FUNCTIONAL DATA ANALYSIS WITH APPLICATION TO TRAFFIC FLOW DATA

By

Yi-Chen Zhang

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Statistics – Doctor of Philosophy

2018

ABSTRACT

FUNCTIONAL DATA ANALYSIS WITH APPLICATION TO TRAFFIC FLOW DATA

By

Yi-Chen Zhang

Functional data has become increasingly popular in the recent statistical literature. Considerable attention has been paid to the development of functional data analysis. This thesis consists of four main chapters to address some important questions that arise from implementing FPCA in practice and to give answer to these questions. In Chapter 2, we investigate the problem of data preprocessing for functional data. We propose and analyzes a nonparametric functional data approach to missing value imputation and outlier detection for functional data. In Chapter 3, a functional naive Bayes classifier has been proposed for functional data which provides a surrogate density estimation for functional random variables that makes a direct extension of density-based classical multivariate classification approaches to functional data classification possible. In Chapter 4, we merge two ideas of functional classification and functional prediction to develop a dynamical prediction for functional data. The proposed functional mixture prediction approach combines functional linear model with functional naive Bayes classifier. In Chapter 5, we suggest a two-step segmentation procedure to estimate both the number and locations of the mean change-points of a functional sequence. Finally, the thesis concludes with a brief discussion of future research directions.

Keywords: Functional data analysis; missing values imputation; outlier detection; functional classification; naive Bayes classifier; functional prediction; functional change-points

This thesis is dedicated to my wonderful wife and loving parents, the hidden strength behind my every success.

ACKNOWLEDGEMENTS

I would like to begin by thanking my advisor Dr. Lyudmila Sakhanenko. I wish to express my deepest gratitude to her. Thanks to her guidance, patience, and thoughtfulness, this work has been made possible. Thank you.

I wish to extend my sincere gratitude to my thesis committee members Drs. David Zhu, Ping-Shou Zhong, and Yuying Xie for their time and interest. Thanks for Dr. Ping-Shou Zhong for his comments and suggestions on the research work in Chapter 3 and Dr. David Zhu for his data and discussion on the future work in Chapter 6.

My sincere thanks go to the staff at the Department of Statistics and Probability, whose kindness provided a very enjoyable study environment. Also, I appreciate the help and support of the Department of Statistics and Probability in the last five years.

Most importantly, I would like to thank Yi-Ru for being a supportive, patient, and a wonderful wife through this entire journey. Her positive attitude and encouragement have always been an inspiration, and give me the much needed strength and confidence, especially during the last stages of my Ph.D training.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

With the progress of technology, data arising in a wide range of fields are often obtained in a form of functions. "Functional data" is a term that refers to data which are be recorded continuously during a time interval, whose graphical representations can be curves, images, shapes, or general objects that vary over time. Although the analysis of functional data and that of multivariate data share many common principles, the infinite-dimensional nature of functional data presents many new challenges that are absent in the traditional multivariate analysis. Functional data analysis (FDA) is an emerging field in statistics that has seen rapid development over the last two decades. As a new branch of statistics, FDA extends existing methodologies and theories from the areas of multivariate data analysis, stochastic processes, generalized linear models, and many others. The book by Ramsay and Silverman (2005) gives a clear account of the basic considerations of FDA, and the book of Ferraty and Vieu (2006) provides a detailed survey of many nonparametric techniques for analyzing functional data.

The term FDA can be dated to Ramsay (1982) and Ramsay and Dalzell (1991). From the 1990s onward, with the development of technology, FDA became a fast growing area of statistics. There are several motivations for studying functional data. In many research fields such as traffic flow in transportation, signal in radar range, weather in meteorology, gene expression in genomics, growth curve in medicine, and brain image in functional magnetic resonance imaging (fMRI), the data generating process is naturally represented in terms of functions. Thus, many practical problems are better approached with the data described as functions. In a conceptual sense, functions are intrinsically infinite-dimensional but are usually measured discretely. Different from multivariate data analysis, FDA treats datum as the continuity of the curves and models the data in the functional space rather than treating them as a set of vectors. With the high intrinsic dimensionality of these data, a

standard multivariate data analysis might not be computationally feasible due to the "curse of dimensionality". Moreover, it should be emphasized that the datum in FDA is a whole function defined on some interval, contrary to simply focusing on the observed value at a particular point in the interval.

Under the FDA framework each sample element is considered to be a curve or a function instead of a finite set of single data points. This has potential advantage to analyzing discrete data as it has fewer assumptions than classical multivariate analysis. Functional data are assumed to be smooth functions that have been measured on a dense grid. One of the most popular tools for FDA is functional principal component analysis (FPCA), which is built on the Karhunen-Loève expansion for stochastic processes and can reduce the random functions to a set of functional principal component (FPC) scores. This thesis focuses extensively on the FPC scores to carry out secondary development for analyzing traffic flow data.

1.1 Motivation

The purpose of this thesis is to address some important questions that arise from implementing FPCA in practice and to give answers to these questions. This topic was chosen after conducting an extensive literature review of functional data analysis. This thesis studies the tasks including:

- Data preprocessing for functional data which proposes and analyzes a nonparametric functional data approach to missing value imputation and outlier detection for functional data;

- The naive Bayes classifier for functional data which provides a surrogate density estimation for functional random variables that makes a direct extension of density-based classical multivariate classification approaches to functional data classification possible;

- Dynamical prediction for functional data which proposes a functional mixture prediction approach that combines functional linear model with functional naive Bayes

classifier;

- Change-points in functional sequence which offers a two-step segmentation procedure to estimate both the number and locations of the mean change-points of a functional sequence.

To illustrate the practical use of our techniques, all of the above topics are implemented on the traffic flow data.

Traffic flow data provide valuable information for highway planning, traffic surveillance, and control purposes. For example, the information of traffic flow is useful to estimate the design-hour volume and annual average daily traffic. In addition, real-time traffic flow data provide essential information for traffic surveillance and control in Intelligent Transportation Systems (ITS).

Applications of traffic monitoring require complete and reliable data. These data can be recorded automatically by various types of vehicle loop detectors, which are usually installed under a planned road at regular intervals. Since loop detectors operate at a rough environment, missing data problems are inevitable due to detector malfunctions or package loss during transmission. Therefore, temporary detector malfunctions that result in loss of data are quite common.

Besides, outlier detection is another important issue in investigating traffic data. These include detecting temporal outliers in terms of magnitude in time and identifying unusual patterns of trajectories, both of which provide useful information for further applications to traffic management.

Classification of the traffic flow is also an important topic for management in ITS. Since the traffic flow can be viewed as a macroscopic traffic characteristic in transportation system, a good classification rule can be very helpful to build a better traffic control strategy.

Furthermore, it is hard not to mention the importance of traffic prediction for ITS. The prediction of traffic rates has long been recognized in many applications. Real-time

forecasting gives travelers the ability to choose better routes, and while it gives authorities the ability to manage the transportation system.

Moreover, change-point problem is also an issue for traffic management. The pattern of traffic flow rate from the downstream or upstream detectors should be roughly similar. An abrupt change indicates something unusual.

1.2 Functional Principal Component Analysis

In the functional data framework, we adopt the notion that each daily traffic flow trajectory is a realization of a random function sampled from a stochastic process. Let X be a random function for the daily traffic flow trajectory in the domain $\mathcal{I} = [0, T]$. We note that the random function X is a square integrable function, that is, $\int_{\mathcal{I}} E(X^2) < \infty$, such that $X \in L_2(\mathcal{I})$. Here, the $L_2(\mathcal{I})$ is the class of random functions with the inner product of any two functions f and g defined as $\langle f, g\rangle = \int_{\mathcal{I}} f(t)g(t)dt$ with the norm $\|f\| = \langle f, f\rangle^{1/2}$. We also assume that the random function X has a smooth mean function μ such that $\mu(t) := E(X(t))$ and the covariance function of X is defined to be the function Γ such that

$$\Gamma(s,t) := cov(X(s), X(t)) = E((X(s) - \mu(s))(X(t) - \mu(t)))$$

for s and t in $\mathcal{I}$. We further assume that the covariance function is continuous and square-integrable, that is, $\int\int \Gamma^2(s,t)dsdt < \infty$. Then the function Γ induces the kernel operator $\Gamma : L_2(\mathcal{I}) \to L_2(\mathcal{I})$, defined by

$$(\Gamma\phi)(s) = \int_{\mathcal{I}} \Gamma(s,t)\phi(t)dt.$$

As noted, FPCA relies on an expansion in terms of the eigenbasis of the covariance function Γ. The existence of an eigenbasis of Γ for $L_2(\mathcal{I})$ is guaranteed by Mercer's lemma, and the expansion of X in this basis is termed the Karhunen-Loève expansion.

Lemma 1. (Mercer's lemma) *Assume that the covariance function Γ as defined is continuous over $\mathcal{I}^2$. Then there exist an orthonormal sequence $\{\phi_j\}$ of continuous function in*

$L_2(\mathcal{I})$, and a non-increasing sequence $\{\lambda_j\}$ of positive numbers, such that

$$(\Gamma\phi_j)(t) = \lambda_j\phi_j(t), \quad t \in \mathcal{I},\ j \in \mathbb{N},$$

and moreover,

$$\Gamma(s,t) = \sum_{j=1}^{\infty} \lambda_j\phi_j(s)\phi_j(t), \quad s,t \in \mathcal{I},$$

where the series converges uniformly on $\mathcal{I}^2$. Hence

$$\sum_{j=1}^{\infty} \lambda_j = \int_{\mathcal{I}} \Gamma(s,s)ds < \infty.$$

The proof of Mercer's lemma can be essentially found in Mercer (1909).

Theorem 1. (Karhunnen-Loève expansion) *Under the assumptions and notations of Mercer's lemma, we have*

$$X(t) = \mu(t) + \sum_{j=1}^{\infty} \xi_j\phi_j(t), \tag{1.1}$$

where $\xi_j = \langle X - \mu, \phi_j\rangle$ is a random variable with $E(\xi_j) = 0$, and $E(\xi_j\xi_k) = \delta_{jk}$, $j,k \in \mathbb{N}$. The δ_{jk} denotes the Kronecker delta. The series (1.1) *converges uniformly on $\mathcal{I}$ with respect to the L_2-norm.*

In equation (1.1), the eigenfunctions ϕ_j are referred to as functional principal components (FPC) with FPC scores ξ_j. The deviation of each sample function from the mean is thus represented as a sum of orthogonal curves with uncorrelated random coefficients. Although the expansion in equation (1.1) is infinite dimensional, it is a common practical experience that the first leading M eigenfunctions can effectively span the process, for an $M < \infty$. In practical applications, this M must be chosen data-adaptively and will be discussed in each chapters for a different purpose. The idea of FPCA is to retain the first M terms in the Karhunen-Loève expansion as an approximation to X

$$X(t) \approx \mu(t) + \sum_{j=1}^{M} \xi_j\phi_j(t) \tag{1.2}$$

and hence to achieve dimension reduction. This can be seen as projecting X onto an M-dimensional space spanned by the first M eigenfunctions with the largest eigenvalues λ_j.

Given the discrete observations $\{(t_{ij}, X_i(t_{ij})),\, i = 1, \cdots, n,\, j = 1, \cdots, m_i\}$ the estimate $\hat{\mu}$ of mean function μ can be estimated by applying the locally weighted least squares method while the estimates $\hat{\phi}_j$ and $\hat{\xi}_{ij}$ of the components ϕ_j and ξ_{ij} rely on the covariance estimate $\widehat{\Gamma}$ of Γ by applying the smoothing scatterplot data $(X_{ij} - \hat{\mu}(t_{ij}))(X_{ij} - \hat{\mu}(t_{il}))$ to fit a local linear plane. Detail of this estimation will be discussed in subsequent chapters. Individual traffic trajectory can then be modeled, using their FPC scores, by

$$\hat{X}_i(t) = \hat{\mu}(t) + \sum_{j=1}^{M} \hat{\xi}_{ij}\hat{\phi}_j(t)$$

for $t \in \mathcal{I}$. We note that the above FPCA method can be implemented on fully observed functional data by Besse (1992), Besse and Ramsay (1986), and Silverman (1996), on densely observed functional data by Castro et al. (1986), Rice and Silverman (1991), and Cardot (2000), and in most common situation of sparely observed functional data by Staniswalis and Lee (1998), James et al. (2000), Rice and Wu (2001), Yao et al. (2005), Yao and Lee (2006), and Paul and Peng (2009).

1.3 Thesis Outline

This thesis considers a FDA approach to address above problems that have been mentioned. We treat the daily traffic flow trajectories as functional data that are sampled from random functions. FDA was introduced nearly two decades ago and various statistical methods for FDA have been intensively developed. Overviews of the FDA methodological foundations can be found in Ramsay and Silverman (2005) and Ferraty and Vieu (2006), as well as the review articles Rice (2004), Müller (2005), and Wang et al. (2016).

The thesis is organized as follows: In Chapter 2, applications of FPCA to the problem of missing data imputation and outlier detection are explored. In Chapter 3, we propose the naive Bayes classifier for functional data. Functional naive Bayes classifier is an extension from multivariate setting to the functional setting. We also investigate properties of the

classifier. Under regularity conditions, the proposed functional naive Bayes classifier has asymptotic equivalence to the true one. In Chapter 4, we propose a functional mixture prediction approach to predict future functional observations. The proposed method combines functional linear regression with the functional naive Bayes classifier. Chapter 5 is devoted to a change-point analysis. We propose a two-step segmentation algorithm for detecting multiple mean change-points in a sequence of functional data. In particular, functional data are transformed into FPC scores and used in the segmentation algorithm for estimating both the number and locations of the mean change-points. In Chapter 6, the concluding chapter, a summary is provided, and further topics in FPCA are briefly mentioned, including generalizations to multivariate FPCA and nonparametric approaches.

CHAPTER 2

DATA PREPROCESSING OF FUNCTIONAL DATA

Applications of traffic monitoring require complete and reliable data. These data can be recorded automatically by various types of vehicle loop detectors, which are usually installed under a planned road at regular intervals. Since loop detectors operate in a rough environment, missing data problems are inevitable due to detector malfunctions or package loss during transmission. Therefore, temporary detector malfunctions that result in loss of data are quite common. While the missing data problem takes place on a detector, the data on neighboring detectors located at the downstream or upstream are often missing as well. One way to deal with missing values is to eliminate sample records with missing values from the original dataset, yet the reduced dataset may lead to biased analysis results. Another approach is to reconstruct missing entries based on the recorded dataset; however, distinct imputation methods have their own advantages and disadvantages with different imputation performances depending on data availability scenarios. Each method may lead to different imputing results. Just like normal data collection or analysis procedures, missing data should be an important consideration in designing traffic data archiving or analysis systems for the purposes of highway planning and traffic surveillance and control, especially for ITS applications. Besides, outlier detection is another important issue in investigating traffic data. These include detecting temporal outliers in terms of magnitude in time (i.e., magnitude outliers) and identifying unusual patterns of trajectories (i.e., shape outliers), which provide useful information for further applications to traffic management.

Comprehensive overview of the issues of missing data can be found in Allison (1999) and Schafer and Graham (2002). Various imputation techniques have been developed in the past decades (see, e.g., Schafer (1999); Collins et al. (2001); King et al. (2001); Graham et al. (2003); Rubin (2004)). There is a large body of literature discussing methods of imputing missing values for multivariate data (Beale and Little (1975); Schafer (1997)) and longitu-

dinal data (Laird (1988); Little (1995); Little and Rubin (2002); Fitzmaurice et al. (2012)). Methods specifically discussed for traffic flow data have attracted significant attention. These include the Kalman filter method (Dailey, 1993), time series modeling (Nihan, 1997), historical (neighboring) imputation (Chen and Shao, 2000), the lane distribution method (Smith and Conklin, 2002), spline regression imputation methods (Chen et al., 2003), genetically designed modeling (Zhong et al., 2004). More recently, Li Qu et al. (2009) proposed Probabilistic Principal Component Analysis (PPCA) and Bayesian Principal Component Analysis (BPCA) imputation algorithms and compared their performance with some conventional methods from the literature. Although historical (or neighboring) imputation and spline (or local regression) imputation are frequently used methods for missing value imputation, they both suffer from some defects. As discussed in Li Qu et al. (2009), they ignore the fact that traffic flows may fluctuate significantly from day to day and contain stochastic variation within the same day. A basic historical imputation utilizes the global information in the sense of closely related or neighboring in historical data while spline imputation uses the local information in the sense of in-a-day flow data. Since the PPCA-based method considers an adaptive fusion of historical and in-a-day information, it outperforms the historical and imputation methods. Although numerous multivariate analysis methods have been developed to deal with missing values, to the best of our knowledge, functional data approaches that take advantage of functional data features have not yet been discussed in relation to imputing missing values for longitudinal or functional data.

As for outlier detection methods, for many statistical analysis procedures an essential step toward obtaining a coherent analysis is the detection of outlying observations. While outlier detection of multivariate data has been developed over several decades, outlier detection of functional data has only been discussed in recent years. Identification of abnormal or unusual patterns of trajectories that significantly deviate from other observations in a homogeneous group can improve the quality of observations and can be useful for further research. Abnormal data may adversely lead to model misspecification, biased parameter

estimation and incorrect results. Methods of outlier detection of functional data in the literature include the use of robust principal component analysis (Hyndman and Ullah, 2007), the successive likelihood ratio test and smoothed bootstrapping (Febrero et al., 2007), singular value decomposition plots (Zhang et al., 2007), rainbow plots, bagplots and boxplots for functional data (Hyndman and Shang, 2010) and functional boxplots (Sun and Genton, 2011).

This study considers a functional data analysis (FDA) approach to missing value imputation, where daily traffic flow trajectories are treated as functional data that are sampled from random functions. We propose to use the conditional expectation approach to FPCA for incompleteness of traffic flow data. Following the missing data imputation method, we also provide two outlier detection methods based on functional principal component (FPC) scores, the modified functional bagplot and the modified functional highest density region (HDR) boxplot, both of which are graphical tools aimed at identifying outlying curves of functional data.

2.1 Missing Values and Outliers in Functional Data

Data quality is an important issue encountered in analysis of traffic flow data. Although the data are automatically recorded by dual loop detectors, data corruption may happen due to short-term software or hardware malfunctions, maintenance operations and detector construction. These may lead to discontinuities or gaps and outliers in the data records, and may create severe obstacles in modeling and identification of the underlying stochastic mechanism. Therefore, it is essential to fill in missing gaps of the data and remove identified outliers before performing statistical analysis.

2.1.1 Patterns of Missing Values

Missing data can be random in nature and are sometimes caused by a detector that does not deliver measurement values or a fault in the measurement tools. Depending on the

measurement facility, missing values can appear as a blank, zero, negative, or not a number. Therefore, missing values are often simple to detect in a recorded dataset.

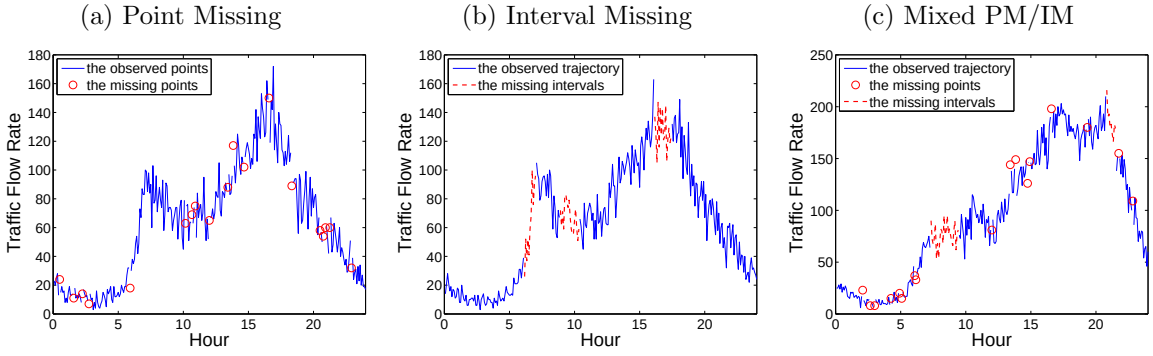


Figure 2.1: Typical missing patterns of traffic flow data. (a) Point Missing (PM): The circles are imputed missing values. (b) Interval Missing (IM): The dotted lines are imputed missing intervals. (c) Mixed PM/IM.

Rubin (1976) first developed a useful terminology for describing the patterns of missing data. Generally speaking, there are three kinds of missing patterns (see, e.g., Rubin (1976); Little and Rubin (2002) for multivariate data and Nakai and Ke (2011) for longitudinal data), including Missing Completely at Random (MCAR), Missing at Random (MAR) and Not Missing at Random (NMAR). When neither MCAR nor MAR holds as the missingness mechanism, it is termed as NMAR. NMAR may be caused by loop detector malfunctioning due to various reasons of machine failure, and it is rare to find an appropriate model for this missingness mechanism. In practice, the missing patterns of real traffic flow data may combine the patterns of MCAR and MAR, called Mixed MCAR/MAR. Since we cannot distinguish MAR and MCAR from NMAR based on the data, we simply classify the missing patterns as PM and IM, and also Mixed PM/IM for traffic flow data as follows. We note that PM and IM correspond to MCAR and MAR, respectively.

- Point Missing (PM): The missing points are completely independent of the observed and unobserved values. The missing points are isolated, grouped or randomly scattered. Both MCR and MR could be special cases of PM. See Figure 2.1(a).

- Interval Missing (IM): This definition is closely related to MR, but has a different focus. In terms of functional data, the data are curves instead of points. Hence, a missing interval means an unobserved interval rather than some unobserved points in a small group. The missing intervals often occur randomly. See Figure 2.1(b).

- Mixed PM/IM: The missing patterns can be PM or IM. See Figure 2.1(c).

Data incompleteness is a troubling feature of many datasets and missing values could be a serious problem as they may distort the properties of the data. Although there may be different patterns of missing values, the imputation method we propose is based on the partially observed trajectories and is not affected by the missing patterns.

2.1.2 Patterns of Outlying Curves

Outlier detection is a prerequisite in many data applications. There are several methods for outlier detection that can be distinguished as univariate versus multivariate techniques and parametric versus non-parametric procedures. For instance, the Mahalanobis distance is a well-known criterion that depends on estimated parameters of the multivariate distribution. Although there are many outlier detection methods for multivariate data, very few of them are for functional data. Defining an outlier or a contamination with a sample of curves is itself a tricky problem. Detecting outlying curves is a challenging task and mistakes or oversights in this area can have serious effects on statistical analysis, including biasing the results.

Following Hyndman and Shang (2010), there are two types of outliers, magnitude outliers and shape outliers. In general, magnitude outliers are distant from the mean and shape outliers have a pattern that is different from the other curves, e.g., see Figures 2.2(a) and 2.2(b), respectively. In practice, outlying curves may exhibit a combination of these features. When analyzing functional data, outliers can greatly affect estimates in many ways, including skewing the summary statistics and distorting the statistical modeling. Further research based

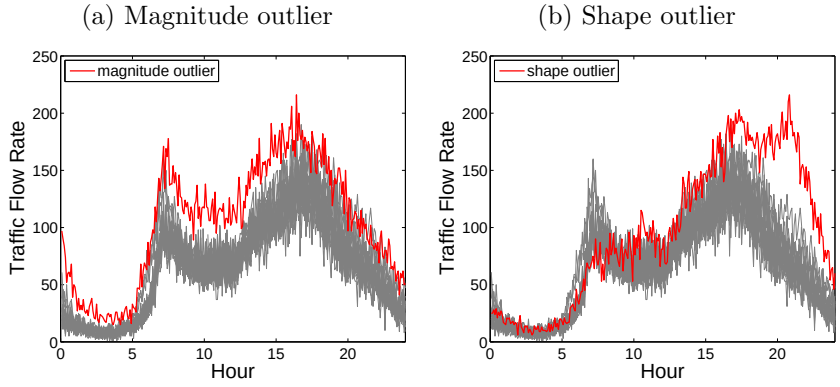


Figure 2.2: Typical outliers of traffic flow rate trajectories. (a) Magnitude outlier; (b) Shape outlier.

on such models and summaries can result in potentially serious failure due to previously undetected errors. Thus, identifying outliers can be important. A good methodology for trying to identify outlying curves should be able to cope with all types of outliers.

In this chapter, our focus is on identifying outlying curves or trajectories that have different patterns in terms of the underlying stochastic structure. Faulty data could be checked by the logic rules and they are distinguished from outliers due to special traffic incidents that cause volume surge or extreme traffic conditions so that the flow rate trajectories are beyond the extent of typical traffic variations. Outliers due to the occurrence of traffic incidents may require different incident detection techniques that should be able to detect abrupt changes in traffic streams; thus, it is beyond the scope of this chapter.

2.2 Functional Principal Components Analysis

Most functional data approaches are nonparametric due to the data features, which impose minimal assumptions on the data that overcome the limitations in parametric modeling. In this section, we make use of functional principal component analysis (FPCA) techniques to impute missing values of functional data.

We adopt the notion that each daily traffic flow trajectory is a realization of a random function. Let X denote the random function for the daily traffic flow trajectory. We further

assume that X has an unknown smooth mean function $EX(t) = \mu(t)$ and covariance function $Cov(X(s), X(t)) = \Gamma(s,t)$, $s, t \in \mathcal{I}$, where $\mathcal{I} = [0, T]$ in the L^2 space. Here we assume that Γ has an orthogonal expansion in L^2, that is, $\Gamma(s,t) = \sum_{k=1}^{\infty} \lambda_k \phi_k(s)\phi_k(t)$, where $\{\lambda_k\}$ is a set of eigenvalues in non-ascending order and $\{\phi_k\}$ is the corresponding set of eigenfunctions that form a basis with a unit norm in L^2. A random trajectory from the traffic flow then has the following Karhunen-Loève representation:

$$X_i(t) = \mu(t) + \sum_{k=1}^{\infty} \xi_{ik}\phi_k(t), \tag{2.1}$$

where $\xi_{ik} = \int (X_i(t) - \mu(t))\phi_k(t)\ dt$ is a random coefficient, projecting $(X_i - \mu)$ in the direction of the k-th eigenfunction ϕ_k, with a mean of zero and variance λ_k.

In practice, the random function X_i is often contaminated with measurement errors. The i-th data object, $i = 1, \ldots, n$, with m_i observations observed at t_{ij} for all t_{ij} in $\mathcal{I}$ and $j = 1, \ldots, m_i$, can be represented as

$$\begin{aligned} Y_i(t_{ij}) &= X_i(t_{ij}) + \varepsilon_{ij} \\ &= \mu(t_{ij}) + \sum_{k=1}^{\infty} \xi_{ik}\phi_k(t_{ij}) + \varepsilon_{ij}, \end{aligned} \tag{2.2}$$

where X_i is the random function described in (2.1), and the random measurement errors ε_{ij} are assumed to be uncorrelated with each other and are independent of ξ_{ik}, with $E(\varepsilon_{ij}) = 0$ and $Var(\varepsilon_{ij}) = \sigma^2$. To obtain the corresponding function estimate in (2.1), we must estimate the model component functions μ and ϕ_k. We apply the locally weighted least squares smoothing method on the pooled data from all trajectories for the estimated mean function μ, where the smoothing parameters can be chosen by various methods, including cross-validation (Rice and Silverman, 1991) or generalized cross-validation (Fan and Gijbels, 1996). To obtain the estimate of Γ, we adopt the techniques proposed in Yao et al. (2005) and smooth the empirical covariances. Then, we obtain the estimated eigenvalue $\hat{\lambda}_k$ and $\hat{\phi}_k$ by applying the eigen-decomposition procedure to the smoothed covariance function estimate.

The random coefficient estimate of ξ_{ik} cannot be obtained easily through $\hat{\xi}_{ik} = \int (X_i(t) - \hat{\mu}(t))\hat{\phi}_k(t)\ dt$. First, this integral approximation method encounters difficulties when there

are many missing entries or if only a few repeated observations available. Second, $X_i(t)$ cannot be observed directly but only through the observations Y_i that are contaminated with measurement errors. Estimating ξ_{ik} by substituting Y_i for X_i may lead to biased FPC scores. To overcome these difficulties, we adopt the approach of Yao et al. (2005) in relation to the conditional expectation by assuming that in (2.2), ξ_{ik} and ε_{ij} are jointly Gaussian. Let $\boldsymbol{Y}_i = (Y_i(t_{i1}), \ldots, Y_i(t_{im_i}))^T$, where m_i is the number of available observations for the i-th trajectory. Let $\boldsymbol{\phi}_{ik}$ be the vector of the values of the k-th eigenfunction, $\boldsymbol{\phi}_{ik} = (\phi_k(t_{i1}), \ldots, \phi_k(t_{im_i}))^T$. Let $\boldsymbol{\Sigma}_{\boldsymbol{Y}_i}$ be the covariance matrix of $\boldsymbol{Y}_i$, and $\boldsymbol{\mu}_i = (\mu(t_{i1}), \ldots, \mu(t_{im_i}))^T$. Under the assumption that the FPC scores ξ_{ik} and error term ε_{ij} are jointly Gaussian, the conditional FPC scores are

$$E(\xi_{ik}|\boldsymbol{Y}_i) = \lambda_k \boldsymbol{\phi}_{ik}^T \boldsymbol{\Sigma}_{\boldsymbol{Y}_i}^{-1}(\boldsymbol{Y}_i - \boldsymbol{\mu}_i). \tag{2.3}$$

The estimated conditional FPC scores in (2.3) are then obtained by substituting the corresponding estimates, giving

$$\hat{\xi}_{ik} = \hat{\lambda}_k \hat{\boldsymbol{\phi}}_{ik}^T \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{Y}_i}^{-1}(\boldsymbol{Y}_i - \hat{\boldsymbol{\mu}}_i), \tag{2.4}$$

where $\hat{\boldsymbol{\phi}}_{ik} = (\hat{\phi}_k(t_{i1}), \ldots, \hat{\phi}_k(t_{im_i}))$ is the estimate of $\boldsymbol{\phi}_{ik}$ and $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{Y}_i} = \{\widehat{\Gamma}(t_{ij}, t_{il}) + \hat{\sigma}^2 \delta_{ij}\}$, for $1 \leq j, l \leq m_i$, is the estimate of $\boldsymbol{\Sigma}_{\boldsymbol{Y}_i}$. Note that $\widehat{\Gamma}(t_{ij}, t_{il})$ and $\hat{\sigma}^2$ are estimates of $\Gamma(t_{ij}, t_{il})$ and σ^2, respectively, and δ_{ij} is the Kronecker delta coefficient. More details about the conditional expectation approach can be found in Yao et al. (2005).

2.2.1 Missing Value Imputation by Functional Principal Component Models

In view of (2.4), this FPC score estimate is applicable to situations where there are missing values and, thus, inspires our missing value imputation method. We truncate the infinite series by L in (2.1) using the fraction of variance explained (FVE) criterion such that the first L components explain at least $\tau_\lambda \times 100\%$ of total variance, i.e.,

$$L = \min \left\{ L \geq 1 : \frac{\sum_{k=1}^{L} \hat{\lambda}_k}{\sum_{k=1}^{M} \hat{\lambda}_k} \geq \tau_\lambda \right\}, \tag{2.5}$$

where M is the largest number of components with $\hat{\lambda}_k > 0$ and τ_λ is a predetermined threshold value, $0 \leq \tau_\lambda \leq 1$. Setting τ_λ equal 0.9 or higher works reasonably well in our simulation study. Based on the estimated model components $\hat{\mu}$, $\hat{\phi}_k$ and $\{\hat{\xi}_{ik}\}_{k=1,\ldots,L}$ for all i, the predicted functions for all of the i-th data object are then given by

$$\hat{Y}_i(t) = \hat{\mu}(t) + \sum_{k=1}^{L} \hat{\xi}_{ik}\hat{\phi}_k(t). \tag{2.6}$$

Since model (2.6) holds for all t in the entire time domain $\mathcal{I}$, the model fits can be used to impute the missing values. That is, if the observations $Y_i(t_{ij})$ are missing for some j, then the missing entries can be imputed by the predicted values $\hat{Y}_i(t_{ij})$. Note that the predicted trajectories $\hat{Y}_i$ include the components of the smoothed mean function and a linear combination of the eigenfunctions, which recover individual trajectories from noise measurements. Additionally, the imputation errors depend on the model complexity while the number of FPC scores are determined by FVE. We note that (2.5) provides a natural method to determine the number of FPC scores, and we will investigate the effect of the number of components on the imputed missing values.

2.2.2 Visualization Tools for Outlier Detection

A visualization tool can be useful to detect abnormal trajectories for functional data when data are contaminated with outlying curves. Two graphical tools were proposed in Hyndman and Shang (2010), functional bagplot and functional highest density region (HDR) boxplot, to detect outliers. Both are based on the first two principal component scores. For this purpose, the robust principal component (Croux and Ruiz-Gazen, 2005) estimation algorithm is applied with a form of projection pursuit because the principal component decomposition could be sensitive to outliers. In addition, this algorithm is more resistant to outliers when the measurement matrix contains outliers. However, this method does not take missing values into account; besides, some difficulties can arise when the sample covariance matrix has very high dimensionality. Computing this sample covariance itself is very costly. Furthermore,

the way of dealing with an incomplete dataset is not clear; particularly as the projection pursuit algorithm requires a complete dataset to project the data onto a lower-dimensional space such that a robust measure of variance of the projected data will be maximized.

To overcome the aforementioned difficulties, instead of using the robust principal component scores, we introduce FPC scores based on conditional expectation in a functional bagplot and functional HDR boxplot. Functional principal component scores can capture much of the information inherent in functional data since the covariance surface has smoothed out some outlying features and noise from the measurement errors. We call these two modified outlier detection tools 'modified functional bagplot' and 'modified functional HDR boxplot'. Details about the functional bagplot and the functional HDR boxplot were discussed in Hyndman and Shang (2010).

2.3 Simulation Study and Data Application

2.3.1 Simulation Study

In the simulation study, we consider various degrees of missingness or missing ratios and compare the conditional expectation approach to FPCA with other methods for missing value imputation. In addition, we investigate the performance of outlier detection for different patterns of outlying curves.

2.3.1.1 Performance Comparison in Missing Value Imputation

For missing value imputation, we compared our FPCA approach with the following methods:

- Functional Mean (FM): The missing values are imputed by the estimated mean function $\hat{\mu}(t)$, which is obtained by smoothing the pooled data from all trajectories.

- Probabilistic Principal Component Analysis (PPCA): This approach was proposed by Tipping and Bishop (1999) who introduced a Gaussian latent variable into classical Principal Component Analysis (PCA), and gave an Expectation-Maximization (EM)

algorithm for estimating the principal subspace. Moreover, the missing entries and the principal axes can be derived by the EM algorithm simultaneously.

- Bayesian Principal Component Analysis (BPCA): Bishop (1999) proposed a Bayesian estimation method to modify PPCA which introduced continuous hyper-parameters to determine the optimal value of the latent space dimensionality.

Both BPCA and PPCA are PCA-based multivariate imputation methods and are shown to outperform many conventional approaches in the literature. Brief reviews of the methods are given in appendix.

In order to evaluate the performance of missing value imputation methods, we use a real dataset that is archived in the Caltrans Performance Measurement System (PeMS) available through the link http://pems.dot.ca.gov/. The traffic flow rates were collected on a 5-min interval from April 1 to April 30 in 2013 (I5-N@CA PM5.32, District 11, San Diego County, City of San Diego. Detector ID: 1114734). We generated artificial missing entries from the real dataset according to the missing mechanisms of the PM, IM and Mixed PM/IM patterns as follows.

1. PM: The missing entries are generated point by point with locations following a uniform distribution.

2. IM: The missing entries are generated with an interval length of size that follows a uniform distribution ranging from half-hour (6 consecutive points) to 2-hour (24 consecutive points) periods. The locations of the missing intervals in each trajectory are uniformly distributed.

3. Mixed PM/IM: The missing entries are generated by combining PM with IM for individual trajectories.

The total number of missing entries amounts to nmp for each missing pattern, where p is the missing ratio, n is the number of trajectories, and m is the number of time points.

Table 2.1: Performance comparison in terms of the RMSE mean, standard error (in parenthesis) and average proportion of total variance explained [in bracket (%)] for FM, BPCA, PPCA and FPCA in PM, IM and Mixed missing patterns based on 100 simulation replications.

Pattern	Method		$p = 0.02$	$p = 0.18$	$p = 0.34$	$p = 0.50$
PM	FM		40.15 (4.07)	40.19 (1.31)	40.38 (0.77)	40.36 (0.58)
	BPCA		29.79 (11.73)	28.39 (10.58)	30.59 (10.60)	31.74 (9.89)
	PPCA (q)	1	20.88 (1.30) [77.2]	21.13 (0.42) [75.8]	21.87 (0.53) [74.1]	23.15 (0.80) [72.0]
		2	19.05 (1.17) [82.5]	19.48 (0.40) [81.5]	20.47 (0.61) [80.4]	22.22 (0.90) [79.3]
		4	19.54 (1.21) [85.2]	20.46 (0.49) [84.8]	22.38 (0.78) [84.6]	26.16 (1.64) [85.0]
		6	20.49 (1.36) [87.3]	21.99 (0.58) [87.4]	25.09 (1.05) [88.0]	30.57 (1.70) [89.9]
	FPCA (L)	1	20.51 (1.04) [91.0]	20.49 (0.45) [90.6]	20.65 (0.28) [90.2]	20.84 (0.28) [89.6]
		2	18.52 (1.24) [97.0]	18.59 (0.38) [96.5]	18.89 (0.28) [96.3]	19.18 (0.28) [95.6]
		4	18.01 (1.26) [99.0]	18.15 (0.38) [98.2]	18.53 (0.27) [98.0]	18.92 (0.28) [97.9]
		6	17.83 (1.31) [99.0]	18.03 (0.37) [99.0]	18.42 (0.27) [99.0]	18.82 (0.27) [99.0]
		AIC	20.59 (1.14) [91.0]	20.51 (0.43) [91.1]	20.65 (0.26) [90.2]	20.86 (0.25) [89.6]
		BIC	20.59 (1.14) [91.0]	20.51 (0.43) [91.1]	20.65 (0.26) [90.2]	20.86 (0.25) [89.6]
IM	FM		36.88 (13.01)	40.56 (3.49)	40.84 (1.87)	41.02 (1.51)
	BPCA		27.17 (12.77)	25.29 (9.33)	26.56 (9.77)	30.75 (9.93)
	PPCA (q)	1	20.90 (2.39) [77.2]	21.39 (0.69) [76.0]	21.91 (0.80) [74.6]	23.33 (1.93) [72.5]
		2	20.33 (2.09) [82.5]	20.80 (0.75) [81.7]	21.59 (0.91) [81.0]	23.87 (2.61) [79.9]
		4	20.06 (2.20) [85.2]	22.01 (2.49) [84.8]	24.11 (2.31) [84.8]	28.13 (3.25) [85.0]
		6	21.63 (3.21) [87.3]	23.97 (2.17) [87.4]	26.99 (2.55) [88.2]	32.75 (3.18) [89.6]
	FPCA (L)	1	20.52 (2.16) [90.9]	20.96 (0.68) [87.9]	21.35 (0.81) [84.7]	22.36 (1.45) [81.2]
		2	18.50 (2.22) [96.8]	19.76 (1.22) [94.5]	20.63 (1.64) [91.4]	21.71 (1.75) [88.7]
		4	18.79 (3.10) [98.8]	20.33 (1.27) [97.7]	21.02 (1.24) [96.5]	22.28 (2.07) [94.7]
		6	19.33 (3.27) [99.0]	20.61 (1.41) [98.9]	21.28 (1.31) [98.5]	22.48 (2.16) [97.1]
		AIC	20.84 (3.16) [90.6]	21.32 (2.38) [89.3]	21.58 (2.51) [88.2]	22.35 (2.97) [88.9]
		BIC	21.02 (2.37) [90.8]	21.53 (1.01) [89.1]	21.80 (1.24) [88.4]	22.61 (1.95) [88.8]
Mixed	FM		40.97 (10.76)	40.82 (2.73)	40.59 (1.70)	40.76 (1.48)
	BPCA		27.46 (12.65)	27.90 (10.92)	29.12 (10.42)	28.64 (9.48)
	PPCA (q)	1	20.68 (1.68) [77.2]	21.14 (0.60) [75.8]	22.02 (0.77) [74.2]	23.03 (0.95) [72.6]
		2	19.91 (1.58) [82.4]	20.51 (0.56) [81.5]	21.68 (0.88) [80.6]	23.15 (1.13) [79.9]
		4	19.60 (1.84) [85.2]	21.13 (1.38) [84.7]	23.11 (1.70) [84.7]	26.45 (2.03) [85.4]
		6	21.03 (2.42) [87.3]	22.82 (1.43) [87.4]	26.10 (2.10) [88.0]	30.82 (2.25) [89.9]
	FPCA (L)	1	20.61 (1.93) [90.9]	20.63 (0.58) [88.9]	20.85 (0.54) [88.3]	21.31 (0.54) [83.8]
		2	18.81 (1.37) [96.9]	18.95 (0.61) [95.0]	19.27 (0.63) [94.2]	20.49 (1.06) [91.0]
		4	18.45 (1.38) [98.7]	19.20 (0.79) [97.6]	19.28 (0.73) [97.2]	20.55 (0.79) [96.4]
		6	18.77 (1.55) [99.0]	19.26 (0.81) [98.9]	19.37 (0.81) [98.7]	20.66 (0.83) [98.2]
		AIC	20.73 (1.77) [90.9]	21.00 (0.64) [88.9]	21.33 (0.69) [88.3]	21.62 (0.98) [88.2]
		BIC	20.73 (1.77) [90.9]	21.00 (0.64) [88.9]	21.33 (0.69) [88.3]	21.62 (0.98) [88.2]

To compare their imputation performance, we use the root mean square error (RMSE)

as a criterion defined by

$$\text{RMSE} = \left\{ {n_0}^{-1} \sum_{i=1}^{n_0} {m_i}^{-1} \sum_{j=1}^{m_i} \{y_i(t_{ij}) - \hat{y}(t_{ij})\}^2 \right\}^{1/2},$$

where m_i is the number of missing points and n_0 is the number of trajectories whose $m_i > 0$. The RMSE measure compares the minimal difference between the imputed data values and the underlying observations, reflecting the performance of missing value imputation. A smaller RMSE indicates a better imputing performance.

