A comparative study of nonparametric spatial models

by

Natalie Grills

B.A., Kansas State University, 2021

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2023

Approved by:

Major Professor
Jingru Mu

Copyright

Abstract

From the mid-twentieth century into the early 2000s, there have been significant developments and expansions in the world of nonparametric regression modeling. In particular, the advancement of nonparametric spatial regression models allows for the creation of models that better fit spatial data. In this report, we conducted a study to illustrate how the development of nonparametric spatial regression methods produced models that more closely fit spatial data. To do so, we compared the fits of select linear regression, nonparametric regression, and spatial regression models for spatial data. The following are the specific models investigated: multivariate linear regression, cubic B-spline, univariate kernel smoothing, geographically weighted regression (GWR), and cubic B-spline tensor product. First, a spatial simulation population data set was created using predefined coefficient functions, and samples of this population were used to fit each of the five models of interest. The mean integrated squared error (MISE) was calculated for the coefficient estimates of the models with functional coefficients. Across the 100 simulated samples, the B-spline tensor product model produced the smallest average mean squared error (MSE), yet larger MISE values for two of the four coefficients compared to the linear regression and GWR models. Second, an analysis of the 1970 Boston housing prices data set, which is provided in the package `mlbench`, was performed using 10-fold cross-validation for each of the models of interest. The mean squared prediction error (MSPE) was calculated for each combination of model and fold. The B-spline tensor product model had the smallest average MSPE, while the linear model had the largest average MSPE.

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I am thankful to my major professor, Dr. Jingru Mu, for giving me the opportunity to learn about nonparametric spatial regression models and their applications outside of the classroom. Her knowledge of the topics covered in this paper was integral to the paper's writing. Many thanks to the members of my supervisory committee for guiding me through this paper and multiple aspects of my education at K-State.

I would like to give special thanks to my peers in the Department of Statistics for the study groups and grading parties during my time as a graduate student at K-State. Without them, obtaining my degree would have been less entertaining by a statistically significant amount.

Finally, the endeavor would not have been possible without the support of the people closest to me. My parents' encouragement throughout my education has always emboldened me to pursue my dreams and tackle projects that I would have otherwise thought to be impossible. I would also like to thank Seth Hoffman for keeping my spirits high and acting as my moral support during this process.

Chapter 1

Introduction

1.1 Background of Methodology

In the past century, considerable strides have been made in the world of nonparametric regression modeling for spatially varying data. To confidently continue moving forward, it is important to reflect on progress that has been made. We will explore foundations on which further developments in spatial statistics can grow. The models discussed in this paper are limited to those listed as precursors to spatially varying coefficient models in the 2017 paper "Estimation and inference in spatially varying coefficient models" (Mu et al., 2018).

Jumping into the world of spatial statistics, it is important to understand the type of data to be analyzed. *Spatial data* refers to data with observations attached to 2- or 3-dimensional spatial objects (i.e. points, lines, or polygons) (Mamoulis, 2012). This paper will focus on data with points, or coordinate pairs, as spatial objects. Coordinate pairs in 3-dimensional (3D) space are based on the geometry of the object that is or contains the area of interest. The most prominent 3D coordinate system is geographic coordinate system, which identifies locations on Earth using angular measures of longitude and latitude. Meanwhile, coordinate pairs in 2-dimensional (2D) space are defined by a grid with equidistant intervals that overlays the area of interest. 3D coordinate systems can be projected, or transformed, to a 2D coordinate system, like maps of the world being presented in 2D instead of on a globe. This paper focuses specifically on what will be called *spatially varying data*, which is data comprised of observations attached to coordinate pairs with at least one predictor whose influence on the response variable varies depending on location in space.

The foundation of multivariate statistical regression models is the *multiple linear regression* (MLR) model. The general model with m predictors $x_1, x_2, ..., x_m$ and no interaction terms is

$$y = \beta_0 + \sum_{j=1}^{m} \beta_j x_j + \epsilon, \tag{1.1}$$

where y denotes the response variable, β_0 is the intercept, m is the number of predictors, β_j denotes the coefficient for the j^{th} predictor x_j, and ϵ denotes the error term in the model. Additionally, ϵ is assumed to follow $N(0, \sigma^2)$ and be identically and independently distributed (iid) (Guo, 2021). A significant benefit of the MLR model is how straight-forward the interpretation of each coefficient is. The coefficients β_j for each predictor x_j can be interpreted as the marginal effect of the corresponding predictor when all other predictors are held constant. For this model, every 1-unit increase of predictor x_j results in a change of β_j in the predicted y value. While there are benefits to MLR models, they can fail to provide a proper fit when the marginal effect of a predictor varies for different values of the predictor, leading to the need for nonparametric regression models.

1.1.1 Nonparametric Nonspatial Regression Models

One of the first significant developments of nonparametric models can be found in the 1987 paper "Generalized Additive Models: Some Applications" by Hastie and Tibshirani. The two proposed an extension of the additive model called the *generalized additive model* (GAM). This model has the form

$$y = \beta_0 + \sum_{j=1}^{m} f_j(x_j) + \epsilon. \tag{1.2}$$

For each of the m predictors, $f_j(x_j)$ denotes a function of the predictor x_j. The m functions, the intercept β_0, and the error term ϵ are then added together to predict the response variable y. (Hastie and Tibshirani, 1987)

The form of this model is directly imitated by the estimation of the B-spline model. In 1947, Curry and Schoenberg were the first to define *B-splines*, which is the shortened term for basis splines. By taking the structure of equation 1.2, defining $f_j(x_j)$ to be the cubic B-spline function for predictor j, and removing β_0, we obtain the model for B-spline regression that was developed in 1978 by de Boor. This model standardizes each predictor to range from 0 to 1 and partitions the values of the standardized predictor into $N+1$ segments using some sequence $\{0 = t_0 < t_1 < \cdots < t_N < t_{N+1} = 1\}$ (De Boor, 2001).

The amount of literature on nonparametric estimation continued to grow in an attempt to avoid the problem of model misspecification. In 1964, the first kernel regression function estimator was defined. The Nadaraya-Watson kernel regression function estimator was developed separately by two statisticians, for which the estimator was named. This estimator transforms multivariate regressions into univariate regressions (Bierens, 1988). This model produces the conditional expected value of the response variable given the values of the dependent and independent variables in the vicinity of the evaluation point. Values of the independent variables that are closer to the evaluation point have a greater influence on the conditional expected value at that evaluation point.

