Unsupervised Learning on Functional Data with an Application to the Analysis of U.S. Temperature Prediction Accuracy

by

Chuyuan Lin

B.Sc., Simon Fraser University, 2017

Project Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY
Spring 2019

Approval

Name: **Chuyuan Lin**

Degree: **Master of Science (Statistics)**

Title: **Unsupervised Learning on Functional Data with an Application to the Analysis of U.S. Temperature Prediction Accuracy**

Examining Committee: **Chair:** Dr. Jinko Graham
Professor
Department of Statistics and Actuarial Science

Dr. Jiguo Cao
Senior Supervisor
Associate Professor
Department of Statistics and Actuarial Science

Dr. Leonid Chindelevitch
Supervisor
Assistant Professor
Department of Computing Science

Dr. Lloyd Elliott
Internal Examiner
Assistant Professor
Department of Statistics and Actuarial Science

Date Defended: **February 7, 2019**

Abstract

Unsupervised learning techniques are widely applied in exploratory analysis as the motivation of further analysis. In functional data analysis, two typical topics of unsupervised learning are functional principal component analysis and functional data clustering analysis. In this study, besides reviewing the developed unsupervised learning techniques, we extend unsupervised random forest clustering method to functional data and detect its shortages and strength through comparisons with other clustering methods in simulation studies. Finally, both proposed method and developed unsupervised learning techniques are conducted on a real data application: the analysis of the accuracy of the U.S. temperature prediction from 2014 to 2017.

Keywords: Unsupervised Learning, Functional Data Analysis, Unsupervised Random Forest, Functional Principal Component Analysis, Hierarchical Clustering, K-means Clustering, Gaussian Mixture Model-based Clustering, Weather Forecast Exploratory Analysis

Dedication

To my beloved parents Mr. Shangpeng Lin and Mrs. Wen Qiu, my family, my friends, and my dogs Huahua and Xiaobai.

Acknowledgements

I want to convey my sincere gratitude to all the people who encourage and support me in my graduate study. In particular, I want to thank my senior supervisor Dr. Jiguo Cao for his vast patience, encouragement and assistance in the process of research and thesis writing.

Besides, I want to express my thanks to my examining committees, Dr Jinko Graham, Dr. Leonid Chindelevitch and Dr. Lloyd Elliott. Thank you for spending you valuable time in attending my defense. Also, I would like to acknowledge all the staff and faculty in the department of statistics and actuarial science, for their great help and kindness.

In addition, I would like to thank all my friends and fellow students, for always supporting and cheering me up when I was depressed or anxious. Especially, I want to acknowledge Zhiyang Zhou for proofreading my thesis. Moreover, I would like to express my appreciation to Ying Yu, Yifan Wu and Dr. Peijun Sang, who brought motivations and offered great help to this thesis. I would also like to thanks my lovely friends and academic sisters and brothers, Dr. Yunlong Nie, Tianyu Guan, Shijia Wang, Jingxue Feng, Anqi Chen, Haoyao Ruan, Jiarui Zhang, Shufei Ge, Yuping Yang, Qi Wen, Mengyang Li, Yan Lin, Trevor Thomson, Jingdan Li, Xinmiao Wang and Dan Lin. Thank you for your companion during this memorable master program.

Last but not least, I am very grateful to my parents and my family, for their invaluable love and support. Thank you for your understanding, encourage and the funny videos from our lovely dogs, which comforted me a lot during my hardest time.

Table of Contents

List of Tables

List of Figures

Chapter 1

Background

1.1 Literature Review

1.1.1 Overview of unsupervised learning

In statistical analysis and machine learning studies, two main categories of methodologies are classified as supervised (with outcome or response) and unsupervised (without outcome or response)[33]. Supervised studies are studies concerning the prediction of one or more response variables $Y_1, Y_2, ..., Y_m$ through predictor variables $X_1, X_2, ..., X_p$ [21]. With some collected data $(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$, a model can be trained to estimate the response variable Y of interest using part of the data as a training set. Then, with the remaining data as a test set, the precision of the model can be evaluated through some loss function $L(y, \hat{y})$, where y is the real value of the response variable, and the $\hat{y}$ is the estimated value from the model. However, in unsupervised studies, one or more variables may be of interest in the data, without a response variable Y, or without a specific idea about what is the response variable [21]. In this case, unsupervised studies studies are usually conducted as exploratory analysis, i.e. with N observations and p variables $(x_1, x_2, ..., x_N)$, an unsupervised study involves detecting the properties or patterns of the joint distribution for the p variables in the data. Thus, without a response variable, supervised models and loss function for model evaluation cannot be specified, and the unsupervised methodologies are developed for further exploratory analysis.

Principal component analysis

The focus of this thesis is on the unsupervised studies with real application. To be more specific, principal component analysis (PCA) and clustering analysis are two main research fields with a wide application to data exploratory analysis [18]. In multivariate data, PCA is a method used for reducing the dimension as well as keeping most of the information or variation in the original data [27]. It is popularly applied in biology [36] and toxicology [26] because the dimension as well as the number of variables in the data are large, sometimes even larger than the number of observations. In biology, gene expression data can be preprocessed by PCA for dimension reduction before further analysis [36]. We will discuss this example in the section of review of clustering analysis. In toxicology, PCA can be used in the estimation of the sources and types of some heavy metal contamination in researches about water or land pollution. For instance, in research for toxic substance source detection, Poland, Loska and Wiechula 2003 [26] discovered suspicious locations of toxic substance sources based on the summary and visualization of PCA result.

The main idea of PCA is to reduce high or medium correlation among variables, through constructing a new coordinate system with a set of transformed linear uncorrelated variables, which are the so-called principal components (PCs). PCs are ordered by how much variation of the original data can be explained, and each observation at each PC has a score; therefore, when the number of PCs is less than the number of variables, all the observations can be represented in a lower dimensional space through their PC scores. Suppose N observations are collected from the distribution of a multivariate variable $\mathbf{X}$, $\{\mathbf{x_1}, ..., \mathbf{x_N}\}$, the classic analysis obtains these scores from the covariance matrix of $\mathbf{X}$, $\Sigma = E[(\mathbf{X} - E(\mathbf{X}))(\mathbf{X} - E(\mathbf{X}))^T]$, which is usually estimated using the sample covariance matrix $S = \frac{1}{N}\sum(\mathbf{x_i} - \bar{\mathbf{x}})(\mathbf{x_i} - \bar{\mathbf{x}})^T$. The procedure to achieve the PC scores can be decomposed into the following steps:

1. Obtain the eigenvalues and the corresponding eigenvectors of V.

2. Order the eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$, and let $\boldsymbol{\alpha_k}$ be the eigenvector of V_k corresponding to the k^{th} largest eigenvalues λ_k.

3. For an original observation $\mathbf{x_i} = (x_{i1}, ..., x_{ip})^T$, its k^{th} PC score $s_{ik} = (\mathbf{x_i} - \bar{\mathbf{x}})^T\boldsymbol{\alpha_k}$, and the observation $\mathbf{x_i}$ can be reformed as $\mathbf{x_i} = \bar{\mathbf{x}} + \sum_{k=1}^{p} s_{ik}\boldsymbol{\alpha_k}$.

The sum of eigenvalues $\lambda_1 + \lambda_2 + ... + \lambda_p = \sum_{i=1}^{p} \mathrm{Var}(\mathbf{x_i})$ is the total variability of the data. For an eigenvalue λ_k, its proportion of variation explained is $\frac{\lambda_k}{\sum_{k=1}^{p}\lambda_k}$. The eigenvectors $\boldsymbol{\alpha_1}, ..., \boldsymbol{\alpha_p}$ are the loadings of the k^{th} PC, and are orthogonal with each other.

Clustering analysis

Clustering analysis, which is usually applied after PCA analysis, is another category of techniques used in unsupervised learning. Its objective is to identify observed data into homogeneous groups as well as clusters without knowing their labels in advance. As a typical exploratory analysis, clustering analysis has been involved in large amount of researches from biology, meteorology to climatology. In biology, many researchers investigate potential gene patterns and functions through classical clustering methods such as K-means or K-centroid clustering, or their modified versions [36][6]. Additionally, in both meteorology and climatology, clustering algorithms can be performed on daily weather forecast data or historical measured climate records, detecting possible regions with similar weather patterns, or tracking change of climate within a wide area partitioned into smaller subregions[11][35].

There are two types of clustering algorithms, hierarchical clustering and non-hierarchical clustering[18]. The main difference between them is that in hierarchical clustering, once an observation is assigned to a cluster, it cannot move to any other clusters when the assigned group is merged during iterations; however, in non-hierarchical clustering, the observations may be assigned to different clusters before the final clustering decision. Hierarchical clustering is a type of classical clustering method which determines the members of the clusters through comparing the difference or distance between observations [21]. Under this setting, with N observation, an $N \times N$ distance matrix is formed with distance measurements of all the pairs of observations. A clustering algorithm is then applied to the distance matrix so that the observations are sequentially clustered together based on the order of their distance measurements. On the other hand, non-hierarchical clustering algorithms contains two sub-types of clustering algorithms, partitioning clustering and model-based clustering. The most famous partitioning clustering algorithm is the K-means algorithm [20]. With a fixed number of clusters K, the K-means algorithm aims to find K clusters such that the variances within each cluster are minimized and the variance between clusters is maximized. In each iteration of the algorithm, the clustering result is re-evaluated for each observation until all the observations are in the clusters with the mean they are closest to. For model-based clustering, mixture distributions are fitted to the observations, and the estimation of the parameters in the probability density functions (PDFs) and the clustering procedure is obtained through maximizing the likelihood function.

1.1.2 Unsupervised learning on time series and functional data

Time series data is a type of high-dimensional multivariate data, because the dimension of it can be considered to be the length of the observed time or the number of the observed time points. The number of time points can be very large with long-term observation or high frequency records such as stocking data, daily temperature records over years and long-run machine monitoring[30]. In clustering analysis of unsupervised studies, two procedures are usually used in the time series data. One is conducting the clustering methodologies on the raw (or smoothed) discrete time series data directly, and the other is conducting the clustering methodologies after converting the time series data into functional data [22]. In 2014, Jacques and Preda graphed the following segmentation to summarize clustering methods for time series or functional data[22].

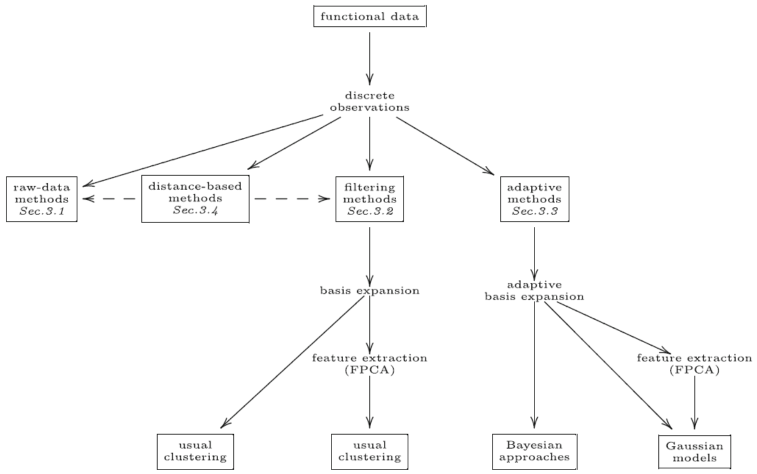

Figure 1.1: Categorization Segmentation of Clustering Methods for Functional Data

Clustering on discrete time series data is classified into raw data methods and distance-based methods in the segmentation in Figure 1.1. The classical idea is to consider each observation as a multivariate case such that each time point is a variable, and apply multivariate clustering methods in section 1.1.1 [22]. According to the research by Aghabozorgi et al.(2015)[4], two of the most common approaches for discrete time series data clustering are hierarchical clustering and partitional clustering. Hierarchical clustering in this case uses dissimilarity to the time series data; and partitional clustering is an extension of K-means clustering to time series data. Clustering in time series data is more complicated and challenging due to the high dimensionality of the input space. Two main difficulties are clustering accuracy and time efficiency[30]. For clustering accuracy, as the noise of the data has sensitive impact on the time series data clustering, some statisticians suggest smoothing the data first before clustering so as to reduce the noise and capture the main pattern through time [4]. In addition, when clustering aims at identifying the shapes of time series data, statisticians introduce a distance called Dynamic Time Warping (DTW) Distance, which measures the dissimilarity between the shapes of two time series observations. This distance measure can even handle time series comparison with different time length [7]. However, another challenge of discrete time series data is not solved: time efficiency. In the DTW distance case, with time series data x with n time points and y with m time points, their DTW distance time complexity is $O(nm)$. To improve algorithm efficiency, dimension reduction is recommended and concerned in the second clustering procedure related to time series clustering, which is described below.

The second clustering procedure treats time series data as functional data during clustering, and it can solve the expensive computation problem and reduce the noise at the same time. In functional data, each observation x is defined as a function of time t, i.e., $x(t)$; in other words, functional data are considered as a set of curves with infinite dimensions [16]. Data smoothing is the first step of all the functional data clustering methods, and is done by fitting each set of time series data to a linear combination of some basis functions, such as spline functions or polynomials [28]. Clustering procedures can then be conducted on these smoothed data using the methods for functional data clustering, which are referred to as filtering methods, adaptive methods and part of distance-based methods in Figure 1.1. The descriptions and examples of the above clustering methods are listed in the following.

1. Distance-based methods. Similar to the multivariate situation, distance-based methods in the functional case conduct clustering based on the distance between the functional objects[22]. Given two functional observations, $x_1(t)$ and $x_2(t)$, the distance or dissimilarity is usually defined as

   $$d_l(x_1(t), x_2(t)) = (\int_T (x_1^{(l)}(t) - x_2^{(l)}(t))^2)^{1/2}, \tag{1.1}$$

   where $x^{(l)}(t)$ is the l-th order derivative of $x(t)$. Using the distance d_0, a robust K-means algorithm has been developed by Cuesta-Albertos and Fraiman [19]. In addition, Ferraty and Vieu 2006[16] proposed the extension of hierarchical clustering to functional data based on the d_0 (as well as L_2) and d_2 distance. Such methods are approximately the same as distance-based clustering on smoothed discrete time series data.

2. Filtering methods. Filtering methods are methods that cluster the observed objects using information after the dimension reduction of the functional data[23]. In filtering methods, instead of directly clustering on fitted functions after smoothing the time series data, some feature information with lower dimensions of the data are extracted to represent the original data and reduce the dimensions for further clustering. The first approach is to use coefficients of the basis functions to represent the data. In 2003, Abraham et al. selected B-splines as basis functions and applied K-means algorithm on the coefficients of B-splines. They also provide a proof of consistency for this method. Similarly, Rossi et al [29] developed a self-organised map for curves clustering through their B-spline coefficients.

   The second approach is using functional principle component scores (FPC scores). This approach is completed through another popular dimension reduction technique, functional principal component analysis (FPCA). FPCA is an extension of principal component analysis (PCA) to functional data. Similar to PCA, FPCA detects the directions that explain the most variation of the data. Those directions amount to eigenfunctions. Then, FPC scores of each observation can be obtained from the eigenfunctions. Besides representing the data through the B-spline coefficients, FPC scores are also recommended for dimension reduction. In 2011, Adelfio, Giada and et al. selected the scores of the first few FPCs for further K-means clustering, where the number of FPCs is based on a desired percentage of the explained variance [3].

3. Adaptive methods. Adaptive methods are model-based clustering methods for functional data. In fact, most adaptive methods model cluster-specific probability distributions using the coefficients of the basis functions or FPC scores. From the obtained distributions of all clusters, the probability of belonging to a specific cluster can be estimated for every observed object. Two most recent adaptive clustering techniques are FunFEM [9] and FunHDDC[31], implemented in the package `FunFEM` and `FunHDDC` respectively in the R language. The FunFEM method is developed for analyzing the bike sharing systems (BSSs) in Europe and exploring patterns of the used rate of bikes, while FunHDDC is used to deal with the multivariate functional data related to pollution in French cities[31]. Both FPCA and functional latent mixture models are applied in both FunFEM and FunHDDC; but the main different between them is that FunFEM accomplishes clustering in a discriminative functional subspace obtained from modified FPCA with a common dimension d, whereas FunHDDC uses FPCA directly to fit the data in cluster-specific subspaces with different dimensions. The models are estimated by EM algorithms in both cases. BIC or integrated completed likelihood (ICL) are commonly used in determining the number of clusters and FPCs.