The sample means with the associated standard errors of the RMSE results for the four methods are presented in Table 2.1, with different missing ratios p, ranging from 2% to 50%, based on 100 simulation replicates for three different missing types. It is clear that FM performs worse than the other methods since it does not take individual fluctuation into account. The RMSE of PPCA increases with missing ratio p, given a fixed number of components. The performance of BPCA looks more robust to missing ratios as compared to PPCA. While this could be the benefit of using a continuous type of hyper-parameter to determine the optimal number of latent space, BPCA also has the largest standard errors, reflecting relatively unstable imputation results. Figure 2.3 displays the RMSE values as a function of the missing ratios under the three missing patterns. The three cases consistently indicate that while the proposed FPCA outperforms the others, PPCA performs much better than BPCA, and FM is the worst. Further, the 95% FVE threshold value generally indicates slightly better performance than that of 90%. In general, the proposed FPCA imputation method performs the best in terms of RMSE including the sample means and standard errors, indicating the capability of capturing the information inherent in the functional data. Similar conclusions can be reached for the IM and Mixed PM/IM cases as well.

In addition, we use the box plots to display the information of RMSE variation between the imputed values and the observations. Figure 2.4 shows that for the PM missing pattern, FM has the largest bias, and BPCA has the largest standard errors, reflecting relatively unstable imputation results. The performance of PPCA depends the dimension of the latent

space (q) and its RMSE increases with the missing ratio. The proposed FPCA approach is robust to missing ratios and outperforms the other methods. Similar conclusions can be reached for the cases of IM and Mixed PM/IM missing patterns.

Here we compare the relative marginal improvement with FM, BPCA, PPCA for different dimensions of the latent space q, and using FPCA for different threshold values π_λ. The imputation performance based on the area under the curve (AUC) measure that considers the missing ratio p is defined as AUC $= \int_{p_L}^{p_U} \text{RMSE}(p)\, \mathrm{d}p$, where we set $p_L = 0.02$ and $p_U = 0.50$ in this study. The AUC results are summarized in Table 2.2. The imputation errors using FPCA are substantially smaller when compared with FM and BPCA, and is a little smaller than these based on PPCA. Choosing the best performance in FPCA in terms of π_λ and PPCA in terms of q, in the Mixed PM/IM case FPCA reduces 7.4% of AUC from PPCA, 44% from BPCA and 106% from FM.

Although FPCA and PPCA are both PCA-based methods, FPCA takes advantage of functional data features and, thus, could perform better than PPCA. In the estimation of the mean and the covariance functions in model (2), FPCA takes into account noise or measurement errors and recovers individual trajectories from noisy measurements. Moreover, the eigen-decomposition is performed on the smoothed covariance, which renders a relatively higher proportion of total variance explained by fewer number of components as

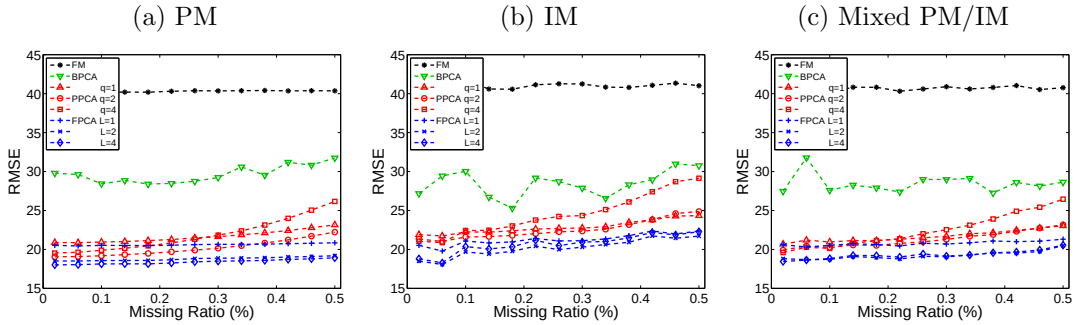


Figure 2.3: RMSE sample means as a function of missing ratios for the imputation methods, FM, BPCA, PPCA (with $q = 1, 2, 4$), and FPCA (with $L = 1, 2, 4$) based on 100 simulation replicates under the three missing patterns.

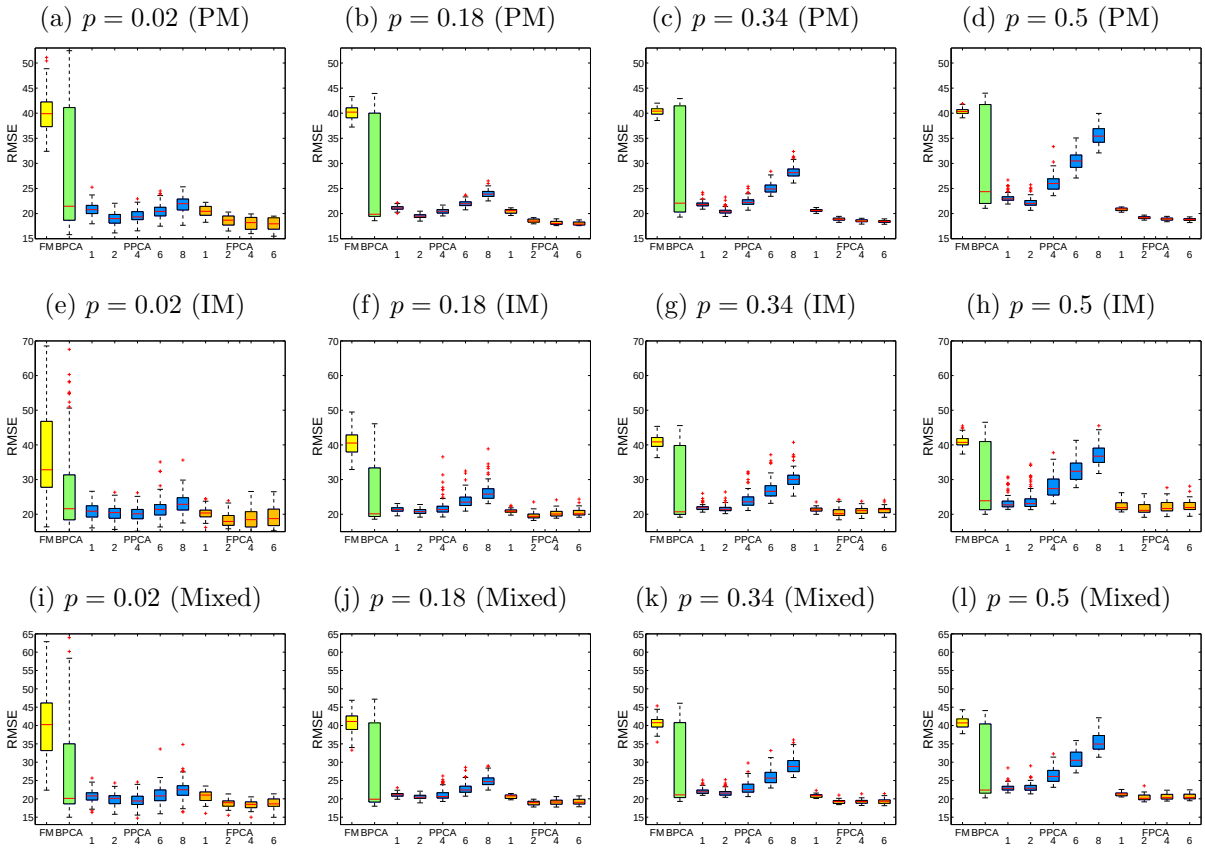


Figure 2.4: Boxplots of RMSE with different missing ratios p using the four methods for the three missing patterns based on 100 simulation replications.

Table 2.2: The AUC of the RMSE for PM, IM and Mixed PM/IM

Method		PM	IM	Mixed
FM		19.34	19.58	19.54
BPCA		14.19	13.64	13.68
PPCA (q)	1	10.38	10.49	10.40
	2	9.67	10.32	10.20
	4	10.46	11.26	10.74
	6	11.52	12.56	11.90
	8	12.82	13.89	13.24
FPCA (L)	1	9.65	10.21	9.96
	2	8.78	9.70	9.20
	4	8.59	9.95	9.26
	6	8.53	10.09	9.31
	AIC	9.89	10.33	10.16
	BIC	9.89	10.44	10.16

compared with those based on the unsmoothed covariance. In contrast, PPCA derives the unknown parameters from the observations via EM algorithm, which contain high noise or measurement errors. In addition, EM algorithm is sensitive to the initial values. Even if a good initial value is given, EM algorithm may converge to a local maximum of the observed data likelihood function rather than the global maximum, and the problem could be serious when the noise is high in the recorded data.

To assess the accuracy of the imputed data values, we calculate the Averaged Standardized Deviation (ASD) of the imputed missing values in the simulation study, where ASD is calculated by

$$\text{ASD} = \frac{1}{n_0}\sum_{i=1}^{n_0}\frac{1}{n_i}\sum_{j=1}^{n_i}\left\{\frac{(y_i(t_{ij})-\hat{y}_i(t_{ij}))^2}{s^2(y(t_{ij}))}\right\}^{1/2},$$

where n_i is the number of missing points and n_0 is the number of trajectories whose $n_i > 0$. Here, $y_i(t_{ij})$ is the observed flow rate of the missing entry, $\hat{y}_i(t_{ij})$ is the estimated missing value, and $s^2(y(t_{ij}))$ is the sample variance at time t_{ij}. The ASD results are summarized in Table 2.3. The results using FPCA indicate that the averaged deviations of the imputed values range from about 0.40 to 0.60 times the sample standard deviation, which appear to be acceptable.

2.3.1.2 Performance Comparison in Outlier Detection

For the performance of outlier detection, we compare the modified functional bagplot and HDR boxplot coupled with the conditional expectation approach, with the robust principal component based functional bagplot and HDR boxplot of Hyndman and Shang (2010). We generate synthetic curves from different models including magnitude outliers and shape outliers as follows.

- Model 1: The synthetic data trajectories are generated from model (2.2) with $L = 2$, the mean function $\mu(t) = t + \sin(t)$, the eigenfunctions $\phi_1(t) = -\cos(\pi t/10)/\sqrt{5}$ and $\phi_2(t) = \sin(\pi t/10)/\sqrt{5}$, $0 \le t \le 10$, the corresponding eigenvalues $\lambda_1 = 4$, $\lambda_2 = 1$, and

Table 2.3: Average standardized deviation of imputed values for the FM, PPCA, BPCA and FPCA methods in terms of the mean and standard error (in parentheses) based on 100 simulation replications.

Pattern	Method		$p = 0.02$	$p = 0.18$	$p = 0.34$	$p = 0.50$
PM	FM		0.80 (0.35)	0.63 (0.06)	0.66 (0.05)	0.66 (0.03)
	BPCA		0.69 (0.41)	0.55 (0.09)	0.58 (0.07)	0.60 (0.05)
	PPCA (q)	1	0.73 (0.30)	0.54 (0.07)	0.59 (0.05)	0.60 (0.03)
		2	0.61 (0.30)	0.50 (0.04)	0.56 (0.06)	0.57 (0.03)
		4	0.66 (0.30)	0.54 (0.05)	0.60 (0.04)	0.61 (0.04)
		6	0.72 (0.31)	0.57 (0.06)	0.65 (0.05)	0.69 (0.08)
		8	0.71 (0.36)	0.65 (0.09)	0.71 (0.07)	0.87 (0.13)
	FPCA (L)	1	0.68 (0.28)	0.52 (0.07)	0.56 (0.05)	0.57 (0.03)
		2	0.60 (0.23)	0.48 (0.06)	0.52 (0.06)	0.54 (0.03)
		4	0.59 (0.21)	0.46 (0.06)	0.52 (0.06)	0.53 (0.04)
		6	0.60 (0.24)	0.47 (0.05)	0.51 (0.06)	0.52 (0.04)
		AIC	0.57 (0.30)	0.47 (0.06)	0.50 (0.05)	0.50 (0.03)
		BIC	0.66 (0.30)	0.52 (0.06)	0.53 (0.05)	0.53 (0.03)
IM	FM		0.88 (0.51)	0.70 (0.10)	0.64 (0.04)	0.66 (0.05)
	BPCA		0.75 (0.56)	0.56 (0.11)	0.58 (0.05)	0.59 (0.08)
	PPCA (q)	1	0.64 (0.27)	0.54 (0.09)	0.58 (0.07)	0.58 (0.06)
		2	0.58 (0.22)	0.52 (0.06)	0.57 (0.04)	0.58 (0.08)
		4	0.56 (0.21)	0.55 (0.07)	0.60 (0.06)	0.66 (0.10)
		6	0.56 (0.19)	0.59 (0.07)	0.66 (0.10)	0.74 (0.04)
		8	0.72 (0.26)	0.64 (0.09)	0.68 (0.12)	0.81 (0.14)
	FPCA (L)	1	0.60 (0.25)	0.53 (0.09)	0.53 (0.06)	0.55 (0.04)
		2	0.53 (0.24)	0.48 (0.06)	0.52 (0.04)	0.52 (0.04)
		4	0.53 (0.25)	0.48 (0.05)	0.52 (0.04)	0.52 (0.05)
		6	0.52 (0.24)	0.49 (0.05)	0.52 (0.05)	0.53 (0.05)
		AIC	0.56 (0.30)	0.49 (0.09)	0.53 (0.07)	0.55 (0.05)
		BIC	0.56 (0.30)	0.49 (0.09)	0.53 (0.07)	0.55 (0.05)
Mixed	FM		0.66 (0.26)	0.64 (0.08)	0.65 (0.05)	0.65 (0.03)
	BPCA		0.50 (0.28)	0.52 (0.11)	0.61 (0.07)	0.60 (0.07)
	PPCA (q)	1	0.46 (0.36)	0.54 (0.10)	0.58 (0.06)	0.58 (0.03)
		2	0.46 (0.34)	0.50 (0.10)	0.56 (0.04)	0.56 (0.03)
		4	0.48 (0.30)	0.53 (0.08)	0.61 (0.06)	0.62 (0.12)
		6	0.54 (0.36)	0.57 (0.07)	0.65 (0.06)	0.72 (0.17)
		8	0.62 (0.35)	0.63 (0.06)	0.73 (0.07)	0.89 (0.23)
	FPCA (L)	1	0.45 (0.40)	0.53 (0.08)	0.56 (0.06)	0.56 (0.05)
		2	0.45 (0.36)	0.49 (0.08)	0.52 (0.05)	0.53 (0.05)
		4	0.44 (0.36)	0.49 (0.08)	0.53 (0.05)	0.54 (0.04)
		6	0.46 (0.36)	0.50 (0.08)	0.51 (0.05)	0.53 (0.03)
		AIC	0.45 (0.30)	0.50 (0.10)	0.52 (0.06)	0.55 (0.04)
		BIC	0.45 (0.30)	0.50 (0.10)	0.52 (0.06)	0.55 (0.04)

the variance of measurement error $\sigma^2 = 0.25$. The FPC scores ξ_{ik} are generated from $N(0, \lambda_k)$ and the measurement errors ε_i are generated from $N(0, \sigma^2)$. We generate

$n = 190$ synthetic curves with 10 additional magnitude outlying curves generated from the same model but with an inflated mean function $\mu(t) = (-1)^r K + t + \sin(t)$, where $1 \leq r \leq 10$ and $K = 10$.

- Model 2: We generate $n = 190$ curves based on the same structure of Model 1 with 10 additional shape outlying curves generated using different eigenfunctions $\phi_1(t) = -\sin(\pi t/10)/\sqrt{5}$ and $\phi_2(t) = \cos(\pi t/10)/\sqrt{5}$.

- Model 3: We simulate 990 curves of the form $Y_i(t) = a_i \sin(t) + b_i \cos(t)$ without measurement errors, where $0 < t < 2\pi$. The coefficients a_i and b_i follow independent uniform distributions on $[0.0, 0.1]$. Ten additional curves are also randomly simulated with the same functional form, but with a_i and b_i following uniform distribution on $[0.1, 0.12]$.

We note that Model 1 includes some symmetric magnitude outliers as shown in Figure 2.5(a), Model 2 contains shape outliers as shown in Figure 2.5(b) and Model 3 is a harmonic signal process with outlying curves that are not far from the median curve as shown in Figure 2.5(c). We note that Model 3 was used in Hyndman and Shang (2010).

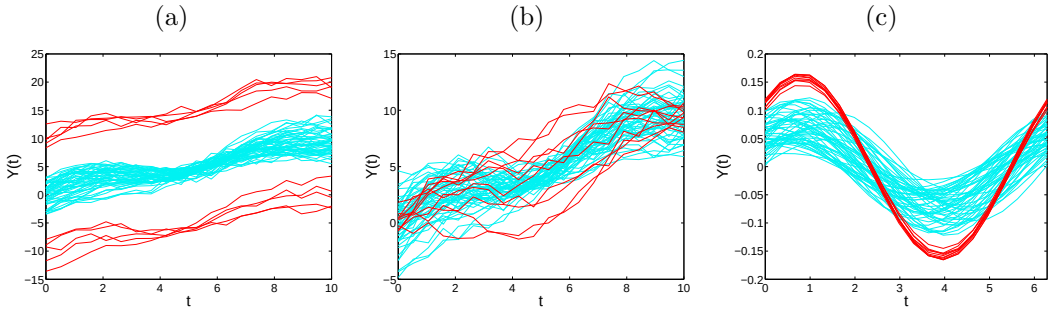


Figure 2.5: Three models of synthetic observations with different patterns of outliers including (a) Model 1 with symmetric magnitude outliers, (b) Model 2 with shape outliers, and (c) Model 3 with a harmonic signal process.

For Models 1 and 2, the first two functional principal components (FPCs) explain 99.07% and 99.70% of the total variation, whereas the first two robust principal components (RPCs) explain only 87.81% and 85.59% of the total variation. The difference occurs because FPCA

takes advantage of the smoothed covariance, which can capture most information inherent in the functional data structure by the first few FPCs; in contrast, the robust PCA is based on the projection pursuit from the robust scale estimator of raw covariance, which explains a relatively low explanation of the first few robust PCs as compared to FPCs. For Model 3, the first two FPCs explain 99.40% of the total variation and the first two robust PCs explain 99.95% of the total variation. Since the data generated from Model 3 is quite smooth and without measurement errors, the robust PCA can take advantage of the robust scale estimator and performs well.

We first compare the performance of the modified functional bagplots and the functional bagplots for Models 1–3. Figure 2.6 shows the outlier detection results of the two methods. It can be seen that the modified functional bagplots (column 2) work slightly better than the functional bagplots (column 4) in Models 1 and 2. For Model 3, the outlier detection results are identical, but both methods have failed to identify many outliers in our simulation study. This is because the curves are not sufficiently distant from the median as was shown by Hyndman and Shang (2010).

To further compare the performance of outlier detection, we introduce two performance measures as defined in Sun and Genton (2011).

- p_c: the percentage of correctly detected outliers defined as the number of correctly detected outliers divided by the total number of outlying curves.

- p_f: the percentage of falsely detected outliers defined as the number of falsely detected outliers divided by the total number of non-outlying curves.

A good outlier detection performance requires a high correct detection percentage p_c and a low false detection percentage p_f.

The percentages of the sample means and standard errors of $\hat{p}_c$ and $\hat{p}_f$ based on 200 simulation replicates are shown in Table 2.4. For Model 1, both methods have 100% correct outlier detection rate, yet the modified functional bagplot has a lower false detection rate

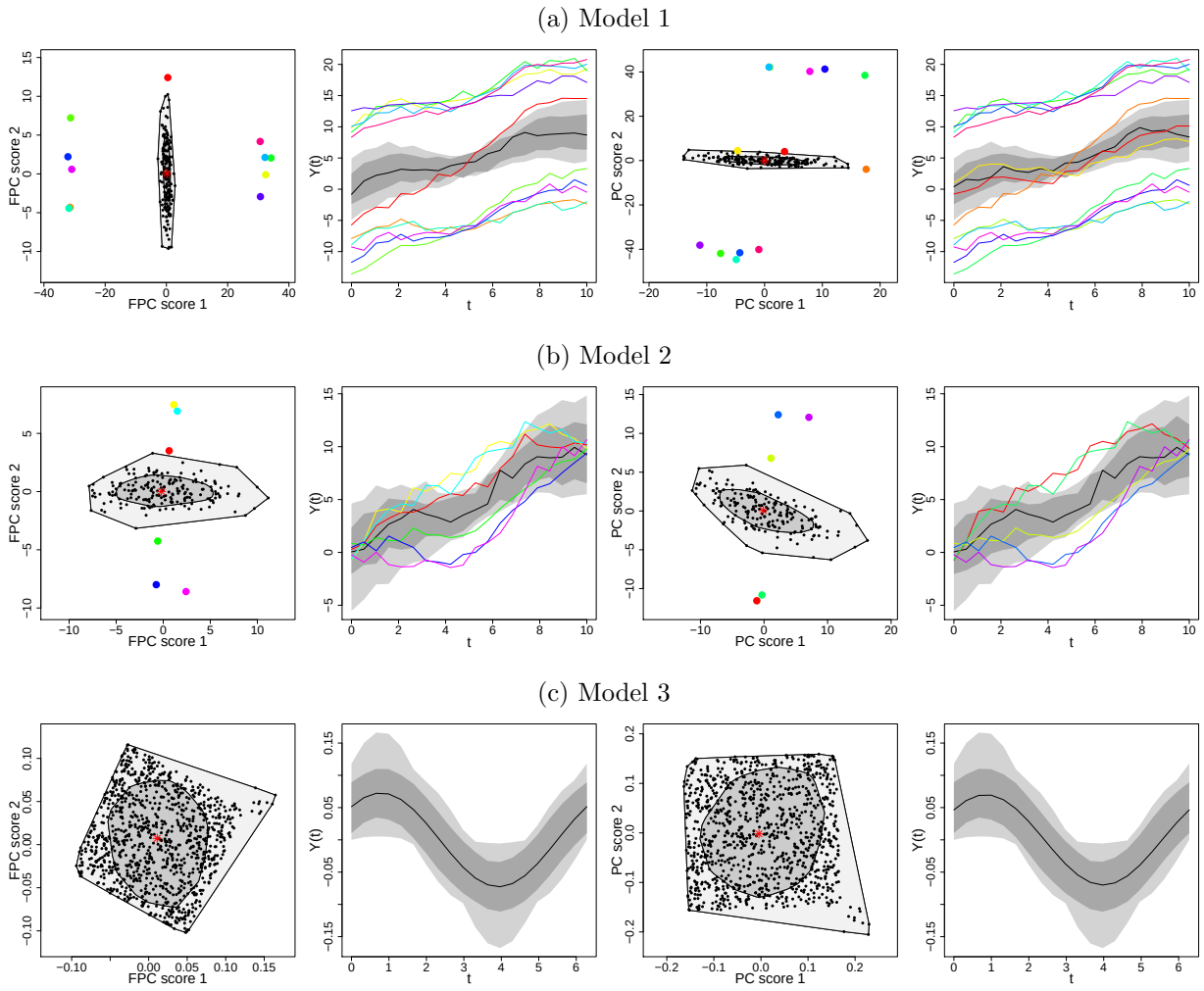


Figure 2.6: The modified bivariate bagplot (column 1), the modified functional bagplot (column 2), the bivariate bagplot (column 3), and the functional bagplot (Column 4) for a sample of synthetic curves for Models 1, 2, and 3.

Table 2.4: The sample means and standard errors (in parentheses) in percentages of $\hat{p}_c$ and $\hat{p}_f$ for the modified functional bagplot and functional bagplot with 200 replications.

Model		Modified Functional bagplot	Functional bagplot
1	$\hat{p}_c$	100.00 (0.00)	100.00 (0.00)
	$\hat{p}_f$	0.82 (0.72)	1.31 (4.69)
2	$\hat{p}_c$	59.55 (15.99)	59 (16.23)
	$\hat{p}_f$	1.41 (4.76)	1.48 (4.81)
3	$\hat{p}_c$	0.20 (1.40)	0.20 (1.40)
	$\hat{p}_f$	0.00 (0.00)	0.00 (0.00)

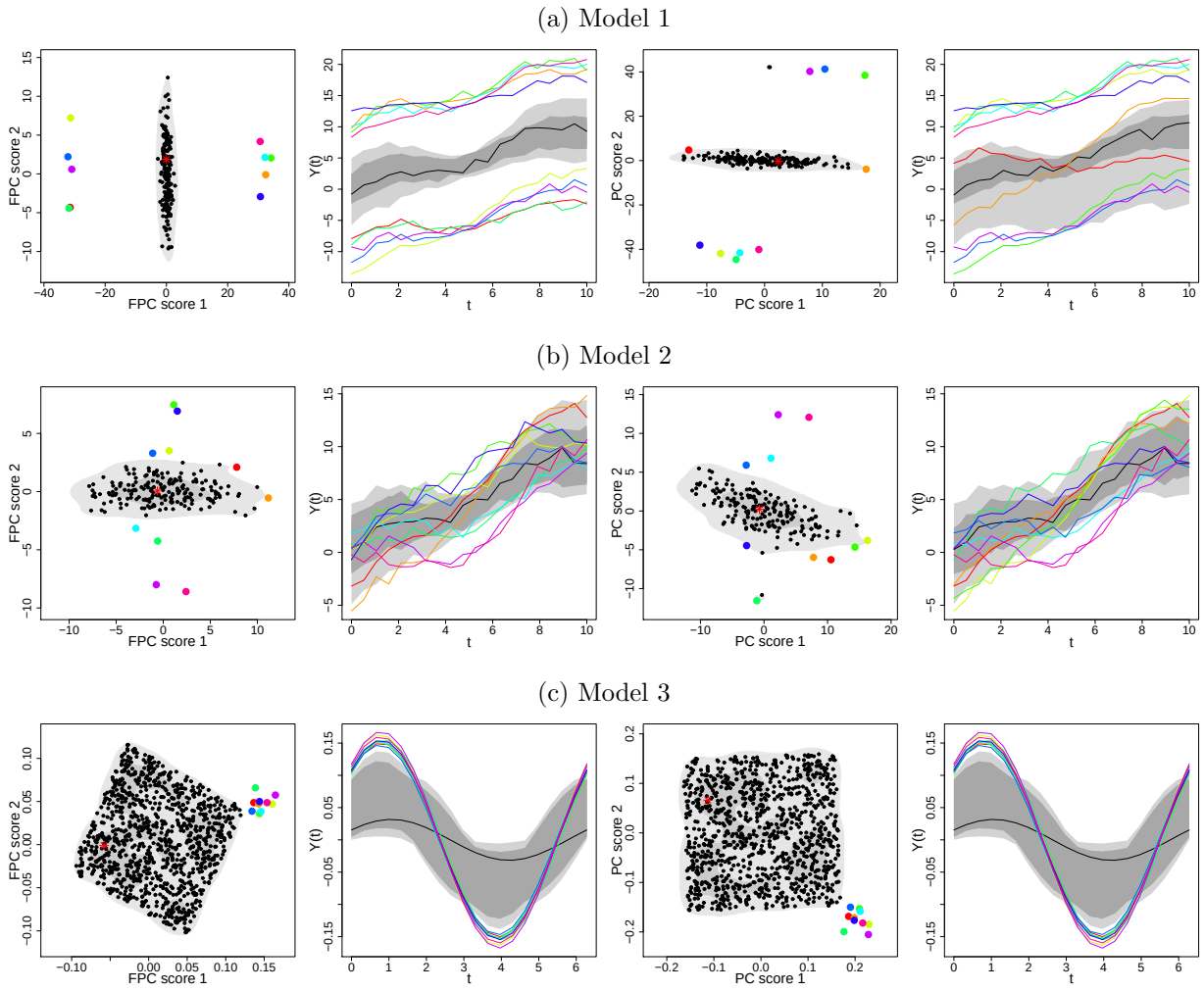


Figure 2.7: The modified bivariate HDR boxplot (column 1), the modified functional HDR boxplot (column 2), the bivariate HDR boxplot (column 3), and the functional HDR boxplot (column 4) for a sample of synthetic curves for Models 1, 2 and 3.

with a smaller standard error. For Model 2, the modified functional bagplot has a slightly higher correct and a slightly lower false outlier detection rate, both with a smaller standard error. For Model 3, both methods work similarly and the results are identical. Focusing on Models 1 and 2, without the help of robust PCA, FPCA still can derive a satisfactory result.

The functional HDR boxplot requires a pre-specified coverage probability, say α, which can be viewed as the percentage of potential outliers. Since we simulate 200 curves, including 10 outliers, for Models 1 and 2, and 1000 curves, including 10 outliers for Model 3, it is reasonable to set $\alpha = 0.05$ for Models 1 and 2, and $\alpha = 0.01$ for Model 3. A sample of the

modified functional HDR boxplots and functional HDR boxplots for the three models are shown in Figure 2.7. The modified functional HDR boxplots (column 2) work better than the functional HDR boxplots (column 4) for Models 1 and 2, especially for Model 1 where the modified functional HDR boxplot correctly identifies all of the outliers without any false detection. Moreover, both methods work equally well on Model 3.

We consider different coverage probabilities of the outlying region based on $\alpha = 0.01, 0.05$ and 0.1 for each model with the sample means and standard errors reported in Table 2.5. Overall, the modified functional HDR boxplot appears to work better than the functional HDR boxplot, except for Model 3 with $\alpha = 0.01$; however, the results of the two methods are not significantly different. In addition, the distribution of $\hat{p}_c$ and $\hat{p}_f$ are identical for $\alpha = 0.05$ and 0.1 in Model 3. For the functional HDR boxplot, $\hat{p}_c$ increases with α so that more outliers are detected, but $\hat{p}_f$ also increases, meaning more non-outlying curves are flagged as potential outliers. When performing functional HDR boxplot, one should be very careful with the pre-specified α since the outlier detection performance depends on the choice of α.

Table 2.5: The sample means and standard errors (in parentheses) of the percentages $\hat{p}_c$ and $\hat{p}_f$ for the functional HDR boxplot and functional HDR boxplot with 200 replications.

Model		Modified Functional HDR boxplot			Functional HDR boxplot		
		$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.1$
1	$\hat{p}_c$	18.10 (4.18)	88.05 (9.49)	100.00 (0.00)	17.55 (4.54)	85.7 (10.49)	100.00 (0.00)
	$\hat{p}_f$	0.10 (0.22)	0.63 (0.50)	5.26 (0.00)	0.13 (0.24)	0.75 (0.55)	5.26 (0.00)
2	$\hat{p}_c$	18.30 (3.90)	59.00 (13.71)	70.00 (14.39)	18.20 (4.22)	58.55 (13.83)	69.25 (14.63)
	$\hat{p}_f$	0.09 (0.21)	2.16 (0.72)	6.84 (0.76)	0.09 (0.22)	2.18 (0.73)	6.88 (0.77)
3	$\hat{p}_c$	97.25 (6.09)	100.00 (0.00)	100.00 (0.00)	97.50 (5.47)	100.00 (0.00)	100.00 (0.00)
	$\hat{p}_f$	0.03 (0.06)	4.04 (0.00)	9.09 (0.00)	0.03 (0.06)	4.04 (0.00)	9.09 (0.00)

2.3.2 Data Application

We implement the proposed missing value imputation and outlier detection methods for traffic flow data collected by a dual loop vehicle detector located at 28.45K northbound on National Highway No. 5 in Taiwan, which is near the entrance of Shea-San Tunnel at 28.11K northbound. The traffic flow rates were collected on a 5-min interval from April 1 to April 30 in 2009. National Highway No. 5 is the major road northbound from Yilan County to Taipei. Yilan County is located nearby Taipei and many people living in Taipei like to go to Yilan County during weekends and holidays for their recreational trips. Therefore, numerous trips northbound on National Highway No. 5 are recreational trips coming back from Yilan County to Taipei, especially starting from the afternoon till evening during weekends and holidays. As shown in Figure 2.8(a) for the observed traffic flow rate trajectories, the peak hours occurred between 14:00 and 21:00. The data set consists of a sample of 30 functional observations, among which 22 were weekdays and 8 were weekends (including Chinese Tomb-Sweeping holiday on April 5), and each sample contains 288 data observations for complete data. There were some missing entries caused by the malfunction of detector, lost packages during transmissions and other reasons. We define the missing ratio of the dataset by $p = \sum_{i=1}^{n} m_i/mn$, where n is the number of observed days, m is the maximal number of observed time points and m_i is the number of missing points of subject i. The overall missing ratio for this data set is 2.28%.

2.3.2.1 Functional Principal Component Analysis

Observing that the traffic flow patterns are distinct on weekends (including holidays) and weekdays, we separate the FPC analysis in these two groups. The observed trajectories and the estimated mean function are displayed in Figures 2.8(a) and 2.9(a). The estimated mean functions indicate the peak hours on weekends occur from 14:00 to 20:00, while there are two peaks on weekdays, one around 8:00 with lower flow rates and smaller variability and the other around 17:00 with relatively high flow rates and variability. The estimated

auto-covariance functions are shown in Figures 2.8(b) and 2.9(b). On weekdays the first peak occurs around from 7:00 to 9:00, which is on-work state, and the second peak occurs from 16:00 to 18:00, which is off-work state. The time from 9:00 to 16:00 shows a regular state. The variability is more complicated on weekends with high variability during peak hours. This smoothed covariance surface reveals the structure of the underlying process, which would be difficult for modeling using traditional parametric approaches. In addition, the eigenfunctions from the decomposition of the estimated covariance are shown in Figures 2.8(c) and 2.9(c). The cumulative fractions of total variance explained by the leading components are displayed in Figures 2.8(d) and 2.9(d). The results indicate that setting π_λ as 0.9 results in 3 components for weekends and 2 components for weekdays, while setting π_λ as 0.95 leads to 4 components for both weekends and weekdays. In addition to the FVE criterion, the AIC- and BIC-like criteria proposed by Yao et al. (2005), which are derived under the Gaussian process assumption, can be used as well. The results using these criteria coincide with those based on the FVE criterion with 0.90 as the threshold in the dataset. The threshold values 0.90 and 0.95 appear to be reasonable choices. However, the threshold 0.9 renders slightly smaller prediction errors in our simulation study, and, thus, we choose 0.9 as the threshold value for our real dataset. The three leading FPCs account for 65.81%, 15.20%, and 11.03% of total variation in weekends, where the first eigenfunction reflects the overall variability in the peak-hour period. In contrast, the two leading components explain 55.01% and 36.42% of total variation in weekdays, where the first and the second eigenfunctions contrast variability between early and late times.

Figure 2.10 displays samples of observed daily traffic flow trajectories, along with the predicted functions and the imputed missing values of different missing patterns. The imputed results appear reasonable. Particularly, the method can catch curvature pattern in interval missing as shown in Figure 2.10(c).

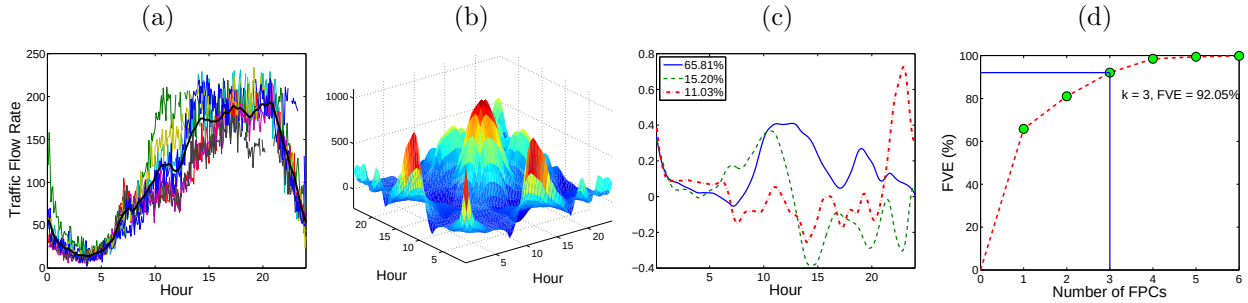


Figure 2.8: (a) Daily traffic flow trajectories superimposed on the estimated mean function, (b) the estimated covariance function, (c) the cumulative fraction of total variance explained by the leading FPCs, and (d) the estimated eigenfunctions for weekends (including holidays).

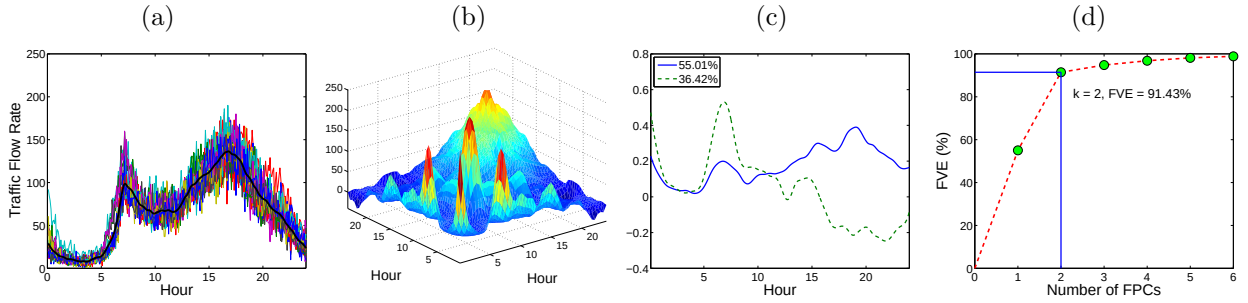


Figure 2.9: (a) Daily traffic flow trajectories superimposed on the estimated mean function, (b) the estimated covariance function, (c) the cumulative fraction of total variance explained by the leading FPCs, and (d) the estimated eigenfunctions for weekdays.

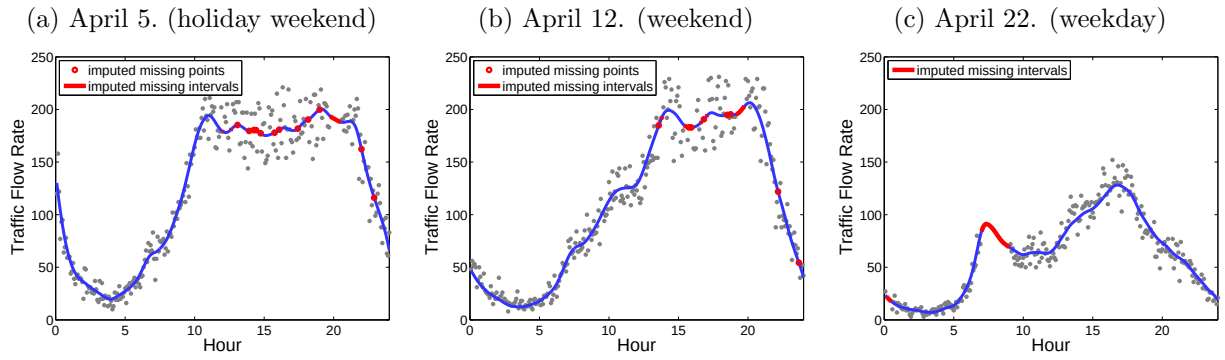


Figure 2.10: Three samples of daily traffic flow rate trajectories with the observations (dots in gray), the predicted trajectories (curves in blue) and the imputed missing values (dots or curves in red).

2.3.2.2 Outlier Detection Results

Detecting extreme traffic flow trajectories through visual inspection can be difficult due to large noise and missing values. We perform outlier detection by applying the modified functional bagplot and the modified functional HDR boxplot, both of which use the functional principle component scores based on the conditional expectation approach.

The outlier detection results based on the modified functional bagplot are shown in Figure 2.11. Figure 2.11(a) displays the modified bivariate bagplot, where the red star marks the Tukey median of the bivariate FPC scores, the dark gray region displays the 50% bag and the light gray region shows the 95% fence. The point at April 4 and April 5, outside the fence are identified as an outlier. The modified functional bagplot is shown in Figure 2.11(b), where the solid black curve (median curve) corresponds to the median point (red star) and the similar shaded dark and light gray region correspond to the bag and fence in the modified functional bagplot, respectively. The outlying curves at April 4 and April 5 are highlighted in green and red. In this dataset, the suspected functional outlier dated April 5 was Chinese Tomb-Sweeping holiday. Figure 2.11(c) illustrates the modified bivariate HDR boxplot using the setting of $\alpha = 0.15$. We note that when using the setting of $\alpha = 0.01$ to 0.1 April 4 and April 5 are identified as outliers. Figure 2.11(d) display the corresponding modified functional HDR boxplots.