1.1.2 Nonparametric Spatial Regression Models

While the previously described models are effective with non-spatially varying data, statisticians continued to develop models that are built specifically to model phenomena over space, such as gas prices, pollution concentrations, and humidity levels. In 1993, Chen and Tsay proposed *functional-coefficient autoregressive* (FAR) models in a paper of the same name. A generalization of the model is

$$y = \beta_0(t) + \sum_{j=1}^{m} \beta_j(t) x_j + \epsilon, \tag{1.3}$$

where $\beta_j(t)$ is a functional coefficient for predictor x_j that is dependent on time point t, $\beta_0(t)$ is the functional intercept dependent on t, and ϵ is the error term of the model, which is iid with a mean of 0 and variance of σ^2. ϵ is also defined to be independent of time points prior to the time point of evaluation (Chen and Tsay, 1993).

Five years after that paper's publication, Brunsdon, Fotheringham, and Charlton proposed the *geographically weighted regression* (GWR) model. This model is similar to the FAR model, but GWR utilizes spatial coordinates to determine the predictors' functional coefficients instead of points in time (Brunsdon et al., 1998).

While the spline tensor product has the oldest theoretical base compared to the other models of interest, its application in statistical modeling is one of the more recent developments. The Kronecker product of matrices was officially defined in 1858 by Johann Georg Zehfuss, but it was named for Leopold Kronecker, who used this product in the 1880s (Liesen and Mehrmann, 2015). This product is also referred to as the *tensor product*, which is how it will be referenced in this paper. It was not until 1994 that the tensor products of polynomial splines started being used to fit regression models. Stone was one of the first to bring tensor products into regression modeling for multivariate function estimation (Stone, 1994). The tensor product of splines regression model can be written as

$$y = F(u, v) = f_0(u, v) + \sum_{j=1}^{m} f_j(u, v) x_j + \epsilon, \tag{1.4}$$

where $f_0(u, v)$ and $f_j(u_v)$ are the tensor product of splines fitted for the intercept and predictor x_j, respectively. Both functions are dependent on coordinate pairs (u, v). In place of standardizing predictors, as seen for the B-spline regression model, the components u and v of the coordinate pairs $s = (u, v)$ are standardized for the tensor product of B-splines regression model. The development of new statistical applications for tensor products is ongoing, including in the realm of spatial analysis.

1.2 Data Description

In section 4, the notable Boston housing price spatial data set will be used to analyze the methodologies compared in this paper. This data set was built from the 1970 Boston census data by Harrison and Rubinfeld in 1979 and was corrected and updated to include spatial point data by Gilley and Pace in 1996. Each row in the data set represents one census tract, which is a county subdivision defined by the U.S. Census Bureau (United States Census Bureau, 2022). A census tract contains between 1,200 to 8,000 people, and the data set contains data for 506 census tracts (Newman et al., 1998). The variables in Table 1.1 were chosen from the Boston data set to mirror the analysis conducted in the paper "A semiparametric spatial dynamic model" (Sun et al., 2014).

Variable Name	Variable Description
CMEDV	Corrected median value of owner-occupied homes in USD 1000's
CRIM	Per capita crime rate by town
RM	Average number of rooms per dwelling
RAD	Index of accessibility to radial highways
TAX	Full-value property-tax rate per USD 10,000
LSTAT	Percentage of lower status of the population
LON	Longitude of census tract
LAT	Latitude of census tract

Table 1.1: `BostonHousing2` data variables from `mlbench` package in R

1.3 Objective

The aim of this paper is to evaluate the fits of MLR, cubic B-splines, univariate kernel smoothing, GWR, and tensor product of cubic B-splines models over spatially varying data, illustrating how the development of nonparametric spatial regression models compare to nonspatial models.

This report is organized as follows. Chapter 2 outlines the specific practices and methods that will be used in chapters 3 and 4. In chapter 3, a simulation study is used to analyze how coefficient estimations of each model of interest differ from the true coefficients. A case study based on 1970 Boston house pricing data is conducted in chapter 4, assessing the differences between model response variable estimates and the observed values. Finally, chapter 5 discusses the findings from the prior chapters.

Chapter 2

Methodology

In this chapter, we will explore the five models of interest in more depth. For simplicity in comparing the models of interest, only the main effects of each predictor are investigated, not taking interaction effects into account.

2.1 Models of Interest

2.1.1 Multiple Linear Regression

The multiple linear regression (MLR) model is fundamental in statistical modeling. The estimated MLR model for m predictors is

$$\hat{y} = \hat{\beta}_0 + \sum_{j=1}^{m} \hat{\beta}_j x_j. \tag{2.1}$$

The coefficients $\beta_0, \beta_1, ..., \beta_m$ in model 1.1 are estimated using the ordinary least squares (OLS) method, meaning that the resulting estimated MLR model minimizes the following equivalent equations.

$$\sum_{i=1}^{n}[y_i - (\hat{\beta}_0 + \sum_{j=1}^{m} \hat{\beta}_j x_{i,j})]^2 \equiv \sum_{i=1}^{n}[y_i - \hat{y}_i]^2 \tag{2.2}$$

$\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_m$ denote estimated coefficients, and i indicates that the x_j and y values are from the i^{th} observation in a sample of size n (Guo, 2021). The R function `lm()` from the package `stats` is used to fit MLR models in this paper.

2.1.2 B-Spline

As mentioned in chapter 1, B-spline regression models are GAMs (as defined in equation 1.2). B-splines are a family of candidate models that use basis functions, for which the method is named after, to estimate the functions $f_j(x_j)$ in GAMs. For a B-spline, the x-axis is standardized from 0 to 1 and partitioned into $N+1$ sections by N points called *knots*. The basis function of a B-spline with a degree of p for the set of knots $K = \{0 = t_0 < t_1 < ... < t_{N+1} = 1\}$ and the standardized predictor fit by the B-spline c_j $(j = 1, ..., m)$ is denoted by $B_{j,p,q}(c_j)$, where $q = 0, 1, 2, ..., N$ indicates t_q to be the knot interval's lower bound of the function. The knots $\{t_1, ..., t_N\}$ are referred to as *internal knots*, while $t_0 = 0$ and $t_{N+1} = 1$ are referred to as *boundary knots*. In this paper, we use a uniformly spaced knot sequence with 5 internal knots for each predictor. Thus, we can further define any knot (t_q) in K to be $q/(N+1) = q/6$. When constructing a B-spline with one predictor, the estimated regression model is

$$\hat{f}(x) = \sum_{q=0}^{N} \hat{\alpha}_q B_q(x). \tag{2.3}$$

Since our simulated and real data sets in the subsequent chapters include multiple independent variables, we used GAMs to add together the B-splines for each independent variable, which can be represented by

$$\hat{y} = \sum_{j=1}^{m} \hat{f}_j(x_j) = \sum_{j=1}^{m} \sum_{q=0}^{N_j} \hat{\alpha}_{j,q} B_{j,q}(x_j), \tag{2.4}$$

where N_j denotes the number of internal knots for the B-spline basis of predictor x_j.