1.2 Motivation

1.2.1 Conducting modern unsupervised machine learning methodologies on functional data

Besides dealing with the unsupervised analysis through the methodologies reviewed in the previous sections, some decision tree-based unsupervised learning methods were developed in early 2000s for dealing with similar analyses. A typical example is the unsupervised random forest (URF), which has been developed and applied to clustering of DNA microarray data [10]. The main contribution of unsupervised random forest(URF) is measuring dissimilarity among the observations through a proximity matrix[37].

URF is developed from the regular supervised random forest for multivariate data used in classification. In 2001, the random forest model was published by Breiman as an ensemble learning method of many uncorrelated and weak individual regression or classification decision trees [18]. A trained random forest for classification can be used to predict the class of a new observation. Each individual tree in the trained random forest votes one class of each input, and the trained random forest achieves the final prediction as the class voted by the most number of trees. Later, through decision trees in classification random forest, Breiman obtained a proximity matrix containing all the similarity measurements between each pair of observations [18]. In addition, Breriman and Cutler suggested that the supervised method should be able to distinguish synthetic data from the original data when generating synthetic data based on the marginal distribution of each variable in the original data and mixing them into the original data[10]. Based on this proposal, unsupervised random forest was put forward with the belief that, with the synthetic data, the unsupervised mode of the random forest turns into a supervised mode. Then, by distinguishing the original and synthetic data, the similarity between the original observations and latent patterns in the original data would be found through the proximity matrix.

Feature information of functional data, such as coefficients of the splines and FPC scores, can be considered as an regularization of the smoothed data to multivariate case; therefore, the common procedure of the functional data clustering is first replacing the functional or time series observations by the feature information, which transform the problem into multivariate clustering case. Secondly, the observations are partitioned into several groups through clustering methods for multivariate data, including hierarchical clustering, partitioning clustering and model-based clustering. Using the idea of feature extraction, some modern unsupervised learning algorithm for multivariate data can also extend to functional data.

Chapter 2

Methodology

2.1 Smoothing Splines and Functional Principal Component Analysis (FPCA)

2.1.1 Non-parametric regression on time series data using B-splines

To reduce the noise and capture the main pattern of time series data, we borrow an idea from linear regression, and fit the data in each state to a smoothed curve as a linear combination of several spline basis functions. In this paper, we used the Schoenbergs B-splines as our basis function family. Consider the time $t \in [a, b]$ and M distinct interior points $\xi_1, \xi_2, ..., \xi_M$ that partition $[a, b]$ into $M + 1$ segments as $a\ (\xi_0) < \xi_1 < \xi_2 ... < \xi_M < b\ (\xi_{M+1})$; then B-spline functions with degree d will be fitted on each interval $[\xi_i, \xi_{i+1}]$ with $d - 1$ continuous derivatives on the open interval (a, b), where $i = 0, 1, ..., M$. With degree d and M interior points $\xi_1, \xi_2, ..., \xi_M$, $M + d + 1$ B-spline basis functions $(B_1(t), B_2(t), ..., B_{M+d+1}(t))$ form the linear space, and an observed curve $x(t)$ can be approximated as a linear combination of the basis functions as

$$s(t) = s(t, \beta) = \sum_{l=1}^{M+d+1} \beta_l B_l(t), \tag{2.1}$$

where $\beta = (\beta_1, \beta_2, ...\beta_{M+d+1})'$ is the coefficients of the corresponding basis function[14]. Similar to the coefficient estimation in linear regression, to estimate β, we firstly transfer the observed time points $t_1, t_2, ..., t_n$ to an $n \times (M + d + 1)$ matrix $\boldsymbol{B}$ with row vector $\boldsymbol{B_i} = (B_1(t_i), ..., B_{M+d+1}(t_i))$. Under the assumption that the $\boldsymbol{B'B}$ is non-singular, β is estimated using least squared error as

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (y_i - s(t_i, \beta))^2 = (\boldsymbol{B'B})^{-1} \boldsymbol{B} y, \tag{2.2}$$

where $\boldsymbol{y} = (y_1, y_2, ..., y_n)$ is the observation of the response variable at time $(t_1, t_2, ..., t_n)$.

2.1.2 Functional principal component analysis

Functional principal component analysis (FPCA) is an extension of principal component analysis to the functional data $x(t)$, where t is a continuous variable[27]. Given a set of functional data, suppose N smoothing curves $\{x_i(t)|i = 1...N, t \in [a, b]\}$, we sample i.i.d from the distribution of a random curve variable $X(t)$ with covariance function $V(s, t) = E[(X(s) - E[X(s)])(X(t) - E[X(t)])]$. Then, the first step of FPCA is to estimate the covariance function as

$$\hat{V}(s, t) = \frac{1}{N-1} \sum_i [x_i(s) - \bar{x}(s)][x_i(t) - \bar{x}(t)], \tag{2.3}$$

where s and t share the same domain $[a, b]$. Using the Karhunen-Loeve decomposition[27], the $v(s, t)$ can be decomposed as

$$V(s, t) = \sum_{j=1}^{\infty} d_j \xi_j(s) \xi_j(t), \tag{2.4}$$

where $\xi_j(t)$ are the eigenfunctions. d_j is the eigenvalue of $\xi_j(t)$usually done with the restricted condition $\int \xi^2(t)dt = 1$. Similar to PCA for multivariate data, d_j and $\xi_j(t)$ satisfy the equation

$$\int v(s, t)\xi_j(t)dt = d_j \xi_j(s) \tag{2.5}$$

and d_j is proportional to the percentage of variation that $\xi_j(t)$ explains. Let $E[X(t)] = \mu(t)$, the j-th PC score of functional data $X_i(t)$ can be calculated as

$$\rho_{ij} = \int \xi_j(t)[X_i(t) - \mu(t)]. \tag{2.6}$$

The FPCA algorithm has been implemented in the function `pca.fd()` in the `fda` package implemented in R programming language and available on CRAN[27]. In FPCA algorithm, eigenfunctions are considered as the linear combinations of the set of basis function $\phi_1(t), ..., \phi_m(t)$, i.e. $\xi(t) = \sum_{l=1}^{m} b_l \phi_l(t)$ [27]. Let $\boldsymbol{\phi} = (\phi_1(t), ..., \phi_m(t))$ be the vector form of basis functions, and let each observations be decomposed as $x_i(t) = \sum_{l=1}^{m} c_{il}\phi_l(t), i = 1, ..., N$, then N observed curves can be expressed as

$$\boldsymbol{x} = \boldsymbol{C}\boldsymbol{\phi}, \tag{2.7}$$

and $\hat{V}(s, t)$ can be expressed as

$$\hat{V}(s, t) = \frac{1}{N} \boldsymbol{\phi}(s)^T C^T C \boldsymbol{\phi}(t). \tag{2.8}$$

Equation 3.5 can then be rewritten in matrix form as

$$\frac{1}{N}\boldsymbol{\phi}(\boldsymbol{s})^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{W}\boldsymbol{b} = d\boldsymbol{\phi}(\boldsymbol{s})^T\boldsymbol{b}, \tag{2.9}$$

where $\boldsymbol{W} = \int \boldsymbol{\phi}^{\boldsymbol{T}}\boldsymbol{\phi}$. Using linear algebra, d and $\boldsymbol{b}$ can be formed by solving eigen-equation 3.9, where the ds are the estimated eigenvalues, and the $\boldsymbol{b}$s are the estimated coefficients of the basis function with corresponding eigenfunctions. The estimated eigenfunctions are then reordered following the size of their eigenvalues from largest to the smallest. The first P PCs are selected with largest eigenvalues that explain most of the variation (i.e. >90%) in the curves. Finally, each $x_i(t)$ can be rewritten and approximated as

$$\begin{aligned} x_i(t) &= \mu(t) + \sum_{j=1}^{\infty} \rho_{ij}\xi_j(t) \qquad (2.10) \\ &\approx \bar{x}(t) + \sum_{j=1}^{P} \hat{\rho}_{ij}\hat{\xi}_j(t), \qquad (2.11) \end{aligned}$$

where $\hat{\rho}_{ij}$ is the estimated PC score of the j-th PC achieved from equation (3.6) using estimated eigenfunction $\hat{\xi}_j(t)$ and sample mean $\bar{x}(t)$. In other word, FPCA provides us with a group of basis functions $\{\bar{x}(t), \xi_1\hat{(t)}, ..., \xi_P\hat{(t)}\}$, and reforms the functional data into another linear combination of the new basis functions, where the coefficient of $\bar{x}(t)$ is always 1, and the coefficient of the $\hat{\xi}_p(t)$ is the estimated score of the j-th PC of the corresponding curve.

2.2 Proposed Method: Unsupervised Random Forest Clustering On Feature Information

Feature information of the functional data can be understood as the different feature values derived from the data, such as B-spline coefficients or FPC scores. With feature information from time series data, we introduce another filtering method through combining classical hierarchical clustering methods and modern machine learning methods, namely, the unsupervised random forest algorithm. The framework of this clustering method is:

1. Obtain feature information;

2. Construct the unsupervised random forest using the original feature information and synthetic dataset based on the marginal distribution of each "variable" in the feature information;

3. Calculate the dissimilarity matrix of the time series data from the proximity matrix of the constructed unsupervised random forest;

4. Conduct hierarchical clustering on the dissimilarity matrix.

2.2.1 Algorithm review: random forests for classification

In 2001, random forests were formally published by Breiman as an ensemble learning method of many uncorrelated and weak individual regression or classification decision trees[18]. The algorithm used to obtain a random forest predictor with B classification decision trees can be decomposed into the following steps

1. Split the data into a training set and a test set, and then train B classification decision trees. For $b = 1, ..., B$

   (a) Draw a sample of size n from the training set using bootstrapping.

   (b) Build a tree T_b on the bootstrapped sample recursively by repeating the following steps on every terminal node of the tree, until reaching the minimum node size n_{min}:

      i. Randomly select m variables from all p variables,

      ii. Pick split points with the least misclassification error among the m variables, and split that node into two children nodes.

2. Return all B classification decision trees as the trained random forest

The random forest for classification trained from the above algorithm can be used to predict the class of a new observation. Each individual tree in the trained random forest votes one class for each observation, and the trained random forest yields a final prediction from the class voted by the most number of trees. Moreover, a proximity matrix can be measured from decision trees in classification random forest, containing the similarity measurement between every two observations. Through the proximity matrix, a dissimilarity matrix can be calculated from the measured distance or difference among observations, which is one of the vital input in many clustering methods, such as hierarchical clustering or partitioning around medoids (PAM).

Proximity matrices and dissimilarity matrices from random forests

With a classification-type random forest trained from N observations, all the observations reach the leaves of some terminal nodes in every classification tree. Therefore, for the i-th and j-th observations with $i, j = 1, ..., N$ and $i \neq j$, the proximity (or similarity) is defined as the fraction of the trees that the i-th and j-th observations are finally split to the same terminal child node as

$$P_{ij} = \frac{1}{B} \sum_{b=1}^{B} I(i^{th} \text{ and } j^{th} \text{ observations in the same final child node} | T_b) \tag{2.12}$$

After measuring the proximity between all of the observations, an $N \times N$ proximity matrix $P = (P_{ij})_{N \times N}$ is formed. According to the definition of P_{ij}, the proximity matrix is a symmetric matrix where all entries are between 0 and 1. To be more specific, the values on the diagonal of the matrix are all 1, and the values off the diagonal are in 0 to 1.

To further our purposes towards clustering, we define the dissimilarity (or distance) matrix as

$$\begin{aligned} D &= 1_{N \times N} - P \qquad (2.13) \\ &= (1 - P_{ij})_{N \times N}. \qquad (2.14) \end{aligned}$$

The dissimilarity matrix measures the degree of difference between observations, which can be directly input into hierarchical clustering methods or partitioning around medoids (PAM).

2.2.2 Unsupervised random forests: extension of classification random forests

The implement of unsupervised random forest can be considered as an extension of classification random forest under an assumption that when the data contain some correlation between the variables or some distinguished patterns between existing but unknown classes, we should be able to separate the original data from the mixture of its randomly generated version and itself. Under this assumption, the algorithm of unsupervised random forests can be described in the following steps:

1. Generate a synthetic dataset with the same size N as the original dataset such that each predictor variable in the synthetic dataset follows the marginal distribution of the corresponding variables in the original dataset, but the relationship between the variables are removed. Two main methods to generate such a synthetic dataset are:
   (a) For each predictor variable, generate N observations from the corresponding marginal distribution in the original dataset, and
   (b) Permute the order of rows for each predictor in the original dataset.
2. Merge the generated dataset and original dataset together, and label the generated rows as ‘Synthetic’ and the original rows as ‘Original’.
3. Apply the random forest training algorithm for a regular supervised two-class classification on the merged dataset, aimed at identifying whether an entry in the merged dataset is from the original or the generated dataset.
4. Calculate the proximity matrix only for the observations from original dataset, and obtain the dissimilarity matrix from the proximity matrix.

2.2.3 Hierarchical clustering using dissimilarity matrices

Given a dissimilarity matrix for N observations, hierarchical clustering method are clustering methods that can sequentially partition the data into n groups, where $n = 1, ..., N$[21]. The observations partitioning process from 1 to N groups can be visualized as a tree diagram. One of the algorithms used to implement hierarchical clustering is called the agglomerative nesting method[18]. The basic idea of this method is described by the following algorithm:

1. Consider each observation as its own cluster at the beginning
2. Agglomerate stepwise
   (a) Join the two clusters that are closest together into one cluster. In this step, some linkage measure must be proposed to define the intergroup distance as well as the closeness of two clusters based on the dissimilarity matrix $D = (D_{ij})_{N\times N}$:
   - Single-linkage. The distance between cluster C_1 and C_2 is defined as

     $$D_{C_1,C_2} = \min_{i\in C_1, j\in C_2}(D_{ij}) \tag{2.15}$$

     and clusters C_1 and C_2 are merged together if D_{C_1,C_2} is smallest compared to the intergroup distance of all other pairs of clusters.
   - Complete-linkage. The distance between cluster C_1 and C_2 is defined as

     $$D_{C_1,C_2} = \max_{i\in C_1, j\in C_2}(D_{ij}) \tag{2.16}$$

     and C_1 and C_2 are merged together if D_{C_1,C_2} is smallest compared to the intergroup distance of all other pairs of clusters.
   - Average-linkage. The distance between cluster C_1 and C_2 is defined as average distance between all pairs of observations in C_1 or C_2. Suppose C_1 and C_2 contains n_1 and n_2 respectively, then the average-linkage distance between two groups is

     $$D_{C_1,C_2} = \frac{1}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} D_{ij} \tag{2.17}$$

     and C_1 and C_2 are merged together if D_{C_1,C_2} is smallest compared to the intergroup distance of all other pairs of clusters.
   - Ward's Minimum Variance Method. Instead of working on the distance between clusters, Ward's method chooses to merge the two clusters whose sum of the squared distance between observations and the centroids is minimized

after merging. In other word, C_1 and C_2 in C groups are selected to be merged if they can minimize

$$W = \sum_{c=1}^{C-1} \sum_{i=1}^{n_c} ||x_i - \hat{x}_c||^2 \tag{2.18}$$

Lance and Williams (1967)[24] discovered the connection between the dissimilarity matrix and Ward's method, and rewrote formula 2.18 using a sum of dissimilarities to update the dissimilarities after merging. This improves efficiency of the implementation of Ward's method.

(b) Repeat (a) until all the data is grouped into one cluster

The hierarchical clustering method with above four types of linkage are implemented in R function `hclust()`, and users can specify the type of linkage through method argument.