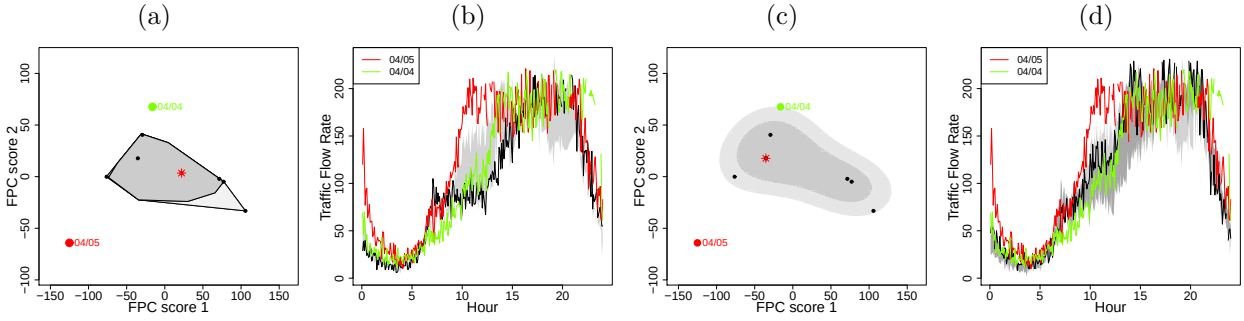


Figure 2.11: (a) The modified bivariate bagplot, (b) the modified functional bagplot, (c) the modified bivariate HDR boxplot (with $\alpha = 0.15$), and (d) The modified functional HDR boxplot for outlier detection on weekends.

More descriptions on the modified bivariate HDR boxplot and functional boxplot will follow. April 5 was on Sunday and it was also the Chinese Tomb-Sweeping holiday. It was a special holiday for people returning back to their hometowns for getting together with their families. In addition to recreational trips, there are many back-to-hometown trips. Therefore, it is understood that the traffic flow pattern on April 4 and April 5 were quite different from other weekends as illustrated in Figure 2.11.

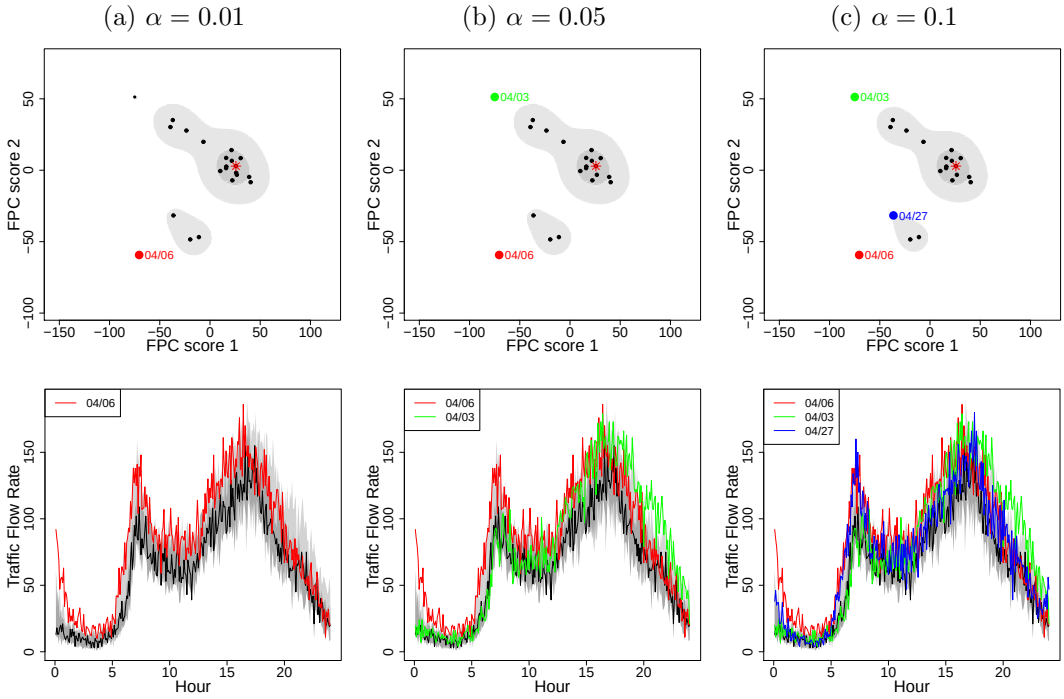


Figure 2.12: The modified bivariate HDR boxplots with $\alpha = 0.01$, $\alpha = 0.05$ and $\alpha = 0.1$. Top panels: the modified bivariate HDR boxplot. Bottom panels: the modified functional HDR boxplot.

For weekdays, no outliers are detected based on the modified functional bagplot approach. To use the functional HDR boxplots, a prespecified coverage probability of the outlying region is needed. We use three coverage probabilities 99%, 95%, and 90%, corresponding to the settings of $\alpha = 0.01$, 0.05, and 0.1, to perform the outlier detection procedure for the traffic flow data. The top panels of Figures 2.12 illustrate the modified bivariate HDR boxplots in which the red star marks the mode of the bivariate FPC scores, the darker gray regions

display the 50% HDR, and the lighter gray regions display the 99%, 95% and 90% HDR, respectively. The points outside the light gray regions are identified as outliers. The bottom panels of Figures 2.12 display the corresponding modified functional HDR boxplots, where the black curves correspond to the mode of bivariate FPC scores, and the shaded dark and light regions correspond to the regions in the modified bivariate HDR boxplots. The modified functional HDR boxplots with $\alpha = 0.01, 0.05$, and 0.1 detect one, two and three outliers in the order of April 6, 3 and 27, which are highlighted in red, blue, and green, respectively. The capability of identifying outliers using the functional HDR boxplots highly depends on the pre-specified α. The results show that more outliers are detected with larger values of α and the strength of potentially flagged outliers can be identified by varying the values of α. While the curve corresponding to April 27 is very close to the boundary of the 90% region, the identified outliers on April 3 (Friday) and April 6 (Monday) are both around the April 5 Chinese Tomb-Sweeping holiday, which gives a reasonable interpretation. Based on the outlier detection results, we found that traffic flow patterns on the special holiday of April 5, and the days before and after the holiday are different from the general weekend or weekday patterns. Thus traffic control strategies for such holidays require separate considerations for the weekends and the weekdays.

2.4 Conclusion

In this study, we proposed a nonparametric functional data approach to missing value imputation and outlier detection for functional data. Our method takes advantage of the functional data features that can be expanded by the FPC models consisting of the mean function and a stochastic component to catch individual variation. We investigate the numerical performance through comparison with popularly used imputation methods in the literature for handling missing traffic flow data. The proposed FPCA based on the conditional expectation approach outperforms PPCA and BPCA in addressing traffic flow data incompleteness. Moreover, we proposed a modified version of the functional bagplot and the

functional HDR boxplot that applies to the FPC scores. One of the advantages of the proposed approach is that it can be used even for incomplete or irregularly collected functional data. The simulation study shows that the proposed FPCA approach for missing value imputation and the modified outlier detection methods can work reasonably well. Although motivated by traffic flow data, the proposed methodology is widely applicable to data that are repeatedly measured over a period of time.

CHAPTER 3

THE NAIVE BAYES CLASSIFIER FOR FUNCTIONAL DATA

The problem of classifying objects is a popular topic in machine learning and statistics. Classification is the problem of identifying a set, where a new observation belongs to, based on a training set of data containing observations whose category memberships are known. The idea is to produce a so-called classifier, which can be viewed as a function induced by a classification algorithm that maps input data to a category. Conventional heuristic methods, such as the naive Bayes classifiers (Langley et al., 1992; Mitchell, 1997) have been widely used in many applications. The naive Bayes classification approach is based on the assumption of the independence between the features, and the output label simply relies on the estimation of univariate conditional probabilities. Experiments demonstrate that despite the naive design and apparently oversimplified assumptions, naive Bayes classifiers bring a competitive performance compared to other state-of-the-art classifiers in many complex real-world situations (Baesens et al., 2003).

In the finite dimensional setting, the multivariate probability density function, provided that it exists, is the main tool for constructing naive Bayes classifier. However, in the infinite dimensional setting, where the data belongs to a functional space, the problem of "the curse of dimensionality" occurs immediately, and as a result the naive Bayes approach is not directly applicable. The difficulty comes from the fact that the notion of a probability density is not well-defined due to the infinite dimensionality of data. Often a probability density function for functional data does not even exist (Delaigle and Hall, 2010). Therefore, a direct extension of density-based classical multivariate classification approaches to functional data cannot be utilized.

On the other hand, functional classification has attracted a lot of attention recently due to a large demand of real world applications including real-time signal analysis (Hall et al., 2001), temporal gene expression curves (Leng and Müller, 2006), traffic flow patterns (Chiou,

2012) and many other fields. Methods of classification of functional data can vary. There has been rapidly expanding amount of research within the statistical field on the development of functional classification methods. These include the k-nearest neighbor (k-NN) methods (Biau et al., 2005; Cérou and Guyader, 2006; Biau et al., 2010), linear discriminant analysis (LDA) (James and Hastie, 2001; Shin, 2008), partial least squares (PLS) (Preda and Saporta, 2005; Preda et al., 2007), logistic regression (Leng and Müller, 2006; Araki et al., 2009), support vector machines (SVM) (Rossi and Villa, 2006; Wu and Liu, 2013; Martin-Barragan et al., 2014), distance-based approaches (Alonso et al., 2012; Galeano et al., 2015), depth-based notions (Cuevas et al., 2007; Sguera et al., 2014), shape descriptors (Epifanio, 2008), nonparametric kernels (Hall et al., 2001; Ferraty and Vieu, 2003), Bayesian methods (Wang et al., 2007), and centroid method (Delaigle and Hall, 2012). More recently, Bongiorno and Goia (2016) employ the idea of pseudo-density to build a classifier based on the Bayes rule. However, their method was based on the multivariate kernel density estimator, which is often restricted to a lower dimension unless the sample size is extremely large since the accuracy of nonparametric density estimators decrease rapidly as dimension increases. Dai et al. (2017) propose to use density ratios of projections on a sequence of common eigenfunctions on the two populations. Our approach is different from theirs, since we work directly with the extension of density-based naive Bayes classifier for functional data.

Despite differences in building classifiers, functional principal component analysis has been used as the main tool in the dimension reduction for functional data. Overviews of the functional data analysis methodological foundations can be found in Ramsay and Silverman (2005) and Ferraty and Vieu (2006), as well as in the review article (Wang et al., 2016). In this chapter, we work toward the construction of naive Bayes classifier for functional data. In order to do so, a density function for a random function is constructed from the functional common principal component (FCPC) scores with an aid of independence assumption. The FCPC scores are the coefficients of the random functions that projected onto an orthogonal basis derived from the decomposition of the common covariance function.

The idea of common principal components can be dated to Flury (1984) in multivariate analysis, and has been extended by Boente et al. (2010) to functional data analysis. This step of construction of density function from FCPC scores makes the naive Bayes classifier possible for functional data. We then prove that under some general assumptions the proposed functional naive Bayes (FNB) classifier has the asymptotic equivalence to the true one. We also compare FNB numerical performance with logistic regression, k-NN, LDA, and SVM. Several simulation experiments suggest that the proposed FNB methods outperform other methods.

3.1 The Naive Bayes Classifier for Functional Data

This section describes a model for functional classification: functional naive Bayes classifier. The setting is as follows. Let X be a random function in $L_2(\mathcal{I})$, which is a Hilbert space of square integrable functions on a compact interval $\mathcal{I}$ equipped with the inner product of two functions f and g defined through the integral operator by $\langle f, g\rangle = \int_{\mathcal{I}} f(t)g(t)dt$ with the associated norm $||\cdot|| = \langle \cdot, \cdot \rangle^{\frac{1}{2}}$. The mean function of X is $E(X(t)) = \mu(t)$ and the covariance function of X is $\text{cov}(X(s), X(t)) = \Gamma(s,t)$, $s, t \in \mathcal{I}$. The covariance Γ is positive definite and has an orthogonal expansion in L_2, $\Gamma(s,t) = \sum_{j=1}^{\infty} \lambda_j \phi_j(s)\phi_j(t)$, where $\{\lambda_j\}_{j=1}^{\infty}$ is a set of eigenvalues and $\{\phi_j\}_{j=1}^{\infty}$ is the corresponding set of eigenfunctions. Assume that there are Π_c, $c \in \{1, \ldots, C\}$, populations, where C is an integer indicating the total number of populations. Let the random variable Y be the group label of X. The prior probability that X is from Π_c is denoted by $\pi_c = P(Y = c)$. We denote by $X^{(c)}$ the X given $Y = c$. Furthermore, we define $\mu^{(c)}$ and $\Gamma^{(c)}$ as the mean and the covariance of $X^{(c)}$. The covariance $\Gamma^{(c)}$ also has an orthogonal expansion in L_2, $\Gamma^{(c)}(s,t) = \sum_{j=1}^{\infty} \lambda_j^{(c)} \phi_j^{(c)}(s)\phi_j^{(c)}(t)$. Our task will be to deduce the group label Y for a new random function X.

The Bayes classifier is a simple probabilistic classifier that assigns each object to the class with the highest conditional probability. Given an observed random function $z \in L_2(\mathcal{I})$, one

predicts the most probable class $\delta(z) \in \{1, \ldots, C\}$ of z by the following classification rule:

$$\delta(z) = \underset{c \in \{1,\ldots,C\}}{\operatorname{argmax}} \; P(Y = c) P(X = z | Y = c). \tag{3.1}$$

If the conditional probability densities of the random function X exist, we denote them as $f^{(c)}$ conditioning the group label Y on c, with $f^{(c)} > 0$. Then the classification rule would become

$$\delta(z) = \underset{c \in \{1,\ldots,C\}}{\operatorname{argmax}} \; \pi_c f^{(c)}(z). \tag{3.2}$$

However, it is clear that the above argument does not work directly as the probability density does not exist for functional data (Delaigle and Hall, 2010). To overcome the aforementioned difficulties, instead of modeling the joint distribution of Y and X to derive the Bayes classifier, we propose to model the joint distribution of Y and the L_2 distance between X and z in sizes of $\varepsilon > 0$, and then use the classifier in (3.2). The small-ball probability $P(\|X - z\| \leq \varepsilon)$ measures the concentration of X for different values of z, so it can be viewed as a "surrogate density" for X.

Now, consider the Karhunen-Loève expansion of a random function X:

$$X(t) = \mu(t) + \sum_{j=1}^{\infty} \xi_j \phi_j(t), \tag{3.3}$$

where $\xi_j = \int_{\mathcal{I}} (X(t) - \mu(t)) \phi_j(t) dt$, $j \geq 1$, are random variables with mean zero and variance λ_j. The quantities ξ_j's are termed as the functional principal component (FPC) scores corresponding to a random function X. We note that FPC scores are always uncorrelated due to the orthogonality of the ϕ_j's. Our goal is to construct the surrogate density for X, that is, we are approximating it by the small-ball probability $P(\|X - z\| \leq \varepsilon)$. A key idea in the naive Bayes classifier for functional data is that we make a strong and naive independence assumption of the ξ_j's. It is close to its analogous definition in the finite-dimensional setting that we typically assume the features are independent in classical naive Bayes classifier. We also note that this is exactly correct if X is a Gaussian process. With the aid of independence assumption of the ξ_j's, Delaigle and Hall (2010) showed that the probability of X belonging

to a ball of radius $\varepsilon > 0$ centered at z can be written as

$$p(z|\varepsilon) = P(\|X - z\| \leq \varepsilon) = \frac{(\varepsilon \pi^{1/2})^J}{\Gamma(J/2+1)} \left\{ \prod_{j=1}^{J} f_j(z_j) \right\} \exp\{o(J)\}, \tag{3.4}$$

where $J = J(\varepsilon)$ diverges to infinity as ε decreases to zero, f_j is the probability density function of ξ_j, and z_j is a projection of $z - \mu$ in the direction of the j-th eigenfunction ϕ_j, i.e., $z_j = \int_{\mathcal{I}} (z(t) - \mu(t))\phi_j(t)dt$, $j \geq 1$, for any square-integrable function z on $\mathcal{I}$. We denote the surrogate density function of X by $f(\cdot|\varepsilon)$ indicating it is related to the choice of ε. It is certainly desirable that, though the density function of X does not exist, the surrogate density function of X satisfy the probability function

$$P(\|X - z\| \leq \varepsilon) = \int_{\mathcal{D}_z} f(u|\varepsilon)du, \tag{3.5}$$

where $\mathcal{D}_z = \{u \in L_2(\mathcal{I}) : \|u - z\| \leq \varepsilon\}$. For a small $\varepsilon > 0$, the relation between (3.4) and (3.5) suggests that

$$f(z|\varepsilon) \propto \prod_{j=1}^{J} f_j(z_j). \tag{3.6}$$

Above equation (3.6) implies that the product of probability density functions f_j, for $j = 1, \ldots J$, can be an approximation for the surrogate density $f(\cdot|\varepsilon)$ of the random function X. In practice the densities f_j's are estimated nonparametrically via kernel density estimation methods. The effective dimension of J for a given value of scale ε will be discussed in Section 3.2.

To estimate the conditional probability densities $f_j^{(c)}$, we need to extract $\xi_j^{(c)}$ from $X^{(c)}$. However, if we simply calculate $\xi_j^{(c)}$ from $\int_{\mathcal{I}} (X^{(c)}(t) - \mu^{(c)}(t))\phi_j^{(c)}(t)dt$, this will lead to a different scale of $\xi_j^{(c)}$, for $c = 1, \ldots, C$, due to the different means and eigenfunctions for each group. It is sensible to subtract $X^{(c)}$ from the overall mean and then project the data onto the same basis. Here we adopt the idea of common principal components from the multivariate analysis. We write the overall mean function $\mu(t) = \sum_{c=1}^{C} \pi_c \mu^{(c)}(t)$ and impose the common eigenfunction assumption on all populations, that is, $\phi_j^{(1)} = \phi_j^{(2)} = \cdots = \phi_j^{(C)} := \phi_j$, for $j \geq 1$. This assumption helps to avoid the comparisons of the modes of variation between different

bases. Under this assumption we can write $\Gamma^{(c)}(s,t) = \sum_{j=1}^{\infty} \lambda_j^{(c)} \phi_j(s)\phi_j(t)$, where the ϕ_j's are the common eigenfunctions. We then define the common covariance function $\Gamma(s,t) = \sum_{c=1}^{C} \pi_c \Gamma^{(c)}(s,t)$. Then ϕ_j is also the j-th eigenfunction of Γ with the pooled eigenvalue $\lambda_j = \sum_{c=1}^{C} \pi_c \lambda_j^{(c)}$. The functional common principal component (FCPC) score under group c is calculated by $\xi_j^{(c)} = \int_{\mathcal{T}} (X^{(c)}(t) - \mu(t))\phi_j(t)dt$, and the conditional probability density $f_j^{(c)}$ is then estimated from the FCPC score $\xi_j^{(c)}$. We note that the FCPC score is not the FPC score in general.

The criterion function for functional naive Bayes classifier can be rewritten as

$$\delta_J(z) = \underset{c \in \{1,\ldots,C\}}{\operatorname{argmax}} \pi_c \prod_{j=1}^{J} f_j^{(c)}(z_j), \tag{3.7}$$

where $f_j^{(c)}$ is the density of j-th FCPC score under group c. This criterion function is also known as the maximum a posteriori (MAP) in Bayesian statistics. Despite the fact that the independence assumption is usually false and the probability estimates of naive Bayes are of low quality, given that we use it only to make the classification and not to accurately predict the actual probabilities, the classification decisions are still quite good. This is also pointed out in Delaigle and Hall (2010) that the small-ball probability in (3.4) can capture the variation with z. Thus, we classify z into population Π_c if and only if the criterion function $\delta_J(z)$ is maximized at the group c. The theoretical properties of the above criterion (3.7) will be discussed in Section 3.2.

3.1.1 Model Estimation

We now consider how the prior probabilities π_c and the conditional probability density functions $f_j^{(c)}$ can be estimated from data. Suppose we have training samples that consist of examples (y_i, x_i) for $i = 1, \ldots, n$. The prior probability π_c can be simply estimated by $\hat{\pi}_c = n_c/n$, where n_c is the number of times that the label c is seen in the training set, i.e., $n_c = \sum_{i=1}^{n} 1\{y_i = c\}$, where 1 is an indicator function to be 1 if $y_i = c$, 0 otherwise. As noted in Dai et al. (2017), when constructing a classifier for functional data, it is often unrealistic

to have fully observed trajectories. In practice, we can only observe a random function at discretized time points rather then the entire trajectory. We denote the discrete observations by $x_{il} = x_i(t_{il})$, where t_{il} is the l-th recording time of the i-th curve in ascending order, $l = 1, \ldots, m_i$; m_i is the number of observations for the i-th curve. Also, the sampled random function is often contaminated with measurement errors. In this case, a presmoothing step is performed on the discrete observations, then we treat the smoothed trajectory as a fully observed trajectory.

Since the overall mean function $\mu(t)$ and the common eigenfunctions $\phi_j(t)$ play a central role in the calculation of the FCPC scores $\xi_j^{(c)}$ for $c = 1, \ldots, C$, we wish to construct estimators $\hat{\mu}$ and $\hat{\phi}_j$ of μ and ϕ_j, respectively. We briefly summarized the procedure as follows. The overall mean function $\mu(t)$ is estimated by $\hat{\mu}(t) = \sum_{c=1}^{C} \hat{\pi}_c \hat{\mu}^{(c)}(t)$, where $\hat{\mu}^{(c)}(t)$ is the estimated mean function under group c. We apply locally weighted polynomial regression (Fan and Gijbels, 1996) to the curves in a given group c:

$$\min_{(\alpha_0,\alpha_1)} \left(\sum_{i=1}^{n_c} \sum_{l=1}^{m_i} \left\{ x_{il}^{(c)} - \alpha_0 - \alpha_1(t_{il} - t) \right\}^2 K_{h_\mu^c}(t_{il} - t) \right),$$

where $K_{h_\mu^c}(\cdot)$ is a known kernel with bandwidth h_μ^c. Take $\hat{\mu}^{(c)}(t) = \hat{\alpha}_0$. Similarly, the common covariance $\Gamma(s,t)$ is estimated by $\widehat{\Gamma}(s,t) = \sum_{c=1}^{C} \hat{\pi}_c \widehat{\Gamma}^{(c)}(s,t)$, where $\widehat{\Gamma}^{(c)}(s,t)$ is the estimated covariance function under group c. We apply two-dimensional scatterplot smoothing to the raw estimates $\Gamma_i^{(c)}(t_{ij}, t_{il})$, where $\Gamma_i^{(c)}(t_{ij}, t_{il}) = (x_{ij}^{(c)} - \hat{\mu}(t_{ij}))(x_{il}^{(c)} - \hat{\mu}(t_{il}))$. To address the problem of measurement errors, only the off-diagonal elements of $\Gamma_i^{(c)}(t_{ij}, t_{il})$, for $j \neq l$, are included in the smoothing step. The covariance estimate is then obtained by fitting a local linear plane,

$$\min_{(\alpha_0,\alpha_1,\alpha_2)} \left(\sum_{i=1}^{n_c} \sum_{1 \leq j \neq l \leq m_i} \left\{ \Gamma_i^{(c)}(t_{ij}, t_{il}) - \alpha_0 - \alpha_1(t_{ij} - s) - \alpha_2(t_{il} - t) \right\}^2 K_{h_\Gamma^c}(s - t_{ij}, t - t_{il}) \right),$$

where $K_{h_\Gamma^c}(\cdot, \cdot)$ is a two-dimensional non-negative kernel function with bandwidth h_Γ^c. Set $\widehat{\Gamma}^{(c)}(s,t) = \hat{\alpha}_0$. The estimates of the common eigenfunctions and the pooled eigenvalues

correspond to the solution $\hat{\phi}_j$ and $\hat{\lambda}_j$ of the eigenequations:

$$\int_{\mathcal{I}} \widehat{\Gamma}(s,t)\hat{\phi}_j(s)ds = \hat{\lambda}_j\hat{\phi}_j(t),$$

subject to $\int_{\mathcal{I}} \hat{\phi}_j^2(t)dt = 1$ for $j = 1, 2, \ldots,$ and $\int_{\mathcal{I}} \hat{\phi}_j(t)\hat{\phi}_k(t)dt = 0$ for $j < k$. We note that the consistency of $\hat{\mu}^{(c)}$ and $\widehat{\Gamma}^{(c)}$ will be discussed in Section 3.2, and the consistency of $\hat{\mu}$ and $\widehat{\Gamma}$ are obtained as a consequence result of the convergence of $\hat{\mu}^{(c)}$ and $\widehat{\Gamma}^{(c)}$ and the convergence of $\hat{\pi}_c$ to π_c. The rates of convergence for $\hat{\phi}_j$ are the same as the rate for $\widehat{\Gamma}$.

Denoting $\xi_{ij}^{(c)}$ as the j-th FCPC score of the i-th observation under group c, then $\xi_{ij}^{(c)}$ is estimated by $\hat{\xi}_{ij}^{(c)} = \int_{\mathcal{I}}(x_i^{(c)}(t) - \hat{\mu}(t))\hat{\phi}_j(t)dt$, for $j = 1, \ldots, J$, via numerical approximation, $\hat{\xi}_{ij}^{(c)} = \sum_{l=1}^{m_i}(x_{il}^{(c)} - \hat{\mu}(t_{il}))\hat{\phi}_j(t_{il})\Delta t_{il}$. Note that $\hat{\xi}_{ij}^{(c)}$ are the empirical versions of $\xi_{ij}^{(c)}$. The nonparametric estimates of the densities for $\xi_{ij}^{(c)}$ are then obtained by Nadaraya-Watson kernel method. The kernel density estimate for the j-th FCPC score in the group c is given by

$$\hat{f}_j^{(c)}(u) = \frac{1}{n_c h_j^c}\sum_{i=1}^{n_c} K\left(\frac{u - \hat{\xi}_{ij}^{(c)}}{h_j^c}\right), \tag{3.8}$$

where $u \in \mathbb{R}$, K is a kernel function, and h_j^c is a bandwidth. For feasibility reason, the bandwidth h_j^c is chosen by a cross-validation procedure. Thus, the bandwidth h_j^c will be the optimal bandwidth that minimizes the asymptotic mean integrated squared error. See Ferraty and Vieu (2006).

Now, we write $\hat{z}_j = \int_{\mathcal{I}}(z(t) - \hat{\mu}(t))\hat{\phi}_j(t)dt$ for any function $z \in L_2(\mathcal{I})$. The kernel density based criterion function for functional naive Bayes classifier (3.7) is thus

$$\hat{\delta}_J(z) = \underset{c\in\{1,\ldots,C\}}{\operatorname{argmax}}\ \hat{\pi}_c \prod_{j=1}^{J} \hat{f}_j^{(c)}(\hat{z}_j). \tag{3.9}$$

We classify a newly observed random function X^* with unknown label Y^* by setting $Y^* = \hat{\delta}_J(X^*)$. In the next section, we study the asymptotic properties of the classifier $\hat{\delta}_J$, and show that it is a consistent estimator of δ_J when n goes to infinity.

3.2 Theoretical Results

In this section we examine theoretical properties of the functional naive Bayes classifier. We begin with the consistency of the estimated overall mean function and the estimated common covariance function. Under the regularity conditions as stated in Yao et al. (2005), we obtain uniform convergence rates for $\hat{\mu}^{(c)}(t)$ of $\mu^{(c)}(t)$ and $\widehat{\Gamma}^{(c)}(s,t)$ of $\Gamma^{(c)}(s,t)$. The consistency of $\hat{\mu}(t)$ for $\mu(t)$ and that of $\widehat{\Gamma}(s,t)$ for $\Gamma(s,t)$ are the immediate corollary of the consistency of $\hat{\pi}_c$ for π_c. Without loss of generality, we make the following simplifications in the theoretical analysis. We assume there is an equal prior probability that an observation is from Π_c, and there is an equal number of observations in each population, i.e., $\pi_1 = \pi_2 = \cdots = \pi_C$ and $n_1 = n_2 = \cdots = n_C$.

Proposition 1. *Assume that the regularity conditions holds on the design points, the kernel functions, the bandwidths of the kernel functions, and the moments of X are listed as (A1.1)-(A4) and (B1.1)-B(2.2) in Yao et al. (2005). Given the group memberships of the observed curves, for each group c, $c = 1, \ldots, C$, the estimated mean function and the estimated covariance function satisfy the uniform consistency properties such that*

$$\sup_{t \in \mathcal{I}} |\hat{\mu}^{(c)}(t) - \mu^{(c)}(t)| = O_p(\tau_n^c) \tag{3.10}$$

and

$$\sup_{s,t \in \mathcal{I}} |\widehat{\Gamma}^{(c)}(s,t) - \Gamma^{(c)}(s,t)| = O_p(\gamma_n^c) \tag{3.11}$$

with the sequences $\tau_n^c = (n_c^{1/2} h_\mu^c)^{-1} \to 0$ and $\gamma_n^c = (n_c^{1/2} (h_\Gamma^c)^2)^{-1} \to 0$, as $n_c \to \infty$, where h_μ^c and h_Γ^c are the bandwidths for estimating mean and covariance functions, respectively.

Proof. The uniform consistency of $\hat{\mu}^{(c)}$ and $\widehat{\Gamma}^{(c)}$ in Proposition 1 are obtained as a consequence of applying *Theorem 1* in Yao et al. (2005) to each group c. □

Proposition 2. *Under the assumptions stated in Proposition 1 the estimated overall mean function and the estimated common covariance function satisfy the uniform consistency prop-*

erties such that

$$\sup_{t\in\mathcal{I}} |\hat{\mu}(t) - \mu(t)| = O_p(\alpha_n) \tag{3.12}$$

and

$$\sup_{s,t\in\mathcal{I}} |\widehat{\Gamma}(s,t) - \Gamma(s,t)| = O_p(\eta_n). \tag{3.13}$$

Furthermore, the common eigenfunctions satisfy the uniform consistency properties such that

$$\sup_{t\in\mathcal{I}} |\hat{\phi}_j(t) - \phi_j(t)| = O_p(\eta_n) \tag{3.14}$$

for fixed j, $j = 1, 2, \ldots,$ with $\alpha_n = \sum_{c=1}^{C} \tau_n^c \to 0$ and $\eta_n = \sum_{c=1}^{C} \gamma_n^c \to 0$, as $n \to \infty$.

Proof. We first establish the consistency of $\hat{\pi}_c$ for π_c. The rate of convergence for $\hat{\pi}_c$ can be easily derived by Hoeffding's inequality. For $\epsilon > 0$, we have

$$P(|\hat{\pi}_c - \pi_c| > \epsilon) \leq 2e^{-2n\epsilon^2}.$$

Let $\delta = 2e^{-2n\epsilon^2}$, then $\epsilon = \sqrt{\frac{1}{2n}\log\left(\frac{2}{\delta}\right)}$. The above inequality yields

$$P\left(\left|\hat{\pi}_c - \pi_c\right| \leq \sqrt{\frac{1}{2n}\log\left(\frac{2}{\delta}\right)}\right) > 1 - \delta,$$

which implies that

$$|\hat{\pi}_c - \pi_c| = O_p(n^{-1/2}). \tag{3.15}$$

The rate of convergence for the overall mean function $\hat{\mu}(t)$ can be derived as follows:

$$\begin{aligned}
\sup_{t\in\mathcal{I}} |\hat{\mu}(t) - \mu(t)| &= \sup_{t\in\mathcal{I}} \left| \sum_{c=1}^{C} \left(\hat{\pi}_c \hat{\mu}^{(c)}(t) - \pi_c \mu^{(c)}(t)\right) \right| \\
&\leq \sup_{t\in\mathcal{I}} \sum_{c=1}^{C} \left|\hat{\pi}_c \big(\hat{\mu}^{(c)}(t) - \mu^{(c)}(t)\big)\right| + \sup_{t\in\mathcal{I}} \sum_{c=1}^{C} \left|(\hat{\pi}_c - \pi_c)\mu^{(c)}(t)\right| \\
&\leq \sum_{c=1}^{C} \sup_{t\in\mathcal{I}} \left|\hat{\mu}^{(c)}(t) - \mu^{(c)}(t)\right| + \sum_{c=1}^{C} \left|\hat{\pi}_c - \pi_c\right| \sup_{t\in\mathcal{I}} \left|\mu^{(c)}(t)\right| \\
&= O_p\left(\sum_{c=1}^{C} \tau_n^c\right) + O_p(n^{-1/2}) = O_p(\alpha_n),
\end{aligned}$$

where the second equality follows from (3.10) and (3.15) with a constant C and $\sup_{t\in\mathcal{I}}|\mu^{(c)}(t)|$ bounded for all $c = 1, \ldots, C$. The last equality is guaranteed by

$$\sum_{c=1}^{C} \tau_n^c = \sum_{c=1}^{C} \frac{1}{\sqrt{n_c} h_\mu^c} \geq \sum_{c=1}^{C} \frac{1}{\sqrt{n_c}} \frac{1}{\max_c h_\mu^c} \geq \frac{1}{\sqrt{n}} \frac{C}{\max_c h_\mu^c}.$$

The rate of $\sum_{c=1}^{C} \tau_n^c$ term is slower than that of $n^{-1/2}$ term. The rate of convergence for the common covariance function $\widehat{\Gamma}(s,t)$ in (3.13) can be derived analogously,

$$\begin{aligned}
\sup_{s,t\in\mathcal{I}} |\widehat{\Gamma}(s,t) - \Gamma(s,t)| &= \sup_{s,t\in\mathcal{I}} \left| \sum_{c=1}^{C} \left(\hat{\pi}_c \widehat{\Gamma}^{(c)}(s,t) - \pi_c \Gamma^{(c)}(s,t) \right) \right| \\
&\leq \sup_{s,t\in\mathcal{I}} \sum_{c=1}^{C} \left| \hat{\pi}_c \big(\widehat{\Gamma}^{(c)}(s,t) - \Gamma^{(c)}(s,t) \big) \right| + \sup_{t\in\mathcal{I}} \sum_{c=1}^{C} \left| (\hat{\pi}_c - \pi_c) \Gamma^{(c)}(s,t) \right| \\
&\leq \sum_{c=1}^{C} \sup_{s,t\in\mathcal{I}} \left| \widehat{\Gamma}^{(c)}(t) - \Gamma^{(c)}(s,t) \right| + \sum_{c=1}^{C} \left| \hat{\pi}_c - \pi_c \right| \sup_{s,t\in\mathcal{I}} \left| \Gamma^{(c)}(s,t) \right| \\
&= O_p \left(\sum_{c=1}^{C} \gamma_n^c \right) + O_p(n^{-1/2}) = O_p(\eta_n),
\end{aligned}$$

and the rate (3.14) is direct consequences of the rate (3.13). □

To investigate the properties of surrogate densities $f^{(c)}(\cdot|\varepsilon)$ for each group $c = 1, \ldots, C$, we make the following assumptions (A1)-(A4), which are parallel to assumptions (3.6)-(3.9) in Delaigle and Hall (2010). These assumptions are made analogous to conditions A1.-A4. in Dai et al. (2017) for two populations. We extend these assumptions to more general case of C populations.

(A1) For all large $D > 0$ and some $\delta > 0$, $\sup_{t\in\mathcal{I}} E\{|X^{(c)}(t)|^D\} < \infty$ and $\sup_{s,t\in\mathcal{I}: s\neq t} E[\{|s-t|^{-\delta}|X^{(c)}(s) - X^{(c)}(t)|\}^D] < \infty$;

(A2) For each integer $r \geq 1$, $(\lambda_j^{(c)})^{-r} E\{\int_{\mathcal{I}} (X^{(c)}(t) - \mu(t))\phi_j(t)dt\}^{2r}$ is bounded uniformly in j;

(A3) The eigenvalues in each of the sequences $\{\lambda_j^{(c)}\}_{j=1}^{\infty}$ are all different, and so are the pooled eigenvalues $\{\lambda_j\}_{j=1}^{\infty}$;

(A4) The density of the j-th standardized FCPC score is bounded and has a bounded derivative; the kernel K is a symmetric, compactly supported probability density function with two bounded derivatives; for some $\delta > 0$, $h_j^c = h_j(n_c) = O(n_c^{-\delta})$ and $n_c^{1-\delta}(h_j^c)^3$ is bounded away from zero as $n_c \to \infty$.

Proposition 3. *Assuming (A1)-(A4) hold and FCPC scores are independent, the surrogate density of each group $f^{(c)}(\cdot|\varepsilon)$ can be approximated by the product of probability density functions $f_j^{(c)}$, for $j = 1, \ldots, J$ with $J = J(\varepsilon) \to \infty$ as $\varepsilon \to 0$.*

$$f^{(c)}(x|\varepsilon) \propto \prod_{j=1}^{J} f_j^{(c)}(x_j). \tag{3.16}$$

Proof. Under assumptions (A1)-(A4) for each group c this is an immediately result from Delaigle and Hall (2010) and (3.4) to (3.6). □

As noted in Delaigle and Hall (2010), the choice of J depends on the rate of convergence of the sequence of eigenvalues λ_j to zero. We list below the effective dimension J for a given value of scale ε associated with the different eigenvalues decays:

- if $\{\lambda_j\}_{j=1}^{\infty}$ decays exponentially

  $$\lambda_j^{-1} \sum_{k \geq j+1} \lambda_k \text{ is bounded as } j \to \infty,$$

  then we take J to be the unique integer for which $\lambda_{J+1} < \varepsilon^2 < \lambda_J$.

- if $\{\lambda_j\}_{j=1}^{\infty}$ decays super-exponentially

  $$\frac{\lambda_{j+1}}{\lambda_j} \to 0 \text{ or, equivalently, } \lambda_j^{-1} \sum_{k \geq j+1} \lambda_k \to 0,$$

  then we take J such that the value of ε^2/λ_J is sufficiently close to 1.

From a practical point of view, the eigenvalue sequence λ_j decreases to zero at an exponential rate is essential in determining the effectiveness of J. In general, the integer J acts as the dimension of the approximated small-ball probability $p(x|\varepsilon)$ at scale ε. In other word, the

value of J can be interpreted as the dimension of the scale space when the unit of scale is ε. Moreover, for a given scale ε the case of eigenvalues that decay exponentially suggests that the effective dimension J can be chosen such that

$$\varepsilon^2 \approx \lambda_J \tag{3.17}$$

In practice, for a predetermined ε, we choose the value of J such that the value of ε^2 is approximately equal to $\hat{\lambda}_J$. Alternatively, one can also simply increases the value of J to assess how the classification performance changes as J increases and determine the optimal value of J empirically.

Next we investigate the asymptotic properties of the FNB classifier $\hat{\delta}_J$ defined in (3.9). We first introduce an useful lemma that states the asymptotic equivalence of the estimated kernel density function to the true one. Let $\mathcal{S}(\alpha) = \{x : \|x\| \leq \alpha\}$ be a bounded set of all square integrable functions for $\alpha > 0$.

Lemma 2. *Assuming (A1)-(A4) hold, for any $j \geq 1$ and $c = 1, \ldots, C$,*

$$\sup_{x \in \mathcal{S}(\alpha)} |\hat{f}_j^{(c)}(\hat{x}_j) - f_j^{(c)}(x_j)| = O_p\left(h_j^c + \left(\frac{n_c h_j^c}{\log n_c}\right)^{-\frac{1}{2}}\right). \tag{3.18}$$

This lemma may essentially be found in Dai et al. (2017). Because it plays such a central role in this paper, we give the proof here since it is brief.

Proof. Let $g_j^{(c)}$ be the density function of the standardized FCPC scores and $\hat{g}_j^{(c)}$ be the kernel density estimates of $g_j^{(c)}$ using the estimated standardized FCPC scores. Without loss of generality, we assume the standardized FCPC scores have zero mean. This assumption can

be easily relaxed by subtracting the standardized FCPC scores from its mean.

$$
\begin{aligned}
&\sup_{x\in\mathcal{S}(\alpha)} |\hat{f}_j^{(c)}(\hat{x}_j) - f_j^{(c)}(x_j)| = \sup_{x\in\mathcal{S}(\alpha)} \left| \frac{1}{\sqrt{\hat{\lambda}_j^{(c)}}} \hat{g}_j^{(c)}(\frac{\hat{x}_j}{\sqrt{\hat{\lambda}_j^{(c)}}}) - \frac{1}{\sqrt{\lambda_j^{(c)}}} g_j^{(c)}(\frac{x_j}{\sqrt{\lambda_j^{(c)}}}) \right| \\
\leq & \sup_{x\in\mathcal{S}(\alpha)} \left(\frac{1}{\sqrt{\hat{\lambda}_j^{(c)}}} \left| \hat{g}_j^{(c)}(\frac{\hat{x}_j}{\sqrt{\hat{\lambda}_j^{(c)}}}) - g_j^{(c)}(\frac{x_j}{\sqrt{\lambda_j^{(c)}}}) \right| + g_j^{(c)}(\frac{x_j}{\sqrt{\lambda_j^{(c)}}}) \left| \frac{1}{\sqrt{\hat{\lambda}_j^{(c)}}} - \frac{1}{\sqrt{\lambda_j^{(c)}}} \right| \right) \\
= & O_p \left(\sup_{x\in\mathcal{S}(\alpha)} \left| \hat{g}_j^{(c)}(\frac{\hat{x}_j}{\sqrt{\hat{\lambda}_j^{(c)}}}) - g_j^{(c)}(\frac{x_j}{\sqrt{\lambda_j^{(c)}}}) \right| \right) + O_p \left(\left| \frac{1}{\sqrt{\hat{\lambda}_j^{(c)}}} - \frac{1}{\sqrt{\lambda_j^{(c)}}} \right| \right) \\
= & O_p \left(h_j^c + \left(\frac{n_c h_j^c}{\log n_c} \right)^{-\frac{1}{2}} \right).
\end{aligned}
$$

□

Theorem 2. *Assuming all populations have the common eigenfunctions, FCPC scores are independent, (A1)-(A4) hold, and the value of J is chosen by* (3.17)*. The criterion function for FNB classifier $\hat{\delta}_J$ is an M-estimator and $P(\hat{\delta}_J(X) \neq \delta_J(X)) \to 0$ as $n \to \infty$.*

Proof. We define

$$
M_n\big(\delta(X)\big) = \hat{\pi}_\delta \prod_{j=1}^{J} \hat{f}_j^{(\delta)}(\hat{x}_j) \quad \text{and} \quad M\big(\delta(X)\big) = \pi_\delta \prod_{j=1}^{J} f_j^{(\delta)}(x_j). \tag{3.19}
$$

It is clear that the criterion function for FNB classifier is to find an estimator $\delta(X)$ such that $M_n(\delta(X))$ can be maximized over the parameter space $\Delta = \{1, \ldots, C\}$. So $\hat{\delta}_J(X)$ is an M-estimator.

