A Monte Carlo Investigation of Smoothing Methods for Error Density Estimation in Functional Data Analysis with an Illustrative Application to a Chemometric Data Set

A MONTE CARLO INVESTIGATION OF SMOOTHING METHODS FOR ERROR DENSITY ESTIMATION IN FUNCTIONAL DATA ANALYSIS WITH AN ILLUSTRATIVE APPLICATION TO A CHEMOMETRIC DATA SET

BY

JOHN RONALD JAMES THOMPSON

A Thesis

submitted to the Department of Mathematics & Statistics

and the School of Graduate Studies

of McMaster University

in partial fulfilment of the requirements

for the degree of

Master of Science

Master of Science (2014) McMaster University

(Mathematics and Statistics) Hamilton, Ontario, Canada

TITLE:	A Monte Carlo Investigation of Smoothing Methods for Error Density Estimation in Functional Data Analysis with an Illustrative Application to a Chemometric Data Set
AUTHOR:	John Ronald James Thompson B.Sc. (Physics) University of Waterloo, Waterloo, Ontario, Canada
SUPERVISOR:	Dr. Jeffrey S. Racine
NUMBER OF PAGES:	xv, 65

For Juliet Jane Thompson, in memoriam

Abstract

Functional data analysis is a field in statistics that analyzes data which are dependent on time or space and from which inference can be conducted. Functional data analysis methods can estimate residuals from functional regression models that in turn require robust univariate density estimators for error density estimation. The accurate estimation of the error density from the residuals allows evaluation of the performance of functional regression estimation. Kernel density estimation using maximum likelihood cross-validation and Bayesian bandwidth selection techniques with a Gaussian kernel are reproduced and compared to least-squares cross-validation and plug-in bandwidth selection methods with an Epanechnikov kernel. For simulated data, Bayesian bandwidth selection methods for kernel density estimation are shown to give the minimum mean expected square error for estimating the error density, but are computationally inefficient and may not be adequately robust for real data. The (bounded) Epanechnikov kernel function is shown to give similar results as the Gaussian kernel function for error density estimation after functional regression. When the functional regression model is applied to a chemometric data set, the local least-squares cross-validation method, used to select the bandwidth for the functional regression estimator, is shown to give a significantly smaller mean square predicted error than that obtained with Bayesian methods.

Acknowledgements

I am most grateful for the supervision and insights provided by Dr. Jeffrey S. Racine, and especially thankful for his expertise in nonparametric statistics. I would like to acknowledge Dr. H.L. Shang for his insightful contributions on Bayesian functional regression and error density estimation. I also thank the SysCom group at the Computer Science Club of the University of Waterloo, especially Murphy Berzish, Anthony Brenna, Jacob Parker, Mark Burns, and Katie S. Hyatt, for their help with configuring the "high-fructose-corn-syrup" machine for computer simulations. Finally, I am grateful to family and friends for their loving support and helpful suggestions in the editing of this manuscript.

Nomenclature

ESE	estimated square error
FNWKE	functional Nadaraya-Watson kernel estimator
i.i.d.	independently and identically distributed
ISE	integrated square error
kNN	k-nearest neighbours
MCMC	Markov chain Monte Carlo
MESE	mean estimated square error
MISE	mean integrated square error
MSE	mean square error
MSPE	mean square predicted error
NPKDE	nonparametric kernel density estimator
NWKE	Nadaraya-Watson kernel estimator
SKDE	Shang kernel density estimator

SNR	signal-to-noise ratio

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Functional data analysis (FDA) is a field in statistics that comprises methods from parametric and nonparametric statistics, functional analysis, computational statistics, and curve smoothing. FDA allows data that are a function of space or time to be analyzed, and inference to be conducted on them. The contributions in this thesis were motivated by the need to advance FDA methods for analysis of complex sets of data representing continuous functional phenomena. For example, FDA methods have been applied to enhance understanding of phenomena in many diverse disciplines, including criminology, economics, archaeology, and neurophysiology (see Ferraty and Vieu, 2006; Ramsay and Silverman, 2005); and more recently meteorology, chemometrics, earthquake and demographics forecasting, earthquake prediction, gene expression, linguistics, and medicine (see Shang, 2013).

1.1 How are data functional?

Data are considered functional if there is a nonrandom quantity on which a measurement depends (e.g., time, space, temperature, wavelength, etc.). What makes data functional is the correlation between a data point and its neighbouring points. Each "string" of data, within which points are highly correlated with respect to a measurable nonrandom quantity, is called a "data curve" or simply a "curve". Functional models can be used to determine how curves relate to other quantities and to perform inference that naturally accounts for high correlation and dimensions (Ferraty and Vieu, 2006). For example, Figure 1.1 shows the heights of 10 girls taken at 31 different ages from the "growth" data set in the **fda** R package (Ramsay and Silverman, 2005). The study contains 310 *recorded* observations, or data points, but there is correlation in that each girl can only grow taller. Therefore, there are 10 *functional* observations with a girl's height being a function of her age. FDA smoothing techniques are useful for estimating each curve, quantifying the differences between curves, and trying to find relationships between a set of curves and other data.

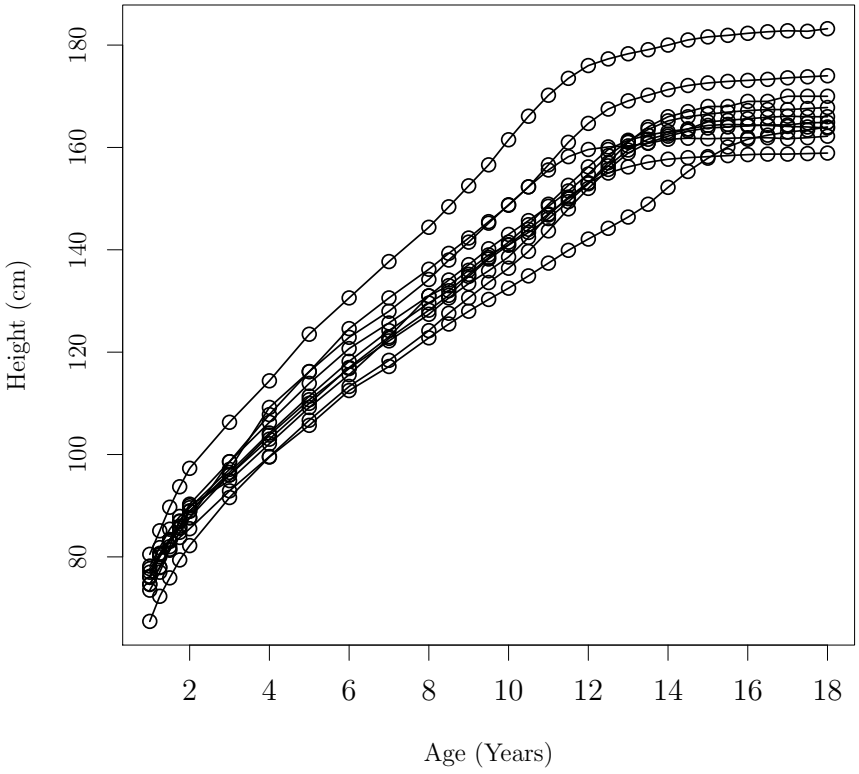

Figure 1.1: The growth of 10 girls measured at 31 different ages. Since the girls were not measured at the *exact* same age, for example Girl A's height is measured at 3 months and 5 days while Girl B's height is measured at 3 months and 9 days, open circles are used for height measurements at every age. This data set can be found in the **fda** R package (Ramsay and Silverman, 2005).

Data curves can be used as explanatory data for a response variable. The relationship between explanatory and response data in the functional setting was analyzed for linear relationships (Hastie and Tibshirani, 1993; Ramsay and Silverman, 2005),

subsequently, nonlinear relationships were estimated with functional polynomial regression models (Yao and Müller, 2010; Horváth and Reeder, 2012), functional additive regression models (Müller and Yao, 2008; Febrero-Bande and González-Manteiga, 2013; Fan and James, 2013), and nonparametric functional regression models (Ferraty and Vieu, 2006; Ferraty et al., 2010). Emphasis in this thesis is placed on analysis using a functional regression model which obtains an estimate of the relationship between a functional variable and a scalar response.

Residuals are calculated from the difference between fitted values of a regression operator and observed response data, and are used as substitutes in error density estimation for the unavailable regression errors (Efromovich, 2005). A nonparametric error density estimate is used to assess the suitability of specified error density assumptions as well as to obtain nonparametric prediction intervals (Akritas and Van Keilegom, 2001). If the data are simulated or come from a real data set, where the error density has a challenging distribution (e.g., bimodal), then nonparametric density estimation is favoured over parametric density estimation. Nonparametric density estimation allows for the data to drive the estimation rather than attempting to specify the shape through a parametric model (Silverman, 1998). A misspecification during the process of parametrically specifying the error density estimation may lead to inaccurate conclusions about the regression estimator's predictive interval. Therefore, sound nonparametric error density estimation using the residuals is paramount for evaluating the predictive capabilities of a regression estimator.

1.2 Scope

The objective of the research presented in this thesis is to evaluate different bandwidth selection methods and kernel functions for univariate error kernel-form density estimation, after functional regression estimation between functional explanatory and scalar response data has been conducted. Chapter 2 describes the methodology of functional regression and univariate kernel-form error density estimation, Chapter 3 presents a simulation study with simulated and real data using methods described in Chapter 2, and Chapter 4 contains conclusions and possible future work for error density estimation in functional regression.

Chapter 2

Methodology

This chapter will describe the different components of FDA that will be used for results detailed in Chapter 3. First, the functional regression model is presented. Second, nonparametric functional regression estimation is described using a functional Nadaraya-Watson kernel estimator with different bandwidth selection methods. Third, a Monte Carlo simulation algorithm for a bootstrap bandwidth selection method is described. Fourth, a Bayesian method for regression estimation and error density estimation is described. Fifth, the Monte Carlo simulation algorithm is modified for use in a two-stage cross-validation procedure, as well as two Bayesian procedures for calculating a functional regression estimate and a kernel-form error density estimate. Last, well-established methods for calculating kernel-form error density estimation using maximum likelihood cross-validation and Bayesian bandwidth selection methods are described. These methods are compared to a proposed least-squares cross-validation bandwidth selection method. Performance under the Gaussian kernel function for error density estimation is compared to that for the (bounded) Epanechnikov kernel function. Chapter 3 contains results obtained with

each of these bandwidth estimation methods and kernel functions.

2.1 Functional regression models

To begin with, the functional regression model that will be used for simulations must be selected. The two models that will be discussed are the functional linear and nonlinear models. The functional linear model is described first, and then extended to the functional nonlinear model. The chemometric data set that the functional nonlinear model will be deployed upon is also described.

2.1.1 Functional linear regression model

The functional linear model (Ramsay and Silverman, 2005) with scalar response data Y, time $t \in (0, 1)$, explanatory functional data $\mathcal{X}(t)$, and error ϵ is written as

$$Y = \alpha + \int_0^1 \rho(t)\mathcal{X}(t)dt + \epsilon. \tag{2.1}$$

This is a parametric model with regression parameters α and ρ (Ferraty and Vieu, 2006). The explanatory functional data are estimated as curves using $\mathcal{X}(t_i) = \sum_{k=1}^{K} c_k\phi_k(t_i)$ with basis functions ϕ_k and their associated weights c_k. The key to fitting a linear model is sound estimation of the parameters α and ρ, while an additional minimization problem is required to estimate the weight functions c_k for the curves.

The choice of basis functions (e.g., B-splines, Fourier series, etc.) is based on the features of the curve data set, such as the periodicity of the data. For example, a curve data set records measurements of temperature at different times during one

year. There are 30 different years and, therefore, 30 functional observations. These data have a period of one year and a Fourier series basis is the suggested basis for this type of data (Ramsay and Silverman, 2005). A B-spline basis is suggested for nonperiodic data (Ramsay and Silverman, 2005) and was used for the heights of girls data set presented in Figure 1.1.

2.1.2 Functional nonlinear regression model

Another approach to FDA extends the parametric functional linear regression model to a nonparametric nonlinear functional regression model (Ferraty and Vieu, 2006). This section considers nonlinear models that may improve the accuracy and robustness of the regression estimator. The functional nonlinear regression model is defined as

$$Y = m(\mathcal{X}) + \epsilon, \tag{2.2}$$

where Y is a scalar response, $\mathcal{X}$ is a data curve, $m(\mathcal{X}) = \mathrm{E}[Y|\mathcal{X}]$ is a regression operator, and ϵ is an error term. A functional data set of N observation curves $\chi_1, \chi_2, \ldots, \chi_N$ is to a "nonfunctional" or real-valued data set of observations $x_1, x_2, \ldots, x_N$ as the functional random variables $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_N$ are to the real-valued random variables $X_1, X_2, \ldots, X_N$. The fixed curve χ in the functional setting is analogous to the fixed point x in the real-valued setting.

A well-adapted space for the infinite dimensional space E for functional data is defined through a semi-norm and semi-metric (Ferraty and Vieu, 2006). A semi-norm, defined in the context of this thesis, is a map that assigns a nonnegative size or length to a curve. A semi-metric, defined in the context of this thesis, is a function that

gives the nonnegative distance between two curves. The choice of semi-norm $||\cdot||$ must by definition satisfy

$$\forall(\lambda, x) \in \mathbb{R} \times E, \|\lambda x\| = |\lambda|\|x\|, \tag{2.3}$$

$$\forall(x, y) \in E \times E, \|x + y\| \leq \|x\| + \|y\|. \tag{2.4}$$

Note that this is equivalent to a norm, except that the condition $\|x\| = 0 \Rightarrow x = 0$ does not need to be satisfied. The semi-metric $d(\cdot, \cdot)$ determined by semi-norm $||\cdot||$ must by definition satisfy

$$\forall x \in E, d(x, x) = 0, \tag{2.5}$$

$$\forall(x, y, z) \in E \times E \times E, d(x, y) \leq d(x, z) + d(z, y). \tag{2.6}$$

The semi-metric $d(\cdot, \cdot)$ is equivalent to a metric, except that the condition $d(x, y) = 0 \Rightarrow x = y$ does not need to be satisfied. Some suggested semi-metrics are based on a derivative of the estimated curves, on principal component analysis, or on partial least-squares (Ferraty and Vieu, 2006). Functional data that appear to be continuous or "smooth" allows for the choice of semi-metric to be based on the derivatives of the data curves. The equation for semi-metrics of the p^{th} derivative of observation curves χ_i and χ_j is

$$d(\chi_i, \chi_j) = \sqrt{\int \left(\chi_i^{(p)}(t) - \chi_j^{(p)}(t)\right)^2 dt}. \tag{2.7}$$

Vertical differences in data curves, as seen below in the chemometric data set in Figure 2.1, are known errors in measurement. The interest lies in comparing a change in

absorbance to a change in wavelength. It can be difficult to shift the curves vertically to compare features like maxima or minima. By choosing $p = 2$ for the order of the derivative in Equation (2.7), the distance between curves can be based on the acceleration of the absorbances. To be able to take a derivative of a curve, smooth continuous basis functions are needed that appropriately estimate each data curve. A B-spline method (de Boor, 1978; Schumaker, 1981) can be used to smooth data. Each data curve $\boldsymbol{\chi}_i = (\chi_i(t_1), \chi_i(t_2), \ldots, \chi_i(t_J))^\top$ is estimated by B-spline basis functions $\{B_1, \ldots, B_B\}$ using the minimization problem (Ferraty and Vieu, 2006)

$$\widehat{\beta}_i = \left(\widehat{\beta}_{i1}, \ldots, \widehat{\beta}_{iB}\right) = \underset{(\alpha_1,\ldots,\alpha_B)\in\mathbb{R}^B}{\arg\inf} \sum_{j=1}^{J} \left(\chi_i(t_j) - \sum_{b=1}^{B} \alpha_b B_b(t_j)\right)^2. \tag{2.8}$$

The estimator for curve $\boldsymbol{\chi}_i$ and its q^{th} derivative is given by

$$\widehat{\chi}_i(\cdot) = \sum_{b=1}^{B} \widehat{\beta}_{ib} B_b(\cdot), \tag{2.9}$$

$$\widehat{\chi}_i^{(q)}(\cdot) = \sum_{b=1}^{B} \widehat{\beta}_{ib} B_b^{(q)}(\cdot). \tag{2.10}$$

The B-spline approach for estimation of each data curve allows for easy calculation of derivatives and is a natural partner to the derivative semi-metric in Equation (2.7). A B-spline basis must be p-times continuously differentiable for the derivative semi-metric (Ferraty and Vieu, 2006; Ramsay and Silverman, 2005).

