations can be written as

$$V\boldsymbol{\psi}_l = \lambda_l \boldsymbol{\psi}_l \Rightarrow \begin{bmatrix} v_{xx} & v_{xy} \\ v_{yx} & v_{yy} \end{bmatrix} \begin{bmatrix} \psi_l^X \\ \psi_l^y \end{bmatrix} = \lambda_l \begin{bmatrix} \psi_l^x \\ \psi_l^y \end{bmatrix},$$

$$v_{xx}\psi_l^x + v_{xy}\psi_l^Y = \lambda_l \psi_l^x,$$

$$v_{yx}\psi_l^x + v_{yy}\psi_l^y = \lambda_l \psi_l^y,$$

$$\int v_{xx}(s,t)\psi_l^x(t)dt + \int v_{xy}(s,t)\psi_l^y(t)dt = \lambda_l \psi_l^x,$$

$$\int v_{yx}(s,t)\psi_l^x(t)dt + \int v_{yy}(s,t)\psi_l^y(t)dt = \lambda_l \psi_l^y.$$

To solve for the eigendecomposition of the covariance operator V, we follow the same procedure as before where we have the following basis expansions of $x_i(t)$ and $y_i(t)$ as follows

$$x_i(t) = \sum_{k=1}^{K} c_{ik}\phi_k(t) = \boldsymbol{\phi}(s)'\boldsymbol{C}'\boldsymbol{C}\boldsymbol{\phi}(t),$$

$$y_i(t) = \sum_{k=1}^{K} d_{ik}\phi_k(t) = \boldsymbol{\phi}(s)'\boldsymbol{D}'\boldsymbol{D}\boldsymbol{\phi}(t),$$

and the estimates of the covariance functions are defined as

$$v_{xx}(s,t) = N^{-1}\sum_{i=1}^{N} x_i(s)x_i(t) = N^{-1}\boldsymbol{\phi}(s)'\boldsymbol{C}'\boldsymbol{C}\boldsymbol{\phi}(s),$$

$$v_{yy}(s,t) = N^{-1}\sum_{i=1}^{N} y_i(s)y_i(t) = N^{-1}\boldsymbol{\phi}(s)'\boldsymbol{D}'\boldsymbol{D}\boldsymbol{\phi}(s),$$

$$v_{xy}(s,t) = N^{-1}\sum_{i=1}^{N} x_i(s)y_i(t) = N^{-1}\boldsymbol{\phi}(s)'\boldsymbol{C}'\boldsymbol{D}\boldsymbol{\phi}(s) = v_{yx}(t,s).$$

For the expansions of the eigenfunctions $\psi_l^x(s)$ and $\psi_l^y(s)$ we have

$$\psi_l^x(s) = \sum_{k=1}^{K} b_{kl}\phi_k(s) = \boldsymbol{\phi}(s)'\boldsymbol{b}_l,$$

$$\psi_l^y(s) = \sum_{k=1}^{K} g_{kl}\phi_k(s) = \boldsymbol{\phi}(s)'\boldsymbol{g}_l.$$

Using the expansions and covariance functions defined above, it can be shown that the eigenequations reduce to

$$N^{-1}\boldsymbol{\phi}(s)'\boldsymbol{C}'\boldsymbol{C}\boldsymbol{W}\boldsymbol{b}_l + N^{-1}\boldsymbol{\phi}(s)'\boldsymbol{C}'\boldsymbol{D}\boldsymbol{W}\boldsymbol{g}_l = \lambda_l\boldsymbol{\phi}(s)\boldsymbol{b}_l, \tag{1.11}$$

$$N^{-1}\boldsymbol{\phi}(s)'\boldsymbol{D}'\boldsymbol{C}\boldsymbol{W}\boldsymbol{b}_l + N^{-1}\boldsymbol{\phi}(s)'\boldsymbol{D}'\boldsymbol{D}\boldsymbol{W}\boldsymbol{g}_l = \lambda_l\boldsymbol{\phi}(s)\boldsymbol{g}_l. \tag{1.12}$$

Since 1.11 and 1.12 must hold for all time points s, this implies the purely matrix equations

$$N^{-1}\boldsymbol{C}'\boldsymbol{C}\boldsymbol{W}\boldsymbol{b}_l + N^{-1}C'\boldsymbol{D}\boldsymbol{W}\boldsymbol{g}_l = \lambda_l\boldsymbol{b}_l,$$

$$N^{-1}\boldsymbol{D}'\boldsymbol{C}\boldsymbol{W}\boldsymbol{b}_l + N^{-1}\boldsymbol{D}'\boldsymbol{D}\boldsymbol{W}\boldsymbol{g}_l = \lambda_l\boldsymbol{g}_l.$$

To enforce the constraint $\|\boldsymbol{\psi}_l\| = 1$ it can be shown that

$$\|\boldsymbol{\psi_l}\| = \int \psi^x_{l1}\psi^x_{l2} + \int \psi^y_{l1}\psi^y_{l2} = \boldsymbol{b}_l' W \boldsymbol{b}_l + \boldsymbol{g}_l' W \boldsymbol{g}_l.$$

Letting $\boldsymbol{u}_l = \boldsymbol{W}^{1/2}\boldsymbol{b}_l$ and $\boldsymbol{q}_l = \boldsymbol{W}^{1/2}\boldsymbol{g}_l$ we get

$$\|\boldsymbol{\psi_l}\| = 1 \Rightarrow \boldsymbol{b}_l' W \boldsymbol{b}_l + \boldsymbol{g}_l' W \boldsymbol{g}_l = 1 \Rightarrow \boldsymbol{u}_l'\boldsymbol{u}_l + \boldsymbol{q}_l'\boldsymbol{q}_l = 1.$$

Using the transformed variables $\boldsymbol{u}$ and $\boldsymbol{q}$, equations (1.12) and (1.13) can now be written as

$$N^{-1}\begin{bmatrix} \boldsymbol{W}^{1/2}\boldsymbol{C}'\boldsymbol{C}\boldsymbol{W}^{1/2} & \boldsymbol{W}^{1/2}\boldsymbol{C}'\boldsymbol{D}\boldsymbol{W}^{1/2} \\ \boldsymbol{W}^{1/2}\boldsymbol{D}'\boldsymbol{C}\boldsymbol{W}^{1/2} & \boldsymbol{W}^{1/2}\boldsymbol{D}'\boldsymbol{D}\boldsymbol{W}^{1/2} \end{bmatrix}\begin{bmatrix} \boldsymbol{u}_l \\ \boldsymbol{q}_l \end{bmatrix} = \lambda_l \begin{bmatrix} \boldsymbol{u}_l \\ \boldsymbol{q}_l \end{bmatrix} \Rightarrow \boldsymbol{F}\begin{bmatrix} \boldsymbol{u}_l \\ \boldsymbol{q}_l \end{bmatrix} = \lambda_l \begin{bmatrix} \boldsymbol{u}_l \\ \boldsymbol{q}_l \end{bmatrix}.$$

For the eigendecomposition of the covariance operator V, we need to solve for the eigendecomposition of $\boldsymbol{F}$. The eigendecomposition of $\boldsymbol{F}$ has $m = 2K$ eigenvalues and eigenvectors. There are K eigenvectors $\boldsymbol{u}_1, \dots, \boldsymbol{u}_K$ and K eigenvectors $\boldsymbol{q}_1, \dots, \boldsymbol{q}_K$. The basis coefficient matrices $\boldsymbol{B}$ and $\boldsymbol{G}$ can be calculated as follows

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1, & \boldsymbol{b}_2, & \dots, & \boldsymbol{b}_K \end{bmatrix} = \boldsymbol{W}^{-1/2}\begin{bmatrix} \boldsymbol{u}_1, & \boldsymbol{u}_2, & \dots, & \boldsymbol{u}_K \end{bmatrix} = \boldsymbol{W}^{-1/2}\boldsymbol{U},$$

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{g}_1, & \boldsymbol{g}_2, & \dots, & \boldsymbol{g}_K \end{bmatrix} = \boldsymbol{W}^{-1/2}\begin{bmatrix} \boldsymbol{q}_1, & \boldsymbol{q}_2, & \dots, & \boldsymbol{q}_K \end{bmatrix} = \boldsymbol{W}^{-1/2}\boldsymbol{Q}.$$

Therefore, each eigenfunction $\boldsymbol{\psi}_l$ for $l = 1, \dots, K$ is defined as

$$\boldsymbol{\psi}_l = \begin{bmatrix} \boldsymbol{\psi}^x_l \\ \boldsymbol{\psi}^y_l \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}'\boldsymbol{b}_l \\ \boldsymbol{\Phi}'\boldsymbol{g}_l \end{bmatrix}.$$

Similarly to section 1.3.2, the eigenfunction $\boldsymbol{\psi}_l$ can't be estimated for all values of s. Therefore, it is represented for discrete values $t = 1, \dots, n$. Then, the matrix of eigenfunctions $\boldsymbol{\Psi}$

is

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\psi}_1, & \boldsymbol{\psi}_2, & \dots, & \boldsymbol{\psi}_K \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}'\boldsymbol{b}_1, & \boldsymbol{\Phi}'\boldsymbol{b}_2, & \dots, & \boldsymbol{\Phi}'\boldsymbol{b}_K \\ \boldsymbol{\Phi}'\boldsymbol{g}_1, & \boldsymbol{\Phi}'\boldsymbol{g}_2, & \dots, & \boldsymbol{\Phi}'\boldsymbol{g}_K \end{bmatrix},$$

where the eigenfunction $\boldsymbol{\psi}_l$ is defined for discrete points $t = 1, \dots, n$. The lth joint principal component score of the ith observation can be calculated as follows

$$\xi_{il} = \int \psi_l^x(t) x_i(t) dt + \int \psi_l^y(t) y_i(t) dt \approx \sum_{k=1}^{K} b_{kl} c_{ik} + \sum_{k=1}^{K} g_{kl} d_{ik} = \tilde{\xi}_{il}.$$

Then for the ith observation $\boldsymbol{F}_i$, the vector of principal component scores vector is

$$\tilde{\boldsymbol{\xi}}_i = \begin{bmatrix} \tilde{\xi}_{i1}, & \tilde{\xi}_{i2}, & \dots, & \tilde{\xi}_{iK} \end{bmatrix}', \ i = 1, \dots, N.$$

1.3.4 Karhunen-Loeve Expansion

The Karhunen-Loeve expansion is a very useful representation of a stochastic process X in terms of its eigendecomposition that facilitates dimension reduction (Wang et al., 2016). This representation aids in the estimation of functional models. Before stating the Karhunen-Loeve expansion Theorem, we first need to state Mercer's Lemma. Let δ denote the Kronecker delta. Then, following Bosq (2000) the Mercer Lemma is defined in Lemma 8.

Lemma 8 *(Mercer Lemma) Let v be a covariance function continuous over $[0,1]^2$. Then there exists a sequence (ψ_k) of continuous functions and a decreasing sequence (λ_k) of positive numbers such that*

$$\int_0^1 v(s,t)\psi_k(s)ds = \lambda_k\psi_k, \;\; t \in [0,1], \;\; k \in \mathbb{N}$$

and,

$$\int_0^1 \psi_k(s)\psi_j(s)ds = \delta_{k,j}; \;\; k,j \in \mathbb{N}. \tag{1.13}$$

Moreover,

$$v(s,t) = \sum_{k=0}^{\infty} \lambda_k\psi_k(s)\psi_k(t); \;\; s,t \in [0,1],$$

where the series converges uniformly on $[0,1]^2$; hence

$$\sum_{k=0}^{\infty} \lambda_k = \int_0^1 v(t,t)dt < \infty.$$

Theorem 9 *(Karhunen-Loeve) Let $X = \{X(t), 0 \leq t \leq 1\}$ be a second order zero-mean measurable process with continuous covariance function v. Then*

$$X(t) = \sum_{k=0}^{\infty} \xi_k\psi_k(t), \;\; t \in [0,1], \tag{1.14}$$

where (ξ_k) is a sequence of real zero-mean random variables such that

$$E(\xi_k\xi_j) = \lambda_k\delta_{k,j}; \;\; k,j \in \mathbb{N};$$

and where the sequence (λ_k, ψ_k) is defined in the Mercer Lemma. The series in 1.14 converges uniformly with respect to the $L^2(\Omega, \mathcal{A}, P)$-norm.

The dimension reduction of X occurs through choosing the first K terms for large enough K in the Karhunen-Loeve expansion which provides a good approximation since 1.14 converges uniformly as $K \to \infty$. The Karhunen-Loeve expansion is used in many different functional data analysis methods as we will see in the next section.

1.4 Models

As in any area of Statistics, there are many different types of models that exist in FDA. The types of models range from simple linear models to complex nonlinear models. One of the key differences in the FDA context is that the response variable or the predictor variables can both be functional data. This of course depends on the type of application and what kinds of questions the researcher wants to answer. Every model has its advantages and disadvantages which can vary from a simple model and easier interpretation to more complex models but more difficult to interpret.

In this dissertation, functional models are used in two different settings. In the first project, the aim is to predict the conditional mean while the second project focuses on predicting the conditional quantile. Functional models can be used to model both types of responses but not all models that can be used for the conditional mean can be used for the conditional quantile. Most of the models described in this section can be used for both responses. The case where the response variable is functional can't be used to predict a conditional quantile.

1.4.1 Functional Regression Model with Scalar Response

One of the most common functional models is an extension of the the classic multivariate regression model. The main difference is that the predictor variable is now a function as opposed to a scalar. With the scalar response $Y \in \mathbb{R}$ and the functional predictor variable X observed on the time domain $t \in \mathcal{T}$, the model is defined as (Cai and Hall, 2006)

$$Y = \eta_0 + \int_{\mathcal{T}} X(t)\eta_1(t)dt + \epsilon. \tag{1.15}$$

The summation in the standard multivariate regression model is replaced by an integral over the time domain. Even though the function X changes continuously with time, the effect of $X(t)$ on Y is minimal but the effect over the entire time domain is more significant. One of the main differences is that now the coefficient of X, η_1, is a function over time rather than a vector. The estimate of the function η_1 can provide meaningful insights about what time points a future observation x of X will have the greatest effect on the response Y. This is one of the many advantages in using a functional model over a multivariate model.

This model can be extended to include multiple functional predictor variables $X_1, X_2, \ldots, X_p$ where each variable can be defined on a separate domain $\mathcal{T}_j$. The model is then defined as

$$Y = \eta_0 + \sum_{j=1}^{p} \int_{\mathcal{T}_j} X_j(t)\eta_j(t)dt + \epsilon.$$

Furthermore, the model above can be extended to include vector covariates $\boldsymbol{Z} = \{Z_1, Z_2, \ldots, Z_q\}^T$

$$Y = \alpha_0 + \boldsymbol{Z}^T\boldsymbol{\alpha} + \sum_{j=1}^{p} \int_{\mathcal{T}_j} X_j(t)\eta_j(t)dt + \epsilon.$$

1.4.2 Functional Regression Model with Functional Response

This model is an extension of the model described in section 1.4.1 where the response variable is now a function. There are two major models that have been considered which are (Wang et al., 2016),

$$Y(t) = \eta_0(t) + X(t)\eta_1(t) + \epsilon(t), \tag{1.16}$$

$$Y(s) = \eta_0(s) + \int_{\mathcal{T}} X(t)\eta_1(s,t)dt + \epsilon(s). \tag{1.17}$$

Model 1.16 implicitly assumes that the time domains of Y and X are equivalent and is most often referred to as a Varying Coefficient Model (VCM). This model assumes that Y is only dependent on X at a specific time t and not on the history of X, $\{X(s), s \leq t\}$. The varying coefficient model is a well-established model where a lot of theoretical and methodological developments have been made. Model 1.17 differs substantially from model 1.16 since the value of Y at any given time point s, depends on the entire trajectory of X.

1.4.3 Spatially Varying Coefficient Models

Varying Coefficient Models (VCM) are very useful models in the area of nonparametric statistics since they allow for modeling of different types of relationships. Traditionally, VCM's are used to model the relationship between two variables that vary with respect to time or with respect to one of the predictor variables. The goal is to determine how the effect of the predictor variable on the response changes over time. In this context, the varying coefficient is a function of time but this can be generalized to any variable such as age and height.

In spatial data analysis, it is often the case that a simple "global" model cannot explain some relationships, this is referred to as "spatial nonstationarity" (Mu et al., 2018). Since in the spatial context variables are changing over a spatial domain, a varying coefficient that is a function of space can help model the nonstationarity of the data. To define the model, suppose we have n independent copies of $\{(Y_i, X_{i0}, X_{i1}, \ldots, X_{ip}, S_{i1}, S_{i2})\}_{i=1}^n$, where Y is the response variable, $\boldsymbol{X} = (X_0, X_1, \ldots, X_p)^T$ is a $(p+1)$-dimensional vector of explanatory variables with $X_0 = 1$, and $\boldsymbol{S} = (S_1, S_2)^T$ is the spatial coordinates of the observation. Following the model proposed in Mu et al. (2018) and Kim and Wang (2021), suppose that $\{(Y_i, \boldsymbol{X}_i, \boldsymbol{S}_i)\}_{i=1}^n$ satisfies the following model:

$$Y_i = \boldsymbol{X}_i^T \boldsymbol{\beta}(\boldsymbol{S}_i) + \epsilon_i = \sum_{k=0}^{p} X_{ik} \beta_k(\boldsymbol{S}_i) + \epsilon_i, i = 1, \ldots, n, \tag{1.18}$$

where $\beta_k(.)$ are unknown varying-coefficient functions, and ϵ_i are *iid* random errors with $E(\epsilon_i) = 0$ and $Var(\epsilon_i) = \sigma^2$ and are independent of $\boldsymbol{X}_i$. The interest is then to estimate the varying coefficient functions $\beta_0, \ldots, \beta_p$ based on the observations $\{(Y_i, \boldsymbol{X}_i, \boldsymbol{S}_i)\}_{i=1}^n$.

1.4.4 Estimation of Functional Models

What is common to the models presented in Section 1.4.1 and 1.4.2 is that the estimation of the coefficients is intrinsically infinite dimensional (Cai and Hall, 2006). Therefore, estimation of these models directly is incredibly difficult. To reduce the dimensionality, the Karhunen-Loeve expansions of $X(t)$ and $\eta_1(t)$ in terms of the eigendecomposition of the covariance operator V are

$$X(t) = \sum_{k=0}^{\infty} \xi_k \psi_k(t), \ t \in [0, 1],$$
$$\eta_1(t) = \sum_{k=0}^{\infty} \beta_k \psi_k(t), \ t \in [0, 1],$$

where $\xi_k = \int_{\mathcal{T}} X(t)\psi_k(t)dt$ and $\beta_k = \int_{\mathcal{T}} \eta(t)\psi_k(t)dt$. Then using the first K terms in the expansion, model (1.16) can be rewritten as

$$\begin{aligned}
Y &= \eta_0 + \int_{\mathcal{T}} X(t)\eta_1(t)dt + \epsilon \\
&\approx \eta_0 + \int_{\mathcal{T}} \Big\{\sum_{k=0}^{K} \xi_k\psi_k(t)\Big\}\Big\{\sum_{k=0}^{K} \beta_k\psi_k(t)\Big\}dt + \epsilon \\
&= \eta_0 + \int_{\mathcal{T}} \Big\{\sum_{k=0}^{K} \xi_k\psi_k(t)\beta_k\psi_k(t) + \sum_{k\neq j} \xi_k\psi_k(t)\beta_j\psi_j(t)\Big\}dt + \epsilon \\
&= \eta_0 + \sum_{k=0}^{K} \xi_k\beta_k \int_{\mathcal{T}} \psi_k(t)\psi_k(t)dt + \sum_{k\neq j} \xi_k\beta_j \int_{\mathcal{T}} \psi_k(t)\psi_j(t)dt + \epsilon. \qquad (1.19)
\end{aligned}$$

By 1.13 in Lemma 8, we have that

$$\int_0^1 \psi_k(s)\psi_j(s)ds = \delta_{k,j};\;\; k, j \in \mathbb{N}.$$

Then 1.19 is expressed as

$$Y \approx \eta_0 + \sum_{k=0}^{K} \xi_k\beta_k + \epsilon. \qquad (1.20)$$

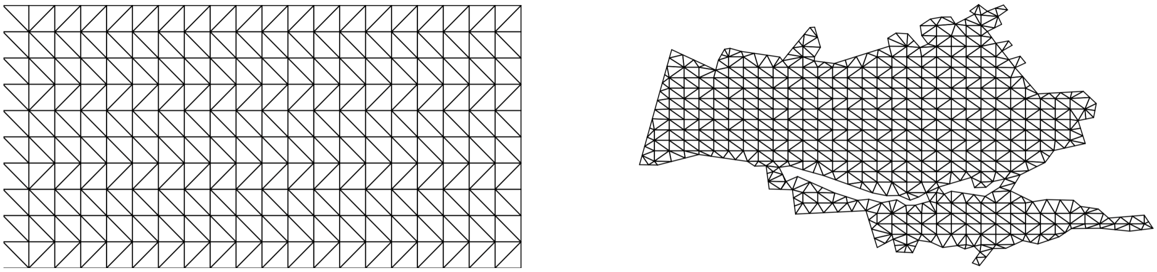
The functional linear regression model in 1.20 has now reduced to a finite dimensional linear regression model. The model in 1.15 can now be estimated directly using classical estimation methods such as least squares.

1.5 Bivariate Splines on Triangulations

1.5.1 Triangulation

The method of triangulation has historically been used in computer vision and graphics to represent the structure of geometric objects. Its methods and theory have been established within mathematics for a long time. It is only in the last decade or so that triangulation has been used in the context of statistics (Lai and Wang, 2013; Mu et al., 2018; Kim and Wang, 2021). More specifically, there is a rich area in mathematics that focuses on the use of splines (of different dimension) on triangulation's. Even though this is the case, the exploration of its uses in statistics has been limited. The method of triangulation can be extended to any kind of polygon shapes but we focus on triangles since any polygonal domain of arbitrary shape can be partitioned into finitely many triangles.

Two-dimensional space takes on many shapes such as rectangles, circles, trapezoids and many other undefined shapes. This is an inherent part of spatial data since phenomena doesn't change over pre-specified domains. When fitting a model to data collected over a spatial domain, there are certain methods that were built specifically for this purpose. Some of the common techniques that have been used are tensor-product splines, kriging, soap film smoothing, and thin-plate splines (TPS). While these methods have been successful at modeling rectangular spatial domains, their accuracy decreases when irregular-shaped spatial domains are used. This is because these methods were developed under the assumption that the domain is rectangular. The issue with complex domains is that the corners and edges are difficult to cover with shapes other than triangles. Following Kim and Wang (2021), assume we observe spatial locations $\boldsymbol{v} = (x, y)$ on a bounded domain $\Omega \subseteq \mathbb{R}^2$ of arbitrary shape. Suppose that Ω can be covered by a finite collection of N triangles $T_1, T_2, \dots, T_N$ such that $\Omega = \bigcup_{i=1}^{N} T_i$. A collection of triangles $\Delta := (T_1, T_2, \dots, T_N)$ is called a triangulation over the domain Ω where it's assumed that if two triangles intersect, then their intersection is either an edge or a vertex. For any triangle $T \in \Delta \subseteq \mathbb{R}^2$ with vertices $\boldsymbol{v} = (x, y) \in \mathbb{R}^2$, T is defined in terms of its vertices where $T = \langle \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3 \rangle$. An example of a triangulation of a rectangular domain and a complex domain can be seen in Figure 1.1.



(a) Triangulation over Rectangular Domain.

(b) Triangulation over Complex Domain.

Figure 1.1: Triangulation Examples.

1.5.2 Bivariate Splines

This section provides the foundations for bivariate splines over triangulations. More details can be found in Lai and Schumaker (2007). Bivariate splines are a simple extension of univariate splines where only the dimension changes. In univariate splines, the domain is split up into a finite set of intervals where a basis is used to fit a curve over each interval. In the case of a two-dimensional domain, the domain is split into a set of triangles as described in Section 1.5.1 where a surface is fit over each triangle. Before defining bivariate splines, bivariate polynomials are defined in Definition 10.

Definition 10 *Given a nonnegative integer d, we write $\mathbb{P}_d$ for the space of bivariate polynomials of degree d, i.e., the linear space of all real-valued functions of the form*

$$p(x,y) = \sum_{0 \le i+j \le d} c_{ij} x^i y^j,$$

where $(c_{ij})_{0 \le i+j \le d}$ are real numbers. The monomials

$$(x^i y^j)_{0 \le i+j \le d}$$

form a basis for $\mathbb{P}_d$.

To arrive at a point where we can fit bivariate splines over triangles, we first need to introduce barycentric coordinates. In the following section, for $\boldsymbol{v} \in \mathbb{R}^2$, define $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$ as the vertices of a triangle T.

Lemma 11 *Every point $\boldsymbol{v} \in \mathbb{R}^2$ has a unique representation in the form*

$$\boldsymbol{v} = b_1 \boldsymbol{v}_1 0 + b_2 \boldsymbol{v}_2 + b_3 \boldsymbol{v}_3 \tag{1.21}$$

with

$$1 = b_1 + b_2 + b_3. \tag{1.22}$$

By Lemma 11, we have that any point $\boldsymbol{v} \in \mathbb{R}^2$ can be uniquely represented by its barycentric coordinates b_1, b_2, b_3 with respect to triangle T as

$$\boldsymbol{v} = b_1 \boldsymbol{v}_1 + b_2 \boldsymbol{v}_2 + b_3 \boldsymbol{v}_3.$$

This is a very unique way to represent a point that lies within a triangle. To gain some intuition, an interesting representation of the barycentric coordinates is a geometric one where given a point $\boldsymbol{v} \in T$, define the triangles $T_1 = \langle \boldsymbol{v}, \boldsymbol{v}_1, \boldsymbol{v}_3 \rangle$, $T_2 = \langle \boldsymbol{v}, \boldsymbol{v}_3, \boldsymbol{v}_1 \rangle$, $T_3 = \langle \boldsymbol{v}, \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle$ which can be seen in Figure 1.2 below.

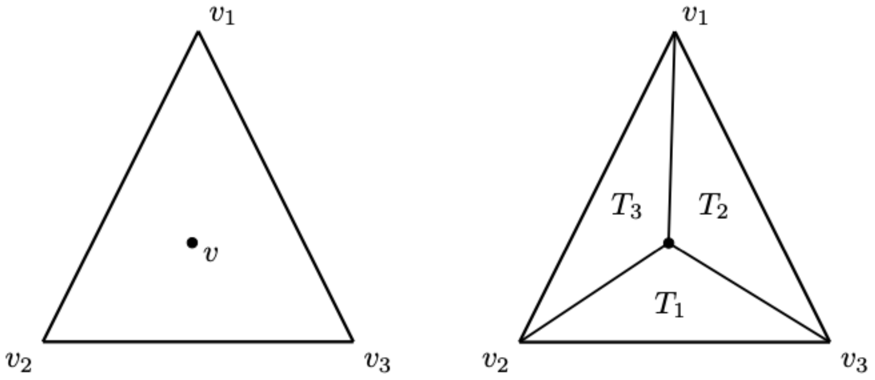


Figure 1.2: Geometric Representation of Barycentric Coordinates.

Then the barycentric coordinates of $\boldsymbol{v}$ relative to T are given by

$$b_i = \frac{A_{T_i}}{A_T}, i = 1, 2, 3,$$

where A represents the area of the corresponding triangle.

Lemma 12 *For each $i = 1, 2, 3$, the function b_i is a linear polynomial in x and y which assumes the value 1 at the vertex $\boldsymbol{v}_i$ and vanishes at all points on the edge of T opposite to $\boldsymbol{v}_i$.*

An example of Lemma 12 for the barycentric coordinate b_1 of a point $\boldsymbol{v} = (x, y) \in \mathbb{R}^2$ is

$$b_1 = \frac{1}{2A_T} det \begin{bmatrix} 1 & 1 & 1 \\ x & x_2 & x_3 \\ y & y_2 & x_3 \end{bmatrix} = \frac{(x_2 y_3 - y_2 x_3) - x(y_3 - y_2) + y(x_3 - x_2)}{2A_T},$$

which shows that b_1 is indeed a linear polynomial. The expressions for b_2 and b_3 follow similarly. Using the barycentric coordinates, we can define bivariate basis functions in Definition 13.

Definition 13 *Given a point $\boldsymbol{v} = (x, y) \in \mathbb{R}^2$ with barycentric coordinates b_1, b_2, b_3 and nonnegative integers summing to d, define the Bernstein basis polynomials of degree $d \geq 1$ relative to T by*

$$B^d_{ijk} = \frac{d!}{i!j!k!} b_1^i b_2^j b_3^k.$$

The Bernstein basis polynomials have many nice properties. One very useful property is described in Theorem 14.

Theorem 14 *The set*

$$\mathcal{B}^d = (B^d_{ijk})_{i+j+k=d}$$

of Bernstein basis polynomials is a basis for the space of all polynomials $\mathbb{P}_d$.

Since the set $\mathcal{B}^d$ of Bernstein basis polynomials is a basis for $\mathbb{P}_d$, we can use the basis functions B^d_{ijk} to represent any functions in the space $\mathbb{P}_d$. Let the space $\mathbb{P}_d(T)$ be the space of all polynomials of degree less than or equal to d with respect to triangle T. Then by Theorem 14, every polynomial $\mathcal{P} \in \mathbb{P}_d(T)$ can be uniquely written as

$$\mathcal{P}(\boldsymbol{s}) = \sum_{i+j+k=d} \gamma_{T;ijk} B_{T;ijk}(\boldsymbol{s}),$$

which is called the B-form of $\mathcal{P}$ relative to T.

Chapter 2

Prediction in Generalized Spatially Varying Functional Regression Model

2.1 Introduction

Technological improvements in data collection have allowed for variables to be continuously measured over time and space which give rise to multivariate spatially varying functional data (Ramsay and Silverman, 2005). Analyzing complex data such as this, creates challenges in gaining insight from the data. There are many existing models for functional data such as the Functional Linear Model (FLM), Functional Additive Model (FAM) and Partially Linear Functional Additive Model (PLFAM) but they don't consider spatial variation (Cai and Hall, 2006). Varying Coefficient Models (VCM) have been used for data that change with respect to some other variable. They are very flexible models since they allow for the behavior to change across the domain of the predictor variables. VCM's were first introduced in the paper by Hastie and Tibshirani (1993). In this paper, they explored a class of regression models and generalized regression models where the coefficients can vary as smooth functions of other variables. The general approach to model estimation in VCM's is the local linear regression method. This method assumes degrees of smoothness for each coefficient. In Fan and Zhang (1999), they introduce a new estimation method when different coefficients have different degrees of smoothness. Furthermore, Fan and Zhang (1999) introduced methods to perform inference with VCM's. There has been significant development in VCM's since their introduction and they are widely applicable in many different settings. The coefficients in a VCM can be a smooth function of any other variable such as space. In the past decade, there has been a strong development in spatial models, more specifically, Spatially Varying Coefficient Models (SVCM). In this chapter, we introduce the Generalized Spatially Varying Functional Model (GSVFM) for predicting a spatial response variable dependent on a spatial predictor variable and multivariate spatially varying functional predictors that is performed by a novel two-step procedure.

Spatially varying functional data provide a rich source of information but is infinite-dimensional. A common approach to reduce the dimensionality is by using multivariate Functional Principal Components Analysis (mFPCA). In most existing functional models, the principal component scores are used as scalar predictor variables. For the proposed two-step procedure, the theoretical model is the GSVFM. There is no closed form for the estimation of the GSVFM since it is a infinite-dimensional model. To estimate the GSVFM, it needs to be reduced to a finite-dimensional model. We project the functional data (at each spatial location) to a low-dimensional space and the projections are used as predictors in a Generalized Spatially Varying Coefficient Model (GSVCM). The GSVCM allows the principal component scores and coefficient functions to vary over space. By allowing the coefficients to vary over space, the GSVCM can capture location-specific effects. This is the main distinction between existing functional models and the two-step procedure through the use of the GSVCM. For spatially varying functional data, using the GSVCM could provide improved model performance. The GSVCM also provides flexibility for different types of data since it allows for different link functions to be used.

There are existing methods that can be used to estimate varying coefficients in SVCM's. One of the popular estimation methods is the local kernel based approach, such as Geographically Weighted Regression proposed by Brunsdon et al. (2010) and local polynomial maximum likelihood method by Sun et al. (2014). The kernel method works well for nonstationary data but can become computationally complex as the sample size grows. In many spatial data applications there is a possibility of complex boundaries as well as interior holes. There is a vast-literature concerning the handling of complex domains. Some of the existing methods include differential regularization (Ramsay, 2002; Sangalli et al., 2013; Azzimonti et al., 2015; Wilhelm and Sangalli, 2016), geodesic low-rank thin plate splines (Wang and Ranalli, 2007), complex region spatial smoother (Scott-Hayward et al., 2014), and soap-film smoothing (Wood et al., 2008). While these methods have been shown to work well for spatial data that can be represented using a single bivariate function, they don't allow for coefficient functions to vary over space. This means that previous models don't allow for any predictor variables in the model, only the spatial locations. This limits the applicability of these models, in applications where spatially varying variables have an effect. To model data distributed over complex domains, the method of Bivariate Penalized Splines over Triangulations (BPST) is introduced in Lai and Wang (2013). The method of triangulation has been shown to be a computationally efficient tool to model complex domains. In Lai and Wang (2013), they were able to show that the BPST method provided better efficiency and optimal convergence rates for the nonparametric components in the model. In Mu et al. (2018), the triangulation method was extended to included varying coefficients for predictor variables. The Generalized Spatially Varying Coefficient Model (Kim and Wang, 2021) extended the SVCM and provided a new estimation method using a quasi-likelihood approach to accommodate different link functions.

The present research is motivated by a crop yield prediction application in agriculture. Agriculture is an industry of major importance in the United States, because it is the source of livelihood for millions and global food security. Getting reliable crop yield predictions in a timely manner is of great importance because of the agriculture industry significance. The temperature data is accessed through publicly available data at NOAA and the agricultural data is accessed through the USDA database. Since county-level daily minimum and maximum temperature time series data are available, it is natural to treat the temperature time series as functional data. It is natural to model the relationship between weather and crop-yield since it is well known that weather has a significant effect on crop yield (Wong et al., 2018). The county-level crop yield, precipitation and proportion of irrigated land are also provided. In the present work, we use data from five Midwestern states, Kansas, Missouri, Indiana, Illinois and Iowa.

In this chapter, we aim to show that the GSVFM is advantageous for spatially varying functional data. Although the GSVFM was introduced in (Park et al., 2022), their approach applies a Bayesian framework while the following research utilizes a frequentist approach. The Bayesian approach worked well compared to existing methods but requires multiple prior distributions. The performance of the GSVFM using the two-step procedure is compared to non-spatial functional models, the PLFAM (Wong et al., 2018) and FLM (Cai and Hall, 2006). This comparison will show that the GSVFM is able to capture location-specific effects present in the data.

The organization of the rest of the chapter is as follows. In Sections 2.2-2.3, we describe the model, methodology and estimation procedure used for the proposed method. We investigate the performance of the proposed method by conducting simulations on spatially varying data in Section 2.4 and apply it to the agriculture data application in Section 2.5. Concluding remarks are discussed in Section 2.6.