We note that

$$
\begin{aligned}
&\sup_{\delta_J\in\Delta} |M_n\big(\delta_J(X)\big) - M\big(\delta_J(X)\big)| \leq \sup_{\delta_J\in\Delta}\sup_{x\in\mathcal{S}(\alpha)} |M_n\big(\delta_J(x)\big) - M\big(\delta_J(x)\big)| \\
\leq\ & \sup_{\delta_J\in\Delta}\sup_{x\in\mathcal{S}(\alpha)} \left|\hat{\pi}_{\delta_J}\prod_{j=1}^{J}\hat{f}_j^{(\delta_J)}(\hat{x}_j) - \pi_{\delta_J}\prod_{j=1}^{J} f_j^{(\delta_J)}(x_j)\right| \\
\leq\ & \sup_{\delta_J\in\Delta}\sup_{x\in\mathcal{S}(\alpha)} \left|\hat{\pi}_{\delta_J}\prod_{j=1}^{J}\left(\hat{f}_j^{(\delta_J)}(\hat{x}_j) - f_j^{(\delta_J)}(x_j)\right)\right| + \sup_{\delta_J\in\Delta}\left|\hat{\pi}_{\delta_J} - \pi_{\delta_J}\right| \sup_{x\in\mathcal{S}(\alpha)}\left|\prod_{j=1}^{J} f_j^{(\delta_J)}(x_j)\right| \\
\leq\ & \sup_{\delta_J\in\Delta}\prod_{j=1}^{J}\sup_{x\in\mathcal{S}(\alpha)}\left|\hat{f}_j^{(\delta_J)}(\hat{x}_j) - f_j^{(\delta_J)}(x_j)\right| + \sup_{\delta_J\in\Delta}\left|\hat{\pi}_{\delta_J} - \pi_{\delta_J}\right| \sup_{x\in\mathcal{S}(\alpha)}\prod_{j=1}^{J}\left|f_j^{(\delta_J)}(x_j)\right| \\
\leq\ & \sup_{\delta_J\in\Delta}\prod_{j=1}^{J} O_p\left(h_j^{\delta_J} + \left(\frac{n_{\delta_J}h_j^{\delta_J}}{\log n_{\delta_J}}\right)^{-\frac{1}{2}}\right) + O_p(n^{-1/2}).
\end{aligned}
$$

The above inequality suggests that

$$\sup_{\delta_J\in\Delta} |M_n\big(\delta_J(X)\big) - M\big(\delta_J(X)\big)| \xrightarrow{p} 0 \ \text{ as } \ n\to\infty.$$

We note that this is a Glivenko-Cantelli class. By the properties of the $\delta_J(X)$, it attains the maximum of M. So for any $\delta_J^{'}(X) \neq \delta_J(X)$, we have

$$\sup_{\delta_J^{'}\in\Delta} M\big(\delta_J^{'}(X)\big) < M\big(\delta_J(X)\big).$$

Then for any estimator $\hat{\delta}_J(X)$ attains the maximum of M_n, we have

$$M_n\big(\hat{\delta}_J(X)\big) \geq M_n\big(\delta_J(X)\big) - o_p(1).$$

Now by the argmax continuous mapping theorem (Vaart, 1998), we conclude that $\hat{\delta}_J(X)$ converges to $\delta_J(X)$ in probability. In other words, $P(\hat{\delta}_J(X) \neq \delta_J(X)) \to 0$ as $n\to\infty$.

□

Theorem 1 provides the asymptotic equivalence of the estimated version of the functional naive Bayes classifier based on the kernel density estimates to the true one.

3.3 Simulation Study and Data Application

3.3.1 Simulation Study

To examine the performance of the proposed method a simulation study was conducted. We consider several simulation settings for various training and testing datasets. Furthermore, we compare the presented method with other algorithms by computing classification error rates. All computations are done with the software MATLAB. We compared the functional naive Bayes (FNB) approach with the following methods:

- Multinomial logistic regression (logistic): The subroutine "mnrfit" is used to fit multinomial logistic regression. The estimated FCPC scores are treated as predictor variables and the group labels are treated as response variables.

- k-nearest neighbors (k-NN): The subroutine "fitcknn" is used to train the k-NN model with the classic L^2 metric, see Ferraty and Vieu (2006) for example. The number of neighbors k were determined by 10-fold cross validations.

- Linear discriminant analysis (LDA): The subroutine "fitcdiscr" is used to train the LDA model, which determine a linear discriminant that classifies FCPC.

- Support vector machines (SVM): the package "libsvm" is used for the SVM with the choice of the RBF kernel for which the cost and gamma parameters were left as the default. The package is available at https://www.csie.ntu.edu.tw/~cjlin/libsvm/.

We note that from the functional data point of view, each datum in functional data is a realization of a sample from an underlying stochastic process. However, in the real world, the data were discretely collected over time, so we can only observe some points that form a vector rather than the entire trajectory. We treat the functional data as a vector when running the SVM.

In order to evaluate the performance of the FNB, we generate synthetic data trajectories based on the model

$$X_{il}^{(c)} = \mu^{(c)}(t_{il}) + \sum_{j=1}^{J} \xi_{ij}^{(c)} \phi_j(t_{il}) + \epsilon_{il}^{(c)},$$

where $i = 1, \ldots, n_c$ and n_c is the number of samples for each group, $c = 1, \ldots, C$. We set $C = 6$ and $J = 20$ for all simulations. The variates $\xi_{ij}^{(c)}$ are generated from independent and identically distributed $N(0, \lambda_j^{(c)})$, where $\lambda_j^{(c)}$ are eigenvalues corresponding to ϕ_j for class c, and the measurement errors $\epsilon_{il}^{(c)}$ are generated from independent and identically distributed $N(0, \sigma^2)$. The basis for the linear combination of functions was specified as the Fourier basis, where the ϕ_j's take the following form: $\phi_1(t) = 1$, $\phi_{2r}(t) = \sqrt{2}\cos(2r\pi t)$, $\phi_{2r+1}(t) = \sqrt{2}\sin(2r\pi t)$, for $r = 1, 2, \ldots$, and $t \in [0, 1]$. The time points are generated from a regular design on 51 equally spaced time points from 0 to 1. Various combinations of the mean function $\mu^{(c)}(t)$ and eigenvalues $\lambda_j^{(c)}$ between groups are described below:

- Model 1: $\mu^{(c)}(t)$ are different and $\lambda_j^{(c)}$ are the same:

  $\mu^{(1)}(t) = 0$, $\mu^{(2)}(t) = \sqrt{2}\sin(4\pi t)$, $\mu^{(3)}(t) = (t + 0.75)^{-3}$,
  $\mu^{(4)}(t) = 2\cos(2\pi t)$, $\mu^{(5)}(t) = 2 - 4\exp(-6t)$, $\mu^{(6)}(t) = -3(t - 0.5)$,
  and $\lambda_j^{(c)} = \exp(-\frac{j}{3})$, for $c = 1, \ldots C$.

- Model 2: $\mu^{(c)}(t)$ are the same and $\lambda_j^{(c)}$ are different:

  $\mu^{(c)}(t) = 0$, for $c = 1, \ldots, C$,
  $\lambda_j^{(1)} = \exp(-j)$, $\lambda_j^{(2)} = \exp(-\frac{1}{2}j)$, $\lambda_j^{(3)} = \exp(-\frac{1}{3}j)$,
  $\lambda_j^{(4)} = \exp(-\frac{2}{3}j)$, $\lambda_j^{(5)} = \exp(-\frac{1}{4}j)$, $\lambda_j^{(6)} = \exp(-\frac{3}{4}j)$.

- Model 3: $\mu^{(c)}(t)$ and $\lambda_j^{(c)}$ are different for all groups:
  the mean functions $\mu^{(c)}(t)$ were chosen as described in Model 1 and the eigenvalues $\lambda_j^{(c)}$ were chosen as described in Model 2.

We remark that the mean functions $\mu(t)$ in Model 1 were used in Coffey et al. (2014), which were chosen to reflect the real-life situation for time-course gene expression data. The

variance of measurement error is set to be $\sigma^2 = 0.01$ to each observation for all Models. In each setting we generated sample sizes of $n = 60$, 120, and 300 for the training set, and sample sizes of $m = 120$, 300, and 600 for the testing set. Each simulated trajectory has $1/C$ probability to belong to group c. Examples for Model 1, Model 2, and Model 3 are plotted in Figure 3.1, Figure 3.2, and Figure 3.3, respectively. This simulation was repeated 200 times.

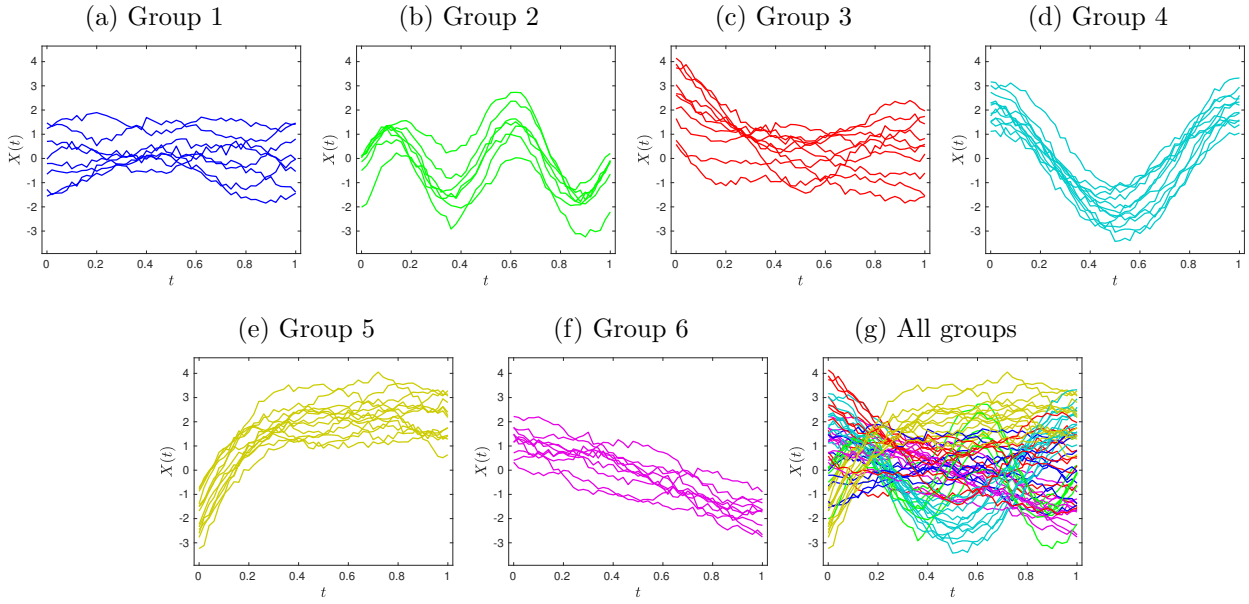


Figure 3.1: Example dataset generated from Model 2 for the simulation study.

The sample means with associated standard errors of the misclassification error rate are presented in Table 3.1 and Table 3.2 with $n = 60, 120, 300$ and $m = 120, 300, 600$ based on 200 simulation replicates for three different models. We explore the behavior of the FNB with $J = 1$, 5, 10 and 20. For Model 1, it is clear that LDA has the best performance and FNB is the second best when $J \geq 10$. This result is not surprising since the LDA created maximizes the differences between groups, when the mean functions are different, the LDA can separate groups very well. For Model 2 and 3, it is clear that FNB has consistently outperformed all other methods for $J \geq 10$. When the mean functions are different (Model 1 and Model 3) the misclassification error rate decreases as the training sample size n increases for all the

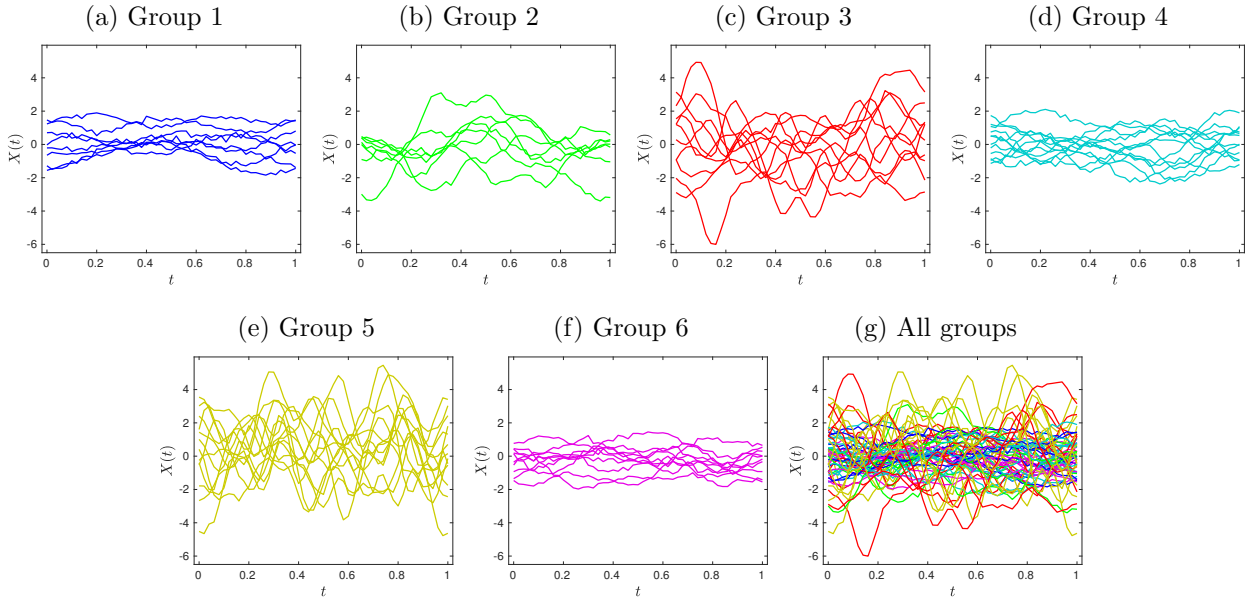


Figure 3.2: Example dataset generated from Model 2 for the simulation study.

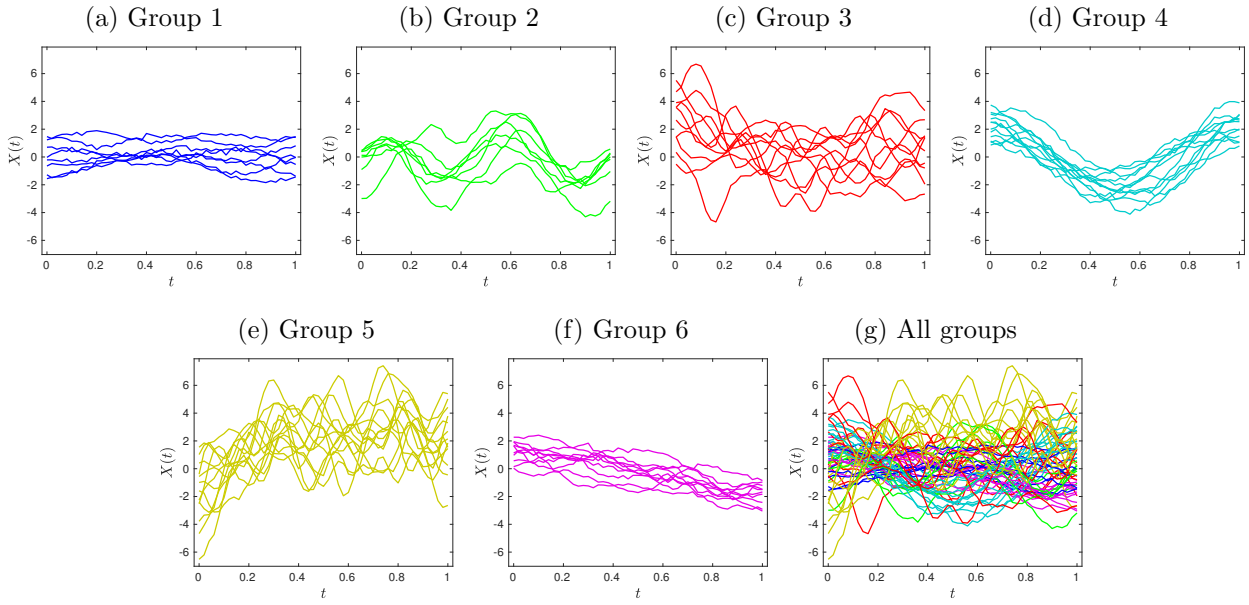


Figure 3.3: Example dataset generated from Model 3 for the simulation study.

methods. On the other hand, when the mean functions are the same (Model 2) the FNB is the only method for which its misclassification error decreases as the training sample size n increases. We observe that the misclassification rate of FNB decreases as the number of n

increases given a fixed number of components J. This illustrates well the consistency result stated in Theorem 1.

The boxplot in Figure 3.4 displays the information about variation of misclassification error rate for Model 1. FNB with $J = 1$ has the largest bias, then this bias is dramatically reduced as J increases. In general when $J \geq 10$, FNB performs the second best for Model 1 among the methods considered. We note that the performance of SVM relies heavily on the choice of the kernel as well as the tuning parameters. There are some situations where more than 80% of the testing data are misclassified when the kernels are misspecified. When the mean functions are different, FNB performs the best for Model 2 and Model 3. The boxplot results are shown in Figure 3.5 and Figure 3.6 for Model 2 and Model 3, respectively.

3.3.2 Data Applications

In this section we illustrate the performance of the proposed FNB approach on four real datasets that belong to very different research areas including traffic flow, fish species, leaf images, and growth curves. In order to evaluate the classification performance, the error rate from 10-fold cross-validation will be used. For each available sample 9/10 of the data are used in training the methods, and the misclassification error is then estimated on the remaining data. We repeat the process 200 times and report the mean misclassification rates, and the standard deviations of the mean estimates.

The first data example concerns with classification of daily traffic volume trends from a highway traffic flow data. We use a dataset that is archived on the Caltrans Performance Measurement System (PeMS) available through the link http://pems.dot.ca.gov/. The traffic flow rate were collected on 5-min intervals from February 8 to July 31 in 2013 (I15-S@CA PM14.3, District 8, Riverside County, City of Murrieta. Detector ID: 817371). Since the traffic flow patterns are distinct on weekdays and weekends (including holidays) and are similar between weekdays, we use only the data from weekdays in this analysis. Each daily curve in the dataset was collected by a vehicle detector at 288 equispaced instants of time in

Table 3.1: The sample means, standard errors (in parentheses) and average proportions of total variance explained [in bracket (%)] for misspecification rates in Model 1, Model 2 and Model 3 with 200 replications.

Model	n	m	Functional Naive Bayes with different J			
			1	5	10	20
1	60	120	0.717 (0.047) [64.30]	0.149 (0.052) [99.26]	0.081 (0.051) [99.90]	0.075 (0.048) [99.99]
		300	0.717 (0.032) [64.01]	0.150 (0.043) [99.26]	0.079 (0.046) [99.90]	0.074 (0.043) [99.99]
		600	0.715 (0.028) [64.07]	0.149 (0.042) [99.25]	0.079 (0.044) [99.90]	0.074 (0.042) [99.99]
	120	120	0.712 (0.046) [63.83]	0.096 (0.029) [99.22]	0.033 (0.021) [99.88]	0.032 (0.020) [99.99]
		300	0.710 (0.031) [64.02]	0.097 (0.021) [99.24]	0.034 (0.017) [99.89]	0.032 (0.014) [99.99]
		600	0.710 (0.025) [64.29]	0.096 (0.018) [99.23]	0.033 (0.013) [99.89]	0.032 (0.012) [99.99]
	300	120	0.702 (0.038) [63.93]	0.077 (0.025) [99.20]	0.028 (0.016) [99.86]	0.022 (0.014) [99.98]
		300	0.699 (0.028) [63.56]	0.077 (0.016) [99.18]	0.028 (0.011) [99.86]	0.023 (0.009) [99.98]
		600	0.699 (0.021) [63.37]	0.075 (0.011) [99.19]	0.028 (0.008) [99.86]	0.022 (0.007) [99.98]
2	60	120	0.824 (0.037) [37.81]	0.703 (0.044) [88.44]	0.589 (0.049) [98.45]	0.563 (0.053) [99.98]
		300	0.824 (0.024) [37.71]	0.703 (0.031) [88.51]	0.589 (0.033) [98.48]	0.562 (0.041) [99.98]
		600	0.823 (0.019) [37.94]	0.704 (0.026) [88.67]	0.588 (0.030) [98.51]	0.560 (0.038) [99.98]
	120	120	0.827 (0.036) [36.23]	0.682 (0.043) [86.69]	0.521 (0.049) [97.81]	0.422 (0.052) [99.97]
		300	0.822 (0.023) [36.61]	0.679 (0.031) [86.95]	0.518 (0.033) [97.88]	0.424 (0.036) [99.97]
		600	0.823 (0.019) [36.74]	0.679 (0.025) [86.82]	0.517 (0.028) [97.84]	0.424 (0.030) [99.97]
	300	120	0.816 (0.034) [35.72]	0.647 (0.047) [85.92]	0.457 (0.046) [97.44]	0.331 (0.045) [99.96]
		300	0.816 (0.024) [35.46]	0.649 (0.029) [85.71]	0.456 (0.030) [97.38]	0.330 (0.029) [99.96]
		600	0.817 (0.018) [32.52]	0.649 (0.020) [85.86]	0.455 (0.024) [97.44]	0.329 (0.024) [99.96]
3	60	120	0.741 (0.049) [37.81]	0.353 (0.065) [88.47]	0.237 (0.061) [98.47]	0.191 (0.074) [99.98]
		300	0.741 (0.037) [37.72]	0.353 (0.056) [88.55]	0.241 (0.055) [98.49]	0.192 (0.068) [99.98]
		600	0.740 (0.034) [37.95]	0.349 (0.052) [88.71]	0.239 (0.052) [98.53]	0.189 (0.066) [99.98]
	120	120	0.739 (0.046) [36.23]	0.262 (0.050) [86.72]	0.130 (0.035) [97.83]	0.063 (0.025) [99.97]
		300	0.735 (0.035) [36.61]	0.264 (0.038) [86.98]	0.127 (0.027) [97.89]	0.063 (0.020) [99.97]
		600	0.735 (0.030) [36.74]	0.258 (0.033) [86.85]	0.124 (0.023) [97.85]	0.062 (0.017) [99.97]
	300	120	0.725 (0.041) [35.72]	0.207 (0.040) [85.94]	0.080 (0.025) [97.45]	0.031 (0.017) [99.96]
		300	0.723 (0.031) [35.46]	0.204 (0.027) [85.73]	0.079 (0.015) [97.39]	0.031 (0.011) [99.96]
		600	0.726 (0.024) [35.52]	0.206 (0.022) [85.88]	0.079 (0.012) [97.45]	0.031 (0.008) [99.96]

the interval $[0, 24]$. Following the procedure described in Chiou et al. (2014b), the missing values were imputed and outliers were identified and removed from the dataset. Overall, 113 daily traffic flow patterns are analyzed: 23 of them are from Mondays, 23 of Tuesdays, 23 of Wednesdays, 21 of Thursdays and 23 of Fridays. The goal of the analysis is to classify the traffic flow patterns based on the day of the week.

The second data example is based on Lee et al. (2008)'s study of fish species recognition and migration monitoring. In their study they suggest that the shape of the fish is the most reliable general characteristic in determining its species. They developed a con-

Table 3.2: The sample means, standard errors (in parentheses) and average proportions of total variance explained [in bracket (%)] for misspecification rates in Model 1, Model 2 and Model 3 with 200 replications. (Continued)

Model	n	m	Logistic	k-NN	LDA	SVM
1	60	120	0.191 (0.051)	0.147 (0.040)	0.029 (0.020)	0.154 (0.052)
		300	0.190 (0.044)	0.148 (0.031)	0.029 (0.015)	0.152 (0.047)
		600	0.190 (0.041)	0.146 (0.027)	0.029 (0.013)	0.151 (0.045)
	120	120	0.140 (0.041)	0.115 (0.033)	0.016 (0.012)	0.102 (0.033)
		300	0.139 (0.032)	0.115 (0.023)	0.016 (0.008)	0.102 (0.024)
		600	0.139 (0.029)	0.113 (0.018)	0.017 (0.006)	0.101 (0.019)
	300	120	0.081 (0.031)	0.088 (0.028)	0.012 (0.011)	0.068 (0.025)
		300	0.079 (0.022)	0.087 (0.017)	0.013 (0.007)	0.067 (0.016)
		600	0.078 (0.020)	0.087 (0.014)	0.012 (0.005)	0.066 (0.012)
2	60	120	0.772 (0.040)	0.777 (0.037)	0.755 (0.045)	0.759 (0.060)
		300	0.771 (0.029)	0.778 (0.024)	0.756 (0.032)	0.757 (0.051)
		600	0.771 (0.024)	0.777 (0.018)	0.755 (0.026)	0.756 (0.048)
	120	120	0.769 (0.043)	0.766 (0.042)	0.765 (0.044)	0.715 (0.051)
		300	0.767 (0.031)	0.766 (0.026)	0.763 (0.032)	0.711 (0.036)
		600	0.764 (0.026)	0.765 (0.018)	0.761 (0.027)	0.712 (0.032)
	300	120	0.771 (0.043)	0.744 (0.039)	0.763 (0.044)	0.670 (0.048)
		300	0.772 (0.032)	0.745 (0.028)	0.765 (0.032)	0.671 (0.040)
		600	0.771 (0.027)	0.745 (0.019)	0.765 (0.027)	0.670 (0.032)
3	60	120	0.350 (0.060)	0.352 (0.048)	0.249 (0.048)	0.281 (0.068)
		300	0.349 (0.053)	0.353 (0.036)	0.247 (0.038)	0.280 (0.063)
		600	0.348 (0.049)	0.352 (0.032)	0.245 (0.035)	0.278 (0.061)
	120	120	0.287 (0.052)	0.310 (0.045)	0.197 (0.038)	0.203 (0.042)
		300	0.288 (0.041)	0.310 (0.031)	0.195 (0.028)	0.205 (0.028)
		600	0.286 (0.037)	0.308 (0.023)	0.193 (0.022)	0.200 (0.022)
	300	120	0.196 (0.038)	0.271 (0.039)	0.162 (0.034)	0.152 (0.036)
		300	0.195 (0.026)	0.268 (0.027)	0.159 (0.022)	0.149 (0.021)
		600	0.196 (0.020)	0.268 (0.022)	0.160 (0.017)	0.150 (0.017)

tour matching algorithm to present the shape of fishes as a modified curve. Each curve in the dataset contained 463 equally spaced points that were mapped from the outline of the fish. There are seven fish species with similar shape characters: salmon, winter coho, brown trout, Bonneville cutthroat, Colorado River cutthroat trout, Yellowstone cutthroat and mountain whitefish. Each species has the sample size of 50, and the total number of fishes is 350. This dataset is available at the UCR Time Series Classification and Clustering website http://www.cs.ucr.edu/~eamonn/time_series_data/. The aim is to discriminate fish species via the shape of the fish.

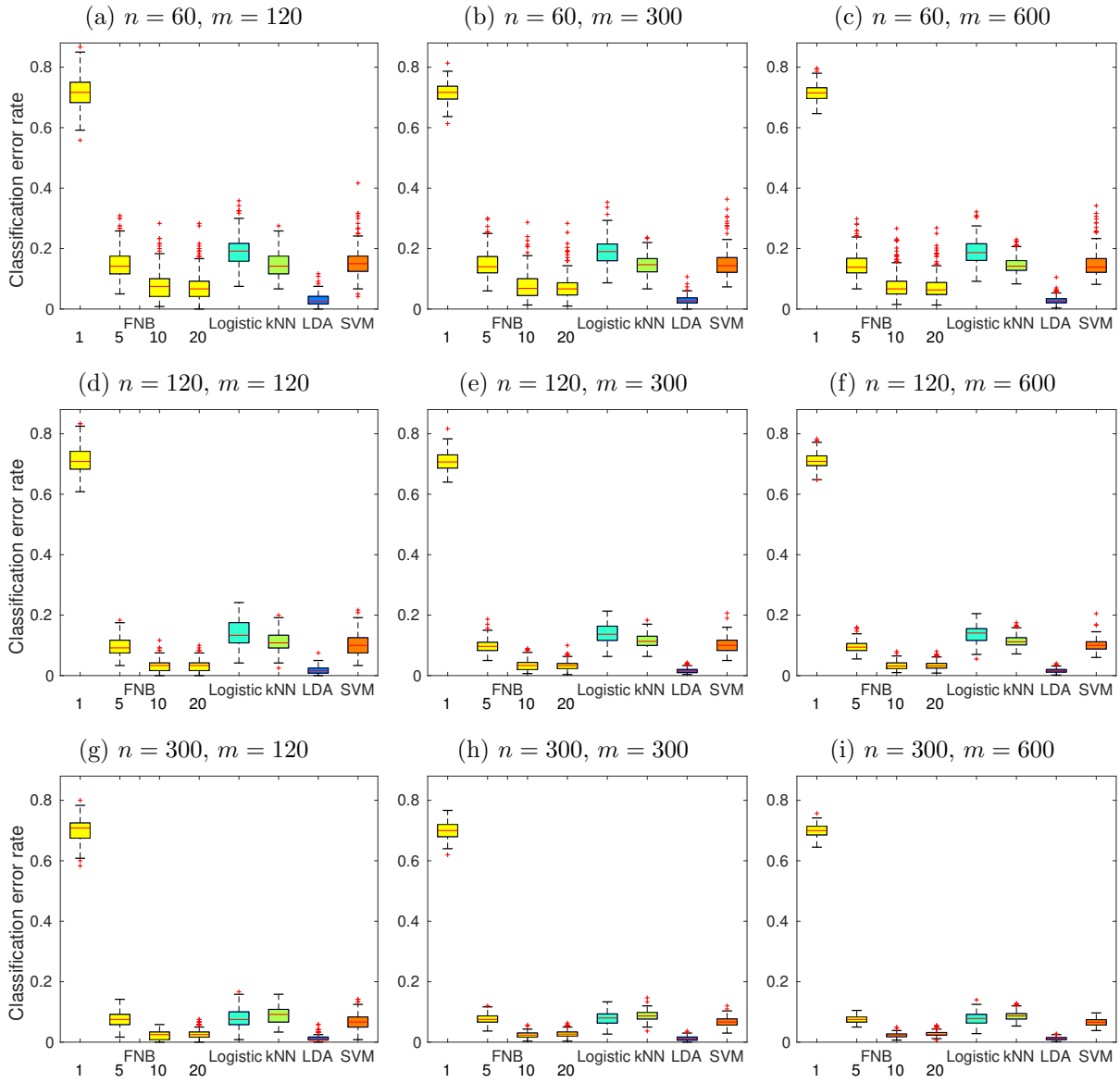


Figure 3.4: Boxplot of misclassification error rates with different number of training n and testing m using the four methods for Model 1 based on 200 replications.

The third data example, described in detail in Grandhi (2002), comes from Oregon State University. According to the original data source with the current growth of digitized data, there is a huge demand for automatic management and retrieval of various images. The OSULeaf data set consists of curves obtained by color image segmentation and boundary extraction in the counter-clockwise direction at 427 equispaced instants from digitized leaf images of six classes. In total 442 leaf images are analyzed: 66 of them are Acer Circinatum,

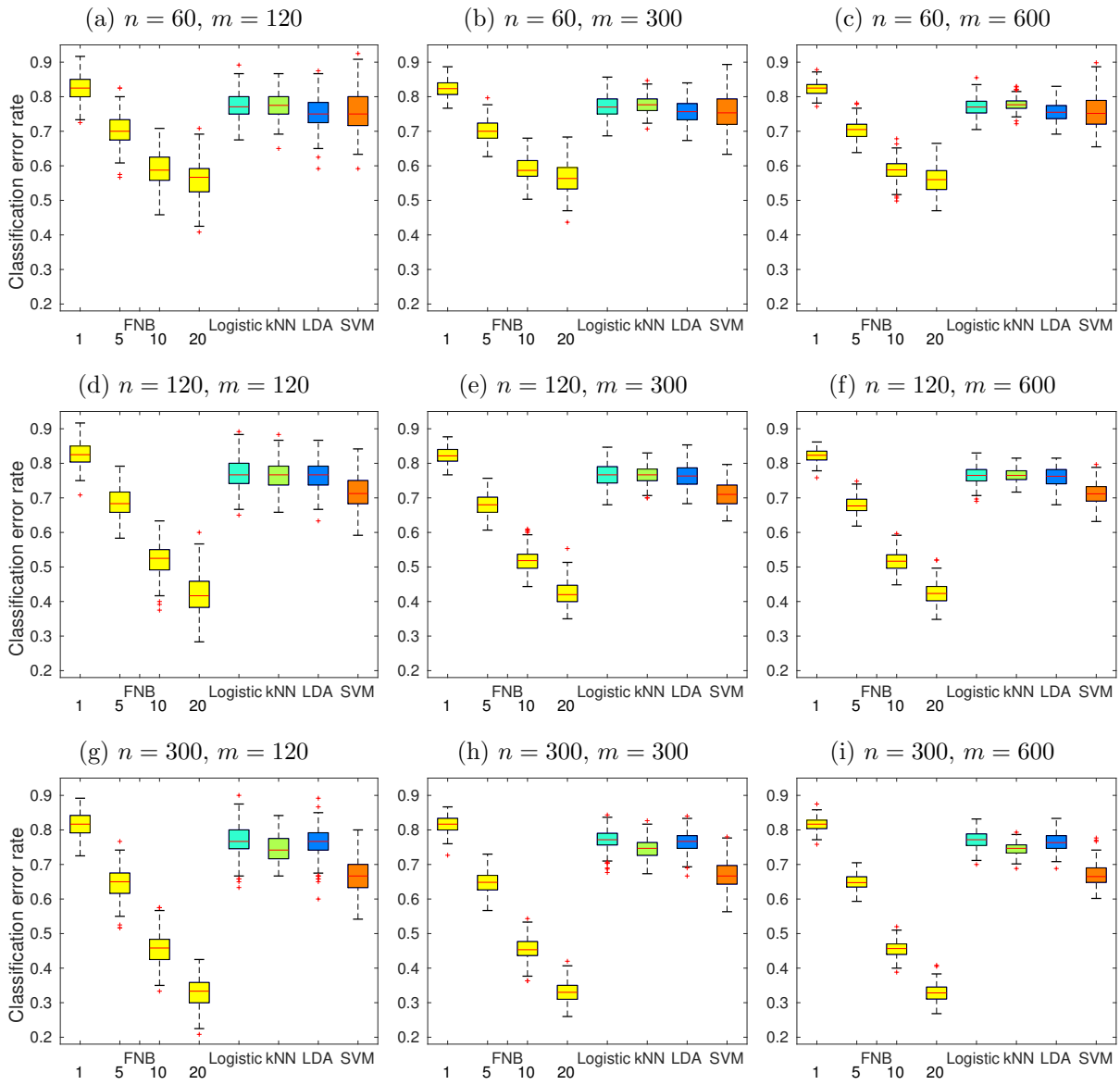


Figure 3.5: Boxplot of misclassification error rates with different number of training n and testing m using the four methods for Model 2 based on 200 replications.

84 of Acer Glabrum, 75 of Acer Macrophyllum, 97 of Acer Negundo, 82 of Quercus Garryana and 38 of Quercus Kelloggii. This dataset is also available at the UCR Time Series Classification and Clustering website. The main objective is to solve the problem of leaf boundary curves classification.

The last data example is the well-known Berkeley growth study (Tuddenham and Snyder, 1954), which is used as an example in various functional clustering studies (Chiou and Li

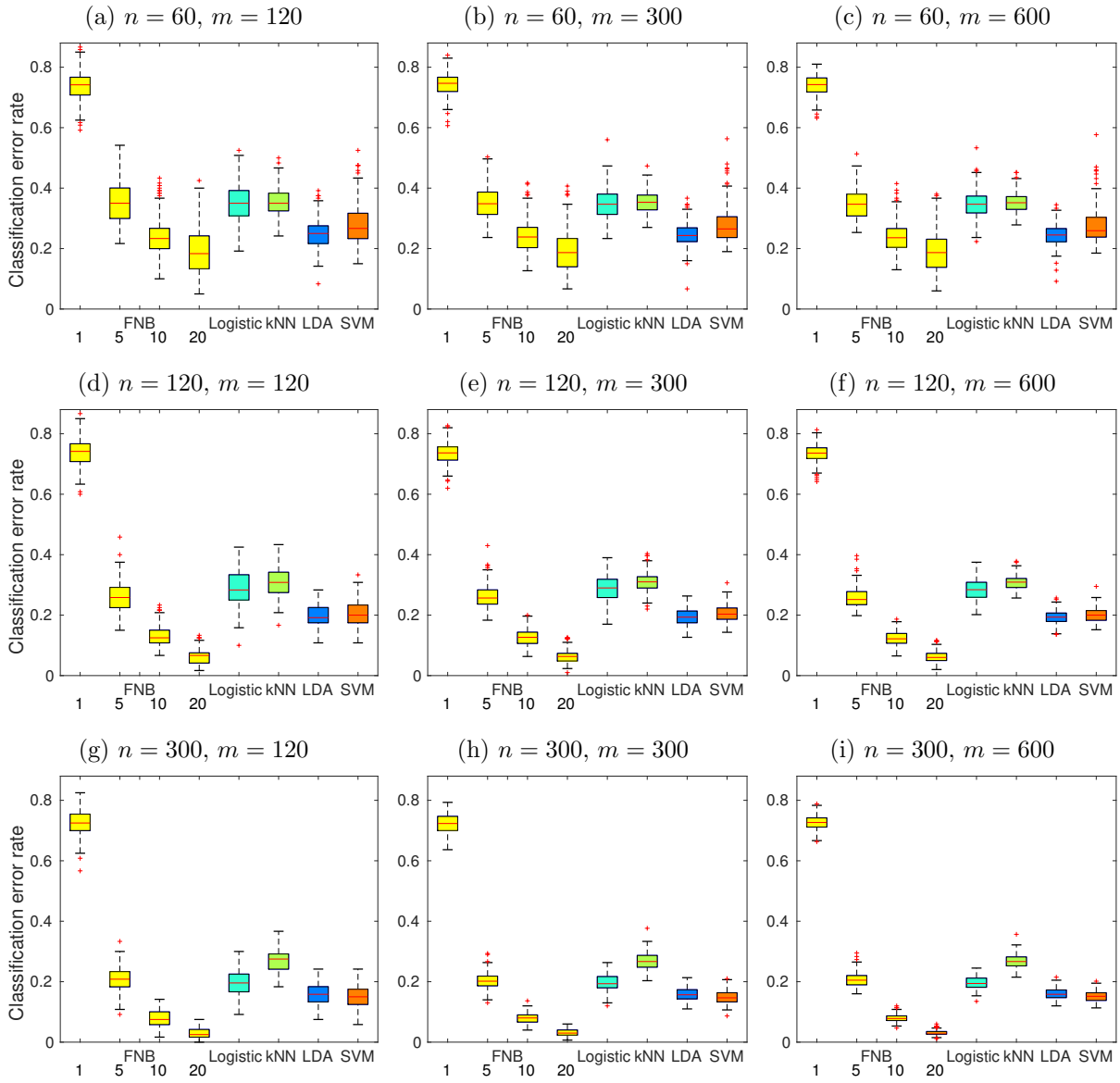


Figure 3.6: Boxplot of misclassification error rates with different number of training n and testing m using the four methods for Model 3 based on 200 replications.

(2007), Jacques and Preda (2014) and Bongiorno and Goia (2016)). In this dataset, the heights of 54 girls and 39 boys are measured at 31 not equally spaced time points, from 1 to 18 years. In total 93 subjects are measured. The goal is to discriminate the growth curves based on the gender differences.

The original trajectories for the four functional dataset are plotted in Figure 3.7. The number of groups for each dataset is reported in parentheses and the group membership of

each trajectory is highlighted in different colors. We presmooth the OSULeaf data since the original observations are quite noisy as clearly indicated in Figure 7 (c).

In Table 3.3, we report the average misclassification rates along with the standard error and the average proportion of total variance explained for the four dataset. As can be seen in Table 3.3, the performance of the proposed FNB method gets better when the number of FCPC increases. For the PeMS, OSULeaf and Growth data, the FNB performs the best when $J = 20$. For the Fish data, the FNB performs the third best, but still comparable with LDA and SVM. We note that the SVM results reported here were obtained under the best choice of kernel and tuning parameters. As we discussed in simulation (Section 3.3.1) there are some situations where more that 80% of test data are misclassified when we specified a wrong kernel or tuning parameters. In general, the results reveal how the FNB behaves with J: The misclassification errors should reduce with increasing J consistently with the proportion of total variance explained.