2.2.4 Problems in unsupervised random forest clustering

Similar to the supervised random forest algorithm, two parameters in an unsupervised random forest, the number of randomly selected variables m and the minimum terminal node size n_{min} should be chosen through tuning. As no actual response classification variable exists in the original dataset, the criterion for choosing the tuning parameters changes from minimizing a loss function such as misclassification rate to minimizing the inner-cluster difference. That is, given a cluster number K, we must pick tuning parameters that can produce a minimum average of inner-cluster variance over K clusters.However, tuning parameters are usually computationally expensive. During the real application with only 50 functional data, the partitioning 4 clusters through unsupervised tuning random forest spent about 20 minutes. However, the clustering procedure of other clustering methods, which were filtering methods (K-means clustering on B-spline coefficients or FPC scores) and adaptive methods (funFEM and funHDDC), only used about 5 minutes.

Another crucial problem in clustering analysis is to determine the number of clusters. The general process of selecting the number of clusters is that given a set of cluster number candidate and a trusted criterion such as CH or Cindex[13], the value of the criterion for each number candidate is calculated, and the candidate number with the best performance under the criterion is selected. Due to the computational expensiveness, unsupervised tuning random forest is not recommended on this problem, and the number of the clusters are suggested to be decided using other clustering methods. Further discussion about the study of the number of clusters is discussed in section 2.3.3 and 2.4.3.

2.3 Filtering Methods: K-means Clustering on Feature Information

The following two clustering methods are extensions of K-means clustering to time series data. To summarize, both methods form time series data into a linear combination of k basis functions $\phi_1, ..., \phi_k$, and conduct K-means clustering method to the estimated coefficients of the basis functions.

2.3.1 K-means clustering on B-spline coefficients

The first method clusters curves by applying K-means clustering to the coefficient vectors β of all smoothed curves. The framework of this clustering method is

1. Suppose we are given n group of time series data with n_i observations in the i^{th} group, $i = 1...n$. In i^{th} group for any i, we approximate the observations $\{(y_j, t_j)|j = 1..n_i\}$ with a smooth curve $y_i(t)$ expressed as a linear combination of B-spline.
2. Cluster the data into K groups by applying K-means clustering to the estimated B-spline coefficients $\{\hat{\beta}_i|i = 1...n\}$.

In the step 2, to cluster the estimated coefficients $\{\hat{\beta}_i|i = 1...n\}$ into K groups through K-means clustering, the main procedure is to search for K partitions, $\{C_1, C_2, ..., C_K\}$, with center vectors $\{c_1, c_2, ..., c_k\}$ which minimize

$$\frac{1}{n}\sum_{j=1}^{K}\sum_{\hat{\beta}_i \in C_j} \|\hat{\beta}_i - c_j\|^2 \tag{2.19}$$

where $\| \cdot \|$ is defined as the Euclidean norm(Hartigan,1975). Given K, the general schema of K-means algorithm is

1. Initially, randomly pick K observations as the center of K clusters,

2. Assign the observations to a cluster, and update its center sequentially through alternating the following two step until the algorithm converges

   - Assign each observation to the cluster whose center has the least Euclidean distance to the observation,
   - Based on the cluster result from previous step, update means as well as the center of all K clusters.

A strong consistency property has been proved for this method, indicating that with an appropriate function basis space, such as b-spline basis, the calculated center of the clusters, $\{c_1, c_2, ..., c_k\}$, will converge to the unique $\{c_1^*, c_2^*, ..., c_k^*\}$ when number of curves within each cluster increase (Abraham, Cornillon and et al., 2003)[2].

2.3.2 K-means clustering on FPC scores

We consider a method in which K-means is used with the PC scores of the curves. The framework of this clustering method is

1. Conduct the FPCA on the curves,

2. Obtain the PC scores of the curves from the first few PCs, explaining more than 90% of the variation of the curves,

3. Cluster the curves through the K-means method on the obtained PC scores.

Similar to the K-means clustering on estimated coefficients of B-spline basis functions, this clustering method applies K-means clustering to the coefficients of eigenfunctions. Therefore, this clustering method may also have a strong consistency property as the K-means clustering method on B-spline coefficients. The consistency of this method has been verify through the simulation study.

2.3.3 Selecting the number of clusters

To accomplish the procedure of the hierarchical and k-means clustering, one of the the requirements is to give the number of clusters to the algorithm. However, it is usually an unknown and challenging problem in real applications. To determine the optimal cluster number, the general idea is to provide a set of possible cluster numbers and select the number with the best clustering results.

In clustering analysis, a criterion to evaluate the clustering result is sometimes called an index. Since 1960s, statisticians have proposed various indices. In 2014, the package `NbClust` was developed for the propose of cluster number determination in hierarchical and K-means clustering[13], including 30 different indices to help users decide the cluster number. Given a specific index, the value of the index for each cluster number candidate is calculated; the optimal cluster number is selected based on the index evaluations for each cluster number candidates. For K-means clustering in filtering methods, 26 indices are involved in the cluster number selection, and the final decision of the cluster number is the number that has been suggested by the most indices.

Twenty-six indices are involved in the cluster number selection in filtering methods. Table 2.1 lists the indices and their criterions of optimal cluster number selection.

Name of Index	Optimal number of clusters		Name of Index	Optimal number of clusters
KL	Maximum value of the index		CH	Maximum value of the index
CCC	Maximum value of the index		Scott	Maximum diference between hierarchy levels of the index
Marriot	Max. value of second differences between levels of the index		Tracew	Max. value of second differences between levels
Trcovw	Maximum difference between hierarchy levels of the index		Friedman	Maximum dffierence between hierarchy levels of the index
Rubin	Minimum value of second differences between levels		Cindex	Minimum value of the index
DB	Minimum value of the index		Silhouette	Maximum value of the index
Duda	Smallest number of clusters such that index $>$ critical-value		Pseudot2	Smallest number of clusters such that index $<$ critical-Value
Beale	Number of clusters such that critical value $>= \alpha$		Ratkowsky	Maximum value of the index
Ball	Maximum difference between hierarchy levels of the index		Ptbiserial	Maximum value of the index
Frey	Cluster level before index value < 1.00		McClain	Minimum value of the index
Dunn	Maximum value of the index		Hubert	Graphical method
SDindex	Minimum value of the index		Dindex	Graphical method
Hartigan	Maximum difference between hierarchy levels of the index		SDbw	Minimum value of the index

Table 2.1: Indices Summary for Cluster Number Selection in Filtering Methods

2.4 Adaptive or Model-based Clustering: FunFEM and FunHDDC

FunFEM [9] and FunHDDC [31] are two model-based clustering methods. The FunFEM is refered to the functional discriminative model estimated through Expectation Maximization (EM) algorithm, while the FunHDDC is refered to the functional High-Dimensional Data Clustering. The first step of the funFEM and funHDDC methods is to smoothen the functional data, which is the same as the previous two clustering methods. Then, the functional data are fitted into a functional latent mixture model with lower dimensional subspaces F. After specifying a cluster number K, inference of the latent mixture model is estimated by the expectation maximization (EM) algorithm. The final cluster result for an observation $x(t)$ is obtained by estimating its probability of belonging to the k^{th} cluster through the latent mixture model, where $k = 1, ..., K$.

2.4.1 FunFEM: clustering functional data using a discriminative functional mixture model

Discriminative functional model (DFM)

Instead of directly grouping the observed functional data $\{x_1(t), x_2(t), ..., x_n(t)\}$ into K clusters, the funFEM introduces an unobserved random variable $Z = (Z_1, ..., Z_K) \in \{0, 1\}^K$. For each functional data observation$X(t)$, $Z_k = 1$ if $X(t)$ belongs to k^{th} group, and $Z_k = 0$ otherwise. Therefore, the clustering amounts to predicting the value of $z_i = (z_{i1}, ..., z_{iK})$ for each observed functional data $x_i(t)$.

We remark that the data have been smoothed as a linear combination of basis functions $\phi_1(t), ..., \phi_p(t)$, i.e. $x_i(t) = \sum_{a=1}^{p} b_{ia}\phi_a(t)$, and the coefficient matrix is written as $B = (b_{ia})$. In funFEM, the functional data is then represented in terms of basis functions $\varphi_1(t), ..., \varphi_d(t)$ with $d < K$ and $d < p$, where $\varphi_j(t)$ is the linear combination of $\phi_l(t)$ as $\varphi_j(t) = \sum_{l=1}^{p} u_{jl}\phi_l(t)$ with a constraint that $u'_{jl}u_{il} = 0$ if $j \neq i$ for $1 \leq j, i \leq d$. In other word, the data are expressed in a lower dimension subspace whose orientation is the orthogonal matrix $U_{p \cdot d} = (u_{jl})$. Finally, the functional data $x_1(t), ..., x_n(t)$ is transformed into $x_i(t) = \sum_{b=1}^{d} \lambda_{ib}\varphi_b(t)$. Define the latent expansion coefficients matrix $\Lambda_{d \cdot n} = (\lambda_{ib})$, the relationship between $U_{p \cdot d}$ and B is then expressed as

$$B = U\Lambda + \varepsilon \tag{2.20}$$

where $\varepsilon \in \Re^p$ is random and independent noise, which is assumed to be distributed following a multivariate Gaussian density as

$$\varepsilon \sim N(0, E). \tag{2.21}$$

The latent expansion coefficients Λ are assumed to be n random vectors. Conditionally on $Z_k = 1$, Λ is assumed following a multivariate Gaussian distribution as

$$\Lambda_{|Z_k=1} \sim N(\mu_k, \Sigma_k) \tag{2.22}$$

where μ_k and Σ_k are the mean and the variance-covariance matrix of the k-th group. Under the above assumptions, the marginal distribution of B follows a mixture of Gaussian distribution as

$$p(b) = \sum_{k=1}^{K} \pi_k \Phi(b; U\mu_k, U^t \Sigma_K U + E) \tag{2.23}$$

where b is the coefficients of the original basis functions $\phi_a(t), a = 1, ..., p$ for a curve $X(t)$, Φ is the standard Gaussian density, and $\pi_k = P(Z_k = 1)$ is the prior probability of the k_{th} group.

Finally, given V as the orthogonal component of U such that $U^T V = 0$, the noise covariance matrix E conditional on $Z_k = 1$ as $E_{Z_k=1} = \text{cov}(W^t B | Z_k = 1) = W^t \Sigma_k W$ can be assumed in the following form:

$$\left(\begin{array}{c|ccc} \boxed{\Sigma_k} & & \mathbf{0} & \\ \hline & \beta & & 0 \\ \mathbf{0} & & \ddots & \\ & 0 & & \beta \end{array}\right) \begin{array}{l} \left.\vphantom{\Sigma_k}\right\} d \\ \left.\vphantom{\begin{array}{c}\beta\\ \ddots\\ \beta\end{array}}\right\} p-d \end{array}$$

To understand the noise covariance matrix form, it can be said that within the k-th cluster, the noise variance is modeled as Σ_k; and outside the cluster, the variance is modeled by the β parameter. This way of modeling the noise covariance matrix of basis functions coefficients $E_{Z_k=1}$ is called discriminative functional model (DFM), and we denote this type of the DFM as $textDFM_{[\Sigma_k \beta]}$.Fraley and Raftery(1999)[17] introduced a family of DFM models including of this $E_{Z=k}$ type. This is also summarized by Bouveyron, Come and Jacques(2015)[9]. Some model selection criterions can assist us in selecting which DFM model to use, the number of clusters K and the number of basis functions d.

FunFEM algorithm on model inference

The EM algorithm is applied in maximizing the likelihood of the above model because the variable z_i of each curve $x_i(t)$ is unknown. Bouveyron, Come and Jacques(2015) proposed a FunFEM algorithm that first estimates the matrix U to obtain the most discriminative subspace, and then conducts the EM algorithm to maximize the model log-likelihood[9]. The FunFEM algorithm obtain the maximum likelihood iteratively; in the q-th iteration, the algorithm alternates over the following three steps:

1. **F step.** Conditional on the posterior probabilities estimated in the $(q-1)$-th iteration, $t_{ik}^{(q)} = E[z_{ik}|b_i, \theta^{(q-1)}]$, the F step determines the matrix U, which represents the orientation of the discriminative latent subspace F where the data are best separated into K clusters. Such a functional subspace F is desired as it maximize the variance between clusters, while minimizing the variance within clusters.

   In Bouveyron, Come and Jacques' paper, they show that finding the subspace F is equivalent to finding the discriminative functions u such that

$$\underset{u}{\operatorname{argmax}} \frac{\operatorname{Var}(E[\Phi(X)|Z])}{\operatorname{Var}(\Phi(X))} \tag{2.24}$$

   where $\Phi(X) = \int_{[0,T]} X(t)u(t)dt$ is the projection of X on the discriminative function u. The solution of equation (4.13) is the eigenfunction u associated with the largest eigenvalue η of the following eigen-equation:

$$Du = \eta Cu, \tag{2.25}$$

   where C is the covariance operator of X as $C(t,s) = E[(X(t) - E[X(t)])(X(s) - E[X(s)])]$, and D is the integral between-cluster covariance operator $D(t,s) = E[E[X(t) - E[X(t)]|Z]E[X(s) - E[X(s)]|Z]]$.

   Numerically, C is estimated by $\hat{C}(t,s) = \frac{1}{n}\phi'(t)B'B\phi(s)$, where B is the $n \times p$ matrix of basis function coefficients and $\phi(t)$ is the p-vector of basis functions $\phi_j(t), j = 1, ..., p$. D is estimated iteratively conditional on the posterior probabilities $t_{ik}^{(q-1)}$ by $\hat{D}^{(q)}(t,s) = \frac{1}{n}\phi'(t)B'TT'B\phi(s)$, where $T = (\frac{t_{ik}^{(q-1)}}{\sqrt{n_k^{(q-1)}}})_{i,k}$ is an $n \times k$ matrix with $n_k^{(q-1)} = \sum_{i=1}^{n} t_{ik}^{(q-1)}$.

Assume that the discriminative function u can be smoothed with the same basis functions as the observed curved $u(t) = \sum_{j=1}^{p} v_j \phi_j(t) = \phi'(t)v$. By substituting the $\hat{D}^{(q)}$ and $\hat{C}$ to the eigen-equation $Du = \eta Cu$, the problem converts to solve the v such that

$$(B'BH)^{-1}B'TT'BHv = \eta v \tag{2.26}$$

where $H = \int_{[0,T]} \phi(s)\phi'(s)ds$. Thus, the final problem is equivalent to finding the first d eigenvectors vs as $v_j = (v_{j1}, ..., v_{jp}), j = 1, ..., d$ for the matrix $(B'BH)^{-1}B'TT'BH$. The d discriminative functions $u_1(t), ..., u_d(t)$ are obtained as the linear combination of basis functions $\phi(t)$ with the coefficient vectors $v_1, ..., v_d$, and the $p \times d$ orientation matrix $U^{(q)} = (v_{jl}^{(q)})_{p\times d}$.

2. **M step.** Conditional on the orientation matrix U obtained from the F step, the M step follows the classical scheme of the EM algorithm to maximize the conditional expectation of the full log-likelihood $Q(\theta; \theta^{(q-1)}) = E[l(\theta; B, z_1, ..., z_n)|B, \theta^{(q-1)}]$. where $\theta = (\pi_k, \mu_k, \Sigma_k, \beta)_k$ for $k = 1, ..., K$. By maximizing the conditional expectation, the following model parameters are updated: $\pi_k^{(q)}, \mu_k^{(q)}, \Sigma_k^{(q)}$ and $\beta^{(q)}$. The updates are as follows

   - $\pi_k^{(q)} = n_k^{(q-1)}/n,$
   - $\mu_k^{(q)} = \frac{1}{n_k^{(q-1)}} \sum_{i=1}^{n} t_{ik}^{(q-1)} U^{(q)}$
   - $\Sigma_k^{(q)} = U^{(q)t} C_k^{(q)} U^{(q)}$
   - $\beta^{(q)} = (trace(C^{(q)}) - \sum_{j=1}^{d} v_j^{(q)t} C^{(q)} v_j^{(q)} / (p - d)$

   where $C_k = \frac{1}{n_k^{(q-1)}} \sum_{i=1}^{n} t_{ik}^{(q-1} (b_i - \mu_k^{(q-1)})(b_i - \mu_k^{(q-1)})^t$. The updated formula varies with different forms of the noise covariance matrix in the DFM family. These forms are obtained by Bouveyron and Brunet in 2012 (2012).