2.2 Methodology

The model in consideration is intended to perform prediction for the scalar response variable $Y(\boldsymbol{s})$. Let $Y(\boldsymbol{s})$ denote the scalar response variable located at $\boldsymbol{s} \in \Omega$ for a spatial domain $\Omega \subset \mathbb{R}^2$. The scalar response $Y(\boldsymbol{s})$ is assumed to be dependent on q functional variables $\boldsymbol{X}(\boldsymbol{s};t) = \{X_1(\boldsymbol{s};t), X_2(\boldsymbol{s};t), \ldots, X_q(\boldsymbol{s};t)\}^T$ defined for $t \in \mathcal{T}$ and m scalar predictors $\boldsymbol{Z}(\boldsymbol{s}) = \{Z_1(\boldsymbol{s}), Z_2(\boldsymbol{s}), \ldots, Z_m(\boldsymbol{s})\}^T$. Without loss of generality, we rescale the time domain to a unit interval where $\mathcal{T} = [0, 1]$. We want to model the spatially varying relationship between the scalar responses and associated predictors. To allow for different families within the exponential family, the conditional density of Y given $(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})$ can be represented as

$$f_{Y|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}}(y) = exp\left\{ \frac{1}{\sigma^2}[y\varrho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{s}) - \mathcal{B}\{\varrho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{s})\}] + \mathcal{C}(y, \sigma^2) \right\},$$

for some functions $\mathcal{B}$ and $\mathcal{C}$, dispersion parameter σ^2 and canonical parameter ϱ. Define the conditional mean $\mu(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}) = E[Y|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}]$. We assume that $\mu(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})$ is modeled by a link function g as follows

$$g\{\mu(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})\} = \eta_0(\boldsymbol{s}) + \boldsymbol{Z}^T(\boldsymbol{s})\boldsymbol{\alpha}(\boldsymbol{s}) + \int_{\mathcal{T}} \boldsymbol{X}^T(\boldsymbol{s}; t)\boldsymbol{\eta}(\boldsymbol{s}; t)dt, \tag{2.1}$$

where $\eta_0(\boldsymbol{s})$ is the location-specific intercept, $\boldsymbol{\alpha}(\boldsymbol{s}) = \{\alpha_1(\boldsymbol{s}), \alpha_2(\boldsymbol{s}), \ldots, \alpha_m(\boldsymbol{s})\}^T$ is a vector of the spatially varying coefficients, and $\boldsymbol{\eta}(\boldsymbol{s}; t) = \{\eta_1(\boldsymbol{s}; t), \eta_2(\boldsymbol{s}; t), \ldots, \eta_q(\boldsymbol{s}; t)\}^T$ is a vector of spatially varying functions over $t \in \mathcal{T}$ modeling the effect of functional trajectories on the scalar response at location $\boldsymbol{s}$. Define the conditional variance of Y as $\text{Var}(Y|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}) = \sigma^2 V\{\mu(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})\}$ for some variance function V.

Since the direct estimation of model 2.1 suffers from the curse of dimensionality, we need a low-dimensional representation of the spatially dependent multivariate functional predictors. To reduce the dimensionality, we use the method of mFPCA for the functional variables. We perform mFPCA assuming second-order stationarity. A direct consequence of second-order stationarity, is that the variance is the same for each location and a constant mean across locations. Since we don't assume the mean is constant across space, we enforce second-order stationarity by centering the functional data by its location-specific mean before performing mFPCA. This step results in the functional data having a constant mean across all locations. To estimate the eigenvalues and eigenfunctions of the functional data $\boldsymbol{X}(\boldsymbol{s};t)$, its covariance function first needs to be defined. The covariance function is defined as follows

$$\mathcal{C}(t_1,t_2) = \mathbb{E}[\{\boldsymbol{X}(\boldsymbol{s};t_1) - \boldsymbol{\mu}(\boldsymbol{s};t_1)\}\{\boldsymbol{X}(\boldsymbol{s};t_1) - \boldsymbol{\mu}(\boldsymbol{s};t_2)\}^T], \tag{2.2}$$

where $\boldsymbol{\mu}(\boldsymbol{s};t) = \mathbb{E}\{X(\boldsymbol{s};t)\} = \{\mu_1(\boldsymbol{s};t), \mu_2(\boldsymbol{s};t), \dots, \mu_q(\boldsymbol{s};t)\}^T$ is the mean function of the multivariate functional predictors. The eigendecomposition of $\mathcal{C}(t_1,t_2)$ for time points $t_1 \neq t_2$, is defined as $(\lambda_r, \boldsymbol{\psi}_r)_{r=1}^{\infty}$. By Mercer's Lemma, the covariance function has the following spectral decomposition,

$$C(t_1,t_2) = \sum_{r=1}^{\infty} \lambda_r \boldsymbol{\psi}_r(t)\boldsymbol{\psi}_r^T(t),$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ are the eigenvalues and $\boldsymbol{\psi}_r = (\psi_{r1}, \dots, \psi_{rq})^T$ are the corresponding eigenfunctions such that $\langle \boldsymbol{\psi}_r, \boldsymbol{\psi}_{r'} \rangle = \int_{\mathcal{T}} \boldsymbol{\psi}_r^T \boldsymbol{\psi}_{r'} dt = I(r = r')$. Then we can define the Karhunen-Loeve expansions of $\boldsymbol{X}(\boldsymbol{s};t)$ and $\boldsymbol{\eta}(\boldsymbol{s};t)$ as follows

$$\boldsymbol{X}(\boldsymbol{s};t) = \boldsymbol{\mu}(\boldsymbol{s};t) + \sum_{r=1}^{\infty} \xi_r(\boldsymbol{s})\boldsymbol{\psi}_r(t),$$
$$\boldsymbol{\eta}(\boldsymbol{s};t) = \sum_{r=1}^{\infty} \beta_r(\boldsymbol{s})\boldsymbol{\psi}_r(t),$$

where $\xi_r(\boldsymbol{s})$ and $\beta_r(\boldsymbol{s})$ are the corresponding principal component scores for $\boldsymbol{X}(\boldsymbol{s};t)$ and $\boldsymbol{\eta}(\boldsymbol{s};t)$, respectively. The functional principal component scores $\xi_r(\boldsymbol{s})$ and $\beta_r(\boldsymbol{s})$ are defined as

$$\xi_r(\boldsymbol{s}) = \int_{\mathcal{T}} \{\boldsymbol{X}(\boldsymbol{s};t) - \boldsymbol{\mu}(\boldsymbol{s};t)\}^T \boldsymbol{\psi}_r(t)dt,$$

$$\beta_r(\boldsymbol{s}) = \int_{\mathcal{T}} \boldsymbol{\eta}^T(\boldsymbol{s};t)\boldsymbol{\psi}_r(t)dt,$$

where $\xi_r(\boldsymbol{s})$ and $\beta_r(\boldsymbol{s})$ are zero-mean random variables with $\mathbb{E}\{\xi_r(\boldsymbol{s})\xi_{r'}(\boldsymbol{s})\} = \lambda_r I(r = r')$ and $\mathbb{E}\{\beta_r(\boldsymbol{s})\beta_{r'}(\boldsymbol{s})\} = \lambda_r I(r = r')$. Using the expansions of $\boldsymbol{X}(\boldsymbol{s};t)$ and $\boldsymbol{\eta}(\boldsymbol{s};t)$, model 2.1 can now be rewritten as a Generalized Spatially Varying Coefficient Model (GSVCM) (Mu et al. (2018); Kim and Wang (2021))

$$g\{\mu(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})\} = \alpha_0(\boldsymbol{s}) + \sum_{l=1}^{m} Z_l(\boldsymbol{s})\alpha_l(\boldsymbol{s}) + \sum_{r=1}^{\infty} \xi_r(\boldsymbol{s})\beta_r(\boldsymbol{s}). \tag{2.3}$$

2.2.1 Estimation in mFPCA

Let $\{Y_k(\boldsymbol{s}_u), \boldsymbol{X}_k(\boldsymbol{s}_u;t), \boldsymbol{Z}_k(\boldsymbol{s}_u)\}_{k=1}^{K_u}$ be an independent copy of K_u spatial replicates from the joint distribution $(Y, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})$ at location $\boldsymbol{s}_u \in \Omega \subset \mathbb{R}^2, u = 1, \ldots, U$. Define the total number of observations by $n = \sum_{u=1}^{U} K_u$ and denote the ith observation as $\{Y_i(\boldsymbol{s}_i), \boldsymbol{X}_i(\boldsymbol{s}_i;t), \boldsymbol{Z}_i(\boldsymbol{s}_i)\}$, for $i = 1, \ldots, n$, where $\boldsymbol{s}_i$ represents the location for the ith observation. Since the covariance function in 2.2 is unknown, we can estimate it as follows

$$\widehat{C}(t_1, t_2) = \frac{1}{n}\sum_{u=1}^{U}\sum_{k=1}^{K_u}\{\boldsymbol{X}_k(\boldsymbol{s}_u;t_1) - \bar{\boldsymbol{X}}(\boldsymbol{s}_u;t_1)\}\{\boldsymbol{X}_k(\boldsymbol{s}_u;t_2) - \bar{\boldsymbol{X}}(\boldsymbol{s}_u;t_2)\}^T \tag{2.4}$$

$$= \sum_{r=1}^{n-1} \widehat{\lambda}_r \widehat{\boldsymbol{\psi}}(t_1)\widehat{\boldsymbol{\psi}}(t_2),\ t_1, t_2 \in [0, 1], \tag{2.5}$$

where $\bar{\boldsymbol{X}}(\boldsymbol{s}_u;t) = \frac{1}{K_u}\sum_{k=1}^{K_u} \boldsymbol{X}_k(\boldsymbol{s}_u;t)$ and $(\widehat{\lambda}_r, \widehat{\boldsymbol{\psi}}_r)_{r=1}^{\infty}$ are the estimated eigenvalues and eigenfunctions. Since $\widehat{C}$ has rank $n - 1$(Wong et al., 2018), it has the following spectral decomposition by Mercer's Lemma (8),

$$\widehat{C}(t_1, t_2) = \sum_{r=1}^{n-1} \widehat{\lambda}_r \widehat{\boldsymbol{\psi}}(t_1)\widehat{\boldsymbol{\psi}}(t_2),\ t_1, t_2 \in [0, 1].$$

To reduce the dimension, the first p functional principal components for large enough p are chosen. Then the r'th functinal principal component score can be estimated by,

$$\widehat{\xi}_{ir} = \int_{\mathcal{T}} \{\boldsymbol{X}_i(\boldsymbol{s}_i; t) - \boldsymbol{\mu}(\boldsymbol{s}_i; t)\}^T \widehat{\boldsymbol{\psi}}_r dt,\ i = 1, \ldots, n,$$

for $r = 1, \ldots, n-1$. The GSVCM introduced in 2.3 can be approximated by

$$g\{\mu(\boldsymbol{X}_i, \boldsymbol{Z}_i, \boldsymbol{s}_i)\} \approx \alpha_0(\boldsymbol{s}_i) + \sum_{l=1}^{m} Z_{il}(\boldsymbol{s}_i)\alpha_l(\boldsymbol{s}_i) + \sum_{r=1}^{p} \widehat{\xi}_{ir}(\boldsymbol{s}_i)\beta_r(\boldsymbol{s}_i). \tag{2.6}$$

Let $\boldsymbol{\mathcal{X}}_i(\boldsymbol{s}_i) = \{\mathcal{X}_{i0}(\boldsymbol{s}_i), \mathcal{X}_{i1}(\boldsymbol{s}_i), \ldots, \mathcal{X}_{i(m+p)}(\boldsymbol{s}_i)\}^T = \{Z_{i0}(\boldsymbol{s}_i) \equiv 1, Z_{i1}(\boldsymbol{s}_i), \ldots, Z_{im}(\boldsymbol{s}_i), \widehat{\xi}_{i1}(\boldsymbol{s}), \ldots, \widehat{\xi}_{ip}(\boldsymbol{s}_i)\}^T$ be the vector of predictor variables and $\boldsymbol{\theta}(\boldsymbol{s}) = \{\theta_0(\boldsymbol{s}), \theta_1(\boldsymbol{s}), \ldots, \theta_{m+p}(\boldsymbol{s})\}^T = \{\alpha_0(\boldsymbol{s}), \alpha_1(\boldsymbol{s}), \ldots, \alpha_m(\boldsymbol{s}), \beta_1(\boldsymbol{s}), \ldots, \beta_p(\boldsymbol{s})\}^T$ the vector of coefficients, for $j = 0, \ldots, (m+p)$, then model 2.6 can be rewritten as

$$g\{\mu(\boldsymbol{X}_i, \boldsymbol{Z}_i, \boldsymbol{s}_i)\} \approx \sum_{j=0}^{m+p} \mathcal{X}_{ij}(\boldsymbol{s})\theta_j(\boldsymbol{s}_i) = \boldsymbol{\mathcal{X}}_i(\boldsymbol{s}_i)^T\boldsymbol{\theta}(\boldsymbol{s}_i). \tag{2.7}$$

We want to find an estimator for $\boldsymbol{\theta}(\boldsymbol{s})$ by using the triangulation method introduced in Section 2.2.2.

2.2.2 Bivariate Spline Approximation over Triangulation

Suppose that Ω can be covered by a finite collection of N triangles $\delta_1, \delta_2, \ldots, \delta_N$ such that $\Omega = \bigcup_{i=1}^{N} \delta_i$. A collection of triangles $\Delta := \{\delta_1, \delta_2, \ldots, \delta_N\}$ is called a triangulation over the domain Ω where it's assumed that if two triangles intersect, then their intersection is either an edge or a vertex. For any triangle $\delta \in \Delta \subseteq \mathbb{R}^2$ with vertices $\boldsymbol{s} = (s_1, s_2) \in \mathbb{R}^2$, δ is defined in terms of its vertices where $\delta = \langle \boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3 \rangle$.

A unique property of triangles, is that for any point $\boldsymbol{s}_u = (s_{u1}, s_{u2}) \in \delta$, $\boldsymbol{s}$ can be represented as $\boldsymbol{s}_u = b_1\boldsymbol{s}_1 + b_2\boldsymbol{s}_2 + b_3\boldsymbol{s}_3$, where b_1, b_2, b_3 are the barycentric coordinates of the point $\boldsymbol{s}$. Barycentric coordinates are useful since they can be used to define basis functions over the complex domain. For a given point $\boldsymbol{s}$ and nonnegative integers i, j, k with degree $d = i + j + k \geq 1$, its Bernstein basis polynomials relative to triangle δ are defined as $B^d_{ijk} = \frac{d!}{i!j!k!} b_1^i b_2^j b_3^k$. An important property of the Bernstein polynomials is that the set $\mathcal{B} = \{B^d_{ijk}(\boldsymbol{s})\}_{i+j+k=d}$ is a basis for the space of all bivariate polynomials of degree d, denoted as $\mathbb{P}_d$, of the form $p(x, y) = \sum_{0 \leq i+j \leq d} c_{ij} x^i y^j$. Based on this property, for a nondegenerate triangle δ and spline coefficients $\{\gamma_{\delta;ijk}\}$, any polynomial $\mathcal{P} \in \mathbb{P}_d(\delta)$, $\mathcal{P}$ can be expressed in terms of the Bernstein basis polynomials where $\mathcal{P}(\boldsymbol{s}) = \sum_{i+j+k=d} \gamma_{\delta;ijk} B^d_{ijk}(\boldsymbol{s})$ which is called the B-form of $\mathcal{P}$ relative to δ.

Given $0 \leq v < d$ and a triangulation Δ, define the spline space of degree d and smoothness v over Δ as $\mathbb{S}^v_d(\Delta) = \{\mathcal{P} \in \mathbb{C}^v(\Omega) : \mathcal{P}|_{\delta_g} \in \mathbb{P}_d(\delta_g), \delta_g \in \Delta, g = 1, \ldots, G\}$, where $\mathbb{C}^v(\Omega)$ is the space of v times continuously differentiable functions over the spatial domain Ω. Now suppose we have a triangulation Δ_j with N_j triangles for each parameter $j = 0, \ldots, m + p$. To estimate the parameter θ_j, define the set of bivariate Bernstein polynomials for the spline space $\mathbb{S}^v_d(\Delta_j)$ as $\{B_{jw}\}_{w \in \mathcal{W}_j}$, where $\mathcal{W}_j$ represents an index set for the basis functions on Δ_j with cardinality $|\mathcal{W}_j| = \frac{N_j(d+1)(d+2)}{2}$. Then, the bivariate functions $\theta_j \in \mathbb{S}^v_d(\Delta_j)$ can be approximated by $\theta_j(\boldsymbol{s}) = \sum_{w \in \mathcal{W}_j} B_{jw}(\boldsymbol{s}_u)\gamma_{jw} = \boldsymbol{B}_j(\boldsymbol{s}_u)^T \boldsymbol{\gamma}_j$ where $\boldsymbol{B}_j(\boldsymbol{s}) = \{B_{jw}, w \in \mathcal{W}_j\}^T$ and $\boldsymbol{\gamma}_j = \{\gamma_{jw}, w \in \mathcal{W}_j\}$ is the vector of bivariate basis functions evaluated at location $\boldsymbol{s}$ and the corresponding spline coefficient vector, respectively. To ensure that the spline surface is smooth across the shared edges of triangles in Δ_j, we enforce global smoothness restrictions in $\mathbb{S}_d^v(\Delta_j)$ where we introduce the constraint matrix $\boldsymbol{H}_j$ that satisfies $\boldsymbol{H}_j\boldsymbol{\gamma}_j = \boldsymbol{0}$ for all $j = 0, \ldots, m+p$. The matrix $\boldsymbol{H}_j$ depends on the smoothness of the basis functions and the structure of the triangulation Δ_j.

2.2.3 Penalized Quasi-Likelihood Method

To estimate the bivariate functions $\theta_j(\boldsymbol{s}),\ j = 0, \ldots, (m+p)$, a quasi-likelihood approach is used (Kim and Wang, 2021). The quasi-likelihood method is quite flexible since it doesn't require any distributional assumptions. If there exists a positive function V for the $\mathrm{Var}(Y|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})$ defined in the beginning of Section 2.2, then the estimation of the mean function $\mu(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})$ in model 2.3 can be achieved by replacing the conditional log-likelihood function $\log\{f_{Y|\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s}}(y)\}$ with a quasi-likelihood function $l_Q\{\mu(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})\}$, which satisfies $\frac{\partial}{\partial\mu} l_Q\{\mu(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})\} = \frac{y-\mu}{\sigma^2 V(\mu)}$. Let $\zeta(\boldsymbol{\mathcal{X}}, \boldsymbol{s}) = \sum_{j=0}^{(m+p)} \mathcal{X}_j(\boldsymbol{s})\theta_j(\boldsymbol{s})$. Then for the quasi-likelihood $l_Q\{g^{-1}(\zeta), y\}$, denote $\phi_1(\zeta, y)$ and $\phi_2(\zeta, y)$ as follows

$$\begin{aligned}
\phi_1(\zeta, y) &= \frac{\partial}{\partial\zeta} l_Q\{g^{-1}(\zeta), y\} = \{y - g^{-1}(\zeta)\}\phi_1(\zeta), \\
\phi_2(\zeta, y) &= \frac{\partial^2}{\partial\zeta^2} l_Q\{g^{-1}(\zeta), y\} = \{y - g^{-1}(\zeta)\}\phi_1'(\zeta) - \phi_2(\zeta),
\end{aligned}$$

where $\phi_h(\zeta)$ is defined as

$$\phi_h(\zeta) = \frac{\{\frac{\partial}{\partial\zeta} g^{-1}(\zeta)\}^h}{[\sigma^2 V\{g^{-1}(\zeta)\}]} = \frac{[g'\{g^{-1}(\zeta)\}]^h}{[\sigma^2 V\{g^{-1}(\zeta)\}]}$$

for $h = 1, 2$.

It is common for points to be irregularly spaced within a spatial domain. This can lead to difficulty in estimating a smooth bivariate function over the domain. Because of this, we use penalization in the estimation through the energy functional penalty. The energy functional penalty is defined as

$$\varepsilon(f) = \int_{\Omega} \{(\nabla_{s_1}^2 f)^2 + 2(\nabla_{s_1}\nabla_{s_2} f)^2 + (\nabla_{s_2}^2 f)^2\}\ ds_1 ds_2,$$

which enforces smoothness conditions between the bivariate functions spread out across the domain. The energy penalty enforces smoothness across space since the second gradient over space is a measure for the curvature of a spatially varying function. Using the quasi-likehood function l_Q and the energy penalty method, the objective function is defined as

$$\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} l_Q \left[g^{-1} \left\{ \sum_{j=0}^{m+p} \mathcal{X}_{ij}(\boldsymbol{s}_i)\theta_j(\boldsymbol{s}_i) \right\}, Y_i(\boldsymbol{s}_i) \right] - \frac{1}{2} \sum_{j=0}^{m+p} \tau_j \varepsilon(\theta_j), \tag{2.8}$$

for nonnegative smoothness parameters $\tau_1, \ldots, \tau_{m+p}$. Although different triangulations can be used for estimation, Bernstein basis polynomials and constraint matrices can be used for each parameter, for simplicity we assume that $\Delta_j = \Delta$, $\boldsymbol{B}_j = \boldsymbol{B} = \{\boldsymbol{B}(\boldsymbol{s}_1), \ldots, \boldsymbol{B}(\boldsymbol{s}_n)\}^T$ and $\boldsymbol{H}_j = \boldsymbol{H}$ where $\boldsymbol{B}_j(\boldsymbol{s}_i) = \boldsymbol{B}(\boldsymbol{s}_i) = \{B_w(\boldsymbol{s}_i), w \in \mathcal{W}_j\}$ $j = 0, \ldots, m+p$ and $i = 1, \ldots, n$. The coefficient $\theta_j \in \mathbb{S}_d^v(\Delta)$ is penalized over the entire spatial domain Ω. Since θ_j can't be calculated for every $\boldsymbol{s} \in \Omega$, it is approximated by $\boldsymbol{B}^T\boldsymbol{\gamma}_j$. Then the energy functional $\varepsilon(\theta_j)$ can be approximated by $\varepsilon(\theta_j) = \varepsilon(\boldsymbol{B}^T\boldsymbol{\gamma}_j) = \boldsymbol{\gamma}_j^T\boldsymbol{K}\boldsymbol{\gamma}_j$, where $\boldsymbol{K}$ is the block diagonal penalty matrix. Using the basis expansions and the constraint matrix $\boldsymbol{H}$, the maximization problem in 2.8 can be rewritten as a minimization problem as follows

$$\min_{\substack{\boldsymbol{\gamma} \\ \boldsymbol{H}\boldsymbol{\gamma}_j = \boldsymbol{0}}} -\sum_{i=1}^{n} l_Q \left[g^{-1} \left\{ \sum_{j=0}^{m+p} \mathcal{X}_{ij}(\boldsymbol{s}_i)\boldsymbol{B}(\boldsymbol{s}_i)^T\boldsymbol{\gamma}_j \right\}, Y_i(\boldsymbol{s}_i) \right] + \frac{1}{2} \sum_{j=0}^{m+p} \tau_j \boldsymbol{\gamma}_j^T \boldsymbol{K}\boldsymbol{\gamma}_j, \tag{2.9}$$

where $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{m+p})$. A key component of solving this minimization problem is removing the constraint $\boldsymbol{H}\boldsymbol{\gamma}_j = \boldsymbol{0}$ by using the QR decomposition of $\boldsymbol{H}^T$ whose rank is the smoothness parameter v. The QR decomposition of $\boldsymbol{H}^T$ is defined as

$$\boldsymbol{H}^T = \boldsymbol{Q}\boldsymbol{R} = \begin{bmatrix} \boldsymbol{Q}_1, & \boldsymbol{Q}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_1 \\ \boldsymbol{R}_2 \end{bmatrix},$$

where $\boldsymbol{Q}$ is an orthogonal matrix and $\boldsymbol{R}$ is an upper triangular matrix. Here, $\boldsymbol{Q}_1$ represents the first v columns of $\boldsymbol{Q}$ and $\boldsymbol{R}_2$ is a matrix of zeros. Then $\boldsymbol{\gamma}_j$ can be reparametrized by using $\boldsymbol{\gamma}_j = \boldsymbol{Q}_2\boldsymbol{\gamma}_j^*$ which guarantees that $\boldsymbol{H}\boldsymbol{\gamma}_j = \boldsymbol{0}$. Now the constrained minimization problem becomes an unconstrained penalized minimization problem where

$$\min_{\boldsymbol{\gamma}^*} -\sum_{i=1}^{n} l_Q \left[g^{-1} \left\{ \sum_{j=0}^{m+p} \mathcal{X}_{ij} \boldsymbol{B}(\boldsymbol{s}_i)^T \boldsymbol{Q}_2 \boldsymbol{\gamma}_j^*, Y_i(\boldsymbol{s}_i) \right\} \right] + \frac{1}{2} \sum_{j=0}^{m+p} \tau_j \boldsymbol{\gamma}_j^{*T} \boldsymbol{Q}_2^T \boldsymbol{K} \boldsymbol{Q}_2 \boldsymbol{\gamma}_j^*, \qquad (2.10)$$

where $\boldsymbol{\gamma}^* = (\boldsymbol{\gamma}_0^{*T}, \ldots, \boldsymbol{\gamma}_{m+p}^{*T})^T$.

2.3 Implementation of GSVFM

We can estimate the spatially varying coefficients $\theta_j(\boldsymbol{s}_i)$ by minimizing the objective function in equation 2.10. For certain link functions, there is a closed-form solution for 2.10. Since this isn't the case for all link functions, the minimization of the objective function is solved by using a Penalized Iteratively Reweighted Least Squares (PIRLS) procedure (Kim and Wang, 2021). Before describing the procedure, some notation needs to be defined. Let $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T$ and $\boldsymbol{\mathcal{X}}_i = (1, \mathcal{X}_{i1}, \ldots, \mathcal{X}_{i(m+p)})$ be the response vector and the ith row vector of the design matrix. Define the matrix $\boldsymbol{\mathcal{X}}^* = (\boldsymbol{\mathcal{X}}_1^*, \ldots, \boldsymbol{\mathcal{X}}_n^*)^T$, where

$\boldsymbol{\mathcal{X}}_i^* = \boldsymbol{\mathcal{X}}_i \otimes \boldsymbol{B}^*(\boldsymbol{s}_i)$ and $\boldsymbol{B}^*(\boldsymbol{s}_i) = \boldsymbol{Q}_2^T \boldsymbol{B}(\boldsymbol{s}_i)$. Define $\zeta(\boldsymbol{\gamma}^*)$ as follows

$$\zeta(\boldsymbol{\gamma}^*) = (\zeta_i)_{i=1}^n = \left\{ \sum_{b=0}^{m+p} \mathcal{X}_{ib} \boldsymbol{B}(\boldsymbol{s}_i)^T \boldsymbol{Q}_2 \gamma_b^* \right\}_{i=1}^n .$$

Futhermore, denote the mean vector $\boldsymbol{\mu}(\boldsymbol{\gamma}^*) = (\mu_i)_{i=1}^n = \{g^{-1}(\zeta_i)\}_{i=1}^n$, the variance function matrix $V = \text{diag}\{V(\mu_i)\}_{i=1}^n$, the diagonal matrix with elements of the derivative of the link function g, $G = \text{diag}\{g'(\mu_i)\}_{i=1}^n$, and the weight matrix $\boldsymbol{W} = \boldsymbol{V}^{-1}\boldsymbol{G}^{-2} = \text{diag}[\{\boldsymbol{V}(\mu_i) g'(\mu_i)^2\}^{-1}]_{i=1}^n$.

To describe the procedure, suppose we are at the jth iteration with the current parameter estimates $\boldsymbol{\gamma}^{*(j)}$, $\boldsymbol{\mu}^{(j)} = \boldsymbol{\mu}(\boldsymbol{\gamma}^{*(j)})$, $\boldsymbol{\zeta}^{(j)} = \boldsymbol{\zeta}(\boldsymbol{\gamma}^{*(j)})$ and $\boldsymbol{V}^{(j)}$. As in any iterative procedure, we want to find the updated estimates at the $(j+1)$'th iteration. To find the estimates, we need to define the objective function in terms of the estimates at the jth iteration. The objective function at the $(j+1)$'th iteration can be defined as follows

$$L_p^{(j+1)} = \|\{\boldsymbol{V}^{(j)-\frac{1}{2}}\{\boldsymbol{Y} - \boldsymbol{\mu}(\boldsymbol{\gamma}^*)\}\}\|^2 + \frac{1}{2}\sum_{b=0}^{m+p} \tau_b \gamma_b^{*T} \boldsymbol{Q}_2^T \boldsymbol{K} \boldsymbol{Q}_2 \gamma_b^* . \tag{2.11}$$

The general procedure of an iterative algorithm is to find an approximation for a function with respect to the variables. In equation 2.11, the expectation function is unknown but we can approximate it by its first-order Taylor expansion around $\boldsymbol{\gamma}^{*(j)}$ as follows

$$\boldsymbol{\mu}(\boldsymbol{\gamma}^*) = \boldsymbol{\mu}^{(j)} + \{\boldsymbol{G}^{(j)}\}^{-1} \boldsymbol{\mathcal{X}}^* (\gamma^* - \gamma^{*(j)}). \tag{2.12}$$

Using the Taylor expansion in 2.12, $L_p^{(j+1)}$ can be approximated as follows

$$
\begin{aligned}
L_p^{(j+1)} &\approx \|\{\boldsymbol{V}^{(j)}\}^{-\frac{1}{2}}(\boldsymbol{Y}-\boldsymbol{\mu}^{(j)}) - \{\boldsymbol{V}^{(j)}\}^{-\frac{1}{2}}\{\boldsymbol{G}^{(j)}\}^{-1}\boldsymbol{\mathcal{X}}(\boldsymbol{\gamma}^* - \boldsymbol{\gamma}^{*(j)})\|^2 + \frac{1}{2}\sum_{b=0}^{m+p}\tau_b\boldsymbol{\gamma}_b^{*T}\boldsymbol{Q}_2^T\boldsymbol{K}\boldsymbol{Q}_2\boldsymbol{\gamma}_b^* \\
&= \|\{\boldsymbol{V}^{(j)}\}^{-\frac{1}{2}}\{\boldsymbol{G}^{(j)}\}^{-1}\boldsymbol{G}^{(j)}(\boldsymbol{Y}-\boldsymbol{\mu}^{(j)}) - \{\boldsymbol{V}^{(j)}\}^{-\frac{1}{2}}\{\boldsymbol{G}^{(j)}\}^{-1}\boldsymbol{\mathcal{X}}(\boldsymbol{\gamma}^* - \boldsymbol{\gamma}^{*(j)})\|^2 \\
&+ \frac{1}{2}\sum_{b=0}^{m+p}\tau_b\boldsymbol{\gamma}_b^{*T}\boldsymbol{Q}_2^T\boldsymbol{K}\boldsymbol{Q}_2\boldsymbol{\gamma}_b^* \\
&= \|\{\boldsymbol{V}^{(j)}\}^{-\frac{1}{2}}\{\boldsymbol{G}^{(j)}\}^{-1}[\boldsymbol{G}^{(j)}(\boldsymbol{Y}-\boldsymbol{\mu}^{(j)}) - \boldsymbol{\mathcal{X}}(\boldsymbol{\gamma}^* - \boldsymbol{\gamma}^{*(j)})]\|^2 + \frac{1}{2}\sum_{b=0}^{m+p}\tau_b\boldsymbol{\gamma}_b^{*T}\boldsymbol{Q}_2^T\boldsymbol{K}\boldsymbol{Q}_2\boldsymbol{\gamma}_b^*.
\end{aligned}
$$

Since $\boldsymbol{W}^{(j)} = \{\boldsymbol{V}^{(j)}\}^{-1}\{\boldsymbol{G}^{(j)}\}^{-2}$, we can define $\{\boldsymbol{W}^{(j)}\}^{\frac{1}{2}}$ to be $\{\boldsymbol{W}^{(j)}\}^{\frac{1}{2}} = \left[\{\boldsymbol{V}^{(j)}\}^{-1}\{\boldsymbol{G}^{(j)}\}^{-2}\right]^{\frac{1}{2}} = \{\boldsymbol{V}^{(j)}\}^{-\frac{1}{2}}\{\boldsymbol{G}^{(j)}\}^{-1}$. Notice that, $\boldsymbol{\mathcal{X}}(\boldsymbol{\gamma}^* - \boldsymbol{\gamma}^{*(j)}) = \boldsymbol{\mathcal{X}}\boldsymbol{\gamma}^* - \boldsymbol{\mathcal{X}}\boldsymbol{\gamma}^{*(j)} = \boldsymbol{\mathcal{X}}\boldsymbol{\gamma}^* - \boldsymbol{\zeta}^{(j)}$. Then we can rewrite the above equation as follows

$$
L_p^{(j+1)} \approx \|\{\boldsymbol{W}^{(j)}\}^{\frac{1}{2}}[\widetilde{\boldsymbol{Y}}^{(j)} - \boldsymbol{\mathcal{X}}^*(\gamma^*)]\|^2 + \frac{1}{2}\sum_{b=0}^{m+p}\tau_b\boldsymbol{\gamma}_b^{*T}\boldsymbol{Q}_2^T\boldsymbol{K}\boldsymbol{Q}_2\boldsymbol{\gamma}_b^*, \tag{2.13}
$$

where $\widetilde{\boldsymbol{Y}}^{(j)} = \{\widetilde{Y}_1^{(j)}, \ldots, \widetilde{Y}_n^{(j)}\}^T = \boldsymbol{G}^{(j)}(\boldsymbol{Y}-\boldsymbol{\mu}^{(j)}) + \boldsymbol{\zeta}^{(j)}$. Using the approximation of the objective function in 2.13, the PIRLS algorithm can be described as in Algorithm 1 below.

Algorithm 1 PIRLS Algorithm

```
Step 1: Initial values need to be chosen for the parameters $\boldsymbol{\zeta}^{(0)}$ and $\boldsymbol{\mu}^{(0)}$. The weight matrix $\boldsymbol{W}^{(0)}$ is initialized by calculating the values of $g'(\mu_i^{(0)})$ and $\boldsymbol{V}(\mu_i^{(0)})$, for $i = 1, \ldots, n$. Using these components, $\widetilde{\boldsymbol{Y}}^{(0)}$ can be calculated.
Step 2: The algorithm is executed for a specified number of iterations $J$ until the sequence of $\boldsymbol{\gamma}^*$'s converges. For each iteration j, the following procedure is implemented:
for each $0 \leq j \leq J$ do
    (i) The current estimates $\boldsymbol{\zeta}^{(j)}$ and $\boldsymbol{\mu}^{(j)}$ are used to update $g'(\mu_i^{(j)})$ and $\boldsymbol{V}(\mu_i^{(j)})$ and then $\boldsymbol{W}^{(j)}$ and $\widetilde{\boldsymbol{Y}}^{(j)}$ can be updated accordingly.
    (ii) Using the updated variables from (i), the objective function in equation 2.13 can be updated and then minimized with respect to $\boldsymbol{\gamma}^*$. The value of $\boldsymbol{\gamma}^*$ that minimizes 2.13, is set to $\boldsymbol{\gamma}^{*(j+1)}$. The solution $\boldsymbol{\gamma}^{*(j+1)}$ is used to update $\boldsymbol{\zeta}^{(j+1)} = \boldsymbol{\zeta}(\boldsymbol{\gamma}^{*(j+1)})$ and $\boldsymbol{\mu}^{(j+1)} = \boldsymbol{\mu}(\boldsymbol{\gamma}^{*(j+1)})$.
    (iii) Set $j$ to $j + 1$.
end for
```

In the following analyses, the initial values for $\boldsymbol{\mu}^{(0)}$ and $\boldsymbol{\zeta}^{(0)}$ are chosen as $\boldsymbol{\mu}^{(0)} = \boldsymbol{Y} + \mathbf{0.1}_n$ and $\boldsymbol{\zeta}^{(0)} = \{g(\mu_1^{(0)}), g(\mu_2^{(0)}), \ldots, g(\mu_n^{(0)})\}^T$, where $\mathbf{0.1}_n$ is a vector of 0.1's of size n.

The PIRLS procedure described above is dependent on the selection of the smoothing parameters $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{m+p})^T$. The purpose of the penalty is to control the smoothness of the coefficient functions. Imposing smoothness on the coefficients controls overfitting but affects the model fit. The smoothing parameters control the balance between the model fit and the variability of the functions. In general, cross-validation is used to determine the optimal smoothing parameters but it is computationally burdernsome. Therefore, the Generalized Cross-Validation (GCV) criterion is considered. Define the matrix $\boldsymbol{D}_\Lambda = \boldsymbol{\Lambda} \otimes \boldsymbol{Q}_2^T \boldsymbol{K} \boldsymbol{Q}_2$, where $\boldsymbol{\Lambda} = \text{diag}(\tau_0, \tau_1, \ldots, \tau_{(m+p)})$. Let $\boldsymbol{M}(\boldsymbol{\tau}) = \boldsymbol{\mathcal{X}}^{*T} \boldsymbol{W} \boldsymbol{\mathcal{X}}^* + \boldsymbol{D}_\Lambda$ and denote the smoothing matrix $\boldsymbol{S}(\boldsymbol{\tau}) = \boldsymbol{\mathcal{X}}^* \{\boldsymbol{M}(\boldsymbol{\tau})\}^{-1} \boldsymbol{\mathcal{X}}^{*T} \boldsymbol{W}$. Then the GCV criterion can be defined as

$$\text{GCV}(\boldsymbol{\tau}) = \frac{n^{-1} \| \boldsymbol{W}^{\frac{1}{2}} \{ \widetilde{\boldsymbol{Y}} - \boldsymbol{S}(\boldsymbol{\tau}) \widetilde{\boldsymbol{Y}} \} \|}{[n^{-1} \text{tr}\{\boldsymbol{I} - \boldsymbol{S}(\boldsymbol{\tau})\}]}. \tag{2.14}$$

The smoothing parameters $\boldsymbol{\tau}$ can then be minimized using the GCV criterion over a grid of values to select the optimal $\boldsymbol{\tau}$.