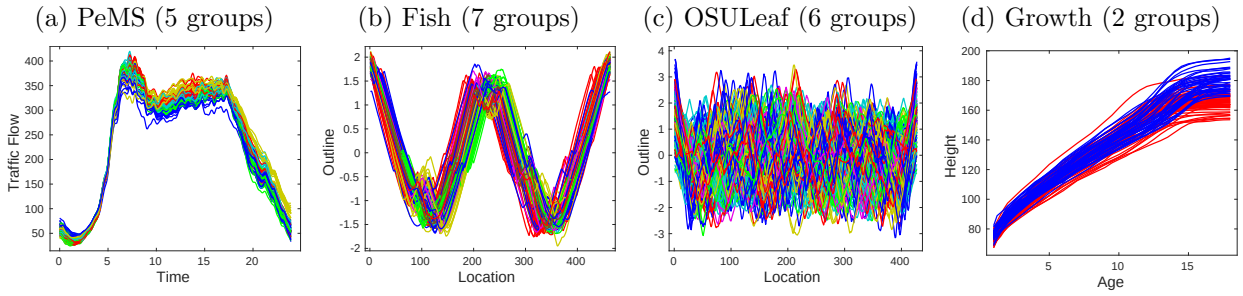


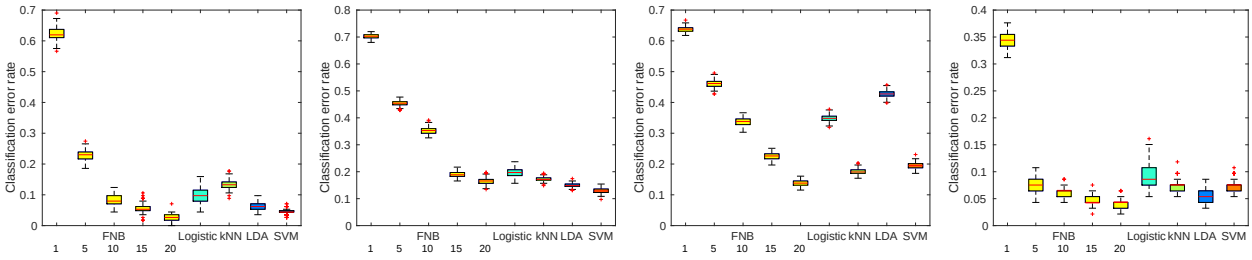
Figure 3.7: The original trajectories for the four functional dataset.

3.4 Conclusions

We have developed a classification method for functional data based on the surrogate densities constructed from FCPC scores. The novelty here is to make a naive assumption of independence of FCPC scores and use the theoretical result from Delaigle and Hall (2010) to construct surrogate densities. The surrogate densities make the density-based naive Bayes classifiers possible for functional data. It is shown that the FCPC score relies on the consis-

Table 3.3: The sample means, standard error (in parentheses) and average proportion of total variance explained [in bracket (%)] for misclassification rates.

Method	J	PeMS	Fish	OSULeaf	Growth
FNB	1	0.624 (0.019) [49.61]	0.703 (0.007) [70.16]	0.637 (0.009) [29.18]	0.346 (0.012) [88.77]
	5	0.229 (0.017) [91.69]	0.454 (0.010) [91.83]	0.460 (0.012) [68.10]	0.077 (0.012) [99.30]
	10	0.084 (0.015) [97.71]	0.353 (0.013) [98.05]	0.338 (0.013) [89.62]	0.061 (0.008) [99.97]
	15	0.058 (0.014) [99.29]	0.190 (0.010) [99.54]	0.225 (0.011) [97.35]	0.045 (0.008) [100.00]
	20	0.025 (0.011) [99.81]	0.165 (0.012) [99.90]	0.139 (0.009) [99.65]	0.040 (0.010) [100.00]
Logistic		0.095 (0.024)	0.196 (0.014)	0.349 (0.010)	0.094 (0.021)
k-NN		0.132 (0.014)	0.172 (0.007)	0.175 (0.009)	0.075 (0.010)
LDA		0.060 (0.011)	0.150 (0.008)	0.428 (0.010)	0.055 (0.010)
SVM		0.046 (0.008)	0.129 (0.009)	0.194 (0.010)	0.071 (0.011)



(a) PeMS (b) Fish (c) OSULeaf (d) Growth

Figure 3.8: Boxplot of misclassification error rate for 10-fold cross-validation based on 200 replications.

tently estimated overall mean and common eigenfunctions that are used for stochastic curve expansion. In addition, the nonparametrically estimated FNB classifier has an asymptotic equivalence to the true one. In the present paper, for ease of presentation we have assumed that there is an equal prior probability and there is an equal number of observations in each population to better demonstrate the consistence of the FNB classifier. It should be noted that this assumption is not essential and can be easily relaxed. Our simulation studies and real data applications show that the proposed FNB classifier performs well. Overall, we conclude that the proposed FNB classifier is conceptually simple, analogous to the classical naive Bayes classifier. The procedure of constructing this classifier is also easily implementable and practically useful.

CHAPTER 4

DYNAMIC FUNCTIONAL PREDICTION AND CLASSIFICATION

The idea of dynamic function prediction follows that of Chiou (2012) as the main reference. We adopt their functional mixture prediction approach combines with the functional naive Bayes classifier that proposed in Chapter 3. This study extends the idea of the functional naive Bayes to identify distinct patterns of traffic flow from the past data.

4.1 Modeling traffic flow trajectries

As we mentioned in Chapter 1.2, the daily traffic flow trajectory can be viewed as a functional data. We use the notation X to be the random function for the daily traffice flow trajectory in the domain $\mathcal{I} = [0, T]$. The random function X has a smooth mean function $E(X(t)) = \mu(t)$ and covariance function $cov(X(s), X(t)) = \Gamma(s, t)$ for s and t in $\mathcal{I}$.

4.1.1 Functional Naive Bayes and Functional Probability Naive Bayes Classifier

We assume the random function X consists of C sub-functions, with each sub-function corresponding to a population. We also assume that there are Π_c, $c \in \{1, \cdots, C\}$, populations, where C is an integer indicating the total number of populations. Let the random variable Y be the group label of X. For each sub-function associated with the group c, we define the conditional mean function $E(X(t)|Y = c) = \mu^{(c)}(t)$ and the covariance function $Cov(X(s), X(t)|Y = c) = \Gamma^{(c)}(s, t)$. Let $(\lambda_j^{(c)}, \phi_j^{(c)})$ be the corresponding eigenvalue-eigenfunction pairs of the covariance kernel $\Gamma^{(c)}$.

Following the conventional approach, the best group membership c given X is determined by maximizing the posterior probability $P_{Y|X}(\cdot|\cdot)$ such that

$$c^*(X) = \underset{c \in \{1, \cdots, C\}}{\operatorname{argmax}} P_{Y|X}(c|X).$$

However, it is clear that the above posterior probability is conditioned on the probability

density function of a random function X, which does not exist as discussed in Delaigle and Hall (2010). Without attempting to model the underlying probability distribution, we propose estimating the posterior membership probability $P(Y = c|X)$ using the following methods:

- Functional Naive Bayes: the posterior probability is either 1 or 0, i.e.,

$$P_{Y|X}(c|X) = \begin{cases} 1, & \text{if } \ c = \delta_J(X) \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

  where $\delta_J(X)$ is the criterion function (3.7) as we discussed in Chapter 3

$$\delta_J(X) = \underset{c\in\{1,\cdots,C\}}{\operatorname{argmax}} \ \pi_c f^{(c)}(X|\varepsilon) \quad \text{and} \quad f^{(c)}(X|\varepsilon) = \prod_{j=1}^{J} f_j^{(c)}(x_j).$$

- Functional Probability Naive Bayes: the posterior group membership probability is estimated using the proportion of posterior probability of each group.

$$P(Y = c|X) = \frac{\pi_c f^{(c)}(X|\varepsilon)}{\sum_{c=1}^{C} \pi_c f^{(c)}(X|\varepsilon)}, \quad \text{for } c = 1, \cdots, C \tag{4.2}$$

In the machine learning literature, the equations (4.1) and (4.2) are called hard- and soft-classification, respectively.

For the purpose of prediction, the time domain $\mathcal{I}$ of the function X is partitioned into two exclusive time domains $\mathcal{S}(\tau) = [0, \tau]$ and $\mathcal{T}(\tau) = [\tau, T]$. Now, let X^* be a newly observed trajectory of the function X, denoted by $X^*_{\mathcal{S}(\tau)}$ as observed up to time τ. We predict the group membership probability of the trajectory X^* based on the known trajectory $X^*_{\mathcal{S}(\tau)}$ observed until time τ, which will then be used to predict the unobserved trajectory $X^*_{\mathcal{T}(\tau)}$.

We define the surrogate density of a random function based on the partially observed $X_{\mathcal{S}(\tau)}$ rather than the entire X since the part $X_{\mathcal{T}(\tau)}$ is not yet observed. Suppose we have the group mean function $\mu^{(c)}(t)$ and covariance function $\Gamma^{(c)}(s,t)$, $c = 1, \cdots, C$, the partially observed mean function and covariance function are denoted by $\mu^{(c)}_{\mathcal{S}(\tau)}$ and $\Gamma^{(c)}_{\mathcal{S}(\tau)}(s,t)$. Then,

the partially observed overall mean function and common covariance function are defined as

$$\mu_{\mathcal{S}(\tau)}(t) = \sum_{c=1}^{C} \pi_c \mu^{(c)}_{\mathcal{S}(\tau)}(t) \quad \text{and} \quad \Gamma_{\mathcal{S}(\tau)}(s,t) = \sum_{c=1}^{C} \pi_c \Gamma^{(c)}_{\mathcal{S}(\tau)}(s,t),$$

respectively. The partially observed common eigenfunction is then derived from the eigen-decomposition of the partially observed common covariance function. We denoted the partially observed common eigenfunction by $\phi_{\mathcal{S}(\tau),j}(t)$, for $j \geq 1$. Taking $\xi^{(c)}_{\mathcal{S}(\tau),j} = \langle X^{(c)}_{\mathcal{S}(\tau)} - \mu_{\mathcal{S}(\tau)}, \phi_{\mathcal{S}(\tau),j} \rangle$, the surrogate density can be approximated by the product of probability density function of $\xi_{\mathcal{S}(\tau),j}$, i.e.,

$$f^{(c)}(X^{(c)}_{\mathcal{S}(\tau)} | \varepsilon) \approx \prod_{j=1}^{J} f^{(c)}_j(\xi^{(c)}_{\mathcal{S}(\tau),j}).$$

We can predict the group membership based on the newly observed $X^*_{\mathcal{S}(\tau)}$ using the functional naive Bayes

$$\delta_J(X^*_{\mathcal{S}(\tau)}) = \underset{c \in \{1,\cdots,C\}}{\operatorname{argmax}} \pi_c f^{(c)}(X^*_{\mathcal{S}(\tau)} | \varepsilon)$$

and the group membership probability using the functional probability naive Bayes

$$P(Y = c | X^*_{\mathcal{S}(\tau)}) = \frac{\pi_c f^{(c)}(X^*_{\mathcal{S}(\tau)} | \varepsilon)}{\sum_{c=1}^{C} \pi_c f^{(c)}(X^*_{\mathcal{S}(\tau)} | \varepsilon)},$$

for $c = 1, \cdots, C$. We note that the prior probability π_c remains the same from the training data.

4.1.2 Estimation for Functional Naive Bayes and Functional Probability Bayes Classifier

In practice, the model components of functional naive Bayes classifier can be estimated from the training data. Given the observations $\{t_{ij}, Y_i, X_i(t_{ij})\}$, $i = 1, \cdots, n$, $j = 1, \cdots, m_i$, we follow the same procedure as discussed in Chapter 3.1.1 to get the estimates of $\{\hat{\pi}_c, \hat{\mu}^{(c)}, \widehat{\Gamma}^{(c)}\}$, $c = 1, \cdots, C$. Then the estimated overall mean function is $\hat{\mu}(t) = \sum_{c=1}^{C} \hat{\pi}_c \hat{\mu}^{(c)}(t)$ and the estimated common covariance function is $\widehat{\Gamma}(s,t) = \sum_{c=1}^{C} \hat{\pi}_c \widehat{\Gamma}^{(c)}(s,t)$. Denoting the j-th eigenvalue-eigenfunction pair of $\widehat{\Gamma}$ by $(\hat{\lambda}_j, \hat{\phi}_j)$, we obtain the FCPC scores for a generic

functional observation X from group c, as $\hat{\xi}_j^{(c)} = \langle X^{(c)} - \hat{\mu}, \hat{\phi}_j \rangle$. The kernel density estimate for the j-th FCPC score in group c is

$$\hat{f}_j^{(c)}(u) = \frac{1}{n_c h_j^c} \sum_{i=1}^{n_c} K\left(\frac{u - \hat{\xi}_{ij}^{(c)}}{h_j^c} \right).$$

Now for any function $z \in L_2(\mathcal{I})$, we write $\hat{z}_j = \langle z - \hat{\mu}, \hat{\phi}_j \rangle$, the surrogate density function is

$$\hat{f}^{(c)}(z|\varepsilon) = \prod_{j=1}^{J} \hat{f}_j^{(c)}(\hat{z}_j).$$

For the details of the procedure we refer to Chapter 3.1.1.

Now, given a newly observed trajectory X^* up to time τ, denoted by $X^*_{\mathcal{S}(\tau)}$, we obtain the estimated surrogate density

$$\hat{f}^{(c)}(X^*_{\mathcal{S}(\tau)}|\varepsilon) = \prod_{j=1}^{J} \hat{f}_j^{(c)}(\hat{\xi}_{\mathcal{S}(\tau),j})$$

for $c = 1, \cdots, C$, where the j-th FCPC score $\hat{\xi}_{\mathcal{S}(\tau),j} = \langle X^*_{\mathcal{S}(\tau)} - \hat{\mu}_{\mathcal{S}(\tau)}, \hat{\phi}_{\mathcal{S}(\tau),j} \rangle$, for $j \geq 1$. To obtain $\{\hat{\phi}_{\mathcal{S}(\tau),j}\}$ we simply decompose the covariance estimate $\widehat{\Gamma}$ into blocks corresponding to the time domains $\mathcal{S}(\tau)$ and $\mathcal{T}(\tau)$ without re-estimating the covariance function. We predict the group membership for the newly observed $X^*_{\mathcal{S}(\tau)}$ by the functional naive Bayes

$$\hat{\delta}_J(X^*_{\mathcal{S}(\tau)}) = \underset{c \in \{1,\cdots,C\}}{\operatorname{argmax}} \hat{\pi}_c \hat{f}^{(c)}(X^*_{\mathcal{S}(\tau)}|\varepsilon) \tag{4.3}$$

and the group membership probability using the functional probability naive Bayes

$$\widehat{P}(Y = c|X^*_{\mathcal{S}(\tau)}) = \frac{\hat{\pi}_c \hat{f}^{(c)}(X^*_{\mathcal{S}(\tau)}|\varepsilon)}{\sum_{c=1}^{C} \hat{\pi}_c \hat{f}^{(c)}(X^*_{\mathcal{S}(\tau)}|\varepsilon)} \tag{4.4}$$

for $c = 1, \cdots, C$.

4.2 Functional Mixture Prediction

In order to accurately predict traffic flow trajectories under various traffic conditions, we propose to combine the functional linear model with functional naive Bayes methods. Given

a newly observed trajectory $X^*_{\mathcal{S}(\tau)}$ of the process X as observed up to time τ, by using law of iterated expectation (Durrett, 2010) one can show that

$$E(X^*_{\mathcal{T}(\tau)}(t)|X^*_{\mathcal{S}(\tau)}) = \sum_{c=1}^{C} E(X^*_{\mathcal{T}(\tau)}(t)|X^*_{\mathcal{S}(\tau)}, Y = c)P(Y = c|X^*_{\mathcal{S}(\tau)}), \tag{4.5}$$

where $E(X_{\mathcal{T}(\tau)}(t)|X^*_{\mathcal{S}(\tau)}, Y = c)$ is the predictive function conditional on group $Y = c$ and $P(Y = c|X^*_{\mathcal{S}(\tau)})$ is the posterior probability of group membership given the newly observed trajectory $X^*_{\mathcal{S}(\tau)}$ up to time τ.

4.2.1 Functional Linear Prediction Model

Given $X^*_{\mathcal{S}(\tau)}$, we aim to predict the values of $X^*_{\mathcal{T}(\tau)}$. Here we consider a functional linear regression model (Ramsay and Silverman, 2005). The process $X(s)$, for $s \in \mathcal{S}(\tau)$ denoted by $X_{\mathcal{S}(\tau)}$, serves as the predictor function and the process $X(t)$, for $t \in \mathcal{T}(\tau)$ denoted by $X_{\mathcal{T}(\tau)}$, is the response function. For each group, the process $X^{(c)}$ is decomposed into $X^{(c)}_{\mathcal{S}(\tau)}$ and $X^{(c)}_{\mathcal{S}(\tau)}$ whose Karhunen-Loève expansions can be obtained such that

$$X^{(c)}_{\mathcal{S}(\tau)}(s) = \mu^{(c)}_{\mathcal{S}(\tau)}(s) + \sum_{j=1}^{\infty} \xi^{(c)}_{\mathcal{S}(\tau),j}\phi^{(c)}_{\mathcal{S}(\tau),j}(s)$$

and

$$X^{(c)}_{\mathcal{T}(\tau)}(t) = \mu^{(c)}_{\mathcal{T}(\tau)}(t) + \sum_{k=1}^{\infty} \xi^{(c)}_{\mathcal{T}(\tau),k}\phi^{(c)}_{\mathcal{T}(\tau),k}(t),$$

where $\xi^{(c)}_{\mathcal{S}(\tau),j} = \langle X^{(c)}_{\mathcal{S}(\tau)} - \mu^{(c)}_{\mathcal{S}(\tau)}, \phi^{(c)}_{\mathcal{S}(\tau),j}\rangle$ and $\xi^{(c)}_{\mathcal{T}(\tau),k} = \langle X^{(c)}_{\mathcal{T}(\tau)} - \mu^{(c)}_{\mathcal{T}(\tau)}, \phi^{(c)}_{\mathcal{T}(\tau),k}\rangle$

Conditioning on the group membership, the functional linear regression model becomes

$$E(X_{\mathcal{T}(\tau)}(t)|X_{\mathcal{S}(\tau)}, Y = c) = \mu^{(c)}_{\mathcal{T}(\tau)}(t) + \int_{\mathcal{S}(\tau)} \beta^{(c)}_{\tau}(s,t)(X_{\mathcal{S}(\tau)}(s) - \mu^{(c)}_{\mathcal{S}(\tau)}(s))ds \tag{4.6}$$

for all $t \in \mathcal{T}(\tau)$. Here, given a fixed value of τ, the bivariate regression function $\beta^{(c)}_{\tau}(s,t)$ is smooth and square integrable, that is, $\int_{\mathcal{T}(\tau)} \int_{\mathcal{S}(\tau)} \beta^{(c)}_{\tau}(s,t)dsdt < \infty$. Under certain regularity conditions which are outlined in He et al. (2000) the regression coefficient function

$\beta_\tau^{(c)}$ has a basis representation such that $\beta_\tau^{(c)}(s,t) = \sum_{k=1}^{\infty}\sum_{j=1}^{\infty}\beta_{\tau,jk}^{(c)}\phi_{\mathcal{S}(\tau),j}^{(c)}(s)\phi_{\mathcal{T}(\tau),k}^{(c)}(t)$, where $\beta_{\tau,jk}^{(c)} = E\left(\xi_{\mathcal{S}(\tau),j}^{(c)}\xi_{\mathcal{T}(\tau),k}^{(c)}\right)\Big/E\left((\xi_{\mathcal{S}(\tau),j}^{(c)})^2\right)$. Model (4.6) thus can be rewritten as

$$E(X_{\mathcal{T}(\tau)}(t)|X_{\mathcal{S}(\tau)}, Y = c) = \mu_{\mathcal{T}(\tau)}^{(c)}(t) + \sum_{k=1}^{\infty}\sum_{j=1}^{\infty}\beta_{\tau,jk}^{(c)}\xi_{\mathcal{S}(\tau),j}^{(c)}\phi_{\mathcal{T}(\tau),k}^{(c)}(t). \tag{4.7}$$

Suppose that the group structure $\mu_{\mathcal{S}(\tau)}^{(c)}$, $\mu_{\mathcal{T}(\tau)}^{(c)}$, $\{\phi_{\mathcal{S}(\tau),j}^{(c)}\}$, $\{\phi_{\mathcal{T}(\tau),k}^{(c)}\}$ and the regression coefficient $\beta_{\tau,jk}^{(c)}$ are given, one can predict the unobserved trajectory $X_{\mathcal{T}(\tau)}^*$ based on the partially observed trajectory $X_{\mathcal{S}(\tau)}^*$ for a specific cluster c by (4.7), that is,

$$E(X_{\mathcal{T}(\tau)}^*(t)|X_{\mathcal{S}(\tau)}^*, Y = c) = \mu_{\mathcal{T}(\tau)}^{(c)}(t) + \sum_{k=1}^{\infty}\sum_{j=1}^{\infty}\beta_{\tau,jk}^{(c)}\xi_{\mathcal{S}(\tau),j}^{*(c)}\phi_{\mathcal{T}(\tau),k}^{(c)}(t) \tag{4.8}$$

for all $t \in \mathcal{T}(\tau)$, where $\xi_{\mathcal{S}(\tau),j}^{*(c)} = \langle X_{\mathcal{S}(\tau)}^{*(c)} - \mu_{\mathcal{S}(\tau)}^{(c)}, \phi_{\mathcal{S}(\tau),j}^{(c)}\rangle$. In practice, the group structure $\mu_{\mathcal{S}(\tau)}^{(c)}$, $\mu_{\mathcal{T}(\tau)}^{(c)}$, $\{\phi_{\mathcal{S}(\tau),j}^{(c)}\}$, and $\{\phi_{\mathcal{T}(\tau),k}^{(c)}\}$ can be estimated analogously from the training data as described in Section 4.1.2. However, the regression coefficients $\beta_{\tau,jk}^{(c)}$ are more complicated. We will summarize the estimation procedure in the next section.

4.2.2 Estimation for Functional Mixture Prediction

Since the bivariate regression function $\beta_\tau^{(c)}(s,t)$ is a smooth function of τ for all s and t, it follows that $\beta_{\tau,jk}^{(c)}$ is also smooth in τ for all j and k. To estimate the regression parameters $\beta_{\tau,jk}^{(c)}$ for each group, one needs to derive the estimated principal component score $\hat{\xi}_{\mathcal{S}(\tau),j}^{(c)}$ and $\hat{\xi}_{\mathcal{T}(\tau),k}^{(c)}$ and the eigenfunction $\hat{\phi}_{\mathcal{S}(\tau),j}^{(c)}(t)$ and $\hat{\phi}_{\mathcal{T}(\tau),k}^{(c)}(t)$ for each group. The estimate of $\beta_{\tau,jk}^{(c)}$ is then derived by

$$\tilde{\beta}_{\tau,jk}^{(c)} = \left\{(n_c - 1)\lambda_{\mathcal{S}(\tau),j}^{(c)}\right\}^{-1}\sum_{i=1}^{n_c}\left(\hat{\xi}_{\mathcal{S}(\tau),i,j}^{(c)} - \bar{\hat{\xi}}_{\mathcal{S}(\tau),j}^{(c)}\right)\left(\hat{\xi}_{\mathcal{T}(\tau),i,k}^{(c)} - \bar{\hat{\xi}}_{\mathcal{T}(\tau),k}^{(c)}\right), \tag{4.9}$$

where $\bar{\hat{\xi}}_{\mathcal{S}(\tau),j}^{(c)}$ and $\bar{\hat{\xi}}_{\mathcal{T}(\tau),k}^{(c)}$ are sample averages of $\hat{\xi}_{\mathcal{S}(\tau),i,j}^{(c)}$ and $\hat{\xi}_{\mathcal{T}(\tau),i,k}^{(c)}$, respectively. The estimate is motivated by $\beta_{\tau,jk}^{(c)} = E\left(\xi_{\mathcal{S}(\tau),j}^{(c)}\xi_{\mathcal{T}(\tau),k}^{(c)}\right)\Big/E\left((\xi_{\mathcal{S}(\tau),j}^{(c)})^2\right)$. The local linear smoothing method (Fan and Gijbels, 1996) is then applied on the estimates $\{\tilde{\beta}_{\tau,jk}^{(c)}, \tau = \tau_1, \cdots, \tau_Q\}$ over

τ to obtain the smoothed estimates $\hat{\beta}^{(c)}_{\tau,ik}$, where Q is the number of time points at which predicting the future trajectory is of interest.

We then proceed to obtain the predicted trajectory conditional on group c by using their estimate

$$\widehat{E}(X^*_{\mathcal{T}(\tau)}(t)|X^*_{\mathcal{S}(\tau)}, Y=c) = \hat{\mu}^{(c)}_{\mathcal{T}(\tau)}(t) + \sum_{k=1}^{L_c}\sum_{j=1}^{L_c} \hat{\beta}^{(c)}_{\tau,jk}\hat{\xi}^{*(c)}_{\mathcal{S}(\tau),j}\hat{\phi}^{(c)}_{\mathcal{T}(\tau),k}(t) \tag{4.10}$$

for all $t \in \mathcal{T}(\tau)$. Here, L_c is determined by (2.5). Finally, the functional mixture prediction model is combining the results of (4.10) with (4.4). We obtain the predicted unobserved traffic flow trajectory

$$\widehat{E}(X^*_{\mathcal{T}(\tau)}(t)|X^*_{\mathcal{S}(\tau)}) = \sum_{c=1}^{C} \widehat{E}(X^*_{\mathcal{T}(\tau)}(t)|X^*_{\mathcal{S}(\tau)}, Y=c)\widehat{P}(Y=c|X^*_{\mathcal{S}(\tau)}). \tag{4.11}$$

The implementation algorithm of functional mixture prediction is summarized in the next section.

4.2.3 Implementation algorithm of functional mixture prediction.

The algorithm of functional mixture prediction is summarized is this section. Suppose we have a newly partially observed trajectory $\{t_j, X^*(t^*_j)\}$ for $t_j < \tau$, and denoted by $X^*_{\mathcal{S}(\tau)}$. The functional mixture prediction that combines the functional linear model and functional probability Naive Bayes classifier is summarized as follows:

Step 1 Train the functional linear regression model and functional Naive Bayes Classifier for each group c, where $c = 1, \cdots, C$.

(i) Fit the functional linear regression model based on the group specified training data and derive the smoothed regression coefficient estimates $\hat{\beta}^{(c)}_{\tau,jk}$.

(ii) Train the functional Naive Bayes classifier and derive the estimates of prior probability $\hat{\pi}_c$ and the estimates of surrogate density $\hat{f}^{(c)}(\cdot|\varepsilon)$.

Step 2 For a new partially observed trajectory $X^*_{\mathcal{S}(\tau)}$, predict the unobserved trajectory $X^*_{\mathcal{T}(\tau)}$ and also the posterior membership probability

(i) The unobserved trajectory is predicted based on group specified functional linear model, that is, fit $\widehat{E}(X_{\mathcal{T}(\tau)}(t)|X_{\mathcal{S}(\tau)}, Y = c)$ in (4.10).

(ii) The unknown group and posterior probability is predicted by functional probability naive Bayes, that is, $\widehat{P}(Y = c|X^*_{\mathcal{S}(\tau)})$ in (4.4).

Step 3 The unobserved trajectory is predicted by the functional mixture prediction model that combines the functional linear model with functional probability naive Bayes, that is $\widehat{E}(X^*_{\mathcal{T}(\tau)}|X^*_{\mathcal{S}(\tau)})$ as derived in (4.5).

The whole process of the functional mixture prediction flow chart is plotted in Figure 4.1. We further note that, in order to obtain the prediction interval of $\widehat{E}(X^*_{\mathcal{T}(\tau)}|X^*_{\mathcal{S}(\tau)})$, the bootstrap resampling method is applied and summarized in the next section.

4.2.4 Bootstrap prediction intervals for functional mixture prediction

In order to obtain the prediction interval of $\hat{X}^*_{\mathcal{T}(\tau)}(t) = \widehat{E}(X^*_{\mathcal{T}(\tau)}|X^*_{\mathcal{S}(\tau)})$, one commonly used method is the bootstrap resampling method. Using the training data, we resample the trajectory $x^{(c)}(t) = \mu^{(c)}(t) + \sum_{j=1}^{Lc} \xi_j^{(c)}\phi_j^{(c)}(t)$ by the following procedures:

Step 1 The mean function $\mu^{(c)}$ and the eigenfunctions $\phi_j^{(c)}$ are treated as fixed components and are replaced with their estimates $\hat{\mu}^{(c)}$ and $\hat{\phi}^{(c)}$.

Step 2 Derive the estimated FPC scores $\hat{\xi}_{ij}^{(c)} = \langle x_i^{(c)} - \hat{\mu}^{(c)}(t), \hat{\phi}_j^{(c)}\rangle$. Then, obtain the estimated residual of each $X_i^{(c)}$ by $\hat{\epsilon}_{il}^{(c)} = x_i^{(c)}(t_{il}) - \hat{\mu}^{(c)}(t_{il}) - \sum_{j=1}^{Lc} \hat{\xi}_{ij}^{(c)}\hat{\phi}_j^{(c)}(t_{il})$.

Step 3 Obtain the b-th bootstrap sample of the FPC scores $\{\hat{\xi}^{(c)}_{1j,b}, \cdots, \hat{\xi}^{(c)}_{n_c j,b}\}$ and the residual $\{\hat{\epsilon}^{(c)}_{1l,b}, \cdots, \hat{\epsilon}^{(c)}_{n_c l,b}\}$ by sampling with replacement from $\{\hat{\xi}_{il}^{(c)}, 1 \le i \le n_c, 1 \le j \le L_c\}$ and $\{\hat{\epsilon}_{1l}^{(c)}, 1 \le i \le n_c, 1 \le l \le m\}$, respectively.

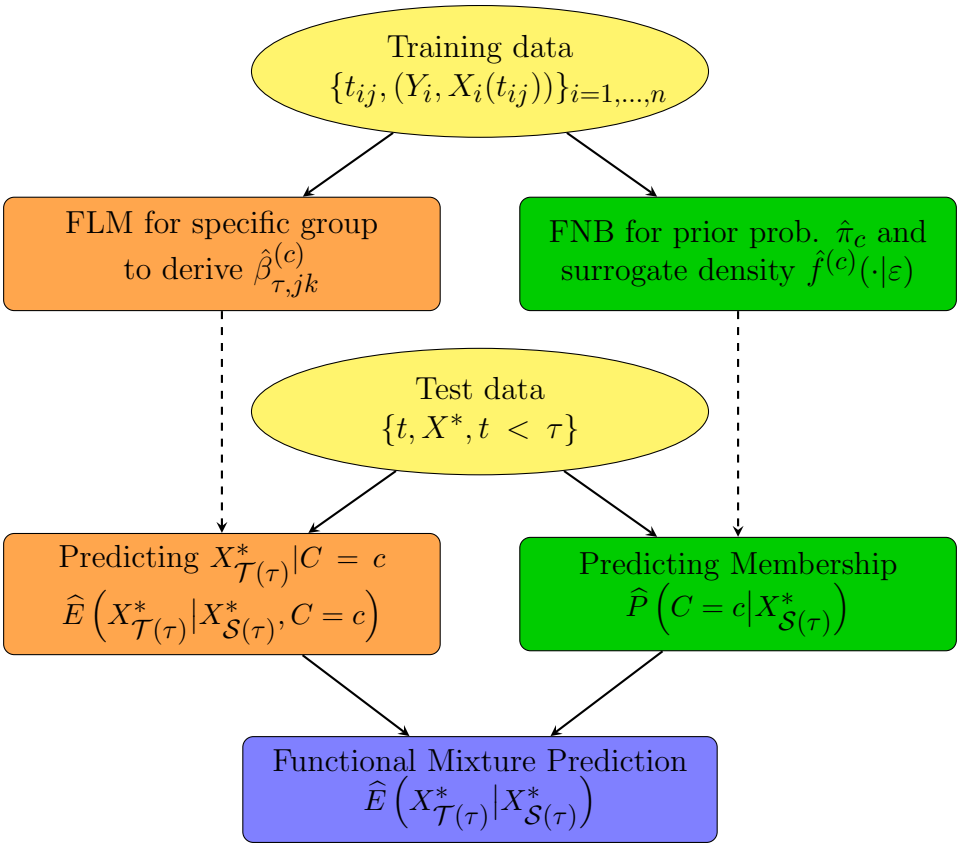


Figure 4.1: Functional mixuture prediction flow chart.

Step 4 The b-th bootstrap sample $\{X^{(c)}_{1,b}, \cdots, X^{(c)}_{n_c,b}\}$ is constructed by $X^{(c)}_{i,b}(t_{il}) = \hat{\mu}^{(c)}(t_{il}) + \sum_{j=1}^{L_c} \hat{\xi}^{(c)}_{ij,b}\hat{\phi}^{(c)}_j(t_{il}) + \epsilon^{(c)}_{il,b}$.

Step 5 Use the b-th bootstrap sample to derived the smoothed regression coefficient estimates $\hat{\beta}^{(c)}_{\tau,b,jk}$ of the functional linear regression model and the surrogated density $\hat{f}^{(c)}_b(\cdot|\varepsilon)$ of the functional naive Bayes classifier.

Step 6 Based on the estimates obtained from each bootstrap sample, we can estimate the posterior probability $\widehat{P}(Y = c|X^*_{\mathcal{S}(\tau)})$ as in (4.4) and the predicted trajectory conditioning on group c, $\widehat{E}(X^*_{\mathcal{T}(\tau)}(t)|X^*_{\mathcal{S}(\tau)}, Y = c)$ as in (4.10).

Step 7 The mixture predicted trajectory $\widehat{E}(X^*_{\mathcal{T}(\tau)}(t)|X^*_{\mathcal{S}(\tau)})$ is then obtained according to (4.5).

Step 8 The 95% prediction interval is constructed by using the 2.5% and 97.5% percentiles of the bootstrap predicted trajectories.

4.3 A Real Data Application

4.3.1 Analysis of Traffic Flow Patterns and Posterior Probabilities

In the section we use one data example to illustrate the proposed functional mixture prediction. This traffic flow data were collected by a dual loop vehicle detector over 15-min time intervals located near Shea-San Tunnel on National Highway 5 in Taiwan in 2009. The trajectories sample 70 days as the training data and the remaining 14 days are used as the test data to validate the prediction performance. The goal is to predict the unobserved traffic flow trajectory for a partial trajectory with updated flow information up to the "current time" τ.

Based on our prior knowledge, this traffic flow data can be divided into three groups. Group 1 contains all holidays, Group 2 comprises weekdays including Mondays through Thursdays, and Group 3 comprises Fridays. The mean functions of the three groups and the overall trajectories are displayed in Figure 4.2. It is clear that Group 1 has a higher mean traffic flow rate than the other two groups, Group 2 and Group 3 have relatively close mean flow rates in terms of shape and magnitude until 11:00, and they diverge thereafter with a higher mean flow rate in Group 3. In Figure 4.3, we plot the mean functions along with observed trajectories, covariance functions, and leading eigenfunctions.

We first use the training data to train the functional linear model and the functional naive Bayes classifier as described in previous section. Given a newly observed trajectory from the test data up to the current time τ, we predict the posterior probabilities by (4.4). The posterior probabilities for some test samples are illustrated in Figure 4.4 with the values of τ from 8:00 to 20:00 by 15-min intervals. In Figure 4.4, Test sample 2, 3, 9 are classified as Group 1 (holidays) for all τ. Other Test samples are classified either in Group 2 (Weekdays)

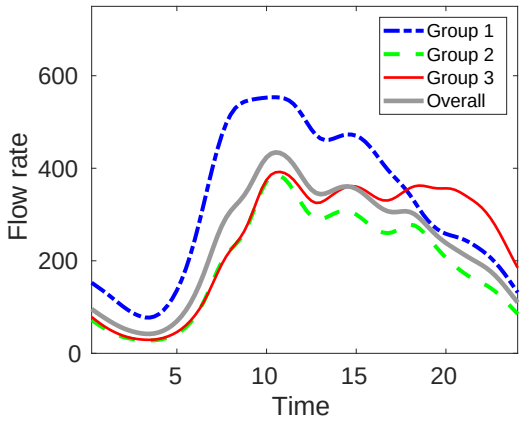


Figure 4.2: Overall and group-specific mean functions of the training data of daily traffic flow rates.

or Group 3 (Fridays). In this case Group 2 and Group 3 have almost the same pattern prior to $\tau = 10:00$, which may lead to misclassification and result in poor prediction accuracy. This issue is resolved as τ moves onward.

4.3.2 Mixture Prediction of Traffic Flow

To examine the performance of mixture prediction, we use partially observed trajectories before time τ to predict the future interval length after time τ. Here we introduce some notations and criterion as used in Chiou et al. (2014b). We define $\mathcal{S}(\tau;\omega) = [\max(0, \tau-\omega), \tau]$ and $\mathcal{T}(\tau;\kappa) = [\tau, \min(\tau+\kappa, T)]$, where ω is the length of the known interval prior to time τ to be used in prediction calculations and κ is the length of the unknown interval to be predicted from time τ onward. Given a sample X_i^* observed up to time τ, denoted by $X_{i,\mathcal{S}(\tau)}^*$, the mean integrated prediction error (MIPE) can be calculated by

$$\text{MIPE}(\tau,\omega,\kappa) = \frac{1}{m_p}\sum_{i=1}^{m_p}\frac{1}{\kappa}\int_0^{\kappa}\left\{\hat{X}_{i,\mathcal{T}(\tau)}^*(t) - X_{i,\mathcal{T}(\tau)}^*(t)\right\}^2 dt,$$

where $\hat{X}_{i,\mathcal{T}(\tau)}^*(t)$ is the predicted trajectory obtained by (4.11) and $X_{i,\mathcal{T}(\tau)}^*$ is the X_i^* observed after time τ. Here the m_p is the number of trajectories in the test data. The total mean

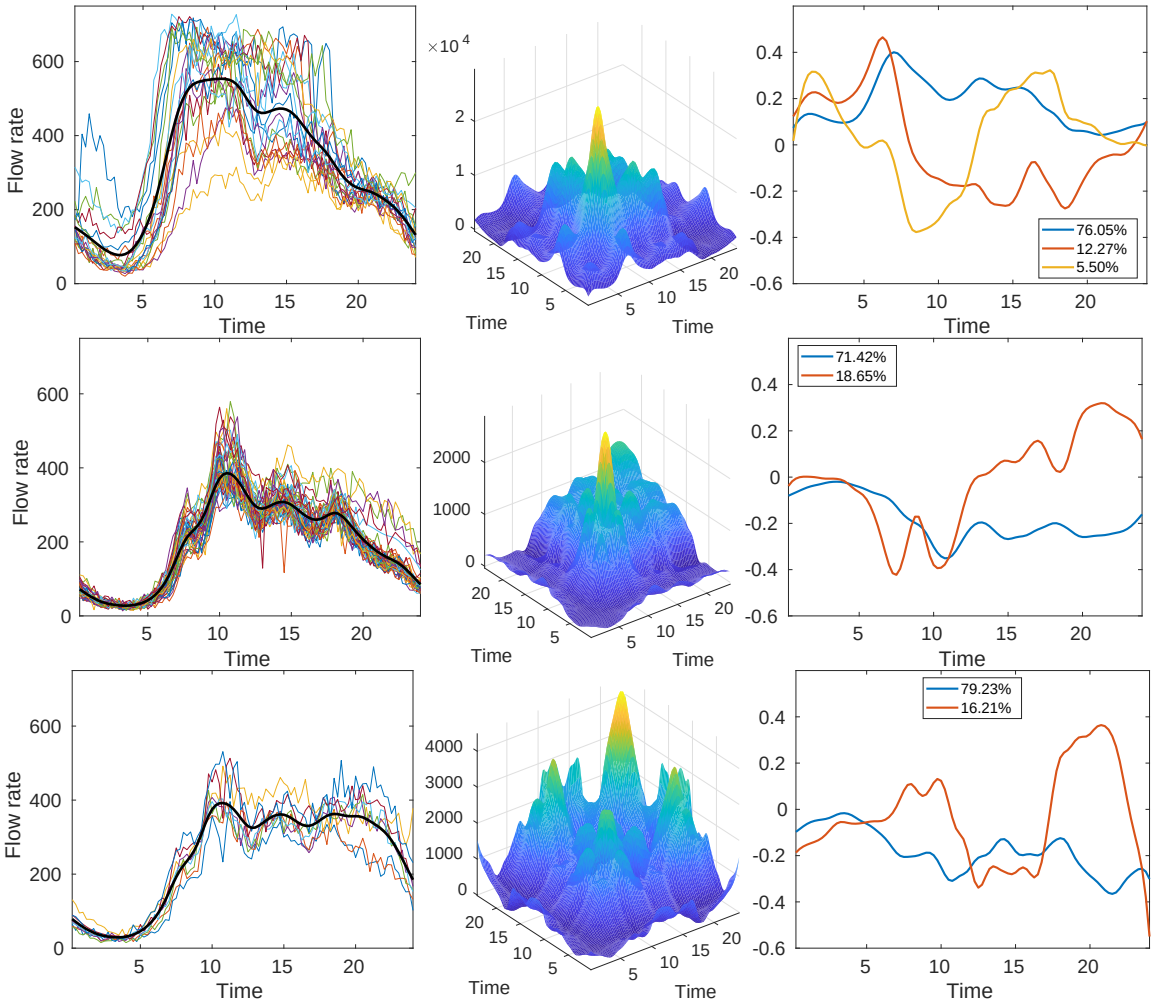


Figure 4.3: Estimated mean functions (left column) superimposed on the observed trajectories, covariance functions (middle column) and the correpsonding eigenfunctions (right column) of group 1-3 (from top to bottom) based on the training data of daily traffic flow trajectories.

integrated prediction error (TMIPE) across different values of τ can be calculated by

$$\text{TMIPE}(\omega, \kappa) \int_{\tau_s}^{\tau_e} \text{MIPE}(s, \omega, \kappa) ds,$$

where $\tau_s = \max(0, \tau - \omega)$ and $\tau_e = \min(\tau + \kappa, T)$, for $\omega, \kappa > 0$. In this study, $T = 24$ (hours) and we set $\tau_s = 8$ and $\tau_e = 20$. For notation convenience we further set $\omega^* = \tau$ to denote the maximal length of the past trajectory information available for prediction and $\kappa^* = 24 - \tau$ to denoted the interval length from the current time to the end of the day.