2.1.3 Analysis of smooth data

A popular data set known as the *tecator* chemometric data set, provided by food industry company Tecator, is often evaluated by functional regression methods. The

tecator data set is ideal for present purposes because it has functional predictor and scalar response spectrometric measurements. This data set is found at `http://lib.stat.cmu.edu/datasets/tecator`, and was originally analyzed using a neural networks approach (Borggaard and Thodberg, 1992). The data set was made with a Tecator Infratec Food and Feed Analyzer using a range of infrared light measuring a 100 channel spectrum of absorbances (between 850 and 1050 nm) on meat samples with different protein, moisture, and fat content. The absorbance is $-\log_{10}$ of the transmittance measured by a spectrometer. Figure 2.1 shows a portion of this data set with 20 out of the total 215 curve observations and 100 evenly spaced wavelength points.

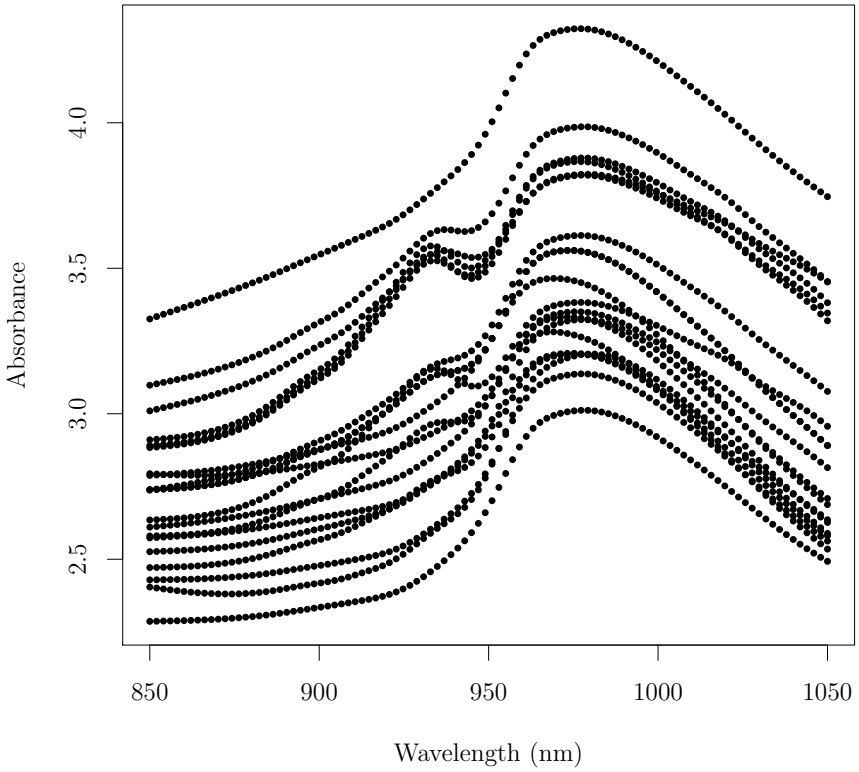

Figure 2.1: The absorbance of infrared light, as a function of the infrared light's wavelength, in meats with different protein, moisture, and fat content for 20 samples. Data set retrieved from `http://lib.stat.cmu.edu/datasets/tecator`.

Using the nonparametric functional regression model, the regression operator m that maps the functional explanatory data (absorbances are a function of wavelength) to a real-valued response (protein, moisture, or fat content) can be estimated. After smoothing using B-splines, one of the functional regression estimators that will be described in Section 2.2 can be used to estimate the regression operator $m(\mathcal{X})$.

2.1.4 Functional regression model summary

The following is a summary of the steps taken when estimating a functional model:

(1) Data are paired $(Y_i, \mathcal{X}_i)$ for $i = 1, \ldots, N$ where N is the sample size.

(2) $\mathcal{X}_i \in E$ is the i^{th} functional random variable.

(3) $Y_i \in \mathbb{R}$ is the i^{th} scalar response.

(4) The nonparametric functional regression model is $Y_i = m(\mathcal{X}_i) + \epsilon_i$.

(5) The regression operator m is a smooth function, defined as $m(\mathcal{X}_i) = E[Y_i|\mathcal{X}_i]$, and its estimate is written as $\widehat{m}(\mathcal{X}_i)$.

(6) The semi-metric $d(\cdot, \cdot)$ is the derivative semi-metric defined in Equation (2.7).

(7) The errors ϵ_i are assumed to satisfy $E[\epsilon_i] = 0$ and $E[\epsilon_i^2] \neq 0$.

(8) The error ϵ_i is assumed to be independent of both the error ϵ_j (where $j \neq i$) and the curve $\mathcal{X}_i$.

(9) The residuals are given by $\widehat{\epsilon_i} = Y_i - \widehat{m}(\mathcal{X}_i)$.

So far, the parametric linear and nonparametric nonlinear functional regression models and the chemometric data set of interest have been presented. Next, methods for estimating the regression operator in the nonparametric functional regression model are present.

2.2 Functional regression estimation

This section describes functional regression estimation with kernel methods. First, kernel functions for the real-valued setting are extended to the functional setting. Second, the Nadaraya-Watson kernel estimator for the real-valued regression setting is extended to the functional setting with global and local cross-validation selected

bandwidths. Third, an algorithm is described to calculate an optimal bandwidth for functional regression using a bootstrap. Fourth, a Bayesian method that calculates an optimal bandwidth for functional regression, and global and local selected bandwidths for error density estimation, is described.

2.2.1 Kernel functions in the functional setting

A kernel local weighting transform Δ_i for the real-valued setting with kernel function K and bandwidth h for independently and identically distributed (i.i.d.) real-valued sample $X_1, X_2, \ldots, X_N$, where $X_i \in \mathbb{R}$, is given by

$$\Delta_i = \Delta_i(x, h, K) = \frac{1}{h} K\left(\frac{x - X_i}{h}\right). \tag{2.11}$$

The bandwidth h is a smoothing parameter. The challenge, when using kernel weighting for regression and error density estimation, is selecting an optimal bandwidth. The selection of a bandwidth for kernels is dependent on the choice of semi-metric used to calculate the difference between curves (Ferraty and Vieu, 2006). The multivariate kernel weighting transform with random vectors $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$, where $\mathbf{X}_i \in \mathbb{R}^q$, is given by

$$\Delta_i = \frac{1}{\prod_{j=1}^{q} h_j} K^*\left(\frac{x - \mathbf{X}_i}{h}\right), \tag{2.12}$$

where $K^*(\mathbf{u})$ is the product of q kernel functions defined as

$$K^*(\mathbf{u}) = K_1(u_1) \times K_2(u_2) \times \cdots \times K_q(u_q) \tag{2.13}$$

for any vector $\mathbf{u} = (u_1, u_2, \ldots, u_p)^\top \in \mathbb{R}^q$.

Kernel functions and weightings can be extended from the multivariate real-valued setting to the functional setting. To illustrate this, consider a random curve sample $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_N$ that lies in an infinite dimensional space E. The naive kernel local weighting transform for the functional setting (Ferraty and Vieu, 2006) can be written as

$$\Delta_i = \frac{1}{V(h)} K\left(\frac{d(\chi, \mathcal{X}_i)}{h}\right), \tag{2.14}$$

where the semi-metric $d(\cdot, \cdot)$ is chosen to the derivative semi-metric in Equation (2.7), and $V(h)$ is the volume of a ball B with radius h and centred at χ, defined as

$$B(\chi, h) = \{\chi' \in E \,:\, d(\chi, \chi') \leq h\}. \tag{2.15}$$

The topology of the ball's surface is induced by the chosen semi-metric $d(\cdot, \cdot)$. The volume of the ball $V(h)$ allows for the normalization of the kernel function in the functional space E. However, the calculation of $V(h)$ requires a measure on the functional space E, and there is no universally accepted reference measure for E like the Lebesque measure for the Euclidean space $\mathbb{R}$ (Ferraty and Vieu, 2006). To free the normalization from this choice, it has been suggested that the normalization factor be based on the probability distribution of the functional random variable $\mathcal{X}_i$ (Ferraty and Vieu, 2006) given by

$$\Delta_i = \frac{K\left(\frac{d(\chi, \mathcal{X}_i)}{h}\right)}{E\left(K\left(\frac{d(\chi, \mathcal{X}_i)}{h}\right)\right)}. \tag{2.16}$$

Basing the weighting on $\mathcal{X}_i$ in this way allows for the data to fully drive the smoothing.

In the univariate real-valued setting, a kernel function K is symmetric, so that $K(t) = K(-t)$. However, in the multivariate and functional setting, a kernel function is restricted to be asymmetric, with $K(t) > 0$, since $d(\cdot, \cdot)$ is defined to be strictly nonnegative (Ferraty and Vieu, 2006). The choice of kernel function K for the chemometric data set is often the asymmetric quadratic kernel (Benhenni et al., 2007; Ferraty et al., 2008, 2010; Shang, 2013) defined as

$$K(u) = \frac{3}{2}(1 - u^2),\ u \in (0, 1). \tag{2.17}$$

2.2.2 The Nadaraya-Watson kernel estimator

The Nadaraya-Watson kernel estimator (NWKE) is used to estimate a nonparametric regression operator for real-valued data (Nadaraya, 1964; Watson, 1964). The NWKE $\widehat{g}(x)$ for the regression operator $g(x) = E[Y|x]$ with paired data $\{(\mathbf{X}_i, Y_i)\}_{i=1,\ldots,n}$ lying in the space $\mathbb{R}^q \times \mathbb{R}$ for the regression model $Y_i = g(X_i) + \epsilon$ is given by

$$\widehat{g}(x) = \frac{\sum_{i=1}^{n} Y_i K^*\left(\frac{x-\mathbf{X}_i}{h}\right)}{\sum_{i=1}^{n} K^*\left(\frac{x-\mathbf{X}_i}{h}\right)}, \tag{2.18}$$

where $K^*\left(\frac{x-\mathbf{X}_i}{h}\right)$ is a product of q asymmetric kernel functions given in Equation (2.13) and $h = (h_1, h_2, \ldots, h_q)$ are global bandwidths. The key to nonparametric estimation of the regression operator using kernel functions is to select a bandwidth h that minimizes an approximation to the mean square error (MSE). Data-driven bandwidth selection methods have been described for the real-valued setting, such as the local constant least-squares cross-validation bandwidth selection method (Li and Racine, 2007). This method for optimizing bandwidth vector $(h_1, h_2, \ldots, h_q)$

minimizes the local constant least-squares cross-validation criterion given by

$$CV_{lc}(h) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{g}_{-i}(\mathbf{X}_i))^2 M(\mathbf{X}_i), \tag{2.19}$$

where $M(\mathbf{X}_i)$ is some weight function and $\widehat{g}_{-i}(\mathbf{X}_i)$ is the leave-one-out kernel estimator defined as

$$\widehat{g}_{-i}(\mathbf{X}_i) = \frac{\sum_{j=1,j\neq i}^{n} Y_j K^*\left(\frac{\mathbf{X}_i - \mathbf{X}_j}{h}\right)}{\sum_{j=1,j\neq i}^{n} K^*\left(\frac{\mathbf{X}_i - \mathbf{X}_j}{h}\right)}. \tag{2.20}$$

Using a bandwidth vector optimized via cross-validation in the regression estimator $\widehat{g}(\mathbf{X}_i)$ defined in Equation (2.18), the residuals can be calculated using $\widehat{\epsilon_i} = Y_i - \widehat{g}(\mathbf{X}_i)$ and an error density estimate can be calculated from the residuals.

2.2.3 Functional regression with a global cross-validated bandwidth

The NWKE in Equation (2.18) can be extended to the functional regression setting. Consider the univariate functional regression model where $(\mathcal{X}_i, Y_i) \in E \times \mathbb{R}$. The extension of the NWKE to the functional setting (Helland, 1990; Ferraty and Vieu, 2006), or the functional Nadaraya-Watson kernel estimator (FNWKE), is written as

$$\widehat{m}(\chi) = \frac{\sum_{i=1}^{n} Y_i K\left(\frac{d(\chi,\mathcal{X}_i)}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{d(\chi,\mathcal{X}_i)}{h}\right)}, \tag{2.21}$$

where bandwidth h is global to all fixed curves χ, and dependent on the curve sample size n (Benhenni et al., 2007). For the FNWKE in Equation (2.21), the upper bounds

on rates of convergence (Ferraty and Vieu, 2006, Chapter 6), the bias, variance, and MSEs, and the asymptotic distribution (Ferraty et al., 2008) are all known.