3. **E step.** Conditional on the updated model parameters from the M step, the E step updates the posterior probabilities $t_{ik}^{(q)} = E[z_{ik}|b_i, \theta^{(q)}]$, as well as the posterior probability that the curve $x_i(t)$ belongs to the k^{th} cluster under the current model. Note that $\theta_k^{(q)} = (\pi_k^{(q)}, \mu_k^{(q)}, \Sigma_k^{(q)}, \beta^{(q)})$. Through the Bayes theorem, the posterior probabilities are

$$t_{ik}^{(q)} = \frac{\pi_k^{(q)} \phi(b_i, \theta_k^{(q)})}{\sum_{l=1}^{K} \pi_l^{(q)} \phi(b_i, \theta_l^{(q)})}. \tag{2.27}$$

2.4.2 FunHDDC: a functional extension of the high-dimensional data clustering (HDDC) algorithm

FunFEM v.s. FunHDDC

FunHDDC is another functional data clustering algorithm developed by Schemutz, Jacques, Bouveyron, etc.(2017)[31], which allows the clustering algorithms to be applied to multivariate functional data. Similar to FunFEM, FunHDDC also reforms the functional data to a lower dimension subspace and then clusters the data through fitting a functional latent mixture model using EM algorithm.

The main difference between FunFEM and FunHDDC is that FunFEM fixes a common d over all K clusters with $d < K$, whereas the FunHDDC applies FPCA more straightforwardly and allows various lower dimension numbers d_k among K clusters, where d_k is not necessarily less than K. Although the FunHDDC develops more flexibility on lower dimension number selection, its within-cluster covariance matrix has to be in a diagonal form.

Generative model for the functional HDDC (FunHDDC) algorithm

In a way similar to FunFEM, given clusters number K, the FunHDDC algorithm defines a latent variable Z_{ik} such that $Z_{ik} = 1$ if $X_i(t)$ belongs to cluster k and $Z_{ik} = 0$ otherwise. With the observed curves $x_1(t), ..., x_n(t)$, if all Z_{ik} are given by z_{ik} for $i = 1, ..., n$ and $k = 1, ..., K$, then the number of the curves in the k-th cluster is $n_k = \sum_{i=1}^{n} z_{ik}$. Given the original P basis functions $\phi_1(t), ..., \phi_P(t)$ we form $x_i(t) = \sum_{a=1}^{P} \beta_{ia}\phi_a(t)$. For a specific cluster k, the curves are restricted to a d_k dimensional functional subspace such that $d_k < P$ for $k = 1, ..., K$. Through the numerical application of FPCA, the first P eigenfunctions $\varphi_m^{(k)}(t)$ can be estimated by a linear transformation of $\phi_j(t)$ such that

$$\varphi_m^{(k)}(t) = \sum_{l=1}^{P} w_{ml}^{(k)} \phi_l(t), m = 1, ..., P \tag{2.28}$$

Then we can define the orthogonal $P \times P$ matrix $W_k = (w_{l_k})$ as a matrix of eigenfunction coefficients. Therefore, the first d_k eigenfunctions form the d_k dimensional functional subspace. The direction of the k-th cluster on the subspace is U_k, and constructed by splitting the W_k into two parts such that $W_k = [U_k, V_k]$ with U_k a $P \times d_k$ matrx, and V_k a $P \times (P - d_k)$ matrix.

Moreover, for all n_k curves in cluster k, the FPCA scores $(\delta_{i_k})_{1 \le i_k \le n_k}$ are assumed to follow a Gaussian distribution as

$$\delta_{i_k} \sim N(\mu_k, \Delta_k) \tag{2.29}$$

where $\mu_k \in \Re^P$ is the mean and Δ_k is the covariance matrix. Similar to DFM model family, Δ_k can be modeled in different forms. One of the forms is the following:

$$\left(\begin{array}{cc}
\boxed{\begin{matrix} a_{k1} & & 0 \\ & \ddots & \\ 0 & & a_{kd_k} \end{matrix}} & \mathbf{0} \\
\mathbf{0} & \boxed{\begin{matrix} b_k & & 0 \\ & \ddots & \\ & & \ddots \\ 0 & & b_k \end{matrix}}
\end{array}\right)
\begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \end{matrix}}\right\} d_k \\ \left.\vphantom{\begin{matrix} a \\ a \\ a \\ a \end{matrix}}\right\} p - d_k \end{matrix}$$

Finally, the FPCA scores δ of functional data $X(t)$ is distributed following a Gaussian mixture model with density

$$p(\delta) = \sum_{k=1}^{K} \pi_k N(\delta; \mu_k, \Delta_k), \tag{2.30}$$

where $N(\cdot)$ is the Gaussian density function and $\pi_k = P(Z_k = 1)$. We use the EM algorithm to conduct the parameters estimation in the model and the z_{ik} prediction sequentially.

The EM algorithm on funHDDC model inference

A standard EM algorithm is applied directly to FunHDDC which in turn calculates the expectation of the complete log-likelihood of z_{ik} (E step) using the model with the parameters estimated from the previous iteration. The parameters are then re-estimated by maximizing the log-likelihood expectation (M step) calculated in the E step. Specific for this FunHDDC method, in the q-th iteration, the algorithm alternatives over the following two steps:

1. **E step.** In this step, the posterior probability of observed curve $x_i(t)$ being assigned to the k-th cluster is computed as

$$t_{ik}^{(q)} = E[Z_{ik}|c_i, \theta^{(q-1)}] = 1/\sum_{l=1}^{K} \exp[\frac{1}{2}(H_k^{(q-1)}\beta_i) - H_l^{(q-1)}(\beta_i))] \tag{2.31}$$

where $\theta^{(q-1)}$ are the model parameters generating from q^{th} iteration with $\theta = (\pi_k, \mu_k, a_{kj}, b_k, w_{kj})_{kj}$ for $1 \leq k \leq K$, $1 \leq j \leq d_k$, and w_{kj} is the j^{th} column of W_k, where $H_k^{(q-1)}(\beta)$ is the defined as the cost function for $\beta \in \Re^P$, and the β is the coefficient vector of the P original basis functions $\phi_i(t)$.

2. **M step.** In this step, the parameters in the model are estimated by maximizing the expectation of the complete log-likelihood conditional on t_{ik} achieved from the E step. The update for the parameters is

   - $\pi_k^{(q)} = \frac{\eta_k^{(q)}}{n}$
   - $\mu_k^{(q)} = \frac{1}{\eta_k^{(q)}} \sum_{i=1}^n t_{ik}^{(q)} c_i$ where $\eta_k^{(q)} = \sum_{i=1}^n t_{ik}^{(q)}$
   - $a_{kj}^{(q)}$ is set to the d_k largest eigenvalues of $H^{1/2} C_k^{(q)} H^{1/2}$
   - $b_k^{(q)} = \frac{1}{P-d_j}[tr(H^{1/2} C_k^{(q)} H^{1/2}) - \sum_{j=1}^{d_k} \hat{a}_{kj}^{(q)}]$

   where $H = \int_0^T \phi'(t)\phi(t)$ and $C_k^{(q)} = \frac{1}{\eta_k^{(q)}} \sum_{i=1}^n t_{ik}^{(q)} (c_i - \mu_k^{(q)})^t (c_i - \mu_k^{(q)})$.

2.4.3 Model selection and choosing the number of clusters

Two crucial parameters in FunFEM and FunHDDC are the dimension of the subspaces and the number of clusters. These parameters should be selected during the clustering procedure. In FunFEM, the number of clusters, common intrinsic dimension d and the model form in the DFM family are selected simultaneously. Akaike information criterion (AIC)[5], Bayesian information criterion (BIC)[32] and Integrated Completed Likelihood (ICL)[8] are recommended for parameter and model selection and implemented by the R package `FunFEM`. Given a fitted model M, the details of these criteria are as follows

- AIC$(M) = l(\hat{\theta}) - \xi(M)$, where $l(\hat{\theta})$ is the log-likelihood of the estimated parameter $\hat{\theta}$, and $\xi(M)$ is the number of free parameters in the models;
- BIC$(M) = -l(\hat{\theta}) + \frac{\xi(M)}{2} log(n)$, where n is the number of observations;
- ICL = BIC - $\sum_{k=1}^K \sum_{i=1}^n Z_{ik} \times \log(z_{ik})$, where Z_{ik} is the indicator for the cluster of the i-th observation such that $Z_{ik} = 1$ if the i^{th} observation belongs to the k^{th} cluster and 0 otherwise.

Compared to BIC, ICL determines the number of cluster through the final allocation results of the observations; moreover, it has been observed to choose the model and cluster number with more separated cluster patterns[31]. Before selection, the set of models and cluster number candidates are established respectively. Then, the values of the criterion are computed for all combinations of model and cluster number. We will determine the model type and number of clusters from the combination with the smallest criterion value.

Similar to FunFEM, FunHDDC chooses the model type and number of clusters using AIC, BIC or ICL. Moreover, Cattell's scree-test[12] is applied for choosing the intrinsic dimensions d_k for each cluster. This test selects a dimension for which subsequent eigenvalues have smaller differences than the thresholds selected from AIC, BIC or ICL. The algorithm and criterions for model and cluster number selection for FunHDDC are implemented in R package `FunHDDC`.

Chapter 3

Simulation Study

In this chapter, a simulation study is set up to validate the performance of the clustering methods that we have described in this work. Two main interests in validation are the accuracy of cluster exchangeability selection and the precision of clustering results. Four clusters of curves with four different mean function respectively are generated. Moreover, the clustering performance under different scenarios, such as different number of curves in the each clusters and different patterns of within-cluster variance, is discussed.

3.1 Overall Simulation Setup

In the simulation study, we fix a real number of clusters $K = 4$, and let each cluster contain an equal quantity of curves. The curves in each cluster are in the following form:

- Group 1: $X(t) = \sin(2t) + \epsilon_1(t)$,
- Group 2: $X(t) = 2\sin(2t) + \epsilon_2(t)$,
- Group 3: $X(t) = \frac{1}{4}\sin(2t) + \epsilon_3(t)$,
- Group 4: $X(t) = \sin(t) + \epsilon_4(t)$,

where the range of t is $t \in [0, 10]$, and $\epsilon_k(t)$ is white noise in the k-th cluster. In addition, we consider four different scenarios of $\epsilon_i(t)$ of $i = 1, ..., 4$ such that

- Scenario 1: the variance of $\epsilon_i(t)$ is the same over time and clusters,
  - i.e. $\epsilon_i(t) \sim N(0, \sigma)$ for all $i = 1, ..., 4$ and $t \in [0, 10]$ where σ is a constant respect to the size of the overall variation;
- Scenario 2: the variance of $\epsilon_i(t)$ varies over clusters and is proportional to the range of the mean curve in each cluster,
  - i.e. $\epsilon_1(t) \sim N(0, \sigma)$, $\epsilon_2(t) \sim N(0, 2\sigma)$, $\epsilon_3(t) \sim N(0, \frac{1}{4}\sigma)$, $\epsilon_4(t) \sim N(0, \frac{1}{2}\sigma)$ for $t \in [0, 10]$;

- Scenario 3: the variance of $\epsilon_i(t)$ varies over time and proportional to the absolute value of the $\sin(at)$ at time t, where $a = 1$ for clusters 1,2,3 and $a = 2$ for cluster 4, i.e. $\epsilon_i(t) \sim N(0, \sigma_i(t))$ with
  - $\sigma_i(t) \approx \sigma \cdot |\sin(2t)|$ for $t \in [0, 10]$ and $i = 1, 2, 3$,
  - $\sigma_i(t) \approx \sigma \cdot |\sin(t)|$ for $t \in [0, 10]$ and $i = 4$;
- Scenario 4: the variance of $\epsilon_i(t)$ varies over clusters and time as a combination of 2 and 3,
  - $\epsilon_1(t) \sim N(0, \sigma \cdot |\sin(2t)|)$, $\epsilon_2(t) \sim N(0, 2\sigma \cdot |\sin(2t)|)$,
  - $\epsilon_3(t) \sim N(0, \frac{1}{4}\sigma \cdot |\sin(2t)|)$, $\epsilon_4(t) \sim N(0, \frac{1}{2}\sigma \cdot |\sin(t)|)$.

During data simulation under each scenario, each curve is generated pointwise with 1001 equidistant observed time points, t = 0, 0.01,...,9.99,10, and then is smoothed by 13 cubic B-splines with 9 equally spaced interior points t = 0.1, 0.2,...,0.9. Additionally, we examinate the performance of clustering methods with different number of curves n in each cluster with $n = 20, 50, 100$, and different constant σ in every variation scenario with $\sigma = 1.25, 2.5$.

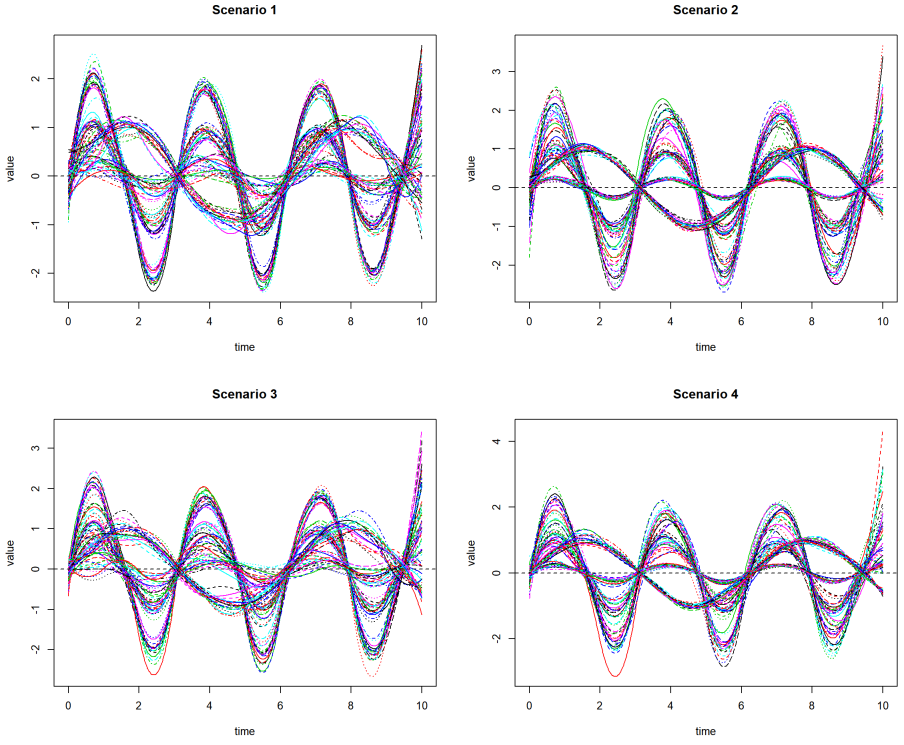

Figure 3.1: Smoothed Simulated Data for Four Different Scenarios with $n = 20$ and $\sigma = 1.25$

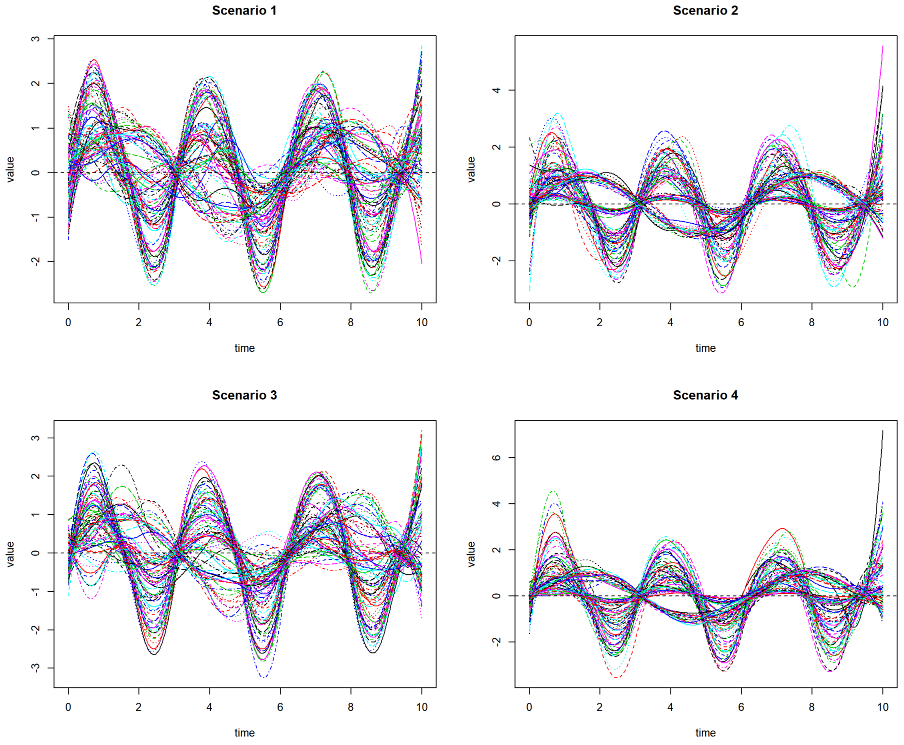

Figure 3.2: Smoothed Simulated Data for Four Different Scenarios with $n = 20$ and $\sigma = 2.5$

3.2 Simulation Study for Selecting the Number of Clusters

The simulation study in this section assesses the ability of filtering and adaptive methods to choose the correct number of cluster. In this study, for each scenario, 200 simulated data sets have been generated through repeating the simulation setup. For each generated dataset, the number of clusters are determined through

- 26 indices from R package `NbClust` for filtering methods, K-means clustering on B-spline coefficients and FPC scores;
- BIC and ICL criterion from the R package `FunFEM` and `FunHDDC` for adaptive methods, FunFEM and FunHDDC.