The prediction performance is then compared with the following methods:

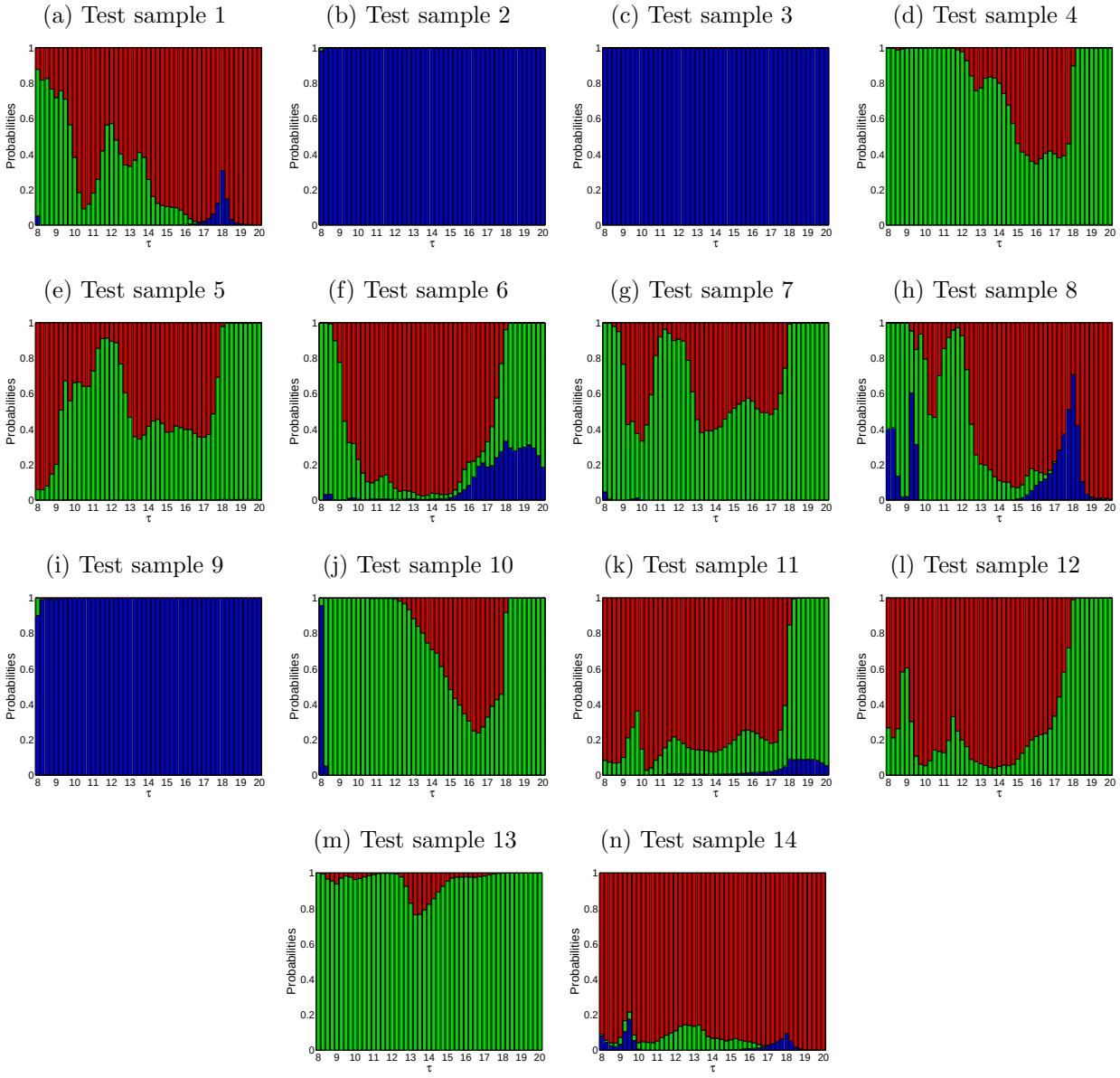


Figure 4.4: The predicted group membership distribution for Groups 1-3 (plotted in blue, green, and red) as a function of the "current time" τ for samples from the test data based on the trajectories observed up to τ.

- FP: Functional prediction based on function linear regression without considering groups. That is, we use all the training data to train the FLM without specific groups.
- FNP: Functional (naive) prediction based on group-specified functional linear regression and the group membership is simply determined by the functional naive Bayes

classifier in (4.3).

- FMP: Function mixture prediction based on group-specified functional linear regression and the posterior probability is calculated by (4.4).

To examine the effects on the prediction performance of interval length after the time τ, we consider various values of ω from $\omega = 1$ to $\omega = 6$ and $\omega = \omega^*$ and that of κ from $\kappa = 1$ to $\kappa = 10$ and $\kappa = \kappa^*$. Table 4.1 indicates that the proposed FMP has the smallest TMIPE compared to these of FP and FNP. It is not surprising that the prediction performance gets worse when κ increases for a fixed ω. In Figure 4.5, we plot the TMIPE as a function of ω for a fixed value of κ. The best prediction strategies for different values of κ are not quite clear to us. For different values of κ the best prediction varies. For example in Figure 4.5(a) for $\kappa = 1$ the best prediction would be $\omega = \omega^*$. However, in Figure 4.5(k) for $\kappa = \kappa^*$ the best prediction is $\omega = 1$. It not only depends on the available information prior to time τ, but also depends on the interval length after it. In Figure 4.6 we plot the TMIPE as a function of κ for a fixed value of ω. All Figures in 4.5 suggest the proposed functional mixture prediction outperform other methods.

4.4 Conclusion

This study proposed a prediction method that incorporates functional linear model and functional naive Bayes model as introduced in Chapter 3. Although it is motivated by the subject of traffic flow prediction, the proposed method can be generally applied to other subjects that are suitable for functional data analysis. Our real data analysis demonstrates that considering the group-specific prediction and posterior probability can work reasonably well to predict traffic flow. We thus conclude that taking the traffic flow patterns into account can greatly improve prediction performance.

Table 4.1: Performance comparisons for FP, FNP, and FMP based on TMIPE ($\times 10^3$) under various values of κ.

	κ \ ω	1	2	3	4	5	6	ω^*
FP	1	4.21	4.99	5.54	4.89	4.64	4.44	5.42
	2	6.16	6.47	6.70	6.02	5.72	5.59	7.05
	3	6.99	7.12	7.16	6.52	6.26	6.21	7.86
	4	7.44	7.69	7.63	7.05	6.81	6.89	8.56
	5	8.07	8.33	8.18	7.60	7.45	7.63	9.21
	6	8.71	8.95	8.67	8.19	8.12	8.36	9.90
	7	9.32	9.59	9.21	8.83	8.82	9.07	10.65
	8	9.92	10.24	9.79	9.48	9.49	9.74	11.38
	9	10.52	10.87	10.34	10.08	10.10	10.39	12.06
	10	11.07	11.43	10.85	10.64	10.68	10.99	12.69
	κ^*	12.34	12.77	12.12	11.97	12.06	12.40	14.11
FNP	1	3.51	3.28	3.22	3.33	3.58	3.58	3.15
	2	4.50	3.90	3.79	4.02	4.25	4.25	3.84
	3	4.67	4.11	4.15	4.38	4.64	4.64	4.20
	4	5.00	4.62	4.69	4.87	5.18	5.24	4.74
	5	5.86	5.48	5.49	5.66	6.01	6.10	5.66
	6	6.85	6.42	6.41	6.59	6.92	7.01	6.65
	7	7.77	7.33	7.32	7.51	7.82	7.90	7.59
	8	8.56	8.14	8.14	8.31	8.63	8.71	8.44
	9	9.31	8.89	8.88	9.05	9.37	9.46	9.25
	10	10.05	9.62	9.61	9.78	10.10	10.20	10.08
	κ^*	11.84	11.49	11.48	11.64	11.94	12.04	12.54
FMP	1	3.31	3.32	3.21	3.33	3.49	3.50	2.95
	2	4.25	3.85	3.75	3.99	4.13	4.13	3.57
	3	4.38	4.05	4.13	4.42	4.61	4.61	4.09
	4	4.48	4.36	4.51	4.77	5.00	5.06	4.62
	5	4.75	4.68	4.81	5.07	5.30	5.40	5.22
	6	5.04	4.95	5.10	5.37	5.60	5.70	5.75
	7	5.39	5.32	5.51	5.79	5.99	6.06	6.31
	8	5.80	5.77	5.96	6.24	6.45	6.54	6.99
	9	6.29	6.27	6.45	6.73	6.95	7.06	7.75
	10	6.75	6.72	6.91	7.19	7.42	7.54	8.48
	κ^*	8.00	8.07	8.26	8.53	8.75	8.87	10.87

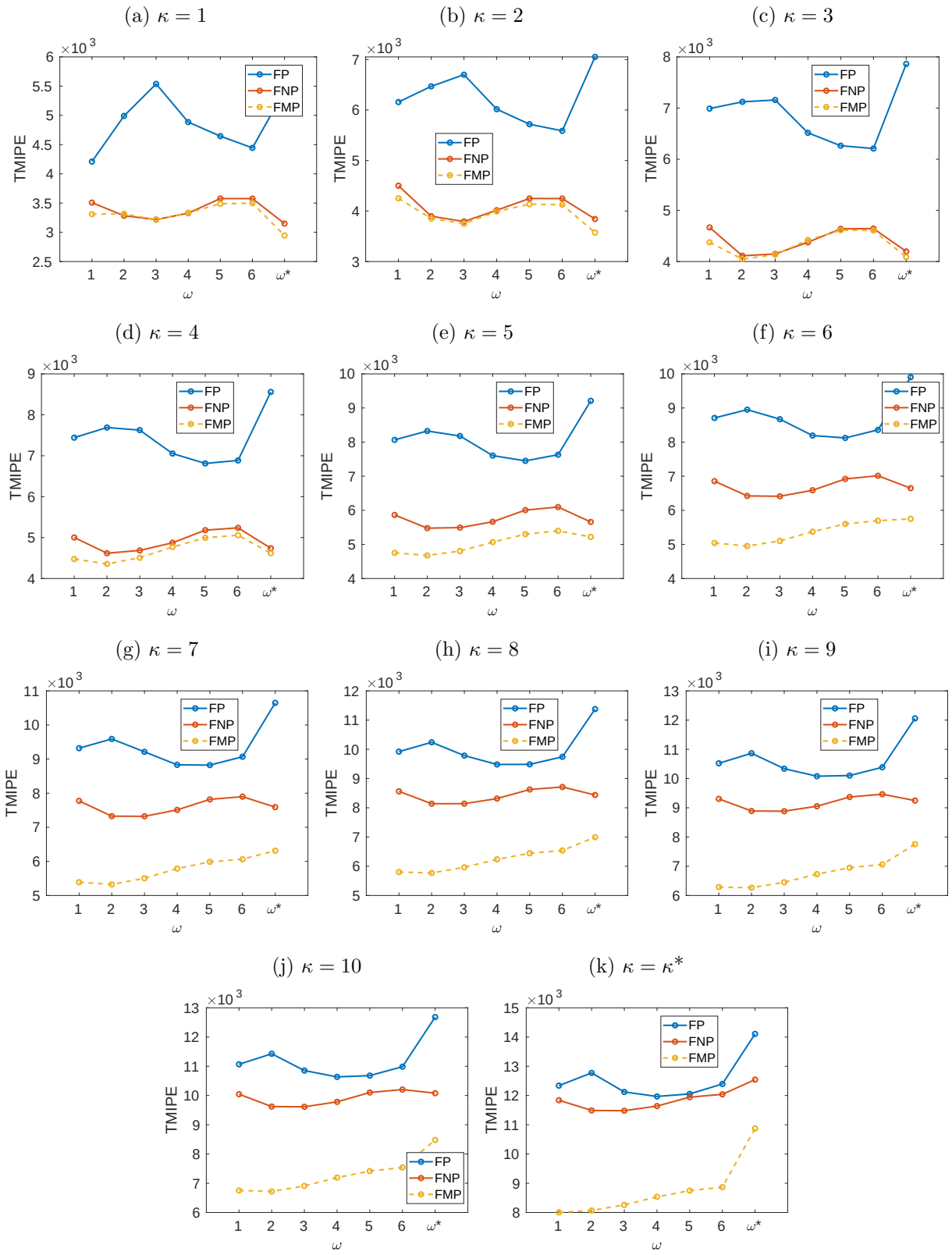


Figure 4.5: Performance comparisons for FP, FNP, and FMP, based on TMIPE, displayed as a function of ω from (a) $\kappa = 1$ to (k) $\kappa = \kappa^*$.

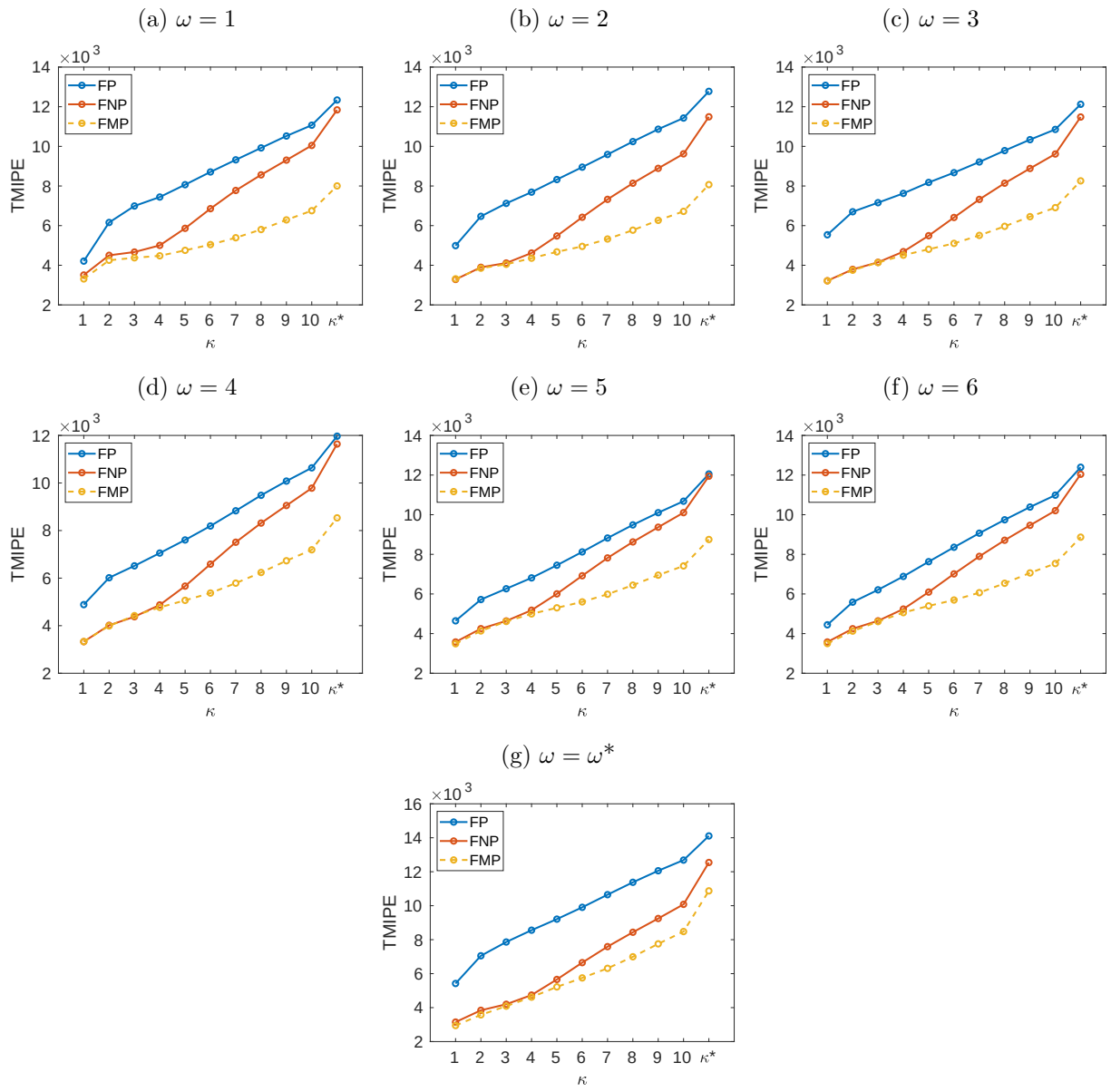


Figure 4.6: Performance comparisons for FP, FNP, and FMP, based on TMIPE, displayed as a function of ω from (a) $\omega = 1$ to (g) $\omega = \omega^*$.

CHAPTER 5

IDENTIFYING MULTIPLE MEAN CHANGE-POINTS OF FUNCTIONAL SEQUENCE

Change-point problem is an important topic in analysis of sequence data. Numerous methods have been implemented for detecting change-points in scalar and vector observations, but for functional observations, the related works are few. In this chapter, we are interested in detecting whether there are abrupt changes in the mean function of the data. Particularly, we assume that there are multiple change-points.

Multiple change-points occurs in many applications, especially when the sequence is long. A typical example is the vehicle volume flows on a highway with several interchanges. Figure 5.1 displays the pre-smoothed trajectories of the daily vehicle volumes on southbound National Highway No. 5 (NH5) in Taiwan from May 14, 2010 to May 16, 2010. The raw data are monitored by 84 electric detectors on a 5-minute interval, which results in the total of 288 records per day. In every panel, the trajectories are classified into three groups by different colors, which consist of detectors No. 1–70 (red), 71–79 (green) and 80–84 (blue) respectively. The partition boundaries, which are located at Toucheng interchange (between No. 70 and 71) and Luodong interchange (between No. 79 and 80), are identified by the algorithm in this chapter. It is easy to recognize that the mean trajectories of these three groups are different. This may imply that the total vehicle volumes may have changed at least twice among these intervals.

A conventional approach to detect and estimate the change-point is through hypothesis testing. Berkes et al. (2009) and Aue et al. (2009) are the first to discuss the mean change-point of a sequence of independent functional data. They introduce the partial sum based test, a common approach in univariate and multivariate scalar data to detect single mean change-point. The functional observations are transformed into a lower dimensional score vectors by functional principal component (FPC) analysis, and the test statistic is

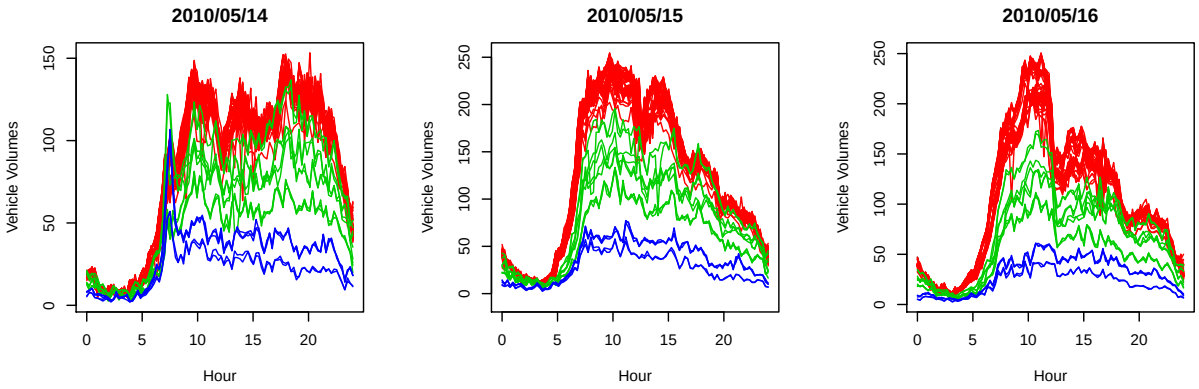


Figure 5.1: The trajectories of the total vehicle volumes on southbound National Highway No. 5 of Taiwan during May 14 – 16, 2010. The data were monitored by 84 electric detectors distributed along the highway, where each curve represents the daily record of a detector. The trajectories are identified as three groups with the segmentation method of this paper, where the red, green and blue lines represents detectors No. 1–70, 71–79 and 80–84 respectively.

constructed by the normalized quadratic form of the partial sum process of this score sequence. Later, the test is extended to serially correlated functional data by Hörmann and Kokoszka (2010) and Aston and Kirch (2012) under their weak dependence frameworks. They use the long-run covariance of the score vectors as the normalizer in the test statistic for standardizing the variance of the partial sum process.

These partial sum based tests are designed to detect single mean change-point. For multiple mean change-points, the methods should be applied through binary segmentation to iteratively find out all the change-points. That is, repeatedly performing the methods on the sub-sequences divided by the estimated change-points in previous tests. But there are two potential issues about this approach. First, these tests rely on the asymptotic distribution whose validity is doubtful as the sub-sequence gets shorter. Furthermore, if the FPC score sequence is dependent, the tests exhibit the non-monotonic power phenomenon due to oversized bandwidth when estimating the long-run covariance (Perron (1991); Vogelsang (1999)). To avoid this phenomenon, Zhang et al. (2011) adopt the adapted self-normalizer statistic as the normalizer to get rid of estimating the bandwidth. However, this test is powerful but the empirical size is often greater than expected in our experience. We are motivated to

develop a procedure that estimates the multiple change-points simultaneously to avoid the aforementioned drawbacks.

The multiple change-points problem can be viewed as partitioning the sequence into non-overlapped segments, where the partition boundaries represent the change-points and the interested statistical property is unchanged within each segment. In the literature, there are quite a few procedures in data mining and signal processing that deal with scalar sequence segmentation. One of the early partition algorithms is the dynamic programming proposed by Bellman (1961). Given the number of partition boundaries M, it starts with the sub-problem of finding the optimal partition with one boundary. The optimization process scans each possible candidate of this boundary and the results are stored. Then the algorithm proceeds to the second sub-problem of finding optimal partition with two boundaries, where the first boundary can be determined with the stored information of the first sub-problem. Similarly, the optimization process continues with further sub-problems until the main problem is solved and at each iteration, the information of previous sub-problems can be recycled to reduce the computation cost. Extension of the dynamic programming include the divide and segment algorithm (Terzi and Tsaparas, 2006), optimal segmentation algorithm (Auger and Lawrence, 1989), and pruned exact linear time algorithm (Killick et al., 2012), who also implement an R package '*changepoint*' (Killick and Eckley, 2014).

Greedy algorithm is another popular approach. It searches the change-points iteratively, one at a time, and stops when a certain criterion is met. The search path can be either forward (top-down) or backward (bottom-up). The former assumes no change-point at first and looks for a new partition boundary at each step. Binary segmentation also belongs to this type of approaches. The backward search assumes all or many of the data are partition boundaries and tries to merge two segments in the iteration by eliminating their boundary. The stopping criterion can be that the number of boundaries reaches a given M or the result of the verification procedure is no more significant. Notable references inlcude Shatkay and Zdonik (1996), Haiminen and Gionis (2004), Palpanas et al. (2004) and Terzi (2006).

Besides the above two types of methods, Himberg et al. (2001) propose the local/global iterative segmentation procedures that recursively adjust the partition boundaries with fixed number of segments. The local approach adjusts each boundary within the neighboring segments and the global approach adjusts it to the optimal position within the whole sequence. It is noted that most of the segmentation methods above focus on finding the positions of the boundaries rather than their number since they are executed under the assumption of fixed segment number.

In this chapter, we propose a two-step segmentation procedure to estimate both the number and locations of the mean change-points among a functional sequence. The first step is to list the possible locations of a given number of mean change-points with a recursive least square segmentation algorithm, and the second step is to remove the redundant points from the list with a backward elimination procedure. The raw functional observations, which are often recorded as high dimensional data with regular or irregular sampling frequencies, are presmoothed by basis functions such as the B-splines and projected on the low dimensional sub-eigenspace spanned by the leading FPC's. As long as the differences in the mean functions are not orthogonal to this sub-eigenspace, there will also be mean changes at the same locations in the sequence of projected score vectors. Thus, we can apply the proposed segmentation procedure on the score sequence to detect the mean change-points.

5.1 Functional Mean Change-Points

5.1.1 Multiple Mean Change-Points Model

Let $\mathcal{H} = \mathcal{L}^2([0,1])$ be a Hilbert space of square integrable functions defined on $[0,1]$, and $\mathcal{L}^p_{\mathcal{H}}$ be the space of $\mathcal{H}$ valued random variables X. Assume that a sequence of functional data $\{X_i\} \in \mathcal{H}$ are observed:

$$X_i(t) = Y_i(t) + \mu_0(t), \quad t \in [0,1],\ i = 1, \cdots, N, \tag{5.1}$$

where $\mu_0 \in \mathcal{H}$ is a deterministic function, and $\{Y_i\} \in \mathcal{H}$ is a sequence of mean zero random functions with $E\|Y_i\|^2 = \int EY_1^2(t)dt < \infty$. We assume that $\{Y_i\}$ have the same covariance function and are serially correlated under some weakly dependent framework. In the following content, we will omit the functional argument t for simplicity when there is no ambiguity.

A multiple change-points model with M change-points is:

$$X_i(t) = Y_i(t) + \sum_{m=1}^{M+1} \mu_m(t) \cdot \mathbf{1}_{(\theta_{m-1},\theta_m]}(i/N), \quad t \in [0,1],\ i = 1, \cdots, N, \tag{5.2}$$

where

$$\mathbf{1}_{(\theta_{m-1},\theta_m]}(x) = \begin{cases} 1, & \theta_{m-1} < x \leq \theta_m; \\ 0, & \text{otherwise}, \end{cases}$$

and $0 < \theta_1 < \theta_2 < \cdots < \theta_M < 1$ are the M positions of change. For notation convenience, we also denote $\theta_0 = 0$ and $\theta_{M+1} = 1$. The function $\mu_m \in \mathcal{H}$ is the deterministic mean function of the segment between the $(m-1)$-th and m-th changes. We also assume that $\mu_m \neq \mu_{m+1},\ \forall m = 1, \cdots, M$ for model identifiability. In this setting, the number of change-points M, their positions $\{\theta_m : m = 1, \cdots, M\}$ and the segmentwise mean functions $\{\mu_m(\cdot) : m = 1, \cdots, M+1\}$ are unknown parameters. Because the mean functions $\{\mu_m\}$ can be consistently estimated by segmentwise sample means as long as we have the estimated positions of the change-points, for our goal we will focus on the estimation of the change-points $\{M, \theta_1, \cdots, \theta_M\}$.

5.1.2 Functional Principal Components

By assuming the data are functional, the raw samples should be preprocessed before applying any statistical method. Both presmoothing and functional principal component (FPC) projection are the basic preprocessing tools in functional data analysis (Ramsay and Silverman, 2005). Presmoothing replaces the raw observations with their smoothed approximations, which reduces the random variations in the functions. The smoothed functions are then

projected into a lower dimensional space in order to reduce the data dimensionality. FPC analysis plays this role of dimension reduction for the infinite dimensional smooth functional data. Particularly, it preserves most of the variances in the space spanned by the leading components. If the data set contains significant mean changes, then the change information will also be counted in this projection space. This fulfills the basic requirement of the functional mean change-point procedures that the mean function differences should not be orthogonal to the projecting space. Therefore, FPC is a suitable choice of dimension reduction for the functional mean change-point problem.

Analogous to the multivariate principal component analysis, the components in FPC analysis are derived from the covariance function of the observations. Let the covariance operator of $\{Y_i\}$ be $C : \mathcal{L}^2([0,1]) \to \mathcal{L}^2([0,1])$, with the integration kernel $c(t,s) = E(Y_i(t)Y_i(s))$, $t, s \in [0,1]$. Assume that $c(t,s)$ satisfies the following decomposition:

$$c(t,s) = \sum_{l=1}^{\infty} \lambda_l v_l(t) v_l(s),$$

where $\lambda_1 > \lambda_2 > \cdots \geq 0$ are the eigenvalues and $\{v_l(\cdot) : l \geq 1\}$ are the corresponding orthonormal eigenfunctions of $c(t,s)$, satisfying

$$\int c(t,s) v_l(s) ds = \lambda_l v_l(t), \quad l = 1, 2, \cdots, \quad t \in [0,1].$$

Then we have the following Karhunen-Loéve decomposition:

$$Y_i(t) = \sum_{l=1}^{\infty} \eta_{i,l} v_l(t), \quad i = 1, \cdots, N, \tag{5.3}$$

where $\eta_{i,l} = \int_0^1 Y_i(t) v_l(t) dt$ is the projected score of Y_i on component v_l. It is noted that for any fixed i, $\{\eta_{i,l} : l = 1, 2, \cdots\}$ is a uncorrelated mean 0 random sequence with variance λ_l. The dimension reduction is done by representing the observed data X_i with its projected scores on the first d eigenfunctions of $c(t,s)$, where d is a positive integer such that $\lambda_1 > \cdots > \lambda_d > 0$.

In practice, $c(t,s)$ is unknown and needs to be estimated. A consistent estimator of the covariance kernel $c(t,s)$ under the model of no change-point (5.1) is the empirical covariance

function:

$$c_N(t,s) = \frac{1}{N}\sum_{i=1}^{N}\left(X_i(t) - \bar{X}_{1,N}(t)\right)\left(X_i(s) - \bar{X}_{1,N}(s)\right),$$

where $\bar{X}_{1,N} = N^{-1}\sum_{i=1}^{N} X_i$ is the overall sample mean. Let the eigenfunctions of $c_N(t,s)$ be $\{\nu_l(\cdot)\}$, and

$$\xi_{i,l} = \int X_i(t)\nu_l(t)dt, \quad l = 1, \cdots, d.$$

These scores $\boldsymbol{\xi}_i = (\xi_{i,1}, \cdots, \xi_{i,d})^T$, for $i = 1, \cdots, N$, will be used in the procedure for estimating the mean change-points.

5.1.3 Single Mean Change-Point Test

As we mentioned above, all currently available works concerning the mean change-point detection in a functional sequence consider the hypothesis of a single change-point, i.e., $M = 1$ in (5.2). Aston and Kirch (2012) also introduce an extra change type called the epidemic change, which states that the sequence changes its mean function at some time and changes back to the original mean at a later time. All these works adopt the partial sum based test, and we will address the steps in the following paragraphs.

Recall that $\eta_{i,l}$ is the projected score of Y_i on the l-th FPC. Denote $\boldsymbol{\eta}_i = (\eta_{i,1}, \cdots, \eta_{i,d})^T$. We note that if $\{Y_i\}$ are independent, then $\{\boldsymbol{\eta}_i\}$ are independent. Also, if $\{Y_i\}$ are weakly dependent, then $\{\boldsymbol{\eta}_i\}$ are weakly dependent in the same framework. From functional central limit theorem, if $\{\boldsymbol{\eta}_i\}$ are independent, then the partial sum process of $\{\boldsymbol{\eta}_i\}$:

$$\left\{ N^{-1/2}\sum_{i=1}^{\lfloor Nx \rfloor}\boldsymbol{\eta}_i : x \in [0,1] \right\} \xrightarrow{d} \left\{ \boldsymbol{\Sigma}^{1/2}\boldsymbol{W}_d(x) : x \in [0,1] \right\}, \tag{5.4}$$

where the covariance matrix $\boldsymbol{\Sigma}$ is diagonal with elements $\{\lambda_1, \cdots, \lambda_d\}$ and $\boldsymbol{W}_d$ is a d-dimensional Wiener process with independent components. If $\{\boldsymbol{\eta}_i\}$ are weakly dependent, then equation (5.4) still holds, but the matrix $\boldsymbol{\Sigma}$ is no longer a diagonal matrix. It becomes the long-run covariance matrix of $\{\boldsymbol{\eta}_i\}$, which is positive definite and defined by:

$$\boldsymbol{\Sigma} = \sum_{h=0}^{\infty} E\boldsymbol{\eta}_i\boldsymbol{\eta}_{i+h}^T + \sum_{h=-1}^{-\infty} E\boldsymbol{\eta}_{i-h}\boldsymbol{\eta}_i^T. \tag{5.5}$$

This indicates that for independent and weakly dependent functional data, their cumulative scores actually converge to the same Wiener process with different covariances.

Based on this property, we can derive a statistic featuring this asymptotic result in the following steps:

1. Project the centered functions $\{X_i - \bar{X}_{1,N}\}$ on the estimated d-dimensional sub-eigenspace through FPC and get scores $\{\boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}}_{1,N}\}$;

2. Compute the estimator of the covariance matrix $\boldsymbol{\Sigma}$ in (5.4), denote it by $\boldsymbol{\Sigma}_N$;

3. Let $\boldsymbol{Q}(x) = \sum_{i=1}^{\lfloor Nx \rfloor}(\boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}}_{1,N})$ be the partial sums of the centered scores, then

$$T_{N,d}(x) = N^{-1}\boldsymbol{Q}(x)^T\boldsymbol{\Sigma}_N^{-1}\boldsymbol{Q}(x) \overset{D^d_{[0,1]}}{\to} \boldsymbol{B}_d(x)^T\boldsymbol{B}_d(x), \tag{5.6}$$

   where $\boldsymbol{B}_d(x)$ is the d-dimensional Brownian bridge with independent components.

Various test statistics can be constructed with $T_{N,d}(x)$. For example, its integral or maximum are the mostly used. The matrix $\boldsymbol{\Sigma}_N$ is used as a normalizer in (5.6). When $\{Y_i\}$ are independent, it is the diagonal matrix with the d estimated eigenvalues of $c_N(t,s)$ as its diagonal elements. When $\{Y_i\}$ are weakly dependent, $\boldsymbol{\Sigma}_N$ is a consistent symmetric positive-definite estimator for $\boldsymbol{\Sigma}$ which is often computed by:

$$\begin{aligned}\boldsymbol{\Sigma}_N =& N^{-1}\sum_{h=0}^{q} w_q(h)\left\{\sum_{i=1}^{N-h}(\boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}}_{1,N})(\boldsymbol{\xi}_{i+h} - \bar{\boldsymbol{\xi}}_{1,N})^T\right\} + \\ & N^{-1}\sum_{h=-1}^{-q} w_q(h)\left\{\sum_{i=1}^{N-|h|}(\boldsymbol{\xi}_{i-h} - \bar{\boldsymbol{\xi}}_{1,N})(\boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}}_{1,N})^T\right\},\end{aligned} \tag{5.7}$$

where w_q is a kernel weight function with bandwidth q and the infinite sums in (5.5) are truncated here to be sums of $2q+1$ terms.

The parameter q can cause serious problem for detecting mean change-points. It controls the number of autocorrelation lags to be counted in the long-run covariance estimator. Intuitively, if the autocorrelation decays slowly, q should be large. Several data-driven methods were proposed to select a proper q among a stationary time series, e.g., Andrews (1991) and

Newey and West (1994), where the former was used in Hörmann and Kokoszka (2010). But when there is a mean change among the observations, the mean difference would enlarge the sample covariance and autocovariance functions, then these data-driven methods tend to provide an over-sized bandwidth. This oversized q would cause over-cumulation of the sums in (5.7), and results in an over-normalized test statistic. The greater the mean difference, the more the test statistic is over-normalized and its power shrinks. This is the so-called non-monotonic power phenomenon observed by Perron (1991) and Vogelsang (1999) in univariate and multivariate time series.

Another inconvenience of the partial sum based tests is the way to handle multiple change-points. They must be applied with binary segmentation to repeatedly find out the change-points. Therefore, at the first few iterations, since there are undetected change-points in the data, it is possible to produce insignificant tests due to the non-monotonic power phenomenon. Moreover, the sub-sequence length gets shorter in the latter iterations and the validity of the asymptotic distribution becomes doubtful.

In the next section, we will propose a procedure to estimate the number and locations of multiple change-points in the mean of a sequence of functional observations, which gets rid of the estimation of long-run covariances and takes the bottom-up like approach to avoid making decisions with contaminated statistics.

5.2 Backward Recursive Least Squares Segmentation

To assess the multiple change-points model (5.2), we need two assumptions.

1. The number of true change-points $M < N^{\alpha}$, for some $\alpha \in (0, 0.5)$.

   This assumption was rarely mentioned in the literature concerning multiple change-points. If we don't put this restriction, M can actually be maximized to N and then (5.2) may not be a proper model since the sequence changes its mean all the times. The upper bound 0.5 for α can be adjusted, according to the application. Because we adopt a bottom-up approach in our proposed procedure, we need this assumption to assure

the model adequacy and constrain M to a moderate size for the sake of shortening the search path and in order to reduce the computational cost.

2. The minimum segment length is greater than some positive number β, that is,

$$\min_{1\leq m\leq M}\{\theta_m - \theta_{m-1}\} > \beta.$$

This assumption also poses an implicit constraint that $M < \beta^{-1}$, but more importantly, it guarantees that there are at least $\lfloor N\beta \rfloor$ observations between any adjacent change-points, which provides sufficient samples for estimating the mean functions in each segment.

The choice of (α, β) is either subjective or data-dependent. For our algorithm, it is more convenient to assign an integer upper bound for M rather than choose α. Moreover, since a greater β would limit both the possible number of change-points and solution space during recursive optimization, we prefer to use simple candidates such as $2/N$ or $3/N$ for β.

We propose a new method to recursively search for the positions of the possible change-points with least squares, and via backward elimination to sequentially test and remove the redundant ones, till all the remaining change-points are significant. This procedure, named the Backward Recursive Least Square Segmentation (BRLSS), is applied on the FPC score vectors $\{\boldsymbol{\xi}_i\}$. It contains two major stages:

1. Partition the score sequence into K non-overlapping segments with recursive least squares segmentation (RLSS), where $K > N^{\alpha}$ is a positive integer;

2. Backward eliminating the estimated partition boundaries in pervious stage.

We will detail the steps in the subsections below.

5.2.1 Recursive Least Squares Segmentation

The idea of RLSS originates from the K-means algorithm for clustering data, because both methods look for subsets or partitions of the data where in each subset or partition, the

data share the same mean. We also adopt the devices of fixed number of segments, initial partitions and recursively updating approach in our algorithm. The main difference between RLSS and K-means is, rather than updating the membership of every observation, we only need to adjust the positions of the partition boundaries until they are not varying any more. This updating algorithm coincides with the local iterative replacement of Himberg et al. (2001), who state the generic algorithm and we give the detailed steps here.

Let $0 \leq a < b < c \leq 1$. Denote

$$S(a,b) = \sum_{i=\lfloor Na \rfloor + 1}^{\lfloor Nb \rfloor} \left(\boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}}_{\lfloor Na \rfloor + 1, \lfloor Nb \rfloor} \right)^T \left(\boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}}_{\lfloor Na \rfloor + 1, \lfloor Nb \rfloor} \right) \tag{5.8}$$

the sum of squared centered distances of the segment $\left\{ \boldsymbol{\xi}_{\lfloor Na \rfloor + 1}, \cdots, \boldsymbol{\xi}_{\lfloor Nb \rfloor} \right\}$, where

$$\bar{\boldsymbol{\xi}}_{\lfloor Na \rfloor + 1, \lfloor Nb \rfloor} = \frac{1}{\lfloor Nb \rfloor - \lfloor Na \rfloor} \sum_{i=\lfloor Na \rfloor + 1}^{\lfloor Nb \rfloor} \boldsymbol{\xi}_i.$$

If there is a mean change among $\left\{ \boldsymbol{\xi}_{\lfloor Na \rfloor + 1}, \cdots, \boldsymbol{\xi}_{\lfloor Nc \rfloor} \right\}$, a natural estimate of the position would be $\text{argmin}_b \{ S(a,b) + S(b,c) \}$. To avoid the ambiguity of multiple solutions and embed the interval length constraint β, we set the change-point estimator as

$$b^* = \inf \left\{ \underset{a+\beta < b \leq c-\beta}{\text{argmin}} \; S(a,b) + S(b,c) \right\}. \tag{5.9}$$

This 2-segmentation is the basic operation in RLSS.

Before executing RLSS, we need to determine an upper bound for the number of change-points, or equivalently, for the number of segments. Let this upper bound be K. In this section, we would regard K to be the upper bound for the segment number because it is easier to introduce our algorithm in the view of segments. Assign an initial partition in K segments for the sequence. This initial partition can be arbitrarily determined, but our simulation shows that an informative partition is preferred, we will provide a simple algorithm in section 5.2.3 for generating an informative partition.

Let the initial partition be $\boldsymbol{\theta}^{(0)} = \left\{ \theta_1^{(0)}, \cdots, \theta_{K-1}^{(0)} \right\}$, where the superscript stands for the number of iterations that the partition boundaries have been updated. For convenience,

we also set $\theta_0^{(r)} = \theta_0 = 0$ and $\theta_K^{(r)} = \theta_K = 1$ for all $r \in \mathbb{N}$. RLSS updates each element in $\boldsymbol{\theta}^{(0)}$ sequentially, one at a time, by applying the 2-segmentation in (5.9) to the concatenated segment made up of the two adjacent segments of the boundary being updated. The next boundary will be updated similarly with its left-hand-side segment partitioned by the latest updated boundary. We get $\boldsymbol{\theta}^{(1)}$ after updating all the boundaries in $\boldsymbol{\theta}^{(0)}$ and the same process repeats on $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \cdots$. RLSS terminates as soon as $\lfloor N\boldsymbol{\theta}^{(r)} \rfloor = \lfloor N\boldsymbol{\theta}^{(r-1)} \rfloor$ for some r, then the final partition boundaries $\boldsymbol{\theta}^{(r-1)}$ will be used in the next stage for backward elimination. The updating process is formulated in Algorithm 1.