The number of bandwidths to optimize has changed from q bandwidths $(h_1, h_2, \ldots, h_q)$ in the multivariate real-valued setting to one bandwidth h in the functional setting. This is a direct result from the change in the space of the explanatory variable from $\mathbb{R}^q$ to E. The local constant least-squares cross-validation optimized bandwidth $\widehat{h}$ (Härdle and Marron, 1985) is given by

$$\widehat{h} = \underset{h \in H_n}{\arg\min}\, GCV_x(h) = \underset{h \in H_n}{\arg\min} \left\{ n^{-1} \sum_{i=1}^{n} (Y_i - \widehat{m}_h^{-i}(\mathcal{X}_i))^2 W(\mathcal{X}_i) \right\}, \tag{2.22}$$

where $W(\mathcal{X}_i)$ is a weight function, H_n is a set of possible bandwidths, and $\widehat{m}_h^{-i}(X_i)$ is the leave-one-curve-out estimator defined as

$$\widehat{m}_h^{-i}(\mathcal{X}_i) = \frac{\sum_{j=1, j\neq i}^{n} Y_j K\left(\frac{\mathcal{X}_i - \mathcal{X}_j}{h}\right)}{\sum_{j=1, j\neq i}^{n} K\left(\frac{\mathcal{X}_i - \mathcal{X}_j}{h}\right)}. \tag{2.23}$$

2.2.4 Functional regression with local cross-validated bandwidths

Local least-squares cross-validation is an extension of the global procedure in the previous section (Benhenni et al., 2007; Ferraty and Vieu, 2006). A local bandwidth refers to a bandwidth that changes depending on the fixed curve χ. The local bandwidth cross-validation procedure has weight functions $W_{n,\chi}$ that depend on the fixed curve χ and the sample size n. The local bandwidth $\widehat{h}_\chi$ calculated using the local

constant least-squares cross-validation procedure (Benhenni et al., 2007) is given by

$$\widehat{h}_{\chi} = \underset{h \in H_n}{\arg\min}\, LCV_{\chi}(h) = \underset{h \in H_n}{\arg\min} \left\{ n^{-1} \sum_{i=1}^{n} (Y_i - \widehat{m_h}^{-i}(\mathcal{X}_i))^2 W_{n,\chi}(\mathcal{X}_i) \right\}. \quad (2.24)$$

The weight functions $W_{n,\chi}$ are given by

$$W_{n,\chi}(\mathcal{X}_i) = \begin{cases} 1, & d(\chi, \mathcal{X}_i) < h \\ 0, & \text{otherwise.} \end{cases} \quad (2.25)$$

If the bandwidth h is selected to include k data curves, this local cross-validation procedure becomes a k-nearest neighbours (kNN) selected bandwidth h_k for nonparametric regression estimators. By choosing the number of neighbours for smoothing, regression estimators adapt to local information in data.

For a kernel (or any) estimator $\widehat{m}(\chi)$ to be consistent, it must converge in probability to the underlying data generating process $m(\chi)$ that it is estimating as the number of samples n grows indefinitely (Casella and Berger, 2002); that is,

$$\lim_{n \to \infty} P(|\widehat{m}(\chi) - m(\chi)| \geq \varepsilon) = 0, \forall \varepsilon > 0, \chi \in E. \quad (2.26)$$

It has been shown that for a kernel estimator to be consistent, the bandwidth h must satisfy $h \to 0$ and $nh \to \infty$ as $n \to \infty$ (Parzen, 1992). The rate of convergence is how quickly the bandwidth converges to zero, relative to the sample size n (no faster than $o(n^{-1})$), as $n \to \infty$. The estimator $\widehat{m}(\chi)$ is said to have a almost complete rate

of convergence of $O(u_n)$ to $m(\chi)$ (Ferraty and Vieu, 2006) if and only if

$$\sum_{n\in\mathbb{N}} P(|\widehat{m}(\chi)| > \varepsilon u_n) < \infty, \forall \varepsilon > 0, \chi \in E. \tag{2.27}$$

The kNN method in the functional setting has an almost complete rate of convergence, a fully developed methodology for bandwidth selection as well as applications using simulated and real data (Burba et al., 2009). The bandwidth h is calculated to include k neighbouring data curves using the chosen semi-metric. The estimate $\widehat{m}_{kNN}(\chi)$ for the regression operator $m(\chi)$ using the kNN method is given by

$$\widehat{m}_{kNN}(\chi) = \sum_{i=1}^{n} Y_i \omega_{i,n}(\chi), \;\; \omega_{i,n}(\chi) = \frac{K(H_{n,k}(\chi)^{-1} d(\chi, \mathcal{X}_i))}{\sum_{i=1}^{n} K(H_{n,k}(\chi)^{-1} d(\chi, \mathcal{X}_i))}, \tag{2.28}$$

where K is an asymmetric kernel function and $H_{n,k}(\chi)$ is a positive random variable given by

$$H_{n,k}(\chi) = \min \left\{ h \in \mathbb{R}^+ \; : \; \sum_{i=1}^{n} 1_{B(\chi,h)}(\mathcal{X}_i) = k \right\}. \tag{2.29}$$

The local bandwidth $H_{n,k}(\chi)$, which depends on the explanatory data $(\mathcal{X}_1, \ldots, \mathcal{X}_n)$, is the minimum bandwidth to contain k data curves in the ball $B(\chi, h)$ in Equation (2.15) centred at the fixed curve χ.

In a simulation involving smooth data with low variability similar to the chemometric data set, the regression estimator with local bandwidths had a lower MSE, and, therefore, estimates the regression operator $m(\chi)$ more accurately, than with a global bandwidth (Benhenni et al., 2007). Local and global bandwidth selection methods were applied to the chemometric data set in Figure 2.1 and it was observed that

locally selecting the bandwidth between curves gave a lower mean square predicted error (MSPE) than selecting it globally (Benhenni et al., 2007). The MSPE measures the difference between an estimator's predicted response and the true response.

2.2.5 Bootstrapping residuals

The purpose of this section is to look at an algorithm that can be used to select an optimal bandwidth for functional regression and to obtain pointwise confidence intervals of a regression estimator. The simulated explanatory and response data, which are used for simulations in this thesis, are

$$\text{Data} \quad : \quad \mathcal{X}_i(t_j) = a_i \cos(2t_j) + b_i \sin(4t_j) + c_i(t_j^2 - \pi t_j + \frac{2}{9}\pi^2), \tag{2.30}$$

$$\text{Model 1} \quad : \quad m(\mathcal{X}_i) = 10(a_i^2 - b_i^2) \tag{2.31}$$

$$\text{Model 2} \quad : \quad m(\mathcal{X}_i) = \int_0^\pi t \cos(t) \left(\mathcal{X}_i'(t)\right)^2 dt, \tag{2.32}$$

where $i = 1, \ldots, N$ and N is the curve sample size, $t_j \in [0, \pi]$ for $j = 1, \ldots, 100$ are equispaced points, and a_i, b_i, and c_i are independent random variables with a $[0, 1]$ uniform distribution. A simulation study was conducted (Ferraty et al., 2008, 2010) using samples from the models in Equations (2.30), (2.31), and (2.32). One sample of size $n_1 = 250$ was used to train the FNWKE and a second sample of size $n_2 = 100$ was used to test the estimator's performance using $N_B = 1000$ bootstrap replicates and $M = 100$ Monte Carlo replicates. Different numbers of bootstrap and Monte Carlo replicates were considered, but did not affect the results significantly enough

to alter the conclusions (Ferraty et al., 2008). Bandwidths were selected from the set

$$h = h(\chi) \in \{h_1, h_2, \ldots, h_{32}\} = H, \tag{2.33}$$

using the kNN method for $k = 1, \ldots, 32$ (Ferraty et al., 2008, 2010).

An algorithm using a "wild" bootstrap procedure to obtain a data-driven optimal bandwidth was developed for the functional setting by Ferraty et al. (2008, 2010). The asymptotic validity of the wild bootstrap in the functional setting is known (Ferraty et al., 2010). The wild bootstrap procedure, with an automatic rule for selecting a bandwidth in the FNWKE, was used on simulated and real data sets by Ferraty et al. (2008). Before using the wild bootstrap procedure, the regression estimate and residuals were calculated with the following algorithm:

1. Generate the simulated curves $\mathcal{X}_i$ where $i = 1, \ldots, N$ using Equation (2.30) and the simulated regression operators $m(\mathcal{X}_i)$ using Equation (2.31) or (2.32).
2. Generate ϵ_i from a Gaussian distribution with different signal-to-noise ratios (SNR), i.e. $\text{var}(\epsilon_i) = \text{SNR} \times \text{var}(\{m(\mathcal{X}_i)\}_{i=1,\ldots,N})$.
3. Compute $Y_i = m(\mathcal{X}_i) + \epsilon_i$.
4. Replicate Steps 1–3 for $s = 1, \ldots, M$ for $M = 100$ Monte Carlo replications to generate $(\mathcal{X}_i^s, Y_i^s)$.
5. Compute N estimates of $\widehat{m}_h^s(\mathcal{X})$ for each Monte Carlo replicate.
6. Estimate the error density $\widehat{f}_{true}(\widehat{\epsilon})$ using a univariate kernel density estimator from $\{\widehat{m}_h^s(\chi) - m(\chi)\}$.

The wild bootstrap procedure (Härdle, 1989; Härdle and Marron, 1991) is used to bootstrap the errors, which are real-valued random variables (Ferraty et al., 2010).

The wild bootstrap algorithm is as follows:

1. Estimate the regression operator $\widehat{m}_b(\chi)$ with some fixed bandwidth b not equal to h and calculate residuals $\{\widehat{\epsilon}_{i,b} = Y_i - \widehat{m}_b(\mathcal{X}_i)\}$.
2. Draw n i.i.d. random variables V_i that are independent of $\{(\mathcal{X}_i, Y_i)\}$, where $E[V_i] = 0$ and $E[V_i^2] = 1$.
3. Calculate $\{\epsilon_i^{\text{boot}} = V_i\widehat{\epsilon}_{i,b}\}_{i=1,\ldots,n}$ and define $Y_i^{\text{boot}} = \widehat{m}_b + \epsilon_i^{\text{boot}}$.
4. Calculate the bootstrapped regression estimate $\widehat{m}_{h,b}^{\text{boot}}(\chi) = \frac{\sum_{i=1}^n Y_i^{\text{boot}} K(h^{-1}d(\mathcal{X}_i,\chi)}{\sum_{i=1}^n K(h^{-1}d(\mathcal{X}_i,\chi))}$.
5. Construct a new regression operator $\widehat{m}_b^s(\chi)$ under a fixed bandwidth b.
6. Construct $B = 1000$ bootstrap estimates $\widehat{m}_{h,b}^{\text{boot}}(\chi)$ from Steps 1–4.
7. Estimate the bootstrapped error density $\widehat{f}_{\text{boot}}(\widehat{\epsilon})$ using a univariate kernel density estimator from $\{\widehat{m}_{h,b}^{\text{boot1}}(\chi) - \widehat{m}_b(\chi), \widehat{m}_{h,b}^{\text{boot2}}(\chi) - \widehat{m}_b(\chi), \ldots, \widehat{m}_{h,b}^{\text{boot1000}}(\chi) - \widehat{m}_b(\chi)\}$.

There are a few options for the distribution of the V_i, such as the Rademacher distribution with outcomes $\pm\frac{1}{2}$ and probability $\frac{1}{2}$ for each outcome, or a sum of two Dirac distributions (Härdle and Marron, 1991; Ferraty et al., 2008) given by

$$V_i = \begin{cases} \frac{-\sqrt{(5)}+1}{2}, & \text{with prob. } \frac{\sqrt{(5)}+1}{2\sqrt{(5)}}, \\ \frac{\sqrt{(5)}+1}{2}, & \text{with prob. } \frac{\sqrt{(5)}-1}{2\sqrt{(5)}}. \end{cases} \tag{2.34}$$

Bandwidth selection is calculated from the wild bootstrap by choosing the bandwidth using

$$h_{opt}(\mathcal{X}) = \underset{h\in H}{\arg\min} \left(\sum_{boot=1}^{N_B} \widehat{m}_{h,b}^{boot} - \widehat{m}_b(\chi) \right). \tag{2.35}$$

The purpose of the bootstrap methodology also includes producing pointwise confidence intervals for the regression operator. Confidence intervals describe how well

the model is fitting the data, but prediction intervals describe a range for which the next realization of the response will fall with a specific probability. Since the development of the functional bootstrap, methods using two-stage cross-validation (to be discussed in Section 2.3) and Bayesian methods have been developed to calculate prediction intervals from error density estimation for functional regression.

2.2.6 Bayesian methods for functional regression estimation

Bayesian methods can be used to estimate the functional regression operator and error density using an algorithm (Shang, 2013) similar to the algorithm that uses the wild bootstrap procedure. Bayesian estimation methods have been shown to perform as well as or better than two-stage cross-validation for estimating both the functional regression operator and the real-valued error density (Shang, 2013). A Bayesian Markov chain Monte Carlo (MCMC) algorithm can be used to select bandwidths for the regression and error density estimates. The optimal regression estimator and density estimator bandwidths (h_n, b_n) are selected by maximizing the kernel likelihood of $\mathbf{y} = (y_1, y_2, \ldots, y_n)^\top$. This method for maximizing the kernel likelihood could be accomplished by directly applying kernel methods. However, there is no known literature for asymptotic support of this method and it is outside the scope of this thesis.

For the Bayesian approach to maximizing the likelihood, the error density can be approximated (Jaki and West, 2008, 2011) by the leave-one-out estimator as

$$\widehat{f}_{-i}(\widehat{\epsilon}_i; b_n) = \frac{1}{n-1} \sum_{j=1, j\neq i}^{n} \frac{1}{b_n} \phi\left(\frac{\widehat{\epsilon}_i - \widehat{\epsilon}_j}{b_n}\right), \tag{2.36}$$

where $\phi(\cdot)$ is a standard Gaussian density function and $\widehat{\epsilon}_i = Y_i - \widehat{m}(\mathcal{X}; h_n)$ is the i^{th} residual. This density estimator can be adapted for a local bandwidth $c(1 + c_\epsilon|\widehat{\epsilon_j}|)$ assigned to $\widehat{\epsilon_j}$ (Shang, 2013) and is given by

$$\widehat{f}(\widehat{\epsilon}_i; c, c_\epsilon) = \frac{1}{n-1} \sum_{j=1, j\neq i}^{n} \frac{1}{c(1 + c_\epsilon|\widehat{\epsilon_j}|)} \phi\left(\frac{\widehat{\epsilon}_i - \widehat{\epsilon}_j}{c(1 + c_\epsilon|\widehat{\epsilon_j}|)}\right). \tag{2.37}$$

Note that only b_n will be used for the bandwidth of the error density for simplicity of notation. The posterior distribution of h_n^2 and b_n^2 is approximated by

$$\pi(h_n^2, b_n^2|\mathbf{y}) \propto \widehat{L}(\mathbf{y}|h_n, b_n)\pi(h_n^2)\pi(b_n^2). \tag{2.38}$$

Since the regression model in Equation (2.2) assumes that the errors and the regression operator are uncorrelated, we can assume that each of the estimator bandwidths are uncorrelated (Shang, 2013). The kernel likelihood is given by

$$\widehat{L}(\mathbf{y}|h_n, b_n) = \prod_{i=1}^{n} \left[\frac{1}{n-1} \sum_{j=1, j\neq i}^{n} \frac{1}{b_n} \phi\left(\frac{\widehat{\epsilon}_i - \widehat{\epsilon}_j}{b_n}\right)\right]. \tag{2.39}$$

Prior densities are assumed to be inverse Gamma distributions with hyperparameters $\alpha_h = \alpha_b = 1.0$ and $\beta_h = \beta_b = 0.05$ (Shang, 2013; Geweke, 2010) given by

$$\pi(h^2) = \frac{(\beta_h)^{\alpha_h}}{\Gamma(\alpha_h)} \left(\frac{1}{h^2}\right)^{\alpha_h+1} \exp\left(-\frac{\beta_h}{h^2}\right), \tag{2.40}$$

$$\pi(b^2) = \frac{(\beta_b)^{\alpha_b}}{\Gamma(\alpha_b)} \left(\frac{1}{b^2}\right)^{\alpha_b+1} \exp\left(-\frac{\beta_b}{b^2}\right). \tag{2.41}$$

An MCMC algorithm is given below (Shang, 2013) with an adaptive block random-walk Metropolis algorithm (Garthwaite et al., 2010) to optimize the bandwidths $\boldsymbol{\theta}_n =$

(h_n^2, b_n^2):

1. Specify the starting point $\boldsymbol{\theta}_n^{(0)} \in U(0,1)$ and adaptive tuning parameter $\tau^{(0)}$.
2. Calculate $\boldsymbol{\theta}_n^{(k)} = \boldsymbol{\theta}_n^{(k-1)} + \tau^{(k-1)}\boldsymbol{\epsilon}$, where the error has standard Gaussian distribution.
3. Accept $\boldsymbol{\theta}_n^{(k)}$ with probability $\min\left\{\frac{\pi(\boldsymbol{\theta}_n^{(k)}|\mathbf{y})}{\pi(\boldsymbol{\theta}_n^{(k-1)}|\mathbf{y})},\ 1\right\}$.
4. Set the tuning parameter $\tau^{(k)}$ using a stochastic search algorithm (Robbins and Monro, 1951).
5. Repeat Steps 2–4 $M+N$ times, discarding M results to allow transients to wear off, and estimate the optimal bandwidths as $\widehat{h}_n = \frac{1}{N}\sum_{k=M+1}^{M+N} h_n^{(k)}$ and $\widehat{b}_n = \frac{1}{N}\sum_{k=M+1}^{M+N} b_n^{(k)}$.