2.4 Simulations

To evaluate the performance of the proposed methodology, it is necessary to conduct simulations using generated data that represents the ideal setting for real world data and compare the model to existing functional models. The following simulations will be used the show the advantage of the GSVFM over models that don't include the spatial component.

2.4.1 Data Generation

The spatial domain is one of the key distinguishing components in the GSVFM. In real data applications, the spatial domain can't be chosen but in simulations there is no restriction. The shape of the spatial domain can be simple or complex. The real data spatial domain is used in the simulations for simplicity and to try to replicate the real-world setting as much as possible. In the agricultural data, the variables are measured across the state of Kansas. In the following simulations, the spatial domain used is all 105 Kansas counties. The spatial domain is defined as $\Omega = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_U) \subset \mathbb{R}^2$, where $U = 105$ counties and can be seen in Figure 2.1.

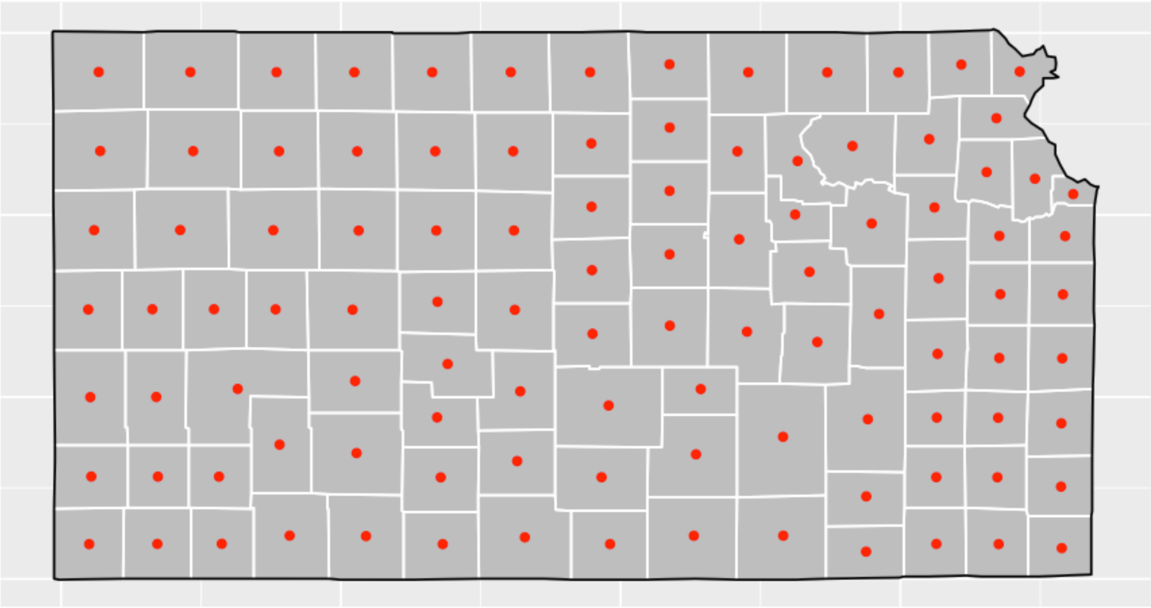


Figure 2.1: Kansas Spatial Domain with County Centroids.

In this case, the shape of the spatial domain can be considered simple since it has minimal complex boundaries. Although the method of Bivariate Splines over Triangulations is advantageous for complex spatial domains, this is not its only benefit. It's other main advantage is that it allows for the coefficient functions to vary over space as opposed to bivariate spline methods that only allowed for a single bivariate function varying over the spatial domain.

Because the agriculture dataset is separated into two distinct datasets, the data generation consists of two parts. The two main data sets that are generated are the functional dataset and the non-functional dataset. The functional dataset includes the daily minimum and maximum temparature time series. The non-functional dataset includes the agricultural yield, precipitation and the county-specific identifiers. All variables are generated with respect to each county (location) in the spatial domain. In the Kansas data, each county has at most 13 years of data. To replicate the setting of the agriculture data, we generate 10 years of data for each county which implies that $K_u = K = 10$. There are a total of $n = K * U = 100 * 105 = 1050$ observations generated for the simulations.

The multivariate spatially varying functional predictor variables $\boldsymbol{X}_k(\boldsymbol{s}_u, t) = \{X_{k1}(\boldsymbol{s}_u, t), X_{k2}(\boldsymbol{s}_u, t)\}^T,\ k = 1, \dots, K_u, u = 1, \dots, 105$, is generated by the Karhunen-Loeve decomposition using $p = 5$ eigenfunctions

$$\boldsymbol{X}_k(\boldsymbol{s}_u; t) = \sum_{r=1}^{5} \xi_{kr}(\boldsymbol{s}_u)\boldsymbol{\psi}_r(t), \tag{2.15}$$

where $\boldsymbol{\psi}_r(t) = \Big\{\cos(2\pi rt), \sin(2\pi rt)\Big\}$ for 100 time points over the time domain $[0, 1]$ and $\boldsymbol{\xi}_{kr} = \{\xi_{kr}(\boldsymbol{s}_1), \dots, \xi_{kr}(\boldsymbol{s}_U)\}$ for $r = 1, \dots, 5$ and $k = 1, \dots, K_u$. In the preprocessing step of smoothing the discrete functional data, a time domain is required. The time domain of [0,1] is used throughout the simulations which is a scaled version of the the time domain of [1,365]. Since the GSVFM allows for the coefficients to spatially vary, there is inherently spatial correlation to some degree. Therefore, it is important to simulate data based on the assumed underlying structure. To determine how much of an effect spatial correlation has on prediction performance, there are two different settings for the generation of the principal component scores. To test the effect of spatial correlation, the FPC scores are generated with spatial correlation and without. In the case of spatial correlation, the FPC score vectors are generated by $\boldsymbol{\xi}_{kr} \sim N(\boldsymbol{0}_U, \lambda_r\boldsymbol{\Sigma}(\phi))$ where $\boldsymbol{\Sigma}(\phi)$ is the Matern covariance function and $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_5) = (9, 4, 3, 2, 1)^T$ where $Var\{\xi_{kr}(\boldsymbol{s}_u)\} = \lambda_r$ for $u = 1, \dots, 105$ and $k = 1, \dots, 10$ (Park et al., 2022). The Matern correlation function describes the correlation between two points separated by d distance units and is defined as

$$\rho(d) = \frac{1}{\gamma(\kappa)2^{\kappa-1}}\Big(\frac{d}{\phi}\Big)^{\kappa} K_{\kappa}\Big(\frac{d}{\phi}\Big),$$

where κ is the smoothness parameter, ϕ the range parameter, and K_κ is the modified Bessel function of the second kind (Stein, 2012). The parameter ϕ controls the rate of decay across space. The larger ϕ is, the slower the rate of decay and thus stronger correlation. The parameters for the Matern correlation function used in the simulation are $\phi = 50/111$ and $\kappa = 1$. For the range parameter ϕ, 50 km represents the decay rate and it is scaled by the number of kilometers per unit of longitude and latitude. In the case of no spatial correlation, the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{I}$, the identity matrix. The identity matrix implies that there is no covariance. When real data is collected, there can be some measurement error introduced because of issues with the measurement devices among other factors. To account for this in the simulations, the multivariate functional predictors $\boldsymbol{X}_k(\boldsymbol{s}_u; t)$ are generated with measurement errors following $N(0, 0.2)$ (Wong et al., 2018). To generate the response variable, the scalar predictor variable and the coefficient functions need to be generated. The scalar predictor variable is generated by $Z_k(\boldsymbol{s}_u) \overset{iid}{\sim} U[0, 3]$ (Park et al., 2022). Since the GSVFM allows the coefficients to behave as smooth bivariate functions, the model should work well with complex functions. In the case where the true coefficients are constant, the GSVFM reduces to a Functional Linear Model since there is no variation across space. In this instance, the Functional Linear Model and the GSVFM should have similar performance. In the case where the coefficient function complexity increases, the GSVFM should perform better than existing functional models. To determine how much of an effect the complexity of the coefficient functions has on the results, there are three different cases for the coefficient functions, constant, linear and complex functions. The constant setting is used to verify that the model is behaving similarly to the Functional

Linear Model. The coefficient functions for the constant setting are

$$\alpha_0(\boldsymbol{s}_u) = 0.004,\ \alpha_1(\boldsymbol{s}_u) = 0.08,\ \beta_1(\boldsymbol{s}_u) = 0.16,\ \beta_2(\boldsymbol{s}_u) = -0.16,\ \beta_3(\boldsymbol{s}_u) = -0.2,$$

$$\beta_4(\boldsymbol{s}_u) = -0.104,\ \beta_5(\boldsymbol{s}_u) = 0.2.$$

For the linear setting, the functions are

$$\begin{aligned} \alpha_0(\boldsymbol{s}_u) &= \frac{1}{65}(1.5s_{1u} + 1.2s_{2u}),\ \alpha_1(\boldsymbol{s}_u) = \frac{1}{65}(2s_{1u} + 2s_{2u}), \\ \beta_1(\boldsymbol{s}_u) &= \frac{1}{65}(4s_{1u} + 6s_{2u}),\ \beta_2(\boldsymbol{s}_u) = \frac{1}{65}(3s_{1u} + 5s_{2u}), \\ \beta_3(\boldsymbol{s}_u) =& \frac{1}{65}(5s_{1u}),\ \beta_4(\boldsymbol{s}_u) = \frac{1}{65}(2s_{1u} + 5s_{2u}),\ \beta_5(\boldsymbol{s}_u) = \frac{1}{65}(2s_{1u} + 2.5s_{2u}). \end{aligned}$$

And for the complex setting we have,

$$\begin{aligned} \alpha_0(\boldsymbol{s}_u) &= \frac{1}{4}\sin\left(\frac{\pi s_{1u}}{6}\right),\ \alpha_1 = \frac{1}{8}\sin(s_{1u} + s_{2u}), \\ \beta_1(\boldsymbol{s}_u) &= \frac{1}{81}\{9 - (3 - s_{1u})^2\}\{9 - (3 - s_{2u})^2\},\ \beta_2(\boldsymbol{s}_u) = \frac{1}{4}\sin\left\{\frac{\pi(s_{1u} + s_{2u})}{6}\right\}, \\ \beta_3(\boldsymbol{s}_u) &= \frac{1}{2640}\{8 + (4 - s_{1u})^2\}\{8 - (4 - s_{2u})^2\},\ \beta_4(\boldsymbol{s}_u) = \frac{1}{4}\sin\{2\pi(s_{1u} + s_{2u})\}, \\ \beta_5(\boldsymbol{s}_u) &= \frac{1}{40}\{3 + (2 - \sqrt{|s_{1u}|})^2\}\{3 - (2 - \sqrt{|s_{1u}|})^2\}. \end{aligned}$$

In all the coefficient settings described, the coefficients were chosen such that the scale of the response variable $Y_k(\boldsymbol{s}_u)$ will be similar in all settings so that their prediction performance can be compared. Since the GSVFM is defined for multiple link functions, two different scenarios are used in the simulations to demonstrate its applicability. The first scenario assumes that $Y_k(\boldsymbol{s}_u) \sim N(\mu_k(\boldsymbol{s}_u), \sigma_\epsilon^2)$ using an identity link function. The second scenario assumes that $Y_k(\boldsymbol{s}_u) \sim \text{Poisson}(\mu_k(\boldsymbol{s}_u))$ using a log link function. Using the data and parameters generated above, the mean $\mu_k(\boldsymbol{s}_u)$ is calculated by using the GSVCM.

- In the scenario where $Y_k(\boldsymbol{s}_u) \sim N(\mu_k(\boldsymbol{s}_u), \sigma_\epsilon^2)$, the model is defined as

  $$\mu_k(\boldsymbol{s}_u) = \alpha_0(\boldsymbol{s}_u) + \alpha_1(\boldsymbol{s}_u)Z_k(\boldsymbol{s}_u) + \sum_{r=1}^{5} \xi_{kr}(\boldsymbol{s}_u)\beta_r(\boldsymbol{s}_u),\ \forall\ k = 1, \dots, 10,\ u = 1, \dots, 105.$$

  The measurement error for the response variable is based on a signal-to-noise ratio (SNR) of 3.3.

- When $Y_k(\boldsymbol{s}_u) \sim \text{Poisson}(\mu_k(\boldsymbol{s}_u))$, the model is defined as

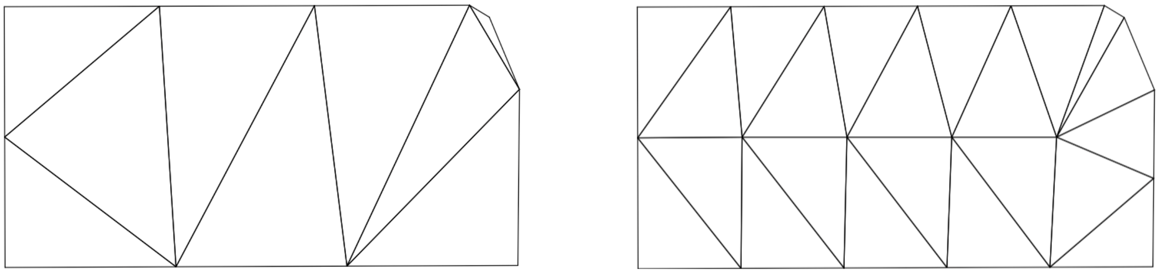
  $$\log\{\mu_k(\boldsymbol{s}_u)\} = \alpha_0(\boldsymbol{s}_u) + \alpha_1(\boldsymbol{s}_u)Z_k(\boldsymbol{s}_u) + \sum_{r=1}^{5} \xi_{kr}(\boldsymbol{s}_u)\beta_r(\boldsymbol{s}_u),\ \forall\ k = 1, \dots, 10,\ u = 1, \dots, 105.$$

  No measurement error is added in the case of a Poisson response since $\mu_k(\boldsymbol{s}_u) = \text{Var}\{Y_k(\boldsymbol{s}_u)\}$ for a Poisson random variable.

2.4.2 Smoothing Parameters

When creating a triangulation of a complex spatial domain, there are many possible triangulations. A triangulation can have anywhere from a small number of very large triangles to a large number of very small triangles. The number and size of the triangles describes the fineness of the triangulation. The number of triangles determines how many locations there are in each triangles. In the case of a small number of triangles, each triangle will contain multiple points. As the number of triangles increases, the number of points in each triangle decreases. The number of triangles has a similar effect to the number of knots for univariate splines. Since the fineness can affect the prediction performance, it is treated as a smoothing parameter. Similar to other smoothing parameters, there is a threshold where the fineness has no effect. Before determining which fineness values to test, it is beneficial to create some triangulations to get a visual of how much the triangulation is changing with the fineness. The best triangulations is where the triangles are approximately uniform in size. When this characteristic is met, each triangle will have a similar number of points. After this initial step, certain values can be chosen for performing cross-validation. The degree of the Bernstein basis polynomials also has an effect on prediction as is common in many other applications of splines. The combination of the degree and spline should be chosen through cross-validation to determine which pair of smoothing parameters results in the lowest MSPE.

To determine what effect the degree and triangulation have on the GSVFM's performance, the following settings are tested, $(d = 2,\ \text{triangles} = 9)$ $(d = 3,\ \text{triangles} = 9)$, $(d = 2,\ \text{triangles} = 22)$ and $(d = 3,\ \text{triangles} = 22)$. The two different triangulations that are used can be seen in Figure 2.2 below.



(a) Triangulation with 9 triangles.

(b) Triangulation with 22 triangles.

Figure 2.2: Triangulations of Kansas.

For the model fitting of the Generalized Spatially Varying Coefficient Model (GSVCM), there is an energy functional penalty with a corresponding penalty parameter. Given a sequence of penalty parameters, the optimal penalty parameter is chosen through generalized cross-validation at each fold.

The simulations are repeated 25 times using 5-fold CV to account for randomness across samples for each setting. For testing the smoothing parameters, only complex coefficient functions are used. To assess the prediction performance, the Mean Squared Prediction Error (MSPE) is used. For a given testing set, let $\{Y_i(\boldsymbol{s}_i)\}_{i=1}^n$ be the true response values and $\{\widehat{Y}_i(\boldsymbol{s}_i)\}_{i=1}^n$ be the corresponding predicted values. Then the MSPE calculated at each fold is defined as follows,

$$\text{MSPE} = \frac{1}{n}\sum_{i=1}^{n}\{Y_i(\boldsymbol{s}_i) - \widehat{Y}_i(\boldsymbol{s}_i)\}^2. \tag{2.16}$$

The MSPE and the training run time (per fold) averaged over the 25 repetitions can be seen in Table 2.1 below.

d	**# triangles**	**MSPE**	**Run Time (sec/fold)**
2	9	0.26	3.42
3	9	0.25	3.28
2	22	0.24	2.95
3	22	0.25	5

Table 2.1: MSPE for combinations of degree and triangles.

Based on the results in Table 2.1, there is a negligible difference in MSPE between the smoothing parameter combinations. Using only the MSPE as the deciding factor, the best combination is $d = 2$ and 22 triangles. It's important to consider that in general, the higher the degree, the more complexity the model is able to capture. Increasing the degree might lead to slightly better results but it can also make interpretation more difficult and increase computation time. The advantage of using more triangles is the ability to capture more variations across space. Although the run time is slightly longer for $d = 3$ and 22 triangles, this is an insignificant difference. Taking all these factors into account and the fact that the triangulation in Figure 2.2a includes one triangle that is significantly different in size than the others, the combination of $d = 3$ and 22 triangles is used for the remaining simulations and real data application in Section 2.5. The smoothness v is set to 1 for all simulations.

2.4.3 Gaussian Response Simulation Results

To evaluate the performance of the GSVFM, it is necessary to compare it to existing functional models. From this comparison, we will be able to determine if the GSVFM performance shows an improvement over existing models. The models that are used for comparison are the Generalized Spatially Varying Functional Model (GSVFM), the Partially Linear Functional Additive Model (PLFAM) and the Functional Linear Model (FLM). The models are defined as follows

$$\textbf{GSVFM};\ g\{\mu(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})\} = \eta_0(\boldsymbol{s}) + \boldsymbol{Z}^T(\boldsymbol{s})\boldsymbol{\alpha}(\boldsymbol{s}) + \int_{\mathcal{T}} \boldsymbol{X}^T(\boldsymbol{s}; t)\boldsymbol{\eta}(\boldsymbol{s}; t)dt,$$

$$\textbf{PLFAM};\ Y = m(\boldsymbol{X}, \boldsymbol{Z}) + \epsilon,$$

$$\textbf{FLM};\ Y = \eta_0 + \boldsymbol{Z}^T\boldsymbol{\alpha} + \int_{\mathcal{T}} \boldsymbol{X}^T(t)\boldsymbol{\eta}(t)dt + \epsilon.$$

As described in Section 2.4.1, the spatial correlation (SC) component is coming from the FPC scores through the use of the Matern correlation function. The GSVFM should perform well in the prescence of spatial correlation. Therefore, it is of interest to see how the model's performance changes depending on different combinations of spatial correlation and function complexity, the following six settings are used: (No SC, Constant), (SC, Constant), (No SC, Linear), (SC, Linear), (No SC, Complex), (SC, Complex). To account for randomness, each setting is repeated 25 times with different seeds to generate the data. For each data generation, 5-fold Cross-Validation MSPE is calculated and the training run time (for each fold) is measured for the GSVFM, PLFAM and FLM. For each model setting, the MSPE and run time are averaged across the 25 repetitions. The MSPE used is defined in 2.16. The results for each setting can be seen in Tables 2.2-2.4

Model	**SC**	**Function**	**MSPE**	**Run Time (sec/fold)**
GSVFM	No	Constant	0.17	8.15
PLFAM	No	Constant	0.18	108.03
FLM	No	Constant	0.19	6.81
GSVFM	Yes	Constant	0.16	7.02
PLFAM	Yes	Constant	0.17	134.77
FLM	Yes	Constant	0.18	4.14

Table 2.2: Gaussian - Constant Functions.

Model	**SC**	**Function**	**MSPE**	**Run Time (sec/fold)**
GSVFM	No	Linear	0.13	4.69
PLFAM	No	Linear	0.55	56.005
FLM	No	Linear	0.55	4.46
GSVFM	Yes	Linear	0.13	12.009
PLFAM	Yes	Linear	0.54	64.29
FLM	Yes	Linear	0.54	4.38

Table 2.3: Gaussian - Linear Functions.

Model	**SC**	**Function**	**MSPE**	**Run Time (sec/fold)**
GSVFM	No	Complex	0.23	6.73
PLFAM	No	Complex	0.63	115.84
FLM	No	Complex	0.62	11.56
GSVFM	Yes	Complex	0.24	4.71
PLFAM	Yes	Complex	0.65	109.39
FLM	Yes	Complex	0.65	6.43

Table 2.4: Gaussian - Complex Functions.

The results for the constant setting are presented in Table 2.2. When the true coefficient functions are constant, there is no significant difference in MSPE between the models. This outcome aligns with expectations, as a spatially dependent model offers no advantage in this scenario. Consequently, the GSVFM behaves similarly to the PLFAM and FLM. In the presence of spatial correlation, the results remain consistent since the coefficient functions are constant. This constant setting was used to verify that the GSVFM operates correctly, but the more interesting case involves coefficient functions that vary across space.

Table 2.3 shows the results for the linear function setting. Here, the true coefficient functions are bivariate linear functions, and we observe a significant difference in MSPE. The competing models' MSPE remains the same and is about four times larger than that of the GSVFM. As expected, the GSVFM shows superior performance when the coefficient functions vary spatially. While it is reassuring that the GSVFM works well for linear functions that change over space, in practice, more complex functions are often encountered. As these functions become more intricate, it becomes increasingly challenging for the model to accurately recover them. The true power and generalizability of the GSVFM can be assessed using more complex functions.

Table 2.4 presents the results when the true coefficient functions are complex. The behavior in this setting is similar to the linear case, with the MSPE for the PLFAM and FLM being equivalent but approximately 2.5 times higher than the GSVFM MSPE, under both spatial correlation settings. The results from both the linear and complex settings highlight the advantages of using the GSVFM over the PLFAM and FLM when spatial correlation is present. Furthermore, even in the absence of spatial correlation, the GSVFM continues to perform well.

In Tables 2.2-2.4, the MSPE values reported are averages across 25 data generations, with five folds per generation. Examining all 125 MSPE values for each setting wouldn't be practical, so using the average provides a general sense of overall model performance. However, relying solely on the mean may not fully capture performance, as it can be influenced by outliers. To gain a clearer understanding of the distribution of MSPE across all data generations, we can generate box plots for each model. Since the complex setting is of greatest interest, the box plots will focus on the complex functions and spatial correlation setting.

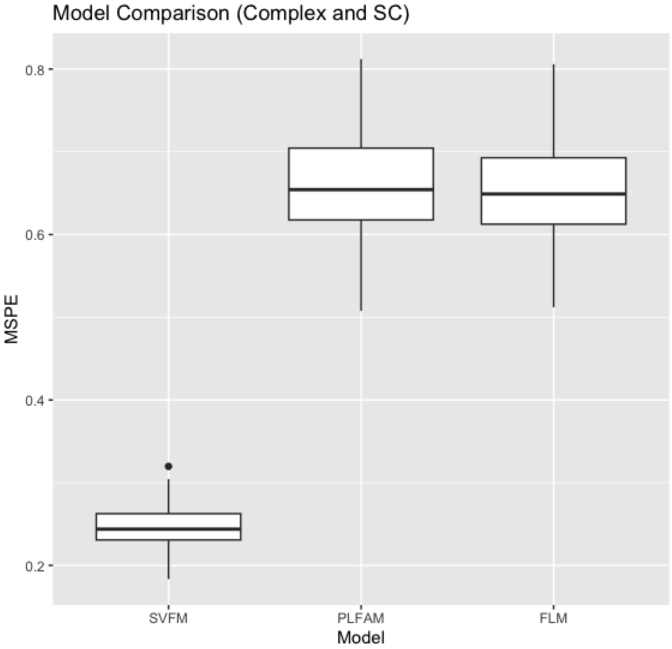


Figure 2.3: Boxplots of the MSPE's.

The boxplots in Figure 2.3 clearly reinforce the findings from Table 2.4, but a key observation is that the spread of the MSPE for the PLFAM and FLM is significantly wider than for the GSVFM. This indicates that, even across different samples, the GSVFM's performance is more stable compared to the PLFAM and FLM. This is a critical aspect of the GSVFM's performance, as in real-world applications, there can be thousands of potential samples. It further highlights the broad applicability of the GSVFM.

The MSPE is just one method for evaluating the GSVFM, but since this is a simulation, we can also assess how well the model recovers the true coefficient functions. If the model is accurate, the estimated coefficient functions should closely match the true ones. Heat maps can be used to visualize this recovery. Since the estimated coefficient functions are discrete—limited to 105 spatial locations in the domain—they are not continuous over space. To better capture the behavior of the coefficient functions across the spatial domain, a grid of values is created, and the estimated coefficient value at each point is calculated. This approach produces a smooth heat map over the spatial domain. By comparing the heat maps of the true and estimated coefficient functions, we can assess the accuracy of the GSVFM's estimations. Figures 2.4-2.13, display the recovery of the true coefficient functions in both the linear and complex settings.

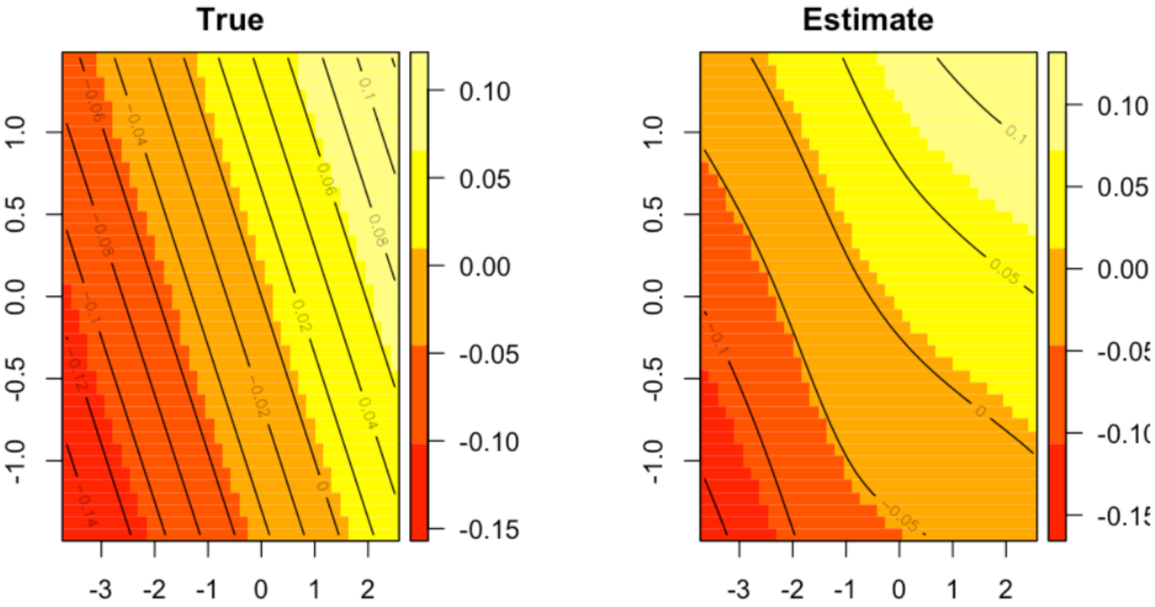


Figure 2.4: α_1 Linear Heatmaps.

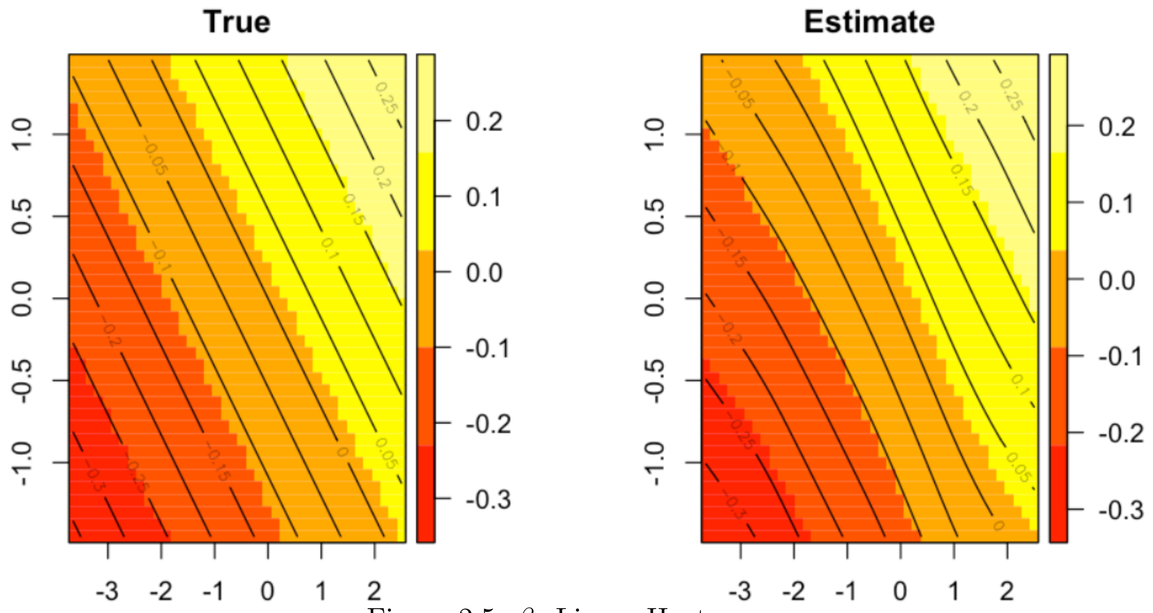


Figure 2.5: β_1 Linear Heatmaps.

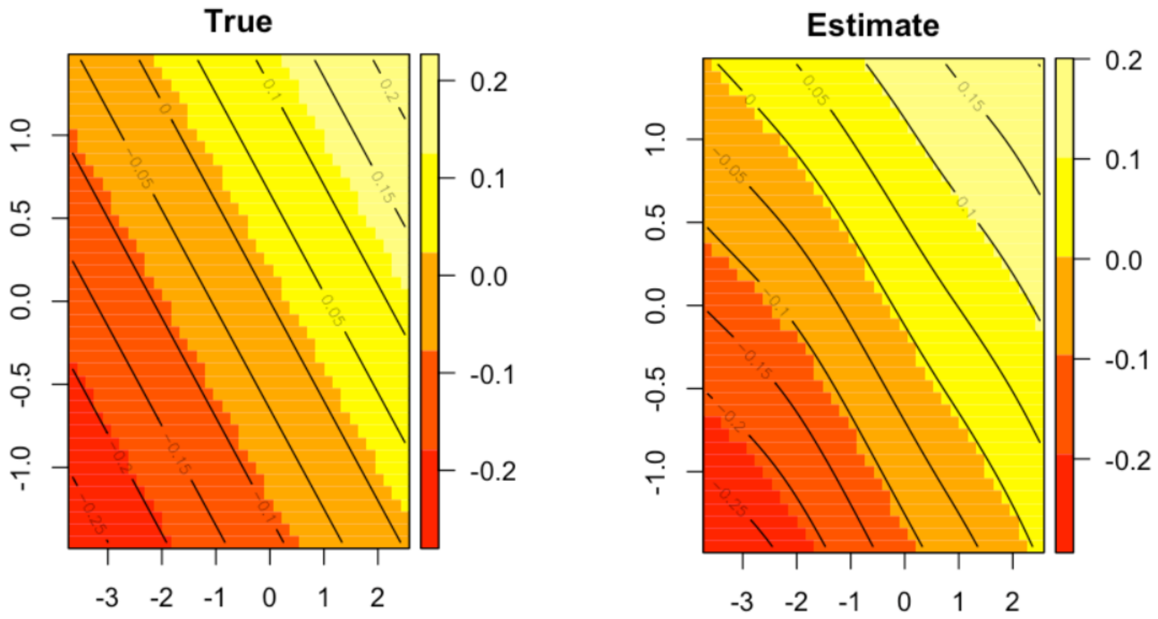


Figure 2.6: β_2 Linear Heatmaps.

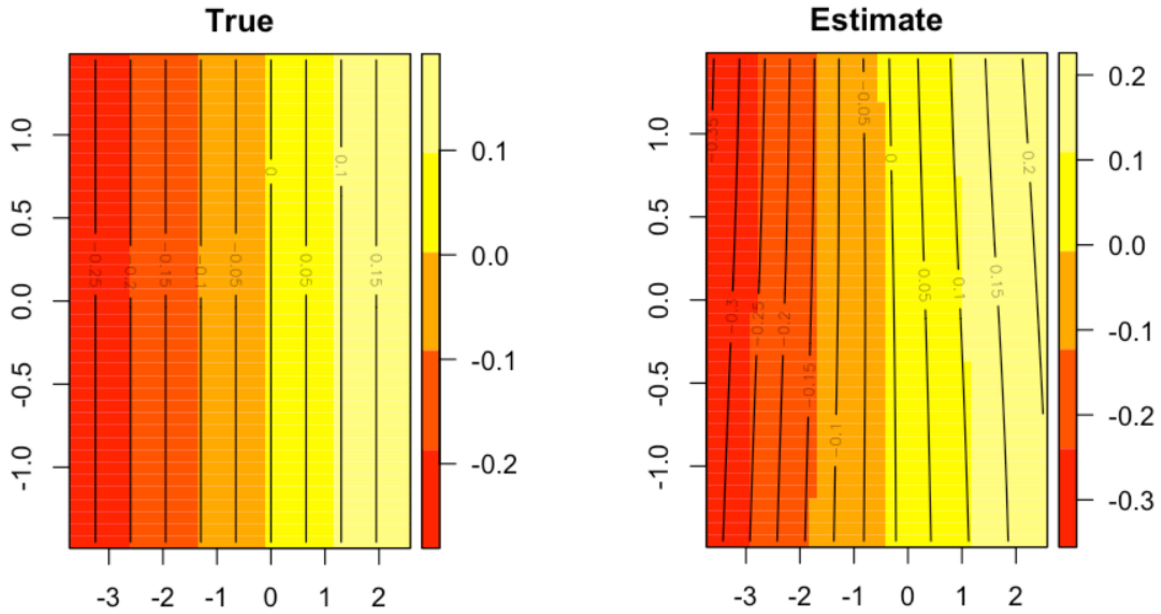


Figure 2.7: β_3 Linear Heatmaps.