To fit a B-Spline to the values of an independent variable x_j that does not have 0 as its minimum and 1 as its maximum, the collected data (corresponding to the set of indices $A = \{1, 2, ..., n\}$ for all n observations) must be standardized. Let $c_{i,j}$ denote the standardized value for predictor j of observation i.

$$c_{i,j} = \frac{x_{i,j} - \min_{i \in A}(x_{i,j})}{\max_{i \in A}(x_{i,j}) - \min_{i \in A}(x_{i,j})} \tag{2.5}$$

Once the data for each independent variable is standardized, the basis function evaluated at $c_{i,j}$ is recursively found by using the following process. First, the basis function for predictor x_j at degree $p = 0$, $B_{j,0,q}(c_{i,j})$, must be determined for every $c_{i,j}$ in every knot interval (t_q, t_{q+1}). Note that the knot sequence K is the same for all predictors now that they all have been standardized and the same number of knots are used for each predictor's B-spline.

$$B_{j,0,q}(c_{i,j}) = \begin{cases} 1 & \text{if } c_{i,j} \in (t_q, t_{q+1}) \\ 0 & \text{otherwise} \end{cases}$$

For every subsequent unit increase in degree p of $c_{i,j}$ in the basis function, the next equation is recursively calculated until the desired degree for the B-spline function is met.

$$B_{j,p,q}(c_{i,j}) = \frac{c_{i,j} - t_q}{t_{q+1} - t_q} B_{j,p-1,q}(c_{i,j}) + \frac{t_{q+2} - c_{i,j}}{t_{q+2} - t_{q+1}} B_{j,p-1,q+1}(c_{i,j})$$

In this paper, cubic B-Splines are used, making the desired degree to be 3. Thus, this step would be repeated 3 times to obtain the cubic B-spline basis for one independent variable. Figure 2.1 is a visualization of the cubic B-spline basis for any one of the predictors.

After the creation of this basis, the final steps would be to plug the values found for $B_{i,j,3,q}(c_{i,j})$ into Equation 2.4, fit a GAM to estimate $\alpha_{j,q}$ for every combination of predictor and knot, and unstandardize the resulting model. Using R programming standardization and the B-spline basis calculations are completed automatically by the `bs()` function. The `gam` package accepts for B-spline components in its fitting.

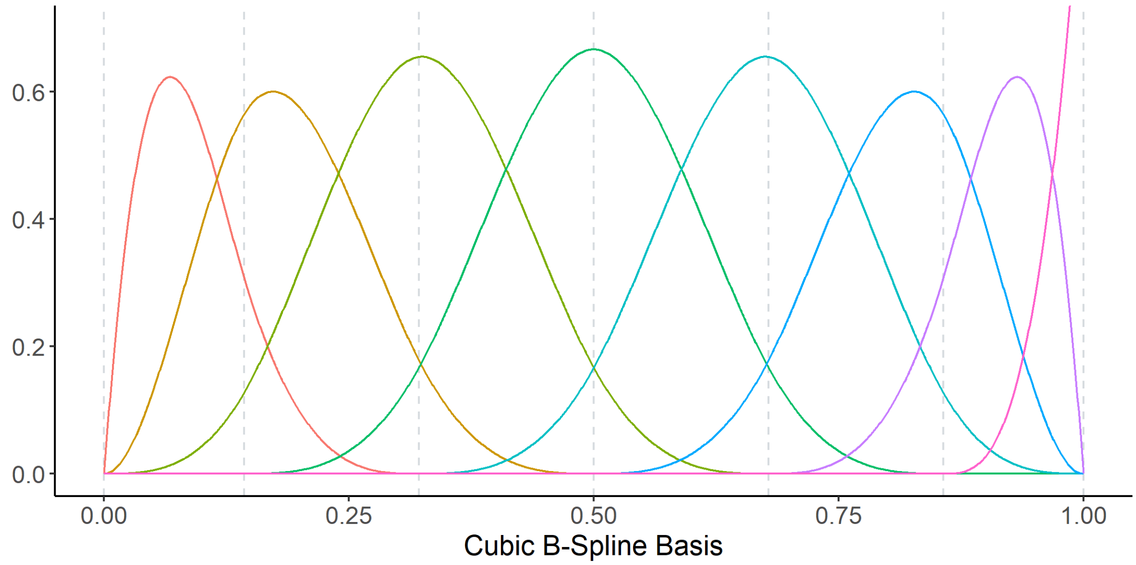


Figure 2.1: The cubic B-spline basis function with 5 internal knots. The knots are marked by gray dashed lines. Each color represents the function for a distinct interval between knots.

2.1.3 Univariate Kernel Smoothing

When performing univariate kernel smoothing regression estimation, the researcher must select the regression estimator prior to fitting. For this paper, the Nadaraya-Watson Regression estimator was chosen. This estimator is calculated using the equation

$$\hat{y} = \hat{f}_{\mathbf{H}}(\mathbf{w}) = \sum_{i=1}^{n} \frac{K_{\mathbf{H}}(\mathbf{w} - \mathbf{W}_i) y_i}{\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{w} - \mathbf{W}_i)}, \tag{2.6}$$

where $\mathbf{w}$ denotes the vector of the m predictors ($[w_1, ..., w_m]^\top$), $\mathbf{W}$ denotes an $m \times n$ matrix with each column i corresponding to the i^{th} observation and each row j corresponding to predictor j. Thus,

$$\mathbf{W} = \begin{pmatrix} W_{1,1} & W_{1,2} & \cdots & W_{1,n} \\ W_{2,1} & W_{2,2} & \cdots & W_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ W_{m,1} & W_{m,2} & \cdots & W_{m,n} \end{pmatrix}.$$

Additionally, y_i denotes the observed response for observation i ($i = 1, 2, ..., n$), and $\mathbf{H}$ denotes a vector of bandwidths ($[h_1, h_2, ..., h_m]^\top$) which each correspond to a predictor. The kernel for each observation can be written as

$$K_{\mathbf{H}}(\mathbf{w} - \mathbf{W}_i) = K_{h_1}(w_1 - W_{i,1}) \times K_{h_2}(w_2 - W_{i,2}) \times \cdots \times K_{h_m}(w_m - W_{i,m}).$$

In this paper, a Gaussian kernel smoothing function was used. Thus, for the predictor w_j

$$K_{h_j}(w_j - W_{i,j}) = \frac{1}{h_j} \cdot \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2} \cdot \left(\frac{w_j - W_{i,j}}{h_j} \right)^2 \right\}$$

The bandwidths were calculated using the `npregbw` function in the R package `np`, which selected bandwidths based on a least-squares cross-validation procedure, and the bandwidths are fixed, meaning that they do not vary for different predictor values (Racine and Hayfield, 2008).