Algorithm 1 The RLSS Algorithm

input: The FPC scores $\{\boldsymbol{\xi}_i\}$, the number of segments K, an initial partition $\left\{\theta_0^{(0)}, \cdots, \theta_K^{(0)}\right\}$, and the minimum segment length β.

Set $r = 0$

repeat

$r = r + 1$

for $k = 1$ to $K - 1$ **do**

Perform 2-segmentation on segment $\left(\theta_{k-1}^{(r)}, \theta_{k+1}^{(r-1)}\right]$:

$$\theta_k^{(r)} = \inf \left\{ \underset{\theta_{k-1}^{(r)} + \beta < \theta \le \theta_{k+1}^{(r-1)} - \beta}{\operatorname{argmin}} S\left(\theta_{k-1}^{(r)}, \theta\right) + S\left(\theta, \theta_{k+1}^{(r-1)}\right) \right\}$$

end for

until $\lfloor N\boldsymbol{\theta}^{(r)} \rfloor = \lfloor N\boldsymbol{\theta}^{(r-1)} \rfloor$

output: The final partition $\boldsymbol{\theta}^{(r-1)} = \left\{\theta_1^{(r-1)}, \cdots, \theta_{K-1}^{(r-1)}\right\}$

RLSS is guaranteed to stop within finite steps by the following theorem. The proof is given in the appendix.

Theorem 3. *Given the number of segments K, the total sums of squared centered distances (TSSCD) at the r-th loop $S_K(\boldsymbol{\theta}^{(r)}) = \sum_{k=1}^{K} S\left(\theta_{k-1}^{(r)}, \theta_k^{(r)}\right)$ will converge in finite r. The estimated partition boundaries $\left\{\boldsymbol{\theta}^{(r)} : r \in \mathbb{N}\right\}$ will also stop varying in finite r.*

5.2.2 Backward Elimination

The $K-1$ partition points $\boldsymbol{\theta}^{(r-1)}$ from the RLSS stage contain $K-1-M$ redundant points which should be removed. At this stage, we will eliminate these points iteratively with a validity check. The basic notion is to verify the most unlikely point in the current list. If it is verified as a true change-point, then the remaining points must all be real ones. But if not, we will discard this point and go on to check the next most unlikely point. This backward elimination will proceed interatively until a true change-point is found or we run out of the candidates.

Let $\boldsymbol{\theta}^{[0]} = \boldsymbol{\theta}^{(r-1)}$. After the k-th iteration, if the last checked most unlikely point is not significant, there will be $K-k-1$ points left on the list. Denote these $K-k-1$ point by $\boldsymbol{\theta}^{[k]} = \{\theta_1^{[k]}, \cdots, \theta_{K-k-1}^{[k]}\}$ and use $\mathcal{A}_j^{[k]} = (\theta_{j-1}^{[k]}, \theta_j^{[k]}]$ to represent the j-th segment in $\boldsymbol{\theta}^{[k]}$. The elimination procedure is described below:

1. Select the most unlikely point:

The sum of squared centered distances is an adequate tool for determining the most unlikely point. The TSSCD of the scores $\{\boldsymbol{\xi}_i\}$ under partition $\boldsymbol{\theta}^{[k]}$ is:

$$S_{K-k}(\boldsymbol{\theta}^{[k]}) = \sum_{j=1}^{K-k} S(\theta_{j-1}^{[k]}, \theta_j^{[k]}), \quad k = 1, \cdots, K-1,$$

here we also set $\theta_0^{[k]} = 0$ and $\theta_{K-k}^{[k]} = 1$ for all k with the same reason as in RLSS. It is easy to see that $S_{K-k}(\boldsymbol{\theta}^{[k]}) \leq S_{K-k-1}(\boldsymbol{\theta}^{[k]} \backslash \{\theta_j^{[k]}\})$ for all $\theta_j^{[k]} \in \boldsymbol{\theta}^{[k]}$.

When $\theta_j^{[k]} \in \boldsymbol{\theta}^{[k]}$ is to be removed at the $(k+1)$-th iteration, the segments $\mathcal{A}_j^{[k]}$ and $\mathcal{A}_{j+1}^{[k]}$ will merge as $\mathcal{A}_j^{[k+1]} = \mathcal{A}_j^{[k]} \bigcup \mathcal{A}_{j+1}^{[k]}$ and we have

$$S(\theta_{j-1}^{[k+1]}, \theta_j^{[k+1]}) - \left\{ S(\theta_{j-1}^{[k]}, \theta_j^{[k]}) + S(\theta_j^{[k]}, \theta_{j+1}^{[k]}) \right\} \geq 0.$$

If $\theta_j^{[k]}$ is a true change-point, this difference would be great due to different means in $\mathcal{A}_j^{[k]}$ and $\mathcal{A}_{j+1}^{[k]}$. But if not, this difference would be small. Therefore, it is reasonable to select the most unlikely point $\theta_{j^*}^{[k]}$ which minimizes this difference, which can also

be formulated as the increment between the two TSSCD's before and after removing it:

$$\theta_{j^*}^{[k]} = \underset{\theta \in \boldsymbol{\theta}^{[k]}}{\operatorname{argmin}} \left\{ S_{K-k}\big(\boldsymbol{\theta}^{[k]}\big) - S_{K-k-1}\big(\boldsymbol{\theta}^{[k]} \backslash \{\theta\}\big) \right\}.$$

2. Test if the most unlikely point is a real change-point:

We will use the fact that removing $\theta_{j^*}^{[k]}$ causes a nonnegative increment in the TSSCD to verify whether $\theta_{j^*}^{[k]}$ is a real change-point. Let $\{\boldsymbol{\zeta}_i^{[k]}\}$ be the score sequence centered under partition $\boldsymbol{\theta}^{[k]}$:

$$\boldsymbol{\zeta}_i^{[k]} = \boldsymbol{\xi}_i - \sum_{j=1}^{K-k} \bar{\boldsymbol{\xi}}_{\lfloor N\theta_{j-1}^{[k]} \rfloor + 1, \lfloor N\theta_j^{[k]} \rfloor} \cdot \mathbf{1}_{(\theta_{j-1}^{[k]}, \theta_j^{[k]}]}(i/N), \quad i = 1, \cdots, N,$$

and $\{\boldsymbol{\zeta}_i^{[k+1]}\}$ be defined similarly under partition $\boldsymbol{\theta}^{[k+1]} = \boldsymbol{\theta}^{[k]} \backslash \{\theta_{j^*}^{[k]}\}$. Both centered sequences have zero mean. If $\theta_{j^*}^{[k]}$ is not a real change-point, then the covariances of $\{\boldsymbol{\zeta}_i^{[k]}\}$ and $\{\boldsymbol{\zeta}_i^{[k+1]}\}$ must be quite close to each other. But if $\theta_{j^*}^{[k]}$ is a real change-point, there will be a mean shift in the segment $\mathcal{A}_{j^*}^{[k+1]}$, hence the sample covariance of $\{\boldsymbol{\zeta}_i^{[k+1]}\}$ will be contaminated by the mean shift and become different from that of $\{\boldsymbol{\zeta}_i^{[k]}\}$.

We adopt the $\mathcal{M}$-test of Box (1949) to compare these two covariance matrices:

$$\mathcal{M} = \frac{12N - 25}{6} \left\{ 2 \log \left| \boldsymbol{C}_{\text{pool}} \right| - \log \left| \boldsymbol{C}^{[k]} \right| - \log \left| \boldsymbol{C}^{[k+1]} \right| \right\},$$

where $\boldsymbol{C}^{[k]}$, $\boldsymbol{C}^{[k+1]}$ are the sample covariances of $\{\boldsymbol{\zeta}_i^{[k]}\}$ and $\{\boldsymbol{\zeta}_i^{[k+1]}\}$, and $\boldsymbol{C}_{\text{pool}} = \{\boldsymbol{C}^{[k]} + \boldsymbol{C}^{[k+1]}\}/2$ is their pooled covariance. If $\mathcal{M} > \chi^2_{\alpha,3}$, where α is the significance level, we reject the null hypothesis that the covariances are equal and conclude that $\theta_{j^*}^{[k]}$ is a change-point, then BRLSS terminates and $\boldsymbol{\theta}^{[k]}$ are the estimates of the real change-points. If the test is not rejected, the elimination procedure is repeated again with $\boldsymbol{\theta}^{[k+1]}$.

The select and test steps are conducted iteratively for $k = 1, \cdots, K-1$ until it is significant at some k. If none of the $K-1$ tests is significant, we conclude that there is no change-point in the sequence. The algorithm of backward elimination is listed in Algorithm 2.

Algorithm 2 The Backward Elimination Algorithm

input: The FPC scores $\{\boldsymbol{\xi}_i\}$, the number of segments K, the partition $\{\theta_1^{[0]}, \cdots, \theta_{K-1}^{[0]}\}$

Set the final number of change-point $M = K$

for $k = 1$ to $K - 1$ **do**

Choose the most unlikely point

$$\theta_{j^*}^{[k]} = \underset{\theta \in \boldsymbol{\theta}^{[k]}}{\operatorname{argmin}} \left\{ S_{K-k}\big(\boldsymbol{\theta}^{[k]}\big) - S_{K-k-1}\big(\boldsymbol{\theta}^{[k]} \backslash \{\theta\}\big) \right\}$$

Compute $\{\boldsymbol{\zeta}_i^{[k]}\}$, $\{\boldsymbol{\zeta}_i^{[k+1]}\}$ and the test statistic $\mathcal{M}$

if $\mathcal{M} > \chi^2_{\alpha,3}$ **then**

$M = K - k$, and the M points $\boldsymbol{\theta}^{[k]}$ are the estimated change-points

break

end if

end for

if $M = K$ **then**

$M = 0$

end if

output: The number of change-points M, and the location estimates $\boldsymbol{\theta}^{[k]}$

5.2.3 Some Remarks of BRLSS

We will discuss the BRLSS algorithm in the following three aspects: the initial partition, the backward elimination procedure, and the data dependency.

1. **Initial partition**:

   In RLSS stage, we need an initial partition and a simple choice is the equally spaced partition. However, like the K-means type algorithms, our algorithm also converges to local minimum sometimes. But if we use an informative initial partition, we can possibly avoid this unfavorable situation.

   A good initial partition should carry the information of the possible segmentation. We provide a small but useful algorithm basing on the 2-segmentation in (5.9), named the *proceeding 2-segmentation* (*P2S*), to generate a suggesting list of the initial partition boundaries. The algorithm is listed in Algorithm 3.

   The P2S algorithm applies the 2-segmentation on the expanding sub-sequences that start at the first observation and the end point proceeds from the given position to the

last observation. The estimated positions $\boldsymbol{\omega}_F$ from these expanding sub-sequences help to depict the profile of the scattering and importance of the changes. The duplicate times of a position reflect the significancy of the change occurs on it, so we can sort the distinct values in $\boldsymbol{\omega}_F$ by their frequencies and select the initial partition boundaries from the leading positions with highest frequencies.

There is a possibility that the most significant change occurs early so that the remaining changes after it are masked in $\boldsymbol{\omega}_F$. Therefore, we suggest to perform P2S on the reversed sequence $\{\boldsymbol{\xi}_{N-i} : i = 0, \cdots, N-1\}$ as well, and summarize the partition boundaries from both $\boldsymbol{\omega}_F$ and $\boldsymbol{\omega}_B$. Please note that the selected partition boundaries must be at least $\lfloor N\beta \rfloor$ points away from each other.

Algorithm 3 The P2S Algorithm

input: the FPC scores $\{\boldsymbol{\xi}_i\}$, the minimum segment length β

for $j = \lfloor N\beta \rfloor + 1$ to N **do**

Perform 2-segmentation on subsequence $\{\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_j\}$:

$$\omega_{F,j} = \inf \left\{ \underset{1 \le \omega \le j}{\operatorname{argmin}} \, S(0, \omega/N) + S(\omega/N, j/N) \right\}$$

Perform 2-segmentation on the reversed subsequence $\{\boldsymbol{\xi}_N, \cdots, \boldsymbol{\xi}_{N-j+1}\}$:

$$\omega_{B,j} = (N+1) - \inf \left\{ \underset{1 \le \omega \le j}{\operatorname{argmin}} \, S(0, \omega/N) + S(\omega/N, j/N) \right\}$$

end for

output: Estimated partition point lists: $\boldsymbol{\omega}_F = \{\omega_{F,j}\}$ and $\boldsymbol{\omega}_B = \{\omega_{B,j}\}$

The lists $\boldsymbol{\omega}_F$ and $\boldsymbol{\omega}_B$ can also reveal the change patterns among the sequence. Where there are many distinct ω_j's in $\boldsymbol{\omega}_F$ and $\boldsymbol{\omega}_B$, and the common elements in both lists are few, it could imply that there is no mean change-point. But if there are some common points in both lists and their frequencies are high, they are very likely to be the true change-points. We would suggest to select the points with relatively higher frequencies, which are $\lfloor N\beta \rfloor$ points apart from each other, to form the initial partition. It is always safe to select K that is greater than the number of the selected points by 2 or 3. The vacancy of these 2 or 3 extra boundaries can be filled with the middle points

of the first 2 or 3 longest segments in the selected partition.

2. **Test procedure in backward elimination**:

   The backward elimination procedure is not limited to the $\mathcal{M}$-test in this paper, it could be any other procedure which can correctly identify whether the removed boundary is a true change-point.

   The reasoning of backward elimination relies on the fact that the segmentwisely centered sequences $\{\boldsymbol{\zeta}_i^{[k]}\}$ and $\{\boldsymbol{\zeta}_i^{[k+1]}\}$ will have different means in $\mathcal{A}_{j^*}^{[k]}$ and $\mathcal{A}_{j^*+1}^{[k]}$ when a true change-point is removed. In fact, we have tried with many test procedures, for example, the two-sample t-test type procedures that verify the means of $\{\boldsymbol{\zeta}_i^{[k+1]}\}$ in these two segments are the same. We found that if we only consider the samples in these two segments, the test performance is usually not good due to limited sample size. On the other hand, if we use the full sequence, the mean change effect in these two segments will be diluted by other unchanged segments and the powers of these methods are also not satisfactory.

   Moreover, when the data actually contains no change-point, we also require this procedure to be insignificant in all iterations. In an analogous sense, we need to control the test size. Out of the many procedures we have tried, testing the equality of the covariance matrices of $\{\boldsymbol{\zeta}_i^{[k]}\}$ and $\{\boldsymbol{\zeta}_i^{[k+1]}\}$ $\mathcal{M}$-test generates the most impressive results, though there are some criticisms about its sensitivity to the departure of Gaussianity. But we would still emphasize that it is not the only answer in our algorithm.

3. **Dependency**:

   There is no special treatment for the serial correlation in BRLSS. When sequence $\{Y_i\}$ possesses a high positive autocorrelation, there would be some pseudo mean effects due to less flunctuations among the observations. For a nonparametric procedure that rely on the data magnitudes, it would often be confused by these pseudo mean effects and make wrong inferences. Our RLSS algorithm also fails at such case (see

Section 5.3.1). We believe that the highly positively correlated cases are hard to solve for many procedures, even the parametric methods. But there is no way to cancel or remove the correlation from the observations without impacting their mean. Recall in Section 5.2.2 that the long-run covariance is used by Hörmann and Kokoszka (2010) in the single change-point test as the normalizer. Their purpose is to derive an asymptotic distribution for the standardized test statistic. In our algorithm, the long-run covariance is irrelevant since we adopt the least square approach.

We cannot use standardized distances, which are normalized by some statistic such as the long-run covariance, in RLSS because the total sums of the squares of these standardized distances in the updating iterations are not necessary a decreasing sequence. This cannot guarantee the convergence of RLSS.

Even though we cannot handle the extreme case of highly positively correlated data, the simulations in Section 5.3.1 show that BRLSS can do well with mildly positively correlated data. BRLSS can also work with negatively correlated, even the highly negatively correlated cases, since there are more flunctuations and less pseudo mean effects in the data.

5.3 Simulation Study and Real Data Application

5.3.1 Simulation Study

In this section, we will use some simulated functional data to verify the cabability of the BRLSS algorithm. Consider the model (5.2) with $M = 0, 1, 2$. The three mean functions to be used:

$$\mu_1(t) = 0.5 - 100(t - 0.1)(t - 0.3)(t - 0.5)(t - 0.9)$$
$$\mu_2(t) = \mu_1(t) + 5t^2 - \exp(1 - 20t)$$
$$\mu_3(t) = \mu_1(t) + \sin(1 + 6\pi t),$$

are drawn in Figure 5.2. Two data generating models of $\{Y_i(t)\}$ are employed, namely the functional basis expansion model and the autoregressive Hilbertian process of order 1, which we denote by FBE and ARH(1) respectively.

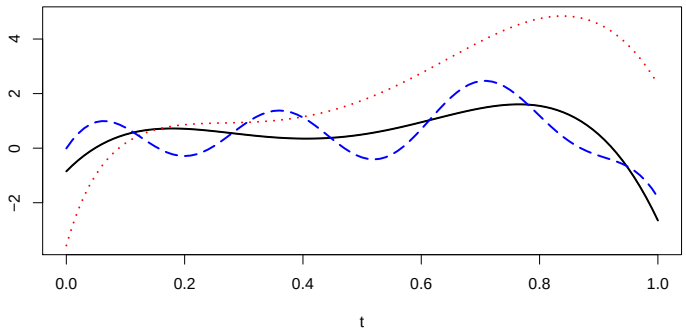


Figure 5.2: The mean functions used in simulation. The black solid line is $\mu_1(t)$, the red point line is $\mu_2(t)$ and the blue dash line is $\mu_3(t)$.

The FBE model is designed to imitate the Karhunen-Loéve decomposition (5.3), which is actually generated by

$$Y_i(t) = \sum_{l=0}^{L} \sqrt{\lambda_l}\, \tau_{i,l}\, \phi_l(t), \quad i = 1, \cdots, N,$$

where $\{\tau_{i,l} : i = 1, \cdots, N\}$ are random scores with zero mean and unit variance, $\{\phi_l(t)\}$ are the orthonormal basis functions that play the role of eigenfunctions and $\{\lambda_l\}$ are the corresponding eigenvalues. We set $L = 150$ and adopt the Fourier basis

$$\phi_l(t) = \begin{cases} \sqrt{2}\sin(2\pi k t - \pi), & l = 2k - 1; \\ \sqrt{2}\cos(2\pi k t - \pi), & l = 2k, \end{cases}$$

along with the strictly decreasing $\lambda_l = 0.7 \times 2^{-l}$.

To generate data with different levels of dependence, the random scores $\{\tau_{i,l}\}$ are simulated by the AR(1) model with different lag-1 coefficients, that is, for each l,

$$\tau_{i,l} = \rho\, \tau_{i-1,l} + \varepsilon_{i,l}, \quad i = 1, \cdots, N,$$

and the noise $\{\varepsilon_{i,l}\}$ are i.i.d. $N(0, 1-\rho^2)$. Four different ρ's are used: 0, 0.2, 0.5 and 0.8, each corresponds to the independent, weakly, mildly, and strongly correlated cases.

The ARH(1) model is defined as:

$$Y_i(t) = \Psi(Y_{i-1})(t) + \epsilon_i(t), \quad i = 1, \cdots, N,$$

with a sequence of independent standard Wiener processes $\{\epsilon_i(t)\}$, and the AR operator $\Psi(\cdot)$. The integral kernel of Ψ is $\psi(s,t) = c\{2-(2s-1)^2-(2t-1)^2\}$, for $s,t \in [0,1]$. The constant c is used to adjust the Hilbert-Schmidt norm $\|\Psi\|_S = \left\{\int\int |\psi(s,t)|^2\, ds\, dt\right\}^{1/2}$ so it would equal to some predefined values. Four values of c are selected so that $\|\Psi\|_S$ would equal 0, 0.2, 0.5, and 0.8. The larger $\|\Psi\|_S$ is, the more $\{Y_i(t)\}$ are correlated and for $c = 0$ (i.e., $\|\Psi\|_S = 0$), $\{Y_i(t)\}$ are independent.

We test for 0, 1 and 2 change-points in this simulation. For 1 change-point data, we designed three settings: $\theta_1 = 0.15, 0.5$, and 0.8 to represent the early, midterm, and late change respectively. Their combinations are used for the 2 change-points cases. These seven different change-point settings would combine with the four different dependencies to generate a total of 28 sub-models. 1000 sequences of length 100 and 500 are generated respectively for each sub-model, where all the functions are simulated on the discrete time points $\{0.01j : j = 1, \cdots, 100\}$. Every realization is pre-smoothed and projected on the FPC with R package '*fda*'.

The minimum segment length for BRLSS is 3, which corresponds to $\beta = 0.03$ for the sequences of length 100 and 0.006 for length 500. Moreover, two kinds of initial partitions with $K = 10$ are used: the equally spaced partition (ESP) and the P2S partition with the given β. The nominal size used by Box's $\mathcal{M}$-test in backward elimination is 0.05.

To analyze the simulation results, two quantities are evaluated: the number of change-points and their estimated locations. The precision of locations depends on the capability of RLSS, while the correctness of change-point number relies on backward elimination. We will check out these quantities by looking at the cases with or without changes separately.

5.3.1.1 Performance for Data without Change-Point

For data without mean change-point, we only need to check the number of estimated change-points. Table 5.1 lists the counts of correctly matched number of change-points for the sub-models that contain no change-point. That is, the number of samples with estimated $M = 0$ out of the 1000 realizations.

Table 5.1: Frequencies of correctly matched change-point number in 1000 no change-point samples.

FBE		ρ				**ARH(1)**		$\|\Psi\|_S$			
Partition	Size	0	0.2	0.5	0.8	Partition	Size	0	0.2	0.5	0.8
ESP	100	1000	1000	977	68	ESP	100	991	983	994	904
	500	1000	1000	1000	703		500	1000	1000	1000	994
P2S	100	1000	1000	980	88	P2S	100	1000	1000	999	889
	500	1000	1000	1000	736		500	1000	1000	1000	994

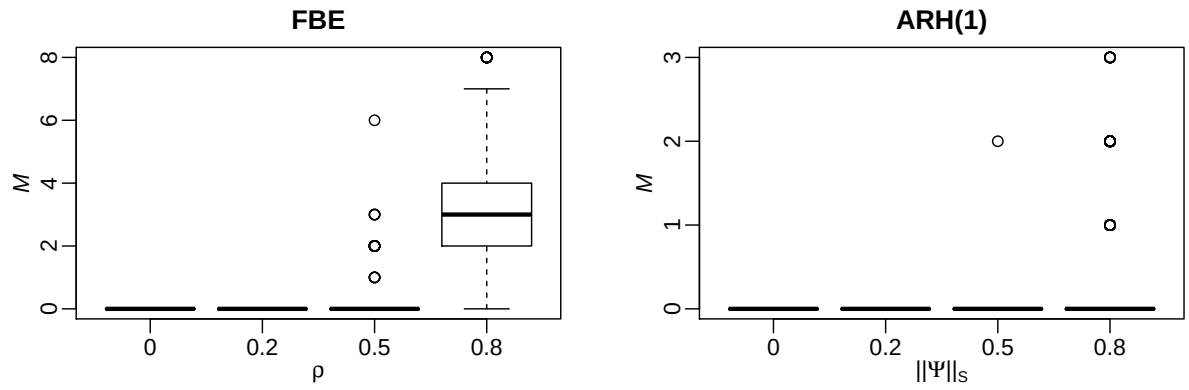


Figure 5.3: Box plots of the estimated number of change-point in the length 100 no change-point samples using P2S initial partition. The results of FBE are plotted in the left panel and ARH(1) in the right. The X-axis represents the correlation categories and the Y-axis is the estimated number of change-points.

The frequencies of correctly matched change-point number are all high except the FBE model with $\rho = 0.8$. Furthermore, almost all the ARH(1) samples with $\|\Psi\|_S \leq 0.5$ perfectly identify M besides the length 100 samples using equally spaced initial partition. This shows that the backward elimination with Box's $\mathcal{M}$-test can effectively identify the data of no

change-point, even when they are weakly dependent. The influence of initial partition can also be seen from these results, since there are even or more matched cases with P2S than with equally spaced partition in most of the sub-models.

We also draw box plots of the estimated M in Figure 5.3, using the length 100 samples with P2S initial partition. It is clear that the number of change-points are overestimated in the FBE model with $\rho = 0.8$, indicating the existence of one or more pseudo mean effects in the data, which are too strong for our method to handle.

5.3.1.2 Performance for Data with Change-Points

For the sub-models with mean change-points, we also look at the estimated number of change-points first. The frequencies of exactly matched number for one and two change-points data are listed in Table 5.2 and Table 5.3 respectively.

Table 5.2: Frequencies of correctly matched change-point number in 1000 one change-point samples.

FBE			ρ				**ARH(1)**			$\|\Psi\|_S$			
Partition	Size	$\boldsymbol{\theta}$	0	0.2	0.5	0.8	Partition	Size	$\boldsymbol{\theta}$	0	0.2	0.5	0.8
ESP	100	0.15	925	947	930	92	ESP	100	0.15	997	995	992	838
		0.50	931	963	932	111			0.50	977	982	991	876
		0.80	892	931	919	87			0.80	967	979	985	854
	500	0.15	988	990	997	650		500	0.15	1000	1000	1000	997
		0.50	989	993	993	602			0.50	1000	1000	1000	968
		0.80	944	978	990	660			0.80	1000	999	1000	956
P2S	100	0.15	998	996	944	124	P2S	100	0.15	1000	1000	997	834
		0.50	999	999	960	148			0.50	1000	1000	998	890
		0.80	1000	997	939	121			0.80	1000	999	998	859
	500	0.15	1000	1000	1000	722		500	0.15	1000	1000	1000	989
		0.50	1000	1000	1000	698			0.50	1000	1000	1000	983
		0.80	1000	1000	1000	725			0.80	1000	1000	1000	984

The patterns in Table 5.2 and Table 5.3 are quite similar to that in Table 5.1: the frequencies are high in most cases except the FBE model with $\rho = 0.8$. The results in these

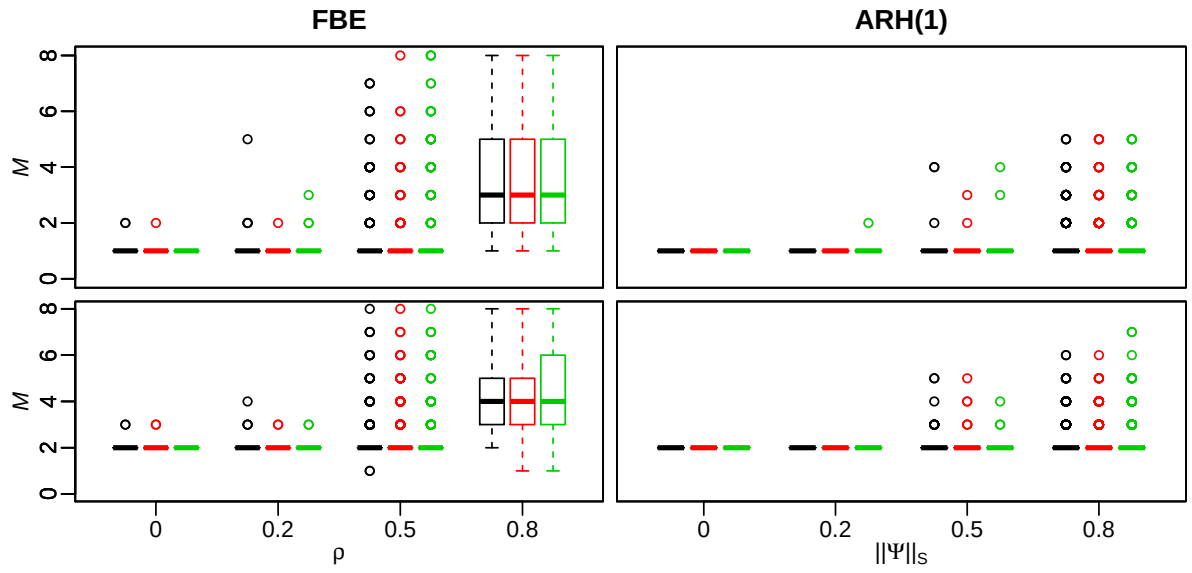


Figure 5.4: Box plots of the estimated number of change-point in length 100 samples with change-points using P2S initial partition. The upper panel are plots of 1 change-point data and the lower panel are of 2 change-points. The results of FBE are plotted in the left panel and ARH(1) in the right. The X-axis represents the correlation categories and the Y-axis is the estimated number of change-points. For each correlation category, different change-point settings are drawn from left to right in black ($\{0.15\}/\{0.15,0.50\}$), red ($\{0.50\}/\{0.15,0.80\}$) and green ($\{0.80\}/\{0.50,0.80\}$) respectively.

two tables demonstrate the capability of our backward elimination procedure, which stops at the exact iteration when the remaining number of change-points are correct.

The box plots of the estimated M in the length 100 samples using P2S initial partition are drawn in Figure 5.4 for one and two change-points data. Under each correlation category, the boxes of different change-point settings are drawn sequantially in different colors. Similarly, the estimated M in FBE model with $\rho = 0.8$ are overestimated and seem to have various answers. But in other sub-models, the estimated M's all concentrate on the estimands. This agrees with the results in Table 5.2 and Table 5.3.

The frequencies of correctly matched number of change-points increase when sample length becomes 500, implying that the change-point number estimator are more consistent as the sample size increases. Further, the influence of initial partition is still significant.

Now we turn to the location estimation of BRLSS. We are interested in the frequencies of

Table 5.3: Frequencies of correctly matched change-point number in 1000 two change-point samples.

FBE			ρ				**ARH(1)**			$\|\Psi\|_S$			
Partition	Size	$\boldsymbol{\theta}$	0	0.2	0.5	0.8	Partition	Size	$\boldsymbol{\theta}$	0	0.2	0.5	0.8
ESP	100	(0.15,0.50)	763	832	897	109	ESP	100	(0.15,0.50)	724	800	906	810
		(0.15,0.80)	733	864	885	104			(0.15,0.80)	849	887	939	812
		(0.50,0.80)	830	898	886	115			(0.50,0.80)	758	821	891	819
	500	(0.15,0.50)	903	955	977	550		500	(0.15,0.50)	959	970	996	952
		(0.15,0.80)	913	962	983	565			(0.15,0.80)	961	976	996	954
		(0.50,0.80)	942	983	985	537			(0.50,0.80)	950	964	992	949
P2S	100	(0.15,0.50)	998	996	944	179	P2S	100	(0.15,0.50)	1000	1000	989	859
		(0.15,0.80)	998	998	935	195			(0.15,0.80)	1000	1000	994	852
		(0.50,0.80)	1000	998	952	188			(0.50,0.80)	1000	1000	997	879
	500	(0.15,0.50)	1000	1000	1000	719		500	(0.15,0.50)	1000	1000	1000	989
		(0.15,0.80)	1000	1000	998	747			(0.15,0.80)	1000	1000	1000	986
		(0.50,0.80)	1000	1000	999	697			(0.50,0.80)	1000	1000	1000	990

both exactly estimated number and correctly matched locations of the change-points. In this simulation, the real change-points are known in advance, so we can count the frequencies of exactly matched locations. In many application when the real change-points are unknown, we wish that our location estimates would be as close to the real ones as possible. Therefore, we also check the frequencies of roughly matched, where the estimated locations are in the neighborhood of the true change-points with ± 2 offsets. The results of FBE and ARH(1) models are listed in Table 5.4 and Table 5.5 respectively.

From Table 5.4, we note that BRLSS correctly estimates both the number and positions of the change-points in more than 70% of the FBE samples with $\rho = 0, 0.2$ and 0.5, if we adopt the P2S initial partition. When we relax the precision of location estimate and allow rough matches within ± 2 offsets, the match percentages jump up to about 90%! This is useful in many applications. Our method can precisely point out the locations where mean changes occur, even if they are not exactly located on the true change-points, they would be only very few positions away. Similar pattern is found in Table 5.5, but the percentages of correctly matched (both exactly and roughly) are even higher.

Table 5.4: Frequencies of matched number and exactly/roughly matched locations of change-points in 1000 FBE samples.

Partition	Size	$\boldsymbol{\theta}$	Exact				Rough			
			0	0.2	0.5	0.8	0	0.2	0.5	0.8
ESP	100	(0.15)	840	855	784	56	923	939	898	73
		(0.50)	843	868	779	77	929	956	898	97
		(0.80)	808	825	755	55	890	924	884	72
		(0.15, 0.50)	659	698	689	67	758	823	860	95
		(0.15, 0.80)	679	740	682	56	768	856	848	77
		(0.50, 0.80)	725	776	685	70	829	894	849	95
	500	(0.15)	907	886	797	324	987	984	934	426
		(0.50)	905	886	835	340	985	989	953	419
		(0.80)	858	879	825	341	942	975	945	435
		(0.15, 0.50)	785	805	714	189	902	948	900	312
		(0.15, 0.80)	799	810	714	204	912	954	895	328
		(0.50, 0.80)	829	830	748	201	938	976	918	299
P2S	100	(0.15)	912	900	793	74	996	992	916	96
		(0.50)	915	889	806	97	999	991	932	126
		(0.80)	915	895	787	70	1000	989	908	94
		(0.15, 0.50)	858	832	697	92	995	982	897	136
		(0.15, 0.80)	874	845	705	98	996	986	892	144
		(0.50, 0.80)	849	817	695	81	993	977	893	116
	500	(0.15)	925	903	816	381	998	995	954	492
		(0.50)	919	891	841	395	998	998	954	504
		(0.80)	906	901	832	409	999	993	955	506
		(0.15, 0.50)	872	836	737	267	997	991	928	401
		(0.15, 0.80)	880	849	745	271	998	995	933	426
		(0.50, 0.80)	850	810	745	258	987	984	913	393

In general, a change-point that occurs in the middle of a sequence, is often easier to be detected than those near both ends. But the results in Table 5.4 and Table 5.5 show that there is no significant distinction among different change-point settings. Hence, BRLSS is robust to the change-point locations.

Table 5.5: Frequencies of matched number and exactly/roughly matched locations of change-points in 1000 ARH(1) samples.

Partition	Size	$\boldsymbol{\theta}$	Exact				Rough			
			0	0.2	0.5	0.8	0	0.2	0.5	0.8
ESP	100	(0.15)	942	925	909	771	996	992	984	824
		(0.50)	918	933	938	816	975	980	988	861
		(0.80)	885	926	931	806	963	976	979	845
		(0.15, 0.50)	665	722	798	727	724	798	898	795
		(0.15, 0.80)	779	795	829	715	849	885	930	795
		(0.50, 0.80)	706	748	823	732	758	818	886	801
	500	(0.15)	944	941	917	824	999	1000	981	912
		(0.50)	937	941	943	850	997	997	995	928
		(0.80)	929	929	939	832	999	996	994	909
		(0.15, 0.50)	877	893	878	727	958	970	977	856
		(0.15, 0.80)	882	896	874	735	960	976	974	873
		(0.50, 0.80)	875	886	900	773	947	961	987	891
P2S	100	(0.15)	954	933	910	761	998	992	984	824
		(0.50)	946	943	924	780	1000	980	988	861
		(0.80)	930	948	929	777	999	976	979	845
		(0.15, 0.50)	913	899	852	714	999	798	898	795
		(0.15, 0.80)	933	899	875	702	998	885	930	795
		(0.50, 0.80)	898	891	874	691	997	818	886	801
	500	(0.15)	946	943	913	806	999	1000	985	912
		(0.50)	947	940	930	817	1000	999	993	914
		(0.80)	929	934	921	824	1000	999	995	913
		(0.15, 0.50)	907	909	865	712	999	1000	984	853
		(0.15, 0.80)	920	918	867	707	999	1000	979	866
		(0.50, 0.80)	898	906	866	742	992	996	978	859

5.3.2 Data Application

In this section, we demonstrate BRLSS on the NH5 vehicle volume data described in the beginning of Chapter 5. NH5 is the first freeway connecting the eastern and western Taiwan. It is famous for the world's 5th longest road tunnel, the Hsuehshan Tunnel. Naturally the traffic flow control is critical for the government and the road users. Along both directions of NH5, a compact monitoring system, including 84 electric detectors on the southbound

and 86 on the northbound, is installed and several traffic indices are monitored. The indices are collected every 5 minutes and the daily profile which consists of 288 observations can be viewed as a functional sample.

Here we use the total volumes of all lanes in each direction during May 10 – 16, 2010. From Figure 5.1, the volume trajectories seem to shift downward several times. We would like to know the actual number and positions of these changes. All the 288-point curves are pre-smoothed and transformed to FPC scores with same programs as in the simulation. Here the FPC dimension is determined by the sum of variance proportions of the leading components that exceeds 80%. We apply BRLSS on these data and use the P2S initial partition with $K = 6$ (i.e., it begins with 5 change-points). The results are listed in Table 5.6.

Table 5.6: The sequentially selected most unlikely points and their Box's $\mathcal{M}$ statistics for vehicle volumes on southbound and northbound NH5 during May 10, 2010 and May 16, 2010. The asterisks denote significant tests under critical value $\chi^2_{0.05,3} = 7.8147$. The bold numbers are the estimated change-points.

		Iteration									
Direction	Date	1		2		3		4		5	
South	May 10	10	0.0124	29	0.1270	74	1.2213	**79**	*13.0161	**70**	32.9286
	May 11	11	0.2690	8	0.1642	74	0.7938	**79**	*12.8263	**70**	28.8395
	May 12	8	0.0301	29	0.5750	74	0.6480	**79**	*11.1850	**70**	30.5788
	May 13	8	0.0132	74	1.2952	29	0.9215	**79**	*13.5420	**70**	33.9575
	May 14	7	0.0036	29	0.5652	74	3.3232	**79**	*20.5770	**70**	40.9360
	May 15	15	0.0960	29	2.2935	74	2.8535	**79**	*20.5580	**70**	53.2198
	May 16	8	0.1313	74	2.3620	30	6.5459	**79**	*9.9222	**70**	47.3120
North	May 10	3	0.1890	9	0.0698	63	0.7259	81	1.2408	**76**	*38.5139
	May 11	33	0.0081	9	0.2376	63	0.5702	81	0.5674	**76**	*24.2468
	May 12	66	0.0367	63	0.2376	81	0.7212	33	0.9632	**76**	*25.9946
	May 13	63	0.0210	72	0.4319	81	0.7458	33	1.0611	**76**	*30.8025
	May 14	9	0.0508	72	0.3813	33	0.6515	81	0.8733	**76**	*32.4544
	May 15	63	0.0371	76	1.3343	33	1.2541	**81**	*8.2028	**71**	30.4828
	May 16	63	0.0410	76	1.5727	33	7.5704	81	3.0006	**72**	*39.5000

The southbound results are consistent, with two critical nodes No. 70 and 79. Both

detectors are at interchange exits, particularly, detector No. 70 is also at the end of Hsuehshan Tunnel. The northbound results are interesting. During May 10 – 14, 2010, which are weekdays, the volume flows change at detector No. 76. But the critical nodes switch to No. 71 and 81 on Saturday and then to No. 72 on Sunday. The northbound detectors No. 72, 76, and 81 are all located at the entrances of different interchanges, while No. 71 is the beginning of Hsuehshan Tunnel. In fact, the estimated northbound detectors can match the estimated southbound detectors geographically. The southbound detector No. 70 is close to the northbound detectors No. 71 and 72, and southbound No. 79 is about at the same place as northbound No. 81. Therefore, it is natural to link the volume mean changes with the vehicle input/output and different driving policies in different road sections divided by the identified interchanges. Furthermore, the weekday/weekend effect is also worth studying.

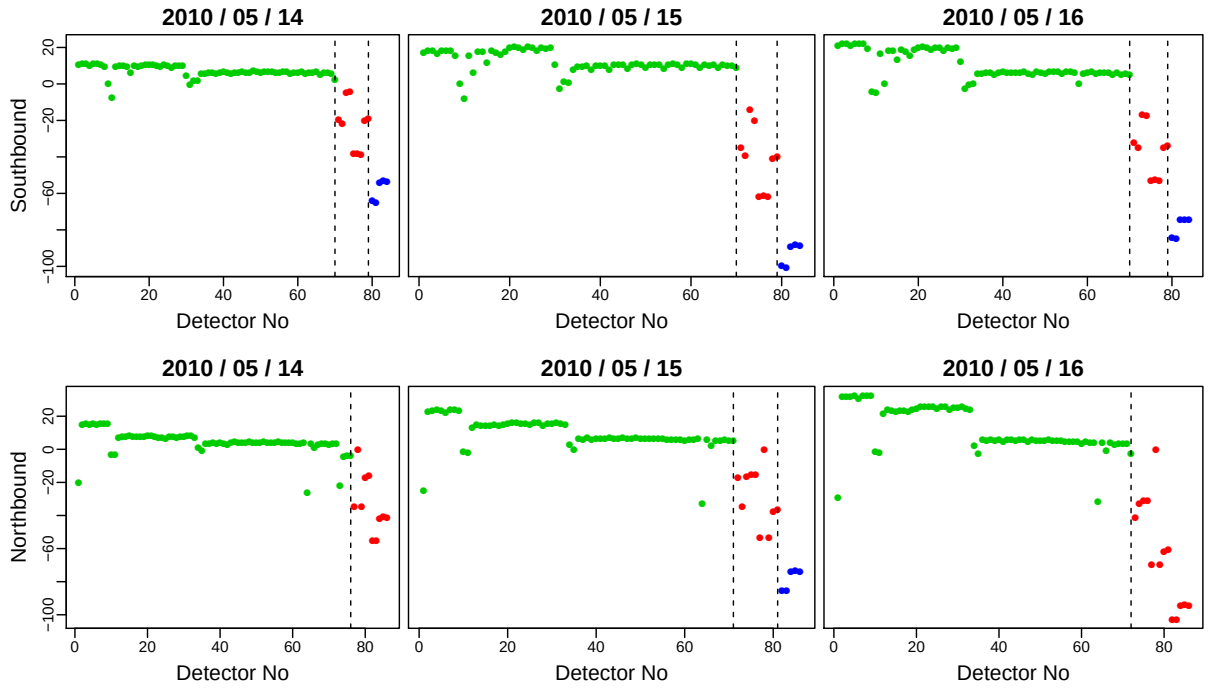


Figure 5.5: The sequential plots of the first FPC scores of the vehicle volumes. The southbound scores are in the upper panel and the northbound scores are in the lower panel. From left to right are May 14, 15 and 16. The dash lines are the estimated locations of the change-points.