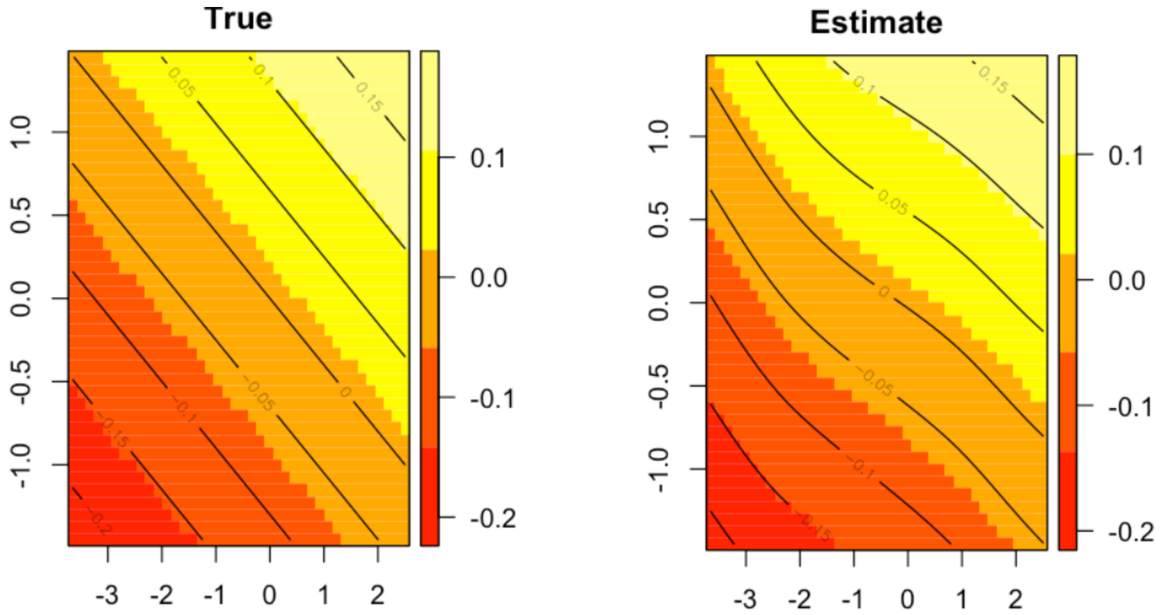


Figure 2.8: β_4 Linear Heatmaps.

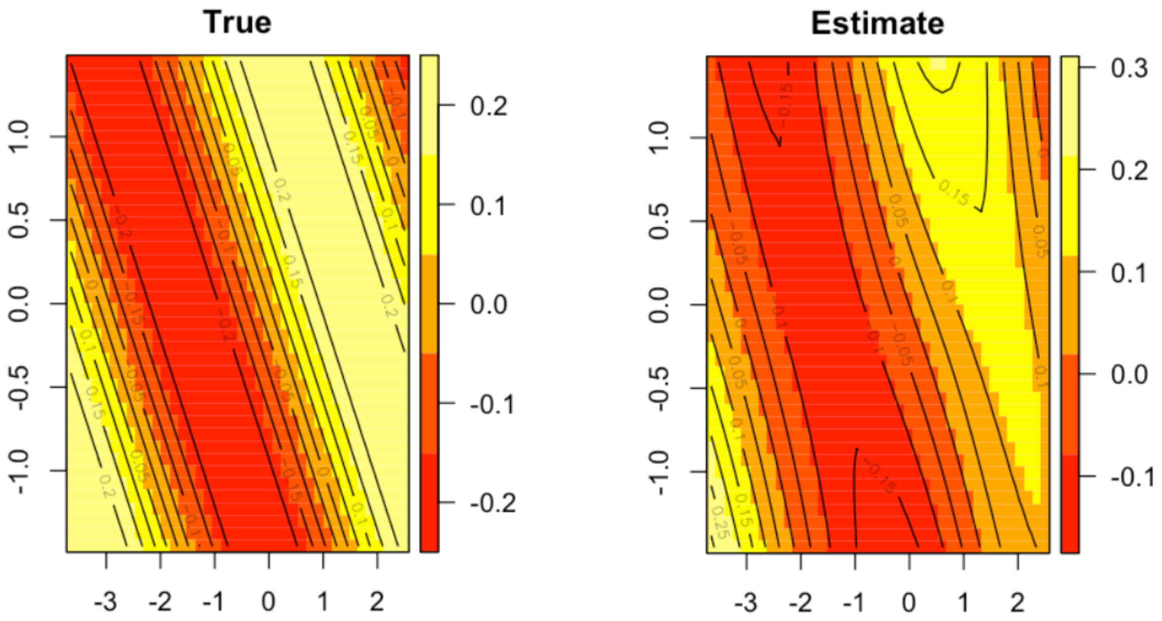


Figure 2.9: α_1 Complex Heatmaps.

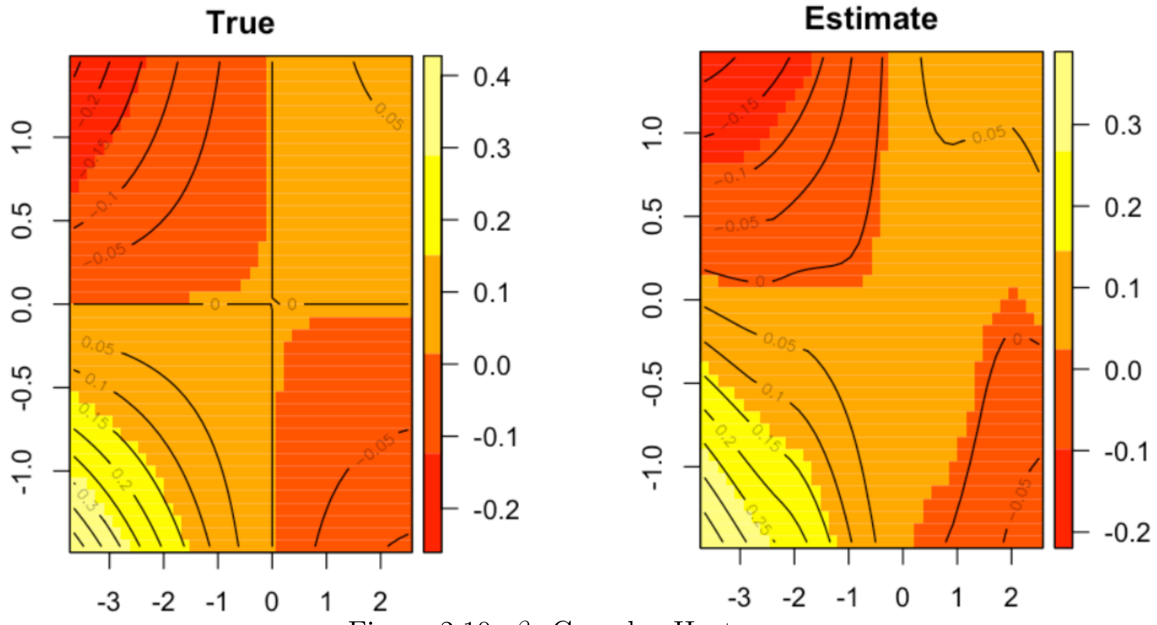


Figure 2.10: β_1 Complex Heatmaps.

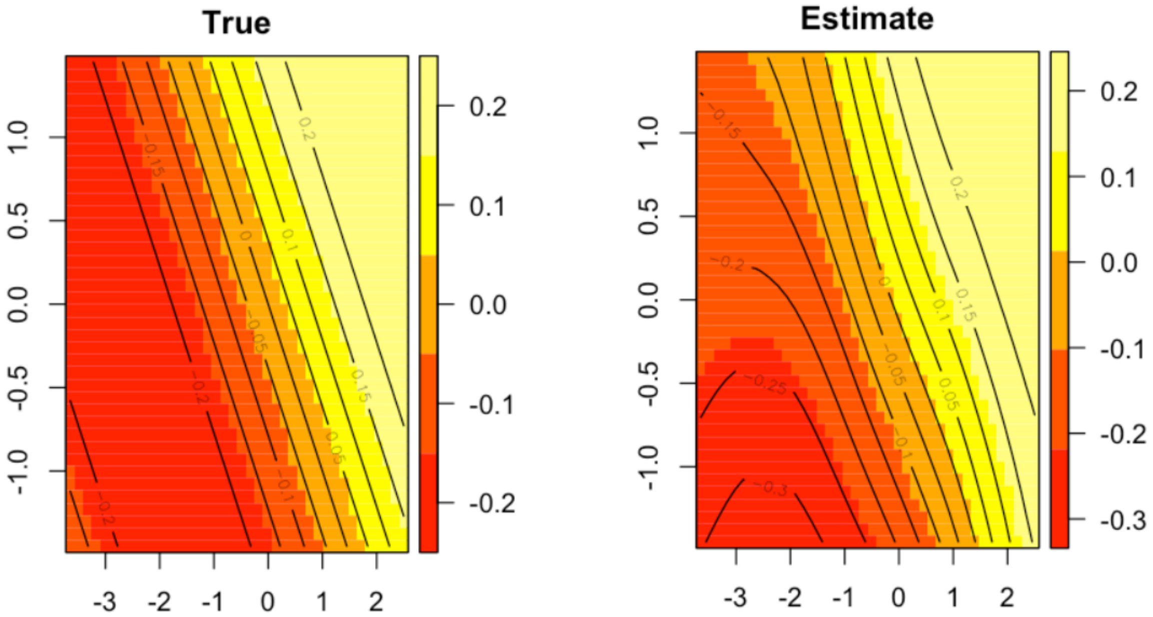


Figure 2.11: β_2 Complex Heatmaps.

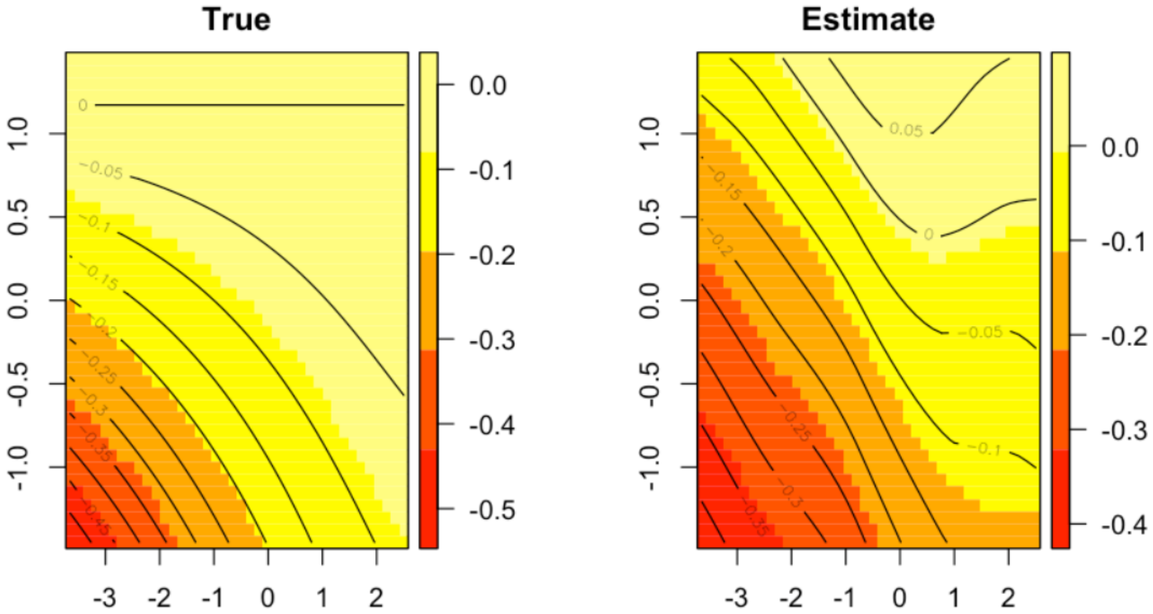


Figure 2.12: β_3 Complex Heatmaps.

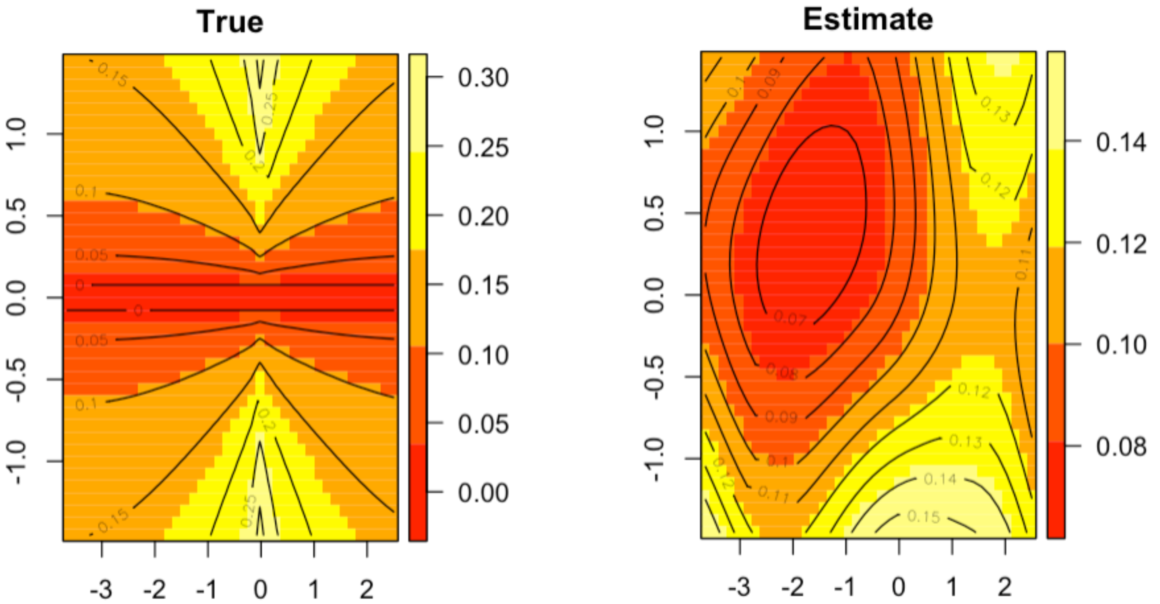


Figure 2.13: β_5 Complex Heatmaps.

At a glance, the heatmaps show that the GSVFM performs quite well in recovering the true coefficient functions, especially in the linear setting, where the estimated coefficients closely follow the behavior of the true functions. Although there is a small discrepancy between the true and estimated linear functions, it is minor and does not significantly affect the overall trend. Only in the case of β_4 in Figure 2.8 do we see slightly more deviation compared to the other coefficients. It's possible that increasing the sample size would bring the estimated linear functions even closer to the true values. The relative simplicity of the linear functions, compared to the complex functions, clearly influences the accuracy of the recovery, as seen in Figures 2.4-2.8. While the recovery of the linear functions is reassuring, the more critical scenario involves the complex functions.

Figures 2.9-2.13, display the heatmaps for the complex functions, where the difference between the true and estimated functions is more noticeable than in the linear case. Some complex functions, such as α_1, β_1, and β_2 in Figures 2.9-2.11 are estimated more accurately than others. However, for β_3 in Figure 2.12 and β_5 in Figure 2.13, the deviation between the true and estimated functions is more pronounced. For instance, in Figure 2.13, the true β_5 heatmap shows a very complex pattern, and the GSVFM has greater difficulty producing a similar estimate. While the estimated heat map for β_3 captures the general trend of the true function, it could be more precise. The higher order FPC scores, β_4 and β_5, have smaller values and thus would had have less of an effect on the response. This could possibly explain a lesser recovery of the original function for higher order FPC coefficient functions. Overall, based on the comparison between the true and estimated coefficient function heatmaps, the GSVFM does a reasonable job of recovering the true functions.

2.4.4 Poisson Response Simulation Results

One of the GSVFM's many advantages, is that it's defined for different link functions. This makes it widely applicable for different kinds of applications where the response variable might not follow a Normal distribution. The GSVFM extends the class of Generalized Linear Models (GLM) to multivariate spatially varying functional data. One of the possible link functions is the log-link function where the goal is to model a Poisson distributed response variable. When the response variable is distributed as Poisson, this means that the response represents the number of times an event occurs in a particular time interval. The model is defined as in Section 2.4.3 but now the response $Y_k(\boldsymbol{s}_u) \sim \text{Poisson}(\mu_k(\boldsymbol{s}_u))$. The data generation is the same as described in Section 2.4.1 instead of the addition of measurement error to the response as stated at the end of Section 2.4.1. The two models used for comparison are the GSVFM and the Generalized Functional Linear Model (GFLM) which are defined as follows

$$\mathbf{GSVFM},\ g\{\mu(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{s})\} = \eta_0(\boldsymbol{s}) + \boldsymbol{Z}^T(\boldsymbol{s})\boldsymbol{\alpha}(\boldsymbol{s}) + \int_{\mathcal{T}} \boldsymbol{X}^T(\boldsymbol{s};t)\boldsymbol{\eta}(\boldsymbol{s};t)dt,$$

$$\mathbf{GFLM},\ g\{\mu(\boldsymbol{X},\boldsymbol{Z})\} = \eta_0 + \boldsymbol{Z}^T\boldsymbol{\alpha} + \int_{\mathcal{T}} \boldsymbol{X}^T(t)\boldsymbol{\eta}(t)dt,$$

where g is a log link function. The simulation results for the constant, linear and complex function settings with respect to spatial correlation, can be in Tables 2.5-2.7

Model	**SC**	**Function**	**MSPE**
GSVFM	No	Constant	1.66
GFLM	No	Constant	1.71
GSVFM	Yes	Constant	1.64
GFLM	Yes	Constant	1.73

Table 2.5: Poisson - Constant Functions.

Model	**SC**	**Function**	**MSPE**
GSVFM	No	Linear	1.54
GFLM	No	Linear	3.75
GSVFM	Yes	Linear	1.55
GFLM	Yes	Linear	3.75

Table 2.6: Poisson - Linear Functions.

Model	**SC**	**Function**	**MSPE**
GSVFM	No	Complex	2.12
GFLM	No	Complex	2.69
GSVFM	Yes	Complex	1.93
GFLM	Yes	Complex	3.34

Table 2.7: Poisson - Complex Functions.

As mentioned in Section 2.4.3, the constant setting is used to verify that the GSVFM is working correctly. In Table 2.5, the MSPE in all settings are fairly close to each other with the GSVFM resulting in a slightly smaller MSPE. The linear and complex settings offer a different perspective of the GSVFM's performance. In Table 2.6, the GSVFM outperforms the GFLM significantly since the GFLM MSPE is around 2.5 times greater than the GSVFM. Looking at the complex setting in Table 2.7, there is similar behavior in the spatial correlation case. When spatial correlation is present, the GFLM MSPE is approximately 1.75 times greater than the GSVFM MSPE. These results confirm what we see in the linear setting but what is interesting is the smaller difference in MSPE under the complex and no spatial correlation setting. In this setting, the GSVFM still outperforms the GFLM but not by as much in the spatial correlation case. A possible explanation for this, is that since the complex functions are periodic, the GFLM may have some cancelation of errors. The results displayed in this section prove that the GSVFM has better prediction performance over the GFLM in the case of a log-link function.

2.5 Application

The motivating application for this research is crop yield prediction for corn yield using weather data and other agricultural variables collected from the Midwest region including the states of Kansas, Missouri, Indiana, Illinois and Iowa over a time period of 21 years from 1999-2019 across 513 counties. For each county, we observe spatial replicates of the average corn yield per acre (measured in bushels per acre), average yearly precipitation, proportion of irrigated land and the minimum and maximum temperature time series observed over the time domain $\mathcal{T} = [0, 365]$. For simplicity, the rescaled time domain of $\mathcal{T} = [0, 1]$ is used throughout this section. The number of available years of data for each county varies across all counties. The data could be missing for all variables for a single year or a combination of variables. There are different ways of handling missing data and the approach used needs to be applied with care. More details concerning missing data handling is covered in Appendix A. The spatial domain Ω is defined to be the entire boundaries of the five Midwest states.

The agriculture industry in the US contributes 136.1 billion dollars to national GDP and 11% of total employment (USDA, 2020a). Agriculture is clearly a significant industry for GDP, employment and food security. The US agriculture industry not only provides food security for the US, it also provides global food security as it is a leading producer in the world. Because of this, obtaining reliable crop-yield predictions is of even more importance. There are different agricultural regions throughout the US, with the Midwest being one of the largest. The Midwest happens to be one of the worlds largest crop production areas with 127 million acres and produces 85% of US corn and soybean

(USDA, 2020b) which only emphasizes how important the US agricultural industry is to the world. Corn is an important crop in the Midwest since it is used for many different purposes such as human food, livestock, and biofuel (ethanol). The Midwest is a prime region for growing corn since the soils are deep, fertile, and rich in organic material and nitrogen, and the land is relatively level. Crop-yield can be affected by a multitude of different variables. There might be certain variables that are very difficult to measure but through multiple research studies, it has been discovered that climate variables such as temperature and precipitation have a significant impact on all plant growth and therefore affect agricultural productivity. One example that provides evidence for this claim is in Ray et al. (2015) which states that approximately 60% of corn yield variability and 36% of soybean yield variability in the Midwest can be explained by climate variability. As one can imagine, the problem of crop-yield prediction is not new and there has been a lot of development in predictive models by crop scientists. Some of these predictive models include, process-based simulation models under comprehensive mechanism processes (Guan et al., 2017; Jones et al., 2017; Peng et al., 2018), empirical statistical crop models based on historical climate observations and yield measurements (Prasad et al., 2006; Gornott and Wechsung, 2016; Kern et al., 2018) and machine learning methods in recent studies (Kang et al., 2020; Van Klompenburg et al., 2020). Although there are differences in these methods, one of their commonalities, is that they all use a summary of climate data such as using monthly or yearly averages instead of complete trajectories. Beyond crop-yield prediction for the importance of sustaining agriculture, prediction models can help us understand the potential impacts of weather patterns on crop productivity which is of great interest to agriculture and the economy as climate change becomes a fast approaching challenge for humanity. The ultimate goal in this application is to use the spatially varying functional model to perform crop yield prediction given a new observation.

In the following section, the aim is to first provide some data exploration in Section 2.5.1 and the remaining sections cover different components of the model fitting. Sections 2.5.2 and 2.5.3 focus on the results for the Midwest agriculture data. The sample size and size of the spatial domain could play a part in the model performance. To contrast the effect of these two characteristics, two separate analyses are performed, one on the Kansas dataset, and one on the complete Midwest dataset. In Section 2.5.4, a Bootstrap Hypothesis Testing procedure is applied to the Midwest data to provide evidence for non-stationarity.

2.5.1 Data Exploration

In any data analysis, exploring the data before fitting different models is important since it delivers useful information that could be of importance in model fitting. Before fitting the model to the Kansas and Midwest data, it is important to get some sense of the behavior of the data. For simplicity, the data exploration is conducted only for the Kansas data but similar behaviors arise for the Midwest data.

The daily minimum and maximum time series data are the functional data in the model that we are using in our model through FPCA. To demonstrate the behavior of the temperature time series, we can plot the behavior for one Kansas county in 1999 which can be seen in Figure 2.14.

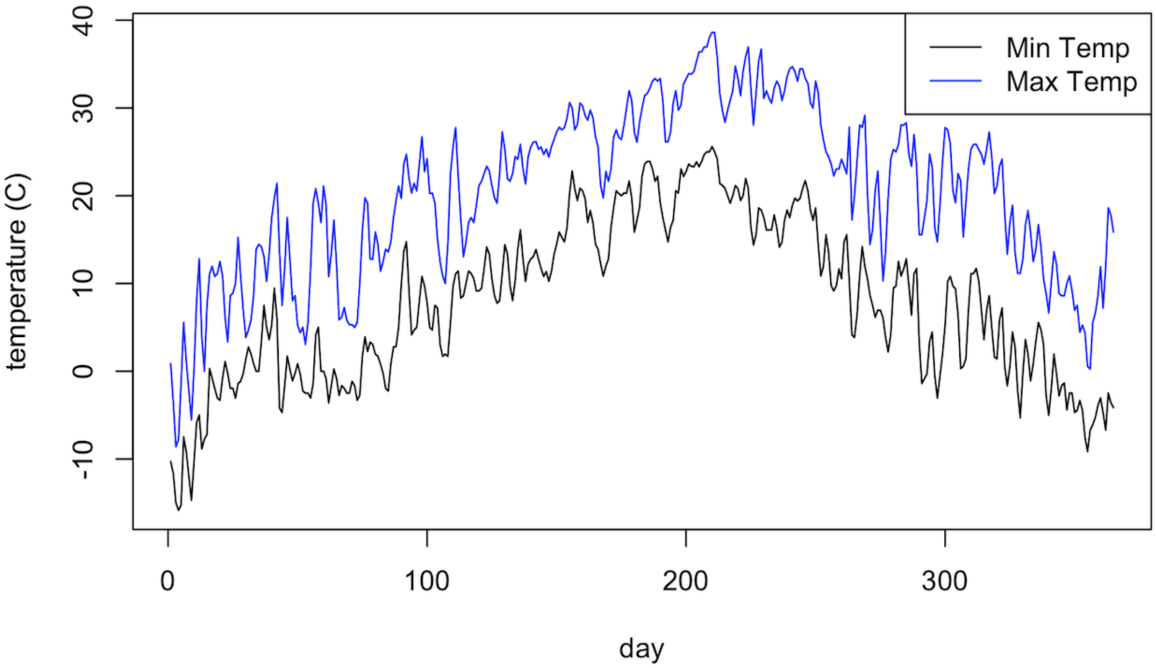
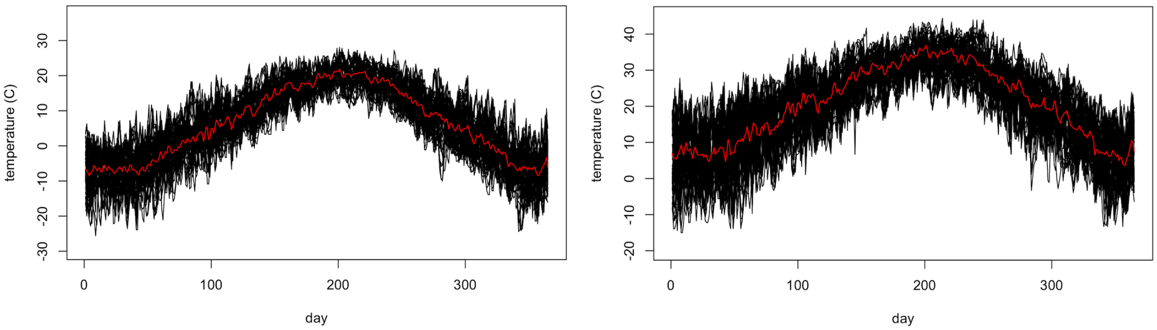


Figure 2.14: Allen County Temperature Curves for the year 1999.

The minimum and maximum temperature fluctuate by about a couple degrees for most days, but for some, it fluctuates between 10-20 degrees from one day to another. As we would expect, the temperature curves start with colder temperatures in the winter and shift to warmer temperatures in the summer and then decreases again from summer to fall. This is common behavior in any location since there are unique seasons for each location. The curves seem to be some kind of linear combination of sine and cosine functions. Figure 2.14 shows us the typical behavior of one county's minimum and maximum time series data but there exists some variation among counties and years. To get a sense of the behavior across different counties and years, 50 randomly selected minimum and maximum temperature time series were for different combinations of county and year. The plot of the time series for these counties and years can be seen in Figure 2.15.

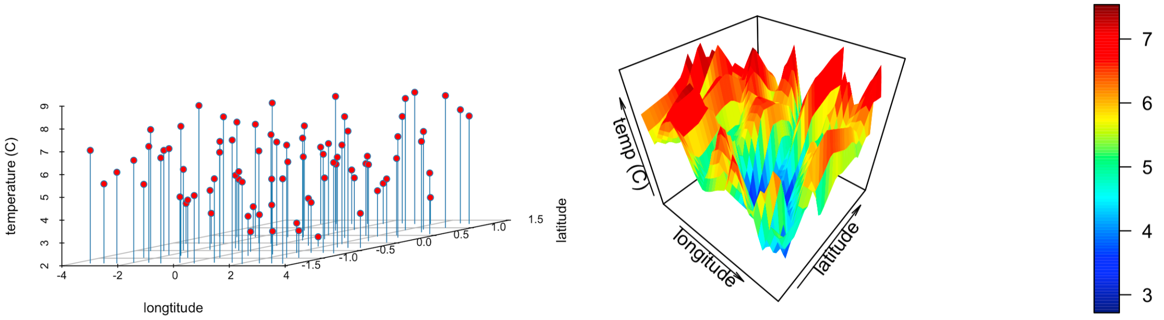


(a) 50 randomly selected minimum temperature curves with the mean curve in red.

(b) 50 randomly selected maximum temperature curves with the mean curve in red.

Figure 2.15: Minimum and Maximum Temperature Curves.

In Figure 2.15, there is some variation across counties and years, especially among the maximum temperature curves. The variation in both plots relative to the mean is anywhere between 10-15 degrees Celsius. Although there is some variation among the curves, which is to be expected, for the most part all curves follow a similar kind of behavior to what was seen in Figure 2.14. There is similar behavior in the curves because all counties are coming from the same region. This does not mean there aren't climate differences but to some degree there are similarities in climate. Figure 2.15 gives one view of the variation across counties. A useful perspective is to get an idea of how the average minimum and maximum temperature change over counties, by creating a 3D scatterplot and surface which can be seen in Figure 2.16.

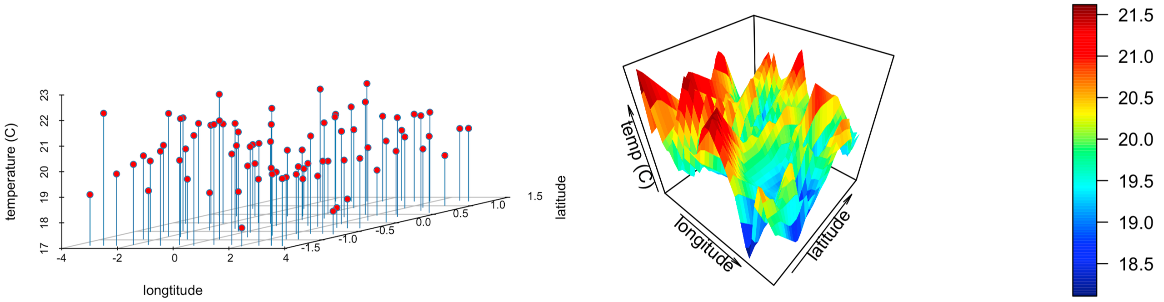


(a) 3D scatterplot of average minimum temperature across counties.

(b) 3D surface of average minimum temperature across counties.

Figure 2.16: Average Minimum Temperature 3D plots.

In Figure 2.16a, the variation among the average minimum temperature from county to county can be seen more distinctly. The averages vary by only a couple degrees since only the average is reported per county. Counties that are closer to each other have similar average minimum temperatures which can be seen more clearly in Figure 2.16b where specific regions are separated by color. The area of Kansas with the highest average minimum temperatures is in red and an area with lowest averages in blue. For the average maximum temperatures, the same plots can be created as seen in Figure 2.17.



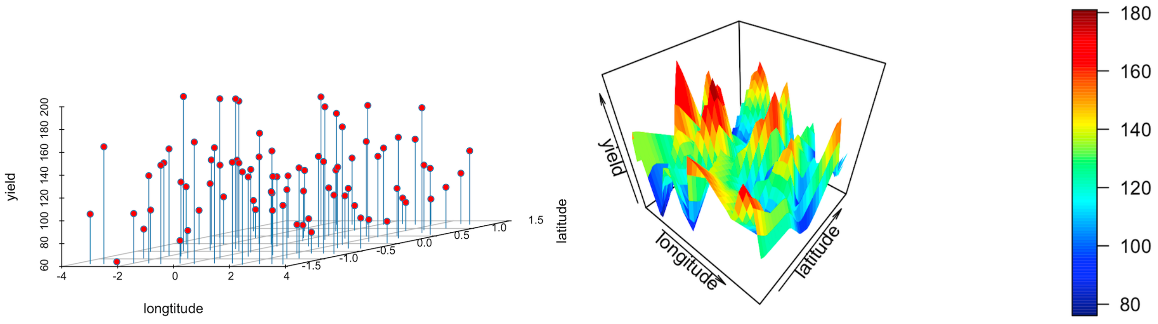
(a) 3D scatterplot of average maximum temperature across counties.

(b) 3D surface of average maximum temperature across counties.

Figure 2.17: Average Maximum Temperature 3D plots.

In Figures 2.17a and 2.17b, there also seems to be quite a bit of variation in the average maximum temperatures across counties but the spread of low and high average maximum temperatures is slightly different than the plots for average minimum temperatures. Unlike the surface plot for average minimum temperatures in Figure 2.16b, in the surface plot in Figure 2.17b, the low and high average maximum temperatures occur in different locations. The region of high averages is different and now there's some low averages in the center of the plot. There is a similar pattern in the average maximum temperatures for counties that are nearby one another. In Figure 2.17b, we can also see a clear separation of regions by the different colors.

The plots in Figures 2.16 and 2.17 give clear evidence that although there are similarities in climate across Kansas, there are distinct regions with unique climates that could have different effects on agriculture. The average temperature plots give us an understanding of where the temperature could have a greater effect on yield but to get a clearer picture, we can look at the same plots for the average yield across counties in Figure 2.18.



(a) 3D Scatterplot of average yield across counties.

(b) 3D surface of average yield across counties.

Figure 2.18: Average Yield 3D Plots.

In Figures 2.18a and 2.18b, the yield varies significantly from county to county with neighboring counties having similar yields. Looking at the surface plot, the yield changes between high and low as we move across the state of Kansas. This kind of variation across counties tells us that the yield could be affected by other variables than just temperature. The average yield could be affected by variables such as the temperature curves, precipitation, proportion of irrigated land among other variables.

Over the past several decades, there have been major advancements in genetics and production technology. Due to these advancements, corn yields have significantly increased. The variation of corn yields across 21 years could be significant and affect the results even though a proportion of the variation in corn yields isn't explained by the predictor variables. Figure 2.19 shows that the average yield per year (across all locations) is affected by the advancements in agriculture. Due to this trend, the yield for a given year is centered by its corresponding average to eliminate this effect.

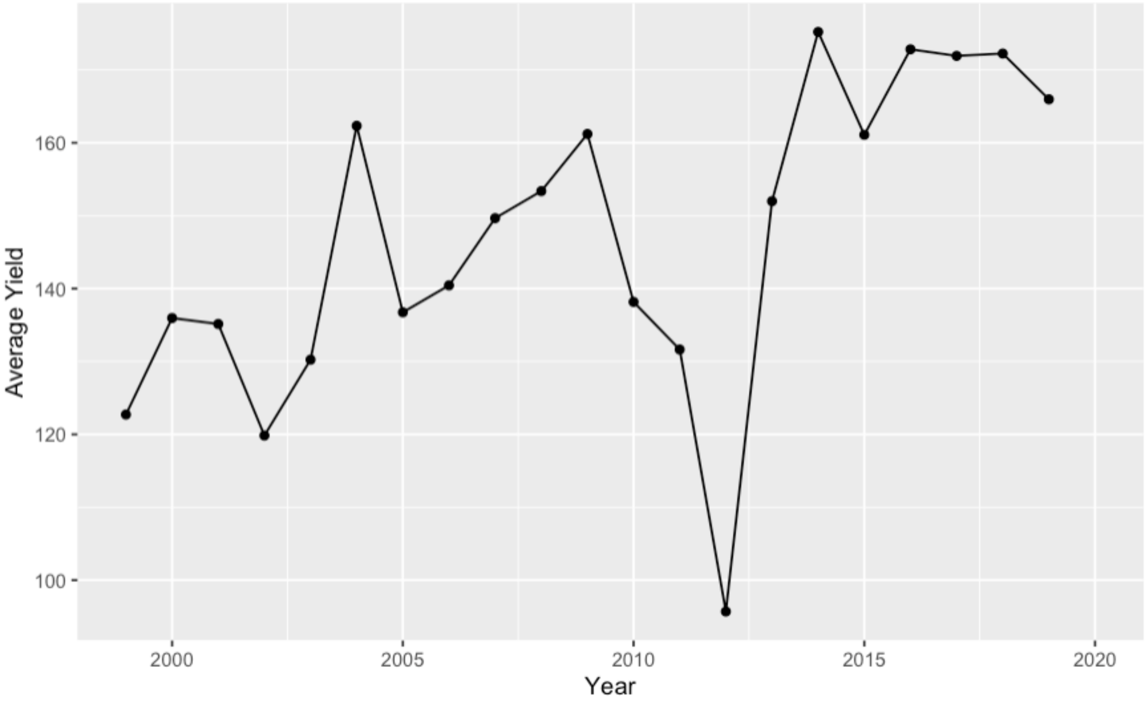


Figure 2.19: Average Yield vs. Year (Midwest).