Given the set of cluster number $\{2, 3, 4, 5, 6\}$, each filtering method suggest one optimal candidate cluster number from the candidate for each simulated data set. Table 3.1 lists the proportion of each number being selected using different clustering methods and under various situations.

Scenario	Clustering Methods	Number K of clusters (n=20/50)				
		2	3	4	5	6
1	K-means on B-spline Coefficients	51.5/71	20/19.5	8.5/9.5	-/-	-/-
	K-means on FPC Scores	-/-	97.5/99.5	2.5/0.5	-/-	-/-
	FunFEM (BIC Criterion)	-/-	-/-	83/**39.5**	17/36.5	-/24
	FunFEM (ICL Criterion)	-/-	1/-	**92**/22.5	6.5/31	0.5/36.5
	FunHDDC (BIC Criterion)	-/-	91/86	9/10	-/3	-/1
	FunHDDC (ICL Criterion)	5/-	91.5/82	3.5/10.5	-/5	-/2.5
2	K-means on B-spline Coefficients	1.5/-	98.5/100	-/-	-/-	-/-
	K-means on FPC Scores	-/-	100/100	-/-	-/-	-/-
	FunFEM (BIC Criterion)	-/-	-/-	**73/31**	26/35.5	1/33.5
	FunFEM (ICL Criterion)	-/	-/	**73.5**/20.5	23.5/34	3/45.5
	FunHDDC (BIC Criterion)	74/32	9/1	9.5/25.5	3.5/12.5	1/29
	FunHDDC (ICL Criterion)	83.5/40.5	4/8	6.5/16.5	4.5/17.5	1.5/17.5
3	K-means on B-spline Coefficients	73/99.5	20/0.5	7/-	-/-	-/-
	K-means on FPC Scores	-/-	85.5/67.5	8/**32.5**	-/-	6.5/-
	FunFEM (BIC Criterion)	-/-	-/-	**79**/0.5	21/66.5	-/33
	FunFEM (ICL Criterion)	-/-	3.5/-	65.5/3.5	27/37	4/59.5
	FunHDDC (BIC Criterion)	66/34.5	27/65.5	7/-	-/-	-/-
	FunHDDC (ICL Criterion)	72.5/43	24/42.5	3/10	0.5/4	-/0.5
4	K-means on B-spline Coefficients	51.5/44	40/56	8.5/-	-/-	-/-
	K-means on FPC Scores	-/-	97.5/100	2.5/-	-/-	-/-
	FunFEM (BIC Criterion)	-/-	-/-	**83/7.5**	17/40.5	-/54
	FunFEM (ICL Criterion)	-/-	0.5/-	59.5/3.5	32.5/47.5	7.5/47
	FunHDDC (BIC Criterion)	-/76.5	91/0.5	9/**7.5**	-/14	-/1.5
	FunHDDC (ICL Criterion)	64.5/78.5	17/3	10.5/5.5	4/7.5	4/5.5

Table 3.1: Percentage of Number of Cluster Selection over 200 Simulations using Different Clustering Methods in 4 Different Scenarios under the Overall Variation $\sigma = 2.5$. Real Number of Clusters is 4.

Table 3.1 is the result of cluster number selection from 200 simulated data sets under $\sigma = 2.5$. It shows that the FunFEM algorithm has a dominant advantage for detecting the real number of clusters compared to the other three methods. When the number of curves in each cluster is smaller (i.e. 20), the accuracy of cluster number detection in FunFEM reaches over 70% under the BIC criterion and over 60% under the ICL criterion; however, when the number of curves increases to 50, FunFEM begins to overestimate the number of clusters. On the other hand, under small number of curves, the other three clustering methods usually underestimate the number of clusters, such that over 70% of selected cluster numbers are 2 and 3 in all 4 scenarios. After the curves number increases to 50, the only conspicuous improvement of the detection rate of correct cluster number 4 is observed with K-means clustering on FPC scores in scenario 3, which is 32.5% and much higher than the rate of FunFEM, 0.5% under BIC and 3.5% under ICL.

Scenarios	Clustering Methods	Number K of clusters (n=20/50)				
		2	3	4	5	6
1	K-means on B-spline Coefficients	-/-	99/98.5	1/1.5	-/-	-/-
	K-means on FPC Scores	-/-	100/100	-/-	-/-	-/-
	FunFEM (BIC Criterion)	-/-	-/-	**99/38.5**	1/33	-/28.5
	FunFEM (ICL Criterion)	-/-	-/-	80.5/23	18/40.5	1.5/36.5
	FunHDDC (BIC Criterion)	1.5/1.5	95.5/65	2/23	-/7.5	1/3
	FunHDDC (ICL Criterion)	9/-	76/72.5	13/19.5	1/5	1/3
2	K-means on B-spline Coefficients	-/-	100/100	-/-	-/-	-/-
	K-means on FPC Scores	-/-	100/100	-/-	-/-	-/-
	FunFEM (BIC Criterion)	-/-	-/-	**88**/33.5	9.5/18	2.5/48.5
	FunFEM (ICL Criterion)	-/-	-/-	73/25.5	22/34.5	5/40
	FunHDDC (BIC Criterion)	8.5/-	36/0.5	22.5/20	21.5/47	11.5/50.5
	FunHDDC (ICL Criterion)	14.5/-	41/21	19.5/**41**	13/19.5	12/18.5
3	K-means on B-spline Coefficients	3/-	80/96	17/4	-/-	-/-
	K-means on FPC Scores	-/-	97.5/99.5	2.5/0.5	-/-	-/-
	FunFEM (BIC Criterion)	-/-	-/-	**77.5**/0.5	21/36.5	1.5/63
	FunFEM (ICL Criterion)	-/-	-/-	65/2.5	30.5/43	4.5/54.5
	FunHDDC (BIC Criterion)	12/2	77.5/60	7.5/**30**	3/0.5	-/7.5
	FunHDDC (ICL Criterion)	14/2	68.5/73	12.5/18.5	4/4	1/2.5
4	K-means on B-spline Coefficients	-/-	100/100	-/-	-/-	-/-
	K-means on FPC Scores	-/-	100/100	-/-	-/-	-/-
	FunFEM (BIC Criterion)	-/-	-/-	**76**/8.5	24/56.5	-/35
	FunFEM (ICL Criterion)	-/-	-/-	60.5/3	33/20	6.5/47
	FunHDDC (BIC Criterion)	44/23	23.5/9	13/**11**	14/25	5.5/32
	FunHDDC (ICL Criterion)	40.5/19.5	20/9.5	14.5/12	12/27	13/32

Table 3.2: Percentage of Number of Cluster Selection over 200 Simulations using Different clustering methods in 4 Different Scenarios under $\sigma = 1.25$. Real Number of Clusters is 4.

Table 3.2 is the result of cluster number selection from 200 simulated data sets under $\sigma = 1.25$. With this smaller σ value, the simulated curves in the same cluster should be closer to each other, which means the patterns of all four clusters are more obvious and distinctive to each other. In this case, FunFEM still performs the best when the number of curves is small. When the number of curves is larger (i.e., 50 per cluster), FunFEM still has the overestimation problem. In this case, FunHDDC performs slightly better than FunFEM. However, the detection rate of the correct cluster numbers can only attain at most 40%. In this case, increase of detection rate of the correct cluster number in scenarios 1 and 2 are observed in FunFEM and FunHDDC, in contrast to table 3.1. This phenomenon does match the expectation because both methods should be good at detecting the clusters with different patterns of variance; and more distinctive clusters should be easier to detect the real unknown number of clusters in reasonable clustering methods. However, in scenario 3 and 4 which involve the overtime changing patterns in variance, the detection rate has no apparent improvement or even a decrease in all the clustering methods. The non-improvement may imply that clustering methods are not good at dealing with the situation such that the variance changes over time.

To conclude, when the number of curves in each cluster is about 20, FunFEM using BIC or ICL criterion is the best tool for cluster number detection based on the simulation result. In this case that detection rate can reach above 60% (ICL) or 70% (BIC). When the number of curves increases to around 50, the FunFEM algorithm has an over-estimation problem and FunHDDC then becomes the best algorithm. However, the FunHDDC in the large curves number case can at most detect 40% of the correct cluster number from 200 simulated data sets; in other word, with a large amount of curves, none of the methods can detect the exact cluster number very well.

3.3 Validation of Clustering Results

In this section, the aim of the simulation study is to evaluate the result of the clustering through different methods when the real number of clusters is given. Inspired by Tibshirani and Walther (2005)[34], the idea of the evaluation is to consider the clustering problem as supervised classification problem. By considering the assigned cluster labels in the original simulated data as the "true" class labels and the cluster labels produced by the clustering methods as the "predicted" class labels, a 2×2 confusion table can be constructed for the further analysis, such as calculation of sensitivity or specificity. Unlike the supervised classification problem, the assigned labels in a clustering study do not have an actual numerical or categorical meaning. Thus, during the construction of the confusion table, instead of judging whether the simulated curves have been assigned to the same class labels, the "true positive" (TP),"true negative" (TN), "false positive" (FP) and "false negative" (FN) are redefined based on the relationship between the curves. For a confusion table with the following form,

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Table 3.3: General Form of Contingency Table

the entries in the table are redefined as

- True Positive(TP): the number of observed curve pairs where both curves have the same class label in both the "true" clusters and "predicted" clusters.

- True Negative(TN): the number of observed curve pairs where two curves have different class labels in both the "true" clusters and "predicted" clusters.

- False Positive(FP): the number of observed curve pairs where two curves have different class labels in the "true" clusters, but have the same class label in the "predicted" clusters.
- False Negative(FN): the number of observed curve pairs where two curves have the same class label in the "true" clusters, but have different class labels in the "predicted" clusters.

Then, the performance of the clustering methods can be evaluated through obtaining the following derivations[15]:

- Sensitivity $= \frac{TP}{TP+FN}$, which represents the proportion of pairs of curves that have been detected in the same cluster during the clustering procedure over all same-cluster pairs of curves in the original simulated data;
- Specificity $= \frac{TN}{TN+FP}$, which represents the proportion of pairs of curves that have been detected in different clusters during the clustering procedure over all different-cluster pairs of curves in the original simulated data;
- Precision $= \frac{TP}{TP+FP}$, which represents the proportion of pairs that are exactly in the same cluster in the original simulated data over all same-cluster pairs of curves detected from the clustering procedure;
- Negative Precision $= \frac{TN}{TN+FN}$, which represents the proportion of pairs that are exactly in different clusters in the original simulated data over all different-cluster pairs of curves detected from the clustering procedure;
- Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$, which represents that overall pairs of generated curves, the proportion of the pairs that their clustering relationships are estimated "correctly" as the same as the ones in the original simulated data.

According to these definitions, the clustering methods are evaluated as having better performance when they can obtain higher value of derivations. The goodness of estimated mean curve of each cluster is another interest in the simulation study. The main idea of the assessment of the mean curve estimation is measuring the difference between the estimated mean curves and the original mean curves. A within-cluster mean squared error is defined to assess the mean curve estimations. Given K clusters and N discrete time observations $t_1, t_2, ..., t_N$, as well as the real and estimated mean curves of each cluster, say $C_1(t), C_2(t), ..., C_K(t)$ and $\hat{C}_1(t), \hat{C}_2(t), ..., \hat{C}_K(t)$ respectively, the within-cluster mean squared error (WCMSE) is then defined as

$$\text{WCMSE} = \frac{1}{K}\frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N}(\hat{C}_j(t_i) - C_j(t_i))^2 \tag{3.1}$$

When WCMSE is smaller, the average difference between the estimated and real within-cluster mean curves is smaller among all observed time points, which indicates that the clustering method performs better in the view of better estimation of mean curve within each cluster. An essential difficulty in WCMSE measurement is how to match the estimated clusters to the original ones; due to the meaninglessness of the cluster label, i.e. the cluster with label "n" in original data may refer to the cluster with a label "m" produced by the clustering algorithm. To solve this matching problem, in each produced cluster, the proportion of every original cluster label is calculated; then the produced cluster is consider as the estimator of the original cluster whose label has the largest proportion.

Tables 3.4 and 3.5 are the summary tables of the clustering result validation for 6 clustering methods, unsupervised random forest clustering on B-spline coefficients or FPC scores, K-means clustering on B-spline coefficients or FPC scores, funFEM and funHDDC. The average of sensitivity, specificity, precision, negative precision and WCMSE are calculated for each clustering method in all the combinations of four different scenarios of variation and three numbers of curves in each cluster, under $\sigma = 2.5$ and real number of cluster $K = 4$. Because the average of sensitivity, specificity and negative precision are very close in every clustering methods, we only list the summary result of the sensitivity in this section. Table 3.4 contains the average of sensitivity and precision, and Table 3.5 contains the average of accuracy and WCMSE. The highlighting in the table indicates the best clustering methods based on the value of the corresponding derivations for every scenario.

		Sensitivity			Precision		
		(Curves Number/Cluster)			(Curves Number/Cluster)		
Scenario	Methods	20	50	100	20	50	100
1	URF on B-spline Coefficients	**0.998**	**0.997**	**0.997**	**0.998**	**0.996**	**0.997**
	URF on FPC Scores	0.982	0.990	0.988	0.979	0.989	0.987
	K-means on B-spline Coefficients	0.953	0.953	0.950	0.880	0.879	0.876
	K-means on FPC Scores	0.951	0.940	0.950	0.860	0.827	0.853
	FunFEM	**0.998**	**0.998**	**0.998**	**0.997**	**0.997**	**0.997**
	FunHDDC	0.898	0.913	0.916	0.773	0.796	0.794
2	URF on B-spline Coefficients	0.980	0.977	0.966	**0.975**	**0.975**	**0.965**
	URF on FPC Scores	0.974	0.975	0.956	0.969	0.960	0.920
	K-means on B-spline Coefficients	0.918	0.928	0.924	0.776	0.804	0.791
	K-means on FPC Scores	0.936	0.939	0.930	0.819	0.827	0.802
	FunFEM	**0.988**	**0.990**	**0.990**	**0.974**	**0.975**	**0.972**
	FunHDDC	0.921	0.948	0.959	0.792	0.873	0.908
3	URF on B-spline Coefficients	**0.964**	**0.968**	**0.957**	**0.960**	**0.967**	**0.956**
	URF on FPC Scores	0.932	0.943	0.931	0.918	0.933	0.920
	K-means on B-spline Coefficients	0.923	0.945	0.941	0.876	0.916	0.905
	K-means on FPC Scores	**0.953**	**0.967**	**0.971**	0.908	0.933	0.949
	FunFEM	**0.959**	**0.967**	**0.970**	**0.952**	**0.954**	**0.960**
	FunHDDC	0.839	0.866	0.873	0.740	0.778	0.788
4	URF on B-spline Coefficients	0.907	0.912	0.901	0.871	0.894	0.887
	URF on FPC Scores	0.880	0.861	0.865	0.848	0.779	0.687
	K-means on B-spline Coefficients	0.863	0.868	0.878	0.728	0.723	0.743
	K-means on FPC Scores	0.904	0.907	0.905	0.788	0.798	0.795
	FunFEM	**0.968**	**0.977**	**0.973**	**0.946**	**0.964**	**0.949**
	FunHDDC	0.843	0.847	0.852	0.689	0.690	0.699

Table 3.4: Summary Table of Average Sensitivity and Precision from the Clustering Results over 200 Simulations in 4 Different Scenarios under $\sigma = 2.5$ and Number of Clusters $K = 4$.