The k^{th} tuning parameter (Robbins and Monro, 1951) is given by

$$\tau^{(k)} = \begin{cases} \tau^{(k-1)}(1 + \frac{1-p}{k}), & \text{if } \boldsymbol{\theta}_n^{(k)} \text{is accepted,} \\ \tau^{(k-1)}(1 + \frac{p}{k}), & \text{if } \boldsymbol{\theta}_n^{(k)} \text{is rejected,} \end{cases} \tag{2.42}$$

where $p = 0.234$ is the optimal acceptance probability for drawing multiple parameters (Roberts and Rosenthal, 2009). Using these methods, bandwidths can be calculated that are simultaneously optimal for functional regression and error density estimation. The objective of this thesis is to compare the performance of regression estimates using Bayesian versus cross-validation bandwidth selection methods.

2.3 Estimation of error density from functional regression residuals

The ability to estimate error density for functional regression models is as important as the ability to estimate the regression operator (Shang, 2013). Error density estimates are used (1) to assess the adequacy of error distribution assumption (Shang, 2013), (2) to test the symmetry of the residual distribution (Neumeyer and Dette, 2007), (3) to quantify statistical inference, prediction, and model validation (Muhsal and Neumeyer, 2010), and (4) to determine the density of the response variable (Escanciano and Jacho-Chávez, 2012). The cumulative distribution function for the error density is used in the calculation of the prediction interval. Consequently, the Bayesian and cross-validation approaches to kernel-form error density estimation in a nonparametric functional regression model with functional predictors and scalar responses have been investigated (Shang, 2013). The contributions in this thesis extend that investigation to improve nonparametric error density estimation for functional regression. The goal is to prescribe guidelines that may increase the accuracy of error density estimation in challenging data sets such as the chemometric data set introduced earlier.

There is a proposed functional adaptation for using a two-stage cross-validation bandwidth selection method to estimate the error density (Samb, 2011). The first stage uses the FNWKE with a least-squares cross-validated bandwidth to estimate the functional regression operator. The second stage uses the univariate kernel density estimator on the residuals with a maximum likelihood cross-validated bandwidth for error density estimation. For the first stage, the functional regression estimate $\widehat{m}(\mathcal{X}_i)$

is calculated using the FNWKE and least-squares cross-validation with the leave-one-curve-out estimator $\widehat{m}_h^{-i}(\mathcal{X}_i)$ given by

$$\widehat{m}_h^{-i}(\mathcal{X}_i) = \frac{\sum_{j=1,j\neq i}^{n} Y_j K_0\left(\frac{d(\mathcal{X}_j,\mathcal{X}_i)}{h_0}\right)}{\sum_{j=1,j\neq i}^{n} K_0\left(\frac{d(\mathcal{X}_j,\mathcal{X}_i)}{h_0}\right)}, \tag{2.43}$$

where K_0 is the asymmetric quadratic kernel function with bandwidth h_0 for estimating the regression operator $m(\chi)$ and $d(\cdot,\cdot)$ is the derivative semi-metric. With an estimate of the regression operator $\widehat{m}(\mathcal{X}_i)$, the residuals can be calculated using $\widehat{\epsilon}_i = Y_i - \widehat{m}(\mathcal{X}_i)$. The error density estimator is given by

$$\widehat{f}_n(\epsilon) = \frac{1}{Nh_1}\sum_{i=1}^{N} K_1\left(\frac{\widehat{\epsilon}_i - \epsilon}{h_1}\right), \tag{2.44}$$

with symmetric kernel K_1 (Shang, 2013). The bandwidth h_1 is optimized using a second stage cross-validation or Bayesian bandwidth selection method (Shang, 2013). The algorithm for estimating the functional regression operator and error density (Shang, 2013) is very similar to the previous algorithm that used a bootstrap procedure in the functional setting (Ferraty et al., 2008, 2010):

1. Generate the simulated curves $\{\mathcal{X}_i\}$ using Equation (2.30), where $i = 1,\ldots,N$, and calculate the simulated regression operators $m(\mathcal{X}_i)$ using Equation (2.31) or (2.32).
2. Generate simulated errors ϵ_i from a Gaussian distribution with different SNRs, i.e. $\text{var}\,(\epsilon_i) = \text{SNR} \times \text{var}(\{m(\mathcal{X}_i)\}_{i=1,\ldots,N})$.
3. Compute the simulated responses $Y_i = m(\mathcal{X}_i) + \epsilon_i$.
4. Replicate Steps 1–3 for $s = 1,\ldots,M$ for $M = 100$ Monte Carlo replications to

generate $(\mathcal{X}_i^s, Y_i^s)$.

5. Compute N estimates $\widehat{m}_h^s(\mathcal{X})$ for each Monte Carlo replicate.
6. Calculate the MSE by averaging over all Monte Carlo replicate square errors $(\widehat{m}_h^s(\mathcal{X}) - m(\mathcal{X}))^2$.

This algorithm is used for the simulation study of this thesis. In the simulation study, the kernel-form error density estimator is applied to the residuals using different bandwidth selection methods and kernel functions, which are described in the next section.

2.4 Univariate kernel-form density estimation

2.4.1 Existing bandwidth selection methods and kernel function

In this section, bandwidth selection methods and kernel functions are described for kernel-form density estimation from the residuals of functional regression (Shang, 2013). The kernel used for error density estimation for the two-stage cross-validation and global and local Bayesian methods is a standard normal Gaussian kernel $K_1(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$, $t \in \mathbb{R}$ for the kernel density estimator in Equation (2.44) (Shang, 2013). Maximum likelihood cross-validation is given by maximizing the leave-one-out log likelihood function for bandwidth h (Li and Racine, 2007) defined as

$$\mathcal{L} = \sum_{i=1}^{n} \ln \widehat{f}_{-i}(X_i), \tag{2.45}$$

where the leave-one-out density estimator $\widehat{f}_{-i}(X_i)$ is given in Equation (2.20). The equation for the optimal bandwidth from maximum likelihood cross-validation with a standard normal Gaussian kernel is

$$h_{opt} = \max_{h \in H} \left[\frac{1}{N} \sum_{i=1}^{N} \ln \left(\frac{1}{\sqrt{2\pi} h (N-1)} \sum_{j=1, j \neq i}^{N} \exp \left\{ -\frac{1}{2} \left(\frac{\widehat{\epsilon_i} - \widehat{\epsilon_j}}{h} \right)^2 \right\} \right) \right], \quad (2.46)$$

where N is the curve sample size and number of residuals, and H is the set of possible bandwidths where H is in the interval $(0, 10)$ (Shang, 2013). It is known that likelihood cross-validation is poor at estimating fat-tailed distributions (Li and Racine, 2007; Hall, 1987a,b). However, the error density for the simulated data in Chapter 3 will be from a Gaussian distribution which is a thin-tailed distribution, and, therefore, likelihood cross-validation is an appropriate bandwidth selection method. Since the distribution of the error $f(\epsilon)$ for the simulated data is known, the integrated square error (ISE) can be used to evaluate the performance of the error density estimate and is approximated by

$$\begin{aligned} \text{ISE}(\widehat{f}) &= \int_{-5}^{5} [f(\epsilon) - \widehat{f}(\epsilon)]^2 d\epsilon \\ &\approx \frac{10}{n} \sum_{i=1}^{n} \left[f\left(-5 + \frac{i-1}{n/10} \right) - \widehat{f}\left(-5 + \frac{i-1}{n/10} \right) \right]^2, \end{aligned} \quad (2.47)$$

where the density is discretized into $n = 10,000$ equally spaced points. The estimated error density $\widehat{f}(\epsilon_i)$ was calculated at each discrete point with a Gaussian kernel, given

by

$$\begin{aligned}\widehat{f}(\epsilon_i) &= \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{1}{h}\left[\frac{\epsilon_i - \widehat{\epsilon}_j}{\sqrt{\text{var}(\widehat{\epsilon})}}\right]\right) \\ &= \frac{1}{Nh\sqrt{2\pi\text{var}(\widehat{\epsilon})}}\sum_{i=1}^{N}\exp\left\{-\frac{1}{2}\left(\frac{1}{h}\left[\frac{\epsilon_i - \widehat{\epsilon}_j}{\sqrt{\text{var}(\widehat{\epsilon})}}\right]\right)^2\right\}. \qquad (2.48)\end{aligned}$$

This estimator will be referred to as the Shang kernel density estimator (SKDE). The algorithm for the second stage of the two-stage cross-validation procedure is calculated from the residuals as follows:

1. Calculate $M = 100$ Monte Carlo replications of N estimated residuals $\{\widehat{\epsilon}_i^s = y_i^s - \widehat{m}_i^s\}_{i=1,\ldots,N,s=1,\ldots,M}$.
2. Estimate the error density using a univariate kernel estimator.
3. Calculate the ISE using Equation (2.47) for each replication and average over all replications to calculate the mean integrated square error (MISE).

Shang (2013) obtained results using simulations for local cross-validation versus Bayesian methods with $N = 50, 250$, and 1000 curve sample sizes and $0.1, 0.5$, and 0.9 SNRs. For functional regression estimation, they showed that Bayesian and cross-validation bandwidth selection methods performed similarly. For error density estimation, they showed that Bayesian bandwidth selection methods gave a lower MISE for large curve sample sizes than cross-validation. When the methods were applied to the chemometric data set, they found that Bayesian bandwidth selection methods gave a lower MSPE and thus more accurately predicts a response outcome than cross-validation.

2.4.2 Univariate kernel-form density estimation using the np R package

Accurate univariate kernel-form error density estimation is essential for evaluating the performance of functional regression estimation. The **np** R packages has functions for kernel estimation of multivariate continuous and ordered or unordered factor data with many different data-driven bandwidth selection procedures (Hayfield and Racine, 2008) and theoretical and practical support (Li and Racine, 2007). For simulations, the kernel-form density estimator found in Equation (2.44) is used and will henceforth be referred to as the nonparametric kernel density estimator (NPKDE). The (bounded) Epanechnikov kernel function (Silverman, 1998) will be investigated and is defined as

$$K_1(t) = \frac{3}{4\sqrt{5}}\left(1 - \frac{1}{5}t^2\right),\ |t| < \sqrt{5}. \tag{2.49}$$

In this thesis, three different bandwidth selection methods will be explored: a local constant least-squares cross-validated bandwidth, a maximum likelihood least-squares cross-validated bandwidth, and the Bayesian method global plug-in bandwidth. Local constant least-squares cross-validation optimizes bandwidth by minimizing the following criterion (Li and Racine, 2007):

$$CV_f(h) = \frac{1}{n^2 h}\sum_i\sum_j \bar{K}\left(\frac{X_j - X_i}{h}\right) - \frac{2}{n(n-1)h}\sum_{j=1}\sum_{i=1, j\neq i} K\left(\frac{X_j - X_i}{h}\right), \tag{2.50}$$

where $\bar{K}(v) = \int K(u)K(v-u)du$ is a two-fold convolution kernel. Maximum likelihood cross-validation uses the same method described for Equation (2.45). For

comparison, the Bayesian optimized global bandwidth is used as a plug-in bandwidth in the density estimator in Equation (2.44) to evaluate its performance in comparison to cross-validation. To compare the effectiveness of each bandwidth selection method, the mean estimated square error (MESE) of the error density estimator is calculated, as given by

$$\text{MESE}(\widehat{f}) = \frac{1}{n}\sum_{i=1}^{n}\left(f(X_i) - \widehat{f}(X_i)\right)^2, \tag{2.51}$$

where X_i, $i = 1, \ldots, n$ is a sample of the residuals, and the error density function f is known for simulated data. Results from previous research (Shang, 2013) are reproduced in Section 3.1.1 for validation. The effects of the aforementioned bandwidth selection procedures and Epanechnikov density function for error density estimation are investigated in Sections 3.1.2 to 3.1.5. These methods are then applied to the chemometric data set in Section 3.2.

2.5 Methodology summary

The functional nonlinear regression model was selected for smooth data using a semi-metric based on a derivative of the estimated curves. The FNWKE was selected to estimate the functional regression operator, with bandwidth selected using local cross-validation or Bayesian bandwidth selection methods. An algorithm for functional regression and for calculating residuals to be used for error density estimation was described. Two univariate kernel density estimators were described using different bandwidth selection methods, and the Epanechnikov and Gaussian kernel functions.

Chapter 3

Simulation Study

This chapter presents results obtained with the methods and algorithms described in Chapter 2. Bayesian methods are found to outperform cross-validation for error density estimation for simulated data. The (bounded) Epanechnikov kernel is found to perform as well as the Gaussian kernel function for cross-validated and global Bayesian bandwidths. These methods are then applied to the chemometric data set, on which the two-stage cross-validation methods outperform Bayesian methods for regression estimation. All calculations are made using the programming language R (R Core Team, 2014).

All of the following simulations were performed using the high-fructose-corn-syrup machine of the Computer Science Club at the University of Waterloo. The computation times are relative to this machine, where the **parallel** R package (R Core Team, 2014) allows the simultaneous use of the 64 processor cores using parallel computation for simulations. The system specifications are as follows:

- 4× AMD Opteron 6272 (2.1 GHz, 16 cores each, 64 total cores)

- 192 GB RAM with shared memory
- Supermicro H8QGi+-F motherboard quad 1944-pin socket
- 500 GB Seagate Barracuda hard drive
- Supermicro case Rackmount CSE-748TQ-R1400B 4U

3.1 Selecting bandwidths using two-stage cross-validation and Bayesian methods for simulated data

This section presents simulation results, first reproducing results found by SKDE studies (Shang, 2013). Second, results obtained with the error density bandwidth using maximum likelihood cross-validation, least-squares cross-validation, and Bayesian plug-in bandwidth in the NPKDE are presented and compared with results obtained with the SKDE. Last, results obtained with Epanechnikov and Gaussian kernel functions using least-squares cross-validation and Bayesian global plugin bandwidths in the NPKDE are compared.