2.5.2 Kansas Data

The Kansas data used in the following analysis is a subset of the original Midwest data. There are 105 counties across Kansas where the yield, precipitation, irrigated land and temperature data are available from 1999-2019. For simplicity, only the years 1999-2011 are used in this analysis with a total of 1218 observations. The spatial domain of Kansas is the same domain that was used in the simulations which can be seen in Figure 2.1. The irrigated land is a variable that measures the average acreage of irrigated land in a given county in thousands. Since this variable has a large scale, it is convenient to scale it to some degree so that it doesn't skew the model coefficients. The total area is also available and we can use it to define a new variable. The proportion of irrigated area is calculated as the ratio between irrigated area and total area so that no information is lost but the new variable is now scaled to be between 0 and 1. As mentioned in Section 2.4, the basis function degree and triangulation fineness plays an important part in the GSVFM's performance. Even though the simulated data was used for cross-validation of these smoothing parameters, it is reasonable to use the same combination in the Kansas analysis. Therefore, the combination used is $d = 3$ and 22 triangles. The smoothness v is set to 1 for the remainder of this section. To fit the GSVCM introduced in Section 2.2, cross-validation needs to be performed for the penalty parameter τ in equation 2.8. The GCV criterion in 2.14 is used to choose the optimal penalty parameter τ at each fold throughout the remainder of this section.

One of the most important smoothing parameter that significantly affects prediction performance is the number of Functional Principal Component (FPC) scores. The number of FPC scores included in the model corresponds to the percentage of variation explained by them. The more FPC scores that are used, the higher percentage of variation that is explained. There is a threshold where the number of scores doesn't change the models performance. The more FPC scores used implies that we are including more of the variation in the model but, this could also lead to overfitting just as in any model. It is necessary to determine the number of scores that results in the best prediction performance. To evaluate the performance, initially a grid of values from 0.15-0.90 by increments of 15%, and 0.95 was used. This was done to get an understanding of the behavior to decide where a finer grid of values should be used to find the minimum test error. The cross-validation was performed using 9 iterations of 5-folds. Both the train and test errors are reported, the train error is used to understand the models ability to fit the data, but the number of FPC scores is chosen based on the test error. The average train and test MSE (across all folds) results can be seen in Figure 2.20.

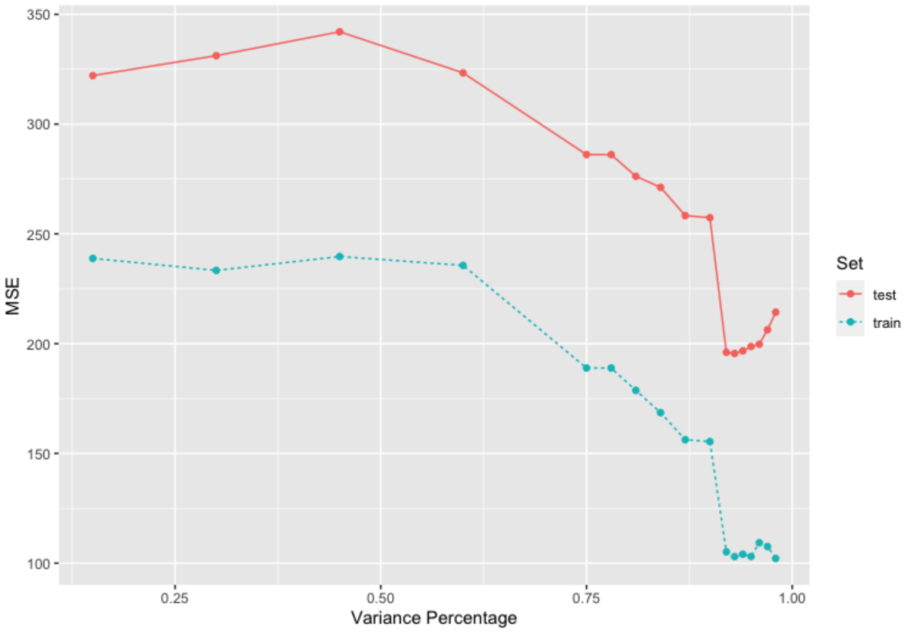


Figure 2.20: MSE vs. Variance Percentage CV (Kansas).

At a glance, it is clear that the MSE doesn't change significantly between the percentages of 15% to 60%. There is a significant drop in the MSE when the variance percentage increases from 60% to 70%. The MSE decreases pretty consistently up until it reaches about 90%. The jump from 90% to 92% is the most drastic, with the MSE decreasing from 155.40 to 103.02 for train and 257.37 to 196.02 for test. There is a small decrease from 92% to 93% with the MSE decreasing from 105.15 to 104.011 for train and 196.02 to 195.52 for test. As the percentage increases from 92%, training error stays almost the same, while test error increases significantly. That's where adding more scores starts having a negative effect. The train and test curves both show similar behavior up until the highest variance percentages. The model is fitted with the training set so it will generally be more biased and result in a better fit than the testing set. The optimal variance percentage is chosen based on the test MSE since the emphasis is on our prediction performance. Given this, the variance percentage of 93% (16 FPC scores) results in the lowest MSE of 195.52 and is used in the remainder of this section.

In the data preparation stage, the smoothing of the discrete functional data is done for each fold separately to ensure that the smoothing parameters are unique to the respective fold. The functional dataset contains more observations than the non-functional dataset. Only observations that are in both datasets are used for model fitting but including more observations in FPCA can increase it's accuracy. Therefore, all the functional training data is used to perform FPCA. It is a common procedure to center the functional data before FPCA. In general, the global mean is used for centering but since the data is spatially varying, the location-specific means differ from each other. Because the means differ across space, it is more sensible to center the functional data by its location-specific mean. In Section 2.5.1, it was shown that the yield should be centered. The yield is centered according to the average yield per year across all counties. The GSVFM is evaluated by performing 5-fold cross-validation across 9 repetitions. This is done to increase the number of validation sets to show how the model can handle variability across datasets.

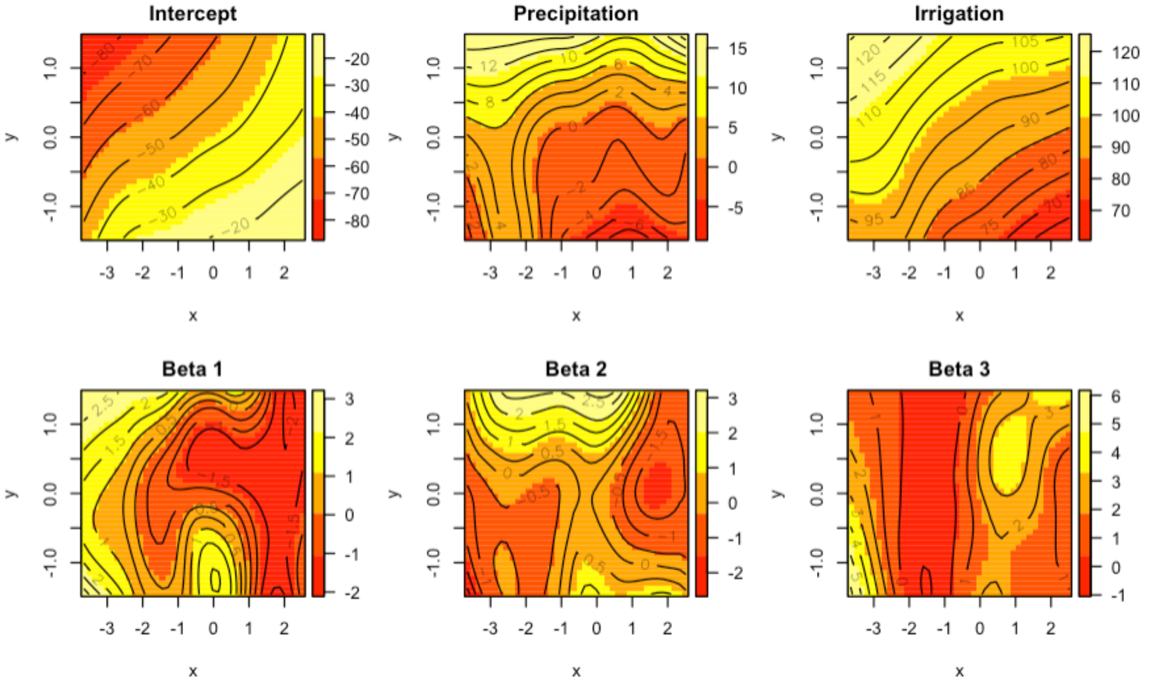
The models used for comparison are the PLFAM and FLM introduced in Section 2.4.3. The PLFAM and FLM have a different approach to FPCA since they do not consider the spatial component. For these models, the functional data is centered by a global mean before FPCA. It is worth mentioning that both models use a variance percentage of 99.9% which selects essentially the maximum number of FPC scores. Both models use a penalty function to prevent overfitting. The PLFAM uses the COSSO penalty first introduced in Lin and Zhang (2006) and the FLM implements a ridge regression penalty. The tuning parameters for COSSO are selected using 5-fold cross-validation and for PLFAM, generalized cross-validation is used. Both models use year indicator variables to account for the consistent increasing trends in yield and an interaction term between proportion of irrigated land and precipitation. There is inherent heteroscedasticity in the errors when not taking the location into account. To account for this in the PLFAM and FLM, they use the size of harvested land as weights in the model fitting. The data is split into five-year windows (i.e. 1999-2003, 2000-2004,..., 2007-2011). For each five-year window, the data is split into five training and testing sets. Since the sizes of harvested land are included in the model, the MSPE is weighted by them as well. The reported MSPE is an average over the 5 validation sets and over all five-year periods. The average MSPE for each model can be seen in Table 2.8

Model	**MSPE**
GSVFM	195.52
PLFAM	317.19
FLM	298.18

Table 2.8: Kansas Model Comparison.

The GSVFM has the smallest MSPE across all models. Its performance is significantly better than the PLFAM and FLM. Part of the reason the GSVFM has significantly better prediction performance, is because it considers location-specific effects while the competing models do not. These results agree with the importance of including the spatial component when the data exhibits variation across space.

Similarly to the simulations in Section 2.4, we can produce heatmaps of the GSVFM model coefficients to understand their behavior across space. In Figure 2.21, the coefficient heat maps for non-functional predictors and the first three functional principal components are displayed over a grid a values across the spatial domain of Kansas.


Intercept

Precipitation

Irrigation

Beta 1

Beta 2

Beta 3

Figure 2.21: Kansas Heatmaps.

The intercept heatmap displays a linear function across space which isn't surprising since the intercept is supposed to capture the global effect at each location. In the red region, we can see that the intercept has the strongest effect which could be because of the other variables being less influential. There is a lot more variation in the precipitation plot with some kind of decreasing exponential function. The irrigation plot also exhibits a linear trend similar to the intercept with its strongest effect being in the upper lefthand corner. In the heatmaps for the coefficients, the behavior is a little bit more scattered than the non-functional variables with no clear pattern. This is to be expected since the FPC scores represent a summary of the functional data with respect to a given eigenfunction. Thus, the interpretation is more difficult when using FPC scores since there isn't a one-to-one correspondence between them and the functional data. We can get some more insight into the FPC scores by looking at the few of the eigenfunctions displayed in Figure 2.22.

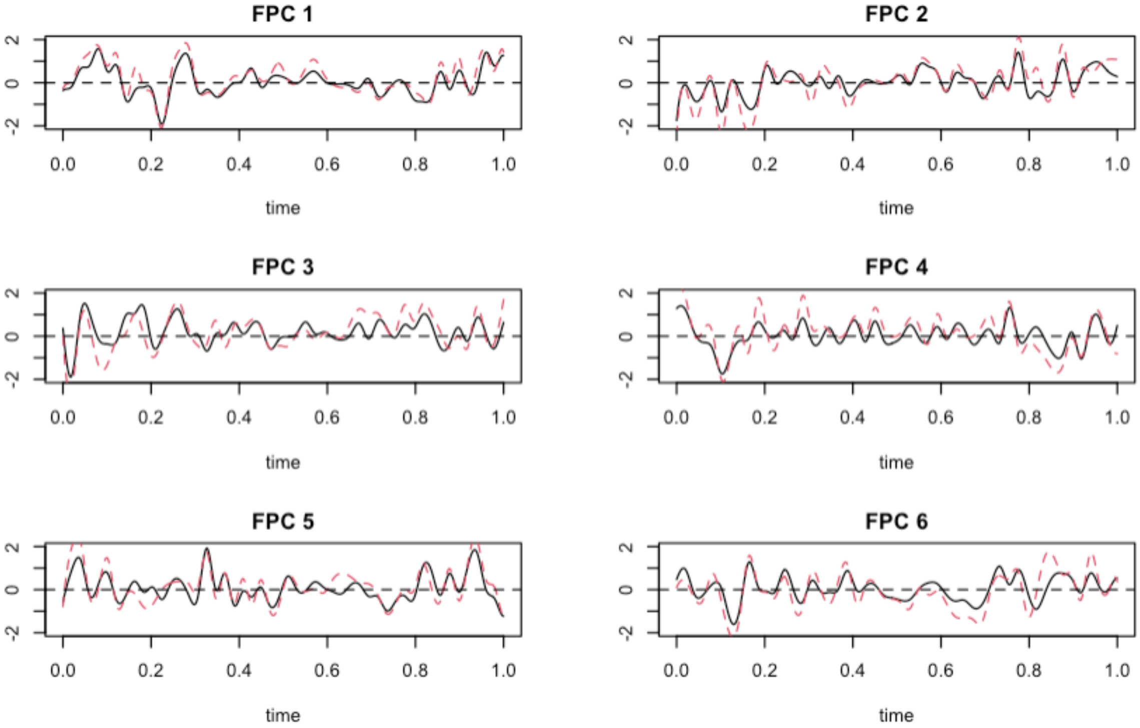


Figure 2.22: Kansas Eigenfunctions.

In all the above plots, the black and dashed red curves represent the maximum and minimum, respectively. The eigenfunctions represent the strongest sources of variation in the functional data. For example, the first eigenfunction shows a lot of variation in the beginning of the year which could be explaining the variation from the winter season. The plot for the second eigenfunction, seems to show a bit more variation towards the end of year. This could be explaining the variation from the fall season. The third eigenfunction exhibits significant variation in the beginning of the year which might explain other sources of variation in the winter season. The minimum and maximum temperature eigenfunctions largely coincide with each other since there is strong correlation between the two functional variables. Overall, the eigenfunctions capture the main sources of variation in the functional data.

2.5.3 Midwest Data

The Midwest data is used in the following analysis which contains data for all variables from Kansas, Missouri, Illinois, Indiana and Iowa over the years 1999-2019 with a total of 8405 observations. The spatial domain of the Midwest region can be seen in Figure 2.24 below.

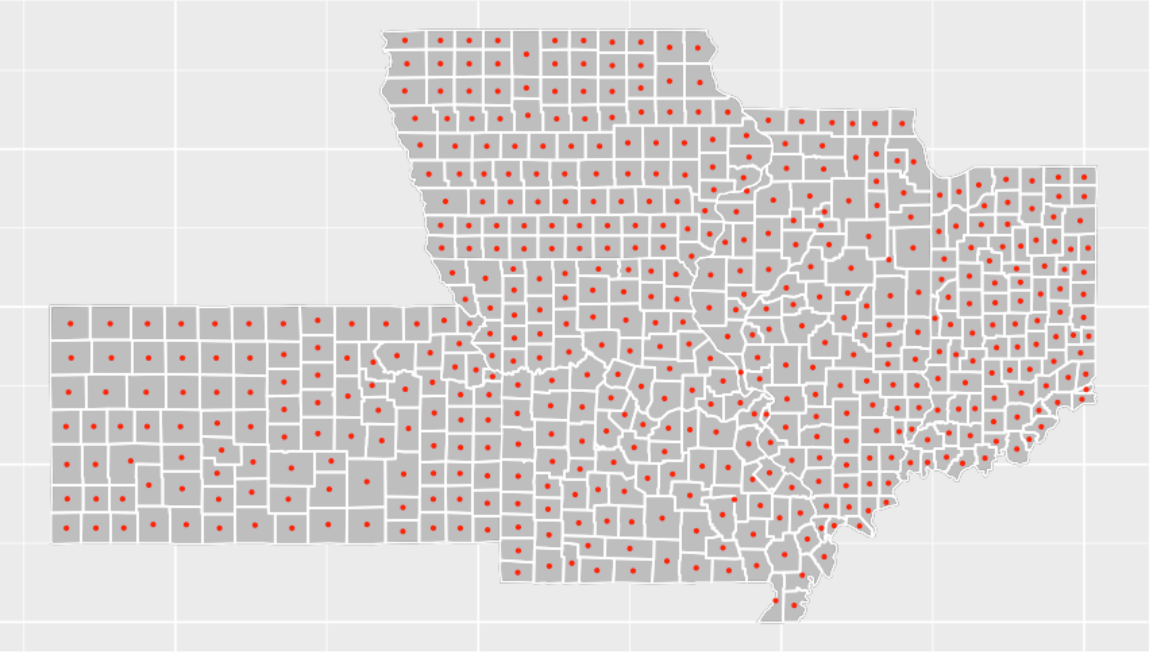


Figure 2.23: Midwest Spatial Domain with County Centroids.

There are 513 counties in the Midwest spatial domain. The triangulation used for the Midwest can be seen in Figure 2.24 below.

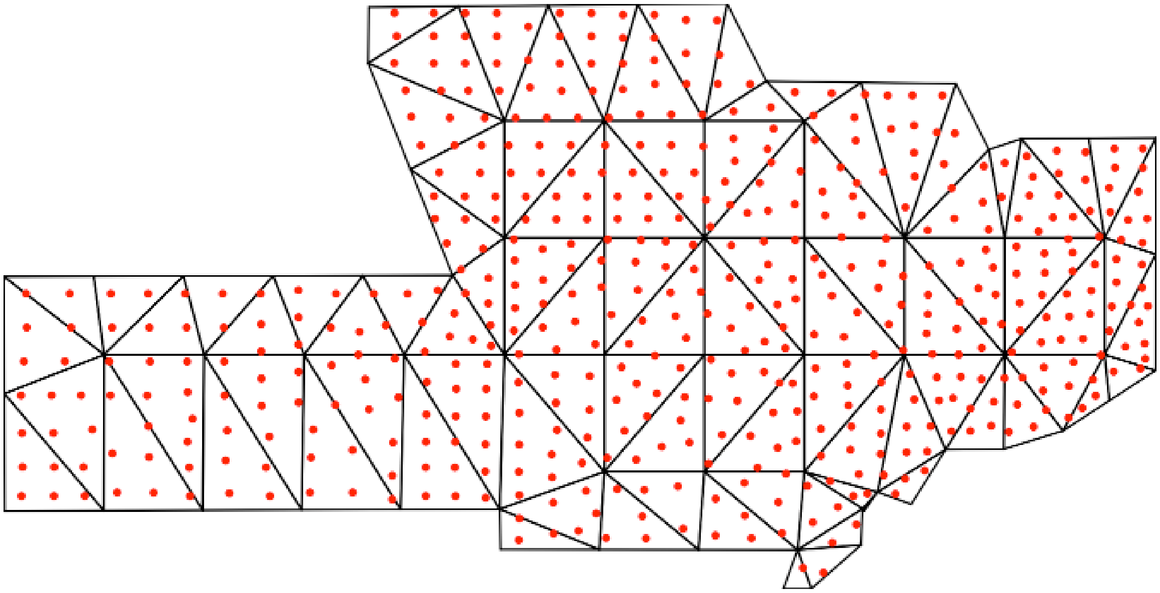


Figure 2.24: Midwest Triangulation with County Centroids.

The Midwest triangulation shown above has 94 triangles and is used throughout the remaining analysis and a basis functions degree of $d = 3$. The smoothness v is set to 1 for the remainder of this section. To fit the GSVCM introduced in Section 2.2, cross-validation needs to be performed for the penalty parameter τ in equation 2.8. The GCV criterion in 2.14 is used to choose the optimal penalty parameter τ at each fold throughout the remainder of this section.

Since the Midwest has significantly more observations than the Kansas dataset, the optimal variance percentage might not be the same as it was in Section 2.5.2. Therefore, it is necessary to perform cross-validation again to ensure the best results. The same approach to cross-validation mentioned in Section 2.5.2, is used for the Midwest data other than the number of repetitions. Running each fold of the Midwest data takes significantly longer than the Kansas data. For this reason, only 3 iterations of 5-fold CV is used. Both the train and test errors are reported, the train error is used to understand the models ability to fit the data, but the number of FPC scores is chosen based on the test error. The results for the cross-validation of the Midwest data is displayed in Figure 2.25.

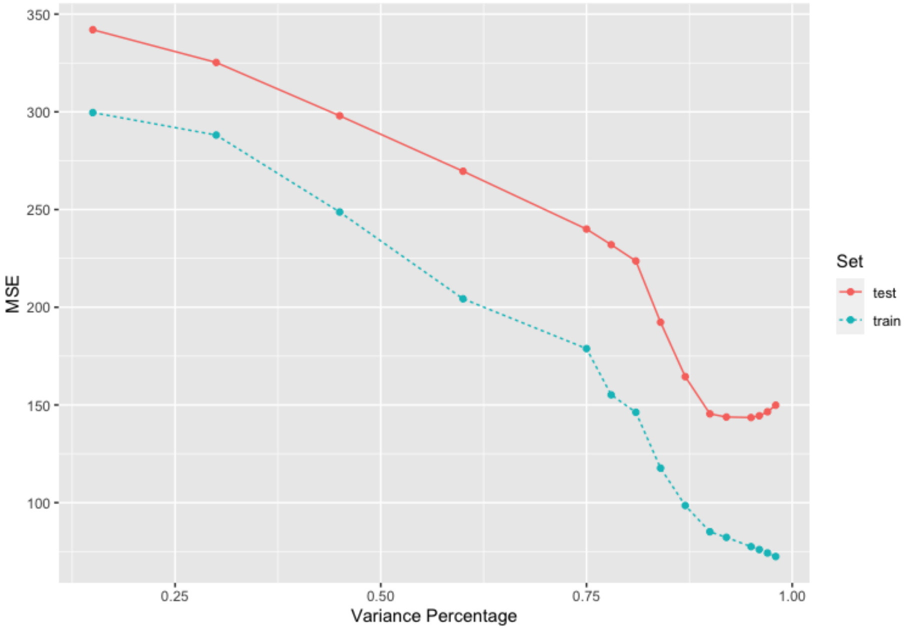


Figure 2.25: MSE vs. Variance Percentage CV (Midwest).

In contrast to Kansas percentage variation CV in Figure 2.20, there is a noticeable decrease in MSE between 15% to 60% for both curves. A possible explanation for this, is that FPCA performs better with more data so the estimated FPC's and FPC scores are improved. There is a consistent decreasing trend throughout the entire range of variance percentage but the sharpest decrease seems to occur between 81% to 84%. The MSE continues to decrease beyond 84% but there is a point where the change is minimal or increasing. Looking at the test curve in red, there is a similar behavior to Figure 2.20 where the MSE starts to increase again after a certain variance percentage. In this case, the threshold is 95% with a train MSE of 77.49 and a test MSE of 143.56. Based strictly on the test MSE, the optimal variance percentage is 95% but this results in a large number of FPC scores. The MSE for

90% is 145.44 which is very close to the MSE for 95%. The number of FPC scores used for 95% variation is 34 and for 90% variation, 21. Since 90% variation uses significantly less scores and results in a comparable test MSE, it is preferable to use 90% variation to avoid the possibility of overfitting. Therefore, 90% variation using 21 FPC scores is used for the remainder of this section.

The data preparation for the GSVFM, PLFAM and FLM follows the same procedure discussed in Section 2.5.2. The reported MSPE for the GSVFM is based on 9 repetitions of 5-folds as was done for the Kansas data. The same calculations for the PLFAM and FLM MSPE are used. The Midwest MSPE results for each model can be seen in Table 2.9.

Model	**MSPE**
GSVFM	144.10
PLFAM	312.87
FLM	387.64

Table 2.9: Midwest Model Comparison.

The GSVFM results in the lowest MSPE with a value of 144.10. This is a significant difference seeing as the PLFAM and FLM MSPE's are more than two times greater than the GSVFM MSPE. It was mentioned in Section 2.5.2 that the main reason for the difference in performance, is that the PLFAM and FLM do not take location-specific effects into consideration. Increasing the size of the spatial domain portrays that location-specific effects become more important as can be seen in the GSVFM performance to competing models. Comparing the Kansas GSVFM MSPE of 195.32 with the Midwest's MSPE of 144.10, further emphasizes the effect a larger spatial domain has on the prediction results.

Akin to the heat maps created for the Kansas data, the Midwest heatmaps are created for the non-functional predictors and the first three FPC scores. The value of the coefficients is displayed for each county rather than a grid of values like in the Kansas data. Gray counties represent missing data. The midwest heatmaps are displayed in Figures 2.26-2.31.

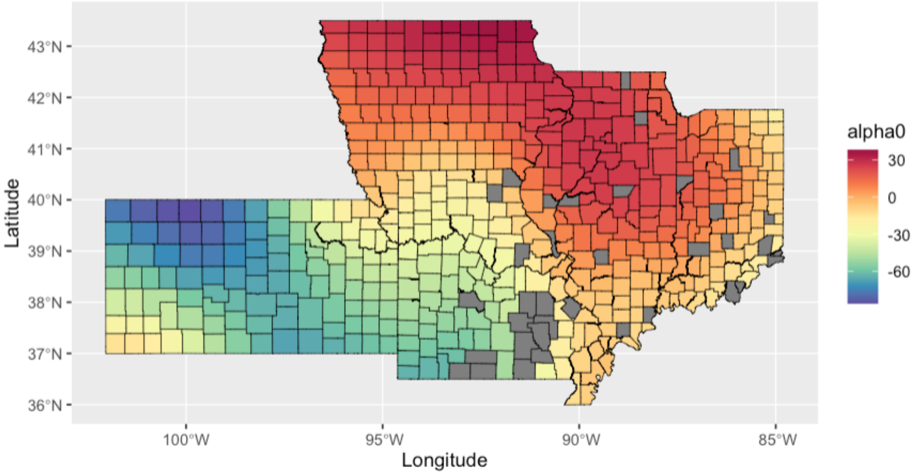


Figure 2.26: α_0 Heatmap (Midwest).

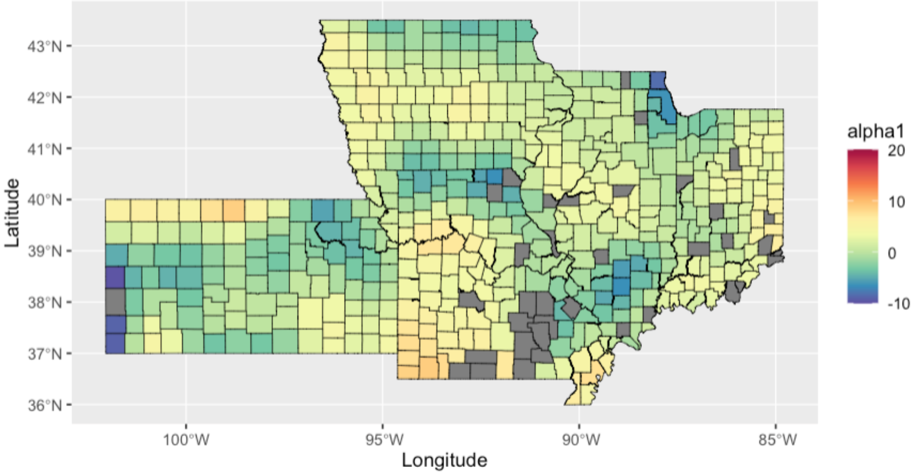


Figure 2.27: α_1 Heatmap (Midwest).

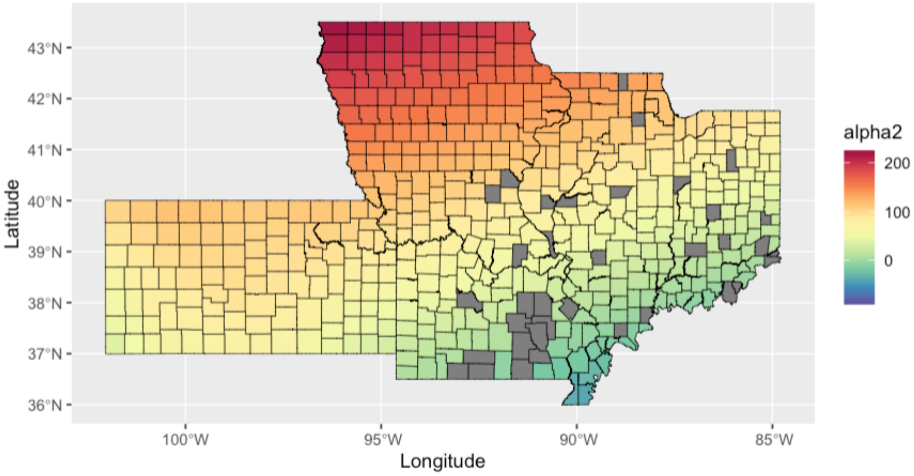


Figure 2.28: α_2 Heatmap (Midwest).

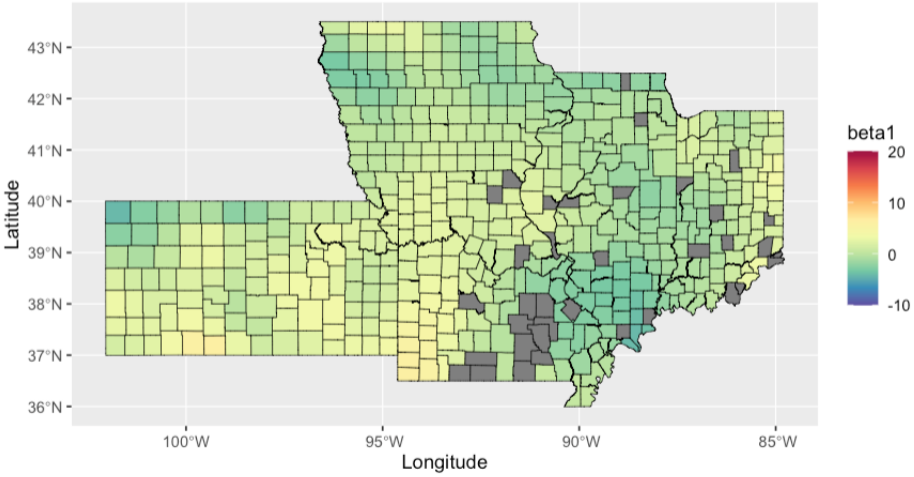


Figure 2.29: β_1 Heatmap (Midwest).

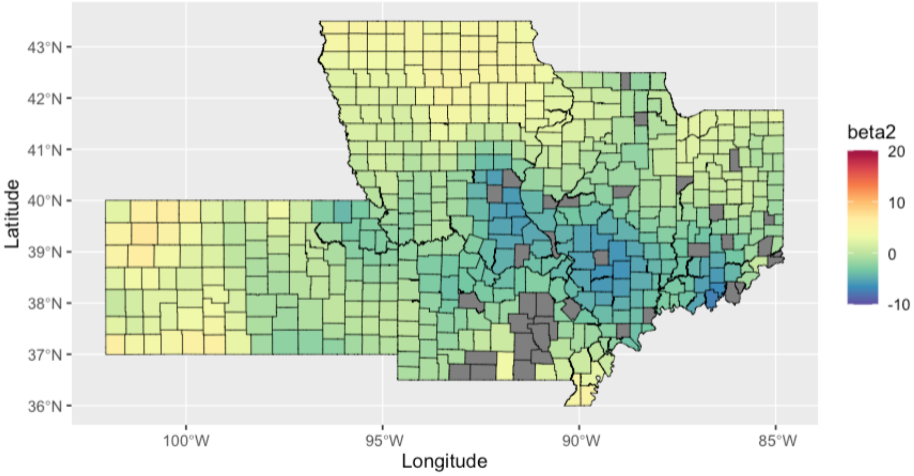


Figure 2.30: β_2 Heatmap (Midwest).

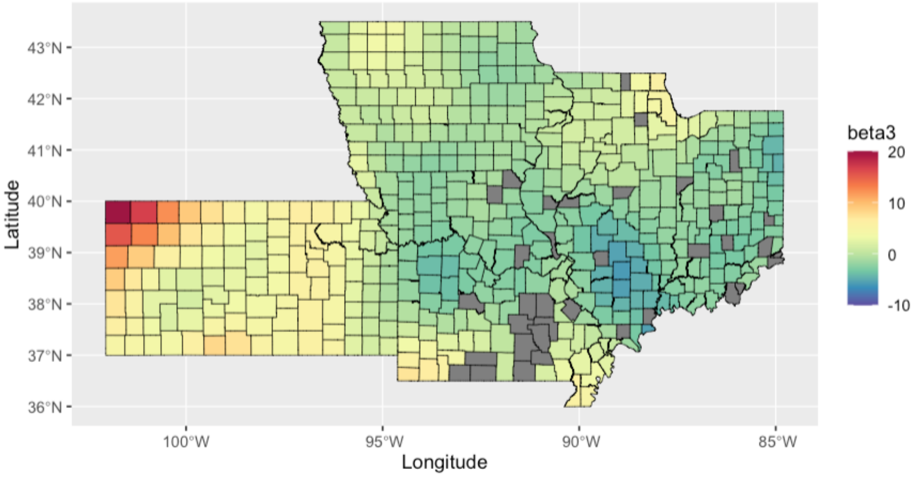


Figure 2.31: β_3 Heatmap (Midwest).

The yield depends on several factors that are included in the model, such as precipitation, irrigation and temperature as well as some factors that are not included such as land quality and soil nutrition. In Figure 2.26, the intercept seems to have a significant influence in multiple counties with only some counties in the center that have an intercept value of zero. This behavior is analogous to the Kansas intercept heatmap in Figure 2.21. As can be seen from the heatmaps, the contribution proportion of the various factors (which has more effect and which has less) does not stay the same in each location. In some locations there are coefficients that are larger than others and in others vice versa. That can be seen through the different range of variations for each coefficient. For example, the top center region of the intercept is very strong in the positive direction which can also be seen in the irrigation heatmap, while the rest of the variables effect is weaker. In contrast, in the center region of the intercept, its values are strong in the negative direction but the irrigation has a larger effect in the positive direction. To accurately study the effect of each factor requires evaluating the magnitude range of the predictor variable with the magnitude of its corresponding coefficient.

As a result of the FPC scores being difficult to intrepret as mentioned in Section 2.5.2, more insight can be gained by plotting the midwest eigenfunctions displayed in Figure 2.32.

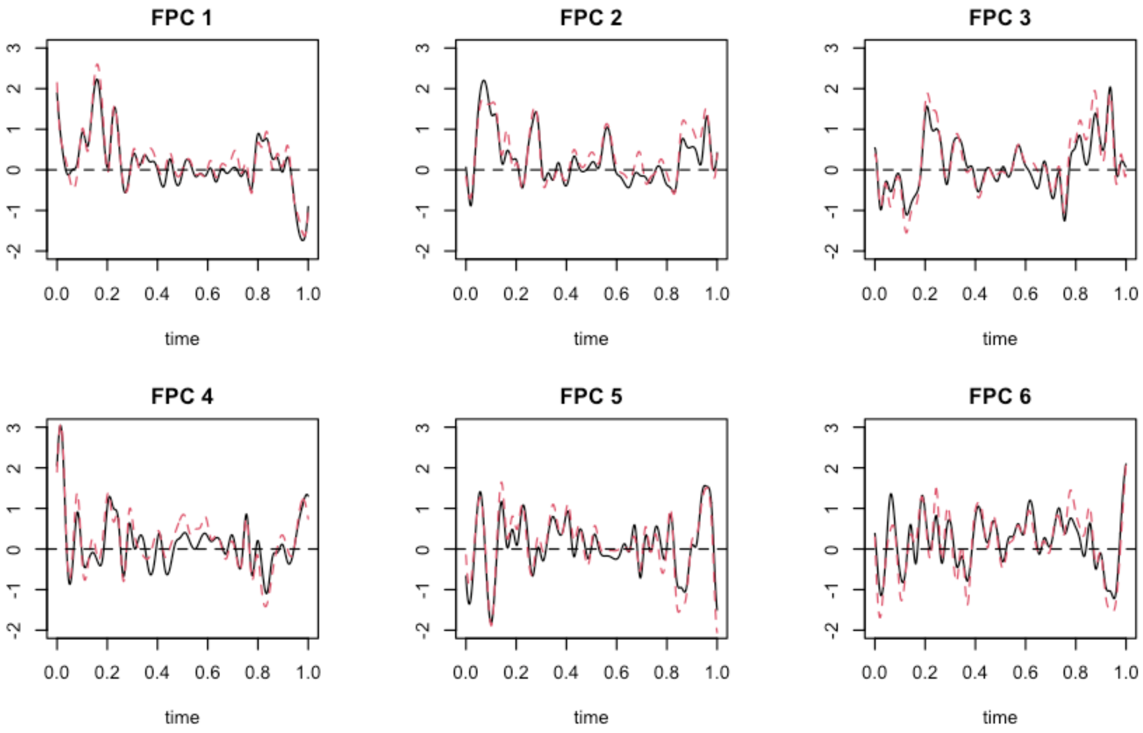


Figure 2.32: Midwest Eigenfunctions.