2.1.4 Geographically Weighted Regression

The GWR model for a sample of size n and m predictors can be written as

$$\hat{y} = \hat{\beta}_0(s) + \sum_{j=1}^{m} \hat{\beta}_j(s) x_j, \tag{2.7}$$

where s denotes the location coordinate pair, $\hat{\beta}_0(s)$ is the estimated intercept functional coefficient, and $\hat{\beta}_j(s)$ denotes the estimated functional coefficient for predictor j. The coefficients for some location k are estimated using the following matrix equation.

$$\hat{\boldsymbol{\beta}}(s_k) = (\boldsymbol{X}^\top \boldsymbol{W}_k \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}_k \boldsymbol{Y}$$

$\boldsymbol{X}$ is the $n \times m$ design matrix that contains the values of the independent variables for each observation, $\boldsymbol{Y}$ is the vector of length n that contains the values of the response variable for each observation, and $\boldsymbol{W}_k$ is the $n \times n$ weighting matrix. $\boldsymbol{W}_k$ contains the weights associated with each observation when estimating values of the functional coefficients $\hat{\boldsymbol{\beta}}(s_k)$ at some location k.

$$\boldsymbol{W}_k = \begin{pmatrix} \alpha_{1,k} & 0 & \cdots & 0 \\ 0 & \alpha_{2,k} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_{n,k} \end{pmatrix}$$

For this paper, we used a Gaussian kernel distance-decay-based weighting function and defined the fixed bandwidth h ($h > 0$) for the regression model using the `gwr.sel` function of the R package `spgwr`. Let $d_{i,k}$ denote the distance between locations i and k. Then,

$$\alpha_{i,k} = \exp(-d_{i,k}^2 / 2h^2)$$

As can be seen in Figure 2.2, the weight of location i on the estimate of the response variable at location k lessens when $d_{i,k}$ increases, and larger bandwidths result in broader distributions of weight.

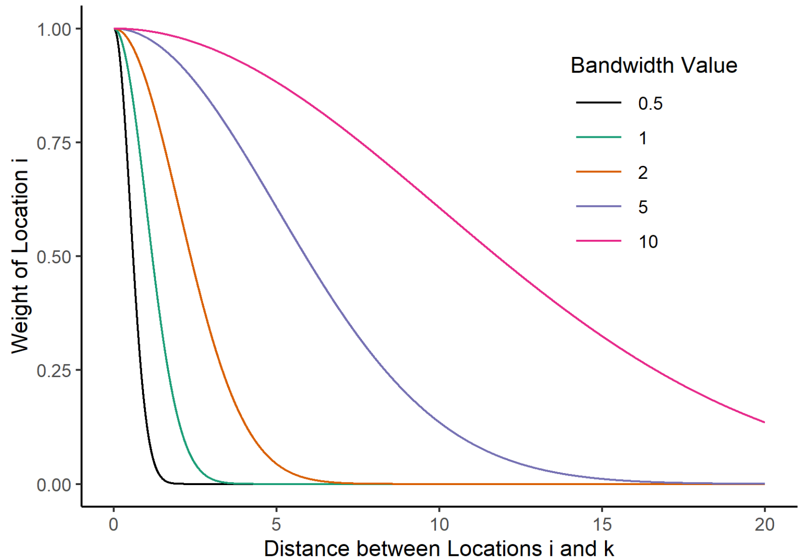


Figure 2.2: The effect of bandwidth in the Gaussian kernel distance-decay-based weighted function

2.1.5 Tensor Product of B-Splines

For the standardized coordinates (u, v), the estimated tensor product of B-splines model is

$$\hat{f}(u, v) = \sum_{k=0}^{N_u} \sum_{\ell=0}^{N_v} \hat{\alpha}_{k,\ell} \left[g_k(u) \otimes h_\ell(v) \right]. \tag{2.8}$$

N_u and N_v are the number of internal knots that span u and v, respectively. Figure 2.3 provides a visual example of this process. In this example, $N_u = 5$ and $N_v = 4$. Like in the model for B-splines, there is a coefficient $\alpha_{k,\ell}$ for every intersection of u and v. The function $g_k(u)$ denotes the B-spline basis function calculated for the knot interval (t_k, t_{k+1}) of u, and similarly, $h_\ell(v)$ denotes the B-spline basis function for the knot interval $(r_\ell, r_{\ell+1})$ of v. These bases are found using the process described in section 2.1.2. The tensor product operation (denoted by $\otimes$) for a data set of size n is performed for $g_k(u)$ and $h_\ell(v)$. This product produces a $n \times N_u N_v$ matrix, in which every value found from $g_k(u)$ for observation i is multiplied by every value found from $h_\ell(v)$ for the same observation i. Once the resulting surface, also called the tensor product B-spline basis, is created, the coefficients $\alpha_{k,\ell}$ are fit for $g_k(u) \otimes h_\ell(v)$ in each of u and v's knot interval intersections using OLS method described

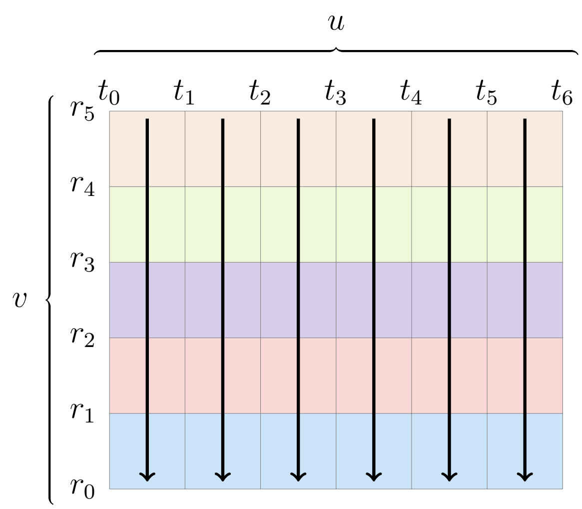


Figure 2.3: A depiction of the tensor product process

in equation 2.2. To match the notation for this model, the OLS criterion is

$$\min \sum_{i=1}^{n} [y_i - (\hat{F}(u_i, v_i))], \tag{2.9}$$

where i indicates the i^{th} observation from a sample of size n.