The sequential plots of the first FPC scores in May 14 – 16 are plotted in Figure 5.5,

each with their estimated change-points marked (the dash lines). The mean shifts can be clearly identified at the estimated locations of the change-points. As argued in Aston and Kirch (2012), if the mean difference is significant, it would impact the FPC such that it is included in the sub-space spanned by the leading components. Hence, the mean difference is not orthogonal to that sub-space and is detectable for the change-point test procedures. For this data set, the changes are so obvious that they can be seen in the first component.

Here the projected FPC dimensions in the seven days are all one, regardless of the direction. That is, the first component would explain more than 80% of the total variation of the raw functions. Under this situation, the other components may carry very few information of the mean differences. Including them in the change-point detection procedure, either our BRLSS or the single change-point test of Berkes et al. (2009), will not help with the test power but sometimes may hamper the detectability. This is because the scores of the selected components are equally treated in the procedure, the importance of the informative component will reduce as more non-informative components are included. The detection results would become varied due to many unnecessary information in the rest components. To this end, we would suggest to use the weighted scores, that is, one can multiply the variance proportion of the component on its projected scores, to reduce the effects of non-informational components.

5.4 Conclusions

In this study, we proposed a two-step segmentation algorithm for detecting multiple mean change-points in a sequence of functional data. Functional data are transformed into functional principal component scores and used in the segmentation algorithm for estimating both the number and locations of the mean change-points. The first step of the algorithm is to estimate a given large number of change-point candidates with recursively least square updating, and the second step removes the redundant candidates with a backward elimination procedure. Simulation shows that this algorithm can accurately estimate both the number

and locations of the change-points among a functional sequence, even when the sequence is weakly dependent with mild autocorrelation. An application is illustrated with a sequence of highway vehicle volume flows in Taiwan. Our real data analysis demonstrates that the mean shifts can be captured at the estimated location of change-points. Overall, we conclude that the proposed BRLSS algorithm is successfully identified the change points in a functional sequence.

CHAPTER 6

CONCLUSIONS AND FUTURE DIRECTIONS

This thesis is concerned with the functional data analysis of daily curves in traffic flow data. The main obstacle in modeling of daily traffic flow trajectories is the problem of the "curse of dimensionality". Since the daily traffic flow data can be viewed as a stochastic process, this leads to the application of the functional data analysis framework. We have first reviewed the theorem of Karhunen-Loève expansion and further developed functional principal component model for modeling of traffic flow trajectories, where the mean function describes the trend of overall traffic flow pattern and the eigenfunctions take variations of each daily traffic flow into account. In addition, the infinite dimension functions can be reduced to a set of finite dimension functional principal component scores, which serves well as the proxy of each daily traffic traffic trajectory. This thesis has focused on the FPC scores to develop further analysis of traffic flow data.

In Chapter 2, we have introduced the functional data approach to deal with missing values in traffic flow data and compared the imputation performance of the proposed functional principal component model with those of probabilistic principal component and Bayesian principal component models. Moreover, based on the FPCA approach, the functional principal component scores can be applied to the functional bagplot and functional HDR boxplot. Using FPC scores derived from conditional expectation makes the outlier detection possible for incomplete functional data. Experiments with a simulated traffic flow data and a real traffic flow data have showed the effectiveness of functional data approach.

In Chapter 3, the naive Bayes classifier for functional data has been proposed. The novelty here is to make a naive assumption of common functional principal component scores to construct surrogate densities. Simulation studies have showed that the proposed functional naive Bayes classifier has a superior performance compared with other state-of-art classifiers. This result should advise a density point of view for the classification of functional

data.

In Chapter 4, we have introduced the functional naive prediction and functional mixture prediction for predicting the unobserved daily traffic flow trajectories. The idea is that we adapt the functional naive Bayes classifier to classify partially observed traffic flow trajectories and consider the group-specific functional linear model to predict the future unobserved traffic flow trajectory for a partially observed flow trajectory. Moreover, the posterior group membership probability can be calculated by surrogate density introduced in Chapter 3. The proposed functional data approaches, including classification and prediction, facilitate accurate prediction of daily traffic flow.

In Chapter 5, a two-step segmentation algorithm for detecting multiple mean change-points in a sequence of functional data has been proposed. Functional data are transformed into functional principal component scores which then are used in the segmentation algorithm for estimating both the number and locations of the mean change-points. The first step of the algorithm is to estimate a given large number of change-point candidates with recursive least square updating, while the second step removes the redundant candidates with a backward elimination procedure. Simulation shows that this algorithm can accurately estimate both the number and locations of the change-points among a functional sequence, even when the sequence is weakly dependent with mild autocorrelation.

In each chapter we have demonstrated that the proposed functional principal component methods have their own merits, and can outperform other investigated methods in traffic data applications. However, in this thesis we only worked with univariate functional data. In many situations, data can be collected on several variables simultaneously. This encourages the study of multivariate functional data and raises a natural motivation to extend the theory of univariate FPCA to the multivariate case. We briefly summarize the future research direction in the following section.

6.1 Future Work

In recent years there has been explosive growth in the number of neuroimaging studies performed using functional Magnetic Resonance Imaging (fMRI). A standard fMRI study gives rise to massive amounts of noisy data with a complicated spatio-temporal correlation structure (Haxby et al., 2001). The spatio-temporal correlation structures in voxels in fMRI data pose a challenge for the univariate functional principal component analysis since the voxels (in the spatio sense) should be considered jointly, and the correlation between them must be taken into account in the kernel covariance operator. Simply ignoring the correlation structure between voxels and performing a separate FPCA for each voxel make the interpretation of the FPCA results difficult. Multivariate functional data analysis (Chiou et al., 2014a) is potentially useful in analyzing such kind of spatio-temporal data because the covariation between voxels can be directly addressed by a single set of multivariate functional principal component scores, which serve well as a proxy for multivariate functional data.

In the study of Henderson et al. (2007), two cortical areas (posterior parahippocampal cortex and retrosplenial cortex) were investigated using fMRI to measure patterns of response while subjects viewed indoor, outdoor, and face pictures. Their findings suggest differences in function in these two areas. Here, we have a different approach. We aim to classify these fMRI brain images based on the pictures that subjects viewed. We extend the functional naive Bayes classifier presented in Chapter 3 from the univariate FPCA to the multivariate FPCA. The relationship between univariate and multivariate FPCA for the Karhunen-Loève representation is discussed in Happ and Greven (2017).

The densities of the first 10 multivariate functional common principal component scores are shown in Figure 6.1. It is clear that the densities for each picture are different and thus can be used to classify fMRI images. This result may lead to an idea of functional classification for fMRI data. Although the technique is still under development, the densities as shown in 6.1 look promising. More work is required to develop multivariate functional naive Bayes classifier.

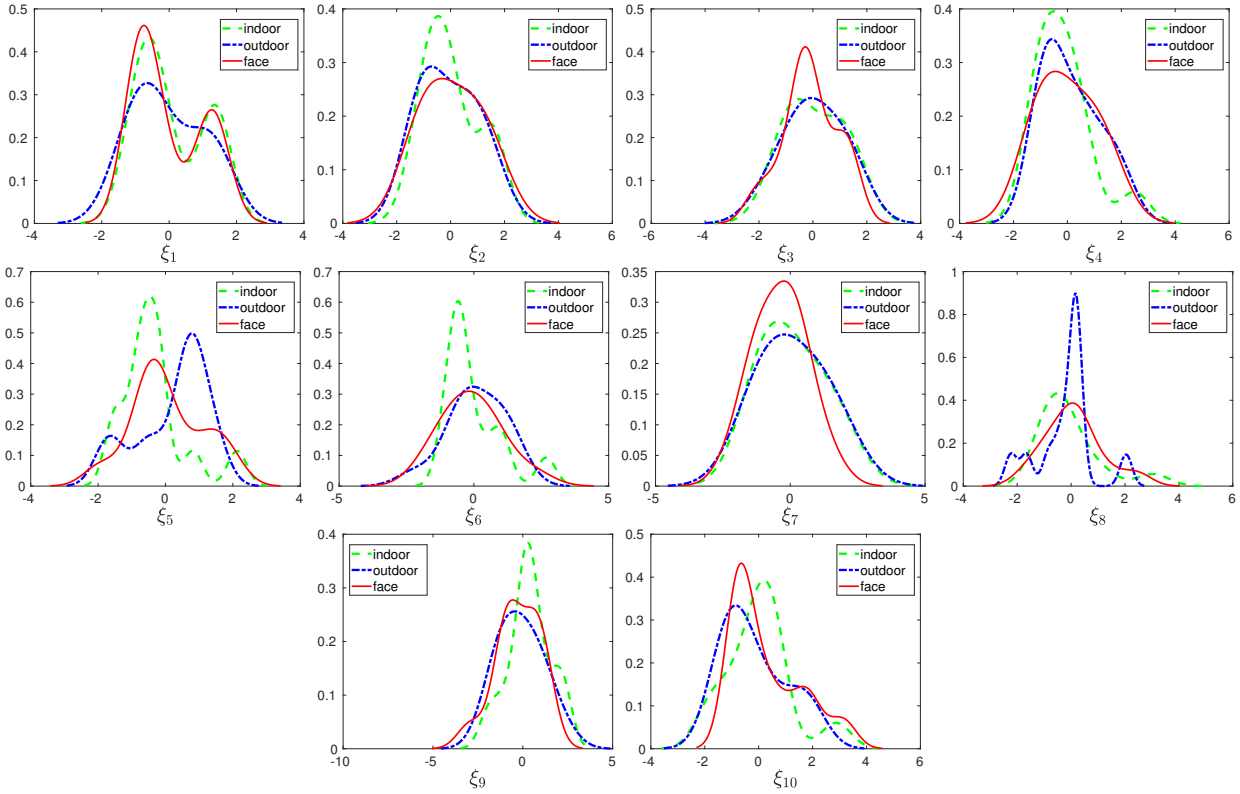


Figure 6.1: Kernel density estimates for the first 10 multivariate functional common principal component scores for the fMRI data.

APPENDIX

Appendix

PPCA Imputation Method

Probabilistic PCA (PPCA) is a probabilistic formulation of PCA based on a Gaussian latent variable model, first introduced by Tipping and Bishop (1999). The goal of such a model is to capture the covariance structure of an observed d dimensional variable $\boldsymbol{y}$ using a corresponding q dimensional latent variable $\boldsymbol{x}$ though a linear transformation function, where $q < d$. Let $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{id})^T$ be a set of observed variables for observation i and $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{iq})$ be a corresponding latent variable, and assume that $\boldsymbol{y}_i$ is produced by a linear transformation from $\boldsymbol{x}_i$ plus additive Gaussian noise. Denoting the transformation by the $d \times q$ matrix $\boldsymbol{W}$ and the d dimensional noise vector by $\boldsymbol{\epsilon}_i$, the PPCA model can be expressed as follows:

$$\boldsymbol{y}_i = \boldsymbol{W}\boldsymbol{x}_i + \boldsymbol{\mu} + \boldsymbol{\epsilon}_i,$$

where $\boldsymbol{\mu}$ permits the model to have nonzero mean. Conventionally, the latent variable $\boldsymbol{x}_i$ is assumed to follow $N_q(\boldsymbol{0}, \boldsymbol{I})$ where $\boldsymbol{I}$ is the identity matrix. Note that $\boldsymbol{x}_i$ can be viewed as the principal component score of observation i in terms of traditional PCA. By additionally specifying the noise to be Gaussian $\boldsymbol{\epsilon}_i \sim N_d(\boldsymbol{0}, \sigma^2\boldsymbol{I})$ and given the latent variable $\boldsymbol{x}_i$, the conditional distribution of the observed data can be expressed as

$$p(\boldsymbol{y}_i|\boldsymbol{x}_i) \sim N_d(\boldsymbol{W}\boldsymbol{x}_i + \boldsymbol{\mu}, \sigma^2\boldsymbol{I}).$$

With a Gaussian prior $\boldsymbol{x}_i$, we can obtain the marginal distribution of the observed data $\boldsymbol{y}_i$,

$$p(\boldsymbol{y}_i) \sim N_d(\boldsymbol{\mu}, \boldsymbol{C}),$$

where $\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2\boldsymbol{I}$. Using Bayes' rule, the posterior distribution of the latent variable $\boldsymbol{x}_i$ given the observed $\boldsymbol{y}_i$ may be calculated

$$p(\boldsymbol{x}_i|\boldsymbol{y}_i) \sim N_q(\boldsymbol{M}^{-1}\boldsymbol{W}^T(\boldsymbol{y}_i - \boldsymbol{\mu}), \sigma^2\boldsymbol{M}^{-1}), \tag{1}$$

where $\boldsymbol{M} = \boldsymbol{W}^T\boldsymbol{W} + \sigma^2\boldsymbol{I}$. Note that $\boldsymbol{M}$ is of dimension $q \times q$ and $\boldsymbol{C}$ is of dimension $d \times d$. Given a set of observed data recorded in vector $\boldsymbol{y} = \{\boldsymbol{y}_i\}$, $i = 1, \ldots, N$, the log-likelihood of the observed data under this model with unknown parameters $\boldsymbol{\theta} = (\boldsymbol{W}, \boldsymbol{\mu}, \sigma^2)$ can be written as

$$L(\boldsymbol{\theta}|\boldsymbol{y}) = \sum_{i=1}^{N} \ln\{p(\boldsymbol{x}_i|\boldsymbol{y}_i, \boldsymbol{\theta})\}.$$

The unknown parameters $\boldsymbol{\theta}$ for this model can be estimated by maximizing the log-likelihood $L(\boldsymbol{\theta}|\boldsymbol{y})$. One can simply apply the EM algorithm to achieve its maximum. In the Expectation-step, we take the expectation of log-likelihood L, which is evaluated using the current estimate for the parameters. In the Maximization-step, we compute parameters $\boldsymbol{W}, \boldsymbol{\mu}$ and σ^2 that maximize the expected log-likelihood found on the Expectation-step. The EM iteration is repeated until the algorithm converges. As shown by Tipping and Bishop (1999), the columns of $\boldsymbol{W}$ will span the principal subspace of conventional PCA when the log likelihood of the PPCA model is maximized. Thus the maximum likelihood estimate of the loadings matrix $\widehat{\boldsymbol{W}}$ in PPCA corresponds exactly to the loading matrix in the conventional PCA.

Now if $\boldsymbol{y}_i$ contains missing values, we can use $\hat{\boldsymbol{y}}_i = \widehat{\boldsymbol{W}}\tilde{\boldsymbol{x}}_i + \hat{\boldsymbol{\mu}}$ as an estimate for y_{ij} if y_{ij} is missing, where $\tilde{\boldsymbol{x}}_i$ is the posterior mean in (1). Further details of PPCA can be found in Tipping and Bishop (1999). The PPCA program is implemented in the package called '*pcaMethods*' in R software.

BPCA Imputation Method

BPCA, a Bayesian estimation method for PPCA, was proposed by Bishop (1999). It introduced some continuous hyper-parameters to determine the optimal value of the latent space dimensionality q which PPCA provides no mechanism for determining. In a fully Bayesian framework, both the number of principal components q and the model parameters $\boldsymbol{\theta}$ are considered to be drawn from appropriate prior distribution. According to Bayesian theory,

the posterior distribution of $\boldsymbol{\theta}$ and $\boldsymbol{x}$ is computed as

$$p(\boldsymbol{\theta}, \boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}, \boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}),$$

where $p(\boldsymbol{\theta})$ is the prior distribution determined before estmation. Following Oba et al. (2003), we assume that there are conjugate priors for $\boldsymbol{\mu}$ and σ^2 and further assume a hierarchical prior for $\boldsymbol{W}$, which is used to determine the best dimension of the latent space. We then have the joint prior:

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \equiv p(\boldsymbol{\mu}, \boldsymbol{W}, \tau|\boldsymbol{\alpha}) = p(\boldsymbol{\mu}|\tau)p(\tau) \prod_{j=1}^{q} p(\boldsymbol{w}_j|\tau, \alpha_j), \tag{2}$$

where $p(\boldsymbol{\mu}|\tau) \sim N_d(\boldsymbol{\mu}_0, (\nu_{\mu_0}\tau)^{-1}\boldsymbol{I})$, $p(\boldsymbol{w}_j|\tau, \alpha_j) \sim N_q(\boldsymbol{0}, (\alpha_j\tau)^{-1}\boldsymbol{I})$ and $p(\tau) \sim G(\tau_0, \nu_{\tau_0})$. Here $\boldsymbol{\alpha}$ is a q dimensional vector of hierarchical parameters used to control a column of $\boldsymbol{W}$ to diminish overfitting problem. The setting of all hyper-parameters are the same as in Oba et al. (2003), corresponding to an almost non-informative prior.

Bayesian inference is achieved by evaluating the posterior distribution of the unknown variables given the observations. In Oba et al. (2003), a variational Bayes algorithm is used to execute Bayesian estimation for both the model parameter $\boldsymbol{\theta}$ and missing values of $\boldsymbol{y}$. The posterior distributions for $\boldsymbol{\theta}$ and missing values of $\boldsymbol{y}$ are obtained via a repetitive algorithm. The missing values are imputed using the expectation of the estimated posterior distribution of missing values of $\boldsymbol{y}$ with optimal q.

Proof of Theorem 3

The proof of this theorem is carried out in two parts:

1. For a fixed K, the sequence $\{S_K\big(\boldsymbol{\theta}^{(r)}\big) : r = 1, 2, \cdots\}$ converges in finite r:

   Recall that in RLSS, we sequentially update each partition boundary with its adjacent segments by 2-segmentation in every iteration. Let $\big\{\boldsymbol{\theta}^{(r)}\big\} = \big\{\theta_0^{(r)}, \theta_1^{(r)}, \cdots, \theta_K^{(r)}\big\}$ be

the final boundaries in the r-th iteration. Because $\theta_0^{(r+1)} = \theta_0^{(r)} = 0$, then

$$\begin{aligned}\theta_1^{(r+1)} &= \inf\left\{\underset{\theta_0^{(r+1)}+\beta<\theta\leq\theta_2^{(r)}-\beta}{\operatorname{argmin}} S\big(\theta_0^{(r+1)},\theta\big) + S\big(\theta,\theta_2^{(r)}\big)\right\} \\ &= \inf\left\{\underset{\theta_0^{(r)}+\beta<\theta\leq\theta_2^{(r)}-\beta}{\operatorname{argmin}} S\big(\theta_0^{(r)},\theta\big) + S\big(\theta,\theta_2^{(r)}\big)\right\}.\end{aligned}$$

Hence, we know that

$$S\big(\theta_0^{(r+1)},\theta_1^{(r+1)}\big) + S\big(\theta_1^{(r+1)},\theta_2^{(r)}\big) \leq S\big(\theta_0^{(r)},\theta_1^{(r)}\big) + S\big(\theta_1^{(r)},\theta_2^{(r)}\big).$$

Similarly, because $\theta_{k-1}^{(r+1)} < \theta_k^{(r)} < \theta_{k+1}^{(r)}$, we also have

$$S\big(\theta_{k-1}^{(r+1)},\theta_k^{(r+1)}\big) + S\big(\theta_k^{(r+1)},\theta_{k+1}^{(r)}\big) \leq S\big(\theta_{k-1}^{(r+1)},\theta_k^{(r)}\big) + S\big(\theta_k^{(r)},\theta_{k+1}^{(r)}\big), \tag{3}$$

for all $k = 1,\cdots,K-1$. Then

$$\begin{aligned}S_K\big(\boldsymbol{\theta}^{(r+1)}\big) &= \sum_{k=1}^{K} S\big(\theta_{k-1}^{(r+1)},\theta_k^{(r+1)}\big) \\ &= \sum_{k=1}^{K-1}\Big\{S\big(\theta_{k-1}^{(r+1)},\theta_k^{(r+1)}\big) + S\big(\theta_k^{(r+1)},\theta_{k+1}^{(r)}\big)\Big\} - \sum_{k=1}^{K-2} S\big(\theta_k^{(r+1)},\theta_{k+1}^{(r)}\big), \\ S_K\big(\boldsymbol{\theta}^{(r)}\big) &= \sum_{k=1}^{K} S\big(\theta_{k-1}^{(r)},\theta_k^{(r)}\big) \\ &= \sum_{k=1}^{K-1}\Big\{S\big(\theta_{k-1}^{(r+1)},\theta_k^{(r)}\big) + S\big(\theta_k^{(r)},\theta_{k+1}^{(r)}\big)\Big\} - \sum_{k=2}^{K-1} S\big(\theta_k^{(r+1)},\theta_{k+1}^{(r)}\big),\end{aligned}$$

and from (3),

$$\begin{aligned}&S_K\big(\boldsymbol{\theta}^{(r+1)}\big) - S_K\big(\boldsymbol{\theta}^{(r)}\big) = \\ &\sum_{k=1}^{K-1}\Big\{S\big(\theta_{k-1}^{(r+1)},\theta_k^{(r+1)}\big) + S\big(\theta_k^{(r+1)},\theta_{k+1}^{(r)}\big) - S\big(\theta_{k-1}^{(r+1)},\theta_k^{(r)}\big) - S\big(\theta_k^{(r)},\theta_{k+1}^{(r)}\big)\Big\} \leq 0.\end{aligned} \tag{4}$$

Since $S_K\big(\boldsymbol{\theta}^{(r)}\big)$ is nonnegative and nonincreasing with r, it will converge in finite r.

2. For a fixed K, sequence $\{S_K(\boldsymbol{\theta}^{(r)}) : r = 1, 2, \cdots\}$ converges in finite r if and only if the partition boundaries $\{(\hat{\theta}_1^{(r)}, \cdots, \hat{\theta}_{K-1}^{(r)}) : r = 1, 2, \cdots\}$ stop varying in finite r.

   The necessity is trivial, hence we shall only prove the sufficiency that:

$$\{S_K(\boldsymbol{\theta}^{(r)})\} \text{ converges in finite } r \;\Rightarrow\; \{\boldsymbol{\theta}^{(r)}\} \text{ stops varying in finite } r.$$

   Suppose that for some $r_0 \in \mathbb{N}$, $S_K(\boldsymbol{\theta}^{(r)}) = S_K(\boldsymbol{\theta}^{(r_0)})$, $\forall\, r \geq r_0$, but $\boldsymbol{\theta}^{(r)} \neq \boldsymbol{\theta}^{(r_0)}$ for at least one element, say $\theta_k^{(r)}$.

   Since $S_K(\boldsymbol{\theta}^{(r_0+1)}) - S_K(\boldsymbol{\theta}^{(r_0)}) = 0$, then from (3) and (4), we have the following $K-1$ equations:

$$S(\theta_{k-1}^{(r_0+1)}, \theta_k^{(r_0+1)}) + S(\theta_k^{(r_0+1)}, \theta_{k+1}^{(r_0)}) = S(\theta_{k-1}^{(r_0+1)}, \theta_k^{(r_0)}) + S(\theta_k^{(r_0)}, \theta_{k+1}^{(r_0)}),$$

   $\forall k = 1, \cdots, K-1$. This means that both $\theta_k^{(r_0)}$ and $\theta_k^{(r_0+1)}$ produce the same TSSCD when performing 2-segmentation on $(\theta_{k-1}^{(r_0+1)}, \theta_{k+1}^{(r_0)}]$. Because $\theta_k^{(r_0+1)}$ is the minimizer of this 2-segmentation, from (5.9), we have $\theta_k^{(r_0+1)} \leq \theta_k^{(r_0)}$. Similar argument applies and results in $\theta_k^{(r_0+h)} \leq \theta_k^{(r_0+h-1)}$, $\forall h \in \mathbb{N}$. The decreasing sequence $\{\theta_k^{(r)}, r \geq r_0\}$ must not fall below $\theta_{k-1}^{(r_0)}$, therefore, it will converge in finite steps.

BIBLIOGRAPHY

BIBLIOGRAPHY

Allison, P. D. (1999). *Missing data*. Sage Thousand Oaks, CA.

Alonso, A. M., Casado, D., and Romo, J. (2012). Supervised classification for functional data: A weighted distance approach. *Computational Statistics & Data Analysis*, 56(7):2334–2346.

Andrews, D. W. K. (1991). Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica*, 59(3):817–858.

Araki, Y., Konishi, S., Kawano, S., and Matsui, H. (2009). Functional logistic discrimination via regularized basis expansions. *Communications in Statistics - Theory and Methods*, 38(16-17):2944–2957.

Aston, J. A. D. and Kirch, C. (2012). Detecting and estimating changes in dependent functional data. *Journal of Multivariate Analysis*, 109:204–220.

Aue, A., Gabrys, R., Horváth, L., and Kokoszka, P. (2009). Estimation of a change-point in the mean function of functional data. *Journal of Multivariate Analysis*, 100(10):2254–2269.

Auger, I. E. and Lawrence, C. E. (1989). Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, 51(1):39–54.

Baesens, B., Gestel, T. V., Viaene, S., Stepanova, M., Suykens, J., and Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, 54(6):627–635.

Beale, E. M. L. and Little, R. J. A. (1975). Missing values in multivariate analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 37(1):129–145.

Bellman, R. (1961). On the approximation of curves by line segments using dynamic programming. *Commun. ACM*, 4(6):284.

Berkes, I., Gabrys, R., Horváth, L., and Kokoszka, P. (2009). Detecting changes in the mean of functional observations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):927–946.

Besse, P. (1992). PCA stability and choice of dimensionality. *Statistics & Probability Letters*, 13(5):405–410.

Besse, P. and Ramsay, J. O. (1986). Principal components analysis of sampled functions. *Psychometrika*, 51(2):285–311.

Biau, G., Bunea, F., and Wegkamp, M. H. (2005). Functional classification in Hilbert spaces. *IEEE Transactions on Information Theory*, 51(6):2163–2172.

Biau, G., Cerou, F., and Guyader, A. (2010). Rates of convergence of the functional k-nearest neighbor estimate. *IEEE Transactions on Information Theory*, 56(4):2034–2040.

Bishop, C. (1999). Bayesian PCA. *Microsoft Research*, 11.

Boente, G., Rodriguez, D., and Sued, M. (2010). Inference under functional proportional and common principal component models. *Journal of Multivariate Analysis*, 101(2):464–475.

Bongiorno, E. G. and Goia, A. (2016). Classification methods for Hilbert data based on surrogate density. *Computational Statistics & Data Analysis*, 99:204–222.

Box, G. E. P. (1949). A general distribution theory for a class of likelihood criteria. *Biometrika*, 36(3/4):317–346.

Cardot, H. (2000). Nonparametric estimation of smoothed principal components analysis of sampled noisy functions. *Journal of Nonparametric Statistics*, 12(4):503–538.

Castro, P. E., Lawton, W. H., and Sylvestre, E. A. (1986). Principal modes of variation for processes with continuous sample curves. *Technometrics*, 28(4):329–337.

Cérou, F. and Guyader, A. (2006). Nearest neighbor classification in infinite dimension. *ESAIM: Probability and Statistics*, 10:340–355.

Chen, C., Kwon, J., Rice, J., Skabardonis, A., and Varaiya, P. (2003). Detecting errors and imputing missing data for single-loop surveillance systems. *Transportation Research Record: Journal of the Transportation Research Board*, 1855:160–167.

Chen, J. and Shao, J. (2000). Nearest neighbor imputation for survey data. *Journal of official statistics*, 16(2):113.

Chiou, J.-M. (2012). Dynamical functional prediction and classification, with application to traffic flow prediction. *The Annals of Applied Statistics*, 6(4):1588–1614.

Chiou, J.-M., Chen, Y.-T., and Yang, Y.-F. (2014a). Multivariate functional principal component anlaysis: A normalizatoin approach. *Statistica Sinica*, 24(4):1571–1596.

Chiou, J.-M. and Li, P.-L. (2007). Functional clustering and identifying substructures of longitudinal data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):679–699.

Chiou, J.-M., Zhang, Y.-C., Chen, W.-H., and Chang, C.-W. (2014b). A functional data approach to missing value imputation and outlier detection for traffic flow data. *Transportmetrica B: Transport Dynamics*, 2(2):106–129.

Coffey, N., Hinde, J., and Holian, E. (2014). Clustering longitudinal profiles using p-splines and mixed effects models applied to time-course gene expression data. *Computational Statistics & Data Analysis*, 71:14–29.

Collins, L. M., Schafer, J. L., and Kam, C. M. (2001). A comparison of inclusive and restrictive strategies in modern missing data procedures. *Psychological Methods*, 6(4):330–351.

Croux, C. and Ruiz-Gazen, A. (2005). High breakdown estimators for principal components: the projection-pursuit approach revisited. *Journal of Multivariate Analysis*, 95(1):206–226.

Cuevas, A., Febrero, M., and Fraiman, R. (2007). Robust estimation and classification for functional data via projection-based depth notions. *Computational Statistics*, 22(3):481–496.

Dai, X., Müller, H.-G., and Yao, F. (2017). Optimal Bayes classifiers for functional data and density ratios. *Biometrika*, 104(3):545–560.

Dailey, D. J. (1993). *Improved error detection for inductive loop sensors*. Washington State Department of Transportation.

Delaigle, A. and Hall, P. (2010). Defining probability density for a distribution of random functions. *The Annals of Statistics*, 38(2):1171–1193.

Delaigle, A. and Hall, P. (2012). Achieving near perfect classification for functional data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):267–286.

Durrett, R. (2010). *Probability: theory and examples*. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, Cambridge ; New York, 4th ed edition. OCLC: ocn607573997.

Epifanio, I. (2008). Shape descriptors for classification of functional data. *Technometrics*, 50(3):284–294.

Fan, J. and Gijbels, I. (1996). *Local polynomial modelling and its applications: Monographs on statistics and applied probability 66*. CRC Press.

Febrero, M., Galeano, P., and González-Manteiga, W. (2007). A functional analysis of NOx levels: location and scale estimation and outlier detection. *Computational Statistics*, 22(3):411–427.

Ferraty, F. and Vieu, P. (2003). Curves discrimination: a nonparametric functional approach. *Computational Statistics & Data Analysis*, 44(1-2):161–173.

Ferraty, F. and Vieu, P. (2006). *Nonparametric functional data analysis: theory and practice*. Springer series in statistics. Springer, New York, NY. OCLC: 255028741.

Fitzmaurice, G. M., Laird, N. M., and Ware, J. H. (2012). *Applied longitudinal analysis*. Wiley, 2 edition edition.

Flury, B. N. (1984). Common principal components in K groups. *Journal of the American Statistical Association*, 79(388):892.

Galeano, P., Joseph, E., and Lillo, R. E. (2015). The Mahalanobis distance for functional data with applications to classification. *Technometrics*, 57(2):281–291.

Graham, J. W., Cumsille, C. P. E., and ElekFisk, E. (2003). Methods for handling missing data. *Handbook of Psychology*.

Grandhi, A. (2002). Content-based image retrieval plant species identification. Master's thesis, Oregon State University.

Haiminen, N. and Gionis, A. (2004). Unimodal segmentation of sequences. In *Fourth IEEE International Conference on Data Mining, 2004. ICDM '04*, pages 106–113.

Hall, P., Poskitt, D. S., and Presnell, B. (2001). A functional data-analytic approach to signal discrimination. *Technometrics*, 43(1):1–9.

Happ, C. and Greven, S. (2017). Multivariate functional principal component analysis for data observed on different (dimensional) domains. *Journal of the American Statistical Association*, pages 1–11.

Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., and Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science (New York, N.Y.)*, 293(5539):2425–2430.

He, G., Mller, H. G., and Wang, J. L. (2000). Extending correlation and regression from multivariate to functional data. *Asymptotics in statistics and probability*, pages 197–210.

Henderson, J. M., Larson, C. L., and Zhu, D. C. (2007). Cortical activation to indoor versus outdoor scenes: an fMRI study. *Experimental Brain Research*, 179(1):75–84.

Himberg, J., Korpiaho, K., Mannila, H., Tikanmäki, J., and Toivonen, H. T. T. (2001). Time series segmentation for context recognition in mobile devices. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 203–210.

Hörmann, S. and Kokoszka, P. (2010). Weakly dependent functional data. *The Annals of Statistics*, 38(3):1845–1884.

Hyndman, R. J. and Shang, H. L. (2010). Rainbow plots, bagplots, and boxplots for functional data. *Journal of Computational and Graphical Statistics*, 19(1):29–45.

Hyndman, R. J. and Ullah, M. S. (2007). Robust forecasting of mortality and fertility rates: A functional data approach. *Computational Statistics & Data Analysis*, 51(10):4942–4956.

Jacques, J. and Preda, C. (2014). Model-based clustering for multivariate functional data. *Computational Statistics & Data Analysis*, 71:92–106.

James, G. M. and Hastie, T. J. (2001). Functional linear discriminant analysis for irregularly sampled curves. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):533–550.

James, G. M., Hastie, T. J., and Sugar, C. A. (2000). Principal component models for sparse functional data. *Biometrika*, 87(3):587–602.

Killick, R. and Eckley, I. A. (2014). changepoint: An R package for changepoint analysis. *Journal of Statisitcal Software*, 58(3):1–19.

Killick, R., Fearnhead, P., and Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598.

King, G., Honaker, J., Joseph, A., and Scheve, K. (2001). Analyzing incomplete political science data: An alternative algorithm for multiple imputation. *American Political Science Review*, 95(1):49–69.

Laird, N. M. (1988). Missing data in longitudinal studies. *Statistics in medicine*, 7(1-2):305–315.

Langley, P., Iba, and, W., and Thompson, K. (1992). An analysis of Bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI'92, pages 223–228, San Jose, California. AAAI Press.

Lee, D.-J., Archibald, J. K., Schoenberger, R. B., Dennis, A. W., and Shiozawa, D. K. (2008). Contour matching for fish species recognition and migration monitoring. In *Applications of Computational Intelligence in Biology*, pages 183–207. Springer.

Leng, X. and Müller, H.-G. (2006). Classification using functional data analysis for temporal gene expression data. *Bioinformatics*, 22(1):68–76.

Li Qu, Jianming Hu, Li Li, and Yi Zhang (2009). PPCA-based missing data imputation for traffic flow volume: A systematical approach. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):512–522.

Little, R. J. A. (1995). Modeling the drop-out mechanism in repeated-measures studies. *Journal of the American Statistical Association*, 90(431):1112–1121.

Little, R. J. A. and Rubin, D. B. (2002). *Statistical analysis with missing data*. John Wiley & Sons.

Martin-Barragan, B., Lillo, R., and Romo, J. (2014). Interpretable support vector machines for functional data. *European Journal of Operational Research*, 232(1):146–155.

Mercer, J. (1909). XVI. Functions of positive and negative type, and their connection the theory of integral equations. *Phil. Trans. R. Soc. Lond. A*, 209(441-458):415–446.

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

Müller, H.-G. (2005). Functional modelling and classification of longitudinal data. *Scandinavian Journal of Statistics*, 32(2):223–240.

Nakai, M. and Ke, W. (2011). Review of the methods for handling missing data in longitudinal data analysis. *International Journal of Mathematical Analysis*, 5(1):1–13.

Newey, W. K. and West, K. D. (1994). Automatic lag selection in covariance matrix estimation. *The Review of Economic Studies*, 61(4):631–653.

Nihan, N. L. (1997). Aid to determining freeway metering rates and detecting loop errors. *Journal of Transportation Engineering*, 123(6):454–458.

Oba, S., Sato, M.-a., Takemasa, I., Monden, M., Matsubara, K.-i., and Ishii, S. (2003). A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16):2088–2096.

Palpanas, T., Vlachos, M., Keogh, E., Gunopulos, D., and Truppel, W. (2004). Online amnesic approximation of streaming time series. In *Proceedings. 20th International Conference on Data Engineering*, pages 339–349.

Paul, D. and Peng, J. (2009). Consistency of restricted maximum likelihood estimators of principal components. *The Annals of Statistics*, 37(3):1229–1271.

Perron, P. (1991). *A test for changes in a polynomial trend function for a dynamic time series*. Econometric Research Program, Princeton University.

Preda, C. and Saporta, G. (2005). PLS regression on a stochastic process. *Computational Statistics & Data Analysis*, 48(1):149–158.

Preda, C., Saporta, G., and Lévéder, C. (2007). PLS classification of functional data. *Computational Statistics*, 22(2):223–235.

Ramsay, J. O. (1982). When the data are functions. *Psychometrika*, 47(4):379–396.

Ramsay, J. O. and Dalzell, C. J. (1991). Some tools for functional data analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):539–572.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional data analysis*. Springer series in statistics. Springer, New York, 2nd ed edition.

Rice, J. A. (2004). Functional and longitudinal data analysis: Perspectives on smoothing. *Statistica Sinica*, 14(3):631–647.

Rice, J. A. and Silverman, B. W. (1991). Estimating the mean and covariance structure nonparametrically when the data are curves. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(1):233–243.

Rice, J. A. and Wu, C. O. (2001). Nonparametric mixed effects models for unequally sampled noisy curves. *Biometrics*, 57(1):253–259.

Rossi, F. and Villa, N. (2006). Support vector machine for functional data classification. *Neurocomputing*, 69(7):730–742.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581.

Rubin, D. B. (2004). *Multiple imputation for nonresponse in surveys*. John Wiley & Sons.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. CRC Press.

Schafer, J. L. (1999). Multiple imputation: A primer. *Statistical Methods in Medical Research*, 8(1):3–15.

Schafer, J. L. and Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, 7(2):147–177.

Sguera, C., Galeano, P., and Lillo, R. (2014). Spatial depth-based classification for functional data. *TEST*, 23(4):725–750.

Shatkay, H. and Zdonik, S. B. (1996). Approximate queries and representations for large data sequences. In *Proceedings of the Twelfth International Conference on Data Engineering*, pages 536–545.

Shin, H. (2008). An extension of Fisher's discriminant analysis for stochastic processes. *Journal of Multivariate Analysis*, 99(6):1191–1216.

Silverman, B. W. (1996). Smoothed functional principal components analysis by choice of norm. *The Annals of Statistics*, 24(1):1–24.

Smith, B. and Conklin, J. (2002). Use of local lane distribution patterns to estimate missing data values from traffic monitoring systems. *Transportation Research Record: Journal of the Transportation Research Board*, 1811:50–56.

Staniswalis, J. G. and Lee, J. J. (1998). Nonparametric regression analysis of longitudinal data. *Journal of the American Statistical Association*, 93(444):1403–1418.

Sun, Y. and Genton, M. G. (2011). Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2):316–334.

Terzi, E. (2006). *Problems and algorithms for sequence segmentations*. PhD thesis, University of Helsinki, Helsinki. OCLC: 500354767.

Terzi, E. and Tsaparas, P. (2006). Efficient algorithms for sequence segmentation. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, Proceedings, pages 316–327. Society for Industrial and Applied Mathematics.

Tipping, M. E. and Bishop, C. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 21/3.

Tuddenham, R. D. and Snyder, M. M. (1954). Physical growth of California boys and girls from birth to eighteen years. *Publications in Child Development. University of California, Berkeley*, 1(2):183–364.

Vaart, A. W. v. d. (1998). *Asymptotic statistics*. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, Cambridge, UK ; New York, NY, USA.

Vogelsang, T. J. (1999). Sources of nonmonotonic power when testing for a shift in mean of a dynamic time series. *Journal of Econometrics*, 88(2):283–299.

Wang, J.-L., Chiou, J.-M., and Müller, H.-G. (2016). Functional data analysis. *Annual Review of Statistics and Its Application*, 3(1):257–295.

Wang, X., Ray, S., and Mallick, B. K. (2007). Bayesian curve classification using wavelets. *Journal of the American Statistical Association*, 102(479):962–973.

Wu, Y. and Liu, Y. (2013). Functional robust support vector machines for sparse and irregular longitudinal data. *Journal of Computational and Graphical Statistics*, 22(2):379–395.

Yao, F. and Lee, T. C. M. (2006). Penalized spline models for functional principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):3–25.

Yao, F., Müller, H.-G., and Wang, J.-L. (2005). Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, 100(470):577–590.

Zhang, L., Marron, J. S., Shen, H., and Zhu, Z. (2007). Singular value decomposition and its visualization. *Journal of Computational and Graphical Statistics*, 16(4):833–854.

Zhang, X., Shao, X., Hayhoe, K., and Wuebbles, D. J. (2011). Testing the structural stability of temporally dependent functional observations and application to climate projections. *Electronic Journal of Statistics*, 5:1765–1796.

Zhong, M., Sharma, S., and Lingras, P. (2004). Genetically designed models for accurate imputation of missing traffic counts. *Transportation Research Record: Journal of the Transportation Research Board*, 1879:71–79.