The first eigenfunction exhibits a lot of variation in the beginning of the year which could be explaining the variation coming from the winter season. In the second eigenfunction plot, there is still exceptional variation in the beginning of the year which may explain other sources of variation coming from winter. The third eigenfunction plot has some variation in the second quarter of the year and towards the end of the year. This might be explaining the variation coming from the spring and fall seasons. What is common across all eigenfunctions, is the minimum and maximum temperature eigenfunctions coincide. This is not surprising, since there is strong correlation between the two functional variables. In summary, the plots suggest that the eigenfunctions capture the most important sources of variation in the functional variables.

2.5.4 Bootstrap Hypothesis Testing Procedure

Throughout this chapter, the spatially varying functional data has been shown to exhibit location-specific effects. These results are based on a finite number of random samples but we need to be wary of making this claim about the population. To determine whether there is evidence for non-stationarity, the coefficient functions are investigated. We can consider the following hypothesis test

$$H_{0j} : \theta_j(\boldsymbol{s}) = \theta_j \text{ vs. } H_{1j} : \theta_j(\boldsymbol{s}) \neq \theta_j. \tag{2.17}$$

If the coefficient functions are truly non-stationary, the data should provide evidence to reject the null hypothesis. In the case that the null hypothesis isn't rejected, then the GSVCM is equivalent to a multiple linear regression model. To perform the hypothesis test in 2.17, we consider a Generalized Quasi-Likelihood Ratio (GQLR) test, where the test statistic is defined to be difference between the quasi-likelihoods under the full and reduced models, respectively. The test statistic is defined as follows

$$\lambda_n(H_0) = \sum_{i=1}^{n} [l_Q\{g^{-1}(\boldsymbol{\mathcal{X}}_i^T \hat{\boldsymbol{\theta}}_F), Y_i\} - l_Q\{g^{-1}(\boldsymbol{\mathcal{X}}_i^T \hat{\boldsymbol{\theta}}_R), Y_i\}], \tag{2.18}$$

where $\hat{\boldsymbol{\theta}}_F$ and $\hat{\boldsymbol{\theta}}_R$ are the estimators under the full and reduced models, respectively. In the full model, all parameters are included in the model. Under the reduced model, all parameters are included and the coefficient being tested, $\theta_j(\boldsymbol{s})$, is treated as a constant coefficient. If there is evidence for non-stationarity, the quasi-likelihood function under the full model should be larger than the quasi-likelihood function under the reduced model. Since the distribution of the test statistic $\lambda_n(H_0)$ is unknown, we use a Bootstrap procedure that is motivated by the GQLR test in Tang et al. (2016). The Bootstrap hypothesis test procedure can be described as in Algorithm 2.

Algorithm 2 GQLR Test

Step 1: Calculate the GQLR test statistic $\lambda_n(H_0)$ based on the data $\{Y_i, \boldsymbol{\mathcal{X}}_i, \boldsymbol{s}_i\}_{i=1}^n$.

Step 2: Generate the response variable $Y_i^* = g(\mu_i^*) + \epsilon_i^*$, $i = 1, \ldots, n$, where $\mu_i^* = \boldsymbol{\mathcal{X}}_i^T \hat{\boldsymbol{\theta}}_R(\boldsymbol{s}_i)$, $\epsilon_i^* = \phi_i \widehat{\epsilon}_i$, $\widehat{\epsilon}_i$'s are residuals from the full model, and ϕ_i's follow a Mammen's two-point distribution with the following probabilities

$$P\left\{\phi_i = \frac{-(\sqrt{5}-1)}{2}\right\} = \frac{(\sqrt{5}+1)}{2\sqrt{5}},$$

$$P\left\{\phi_i = \frac{(\sqrt{5}+1)}{2}\right\} = \frac{(\sqrt{5}-1)}{2\sqrt{5}}.$$

Step 3: For iterations $b = 1, \ldots, B$, calculate the GQLR test statistic $\lambda_n(H_0)$ in equation 2.18 based on the bootstrap sample $\{Y_i^*, \boldsymbol{\mathcal{X}}_i, \boldsymbol{s}_i\}_{i=1}^n$ generated in step 2.

Step 4: Repeat steps 2 and 3 B times and obtain the estimated p-value which is calculated by $\widehat{p} = \sum_{b=1}^B I\{\lambda_n^{(b)}(H_0) \geq \lambda_n(H_0)\}$, where $I(.)$ is the indicator function.

The Bootstrap hypothesis test procedure described in Algorithm 2, is applied to the Kansas and Midwest datasets. For each variable, $B = 500$ replications are used to stabilize the results. All hyptohesis tests are conducted using an significance-level of $\alpha = 0.05$. In the Kansas data, nearly all coefficients are statistically significant with p-values smaller than 0.002 other than β_1 (prcp) and β_{16} (14th FPC score) with p-values of 0.02 and 0.022, respectively. The coefficient of the 15th FPC score, β_{17} is not statistically significant with a p-value of 0.084. In the Midwest data, all coefficients are statistically significant with p-values smaller than 0.002. These results provide sufficient evidence that the coefficient functions are non-stationary.

2.6 Conclusion

In this chapter, the GSVFM was introduced along with its estimation method, which uses a novel two-step procedure. The GSVFM extends the generalized functional linear model by incorporating spatially varying functional data. The two-step procedure, described in Section 2.2, effectively captures variations across both time and space. This approach leverages the mFPCA method to explain the variation in functional data and the GSVCM to account for spatial variation. The results from both simulations and real-world application data presented in this chapter highlight the significance of considering location-specific effects in the presence of spatial variation. The GSVFM demonstrates strong predictive performance for spatially varying functional data and surpasses existing functional models.

Chapter 3

Prediction in Spatially Varying Functional Quantile Model

3.1 Introduction

Most regression models in Statistics are designed to estimate the conditional mean, typically expressed as $\mu = E[Y|X]$, since it serves as a useful measure of central tendency. There are many types of regression models tailored for this purpose, and the theory surrounding them is well established. In numerous cases, these models have proven to work effectively, offering interpretable results. While conditional mean regression models are still widely used today, they are not always the best choice. It's well known that the mean can be sensitive to outliers, making it a less reliable measure of central tendency in some datasets. In such cases, conditional mean regression models may provide adequate but not optimal results.

When applying statistics to various fields, the goal is to find the method that best fits the data structure and the analysis objectives. For data prone to outliers, where the conditional mean might not be sufficient, the GSVFM discussed in Chapter 2 may not be the ideal approach. The median, often regarded as a more robust measure of central tendency in the presence of outliers, is a better alternative in these situations. Additionally, there are cases where estimating quantiles is more relevant than focusing solely on the conditional mean, especially in the presence of heteroscedasticity. In such scenarios, multiple quantiles provide a more comprehensive view of the distribution than the mean alone. To address these problems, the method of quantile regression was first proposed in Koenker and Bassett (1978). In this paper, the goal was to develop a class of linear models to estimate conditional quantiles of the form $Q_Y(\tau|X)$ for the τth quantile. This development arised from a simple minimization problem providing the ordinary sample quantiles in the location model. They showed that this minimization problem can be generalized to the linear model. This generated a new class of Statistics called regression quantiles. Although quantile regression offers considerable flexibility, it also presents certain challenges, particularly from a computational perspective. The method of ordinary least squares is particularly attractive not only for its utility, but also its ease of computation. In the case of quantile regression, the minimization of absolute residuals is used through the check loss function which complicates the model estimation. For the τth quantile, the check function is defined as $\rho_\tau(a) = a \times (\tau - I(a < 0))$. Since the check function is an absolute value, its derivative is undefined. There are different ways to solve this kind of objective function, mainly methods involving linear programming and MM (Majorize-Minorize) algorithms. In Koenker and Bassett

(1978), the linear programming approach is used through the use of the simplex method. Since the introduction of quantile regression, there have been many developments. In Yu and Jones (1998), a local linear estimator was introduced for quantile regression through two distinct methods. The first method is more direct which involves the minimization of a local linear kernel weighted version expected loss function $\mathbb{E}\{\rho_\tau(Y - a|X = x)\}$. The second method involves inverting a local linear estimator of the conditional distribution $F_{Y|X}(y|X = x)$. Another nonparameteric estimation method for quantile regression was introduced in Koenker et al. (1994) where they explored a class of quantile smoothing splines with a roughness penalty. Koenker and Park (1996) created a new alogrithm for computing quantile regression estimators where the response is nonlinear in the parameters. Their approach to estimation is based on interior point methods. An alternative estimation method to nonlinear models is through an MM algorithm introduced in Hunter and Lange (2000). As in any area of Statistics, it is important to create tools to perform inference in the quantile regression process. Most inference procedures for quantile regression were based on nonparameteric tests but in Koenker and Xiao (2002), they introduce parametric hypothesis tests that can be applied to a variety of inference problems. Since it is of interest to test hypotheses including nuisance parameters, this jeopardizes the distribution free character of nonparametric tests. This can be characterized as "the Durbin problem" as mentioned in Watson (1973). Their approach to the Durbin problem involes the use of a martingale transformation of the parametric empirical process suggested by Khmaladze (1982).

There have been numerous developments in both theory and methodology in quantile regression since it was proposed by Koenker and Bassett (1978). The research for conditional quantiles using functional data is a relatively newer area of research but with significant developments since its introduction. Conditional quantile estimation where the covariates are functions, was first introduced in Cardot et al. (2007) where they proposed a smoothing spline estimator and studied its asymptotic behavior. Similar to the estimation approach used in Yu and Jones (1998), in Ferraty et al. (2005), they also used the inversion of conditional distribution function which is estimated by a kernel estimator. In contrast to this method, Chen and Müller (2011) developed a new inversion method where the conditional distribution function is represented as a functional generalized linear model. The method of FPCA is frequently used in functional models and was first used in estimating conditional quantiles in Kato (2012) through the plug-in method. They establish some convergence rates for the estimator in a minimax sense. Building on the research done in Kato (2012), Yao et al. (2017) implemented FPCA in their proposed regularized partially functional linear quantile regression model with high-dimensional scalar covariates. A shrinkage penalty is used to select the most important variables from the high-dimensional covariates.

Although there has been notable progress in functional quantile regression, the functional data is only allowed to vary with respect to time. As was described in Chapter 2, there is the possibility for location-specific effects in the presence spatially varying functional data. Not including the spatial component could affect prediction performance. Therefore, in this chapter we introduce the Spatially Varying Functional Quantile Regression Model (SVFQM) that extends the methodology of the GSVFM introduced in Section 2.2. The spatial conditional quantile is estimated through scalar predictor variables and spatially varying functional data. Other than the quantity being estimated, the main distinction is that the SVFQM doesn't allow for any link function. The approach we utilize uses the novel two-step procedure introduced in Section 2.2 with some inherent differences. The advantage of the SVFQM over existing functional quantile regression models, is shown through simulations and the analysis of the agriculture data.

The organization of the rest of the chapter is as follows. In Sections 3.2-2.3, we describe the model, methodology and estimation used for the SVFQM. The SVFQM's performance is investigated by conducting simulations on spatially varying data in Section 2.4 and applying it to the agriculture data application in Section 3.3. Concluding remarks are discussed in Section 3.3.

3.2 Methodology

The model in consideration is intended to perform prediction for the spatially varying conditional quantile. Let $Y(\boldsymbol{s})$ denote the scalar response variable located at $\boldsymbol{s} \in \Omega$ for a spatial domain $\Omega \subset \mathbb{R}^2$. The scalar response $Y(\boldsymbol{s})$ is assumed to be dependent on q functional variables $\boldsymbol{X}(\boldsymbol{s}; t) = \{X_1(\boldsymbol{s}; t), X_2(\boldsymbol{s}; t), \ldots, X_q(\boldsymbol{s}; t)\}^T$ defined for $t \in \mathcal{T}$ and m spatially varying predictors $\boldsymbol{Z}(\boldsymbol{s}) = \{Z_1(\boldsymbol{s}), Z_2(\boldsymbol{s}), \ldots, Z_m(\boldsymbol{s})\}^T$. Without loss of generality, the time domain is rescaled to a unit interval where $\mathcal{T} = [0, 1]$. Then for a quantile $\tau \in (0, 1)$, the SVFQM is defined as follows

$$Y(\boldsymbol{s}) = \eta_{0;\tau}(\boldsymbol{s}) + \boldsymbol{Z}^T(\boldsymbol{s})\boldsymbol{\alpha}_\tau(\boldsymbol{s}) + \int_{\mathcal{T}} \boldsymbol{X}^T(\boldsymbol{s}; t)\boldsymbol{\eta}_\tau(\boldsymbol{s}; t)dt + \epsilon(\boldsymbol{s}), \tag{3.1}$$

where the $P(\epsilon(\boldsymbol{s}) < 0|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}) = \tau$ and $\eta_{0;\tau}(\boldsymbol{s})$ is the location-specific intercept, $\boldsymbol{\alpha}_\tau(\boldsymbol{s}) = \{\alpha_{1;\tau}(\boldsymbol{s}), \alpha_{2;\tau}(\boldsymbol{s}), \ldots, \alpha_{m;\tau}(\boldsymbol{s})\}^T$ is a vector of the spatially varying coefficients, and $\boldsymbol{\eta}_\tau(\boldsymbol{s};t) = \{\eta_{1;\tau}(\boldsymbol{s};t), \eta_{2;\tau}(\boldsymbol{s};t), \dots, \eta_{q;\tau}(\boldsymbol{s};t)\}^T$ is a vector of spatially varying functions over $t \in \mathcal{T}$ modeling the effect of functional trajectories on the scalar response at location $\boldsymbol{s}$. The conditional distribution function of $Y|(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})$ is denoted as

$$F_Y(.|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}) = P(Y \leq .|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}).$$

The τth conditional quantile of $Y|(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})$ can then be defined as follows

$$Q_Y(\tau|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}) = \inf\{y : F_Y(y|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}) \geq \tau\}.$$

For $0 < \tau < 1$, we assume that $Q_Y(\tau|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})$ has a functional linear structure. Therefore, the SVFQM in 3.1 can be rewritten as

$$Q_Y(\tau|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}) = F_\epsilon^{-1}(\tau) + \eta_{0;\tau}(\boldsymbol{s}) + \boldsymbol{Z}^T(\boldsymbol{s})\boldsymbol{\alpha}_\tau(\boldsymbol{s}) + \int_{\mathcal{T}} \boldsymbol{X}^T(\boldsymbol{s};t)\boldsymbol{\eta}_\tau(\boldsymbol{s};t)dt, \qquad (3.2)$$

where $F_\epsilon^{-1}(\tau)$ is the τth quantile of ϵ. The model in 3.2 has an infinite-dimensionality problem exactly like the GSVFM model introduced in Section 2.2. To reduce the SVFQM's dimensionality, the method of mFPCA used in Section 2.2 is performed using the functional data $\boldsymbol{X}(\boldsymbol{s};t)$ and we obtain the Karhunen-Loeve expansions of $\boldsymbol{X}(\boldsymbol{s};t)$ and $\boldsymbol{\eta}(\boldsymbol{s};t)$, respectively. Using these expansions, model 3.2 can be rewritten as

$$Q_Y(\tau|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}) = F_\epsilon^{-1}(\tau) + \alpha_{0;\tau}(\boldsymbol{s}) + \boldsymbol{Z}^T(\boldsymbol{s})\boldsymbol{\alpha}_\tau(\boldsymbol{s}) + \sum_{r=1}^{\infty} \xi_r(\boldsymbol{s})\beta_{r;\tau}(\boldsymbol{s}), \qquad (3.3)$$

which is a quantile spatially varying coefficient model. The FPC scores in model 3.3 random variables and are therefore unobservable. A finite number p of estimated FPC scores $\widehat{\xi}_1(\boldsymbol{s}), \dots, \widehat{\xi}_p(\boldsymbol{s})$ are used instead and the model can be approximated by

$$Q_Y(\tau|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}) \approx F_\epsilon^{-1}(\tau) + \alpha_{0;\tau}(\boldsymbol{s}) + \boldsymbol{Z}^T(\boldsymbol{s})\boldsymbol{\alpha}_\tau(\boldsymbol{s}) + \sum_{r=1}^{p} \widehat{\xi}_r(\boldsymbol{s})\beta_{r;\tau}(\boldsymbol{s}). \qquad (3.4)$$

Letting $\boldsymbol{\mathcal{X}}(\boldsymbol{s}) = \{\mathcal{X}_0(\boldsymbol{s}), \mathcal{X}_1(\boldsymbol{s}), \ldots, \mathcal{X}_{m+p}(\boldsymbol{s})\}^T = \{Z_0(\boldsymbol{s}) \equiv 1, Z_1(\boldsymbol{s}), \ldots, Z_m(\boldsymbol{s}), \widehat{\xi}_1(\boldsymbol{s}), \ldots, \widehat{\xi}_p(\boldsymbol{s})\}^T$ be the vector of predictor variables and $\boldsymbol{\theta}(\boldsymbol{s}) = \{\theta_{0;\tau}, \theta_{1;\tau}, \ldots, \theta_{m+p;\tau}\}^T = \{\alpha_{0;\tau}(\boldsymbol{s}), \alpha_1(\boldsymbol{s}), \ldots, \alpha_{m;\tau}(\boldsymbol{s}), \beta_{1;\tau}(\boldsymbol{s}), \ldots, \beta_{p;\tau}(\boldsymbol{s})\}^T$ the vector of coefficients, for $j = 0, \ldots, (m+p)$, model 3.4 can be rewritten as follows

$$Q_Y(\tau|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{s}) \approx F_\epsilon^{-1}(\tau) + \sum_{j=0}^{m+p} \mathcal{X}_j(\boldsymbol{s})\theta_{j;\tau}(\boldsymbol{s}) = F_\epsilon^{-1}(\tau) + \boldsymbol{\mathcal{X}}^T(\boldsymbol{s})\boldsymbol{\theta}_\tau(\boldsymbol{s}). \tag{3.5}$$

3.3 Estimation

Suppose we observe K_u spatial replicates $\{Y_k(\boldsymbol{s}_u), \boldsymbol{X}_k(\boldsymbol{s}_u; t), \boldsymbol{Z}_k(\boldsymbol{s}_u)\}_{k=1}^{K_u}$ at location $\boldsymbol{s}_u \in \Omega \subset \mathbb{R}^2, u = 1, \ldots, U$. Define the total number of observations by $n = \sum_{u=1}^{U} K_u$ and denote the ith observation as $\{Y_i(\boldsymbol{s}_i), \boldsymbol{X}_i(\boldsymbol{s}_i; t), \boldsymbol{Z}_i(\boldsymbol{s}_i)\}$, for $i = 1, \ldots, n$, where $\boldsymbol{s}_i$ represents the location for the ith observation. For the ith observation, there are p principal component scores $\{\widehat{\xi}_1(\boldsymbol{s}), \ldots, \widehat{\xi}_p(\boldsymbol{s})\}$. The ith observation can be denoted by $\{Y_i(\boldsymbol{s}_i), \boldsymbol{\mathcal{X}}_i(\boldsymbol{s}_i)\}$, for $i = 1, \ldots, n$.

The estimation of the model coefficients $\boldsymbol{\theta}$ in model 3.5, we use the method of Bivariate Penalized Splines over Triangulations introduced in Sections 2.2.2 and 2.2.3. The objective function can be defined as follows

$$\frac{1}{n}\sum_{i=1}^{n} \rho_\tau\Big(Y_i - \sum_{j=0}^{m+p} \mathcal{X}_{ij}(\boldsymbol{s}_i)\theta_{j;\tau}(\boldsymbol{s}_i)\Big) + \sum_{j=0}^{m+p} \nu_j \varepsilon(\theta_{j;\tau}), \tag{3.6}$$

where $\rho_\tau(.)$ is the check-loss function defined as $\rho_\tau(a) = a \times (\tau - I(a < 0))$ and $\varepsilon(.)$ is the energy functional penalty. Using the Bernstein basis expansion defined in Section 2.2.2, each coefficient $\theta_{j;\tau}(\boldsymbol{s})$ can be approximated by $\theta_{j;\tau}(\boldsymbol{s}) = \boldsymbol{B}(\boldsymbol{s})^T\boldsymbol{\gamma}_{j;\tau}$, where $\boldsymbol{B}(\boldsymbol{s}) = \{B_{jw}, w \in \mathcal{W}_j\}^T$ and $\boldsymbol{\gamma}_{j;\tau} = \{\gamma_{jw}, w \in \mathcal{W}_j\}^T$ is the vector of bivariate basis functions evaluated at location $\boldsymbol{s}$ and the corresponding spline coefficient vector, respectively. The coefficient

$\theta_{j;\tau} \in \mathbb{S}_d^v(\Delta)$ is penalized over the entire spatial domain Ω. Since $\theta_{j;\tau}$ can't be calculated for every $\boldsymbol{s} \in \Omega$, it is approximated by $\boldsymbol{B}^T\boldsymbol{\gamma}_{j;\tau}$ where $\boldsymbol{B} = \{\boldsymbol{B}(\boldsymbol{s}_1), \ldots, \boldsymbol{B}(\boldsymbol{s}_n)\}^T$. Then the energy functional penalty $\varepsilon(\theta_{j;\tau})$ can be approximated by $\varepsilon(\boldsymbol{B}\boldsymbol{\gamma}_{j;\tau}) = \boldsymbol{\gamma}_{j;\tau}^T\boldsymbol{K}\boldsymbol{\gamma}_{j;\tau}$ where $\boldsymbol{K}$ is the block diagonal penalty matrix. The objective function in 3.6 can be rewritten as

$$\frac{1}{n}\sum_{i=1}^{n}\rho_\tau\Big(Y_i - \sum_{j=0}^{m+p}\mathcal{X}_{ij}(\boldsymbol{s}_i)\boldsymbol{B}(\boldsymbol{s}_i)^T\boldsymbol{\gamma}_{j;\tau}\Big) + \sum_{j=0}^{m+p}\nu_j\boldsymbol{\gamma}_{j;\tau}^T\boldsymbol{K}\boldsymbol{\gamma}_{j;\tau}. \qquad (3.7)$$

It is necessary to enforce smoothness across the shared edges of the triangulation Δ. This is accomplished by the constraint matrix $\boldsymbol{H}$ that satisfies $\boldsymbol{H}\boldsymbol{\gamma}_j = \boldsymbol{0}\ \forall\ j = 1, \ldots, (m+p)$. The constraints in $\boldsymbol{H}$ depend on the degree of smoothness for the basis functions and the structure of the triangulation Δ. We remove the constraint $\boldsymbol{H}\boldsymbol{\gamma}_{j;\tau} = \boldsymbol{0}$ by the QR decomposition. Through the QR decomposition introduced in Section 2.2.3, $\boldsymbol{\gamma}_{j;\tau}$ can be reparametrized by $\boldsymbol{\gamma}_{j;\tau} = \boldsymbol{Q}_2\boldsymbol{\gamma}_{j;\tau}^*$ which guarantees that $\boldsymbol{H}\boldsymbol{\gamma}_{j;\tau} = \boldsymbol{0}$. Plugging in this expression for $\boldsymbol{\gamma}_{j;\tau}$, we now have the following

$$\frac{1}{n}\sum_{i=1}^{n}\rho_\tau\Big(Y_i - \sum_{j=0}^{m+p}\mathcal{X}_{ij}(\boldsymbol{s}_i)\boldsymbol{B}(\boldsymbol{s}_i)^T\boldsymbol{Q}_2\boldsymbol{\gamma}_{j;\tau}^*\Big) + \sum_{j=0}^{m+p}\nu_j\boldsymbol{\gamma}_{j;\tau}^{*T}\boldsymbol{Q}_2^T\boldsymbol{K}\boldsymbol{Q}_2\boldsymbol{\gamma}_{j;\tau}^*. \qquad (3.8)$$

Letting $\boldsymbol{\gamma}_\tau^* = \{\boldsymbol{\gamma}_{1:\tau}^{*T}, \ldots, \boldsymbol{\gamma}_{m+p;\tau}^{*T}\}^T$, we have the following minimization problem

$$\min_{\boldsymbol{\gamma}_\tau^*}\frac{1}{n}\sum_{i=1}^{n}\rho_\tau\Big(Y_i - \sum_{j=0}^{m+p}\mathcal{X}_{ij}(\boldsymbol{s}_i)\boldsymbol{B}(\boldsymbol{s}_i)^T\boldsymbol{Q}_2\boldsymbol{\gamma}_{j;\tau}^*\Big) + \sum_{j=0}^{m+p}\nu_j\boldsymbol{\gamma}_{j;\tau}^{*T}\boldsymbol{Q}_2^T\boldsymbol{K}\boldsymbol{Q}_2\boldsymbol{\gamma}_{j;\tau}^*. \qquad (3.9)$$

This is a similar minimization problem compared with equation 2.10 in Section 2.3. The main difference is the check loss function is used as opposed to the quasi-likelihood function $l_Q(.)$. Therefore, the PIRLS algorithm described in Section 2.5.4, can't be used to minimize the objective function in equation 3.9. To minimize this objective function, we can use a quantile linear regression model function from an existing package in R. There is no existing quantile linear regression function that can minimize the objective function in 3.9. In order to use an existing quantile linear regression function in R, the objective function needs to be reparametrized so that it is equivalent to a quantile linear regression model with a ridge penalty.

3.4 Reparametrization Method

To convert the objective function in equation 3.9, we can use a reparameterization method. The objective function can be rewritten by letting $\boldsymbol{B}^*(\boldsymbol{s}_i) = \boldsymbol{Q}_2^T\boldsymbol{B}(\boldsymbol{s}_i)$ and $\boldsymbol{\mathcal{X}}_{ij}^*(\boldsymbol{s}_i) = \mathcal{X}_{ij}(\boldsymbol{s}_i)\boldsymbol{B}_{ij}^*(\boldsymbol{s}_i)$. Then we can rewrite the objective function as

$$\min_{\boldsymbol{\gamma}^*} \frac{1}{n}\sum_{i=1}^{n}\rho_\tau\Big(Y_i - \sum_{j=0}^{m+p}\boldsymbol{\mathcal{X}}_{ij}^{*T}(\boldsymbol{s}_i)\boldsymbol{\gamma}_{j;\tau}^*\Big) + \sum_{j=0}^{m+p}\nu_j\boldsymbol{\gamma}_{j;\tau}^{*T}\boldsymbol{Q}_2^T\boldsymbol{K}\boldsymbol{Q}_2\boldsymbol{\gamma}_{j;\tau}^*. \tag{3.10}$$

The key to reparametrizing the objective function is by using a matrix decomposition. Let $\boldsymbol{D} = \boldsymbol{Q}_2^T\boldsymbol{K}\boldsymbol{Q}_2$. We want to find a decomposition for $\boldsymbol{D}$ such that $\boldsymbol{D} = \boldsymbol{R}\boldsymbol{R}^T$ for some matrix $\boldsymbol{R}$. The square root of the matrix $\boldsymbol{D}$, $\boldsymbol{D}^{\frac{1}{2}}$ satisfies this where $\boldsymbol{D} = \boldsymbol{D}^{\frac{1}{2}}(\boldsymbol{D}^{\frac{1}{2}})^T = \boldsymbol{D}^{\frac{1}{2}}\boldsymbol{D}^{\frac{1}{2}}$. Using this decomposition, the objective function can be rewritten as

$$\min_{\boldsymbol{\gamma}_\tau^*} \frac{1}{n}\sum_{i=1}^{n}\rho_\tau\Big(Y_i - \sum_{j=0}^{m+p}\boldsymbol{\mathcal{X}}_{ij}^{*T}(\boldsymbol{s}_i)\boldsymbol{\gamma}_{j;\tau}^*\Big) + \sum_{j=0}^{m+p}\nu_j\boldsymbol{\gamma}_{j;\tau}^{*T}\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{\gamma}_{j;\tau}^*. \tag{3.11}$$

Since the error component of the objective function is already in terms of a linear quantile regression model with respect to $\boldsymbol{\gamma}_{j;\tau}^*$, we want the penalty component to represent a ridge penalty. This can be accomplished by defining new variables in terms of the square root of the penalty, $\boldsymbol{D}^{\frac{1}{2}}$. Let $\widetilde{\boldsymbol{\gamma}}_{j;\tau} = (\boldsymbol{D}^{\frac{1}{2}})^T\boldsymbol{\gamma}_{j;\tau}^* = \boldsymbol{D}^{\frac{1}{2}}\boldsymbol{\gamma}_{j;\tau}^*$ and $\widetilde{\boldsymbol{\mathcal{X}}}_{ij} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{\mathcal{X}}_{ij}^*$. The objective function in equation 3.11 can be rewritten

$$\begin{aligned}
&\min_{\widetilde{\gamma}_\tau} \frac{1}{n}\sum_{i=1}^{n}\rho_\tau\Big(Y_i - \sum_{j=0}^{m+p}\widetilde{\boldsymbol{\mathcal{X}}}_{ij}^{T}(\boldsymbol{s}_i)\widetilde{\boldsymbol{\gamma}}_{j;\tau}\Big) + \sum_{j=0}^{m+p}\nu_j\widetilde{\boldsymbol{\gamma}}_{j;\tau}^{T}\widetilde{\boldsymbol{\gamma}}_{j;\tau} \\
=&\min_{\widetilde{\gamma}_\tau} \frac{1}{n}\sum_{i=1}^{n}\rho_\tau\Big(Y_i - \sum_{j=0}^{m+p}\widetilde{\boldsymbol{\mathcal{X}}}_{ij}^{T}(\boldsymbol{s}_i)\widetilde{\boldsymbol{\gamma}}_{j;\tau}\Big) + \sum_{j=0}^{m+p}\nu_j\|\widetilde{\boldsymbol{\gamma}}_{j;\tau}\|_2^2,
\end{aligned} \tag{3.12}$$

which is equivalent to the objective function of a quantile linear regression model with a ridge penalty. Using the estimates of $\widetilde{\boldsymbol{\gamma}}_{j;\tau}$, $\widehat{\widetilde{\boldsymbol{\gamma}}}_{j;\tau}$, we can solve for the original spline coefficients $\boldsymbol{\gamma}_j^*$ by the following transformation

$$\widetilde{\boldsymbol{\gamma}}_{j;\tau} = \boldsymbol{D}^{\frac{1}{2}}\boldsymbol{\gamma}_{j;\tau}^{*} \Rightarrow \boldsymbol{\gamma}_{j;\tau}^{*} = \boldsymbol{D}^{-\frac{1}{2}}\widetilde{\boldsymbol{\gamma}}_{j;\tau} \Rightarrow \widehat{\boldsymbol{\gamma}}_{j;\tau}^{*} = \boldsymbol{D}^{-\frac{1}{2}}\widehat{\widetilde{\boldsymbol{\gamma}}}_{j;\tau}.$$

The estimated model coefficients $\widehat{\theta}_{j;\tau}(\boldsymbol{s})$, can be found by

$$\begin{aligned}
&\theta_{j;\tau}(\boldsymbol{s}) = \boldsymbol{B}(\boldsymbol{s})^T\boldsymbol{\gamma}_{j;\tau} = \boldsymbol{B}(\boldsymbol{s})^T\boldsymbol{Q}_2\boldsymbol{\gamma}_{j;\tau}^{*} = \boldsymbol{B}(\boldsymbol{s})^{*T}\boldsymbol{\gamma}_{j;\tau}^{*}, \\
\Rightarrow\ &\widehat{\theta}_{j;\tau}(\boldsymbol{s}) = \boldsymbol{B}(\boldsymbol{s})^{*T}\widehat{\boldsymbol{\gamma}}_{j;\tau}^{*},\ j = 0, \ldots, (m+p).
\end{aligned}$$

3.5 Implementation in R

The minimization of the objective function in equation 3.12, can be accomplished through the implementation of the rqPen package in R. The rqPen package fits linear quantile regression with different penalties such as LASSO, ridge, and elastic net. There are multiple functions within this package but the main functions used are rq.pen and rq.pen.cv. The rq.pen function performs model fitting and rq.pen.cv performs cross-validation for the penalty smoothing parameter ν. As was mentioned in Section 3.1, linear programming methods are generally used to fit quantile regression models. In the case when the number of observations n and variables $(m + p)$ are large, the linear programming algorithms become computationally burdensome. In Yi and Huang (2015), they proposed a method that consists of approximating the quantile loss function by a Huber-like function and a new coordinate descent algorithm which necessitates a differentiable loss function. The rq.pen function minimizes the objective function by using this approach. The proposed method is based on the Huber loss function proposed by Huber (1973) which is defined as

$$h_\omega(t) = \begin{cases} \frac{t^2}{2\omega}, & \text{if } |t| \leq \omega \\ \\ |t| - \frac{\omega}{2}, & \text{if } |t| \geq \omega. \end{cases} \tag{3.13}$$

It is important to recognize that the check loss function $\rho_\tau(a)$ can be written as follows

$$\rho_\omega^\tau(a) = a[\tau - I(a \leq 0)] = \frac{1}{2}(|a| + (2\tau - 1)a), \tag{3.14}$$

and for sufficiently small ω, $|a| = h_\omega(a)$. The Huber-approximated quantile loss function is defined as

$$h_\omega^\tau(a) = h_\omega(a) + (2\tau - 1)a, \tag{3.15}$$

and for small ω, $\rho_\tau(a) \approx \frac{1}{2}h_\omega^\tau(a)$. The advantage of the Huber loss function, is that it is a differentiable function which is important for developing an algorithm. The algorithm proposed by Yi and Huang (2015) is implemented in the hqreg package in R. The function hqreg solves the following minimization problem

$$\min_{\widetilde{\gamma}} \tau \frac{1}{2n} \sum_{i=1}^{n} h_\omega^\tau\Big(Y_i - \sum_{j=0}^{m+p} \widetilde{\boldsymbol{\mathcal{X}}}_{ij}^T(\boldsymbol{s}_i)\widetilde{\boldsymbol{\gamma}}_{j;\tau}\Big) + \nu \sum_{j=0}^{m+p} \|\widetilde{\boldsymbol{\gamma}}_{j;\tau}\|_2^2. \tag{3.16}$$

Cross-validation for penalty parameter ν is performed using the rq.pen.cv function. A sequence of values of ν is chosen automatically according to a method integrated in rq.pen.cv.

3.6 Simulations

In this section, we conduct simulation studies to evaluate the performance of the SVFQM introduced in Section 3.2. It is of importance to compare the SVFQM to existing quantile linear regression models, to determine if the SVFQM has any advantage over them.