We can fit Model 2.7 by using this spline approach and obtain the estimators $\hat{\beta}_j(\cdot)$ in this section for $j = 1, ...m$ by redefining $\hat{\beta}_j(s)$ to be $\hat{f}_j(u, v)$ using the notation found in this section.

$$\hat{y} = \hat{F}(u, v) = \hat{f}_0(u, v) + \sum_{j=1}^{m} \hat{f}_j(u, v) x_j \tag{2.10}$$

With larger data sets, this process becomes tedious and time-consuming when performed manually. Thus, R programming is used in Chapters 3 and 4 to complete the process more efficiently. The Generalized Linear Model (`glm`) function in R was used to fit the tensor product basis surface to the standardized coordinates, collecting the estimates of $\alpha_{k,\ell}$. The error distribution for this generalized linear model is specified to be Gaussian.

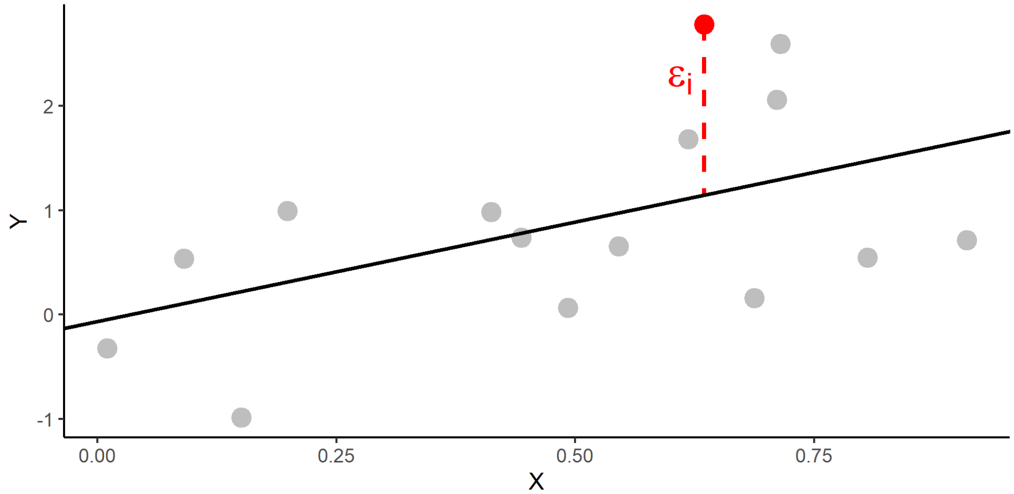


Figure 2.4: Illustration of ϵ_i for calculating MSE

2.2 Measurements of Error

In this paper, three measures of error are used to evaluate the accuracy of estimated values $\hat{y}$, predicted values $\tilde{y}$, and the functional coefficient estimates $\hat{\beta}_j(\cdot)$.

2.2.1 Mean Squared Error

The first measure of error to discuss is *mean squared error* (MSE) of y, which assesses how accurate the estimated values are compared to the observed values of y. For a sample of size n, MSE of y is

$$MSE(y) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2, \tag{2.11}$$

where ϵ_i denotes the difference between the observed values y and the estimated values $\hat{y}$, as depicted in figure 2.4 (Harville and Jeske, 1992). Note that when the OLS method is used to estimate a regression model, the $MSE(y)$ calculated for the resulting regression model is the smallest $MSE(y)$ for the given set of predictors. This can be confirmed by comparing equation 2.2 to equation 2.11.

2.2.2 Mean Squared Prediction Error

Similar to MSE, *mean squared prediction error* (MSPE), also called mean squared error of prediction, is the average of the squared differences between observed and predicted values. However, the two errors differ in their usage. MSPE can be used in partial linear regression to evaluate the fit of predictive models and in the cross-validation of modeling methods (Denham, 2000). This paper focuses on the latter when discussing MSPE and uses it to perform 10-fold cross-validation. For this, a sample is partitioned into 10 equal or relatively equal subsamples, called *folds*. Then, a model is fit to all observations aside from those in the first fold. This process is repeated 10 times, with each repetition excluding a different fold. For each repetition, response values are predicted for every observation, including those from the excluded fold. The following is calculated for each excluded fold k.

$$P_k = \sum_{i=1}^{n} (y_i - \tilde{y}_{i,k})^2 \tag{2.12}$$

$\tilde{y}_{i,k}$ denotes the predicted value of y_i from the model trained by all of the sample data excluding the k^{th} fold. Then, the MSPE can be calculated by

$$MSPE = \frac{1}{n} \sum_{k=1}^{10} P_k. \tag{2.13}$$

The goal of this cross-validation is to minimize the MSPE, creating a better predictive model for the population and avoiding overfitting of the training data.

2.2.3 Mean Integrated Squared Error

Unlike MSE and MSPE, *mean integrated squared error* (MISE) evaluates the fit of an estimated function to a true, known function. The accuracy of functional coefficient estimates in this paper is measured using this metric. For predictor x_j, the accuracy of the functional coefficient is evaluated using

$$MISE_j = E\left\{\int_{-\infty}^{\infty}\left[\hat{\beta}_j(z) - \beta_j(z)\right]^2 dz\right\}, \tag{2.14}$$

which can be approximated by

$$\frac{1}{N}\sum_{i=1}^{N}\left[\hat{\beta}_j(z_i) - \beta_j(z_i)\right]^2. \tag{2.15}$$

In both equations, $\hat{\beta}_j(z)$ and $\beta_j(z)$ are the estimated and true functional coefficients for x_j, respectively (Ghosh, 2018). Both are dependent on some other variable z. For this paper, z is either x_j or the position of the observation in space represented by s depending on the model.

Chapter 3

Simulation Analysis

In this chapter, simulation data will be used to analyze and compare the coefficient estimations and accuracy of the models of interest in this paper. We compute the MISEs of the functional coefficients' estimators from the MLR, GWR, and tensor product models for each sample and the MSEs and MSPEs of the predicted response variable ($\hat{y}$) for each model and sample.

3.1 Simulation Data Creation

To start, we set the true simulated population function to be

$$y_i = \beta_0(s_i) + \beta_1(s_i)x_{1i} + \beta_2(s_i)x_{2i} + \epsilon_i, \tag{3.1}$$

where the coefficients take the following values or functions.

$$\beta_0(u) = 2\sin\left(\frac{\pi u}{6}\right), \quad \beta_1(s) = \sin(2\|s\|^2), \quad \beta_2(s) = \cos(3\|s\|^2)$$

$\beta_1(s)$ and $\beta_2(s)$ mirror the functional coefficients in "A semiparametric spatial dynamic model" (Sun et al., 2014) and $\beta_0(u)$ follows the same form as the functional intercept coefficient in "Estimation and inference in spatially varying coefficient models" (Mu et al., 2018).