3.1.1 Simulating functional data and reproducing previous results

Figure 3.1 shows a sample of $N = 250$ simulated explanatory curves from the model in Equation (2.30) in Section 2.2.5, and the simulated scalar response data is calculated using Equation (2.31). The simulated error ϵ in the nonparametric functional regression in Equation (2.2) is generated from a Gaussian distribution described in

the algorithm in Section 2.3 with a 0.1 SNR.

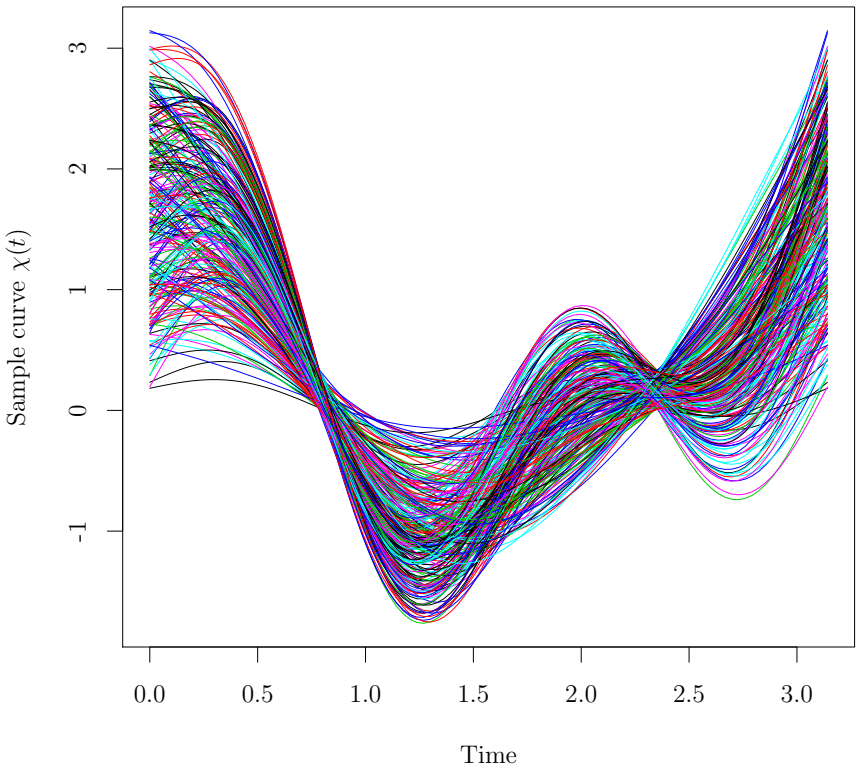

Figure 3.1: Simulation of $N = 250$ sample curves for the model in Equation (2.30)

Table 3.1 shows that the simulation results reproduce previous results (Shang, 2013) using the SKDE in Equation 2.48. Table 3.2 shows the computation-intensive nature of the Bayesian bandwidth selection methods versus the relatively short cross-validation bandwidth selection method.

Table 3.1: Comparison of MISE for Model 1 with SNR 0.1 using the SKDE for 100 Monte Carlo replicates.

Sample Size	Cross-Validation	Bayesian Global	Bayesian Local
50	0.0632	0.0408	0.0388
250	0.0171	0.0094	0.0089
1000	0.0075	0.0030	0.0028

Table 3.2: Comparison of calculation times in minutes for Model 1 with SNR 0.1 using the SKDE for 100 Monte Carlo replicates.

Sample Size	Cross-Validation	Bayesian Global	Bayesian Local
50	0.08	19.00	45.04
250	0.40	425.34	478.79
1000	4.76	8795.88	13329.28

3.1.2 Maximum likelihood and least-squares cross-validated bandwidth selection

This section presents simulation calculations obtained with the maximum likelihood and least-squares cross-validated bandwidth selection methods outlined in Sections 2.4.1 and 2.4.2, respectively, for error density estimation. Table 3.3 presents results that show cross-validated bandwidths from Section 2.4.2 in the NPKDE in Equation (2.4.2) give a smaller MESE than the cross-validation method in Section 2.4.1 for the SKDE in Equation (2.48) for small sample sizes. For larger sample sizes, the SKDE increasingly gives a smaller MESE than the NPKDE. Table 3.4 presents results that show that the Bayesian plug-in bandwidth from Section 2.2.6 in the SKDE in Equation (2.48) gives a smaller MESE than both cross-validation bandwidths in Section 2.4.2 in the NPKDE from Equation (2.44).

Table 3.3: Comparison of MESE of the error density for Model 1 with SNR 0.1: maximum likelihood and least-squares cross-validation (MLCV and LSCV, respectively) with 10 Multistarts and a Gaussian kernel. This table is for two-stage cross-validation bandwidth selection only.

Sample Size	SKDE Two-Staged	NPKDE MLCV	NPKDE LSCV
50	0.0632	0.0611	0.0573
250	0.0171	0.0178	0.0180
1000	0.0075	0.0091	0.0091

Table 3.4: Comparison of MESE of the error density for Model 1 with SNR 0.1: maximum likelihood and least-squares cross-validation (MLCV and LSCV, respectively) with 10 Multistarts and a Gaussian kernel. This table is for Bayesian regression estimation only. The SKDE was calculated with a Bayesian plugin bandwidth and the NPKDE was calculated with two-stage cross-validation.

Sample Size	SKDE Bayesian Global	NPKDE MLCV	NPKDE LSCV
50	0.0408	0.0646	0.0596
250	0.0094	0.0170	0.0171
1000	0.0030	0.0059	0.0058

3.1.3 Bayesian plug-in bandwidth

The Bayesian bandwidth selection method from Section 2.2.6 can be used as a plug-in bandwidth in the NPKDE in Equation (2.44). Table 3.5 shows results obtained with global Bayesian plug-in bandwidth from Section 2.2.6 in the NPKDE from Equation (2.44) in comparison to the SKDE in Equation (2.48). For the FNWKE with a Bayesian global bandwidth, Table 3.5 shows that the Bayesian plug-in bandwidth in the NPKDE gives a smaller MESE than cross-validated bandwidths in the NPKDE from Table 3.4. However, using the Bayesian plug-in bandwidth in the NPKDE shows that it does not perform as well as the Bayesian plug-in bandwidth in the SKDE.

Table 3.5: Comparison of MESE of the error density for Model 1 with SNR 0.1: Bayesian plug-in bandwidth with a Gaussian kernel.

Sample Size	SKDE estimator	NPKDE estimator
50	0.0408	0.0679
250	0.0094	0.0149
1000	0.0030	0.0052

3.1.4 Epanechnikov kernel function

This section presents the effect of using the Epanechnikov kernel function in Equation (2.49) in place of the Gaussian kernel function. Table 3.6 presents results obtained with Gaussian and Epanechnikov kernel functions for two-stage cross-validation, with the second stage being least-squares cross-validation from Section 2.4.2 for the NPKDE in Equation (2.44). The kernel functions in this estimator give very similar MESEs for a cross-validated bandwidth at all sample sizes. For a Bayesian plug-in bandwidth in the NPKDE, Table 3.7 shows that the Epanechnikov kernel function gives very similar MESEs to the Gaussian kernel function at all sample sizes. This shows that there is no apparent difference between using a bounded or unbounded kernel function for error density estimation. These results are to be expected, since the driving force behind sound kernel density estimation is bandwidth selection, not kernel function selection.

Table 3.6: Comparison of MESE of the error density for Model 1 with SNR 0.1: least-squares cross-validation with 10 Multistarts and an Epanechnikov or Gaussian kernel. This is for the two-stage cross-validation only.

Sample Size	Epanechnikov	Gaussian
50	0.0613	0.0611
250	0.0175	0.0178
1000	0.0090	0.0091

Table 3.7: Comparison of MESE of the error density for Model 1 with SNR 0.1: least-squares cross-validation with 10 Multistarts and an Epanechnikov or Gaussian kernel. This is for the global plug-in bandwidth only.

Sample Size	Epanechnikov	Gaussian
50	0.0660	0.0679
250	0.0148	0.0149
1000	0.0051	0.0052

3.1.5 Methodology comparison

The relative performance of each bandwidth selection procedure for regression and error density estimation is shown using comparative box and whisker plots in Figures 3.2 to 3.7 (see Table 3.1.5 for an acronyms reference). Figure 3.2 shows that the SKDE gives smaller estimated square errors (ESEs) than the NPKDE. As the sample size increases in Figures 3.3 and 3.4, all the error densities being estimated from Bayesian global functional regression can be seen to give smaller ESEs than cross-validation functional regression.

Figures 3.5, 3.6, and 3.7 present only error densities produced from Bayesian functional regression estimation. As the sample size increases, it can be seen that the SKDE consistently gives smaller ESEs than the NPKDE. It can also be seen that each method of selecting the bandwidth for the NPKDEs gives very similar results, with Bayesian plug-in bandwidth choosing the more optimal bandwidth than the other methods.

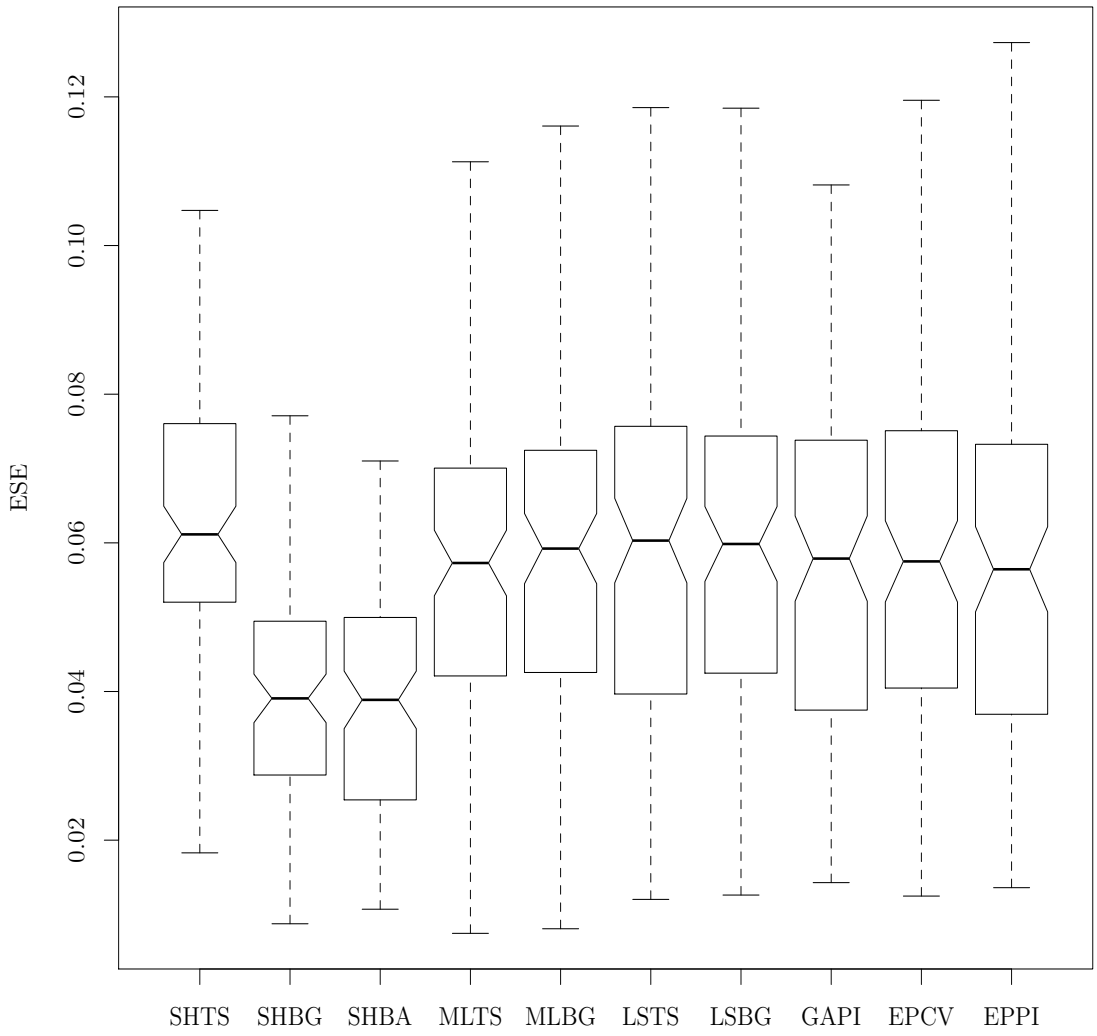

Figure 3.2: Comparison of the estimated squared errors of nonparametric kernel methods for the Bayesian and two-stage cross-validation methods with curve sample size $N = 50$ and $M = 100$ Monte Carlo replicates (refer to Table 3.1.5 for definitions of the acronyms SHTS, etc.).

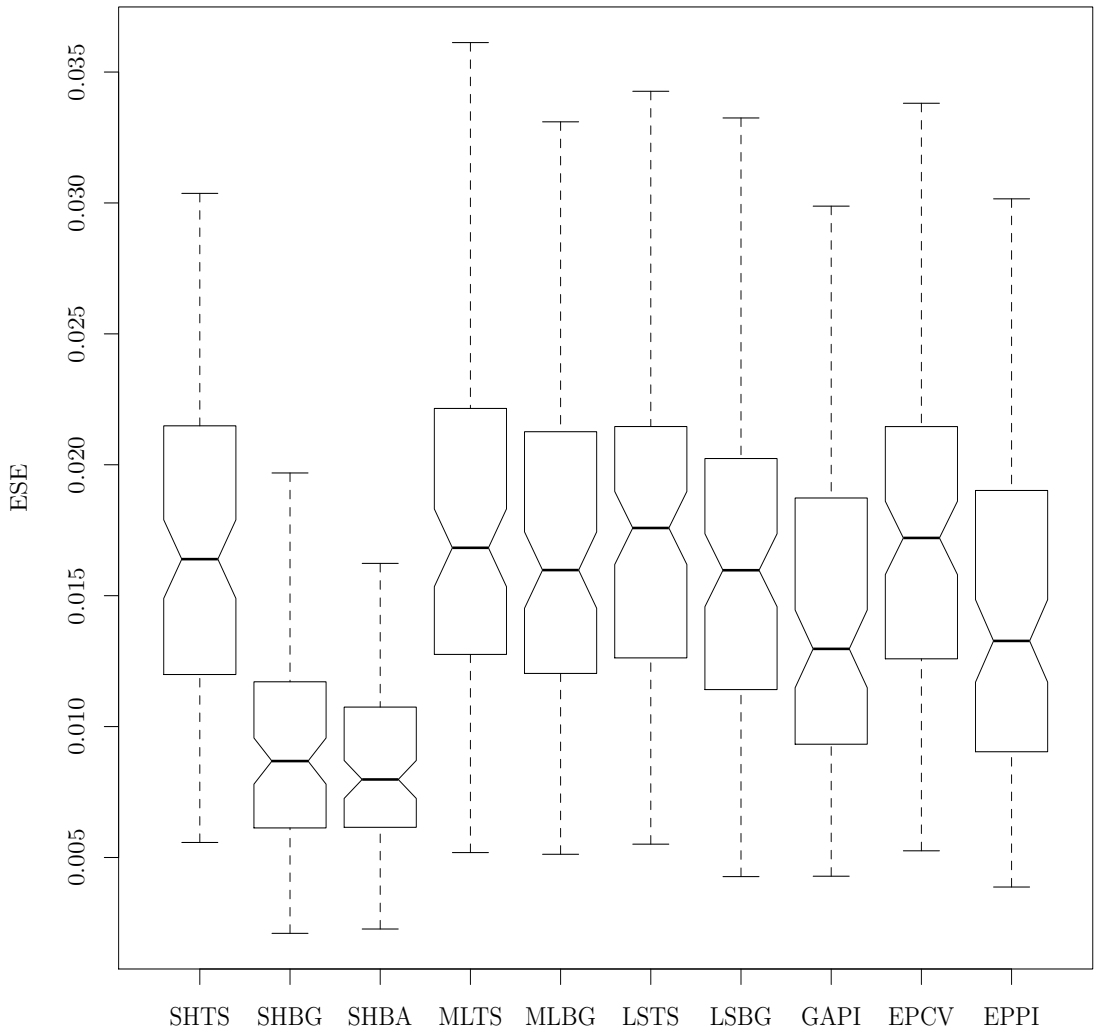

Figure 3.3: Comparison of the estimated squared errors of nonparametric kernel methods for the Bayesian and two-stage cross-validation methods with curve sample size $N = 250$ and $M = 100$ Monte Carlo replicates (refer to Table 3.1.5 for definitions of the acronyms SHTS, etc.).