The data generation for the SVFQM follows the same procedure introduced in Section 2.4.1 with a few small alterations. The main difference is the generation of the true conditional quantiles $Q_Y(\tau|\boldsymbol{X}_k, Z_k, \boldsymbol{s}_u)$ for $k = 1, \ldots, K,\ u = 1, \ldots, U$ where $K = 10$ replications and $U = 105$ locations. This is because the true conditional quantile is modeled as

$$Q_Y(\tau|\boldsymbol{X}_k, Z_k, \boldsymbol{s}_u) = F_\epsilon^{-1}(\tau) + \eta_{0;\tau}(\boldsymbol{s}_u) + Z_k^T(\boldsymbol{s}_u)\alpha_{1;\tau}(\boldsymbol{s}_u) + \int_{\mathcal{T}} \boldsymbol{X}_k^T(\boldsymbol{s}_u;t)\boldsymbol{\eta}_\tau(\boldsymbol{s}_u;t)dt,$$

where $F_\epsilon^{-1}(\tau)$ is the τth quantile of the errors. The kth replication for location u, $X_k(\boldsymbol{s}_u;t)$, is generated using the Karhunen-Loeve expansion of the $X(\boldsymbol{s};t)$ with $p = 5$ eigenfunctions where $\boldsymbol{X}_k(\boldsymbol{s}_u, t) = \{X_{k1}(\boldsymbol{s}_u, t), X_{k2}(\boldsymbol{s}_u, t)\}^T$, $k = 1, \ldots, K, u = 1, \ldots, U$, is generated by the Karhunen-Loeve decomposition using $p = 5$ eigenfunctions

$$\boldsymbol{X}_k(\boldsymbol{s}_u;t) = \sum_{r=1}^{5} \xi_{kr}(\boldsymbol{s}_u)\boldsymbol{\psi}_r(t), \tag{3.17}$$

where $\boldsymbol{\psi}_r(t) = \Big\{\cos(2\pi rt), \sin(2\pi rt)\Big\}$ for 100 time points over the time domain $[0, 1]$ and $\boldsymbol{\xi}_{kr} = \{\xi_{kr}(\boldsymbol{s}_1), \ldots, \xi_{kr}(\boldsymbol{s}_U)\}$ for $r = 1, \ldots, 5$ and $k = 1, \ldots, K$. The FPC scores $\boldsymbol{\xi}_{kr} \sim N(\boldsymbol{0}, \lambda_r\boldsymbol{\Sigma})$ where λ_r is the corresponding eigenvalue and $\boldsymbol{\Sigma}$ is the Matern correlation function introduced in Section 2.4.1. This can be used to rewrite $Q_Y(\tau|\boldsymbol{X}_k, Z_k, \boldsymbol{s}_u)$ as follows

$$Q_Y(\tau|\boldsymbol{X}_k, Z_k, \boldsymbol{s}_u) = F_\epsilon^{-1}(\tau) + \alpha_{0;\tau}(\boldsymbol{s}_u) + Z_k^T(\boldsymbol{s}_u)\alpha_{1;\tau}(\boldsymbol{s}_u) + \sum_{r=1}^{5} \xi_{kr}(\boldsymbol{s})\beta_{r;\tau}(\boldsymbol{s}),$$

where $\alpha_{0;\tau}, \alpha_{1:\tau}, \beta_{1;\tau}, \dots, \beta_{5;\tau}$ are generated as described in Section 2.4.1. The measurement errors for $Y_k(\boldsymbol{s}_u)$ are generated based on a SNR value of 3.3 as in Section 2.4.1 and their corresponding τth quantile $F_\epsilon^{-1}(\tau)$ is calculated. Then $Q_Y(\tau|\boldsymbol{X}_k, Z_k, \boldsymbol{s}_u)$ can be calculated for $k = 1, \dots, 10,\ u = 1, \dots, 105$.

Similarly to Section 2.4.3, the simulations are performed using six different settings to determine the applicability of the SVFQM. One of the advantages of quantile regression is the ability to gain a better understanding of the response variables distribution by using multiple quantiles. Any quantile can be chosen for $\tau \in (0, 1)$ but for simplicity, the quantiles used in the following simulations are 0.25, 0.50 and 0.75. Cross-validation is performed for the penalty parameter ν by using a sequence of ten automatically generated values provided by the rq.pen.cv function as discussed in Section 3.5. The models used in the following simulations are the SVFQM and the Functional Quantile Linear Regression Model (FQR). They are defined as

$$\textbf{SVFQM};\ Q_Y(\tau|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s}) = F_\epsilon^{-1}(\tau) + \eta_0\tau(\boldsymbol{s}) + \boldsymbol{Z}^T(\boldsymbol{s})\boldsymbol{\alpha}_\tau(\boldsymbol{s}) + \int_{\mathcal{T}} \boldsymbol{X}^T(\boldsymbol{s};t)\boldsymbol{\eta}_\tau(\boldsymbol{s};t)dt, \text{ and}$$

$$\textbf{FQR};\ Q_Y(\tau|\boldsymbol{X}, \boldsymbol{Z}) = F_\epsilon^{-1}(\tau) + \eta_{0;\tau} + \boldsymbol{Z}^T\boldsymbol{\alpha}_\tau + \int_{\mathcal{T}} \boldsymbol{X}^T(t)\boldsymbol{\eta}_\tau(t)dt.$$

3.6.1 Simulation Results

The simulation results for each setting were run for quantiles 0.25, 0.50, and 0.75. A similar measure to the MSE is used to assess the model performance. For a given train or testing set, let $\{Q_Y(\tau|\boldsymbol{X}_i, \boldsymbol{Z}_i, \boldsymbol{s}_i)\}_{i=1}^n$ be the true quantile values and $\{\widehat{Q}_Y(\tau|\boldsymbol{X}_i, \boldsymbol{Z}_i, \boldsymbol{s}_i)\}_{i=1}^n$ be the corresponding predicted quantiles. Then, the Quantile Squared Error (QSE) calculated at each fold is defined as follows,

$$\text{QSE}\{\widehat{Q}_Y(\tau|\boldsymbol{X}_i, \boldsymbol{Z}_i, \boldsymbol{s}_i)\} = \sum_{i=1}^{n}\{\widehat{Q}_Y(\tau|\boldsymbol{X}_i, \boldsymbol{Z}_i, \boldsymbol{s}_i) - Q_Y(\tau|\boldsymbol{X}_i, \boldsymbol{Z}_i, \boldsymbol{s}_i)\}^2.$$

The main smoothing parameter in the simulation setting is the penalty paramater ν. At each fold, the optimal penalty parameter is chosen over a grid of ten values through the cross-validation procedure described in Section 3.5. For each setting, the model fitting is repeated 25 times where each repetition uses 4 folds. The average error is calculated across all repetitions. The results for each quantiles can be seen in Tables 3.1-3.3 below.

		Quantile ($\tau = 0.25$)					
		Constant		**Linear**		**Complex**	
Model	**Train/Test**	**SC**	**No SC**	**SC**	**No SC**	**SC**	**No SC**
SVFQM	Train	0.07	0.073	0.038	0.039	0.27	0.27
FQR	Train	0.0024	0.0026	0.46	0.46	0.56	0.56
SVFQM	Test	0.073	0.071	0.038	0.039	0.29	0.28
FQR	Test	0.0024	0.0026	0.47	0.45	0.58	0.56

Table 3.1: Simulation Results for $\tau = 0.25$.

		Quantile ($\tau = 0.50$)					
		Constant		**Linear**		**Complex**	
Model	**Train/Test**	**SC**	**No SC**	**SC**	**No SC**	**SC**	**No SC**
SVFQM	Train	0.072	0.076	0.038	0.037	0.25	0.25
FQR	Train	0.0021	0.0024	0.42	0.42	0.48	0.48
SVFQM	Test	0.037	0.074	0.037	0.037	0.26	0.25
FQR	Test	0.0021	0.0025	0.43	0.42	0.49	0.48

Table 3.2: Simulation Results for $\tau = 0.50$.

		Quantile ($\tau = 0.75$)					
		Constant		**Linear**		**Complex**	
Model	**Train/Test**	**SC**	**No SC**	**SC**	**No SC**	**SC**	**No SC**
SVFQM	Train	0.071	0.073	0.043	0.042	0.26	0.26
FQR	Train	0.0026	0.0028	0.47	0.47	0.58	0.57
SVFQM	Test	0.074	0.073	0.043	0.042	0.27	0.26
FQR	Test	0.0026	0.0029	0.47	0.46	0.58	0.58

Table 3.3: Simulation Results for $\tau = 0.75$.

For all quantiles, the SVFQM outperforms the FQR in both the linear and complex settings, whereas the FQR performs better in the constant setting. This difference could be attributed to the SVFQM's greater ability to capture variations across locations, making it a more powerful model. However, in the case of constant coefficients, the SVFQM may be too sensitive, reacting to minor differences in distribution between locations. In contrast, the

FQR treats all observations collectively, which could explain its better performance in the constant setting.

There is no notable difference in the results between spatial correlation and no spatial correlation. In the case of the linear functions setting, the difference is drastic where the FQR error is around ten times greater than the SVFQM error. Under the complex setting, the difference is less stark but still exceptional with the FQR error being around two times greater than the SVFQM error. Across all quantiles, there is consistent behavior under the linear and complex settings, where the SVFQM outperforms the FQR pretty significantly. Overall, through these simulations we were able to see that the SVFQM not only works well for quantile regression but it also has shows an improvement over the FQR for spatially varying data. The SVFQM is able to achieve this because it considers location-specific effects where as the FQR does not.

3.7 Application

The results presented in section 3.6 provides some assurance regarding the strength of the SVFQM. Although this offers some insight into the SVFQM, it is necessary to assess its utility for real world data. In this section, the SVFQM is used to analyze the agricutural data presented in Section 2.5. Both the Kansas and Midwest data are used in the following section. Similarly to the simulations in Section 3.6, model fitting is applied for the 25th, 50th and 75th quantiles to obtain a comprehensive understanding of the response variables distribution. In the following section, we use the Quantile Exceedance Proportion (QEP) statistic (Zhu et al., 2022) to measure the prediction error for an estimator $\widehat{Q}_Y(\tau|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})$

of $Q_Y(\tau|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})$ which is defined as

$$QEP\{\widehat{Q}_Y(\tau|\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{s})\} = \frac{1}{n}\sum_{i=1}^{n} I(Y_i - \widehat{Q}_Y(\tau|\boldsymbol{X}_i, \boldsymbol{Z}_i, \boldsymbol{s}_i) > 0) - (1-\tau). \tag{3.18}$$

The QEP measures the models ability to estimate the $(1-\tau)$ quantile by comparing the observed proportion with the true proportion, $(1-\tau)$. If the model performs well, the QEP should be close to 0.

3.7.1 Kansas Results

In the following section, the SVFQM and FQR are applied to the Kansas dataset. There are two main smoothing parameters that need to be chosen, the penalty parameter ν and the number of FPC scores. Both the train and test errors are reported, the train error is used to understand the models ability to fit the data, but the number of FPC scores and the optimal penalty parameter are chosen based on the test error.

The penalty parameter ν could have a significant effect on the model performance. To determine the best value of ν, 5-fold CV is performed for each quantile $\tau \in [0.25, 0.50, 0.75]$ over a sequence of values. The optimal number of FPC scores could change depending on the quantile but for simplicity, 60% variation is used for the penalty parameter CV. The average of the absolute value of the QEP train and test results can be seen in Figures 3.1-3.3.

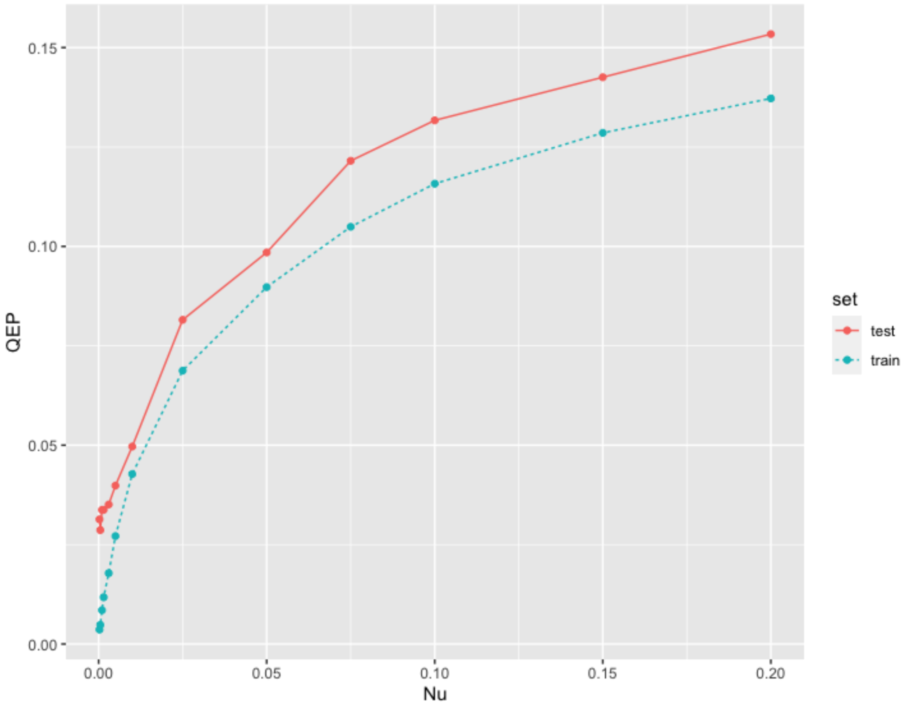


Figure 3.1: SVFQM Penalty Parameter CV for $\tau = 0.25$ (Kansas).

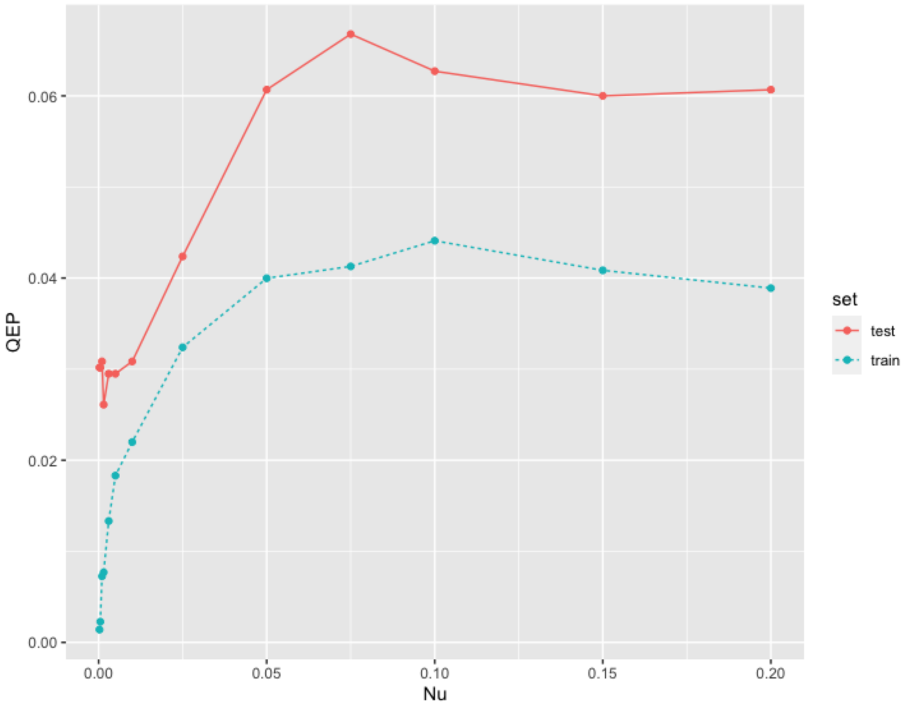


Figure 3.2: SVFQM Penalty Parameter CV for $\tau = 0.5$ (Kansas).

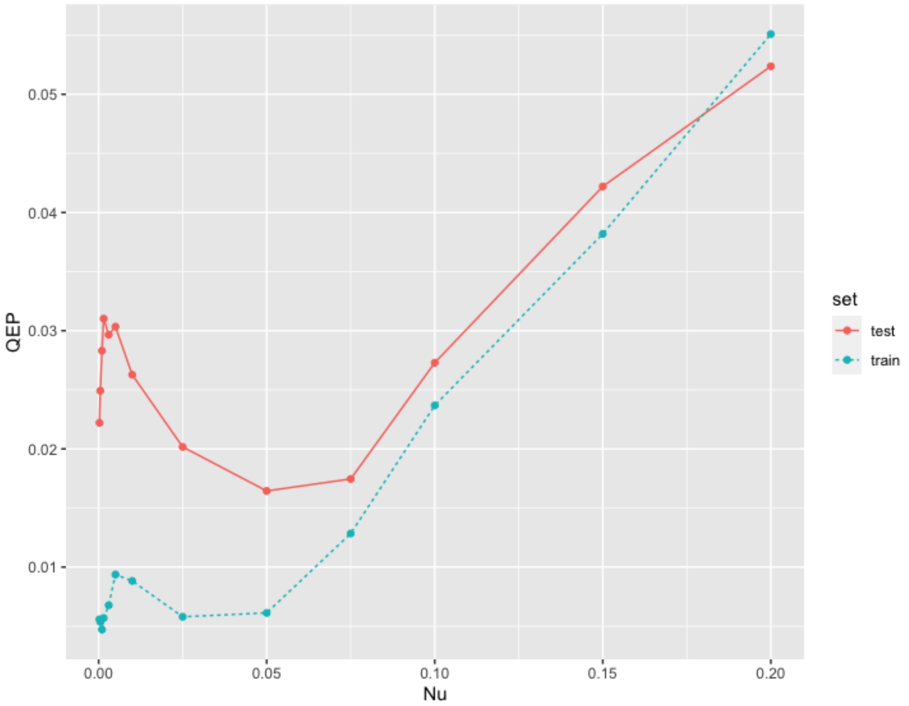
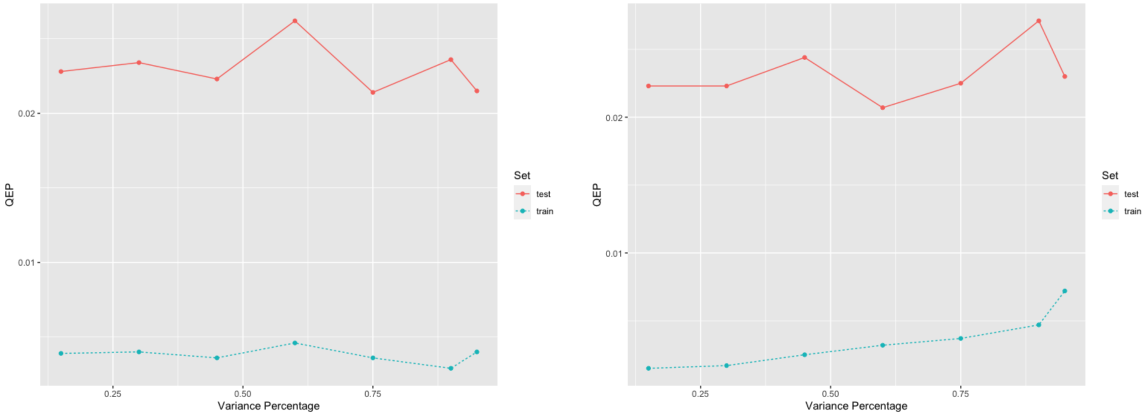


Figure 3.3: SVFQM Penalty Parameter CV for $\tau = 0.75$ (Kansas).

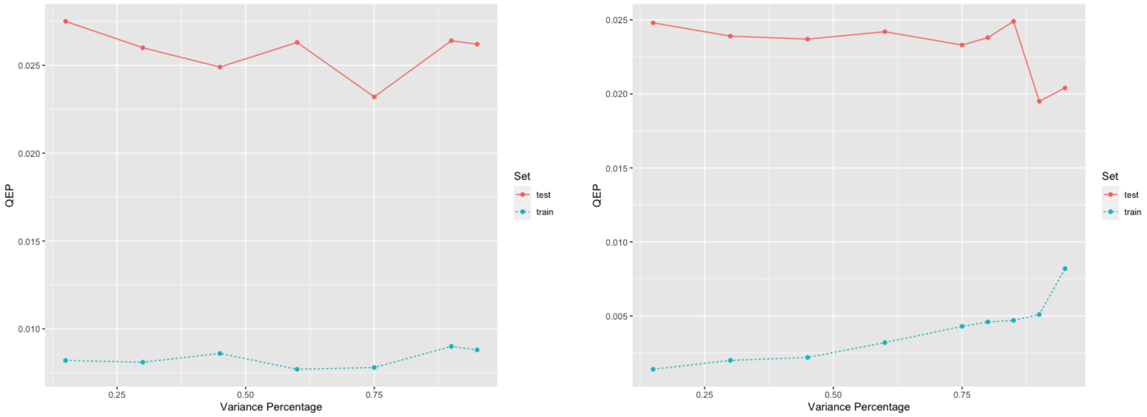
In Figure 3.1, there is a clear increasing trend in the QEP as ν increases. The minimum test QEP value for $\tau = 0.25$ corresponds to the smallest value of ν at 0.0005. Figure 3.2 shows a slightly different pattern, especially in the test curve in red. There is a slight decrease in the beginning and then the QEP increases pretty significantly up until around $\nu = 0.075$ but stabilizes after. The minimum test QEP value for $\tau = 0.5$ corresponds to $\nu = 0.0015$. In Figure 3.3, there is a slightly different behavior where the QEP decreases up until $\nu = 0.005$ and then increases significantly for the rest of the penalty values. This is the case for both the train and test curves but especially in the test curve there is a sharp dip in the QEP. The minimum test QEP value for $\tau = 0.75$ corresponds to $\nu = 0.05$. These values of ν are used throughout the remaining Kansas analysis.

As was seen in sections 2.5.2 and 2.5.3, the number of FPC scores play an important role in a models performance. The number of scores corresponds to the percentage of variation the scores explain. To determine the optimal number of scores, nine repetitions of 5-fold cross-validation is performed over a grid of values for the percentage of variation. This is done for the SVFQM and the FQR for the 25th, 50th and 75th quantiles. The average of the absolute value of the QEP train and test results for the Kansas data can be seen in Figures 3.4-3.6 below.



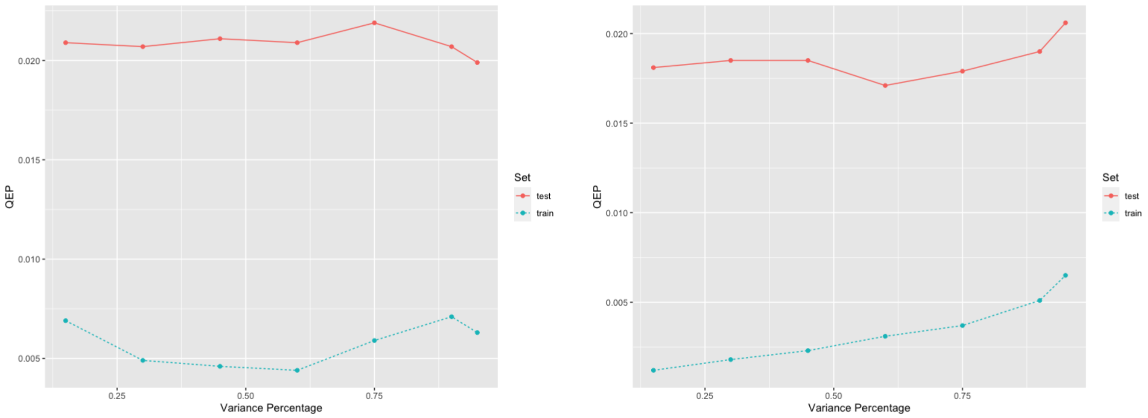
(a) SVFQM (b) FQR

Figure 3.4: QEP vs. Variance Percentage CV for $\tau = 0.25$ (Kansas).



(a) SVFQM (b) FQR

Figure 3.5: QEP vs. Variance Percentage CV for $\tau = 0.50$ (Kansas).



(a) SVFQM

(b) FQR

Figure 3.6: QEP vs. Variance Percentage CV for $\tau = 0.75$ (Kansas).

What is common to all quantiles and models, is that the number of FPC scores has a minimal effect on the QEP. Most notably, for the SVFQM in Figures 3.4a- 3.6a is that there is almost no change in the train QEP (in blue) as the number of FPC scores increases. There is a similar pattern in the FQR plots in Figures 3.4b-3.6b, where there are small increases in the train QEP but no significant changes. There is some change in the testing QEP in Figures 3.4a, 3.4b and 3.5a. With respect to the training curves, we can conclude that the model fitting isn't very sensitive to the number of FPC scores. The testing curves exhibit a similar behavior. The optimal number of scores is determined for each model and quantile based on the smallest value of the test QEP. The average of the absolute value of the QEP for all models and quantiles are displayed in Table 3.4 with respect to the optimal number of FPC scores.

Quantile	**Variance Percentage**	**Model**	**QEP**
0.25	0.75	SVFQM	0.0214
0.25	0.60	FQR	0.0207
0.50	0.75	SVFQM	0.0232
0.50	0.90	FQR	0.0195
0.75	0.90	SVFQM	0.0207
0.75	0.60	FQR	0.0171

Table 3.4: Kansas Model Comparison Results.

There is similar performance across all quantiles with the FQR performing slightly better than the SVFQM. The performance for $\tau = 0.25$ is almost equivalent where as for $\tau = 0.50$ and $\tau = 0.75$ there is a more of a difference between the SVFQM and FQR. The SVFQM should have better performance if there is spatial variation since it's able to model complicated functions across space. To determine if there is evidence for spatial variation, we can obtain another perspective of the SVFQM fit by plotting the coefficient heatmaps across the spatial domain which can be seen in Figures 3.7-3.9.

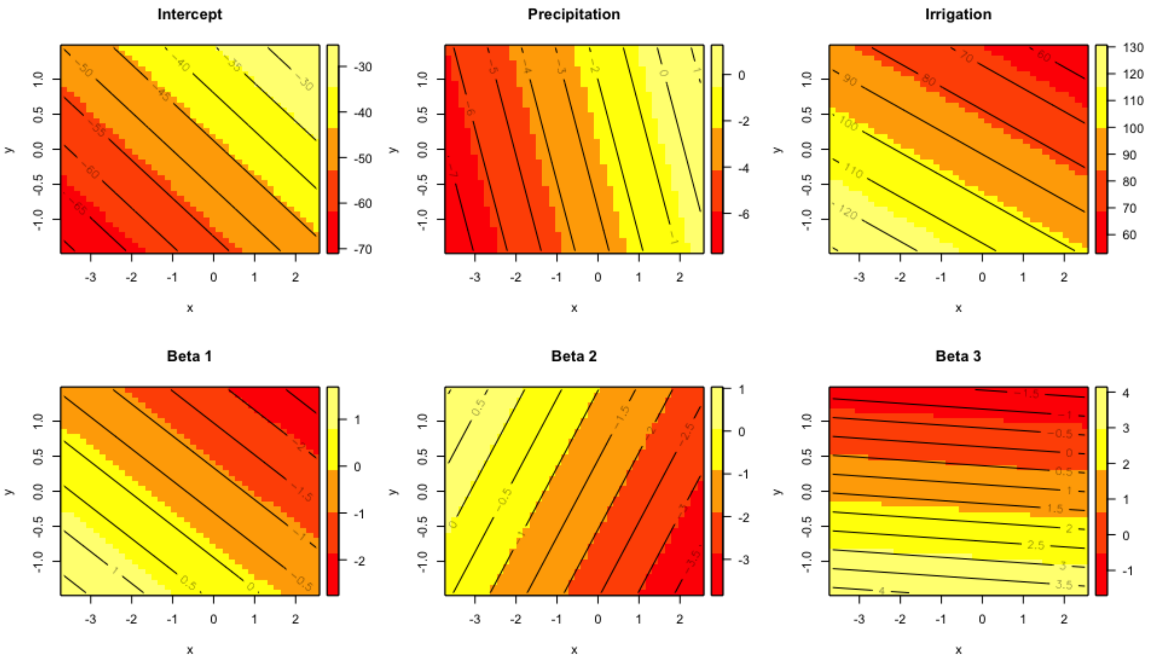


Figure 3.7: Kansas Coefficient Heatmaps for $\tau = 0.25$.

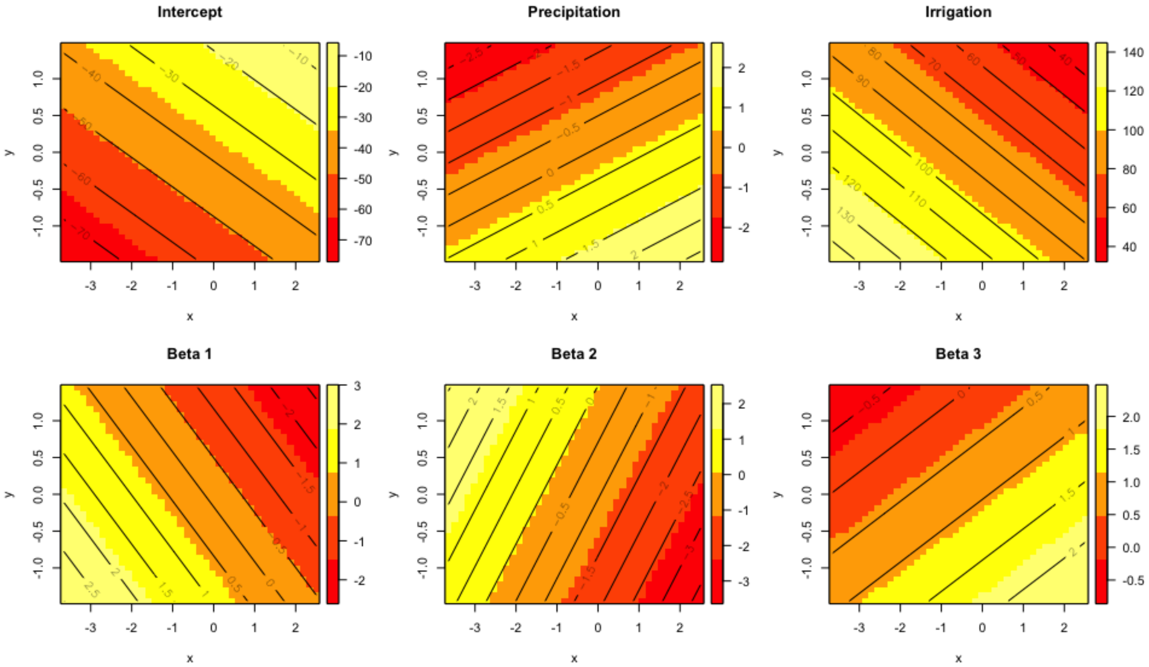


Figure 3.8: Kansas Coefficient Heatmaps for $\tau = 0.50$.

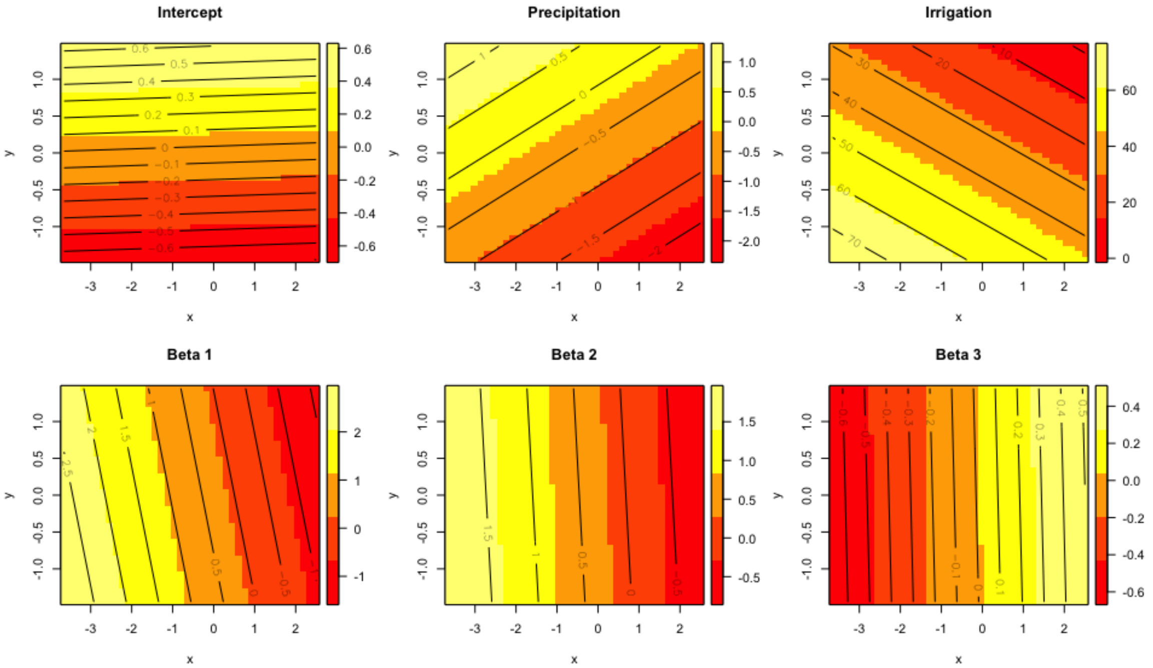


Figure 3.9: Kansas Coefficient Heatmaps for $\tau = 0.75$.

For all quantiles, the coefficient heatmaps clearly exhibit spatial variation across Kansas. Every coefficient heatmap displays a pattern of linear behavior across space. For example, the precipitation heatmap in Figure 3.7 has a slight positive linear behavior where as the β_2 heatmap has a slight negative linear behavior. In Figure 3.8, the β_1 heatmap has a negative linear behavior and the irrigation heatmap has a strong negative linear behavior. In Figure 3.9, the irrigation heatmap also has a strong negative linear behavior. The β_3 heatmap exhibits a weak positive linear behavior. Overall, the heatmaps in Figures 3.7-3.9 display considerable spatial variation. Based on this observation, the SVFQM should perform better than the FQR since it is able to model complicated spatially varying functions. Since this is not the case, there is another underlying problem with the SVFQM's performance.

3.7.2 Midwest Results

In the following section, the SVFQM and FQR are applied to the Midwest dataset. Both the train and test errors are reported, the train error is used to understand the models ability to fit the data, but the penalty parameter and number of FPC scores are chosen based on the test error.

Similarly to the Kansas data, cross-validation for the penalty parameter ν is performed for the Midwest data. 5-fold CV is performed for each quantile $\tau \in [0.25, 0.50, 0.75]$ over a grid of values. For consistency, 60% variation is used for penalty parameter CV. The average of the absoluate value of the QEP train and test results can be seen in Figures 3.10-3.12

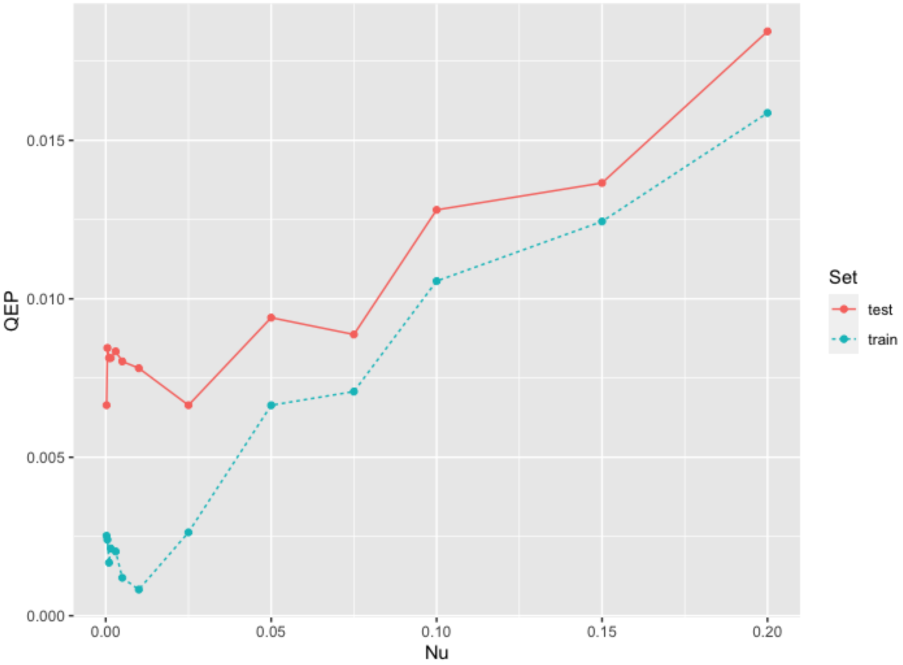


Figure 3.10: SVFQM Penalty Parameter CV for $\tau = 0.25$ (Midwest).