The two best clustering methods suggested by Table 3.4 are the unsupervised random forest clustering algorithm on B-spline coefficients and FunFEM model-based clustering algorithm. In Table 3.4, all 6 methods have best performance with higher sensitivity and precision in both scenarios 1 and 2, the scenarios without the time-varying pattern in variance. In scenarios 1 and 2, the overall average sensitivity and precision reaches 90% and 80% respectively, and the average sensitivity and precision of two best clustering methods reach 99% and 97%. However, when the variance changes over time in scenarios 3 and 4, the average sensitivity and precision in most methods decrease, to no more than 96%.

		Accuracy			WCMSE		
		(Curves Number/Cluster)			(Curves Number/Cluster)		
Scenario	Methods	20	50	100	20	50	100
1	URF on B-spline Coefficients	**0.999**	**0.998**	**0.998**	**0.009**	**0.007**	**0.006**
	URF on FPC Scores	**0.991**	**0.995**	**0.994**	**0.009**	**0.007**	**0.006**
	K-means on B-spline Coefficients	0.949	0.948	0.946	0.027	0.024	0.024
	K-means on FPC Scores	0.941	0.926	0.937	0.024	0.024	0.020
	FunFEM	**0.999**	**0.998**	**0.999**	**0.010**	**0.007**	**0.006**
	FunHDDC	0.902	0.911	0.910	0.045	0.041	0.042
2	URF on B-spline Coefficients	**0.989**	**0.988**	**0.983**	**0.012**	**0.008**	**0.007**
	URF on FPC Scores	0.986	0.986	0.965	0.012	0.009	0.016
	K-means on B-spline Coefficients	0.905	0.916	0.910	0.042	0.036	0.036
	K-means on FPC Scores	0.924	0.924	0.914	0.034	0.031	0.033
	FunFEM	**0.989**	**0.989**	**0.988**	**0.014**	**0.011**	**0.011**
	FunHDDC	0.910	0.944	0.960	0.037	0.022	0.022
3	URF on B-spline Coefficients	**0.982**	**0.984**	**0.978**	**0.011**	**0.008**	**0.007**
	URF on FPC Scores	0.963	0.969	0.963	**0.012**	**0.009**	**0.008**
	K-means on B-spline Coefficients	0.945	0.961	0.956	0.026	0.018	0.019
	K-means on FPC Scores	0.960	0.970	0.977	0.019	0.013	0.010
	FunFEM	**0.978**	**0.979**	**0.981**	**0.013**	**0.010**	**0.008**
	FunHDDC	0.883	0.897	0.901	0.050	0.041	0.038
4	URF on B-spline Coefficients	0.944	0.951	0.947	**0.024**	**0.014**	**0.012**
	URF on FPC Scores	0.932	0.902	0.865	0.032	0.048	0.067
	K-means on B-spline Coefficients	0.881	0.877	0.885	0.052	0.047	0.040
	K-means on FPC Scores	0.909	0.911	0.909	0.038	0.033	0.031
	FunFEM	**0.976**	**0.984**	**0.977**	**0.019**	**0.012**	**0.013**
	FunHDDC	0.863	0.862	0.865	0.056	0.049	0.044

Table 3.5: Summary Table of Average Accuracy and WCMSE from the Clustering Results over 200 Simulations in 4 Different Scenarios under $\sigma = 2.5$ and Number of Clusters $K = 4$.

Table 3.5 provides the same suggestion for the best clustering methods in Table 3.4. In Table 3.5, unsupervised random forest clustering and FunFEM model-based clustering have the highest accuracy and lowest WCMSE among all 6 methods. The overall average accuracy reaches 90% in the scenarios 1 and 2, while that of the two best methods achieves 99%. The overall accuracy reduces by about 3% or 4% in scenarios 3 and 4, while the two best methods can still maintain an accuracy about 98%. In addition, the overall average WCMSE is larger in scenarios 3 and 4 than in scenarios 1 and 2. The WCMSE is about 0.007 for the two best methods in scenarios 1, 2 and 3, and is about 0.015 in scenarios 3 and 4.

To conclude, both Table 3.4 and 3.5 demonstrate that dealing with the clustering methods on curves with within cluster variance varies over time with lower derivations and larger WCMSE. Unsupervised random forest clustering and FunFEM based clustering are the two competitive methods with the best clustering results in terms of sensitivity, precision, accuracy and WCMSE. A simulation study with the same set-up except with a smaller variance as $\sigma = 1.25$ is also conducted. The conclusion show the same trend, and the summary tables for $\sigma = 1.25$ have no obvious difference compared to Table 3.4 and 3.5. Details for the simulation with $\sigma = 1.25$ are given in the Appendix.

Chapter 4

Real Data Application

In this chapter, we apply existing and proposed unsupervised learning methods to the evaluation analysis of U.S. temperature forecast data. The goal of the analysis is to investigate potential covariates correlated to weather prediction performance in the U.S, especially to explore the spatial and time effects in prediction accuracy. The data involved in the analysis come from the Data Expo Case Competition in the Joint General Meeting (JSM) in 2018, which contain 3-year weather forecast and historical measurements records across 113 U.S. cities in 50 states from September 2014 to August 2017[1]. Further details of the data are described in the following section 4.1.

4.1 Data Description

The U.S. temperature forecast data are formed in three parts: forecast weather records, historical weather measurement records and geographical information records[1]. Forecast weather records consist of different measures of weather that the forecast was for over the 3-year period, including minimum temperature, maximum temperature, and the probability of precipitation, and specify the date that was forecast and the date that the forecast was made on. Historical weather records comprise different weather measures in each city, such as maximum and minimum temperature, humidity and sea level pressure, etc. The geographical information of the cities for which the forecast was made is also available. Each city is documented with its corresponding state, geographical coordinates (i.e. longitude and latitude) and airport code (AirPtCd). AirPtCd provides information regarding the airport closest to the city, as well as the place that the historical data was measured. Details of the variables are summarized in Table 4.1.

Variable Category	**Variable Name**	**Description**
Forecast Weather Information	City_Index	The city where the forecasts were made on.
	Forecast_Date	The date that was forecast.
	Forecast_Made_On	The date that the forecast was made on.
	Forecast_Value	Indicate what value is being forecast.
Historical Weather Information	Date	Date that has been forecast.
	Min_TemperatureF	Real minimum temperature measured in Fahrenheit.
	Max_TemperatureF	Real maximum temperature measured in Fahrenheit.
	Mean_TemperatureF	Average of minimum and maximum temperature.
	Mean_DewpointF	
	Mean_Sea_Level_PressureIn	
	Mean_VisibilityMiles	
	Mean_Wind_SpeedMPH	
	PrecipitationIn	
Geographical Information	City	
	State	
	Longitude	
	Latitude	
	AirPtCd	Airport code of the airport closest to the city.

Table 4.1: Description of variables in the dataset

4.2 Objectives

To evaluate the prediction performance, we defined our response variable as the absolute value of the prediction error for the minimum temperature:

$$\varepsilon_t = |T_t^{\text{real}} - T_t^{\text{fore}}|,$$

where T_t^{real} and T_t^{fore} are the real and forecast temperatures at time point t, respectively. According to survey about Americans' interests of weather forest, people are most interested in short-term weather forecasts as it provides direct guidance on planning day-to-day activities ([25]); therefore we only evaluated the overall accuracy of 1-day forecast in this study. Furthermore, with the geographical and temporal information in the collected data, the goals of our analysis are:

1. Explain how prediction performance changes over time.
2. Explore variations in weather forecast accuracy across different geographical locations in the U.S., and identify the most and least predictable regions.

4.3 Exploratory data analysis

Before conducting unsupervised learning methods, we first perform exploratory data analysis (EDA), which helps empirically detect trends in data and plays a foundation for our further studies. The following sub-sections explore variation of prediction error from two different aspects, supported by data summary statistics and plots. These explorations motivate us to find potential methods to explain and model the discovered phenomena in the data.

Seasonal Patterns

The reason for studying the first objective is because time usually plays an important role in determining future climate expectations, so the error that ε_t may also be affected by time and seasons. Intuitively speaking, cold seasons may cause significant uncertainty in forecast guidance and are expected to be less predictable than the warm seasons. This is well illustrated in Figure 4.1, which shows that the performance of weather prediction varies over time from September 2014 to August 2017. The red and blue regions represent Winter (December to February) and Summer (June to August) period, respectively. The variation of ε_t shows periodicity within each year; specifically, the prediction is more variable in Winter compared to Summer.

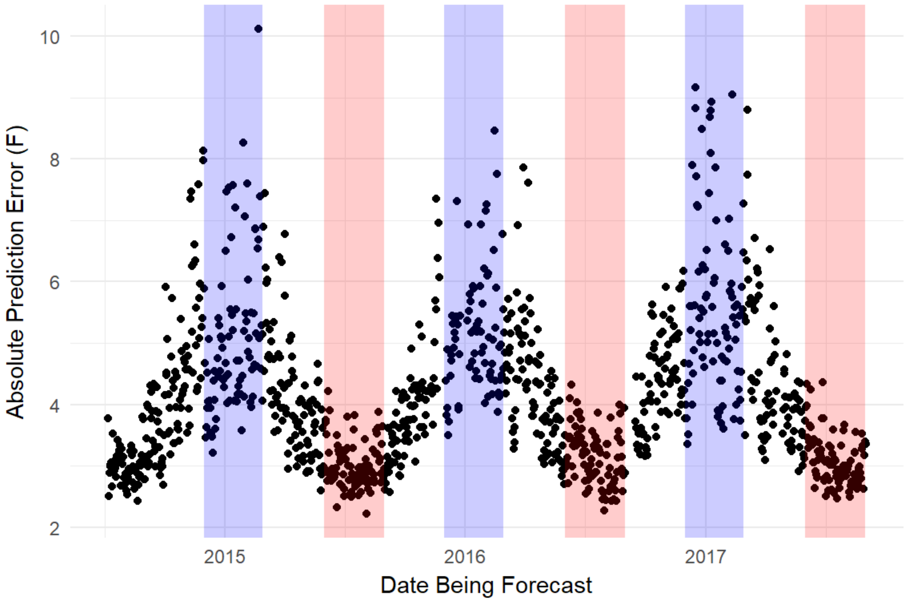

Figure 4.1: Absolute prediction error (F) vs Forecast date

Geographical regional pattern

To investigate our second objective, we generated the following graph to compare the forecast accuracy of different geographical locations across the U.S. We consider the prediction to be accurate if $\varepsilon_t < 4$ (i.e. the prediction error is within 4 Fahrenheit), and the accuracy is evaluated as the percentage of predictions satisfying this condition within each state. More blue represents regions with higher prediction accuracy and less blue represents regions with lower prediction accuracy.

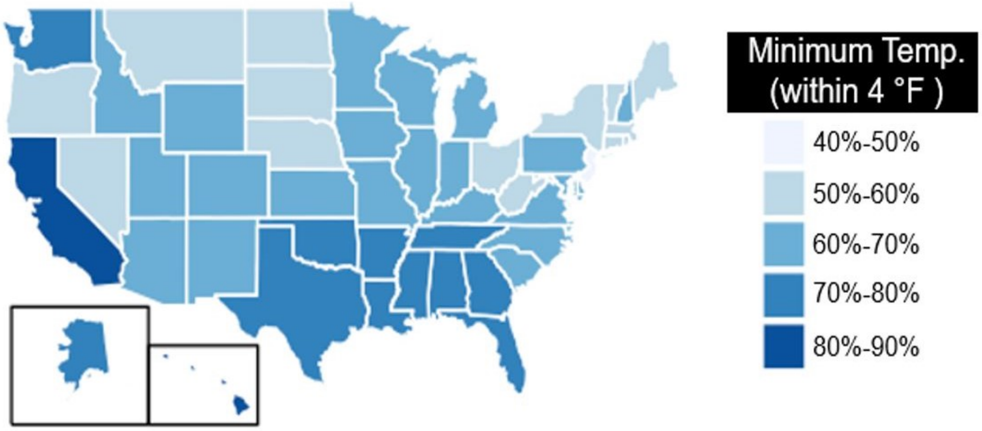

Figure 4.2: Prediction accuracy of minimum temperature (F) for each state

In general, neighbouring states tend to have similar prediction performance. However, for those states which are not close to each other, such as Washington, California, Hawaii and Florida, there is similar prediction accuracy because they can share similar weather conditions with respect to temperature. Therefore, the pattern similarity of prediction performance among the U.S. states may not only relate to geographical locations, but also to the similarity of climate conditions. For example, coastal states with mild climate are more likely to be clustered together and have better forecast performance than the inland states with more extreme weather.

From the result of the exploratory analysis, we suspect that the absolute prediction error of daily minimum temperature is affected by the joint effect of geographical location and climate in weather forecasts, which we refer as the "spatio-climate effect". To illustrate this spatio-climate effect in the further analysis, we utilize the FPCA to investigate the general pattern of the variation of ε_t over time and divide the U.S. into different regions based on weather prediction performance through different clustering methods to. We then identify the most and least predictable U.S. states.

4.4 Results

In this section, the unsupervised analysis result of the U.S. temperature prediction data is presented. We first describe the discrete absolute prediction error time series data ε_t for every state as the average of daily absolute prediction error. Then, ε_t for all 50 U.S. states are transformed into functional data using cubic B-spline. Furthermore, we obtain the overtime pattern of variation across the U.S. through FPCA. Finally, all clustering methods are applied to detect the patterns of ε_t and to group the states with similar ε_t patterns.

4.4.1 Smoothing Splines and FPCA

During the smoothing procedure, we used 17 distinct interior quantile points to divide the 3-year period into 18 time intervals with the same amount of data, so each time interval contains about 2 months of data. Fifty smoothed curves of $\epsilon_i(t), i = 1, ..., 50$ for all U.S. states are obtained and plotted in Figure 4.3.

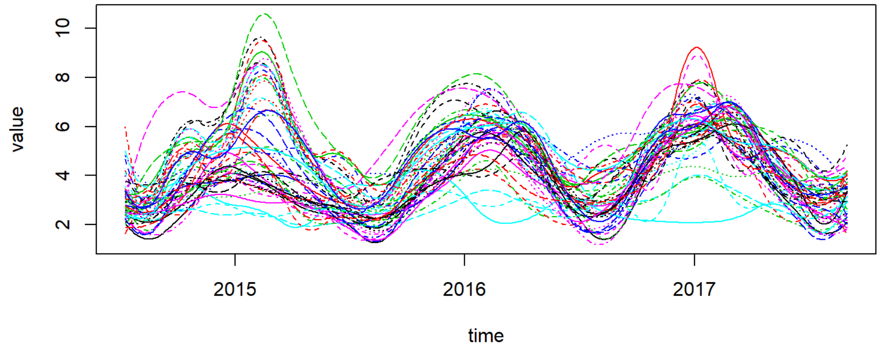

Figure 4.3: Smoothed Curves of Absolute Prediction Error for 50 U.S. States

In Figure 4.3, we observe that the curves vary most during the winter time around the beginning of every year. To further understand the curves variation pattern, we estimate PC functions which explain 90% of the variation through FPCA, which is shown in Figure 4.4.