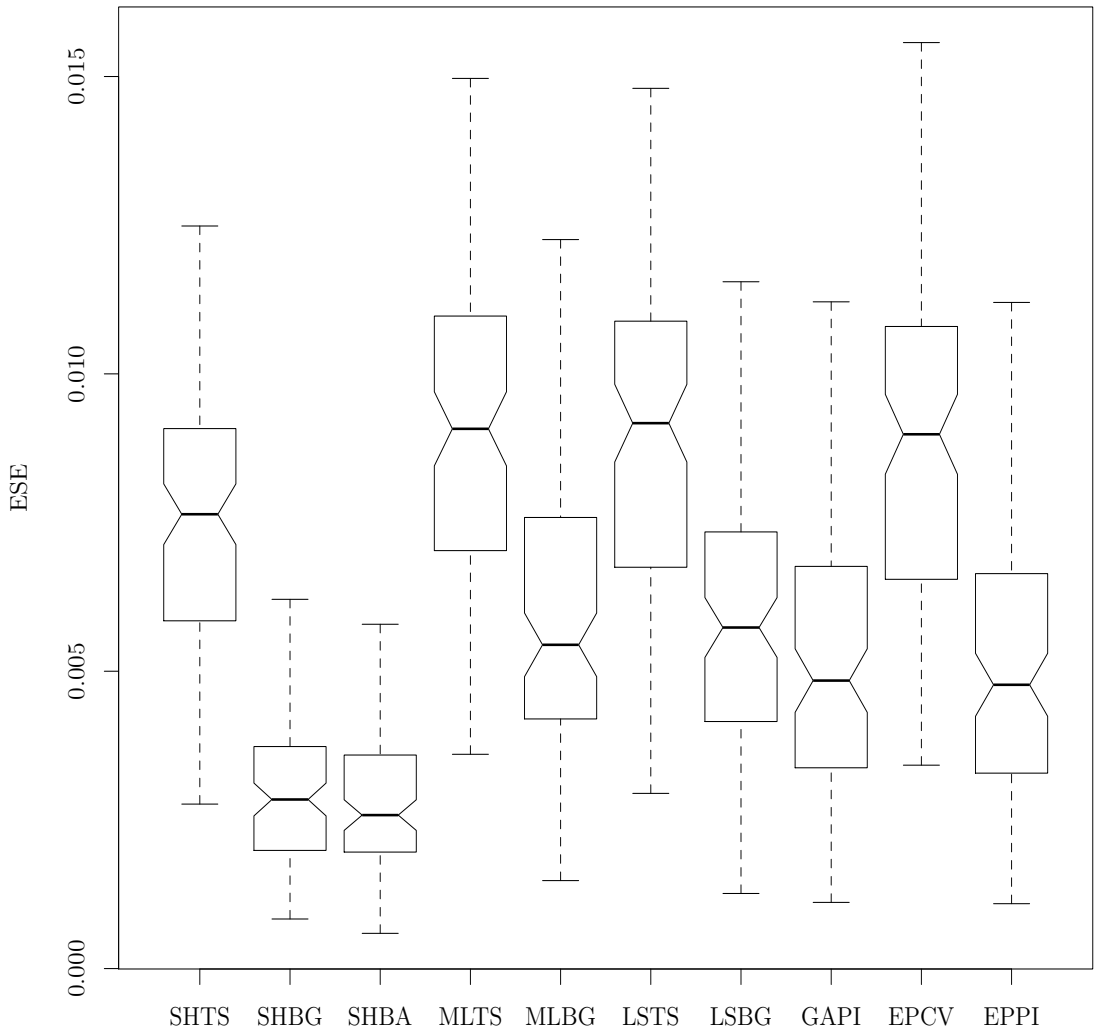

Figure 3.4: Comparison of the estimated squared errors of nonparametric kernel methods for the Bayesian and two-stage cross-validation methods with curve sample size $N = 1000$ and $M = 100$ Monte Carlo replicates (refer to Table 3.1.5 for definitions of the acronyms SHTS, etc.).

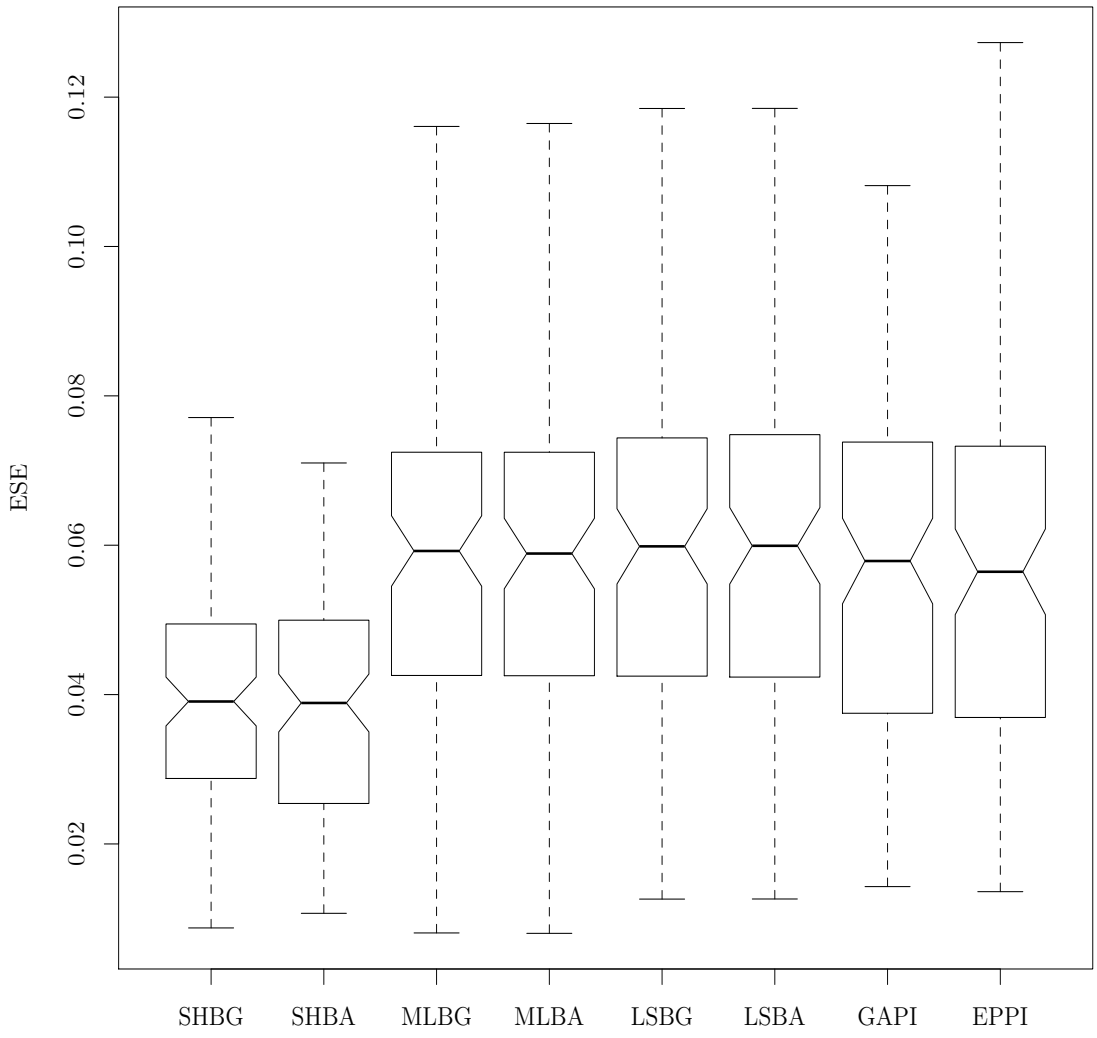

Figure 3.5: Comparison of the estimated squared errors of nonparametric kernel methods for the Bayesian methods only with curve sample size $N = 50$ and $M = 100$ Monte Carlo replicates (refer to Table 3.1.5 for definitions of the acronyms SHBG, etc.).

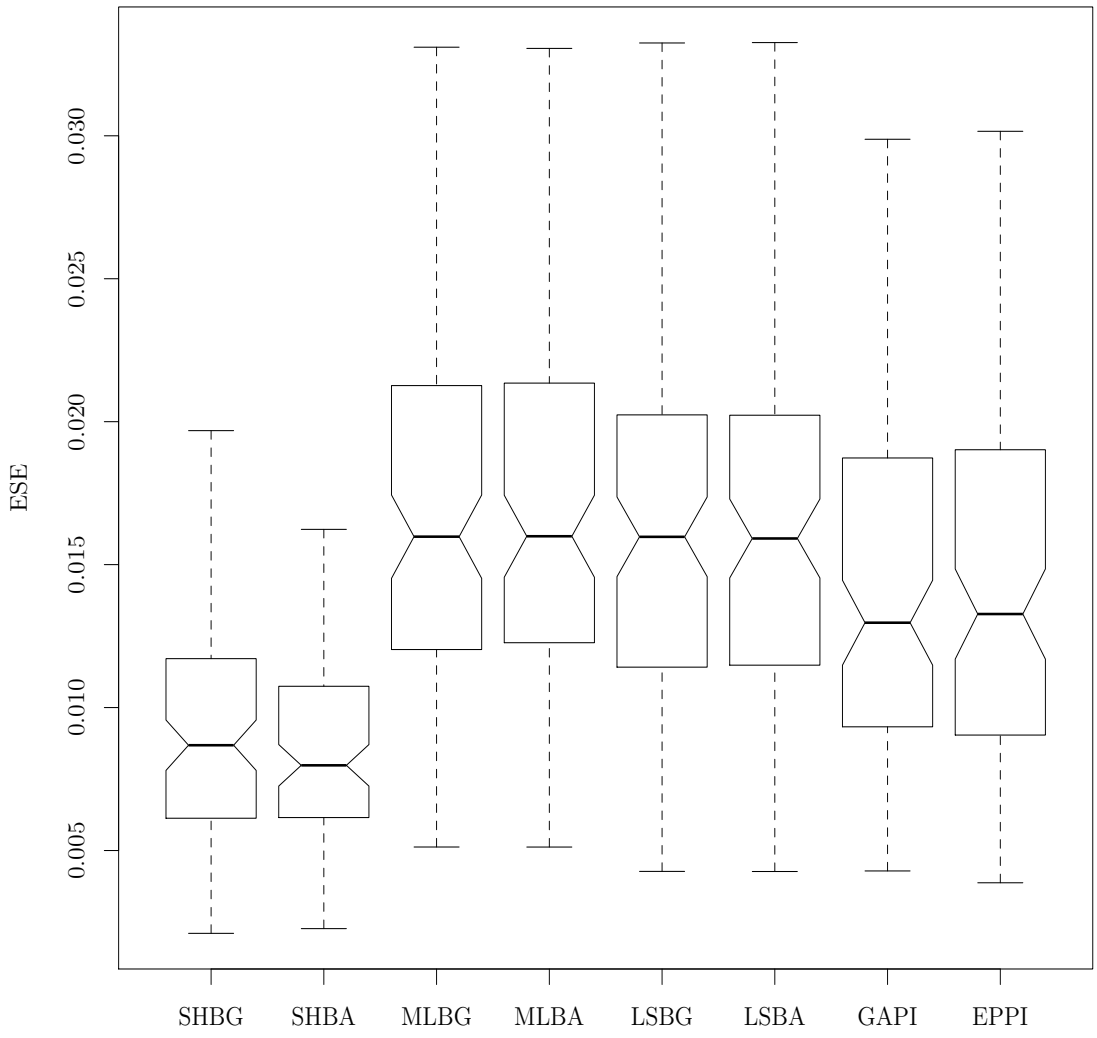

Figure 3.6: Comparison of the estimated squared errors of nonparametric kernel methods for the Bayesian methods only with curve sample size $N = 250$ and $M = 100$ Monte Carlo replicates (refer to Table 3.1.5 for definitions of the acronyms SHBG, etc.).