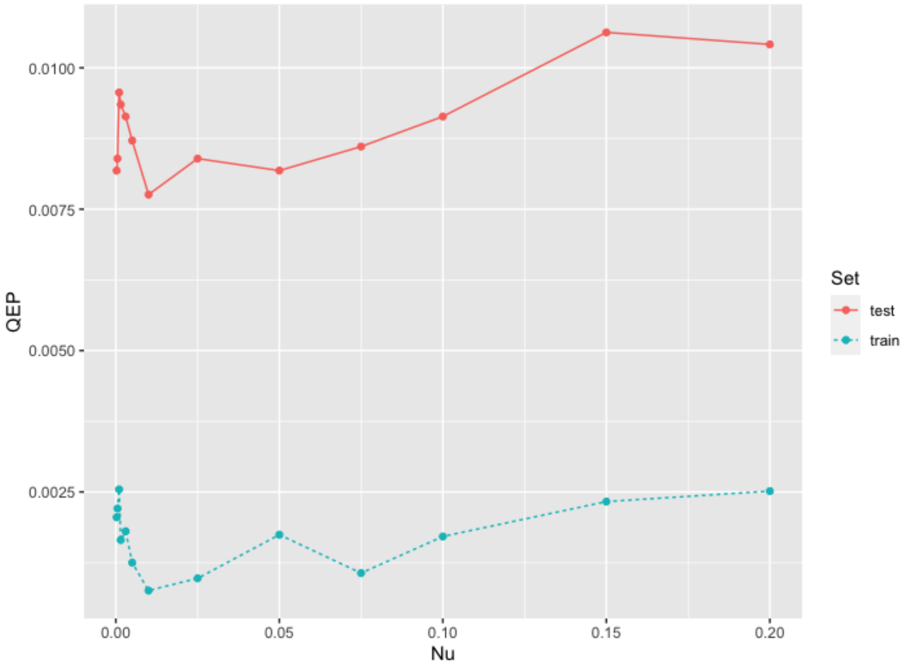


Figure 3.11: SVFQM Penalty Parameter CV for $\tau = 0.5$ (Midwest).

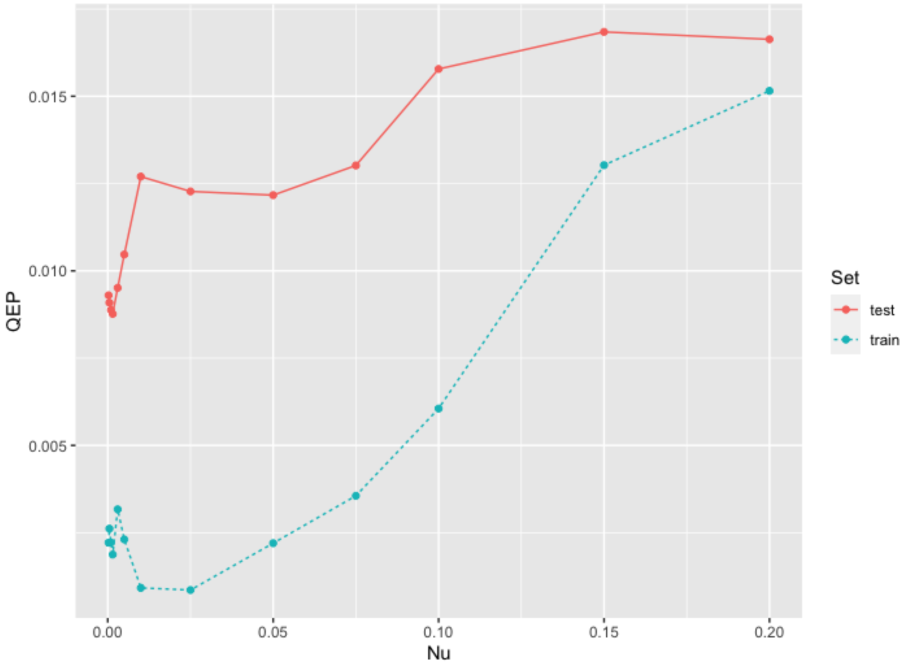
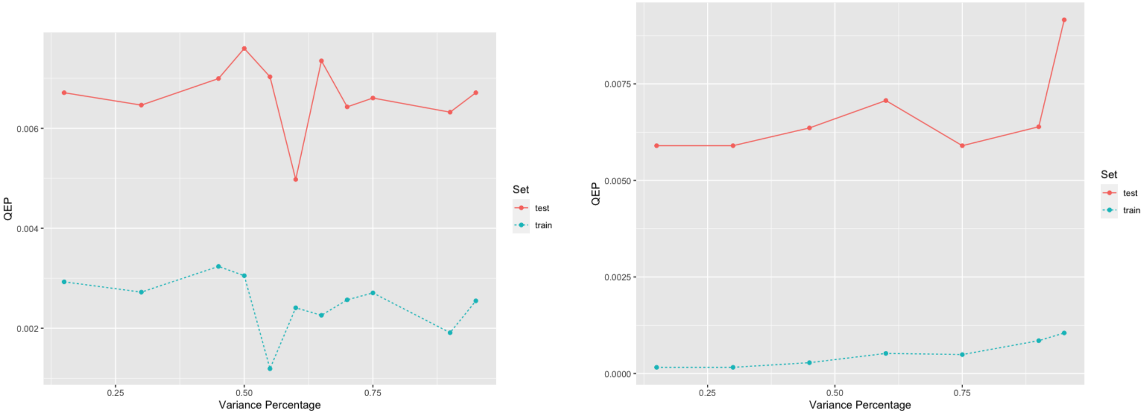


Figure 3.12: SVFQM Penalty Parameter CV for $\tau = 0.75$ (Midwest).

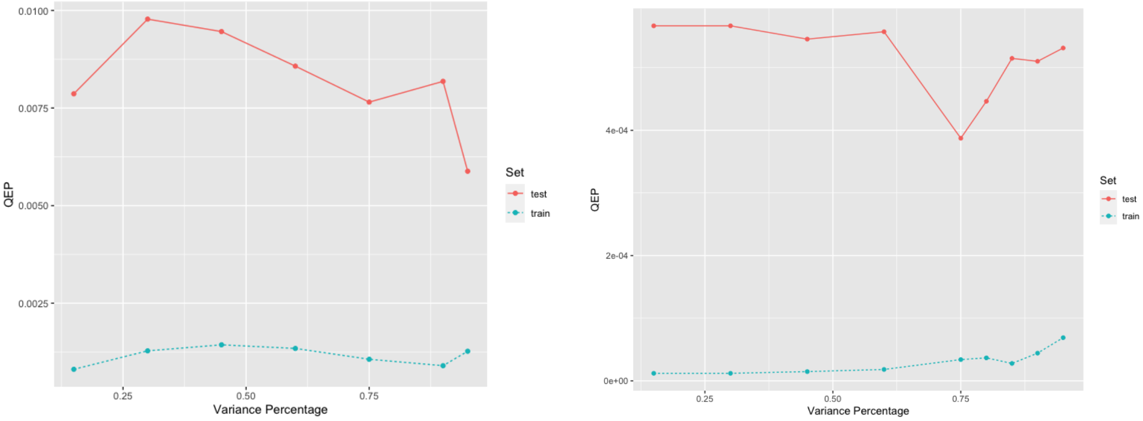
In Figure 3.10, there is a slight decrease in the QEP in the beginning but a sharp increase in the QEP as ν increases from 0.025 in both curves. The minimum test QEP value occurs at the values of $\nu = 0.0025$ and $\nu = 0.025$. The value of $\nu = 0.025$ is used for $\tau = 0.25$. In Figure 3.11 there is a slight decrease in the beginning of the plot but only minor increases in the QEP as ν increases, particularly in the training curve. The minimum test QEP value for $\tau = 0.5$ corresponds to $\nu = 0.01$. In Figure 3.12, the QEP values vary significantly as ν increases, especially in the training curve. There is a sharp increase in the testing curve in the beginning but then smaller increases. In the training curve, there is a slight decrease in the beginning and then a sharp increase for the remaining values of ν. The minimum test QEP value for $\tau = 0.75$ corresponds to $\nu = 0.0015$.

Similarly to Section 2.5.3, the optimal number of FPC scores for the Midwest could differ from Kansas. To determine the optimal number of FPC scores, nine repetitions of 5-fold cross-validation is performed over a grid of values for the percentage of variation. This is done for the SVFQM and the FQR for the 25th, 50th and 75th quantiles. The average of the absolute value of the QEP train and test results for the Midwest data can be seen in Figures 3.13-3.15 below.



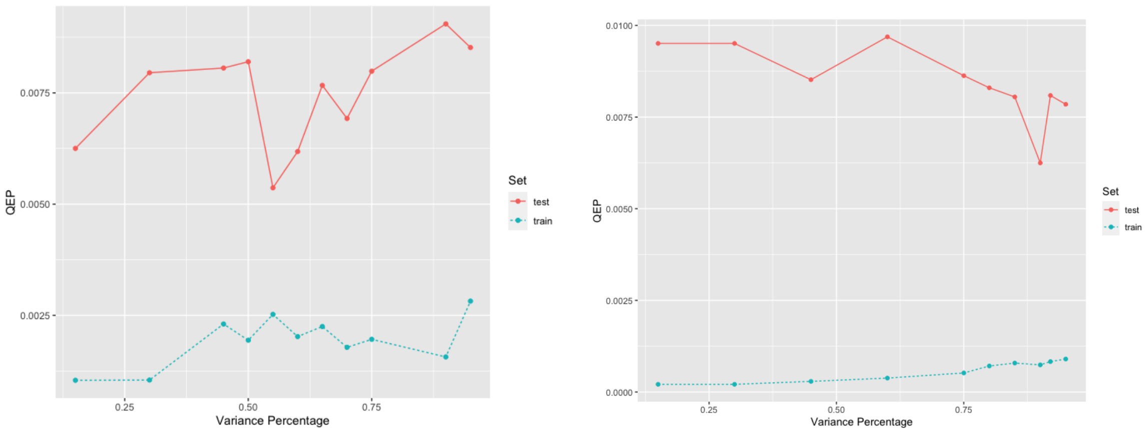
(a) SVFQM (b) FQR

Figure 3.13: QEP vs. Variance Percentage CV for $\tau = 0.25$ (Midwest).



(a) SVFQM (b) FQR

Figure 3.14: QEP vs. Variance Percentage CV for $\tau = 0.50$ (Midwest).



(a) SVFQM

(b) FQR

Figure 3.15: QEP vs. Variance Percentage CV for $\tau = 0.75$ (Midwest).

In contrast to the Kansas plots in Figures 3.4-3.6, there is more variation with respect to the number of FPC scores for the Midwest CV plots. This is especially apparent for the SVFQM in Figures 3.13a-3.15a where there is sharp changes in the test QEP as the variance percentage increases. There is noticeably less of a difference across the variance percentage for the FQR in Figures 3.13b-3.15b, especially in the training curve which is similar to the behavior seen in the Kansas CV plots. For the Midwest data, it seems that the number of FPC scores has a more significant affect on the SVFQM than on the FQR. This is quite different behavior from what was seen for the Kansas data since the FPC scores didn't have much of an effect on either model. An interesting observation is that the SVFQM training values are significantly larger than the FQR training values relative to the Kansas training results in Figures 3.4-3.6. This means that the SVFQM's ability to fit the data is notably worse than that of FQR where as in the Kansas data, the SVFQM and FQR's ability to fit the data was around the same. This could be attributed to the fact that there are significantly more parameters in the Midwest model compared to the Kansas model due to triangulation over a much larger spatial domain. The optimal number of scores is determined based on the testing QEP. The average of the absolute value of the QEP for all models and quantiles can be seen in Table 3.5.

Quantile	**Variance Percentage**	**Model**	**test QEP**
0.25	0.60	SVFQM	0.00721
0.25	0.15	FQR	0.00809
0.50	0.95	SVFQM	0.00955
0.50	0.80	FQR	0.00902
0.75	0.55	SVFQM	0.00612
0.75	0.90	FQR	0.00851

Table 3.5: Midwest Model Comparison Results.

For $\tau = 0.25$ and $\tau = 0.75$, the SVFQM test QEP values are smaller than the SVFQM. The FQR has a slight advantage over the SVFQM for $\tau = 0.50$. Even though the SVFQM's performance is better on average for $\tau = 0.25$ and $\tau = 0.75$, the difference isn't significant. This could be because of the difference in the SVFQM and FQR training QEP values that was seen in Figures 3.13a-3.15a. The SVFQM is a stronger model in the sense that it can model complicated functions over the spatial domain. To determine if the coefficients are spatially varying or not, the SVFQM coefficient heatmaps can be plotted across the spatial domain. The coefficient heatmaps for each value of τ can be seen in Figures 3.16 -3.18.

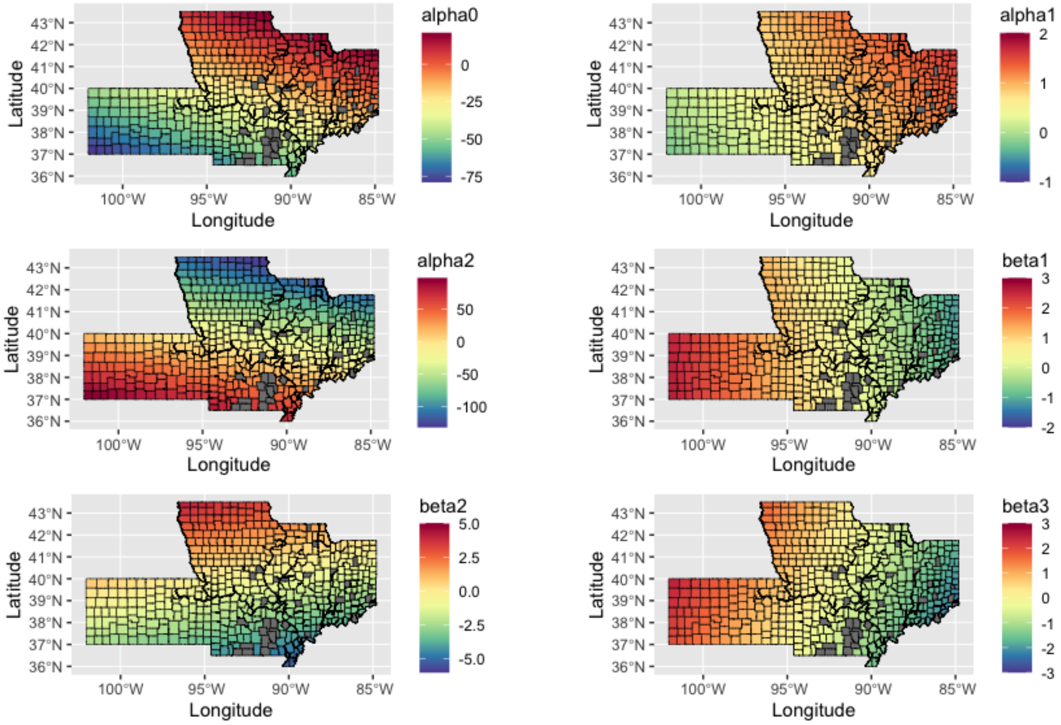


Figure 3.16: Midwest Coefficient Heatmaps for $\tau = 0.25$.

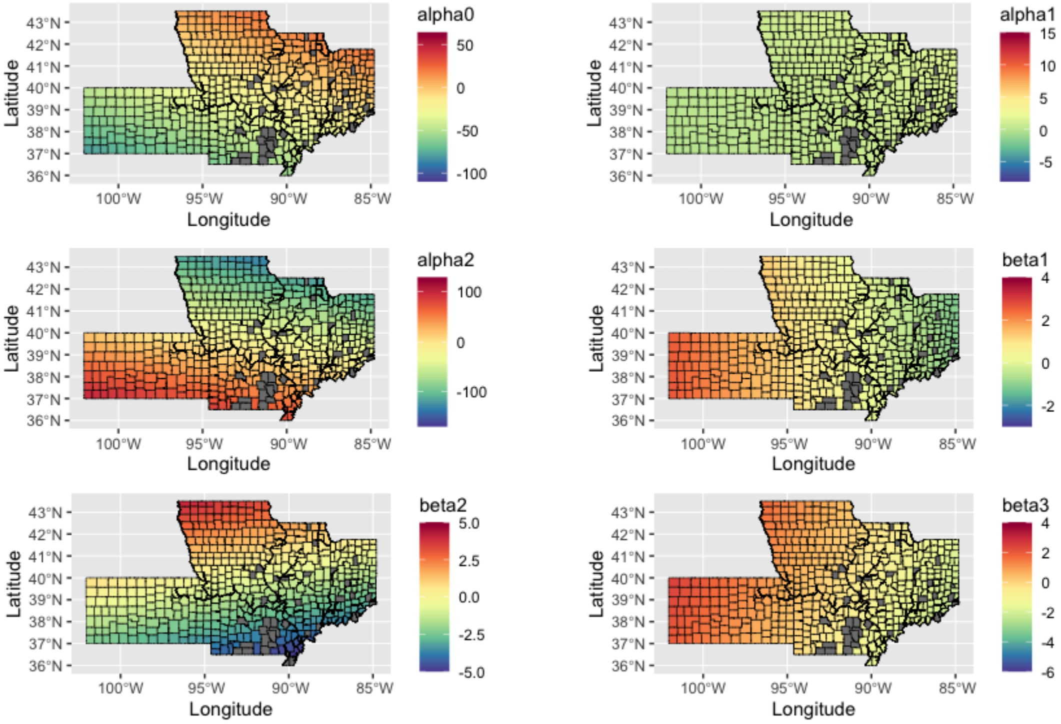


Figure 3.17: Midwest Coefficient Heatmaps for $\tau = 0.50$.

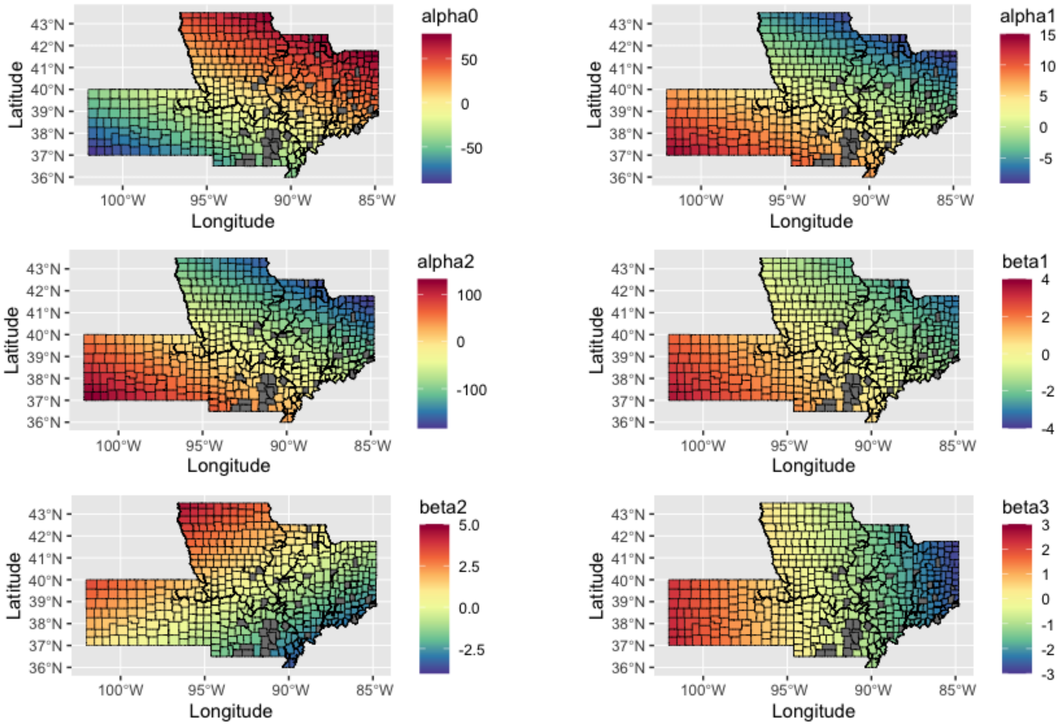


Figure 3.18: Midwest Coefficient Heatmaps for $\tau = 0.75$.

Across all quantiles, it is immediately clear that there is some degree of spatial variation with respect to the coefficients. There are some locations where the contribution proportion of a given variable is stronger than others. For example, in Figure 3.16, the contribution in the top center region of α_0 and β_2 heatmaps is very strong in the positive direction while the α_2 heatmap (irrigation) is very strong in the negative direction. In contrast, in the β_1 heatmap, the contribution in the same region is essentially zero. In the lower left region of the heatmaps for α_2, β_1, and β_3, there is a strong contribution in the positive direction while in the α_0 heatmap, there is a strong contribution in the negative direction. In the α_1 heatmap, there is almost no effect in the same region. In Figure 3.17, the top center region of the α_0, β_2 and β_3 heatmaps, have a moderate positive effect where as in the α_1 and β_1 heatmaps, there is almost no effect. In Figure 3.9, the top center region of the α_0 and β_2

heatmaps, there is a strong contribution in the positive direction while there is a strong negative contribution in the α_1 and α_2 heatmaps. An interesting observation is that in the lower left region of the heatmaps for $\alpha_1, \alpha_2, \beta_1$, and β_3 has a strong positive effect while the α_0 and β_2 heatmaps have a strong negative and minimal effect, respectively. The heatmaps displayed in Figures 3.16-3.18 clearly exhibit spatial variation. The SVFQM should be able to perform significantly better than the FQR under this setting. This means that there is another underlying factor that is affecting the SVFQM's performance.

3.8 Conclusion

In this chapter, the SVFQM was introduced as an extension of the GSVFM to estimate conditional quantiles using spatially varying functional data. The estimation method used for the SVFQM involves a reparametrization method. In the simulations, the SVFQM showed a significant improvement over the FQR model. The real data results didn't provide the same kind of performance since for the Kansas data, both models have similar performance, and for the Midwest data, the SVFQM only has a slight advantage over the FQR. As was highlighted in Section 3.7.2, this could be seen by comparing the SVFQM training values to the FQR training values. This behavior could be attributed to the increase in parameters in the Midwest SVFQM. This type of behavior can also be seen in the Kansas data when the number of triangles is increased. The Kansas and Midwest heatmaps displayed in Sections 3.7.1 and 3.7.2 clearly depict spatial variation, raising the question of why the SVFQM does not significantly outperform the FQR. One possible explanation is that the SVFQM's location-specific coefficients are based on only a small number of observations in the vicinity, whereas the FQR uses all observations across the spatial domain to calculate its coefficients. The results in this chapter provide evidence that the SVFQM provides reasonable results but needs to be investigated further.

Chapter 4

Conclusions

4.1 Summary

In Chapter 2, the GSVFM was introduced as a functional model that accounts for spatially varying functional data, marking a significant methodological advancement since earlier functional models did not incorporate spatial effects. The SVFM, initially introduced by Park et al. (2022), utilized a Bayesian framework for estimation, but Bayesian models often involve complex and time-consuming computations. The GSVFM, on the other hand, offers a more efficient estimation framework through a novel two-step procedure and generalizes the SVFM by allowing for different link functions. These features make the GSVFM versatile and applicable to a wide range of settings. The simulation and real data results presented emphasize the importance of including location-specific effects when analyzing spatially varying functional data. The use of triangulation to account for these effects allows for the estimation of bivariate functions and provides flexibility when dealing with irregular spatial domains.

In Chapter 3, the SVFQM was introduced to underscore the value of considering location-specific effects when predicting quantiles. This represents a major advancement, as existing functional quantile regression models do not account for spatial effects. The SVFQM shares a similar model formulation with the GSVFM but differs in its estimation and interpretation. The SVFQM estimation leverages a novel reparametrization approach to enable penalization. The simulation results highlight the importance of incorporating location-specific effects while the real data results show promise but still require some improvements.

4.2 Future Work

There are several potential directions for future work on the GSVFM. One key area to explore is developing convergence rates for the GSVFM's functional coefficient estimator, which would provide strong theoretical guarantees for the model's overall performance. Additionally, the GSVFM framework could be extended to handle a 3D spatial domain, which would be useful in applications such as weather forecasting and brain imaging data. Another promising avenue is integrating the functional and triangulation methods with deep neural network models to enhance predictive capabilities.

While the SVFQM demonstrates reasonable predictive performance, there is still room for improvement. The reparametrization method was employed to incorporate the spatial penalty into the model due to the lack of existing R packages capable of handling the original quantile regression objective function with a penalty. Although this approach has been effective, developing an algorithm to solve for the estimators using the original objective function could potentially enhance prediction performance. One additional goal in creating a new algorithm would be to improve performance when dealing with a larger number of parameters. Currently, in both the GSVFM and SVFQM, cross-validation for the penalty parameter and the number of FPC scores is performed independently. It's possible that a different combination of the penalty parameter and the number of scores could be selected if cross-validation is performed simultaneously, rather than choosing the smoothing parameters independently. By optimizing both parameters together, the model might achieve a more balanced trade-off between smoothness and flexibility, potentially leading to improved prediction performance. Therefore, developing a two-dimensional cross-validation approach for jointly selecting the penalty parameter and the number of FPC scores is of particular interest.

Bibliography

Azzimonti, L., L. M. Sangalli, P. Secchi, M. Domanin, and F. Nobile (2015). Blood flow velocity field estimation via spatial regression with pde penalization. *Journal of the American Statistical Association 110*, 1057 – 1071.

Bosq, D. (2000). *Linear Processes in Function Spaces: Theory and Applications.* Lecture Notes in Statistics. Springer New York.

Brunsdon, C., S. A. Fotheringham, and M. Charlton (2010). Geographically weighted regression: A method for exploring spatial nonstationarity. *Geographical Analysis 28*, 281–298.

Cai, T. T. and P. Hall (2006). Prediction in functional linear regression. *Annals of Statistics 34*, 2159–2179.

Cai, Z., J. Fan, and R. Li (2000). Efficient estimation and inferences for varying-coefficient models. *Journal of the American Statistical Association 95*, 888 – 902.

Cardot, H., C. Crambes, and P. Sarda (2007, March). Quantile regression when the covariates are functions. *arXiv Mathematics e-prints*, math/0703056.

Chen, K. and H.-G. Müller (2011, 10). Conditional Quantile Analysis When Covariates are Functions, with Application to Growth Data. *Journal of the Royal Statistical Society Series B: Statistical Methodology 74*(1), 67–89.

Fan, J. and W. Zhang (1999). Statistical estimation in varying coefficient models. *The Annals of Statistics 27*(5), 1491 – 1518.

Ferraty, F., A. Rabhi, and P. Vieu (2005). Conditional quantiles for dependent functional data with application to the climatic "el niño" phenomenon. *Sankhyā: The Indian Journal of Statistics (2003-2007) 67*(2), 378–398.

Gornott, C. and F. Wechsung (2016). Statistical regression models for assessing climate impacts on crop yields: A validation study for winter wheat and silage maize in germany. *Agricultural and Forest Meteorology 217*, 89–100.

Guan, K., B. Sultan, M. Biasutti, C. Baron, and D. B. Lobell (2017). Assessing climate adaptation options and uncertainties for cereal systems in west africa. *Agricultural and Forest Meteorology 232*, 291–305.

Hastie, T. J. and R. Tibshirani (1993). Varying-coefficient models. *Journal of the royal statistical society series b-methodological 55*, 757–779.

Horváth, L. and P. Kokoszka (2012). *Inference for Functional Data with Applications.* Springer Series in Statistics. Springer New York.

Huber, P. J. (1973). Robust regression: Asymptotics, conjectures and monte carlo. *Annals of Statistics 1*, 799–821.

Hunter, D. R. and K. Lange (2000). Quantile regression via an mm algorithm. *Journal of Computational and Graphical Statistics 9*(1), 60–77.

Jones, J. W., J. M. Antle, B. Basso, K. J. Boote, R. T. Conant, I. Foster, H. C. J. Godfray, M. Herrero, R. E. Howitt, S. Janssen, et al. (2017). Brief history of agricultural systems modeling. *Agricultural systems 155*, 240–254.

Kang, Y., M. Ozdogan, X. Zhu, Z. Ye, C. Hain, and M. Anderson (2020). Comparative assessment of environmental variables and machine learning algorithms for maize yield prediction in the us midwest. *Environmental Research Letters 15*(6), 064005.

Kato, K. (2012). Estimation in functional linear quantile regression. *The Annals of Statistics 40*(6), 3108–3136.

Kern, A., Z. Barcza, H. Marjanović, T. Árendás, N. Fodor, P. Bónis, P. Bognár, and J. Lichtenberger (2018). Statistical modelling of crop yield in central europe using climate data and remote sensing vegetation indices. *Agricultural and forest meteorology 260*, 300–320.

Khmaladze, E. V. (1982). Martingale approach in the theory of goodness-of-fit tests. *Theory of Probability and Its Applications 26*, 240–257.

Kim, M. and L. Wang (2021). Generalized spatially varying coefficient models. *Journal of Computational and Graphical Statistics 30*, 1 – 10.

Koenker, R. and G. Bassett (1978). Regression quantiles. *Econometrica 46*(1), 33–50.

Koenker, R., P. Ng, and S. Portnoy (1994). Quantile smoothing splines. *Biometrika 81*(4), 673–680.

Koenker, R. and B. J. Park (1996). An interior point algorithm for nonlinear quantile regression. *Journal of Econometrics 71*(1), 265–283.

Koenker, R. and Z. Xiao (2002). Inference on the quantile regression process. *Econometrica 70*(4), 1583–1612.

Kokoszka, P. and M. L. Reimherr (2017). Introduction to functional data analysis. Chapman and Hall/CRC.

Lai, M. J. and L. L. Schumaker (2007). Spline functions on triangulations. In *Encyclopedia of mathematics and its applications.*

Lai, M. J. and L. Wang (2013). Bivariate penalized splines for regression. *Statistica Sinica.*

Lin, Y. and H. H. Zhang (2006). Component selection and smoothing in multivariate nonparametric regression. *Annals of Statistics 34*, 2272–2297.

Mu, J., G. Wang, and L. Wang (2018). Estimation and inference in spatially varying coefficient models. *Environmetrics 29.*

Park, Y., B. Li, and Y. Li (2022). Crop yield prediction using bayesian spatially varying coefficient models with functional predictors. *Journal of the American Statistical Association.*

Peng, B., K. Guan, M. Chen, D. M. Lawrence, Y. Pokhrel, A. Suyker, T. Arkebauer, and Y. Lu (2018). Improving maize growth processes in the community land model: Implementation and evaluation. *Agricultural and forest meteorology 250*, 64–89.

Prasad, A. K., L. Chai, R. P. Singh, and M. Kafatos (2006). Crop yield estimation model for iowa using remote sensing and surface parameters. *International Journal of Applied earth observation and geoinformation 8*(1), 26–33.

Ramsay, J. and B. Silverman (2005). *Functional Data Analysis: With 151 illustrations.* Springer.

Ramsay, T. (2002). Spline smoothing over difficult regions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64*.

Ray, D. K., J. S. Gerber, G. K. MacDonald, and P. C. West (2015). Climate variation explains a third of global crop yield variability. *Nature Communications 6*.

Sangalli, L. M., J. O. Ramsay, and T. Ramsay (2013). Spatial spline regression models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 75*.

Schabenberger, O. and C. A. Gotway (2005). *Statistical Methods for Spatial Data Analysis.* Chapman Hall/CRC.

Scott-Hayward, L., M. L. Mackenzie, C. Donovan, C. G. Walker, and E. Ashe (2014). Complex region spatial smoother (cress). *Journal of Computational and Graphical Statistics 23*, 340 – 360.

Stein, M. (2012). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Series in Statistics. Springer New York.

Sun, Y., H. Yan, W. Zhang, and Z. Lu (2014). A semiparametric spatial dynamic model. *Annals of Statistics 42*, 700–727.

Tang, J., Y. Li, and Y. Guan (2016). Generalized quasi-likelihood ratio tests for semiparametric analysis of covariance models in longitudinal data. *Journal of the American Statistical Association 111*, 736 – 747.

USDA (2020a). Crop Production 2019 Summary. `https://www.nass.usda.gov/Publications/Todays_Reports/reports/cropan20.pdf`.

USDA (2020b). What is agriculture's share of the overall U.S. economy? `https://www.ers.usda.gov/data-products/chart-gallery/gallery/chart-detail/?chartId=58270`.

Van Klompenburg, T., A. Kassahun, and C. Catal (2020). Crop yield prediction using machine learning: A systematic literature review. *Computers and electronics in agriculture 177*, 105709.

Wang, H. and M. G. Ranalli (2007). Low-rank smoothing splines on complicated domains. *Biometrics 63*.

Wang, J.-L., J.-M. Chiou, and H.-G. Müller (2016). Functional data analysis. *Annual Review of Statistics and Its Application 3*(Volume 3, 2016), 257–295.

Watson, G. S. (1973). Distribution theory for tests based on the sample distribution function. Society for Industrial and Applied Mathematics.

Wilhelm, M. and L. M. Sangalli (2016). Generalized spatial regression with differential regularization. *Journal of Statistical Computation and Simulation 86*(13), 2497–2518.

Wong, R. K. W., Y. Li, and Z. Zhu (2018). Partially linear functional additive models for multivariate functional data. *Journal of the American Statistical Association 114*, 406 – 418.

Wood, S. N., M. V. Bravington, and S. L. Hedley (2008). Soap film smoothing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70.*

Yao, F., S. Sue-Chee, and F. Wang (2017). Regularized partially functional quantile regression. *Journal of Multivariate Analysis 156*, 39–56.

Yi, C. and J. Huang (2015). Semismooth newton coordinate descent algorithm for elastic-net penalized huber loss regression and quantile regression. *Journal of Computational and Graphical Statistics 26*, 547 – 557.

Yu, K. and M. C. Jones (1998). Local linear quantile regression. *Journal of the American Statistical Association 93*, 228–237.

Zhu, H., R. Zhang, Y. Li, and W. Yao (2022). Estimation for extreme conditional quantiles of functional quantile regression. *Statistica Sinica 32 4*, 1767–1787.

Appendix A

Data Preparation

In any data analysis, the quality of the results heavily depends on the quality of the data. If the data is not properly collected and processed, there is a risk of producing misleading conclusions. Therefore, ensuring accurate data collection and careful processing is essential and should be handled with the utmost care.

The Midwest data used in this dissertation was obtained from public government databases, specifically the United States Department of Agriculture (USDA) and the National Oceanic and Atmospheric Administration (NOAA). Two distinct datasets were utilized in the research. The non-functional data, sourced from the USDA, includes agricultural information such as yield, area, and irrigated area, along with identifiers for location and year. The functional data, sourced from NOAA, consists of time series data for minimum and maximum temperatures and precipitation across five Midwest states: Kansas, Missouri, Illinois, Indiana, and Iowa.

A.1 Non-functional Data

The agricultural data was obtained from the USDA database, accessible for all states via `quickstats.nass.usda.gov`. To retrieve the desired data for a specific state, a set of query variables must be selected, such as program, sector, commodity, category, and year. The primary variables collected for each Midwest state include corn yield, total area, irrigated area, and associated identifier variables.

The data provided by the USDA is not "clean", requiring several preprocessing steps. Many observations (county and year combinations) have missing yield values. Since the yield is an average across hundreds of farms within a county, it is not practical to impute missing values from nearby counties, so these observations are removed from the dataset.

For the irrigated area variable, which is crucial to the analysis, there are also many missing values. Instead of discarding these observations, the missing values are filled by calculating the average irrigated area (over all years) for each county in the Midwest. Missing values for a given county are then replaced with this average. Additionally, the scale of the irrigated area variable is quite large, which could potentially skew the results. Since the raw values of irrigated area don't provide direct insight (e.g., we don't know what percentage of the total area is irrigated), the variable is transformed into a proportion. This is done by dividing the irrigated area by the total area for each county. Even after this transformation, some missing values remain, which are replaced with 0 to complete the dataset.

A.2 Functional Data

The functional data in this analysis consists of weather data, including daily minimum and maximum temperatures and daily precipitation. This data is accessed through NOAA using an API, handled via ACIS web services, which can be found at `rcc-acis.org/docs_webservices.html`. The first step in extracting the weather data involves connecting to a server to retrieve the data for the desired counties and years. This process is performed separately for each Midwest state.

After extracting the minimum and maximum temperature, and the precipitation data, some missing values need to be addressed. It is reasonable to assume that nearby counties will have similar weather on a given day. Therefore, for each county, missing data for a specific day is replaced with the weather data from its nearest neighboring county (within a certain distance). However, some missing values may still remain if neighboring counties also have missing data. To handle these, each year is examined, and if a year has more than 151 missing values (days), it is excluded from the dataset. For years with fewer than 151 missing values, consecutive sequences of missing data are identified, ensuring the sequence length does not exceed $\frac{365}{3.5}$ days. These consecutive sequences of missing values are filled using the average of the values at the beginning and end of the sequence, respectively. Any remaining years with missing values after this process are removed from the dataset.

Daily precipitation is recorded for each county and year combination. Since daily precipitation likely has minimal impact on the total yield of a given harvest over several months, the yearly average precipitation is used instead. This average is calculated for each county-year combination and is included as a variable in the non-functional dataset.