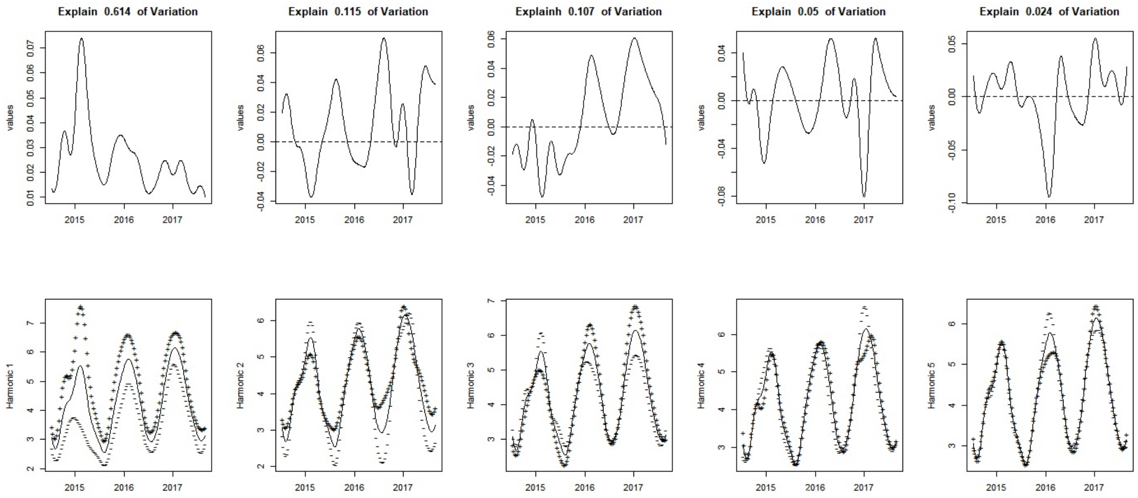

Figure 4.4: First 5 PC Functions of U.S. states Absolute Prediction Error Data

In Figure 4.4, the 5 plots in the first row are the raw eigenfunctions corresponding to the first 5 PCs explaining 91% the variation of the original functional data. The 5 plots in the second row are the mean function $\bar{x}(t)$ plus or minus the same proportion of the corresponding PC's eigenfunction. The curves plotted with plus signs are the new curves produced by the mean function plus a proportion of the corresponding PC function, and the same idea is used to plot the curves with the minus signs. The sign of the PC functions has no meaning, so the interpretation explains only the pattern of the change of the absolute value in the

PC functions. The first 3 PC function explain most of the variation, about 85%, but the fourth and fifth PCs only explain 3%. Therefore, we conclude the main features of the first 3 PC functions.

The first 3 PC functions explain 61.4%, 11.5% and 10.7% of the variation respectively. From the plot of the first PC function explaining most of the variation, we observe that all the PC function values are positive, and three local maximums appear at every mid winter time from 2015 to 2017; moreover, the local maximums are decreasing over years. This phenomenon shows that the absolute prediction error mainly varies at the mid winter time, but such variation decreases over year. Similarly, the second and third PC functions demonstrate that part of the variability exists at the mid summer and mid-winter.

4.4.2 Clustering

The next aim of the study is to find groups of states that have a similar patterns of over-time absolute prediction error and rank the performance of the prediction based on the patterns of the curves. We first determine the number of the clusters from the candidate set $\{2, 3, 4, 5, 6, 7, 8\}$. The result of cluster number selection are stated in Table 4.2.

Algorithm	Criterion	Selected Cluster Number
K-means on B-spline Coefficients	26 Indices in NbClust	3
K-means on FPC Scores		2
FunFEM	BIC	4
	ICL	5
FunHDDC	BIC	2
	ICL	2

Table 4.2: Cluster Number Selection for Smoothed Absolute Prediction Error Curves among 50 U.S. States

The total observed curves number is small (50), which means the average number of curves in each cluster is no more than 25 when the number of clusters no less than 2. Based on the result of the simulation study, the K-means algorithm on feature information usually underestimates the number of clusters, whereas the number selected by the FunFEM algorithm and the BIC criterion is more valid for a small numbers of observations. Therefore, further clustering procedures is conducted with $K = 4$, the cluster number selected by FunFEM and the BIC criterion.

Given $K = 4$, 50 smoothed curves are then clustered into 4 groups through 6 methods: unsupervised random forest clustering on B-spline coefficients or FPC scores, K-means clustering on B-spline coefficients or FPC scores, FunFEM and FunHDDC. To make the curves within each cluster as close as possible, eight different seeds (8, 88, 888, 8888, 88888, 888888, 8888888, 88888888) are tested for each method. Then, the clustering result with the smallest integral of the average within-cluster standard derivation (SD) function is considered as the final clustering result. Table 4.3 shows the smallest integral value of the average within-cluster SD function that each clustering method obtained.

Algorithm	Integral of Average Within-cluster SD Function
URF on B-spline Coefficients	739.3
URF on FPC Scores	717.6
K-means on B-spline Coefficients	692.7
K-means on FPC Scores	710.1
FunFEM with BIC criterion	714.1
FunFEM with ICL criterion	714.1
FunHDDC	Diverge on both BIC and ICL criterion

Table 4.3: Integral Value of Average Within-cluster SD Function from the Result of Each Clustering Method

For these data, the FunHDDC algorithm fails due to divergence of the EM algorithm. The clusters obtained by K-means algorithm on B-spline coefficients achieves the smallest integral value of the average within-cluster SD function, while URF clustering on B-spline coefficients achieve the largest integral value. In the view of minimizing overall within-cluster variance, the clustering result from the K-means algorithm on B-spline coefficients achieves the best result, while the result from URF clustering on B-spline coefficients performs most poorly. However, some clusters may contain curves that have large variance, so determining the final clusters by comparing the overall within-cluster variance is insufficient. Thus, we should also look into the patterns of the curves to determine the quality of the cluster results. Figure 4.5 and 4.6 plot the 4 obtained clusters from k-means and URF clustering respectively.

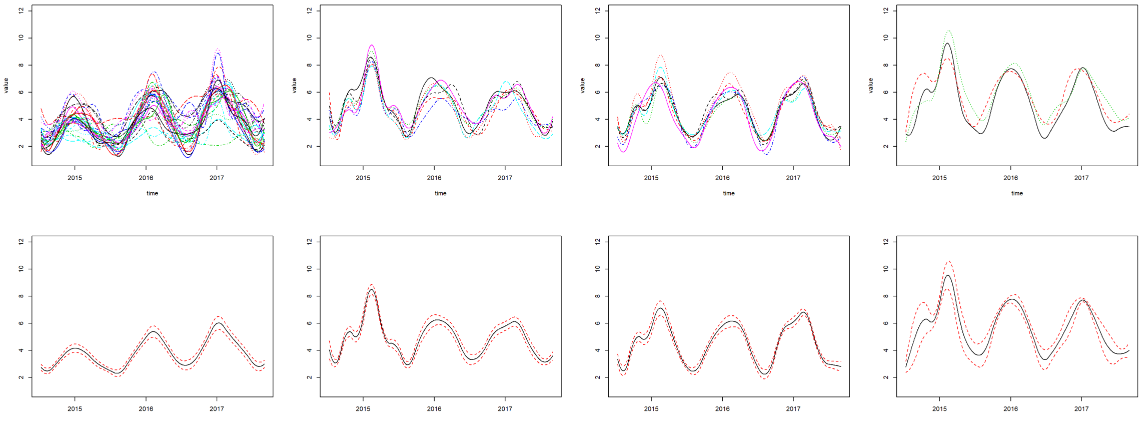

Figure 4.5: Cluster Result from K-means Clustering on B-spline Coefficients

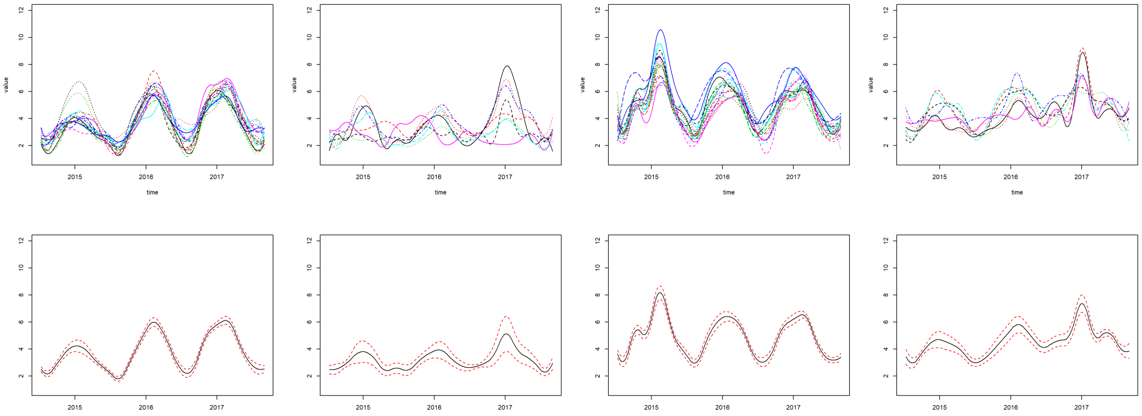

Figure 4.6: Cluster Result from Unsupervised Random Forest Clustering on B-spline Coefficients

In both Figure 4.5 and 4.6, the plots in the first row visualize smoothed curves within each cluster, and the plots in the second row are the 95% pointwise confidence interval of each cluster. Through the comparison of the two figures, we can observe more obvious cluster patterns from URF than from K-means. Figure 4.5 shows that in the result of K-means algorithm, the third and the fourth clusters are very similar, but some curves in the first cluster are visibly flatter than the other curves. This is probably due to misclustering. However, the URF method does detect the flatter curves and produce the second cluster in Figure 4.6; also, it discovers similar first and third clusters as K-means algorithm and discovers the fourth cluster whose pattern is an increasing trend over a year. Consequently, we prefer the clustering result from URF more than the result from K-means. However, some potential misclustered curves still exist in the URF clustering result, such that one curve in cluster 1 has a much higher peak than the other curves in 2015. Following this logic, we should look into the clustering the results from the other four methods to determine the most reasonable clusters.

However, the comparison of the cluster curves plot for four more methods simultaneously are complicated. Also, we remark that the goal of this study is not only to find the main pattern of each cluster but also to find both the geo-correlation of these clusters. Therefore, one shortage of the cluster curve plots like Figures 4.5 and 4.6 is that we cannot visualize the geographic information about the curves. This means we cannot confirm whether the similar patterns between two clustering methods is due to similar states. In this case, in Figures 4.7 to 4.12, the U.S. maps partitioned by different clustering methods are shown to display the result as the complements of Figures 4.5 and 4.6.

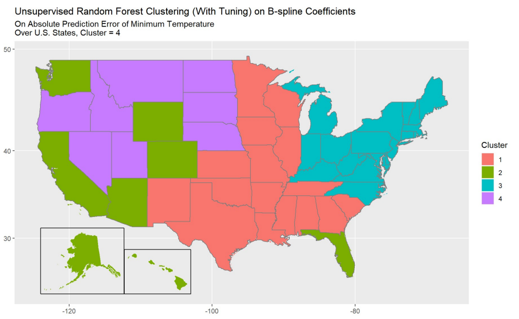
Unsupervised Random Forest Clustering (With Tuning) on B-spline Coefficients
On Absolute Prediction Error of Minimum Temperature
Over U.S. States, Cluster = 4
50
40
30
-120
-100
-80
Cluster
1
2
3
4

Figure 4.7: U.S. Map Partitioned by Unsupervised Random Forest Clustering on B-spline Coefficients

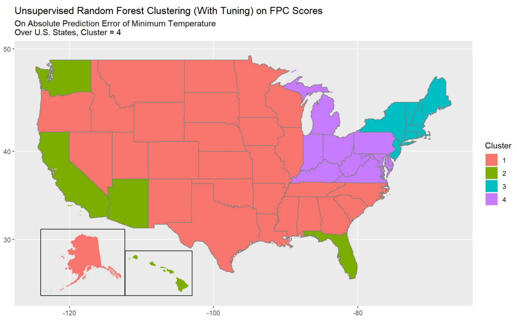
Unsupervised Random Forest Clustering (With Tuning) on FPC Scores
On Absolute Prediction Error of Minimum Temperature
Over U.S. States, Cluster = 4
50
40
30
-120
-100
-80
Cluster
1
2
3
4

Figure 4.8: U.S. Map Partitioned by Unsupervised Random Forest Clustering on FPC Scores

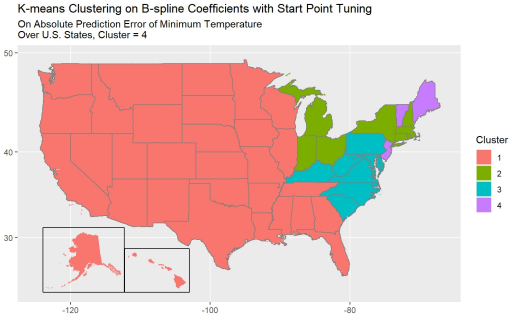
K-means Clustering on B-spline Coefficients with Start Point Tuning
On Absolute Prediction Error of Minimum Temperature
Over U.S. States, Cluster = 4
50
40
30
-120
-100
-80
Cluster
1
2
3
4

Figure 4.9: U.S. Map Partitioned by K-means Clustering on B-spline Coefficients

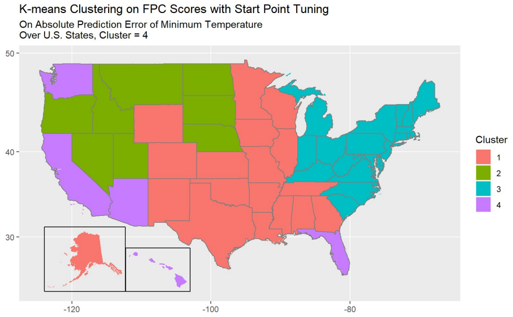
K-means Clustering on FPC Scores with Start Point Tuning
On Absolute Prediction Error of Minimum Temperature
Over U.S. States, Cluster = 4
50
40
30
-120
-100
-80
Cluster
1
2
3
4

Figure 4.10: U.S. Map Partitioned by K-means Clustering on FPC Scores

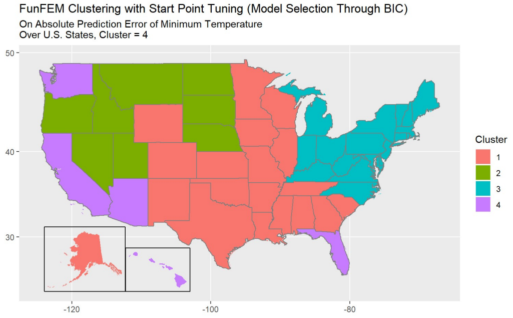
FunFEM Clustering with Start Point Tuning (Model Selection Through BIC)
On Absolute Prediction Error of Minimum Temperature
Over U.S. States, Cluster = 4
50
40
30
-120
-100
-80
Cluster
1
2
3
4

Figure 4.11: U.S. Map Partitioned by FunFEM (Model Selected by BIC)

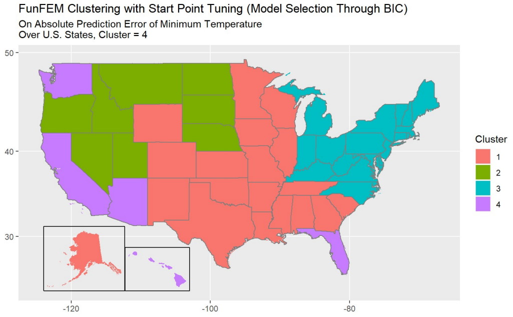
FunFEM Clustering with Start Point Tuning (Model Selection Through BIC)
On Absolute Prediction Error of Minimum Temperature
Over U.S. States, Cluster = 4
50
40
30
-120
-100
-80
Cluster
1
2
3
4

Figure 4.12: U.S. Map Partitioned by FunFEM (Model Selected by ICL)

The clustering results from FunFEM using either BIC or ICL are exactly the same, and they are very similar to the results of URF clustering on B-spline coefficients and K-means clustering on FPC scores. FunFEM clusters Alaska, Wyoming and Colorado in cluster 1 while URF clustering on B-spline coefficients assign them to cluster 2. Besides, FunFEM clusters the South Carolina to cluster 1 while the K-means on FPC scores assigns it to cluster 3. However, the partitions of the U.S. map produced by k-means on B-spline coefficients and by URF clustering on FPC scores are very different from the ones by FunFEM and other two methods. Based on the comparison of the maps and the previous conclusion that the partitions by URF on B-spline coeffcients is better than that by K-means on B-spline coefficients, we determine the final cluster result by firstly electing one result from the FunFEM, K-means clustering on FPC scores and URF clustering on B-spline coefficients, and then selecting the final result by comparing the result from URF clustering from FPC scores and the selected one from the previous step.