For the population of size 10,000, we independently generated x_1 and x_2 from $N(\mathbf{0}_2, I_2)$,

s from $U[0,1]^2$ (where $s = (u, v)$), ϵ is iid from $N(0,1)$, and y is generated through model 3.1. From the simulated population, 100 samples of size $n = 500$ were drawn randomly with replacement. We let the $x_{i,j,k}$ indicate the i^{th} value for the j^{th} predictor in the k^{th} sample. We fit our models of interest to each of the simulated samples and examined the accuracy of each model's coefficient estimates and overall fit.

3.2 Evaluation of Coefficient Estimates

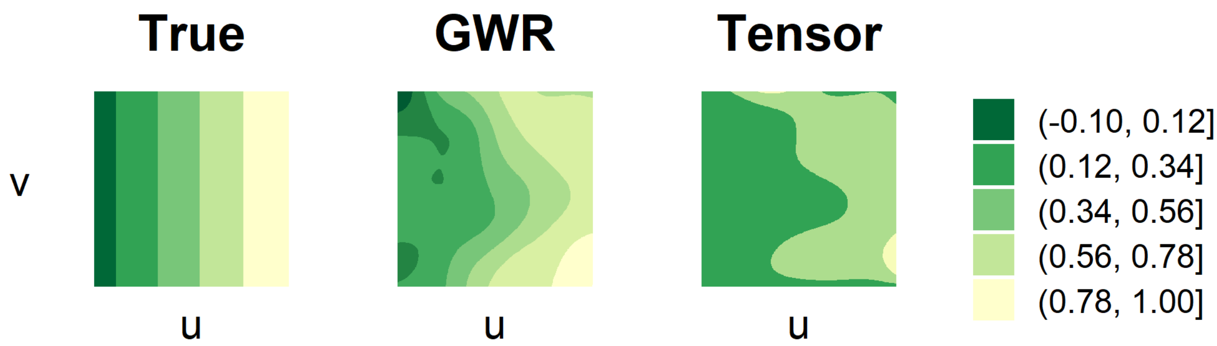
First, we will analyze coefficient estimation for the MLR, GWR, and tensor product models. The MISEs of the B-spline and univariate kernel smoothing models are not calculated, because they do not provide estimates of $\beta_{j,k}(s)x_j$ for each predictor j. The B-spline model produces $\hat{f}_{j,k}(x_j)$ (as defined in equation 2.3) in place of $\beta_{j,k}(s)x_j$, and the kernel smoothing model transforms the independent variables into a single variable. The obtained average MISE values are presented in table 3.1. Table 3.1 shows that the GWR model produces smaller

Table 3.1: MISEs across 100 samples of size 500 from the simulated population

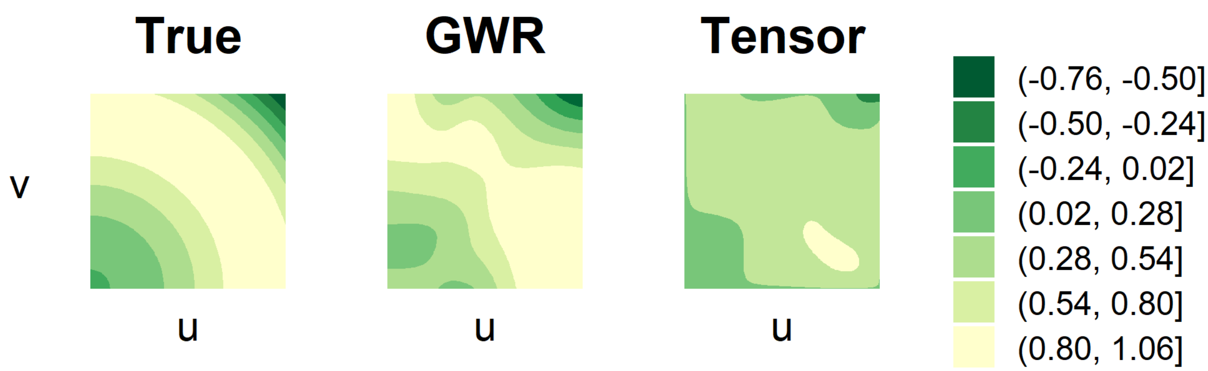
	$\beta_0(s)$	$\beta_1(s)$	$\beta_2(s)$
MLR	0.08959	0.11788	0.50780
GWR	0.02058	0.02748	0.04016
Tensor	0.24959	0.32720	1.05635

average MISEs for its estimates of $\beta_0(s)$, $\beta_1(s)$, $\beta_2(s)$ compared to the MLR and tensor product models. The average MISEs produced by the GWR model for $\beta_0(s)$ and $\beta_1(s)$ are approximately 4 times smaller than the MISEs produced by the MLR model for the same functional coefficients, while the GWR model's average MISE for $\beta_2(s)$ is about 12 times smaller than the MLR model's. The tensor product model, on the other hand, produced the largest average MISEs for each functional coefficient estimate.

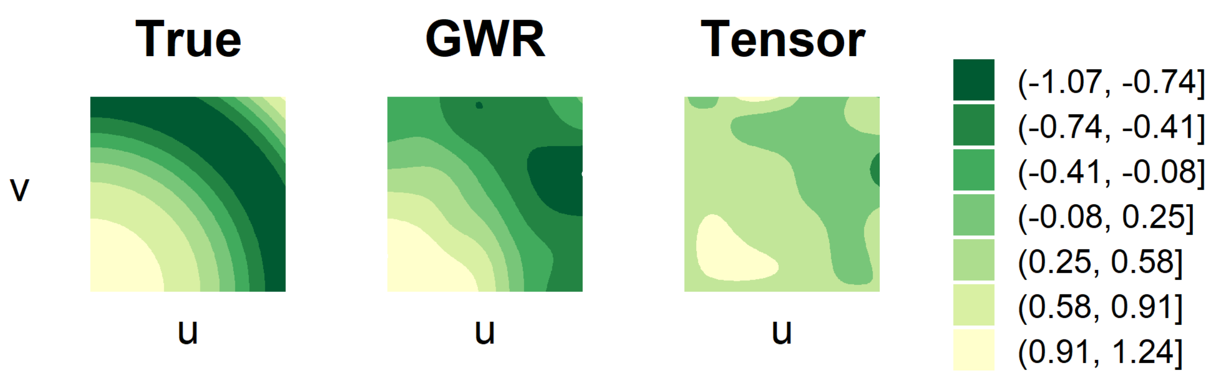
Contour plots for the true values of the functional coefficients spanning $[0,1]^2$ are provided in the left column of Figure 3.1. The middle column of Figure 3.1 contains the estimated values of the function coefficients from the GWR model, and the right column contains the