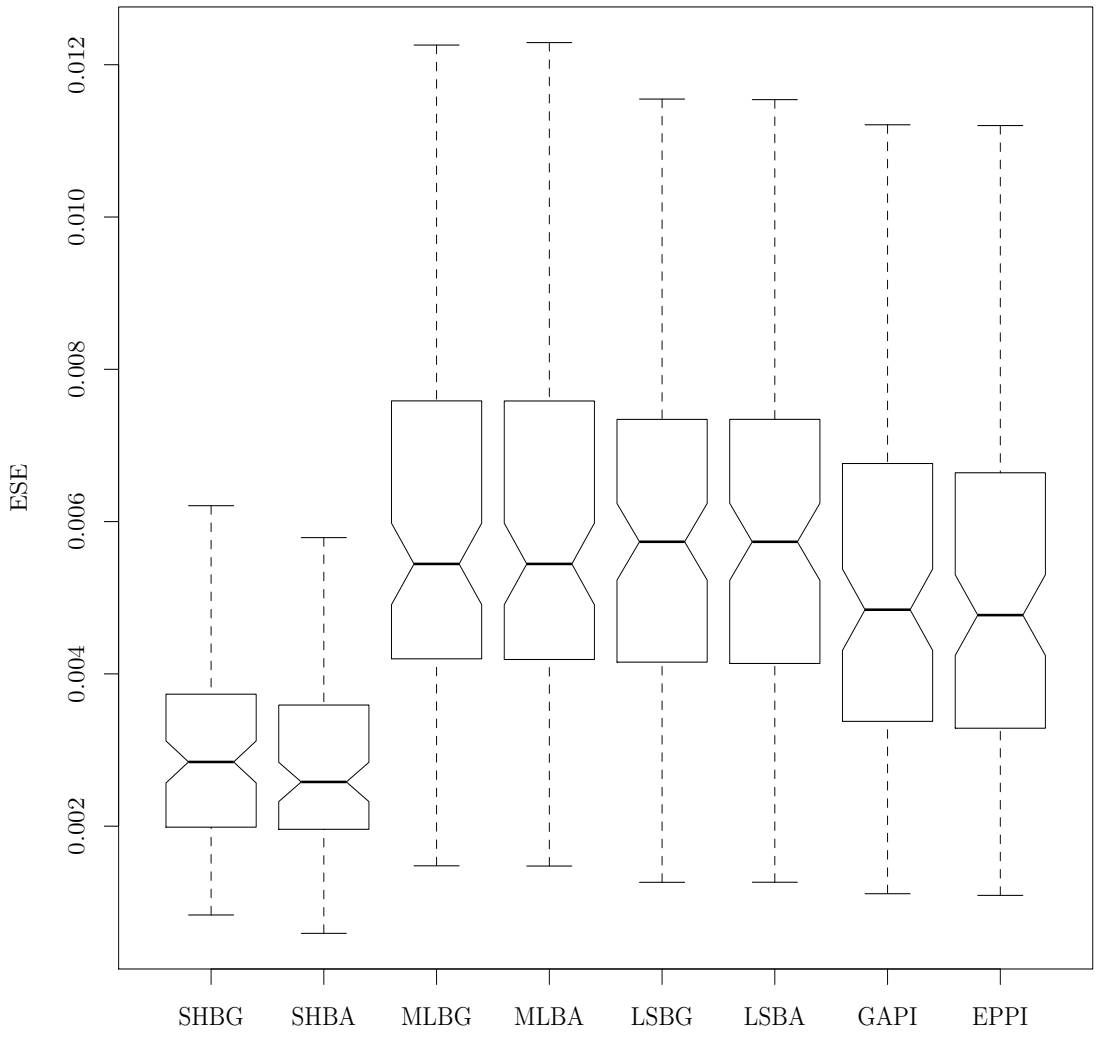

Figure 3.7: Comparison of the estimated squared errors of nonparametric kernel methods for the Bayesian methods only with curve sample size $N = 1000$ and $M = 100$ Monte Carlo replicates (refer to Table 3.1.5 for definitions of the acronyms SHBG, etc.).

Table 3.8: Acronyms for comparative box and whisker plots

Acronym	Bandwidth selection method	Kernel function	Functional regression bandwidth selection method
SHTS	SKDE maximum likelihood cross-validation	Gaussian	Cross-validation
SHBG	SKDE Bayesian global plug-in	Gaussian	Bayesian Global
SHBA	SKDE Bayesian local plug-in	Gaussian	Bayesian Local
MLTS	Maximum likelihood cross-validation	Gaussian	Cross-validation
MLBG	Maximum likelihood cross-validation	Gaussian	Bayesian Global
MLBA	Maximum likelihood cross-validation	Gaussian	Bayesian Local
LSTS	Least-squares cross-validation	Gaussian	Cross-validation
LSBG	Least-squares cross-validation	Gaussian	Bayesian Global
LSBA	Least-squares cross-validation	Gaussian	Bayesian Local
EPCV	Least-squares cross-validation	Epanechnikov	Cross-validation
EPPI	SKDE Bayesian global plug-in	Epanechnikov	Bayesian Global
GAPI	Bayesian global plug-in	Gaussian	Bayesian Global

3.2 Application to the chemometric data set

In this section, functional regression is performed on the chemometric data set using functional cross-validation and Bayesian bandwidth estimation methods, and the error density estimators are applied to the predicted residuals. The chemometric data set *tecator*, described in Section 2.1.3, has 215 paired curves and responses, which are split into a training set of the first $n_1 = 160$ data pairs and an evaluation set of the last $n_2 = 55$ data pairs. This method is used in previous work by Ferraty et al. (2010); Shang (2013), and is extended below to shuffling the data before splitting into two separate sets. Table 3.9 shows that for the original sample, cross-validation gives a smaller MSPE than Bayesian methods for fat, protein, and moisture.

Table 3.9: Comparison of MSPE of the residuals after training each method on the first 160 observations and evaluating on the final 55 observations.

	Fat	Protein	Moisture
Two-Stage CV	5.3679	2.5687	4.2822
Bayesian Global	176.8856	9.3990	103.3482
Bayesian Local	176.8896	9.3990	103.3505

The residuals and estimated error densities for two-stage cross-validation, Bayesian global, and Bayesian local regression estimation are shown in Figures 3.8, 3.9, and 3.10, respectively. Figure 3.8 shows the predicted residuals from predicting the last 55 observations for two-stage cross-validation. In these cases, the SKDE is oversmoothing the data compared to the NPKDE. Figure 3.9 and 3.10 show Bayesian global and local regression estimation residuals. The SKDE is undersmoothing the data in comparison to the NPKDE.

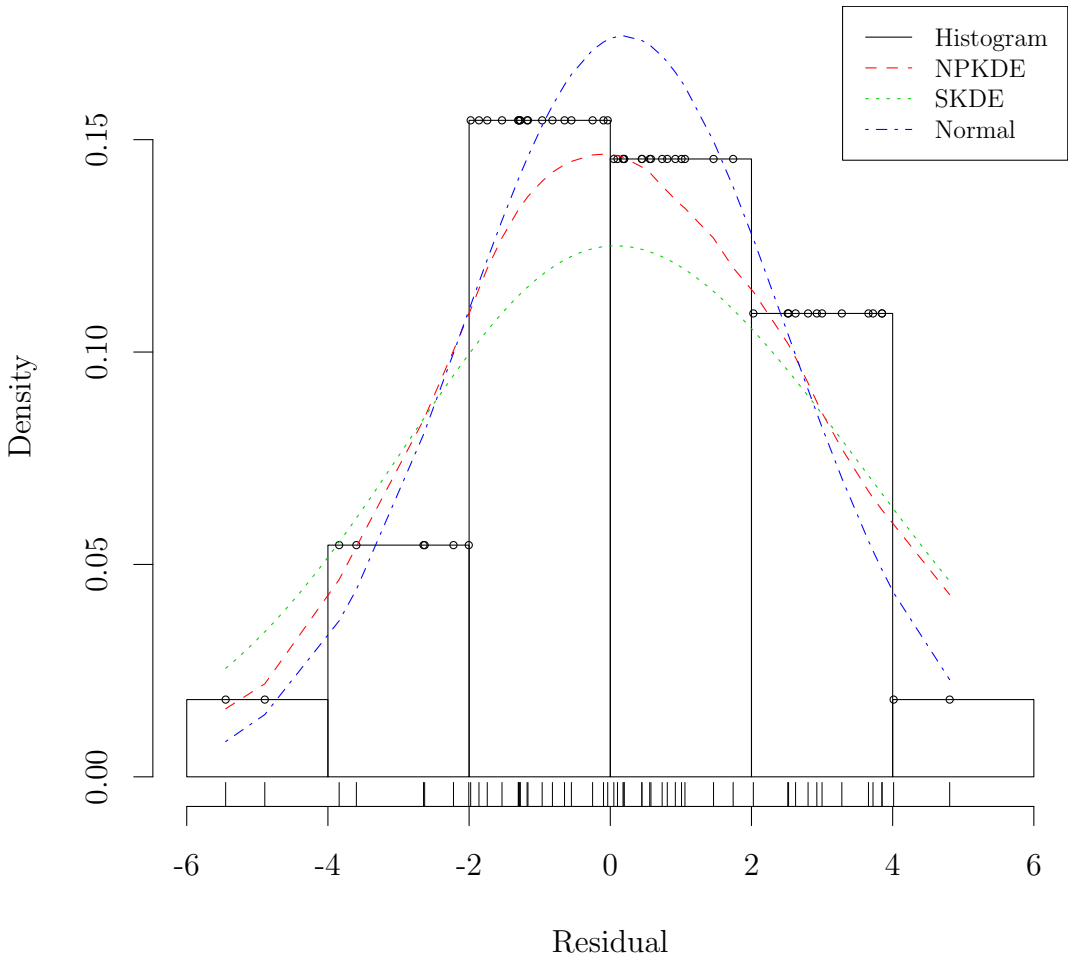

Figure 3.8: Comparison of error density estimators for residuals calculated from the FNWKE with a cross-validated bandwidth. The SKDE is shown in comparison to using the NPKDE with an Epanechnikov kernel and least-squares cross-validated bandwidth. A histogram estimator is used for reference. A standard Gaussian density with maximum likelihood estimates for the mean and standard distribution is used for comparison.

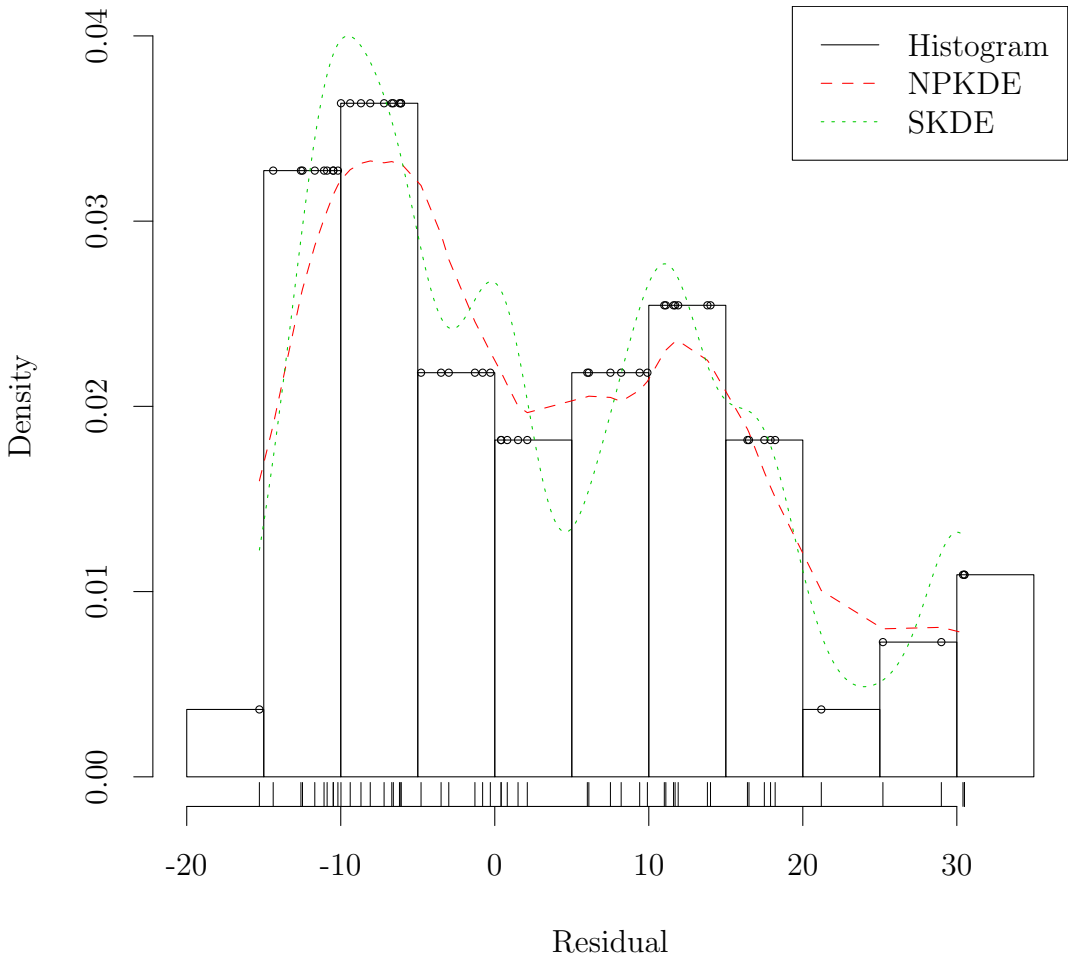

Figure 3.9: Comparison of error density estimators for residuals calculated from the FNWKE with a Bayesian global bandwidth. The SKDE is shown in comparison to using the NPKDE with an Epanechnikov kernel and least-squares cross-validated bandwidth. A histogram estimator is used for reference.

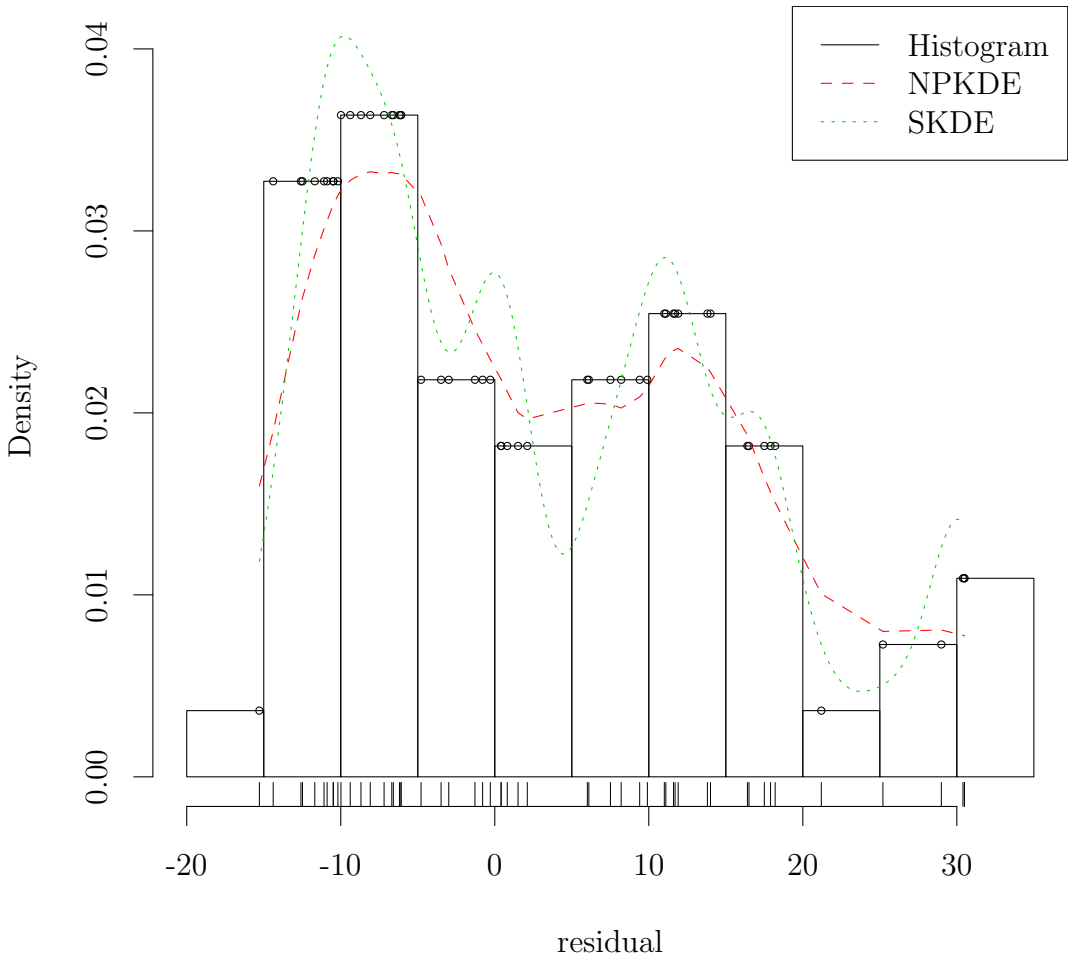

Figure 3.10: Comparison of error density estimators for residuals calculated from the FNWKE with a Bayesian local bandwidth. The SKDE is shown in comparison to using the NPKDE with an Epanechnikov kernel and least-squares cross-validated bandwidth. A histogram estimator is used for reference.