(a) $\beta_0(s)$



(b) $\beta_1(s)$



(c) $\beta_2(s)$

Figure 3.1: Contour plots of the true and estimated values of $\beta_0(s)$, $\beta_1(s)$ and $\beta_2(s)$ for the simulated population. The estimated values were produced by the GWR and tensor product of B-splines models fitted by sample 28. The legends on the far right are applicable to the entire row that they are in.

estimates from the tensor product of B-splines model. Of the 100 samples, the 28^{th} sample was chosen at random. The estimated values of the coefficients for this sample were used to create the contour plots in the right 2 columns. Contour plots for the MLR model are not displayed, as the coefficient estimates do not vary across locations.

The contour plots in Figure 3.1 show that the GWR model was able to approximately estimate the functions of the true functional coefficients. The tensor product model appears to poorly estimate the functions of the true functional coefficients. The visible patterns of the true functional coefficient values in the first column of the contour plots can be detected in the contour plots of the estimated functional coefficient values for the GWR model, while the patterns shown in the contour plots for the tensor product model appear to be loosely related on the true functional coefficient values. Compared to the GWR model, the tensor product model appears to level out its estimates more, creating a less hilly surface.

Overall, the average MISEs in table 3.1 and the contour plots in Figure 3.1 indicate that the coefficient estimates produced by the GWR model are closer to the values produced by the true functional coefficients compared to the MLR and tensor product models with the tensor product model producing larger average MISEs than the MLR model.

3.3 Evaluation of Model Accuracy

For each combination of model and sample, we calculated the MSE to evaluate how well the model fits the sample data, and the MSPE was calculated to evaluate how well the model predicts the response variable for observations outside of the sample.

	MLR	B-Spline	Kernel	GWR	Tensor
MSE	1.71231	1.63024	1.60583	0.90656	0.65620
MSPE	1.75600	203,000	1.79728	1.10745	1.89412

Table 3.2: MSEs and MSPEs across samples of the simulated population

Table 3.2 indicates that the B-spline tensor product model has the lowest MSE compared to the other models of interest, shortly followed by the GWR model. The nonparametric nonspatial models have the third and fourth smallest MSEs with the univariate kernel smoothing model as third and the B-spline model as fourth. The model that produced the largest MSE was the MLR model.

Additionally, table 3.2 shows how well the models of interest fit the population data that was not included in the sample through the MSPEs. The MLR, kernel smoothing, and GWR models have MSPE values that are larger than their MSE values, but not by much. These 3 models have the smallest MSPE values, as well. The smallest MSPE is from the GWR model, the second smallest is from the MLR model, and the third smallest is from the kernel smoothing model. While the B-spline model fits the sample data well, it was far from accurate when predicting the response variable for the rest of the population. The MSPE values for this model ranged from 1.79 to 9.66×10^6, making the accuracy of prediction heavily dependent on the sample data used to fit the B-spline model. The relationship between MSE and MSPE for the tensor product model is similar, yet less extreme, in comparison to the B-spline model. This model produced the smallest MSE value compared to all of the models of interest, but its MSPE is the second largest. For both the B-spline and tensor product models, the models appear to fit the sample data closely, making it less accurate to generalize the model for the entire population.

Thus, the MSEs indicate that nonparametric spatial models had greater accuracy in estimating a response variable for data with marginal effects that vary over space compared to nonparametric nonspatial models, which had greater accuracy compared to a parametric nonspatial model. Furthermore, the MSPEs indicate that the MLR, kernel smoothing, and GWR models can predict the response variable for a population at a decent level of accuracy, while the B-spline and tensor product models tend to overfit the sample data.

Chapter 4

Analysis of Boston Housing Data

In this chapter, we are going to apply the models of interest to analyze the Boston house price data set. We are going to explore how the corrected median value of owner-occupied homes in \$1,000's (denoted by CMEDV) is affected by the per capita crime rate by town (denoted by CRIM), the average number of rooms per dwelling (denoted by RM), the full-value property-tax rate per \$10,000 (denoted by TAX), and the percentage of lower status of the population (denoted by LSTAT). Since we are not attempting to identify which predictors have effects that vary over space, all predictors will be treated like their effects do vary over space. To be similar to the notation used in chapter 2, each variable will be expressed by the notation in table 4.1.

Table 4.1: Notation used for variables in Boston housing price data set

Variable	CMEDV	CRIM	RM	RAD	TAX	LSTAT
Notation	y_i	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	$x_{i,5}$

Prior to model fittings, the rows of the original Boston housing price data were randomized. The rows were then partitioned into 10 folds, preparing the data for 10-fold cross-validation. 9 folds were of size 50, and 1 was of size 56, as there were 506 total observations. For each fold, all of the models of interest were fit using all observations except the ones in the fold, which will be referred to as the training data set. Those models were then used to predict

CMEDV for all observations in the fold. The mean squared prediction error (MSPE) was calculated for each combination of model and fold. The mean squared error (MSE) was calculated for each model for the entire set of data. The resulting MSPEs and MSEs are displayed in table 4.3.

The MSPEs for the models of interest show that the kernel smoothing model was the most accurate at estimating CMEDV for observations outside of the training set with the lowest MSPE. The GWR models produced a slightly larger MSPE than the kernel smoothing model. The MLR model yields an MSPE that is greater than the two aforementioned models, but the MSPEs for the cubic B-splines regression model and the tensor product of cubic B-splines regression model are notably large compared to the other models of interest. For the cubic B-splines regression model, the MSPEs of folds 2 and 4 are significantly higher than any MSPEs produced by this model. For both folds, there is 1 observation in the fold data set with a CRIM ($x_{i,1}$) value greater than 65. As is shown in table 4.2, the values of CRIM are heavily skewed to the right. With the B-spline knots being evenly spread across

Min	1^{st} Quartile	Median	3^{rd} Quartile	Max
0.00632	0.08199	0.25651	3.67822	88.97620