3.2.1 Shuffled samples

To show that this is not just a single sample result, the data are shuffled 100 times and for each shuffle, split into a training set of the first $n_1 = 160$ data pairs and an evaluation set of the last $n_2 = 55$ data pairs. Averaging over the shuffles, Table 3.10 shows that Bayesian methods have a consistently higher MSPE than cross-validation for estimating the regression operator.

Table 3.10: Comparison of MSPE of the residuals after shuffling the data 100 times. The regression estimator for each method is trained on the first 160 observations and evaluated on the final 55 observations per shuffle.

	Fat	Protein	Moisture
Two-Stage CV	7.0864	2.7086	5.7874
Bayesian Global	160.9720	9.0352	97.1804
Bayesian Local	160.9722	9.0352	97.1804

Chapter 4

Conclusions and Recommendations

For simulated data, Bayesian plug-in bandwidth selection methods in the SKDE have been shown to best estimate the error density. For the NPKDE, Bayesian global bandwidth selection methods for error density estimation have been shown to give a slightly smaller MESE than cross-validation methods.

For all sample sizes, the (bounded) Epanechnikov and Gaussian kernel functions have been shown to give a similar MESE for both two-stage cross-validation and Bayesian global plug-in bandwidths in the NPKDE. This was expected as the bandwidth selection method is a more significant factor than kernel selection in accurate density estimation.

For the FNWKE calculated with Bayesian compared to two-stage cross-validation methods, the significant reduction in the MESE for error densities is likely caused by a better estimate of the regression operator, and by not the bandwidth selection method for the error density estimator. However, there is no need to re-select the error density bandwidth using cross-validation after Bayesian functional regression since it simultaneously calculates functional regression and error density bandwidths.

For the chemometric data set, estimating the FNWKE using cross-validated bandwidths was shown to give a significantly lower MSPE than using Bayesian bandwidths. The same Bayesian bandwidth estimation model was used for simulated and real data. The Bayesian bandwidth selection method was shown to be adequate for simulated data, but may have not chosen the best prior distributions for the bandwidths that cause the functional estimates to be inadequate for this real data set.

The small number of Monte Carlo replicates and curve samples, due to the computationally intensive nature of the Bayesian method, do not allow strong conclusions to be drawn from results presented in this thesis. To illustrate how time consuming each method is, consider a sample of 50 curves that has $50 \times 100 = 5,000$ sample points and a sample of 1000 curves that has 100,000 sample points. A core minute/day/year/etc. is the total time a single core would take to run a simulation. The local Bayesian method took approximately 1.6 core days or 22.5 minutes on one core for one replication for 50 curves, and 66 core weeks or 2.4 seconds on one core for one replication for 1000 curves. The cross-validation method took approximately 4 core minutes or 2.4 seconds on one core for one replication for 50 curves, and approximately 4.0 core hours or 2.4 minutes on one core for one replication for 1000 curves. For a simulation of 1000 Monte Carlo replicates, 1000 curves for Bayesian methods could take more than one core year of computation time. For the same simulation, cross-validation methods would take approximately 1.6 core days. This implies that while Bayesian methods have better performance for estimating functional regression and error density, the computational trade-off is significant. The choice is whether or not to invest computation time into calculating Bayesian methods. Each calculation is made using an Opteron core processor, where a more powerful core processor will likely lead to

shorter computation time.

4.1 Recommendations for future work

Based on the results presented in this thesis, some recommendations arise for future work:

- Advances in the asymptotics of local linear cross-validation for functional regression have been recently developed (Rachdi et al., 2014) and would be an interesting comparison to Bayesian bandwidth selection methods.
- Another aspect of functional nonparametric methods is the optimal knot placement for B-splines. Little discussion on knot placement for functional regression was found in the course of a literature review. Simultaneous optimization for B-spline basis, knot placement, and bandwidth for functional regression seems to be an area deserving of attention.
- Theoretical and applied maximization of the kernel likelihood using kernel methods directly is another area deserving of attention and worthy of future investigation.

Other aspects that could usefully be addressed include:

- Investigating the kNN bandwidth estimation for error density estimation (Li and Racine, 2007) of the residuals from functional regression.
- Extending the simulations of this thesis to different SNRs.

- Exploring larger sample sizes (> 1000 curve samples) and Monte Carlo simulations (> 100 replicates) in both functional regression and real-valued error density estimation for all bandwidth estimation methods.
- Exploring different density estimators for the error density (Shang, 2013) such as iterative methods (Müller and Wang, 1990; Jones et al., 1991).
- Investigating Bayesian methods for simulated data where the simulated error is non-Gaussian.
- Using different symmetric kernel functions for error density estimation in the Bayesian MCMC method such as the (bounded) Epanechnikov kernel.

Appendix A

R packages and Functions for Functional Data Analysis

This appendix outlines the different packages and functions used in R for calculations. The R script files for all calculations in this thesis are available upon request.

A.1 The fda R package

The **fda** R package is the starter FDA package for smooth functional data with theoretical (Ramsay and Silverman, 2005) and practical support (Ramsay et al., 2009). It is an excellent starting point for experimenting with the functional linear regression model for FDA. There are functional data applications, with practical examples, to introduce the use of the package to beginner practitioners, including how to specify a basis, build functional data objects, apply smoothing methods, present analysis methods for different kinds of functional data, and work with curve registration and regression analysis.

A.2 The fda.usc R package: functions for the fda package

The package **fda.usc** for statistical computing in FDA (Febrero-Bande and Oviedo de la Fuente, 2012) has functions for nonparametric functional regression and basis representation for the **fda** R package. There are functions that come naturally from nonparametric functional regression such as functional linear models and semi-functional partial linear models. The package provides methods for exploring functional data, conducting analysis, curve outlier detection, functional analysis of variance, and more. This package also contains the *tecator* data set shown in Figure 2.1.

A.3 The "npfda" R functions

These functions are provided on the website `http://www.math.univ-toulouse.fr/staph/npfda/` with theoretical (Ferraty and Vieu, 2006) and practical (`http://www.math.univ-toulouse.fr/~ferraty/SOFTWARES/NPFDA/npfda-help-files.pdf`) support. Many useful routines are included, such as calculating semi-metrics between curves, determining optimal bandwidth, and using different regression techniques in the functional setting. These functions were used to calculate the optimal bandwidths for functional regression using cross-validation and all FNWKEs. Note that this is not a formal package and is not maintained on the CRAN website.

A.4 The np R package: nonparametric kernel density estimation for mixed data types

The **np** package (Hayfield and Racine, 2008) is used for nonparametric kernel density estimation during simulations. This package provides many functions for calculating the nonparametric kernel smoothing methods for continuous and ordered or unordered discrete data types with theoretical support (see Li and Racine, 2007). The kernel smoothing methods include univariate density estimation for continuous and discrete data types, mixed data type multivariate regression, mixed type conditional density estimation, and so on.

A.5 The parallel R package: parallel computation

The **parallel** R package (R Core Team, 2014) is used for parallel computation using R on computers with shared memory between cores. Since R is designed to use only one core per R session, this package uses different methods, such as spawning or forking, to allow for simultaneous multi-core usage. Parallel computation is based around the idea of a master that communicates jobs to workers who all perform the same calculation, with parameters as instructed by the master. This is useful for large simulations for computers with a large number of cores. There exist other packages for communication between computers that do not have shared memory, such as **Rmpi** (Yu, 2002) for R and **npRmpi** (Hayfield and Racine, 2008) for the **np** R package.

Bibliography

Akritas, M. and Van Keilegom, I. (2001). Non-parametric estimation of the residual distribution. *Scandinavian Journal of Statistics*, **28**(3), 549–567.

Benhenni, K., Ferraty, F., Rachdi, M., and Vieu, P. (2007). Local smoothing regression with functional data. *Computational Statistics*, **22**, 353–369.

Borggaard, C. and Thodberg, H. (1992). Optimal minimal neural interpretation of spectra. *Analytical Chemistry*, **64**, 545–551.

Burba, F., Ferraty, F., and Vieu, P. (2009). k-nearest neighbour method in functional nonparametric regression. *Journal of Nonparametric Statistics*, **21**(4), 453–469.

Casella, G. and Berger, R. (2002). *Statistical Inference*. Thomson Learning.

de Boor, C. (1978). *A Practical Guide to Splines*. New York: Springer.

Efromovich, S. (2005). Estimation of the density of regression errors. *The Annals of Statistics*, **33**(5), 2194–2227.

Escanciano, J. and Jacho-Chávez, D. (2012). $\sqrt{n}$ uniformly consistent density estimation in nonparametric regression models. *Journal of Econometrics*, **167**(2), 305–316.

Fan, Y. and James, G. (2013). Functional additive regression. Working paper. URL: `http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/`.

Febrero-Bande, M. and González-Manteiga, W. (2013). Generalized additive models for functional data. *Test*, **22**(2), 278–292.

Febrero-Bande, M. and Oviedo de la Fuente, M. (2012). Statistical computing in functional data analysis: The R package fda.usc. *Journal of Statistical Software*, **51**, 757–796.

Ferraty, F. and Vieu, P. (2006). *Nonparametric Functional Analysis*. Springer-Verlag.

Ferraty, F., Mas, A., and Vieu, P. (2008). Nonparametric regression on functional data: inference and practical aspects. *Austrailian and New Zealand Journal of Statistics*, **49**(3), 267–286.

Ferraty, F., Van Keilegom, I., and Vieu, P. (2010). On the validity of the bootstrap in non-parametric functional regression. *Scandinavian Journal of Statistics*, **37**(2), 286–306.

Garthwaite, P., Fan, Y., and Sisson, S. (2010). Adaptive optimal scaling of metropolis-hastings algorithms using the Robbins-Monro process. Working paper. University of New South Wales.

Geweke, J. (2010). *Complete and Incomplete Econometric Models*. Princeton, NJ: Princeton University Press.

Hall, P. (1987a). On Kullback-Leibler loss and density estimation. *The Annals of Statistics*, **15**, 1491–1519.

Hall, P. (1987b). On the use of compactly supported density estimates in the problems of discrimination. *Journal of Multivariate Analysis*, **23**, 131–158.

Härdle, W. (1989). Resampling for inference from curves. *Proceedings of the 47th Session of the International Statistical Institute Bulletin de l'Institut International de Statistique*, **53**(3), 53–64.

Härdle, W. and Marron, J. (1985). Optimal bandwidth selection in nonparametric regression function estimation. *The Annals of Statistics*, **13**(4), 1465–1481.

Härdle, W. and Marron, J. (1991). Bootstrap simultaneous error bars for nonparametric regression. *The Annals of Statistics*, **16**, 1696–1708.

Hastie, T. and Tibshirani, R. (1993). Varying-coefficient models (with discussion). *Journal of the Royal Statistical Society, Series B*, **55**, 757–796.

Hayfield, T. and Racine, J. (2008). Nonparametric econometrics: The **np** package. *Journal of Statistical Software*, **27**(5).

Helland, I. (1990). PLS regression and statistical models. *Scandinavian Journal of Statistics*, **17**, 97–114.

Horváth, L. and Reeder, R. (2012). A test of significance in functional quadratic regression. In L. Horváth, P. Kokoszka (Eds.). *Inference for Functional Data with Applications.* Springer, pages 225–232.

Jaki, T. and West, R. (2008). Maximum kernel likelihood estimation. *Journal of Computational and Graphical Statistics*, **17**(4), 976–993.

Jaki, T. and West, R. (2011). Symmetric maximum kernel likelihood estimation. *Journal of Statistical Computation and Simulation*, **81**(2), 193–206.

Jones, M., Marron, J., and Park, B. (1991). A simple root-n bandwidth selector. *The Annals of Statistics*, **19**(4), 1919–1932.

Li, Q. and Racine, J. (2007). *Nonparametric Econometrics*. Princeton, NJ: Princeton University Press.

Muhsal, B. and Neumeyer, N. (2010). A note on residual-based empirical likelihood kernel density estimation. *Electronic Journal of Statistics*, **4**, 1386–1401.

Müller, H.-G. and Wang, J. (1990). Locally adaptive hazard smoothing. *Probability Theory and Related Fields*, **85**(4), 523–538.

Müller, H.-G. and Yao, F. (2008). Functional additive models. *Journal of the American Statistical Association*, **103**(484), 1534–1544.

Nadaraya, N. (1964). On estimating regression. *Theory of Probability and its Application*, **9**, 141–142.

Neumeyer, N. and Dette, H. (2007). Testing for symmetric error distribution in nonparametric regression models. *Statistica Sinica*, **17**(2), 775–795.

Parzen, E. (1992). On the estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, **33**(3), 1065–1076.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rachdi, M., Laksaci, A., Demongeot, J., Abdali, A., and Madani, F. (2014). Theoretical and practical aspects of the quadratic error in the local linear estimaiton of the conditional density for functional data. *Computational Statistics and Data Analysis*, **73**, 53–68.

Ramsay, J. and Silverman, B. (2005). *Functional Data Analysis,* 2nd ed. New York: Springer.

Ramsay, J., Hooker, G., and Graves, S. (2009). *Functional Data Analysis with R and MATLAB*. New York: Springer-Science.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, **22**(3), 400–407.

Roberts, G. and Rosenthal, J. (2009). Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, **18**(2), 349–367.

Samb, R. (2011). Nonparametric estimation of the density of regression errors. *Comptes Rendus - Mathematique*, **349**(23-24), 1281–1324.

Schumaker, M. (1981). *Spline Functions: Basic Theory*. Cambridge, UK: Cambridge University Press.

Shang, H. (2013). Bayesian bandwidth estimation for a nonparametric functional regression model with unknown error density. *Computational Statistics and Data Analysis*, **67**, 185–198.

Silverman, B. (1998). *Density Estimation for Statistics and Data Analysis*. Boca Raton, FL: Chapman and Hall.

Watson, G. (1964). Smooth regression analysis. *Sankhya Series A*, **26**, 359–372.

Yao, F. and Müller, H.-G. (2010). Functional quadratic regression. *Biometrika*, **97**(1), 49–64.

Yu, H. (2002). Rmpi: Parallel statistical computing in R. *R News*, **2**(2), 10–14.