Table 4.2: Five number summary for CRIM ($x_{i,1}$) in Boston housing price data set

the range of CRIM, the final 2 knot intervals for CRIM [(59.35, 74.21) and (74.21, 89.08)] contain 2 and 1 observations, respectively. With so little data in these knot intervals, any observations removed from the training data that fall in one of these intervals make the model unable to properly fit B-splines to the last 2 knot intervals. Each of the data sets for fold 2 and fold 4 contain 1 of these observations, leaving little to no data in the corresponding knot interval of the training data set, which resulted in higher MSPEs than the other folds. Similarly, the tensor product of cubic B-splines model with 5 internal knots has intersections of the longitude and latitude knot intervals that contain few data points. When 1 or more observations located within a scarcely dense knot interval intersection are taken from the training data set and put into the fold data set, the model attempts estimation with little information on the area when fitting the model. This results in significant inaccuracies in predictions of the response variable outside of the training data set, leading to large MSPE values, as shown in table 4.3. The MSE values in table 4.3 indicate that the kernel smoothing

	Linear	B-Spline	Kernel	GWR	Tensor
MSPE	30.0000	2,741.96	14.7889	16.5945	2,044,877
MSE	28.702	14.556	1.8174	11.578	203.88

Table 4.3: 10-fold cross-validation for Boston housing price data set using MSE and MSPE for each model of interest

model provided the most accurate CMEDV estimates for the Boston housing price data set. The GWR model produced the second smallest MSE, which was closely followed by the B-spline regression model. The MSE for the MLR model was double the size of its predecessor, the GWR model. The model with the highest MSE, indicating the worst model fit of the models of interest, was the tensor product of B-splines model.

Looking at the MSE and the MSPE values, the tensor product of cubic B-splines regression model yielded the least accurate CMEDV estimates for the Boston housing price data set and the least accurate predictions of CMEDV for the fold data sets. The most accurate CMEDV estimates for the Boston housing price data and for the predictions of the fold data sets were produced by the univariate kernel smoothing model.

Chapter 5

Conclusion

In this study, we examined the model accuracy and coefficient estimations of the multiple linear regression, B-spline regression, univariate kernel smoothing regression, geographically weighted regression, and tensor product of B-splines regression models. The results of the simulation study in chapter 3 show that for a spatially dependent data set where all coefficients are known to vary by location, the nonparametric spatial models (i.e. the GWR and tensor product of B-splines models) were able to better estimate the values of the response variable for a given training data set compared to the nonspatial models, as they produced the smallest MSEs. However, when predicting values of the response variable for the population given a training data set, the B-spline-based models (i.e. the B-spline and tensor product of B-splines models) tend to overfit the data, resulting in less accurate predictions on the whole population. For these two models, decreasing the number of internal knots or increasing the size of the training data set would likely produce smaller MSPEs. This study could be extended by exploring the effects of performing the previously mentioned alterations with the models of interest presented here. The real data analysis conducted with the Boston housing price data set in chapter 4 indicates that, of the models of interest, the univariate kernel smoothing model provided the best fit for the overall Boston data set and produced the most accurate predictions in our 10-fold cross-validation procedure. Since the kernel smoothing model is a nonspatial model, it having the best performance implies that at least one coefficient does not vary by location. Thus, in future studies, a model selection procedure should be performed to more accurately capture the relationship between the predictors and response variable. Additionally, interaction terms should be investigated, as real-world phenomena have factors that interact, and this should be investigated spatially. Overall, this study shows that nonparametric spatial models are ideal for spatially varying data sets that are known to have all coefficients vary over space, but the investigation of real-world data sets through model selection prior to model fitting will lead to more accurate models for data sets where not all predictors have spatially dependent coefficients.

Bibliography

H. J. Bierens. The Nadaraya-Watson Kernel Regression Function Estimator. *Serie Research Memoranda*, pages 212–247, 1988. doi: 10.1017/CBO9780511599279.011.

C. Brunsdon, S. Fotheringham, and M. Charlton. Geographically weighted regression. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(3):431–443, 1998.

R. Chen and R. S. Tsay. Functional-coefficient autoregressive models. *Journal of the American Statistical Association*, 88(421):298–308, 1993.

C. De Boor. *A practical guide to splines: with 32 figures.* Springer, 2001.

M. C. Denham. Choosing the number of factors in partial least squares regression: estimating and minimizing the mean squared error of prediction. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 14(4):351–361, 2000.

S. Ghosh. *Kernel smoothing: Principles, methods and applications.* John Wiley & Sons, 2018.

Y. Guo. Unbiased Least Squares Regression Coefficients for Multiple Linear Regression Mathematical Models. In *2021 4th International Conference on Information Systems and Computer Aided Education*, ICISCAE 2021, page 2063–2066, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450390255. doi: 10.1145/3482632.3484099.

D. A. Harville and D. R. Jeske. Mean squared error of estimation or prediction under a general linear model. *Journal of the American Statistical Association*, 87(419):724–731, 1992.

T. Hastie and R. Tibshirani. Generalized additive models: some applications. *Journal of the American Statistical Association*, 82(398):371–386, 1987.

J. Liesen and V. Mehrmann. *The Kronecker Product and Linear Matrix Equations*, pages 303–310. Springer International Publishing, Cham, 2015. ISBN 978-3-319-24346-7. doi: 10.1007/978-3-319-24346-7_20. URL https://doi.org/10.1007/978-3-319-24346-7_20.

N. Mamoulis. *Spatial Data*, pages 11–19. Springer International Publishing, Cham, 2012. ISBN 978-3-031-01884-8. doi: 10.1007/978-3-031-01884-8_2.

J. Mu, G. Wang, and L. Wang. Estimation and inference in spatially varying coefficient models. *Environmetrics*, 29(1):e2485, 2018.

D. Newman, S. Hettich, C. Blake, and C. Merz. UCI Repository of machine learning databases, 1998. URL http://www.ics.uci.edu/~mlearn/MLRepository.html.

J. S. Racine and T. Hayfield. Nonparametric Econometrics: The np Package. *Journal of Statistical Software*, 27(5):1–32, 2008. doi: 10.18637/jss.v027.i05. URL https://www.jstatsoft.org/index.php/jss/article/view/v027i05.

C. J. Stone. The Use of Polynomial Splines and Their Tensor Products in Multivariate Function Estimation. *The Annals of Statistics*, 22(1):118 – 171, 1994. doi: 10.1214/aos/1176325361.

Y. Sun, H. Yan, W. Zhang, and Z. Lu. A semiparametric spatial dynamic model. *The Annals of Statistics*, 42(2), 2014. doi: 10.1214/13-aos1201.

United States Census Bureau. Glossary, Apr 2022. URL https://www.census.gov/programs-surveys/geography/about/glossary.html.