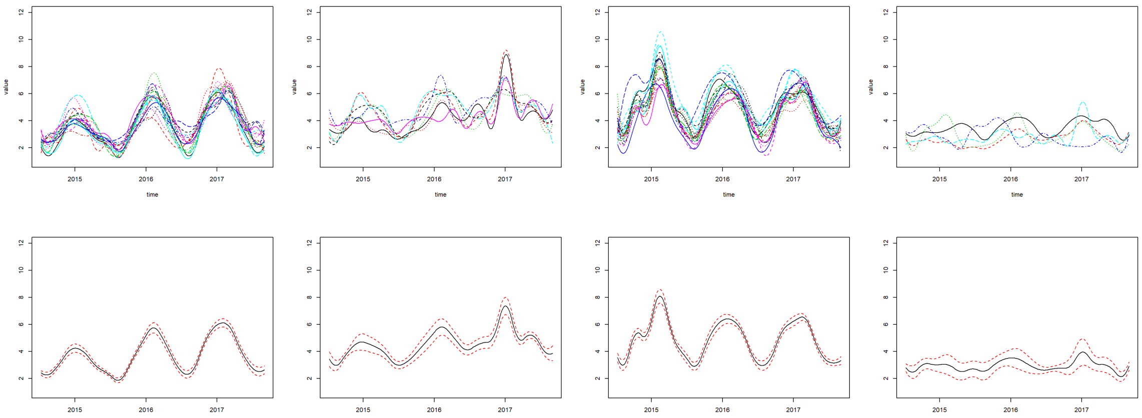

Figure 4.13: Cluster Result from K-means Clustering on FPC Scores

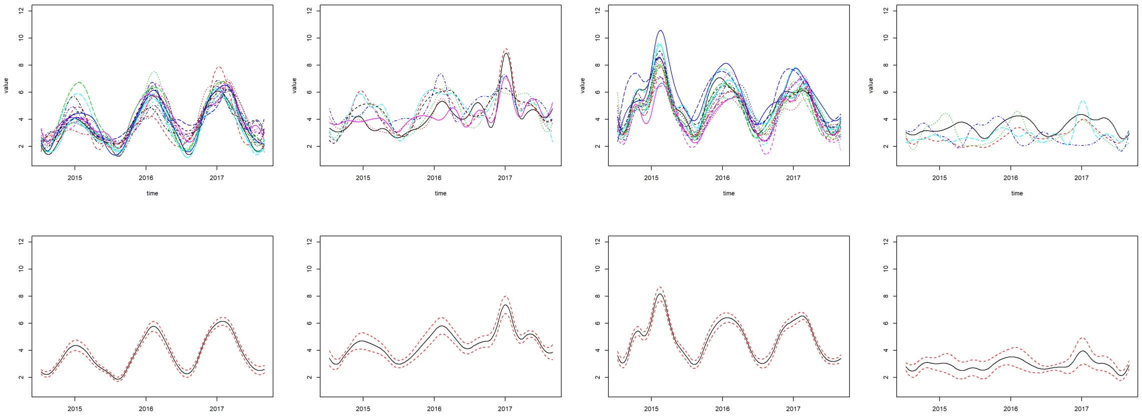

Figure 4.14: Cluster Result from FunFEM

Figure 4.13 and 4.14 are the curve plots of the clustering result for K-means clustering on FPC Scores and FunFEM. Compared to the clustering result of URF clustering on B-spline plotted in Figure 4.6, the cluster results are better because there are fewer potential misclustered curves in cluster 1, and the states with flatter pattern in cluster 4 are detected. The difference between Figure 4.13 and 4.14 is not obvious, but we choose the result from K-means on FPC scores as it has smaller average within-cluster variance.

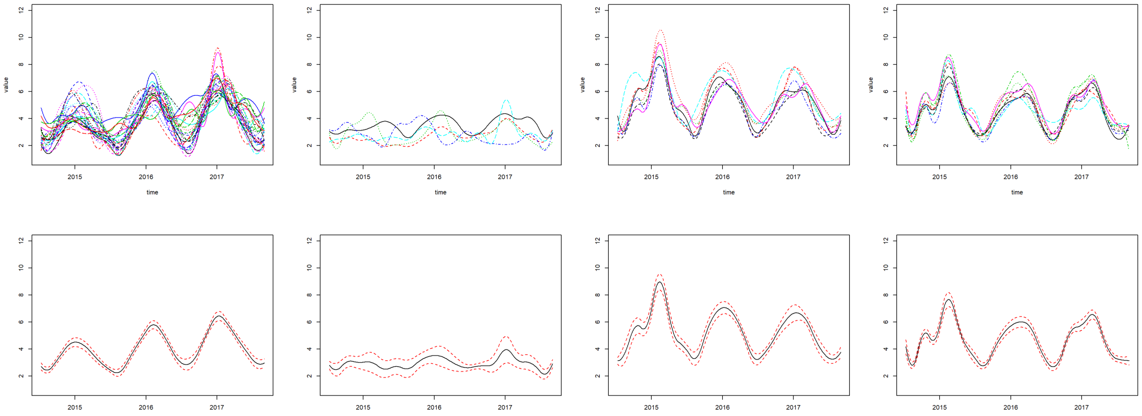

Figure 4.15: Cluster Result from URF Clustering on FPC Scores

By comparing the clustering result plotted in Figure 4.14 and 4.15, we suggest the results from K-means on FPC scores because, in the results from URF clustering on FPC scores, cluster 1 in Figure 4.15 has more potential misclustered curves with different patterns such as apparent increasing trends over time or higher peaks in the Winter time.

Therefore, our final selected result is from K-means on FPC scores who are plotted in Figure 4.10 and 4.13. The results divide the U.S. into the following four regions based on the pattern overtime of the absolute prediction error of minimum temperature, which are

1. states near the west coastline and east south coastline,
2. states in middle inlet, west of Great Lakes, south of U.S and Alaska,
3. states in the north of U.S,
4. states near the east and northeast coastline.

Finally, we ranked the prediction accuracy of four regions through ordering the integration of the mean curves from smallest to largest. The higher ranking represents that the states in that region are more predictable for cold temperatures.

Rank	**Cluster**	**Overall Integral**	**Representative States**
1	Cluster 4	3397.3	California, Florida
2	Cluster 1	4284.2	Alaska, Texas
3	Cluster 2	5248.2	Oregan, North Dakota
4	Cluster 3	5630.1	Pennsylvania, North Carolina

Table 4.4: Cluster Ranking through Mean Curve Integration

Chapter 5

Conclusion and Discussion

In this thesis, the two main contributions are extending unsupervised random forest clustering (URF clustering) to functional data and providing a comprehensive simulation study for evaluating several clustering methodologies for functional data under different scenarios. The proposed functional-case URF clustering is a filtering methods, which transfer the data from a functional format into a multivariate format by replacing observed curves with their feature information, such as B-spline or FPC scores. A drawback of this proposed method is that it cannot select the number of clusters because the parameters in the random forest can only be tuned with a known number of cluster.

Therefore, our simulation study is formed in two component. A study on the accuracy of selecting the number of clusters and an evaluation study of the clustering results when the number of clusters is known. In the simulation study, 6 clustering methods are involved. These methods are URF forest clustering on B-spline coefficients or FPC scores, K-means clustering on B-spline coefficients or FPC scores, FunFEM and FunHDDC. The first component of the simulation study shows that the model-based clustering method FunFEM has excellent performance for detecting the correct cluster number when the number of curves in each cluster is small, but no clustering method has outstanding performance when the number of the curves is large. In the second component of the simulation study, URF clustering methods and FunFEM algorithm are found to be two competitive methods with the best performance.

Finally, we apply the 6 clustering methods in the simulation study to the analysis of U.S. weather forest data. With the interest of the patterns of the prediction error of minimum temperature in 50 U.S. states, we firstly detect 4 clusters using FunFEM. After comparing the clustering results among different methods, we consider that K-means clustering on FPC scores has the most appropriate clustering result. From the result, we find that weather in states near the west coastline and the east south coastline are most predictable, while the states near the east and northeast coastline are least predictable.

By comparing the simulation study and real data application, we can see that during the analysis of real data, the procedure of selecting an appropriate clustering result can be complicated and subjective. The reason for this problem is that clustering is a type of unsupervised learning that is usually used in the exploratory analysis. Unlike the simulation study, we usually do not know the number of clusters or which cluster the curves belong to. Therefore, we may use multiple techniques or visualizations to select clustering results such that the patterns between different clusters are as different as possible, and the curves within a cluster are close to each other. The selection procedure may be very subjective, but because these clustering methods are for exploratory analysis, the clustering result is meaningful once it can provide useful information or motivation for the further analysis.

Bibliography

[1] ASA Statistical Computing and Statistical Graphics Sections data expo 2018. `https://community.amstat.org/stat-computing/data-expo/data-expo-2018`. Accessed: 2018-09-30.

[2] Christophe Abraham, Pierre-André Cornillon, ERIC Matzner-Løber, and Nicolas Molinari. Unsupervised curve clustering using B-splines. *Scandinavian journal of statistics*, 30(3):581–595, 2003.

[3] Giada Adelfio, Marcello Chiodi, Antonino D'Alessandro, and Dario Luzio. FPCA algorithm for waveform clustering. *Journal of Communication and Computer*, 8(6):494–502, 2011.

[4] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering–a decade review. *Information Systems*, 53:16–38, 2015.

[5] Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[6] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Journal of computational biology*, 6(3-4):281–297, 1999.

[7] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[8] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence*, 22(7):719–725, 2000.

[9] Charles Bouveyron, Etienne Côme, Julien Jacques, et al. The discriminative functional mixture model for a comparative analysis of bike sharing systems. *The Annals of Applied Statistics*, 9(4):1726–1760, 2015.

[10] L Breiman and A Cutler. Random forest-manual. *Online: http://www. stat. berkeley. edu/˜ breiman/RandomForests/cc_manual. htm*, 2004.

[11] Joseph H Casola and John M Wallace. Identifying weather regimes in the wintertime 500-hPa geopotential height field for the Pacific–North American sector using a limited-contour clustering technique. *Journal of Applied Meteorology and Climatology*, 46(10):1619–1630, 2007.

[12] Raymond B Cattell. The scree test for the number of factors. *Multivariate behavioral research*, 1(2):245–276, 1966.

[13] Malika Charrad, Nadia Ghazzali, Véronique Boiteau, and Azam Niknafs. NbClust package: finding the relevant number of clusters in a dataset. *UseR! 2012*, 2012.

[14] Haskell Brooks Curry and Isaac J Schoenberg. On Pólya frequency functions IV: the fundamental spline functions and their limits. *Journal dâĂŹanalyse mathématique*, 17(1):71–107, 1966.

[15] Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[16] Frédéric Ferraty and Philippe Vieu. *Nonparametric functional data analysis: theory and practice.* Springer Science & Business Media, 2006.

[17] Chris Fraley and Adrian E Raftery. Mclust: Software for model-based cluster analysis. *Journal of classification*, 16(2):297–306, 1999.

[18] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.

[19] Luis Angel Garcia-Escudero and Alfonso Gordaliza. A proposal for robust curve clustering. *Journal of classification*, 22(2):185–201, 2005.

[20] John A Hartigan and Manchek A Wong. Algorithm as 136: A *k*-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[21] Alan Julian Izenman. Modern multivariate statistical techniques. *Regression, classification and manifold learning*, 2008.

[22] Julien Jacques and Cristian Preda. Functional data clustering: a survey. *Advances in Data Analysis and Classification*, 8(3):231–255, Sep 2014.

[23] Gareth M James and Catherine A Sugar. Clustering for sparsely sampled functional data. *Journal of the American Statistical Association*, 98(462):397–408, 2003.

[24] Godfrey N Lance and William Thomas Williams. A general theory of classificatory sorting strategies: 1. hierarchical systems. *The computer journal*, 9(4):373–380, 1967.

[25] Jeffrey K Lazo, Rebecca E Morss, and Julie L Demuth. 300 billion served: Sources, perceptions, uses, and values of weather forecasts. *Bulletin of the American Meteorological Society*, 90(6):785–798, 2009.

[26] Krzysztof Loska and Danuta Wiechuła. Application of principal component analysis for the estimation of source of heavy metal contamination in surface sediments from the Rybnik Reservoir. *Chemosphere*, 51(8):723–733, 2003.

[27] James Ramsay, Giles Hooker, and Spencer Graves. *Functional data analysis with R and MATLAB.* Springer Science & Business Media, 2009.

[28] James O Ramsay and Bernard W Silverman. *Applied functional data analysis: methods and case studies.* Springer, 2007.

[29] Fabrice Rossi, Brieuc Conan-Guez, and Aïcha El Golli. Clustering functional data with the SOM algorithm. In *ESANN*, pages 305–312, 2004.

[30] Alexis Sardá-Espinosa. Comparing time-series clustering algorithms in R using the dtwclust package. *R package vignette*, 12, 2017.

[31] Amandine Schmutz, Julien Jacques, Charles Bouveyron, Laurence Cheze, and Pauline Martin. Clustering multivariate functional data in group-specific functional subspaces. 2018.

[32] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[33] Tao Shi and Steve Horvath. Unsupervised learning with random forest predictors. *Journal of Computational and Graphical Statistics*, 15(1):118–138, 2006.

[34] Robert Tibshirani and Guenther Walther. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528, 2005.

[35] Isabel F Trigo, Trevor D Davies, and Grant R Bigg. Objective climatology of cyclones in the Mediterranean region. *Journal of Climate*, 12(6):1685–1696, 1999.

[36] Ka Yee Yeung and Walter L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.

[37] Qifeng Zhou, Wencai Hong, Linkai Luo, and Fan Yang. Gene selection using random forest and proximity differences criterion on DNA microarray data. *Journal of Convergence Information Technology*, 5(6):161–170, 2010.

Appendix A

Summary Tables of Simulation Study under $\sigma = 1.25$ and Number of Clusters $K = 4$.

Table A.1: Summary Table of Average Sensitivity and Precision from the Clustering Results over 200 Simulations in 4 Different Scenarios under $\sigma = 1.25$ and Number of Clusters $K = 4$.

		Sensitivity			Precision		
		(Curves Number/Cluster)			(Curves Number/Cluster)		
Scenario	Methods	20	50	100	20	50	100
1	URF on B-spline Coefficients	**1**	**1**	**1**	**1**	**1**	**1**
	URF on FPC Scores	**0.998**	**1**	**1**	**0.998**	**1**	**1**
	K-means on B-spline Coefficients	0.938	0.940	0.940	0.812	0.819	0.821
	K-means on FPC Scores	0.937	0.937	0.936	0.802	0.813	0.817
	FunFEM	**0.998**	**0.999**	**0.998**	**0.994**	**0.998**	**0.993**
	FunHDDC	0.906	0.940	0.991	0.712	0.811	0.973
2	URF on B-spline Coefficients	**1**	**1**	**0.999**	**1**	**1**	**0.999**
	URF on FPC Scores	**0.998**	**1**	**1**	**0.998**	**1**	**1**
	K-means on B-spline Coefficients	0.939	0.932	0.993	0.810	0.793	0.801
	K-means on FPC Scores	0.936	0.928	0.932	0.800	0.780	0.793
	FunFEM	**0.992**	**0.990**	**0.991**	0.974	0.970	0.969
	FunHDDC	0.969	0.994	0.998	0.905	0.983	0.992
3	URF on B-spline Coefficients	**1**	**1**	**1**	**1**	**1**	**1**
	URF on FPC Scores	**0.996**	**0.998**	**0.999**	**0.996**	**0.996**	**0.999**
	K-means on B-spline Coefficients	0.938	0.935	0.940	0.814	0.808	0.826
	K-means on FPC Scores	0.942	0.941	0.950	0.825	0.828	0.855
	FunFEM	**0.998**	**0.996**	**0.993**	**0.994**	**0.988**	**0.979**
	FunHDDC	0.903	0.899	0.899	0.711	0.700	0.695
4	URF on B-spline Coefficients	**0.999**	**0.999**	**.996**	**0.999**	**0.999**	**0.996**
	URF on FPC Scores	**0.994**	**0.997**	**0.999**	**0.993**	**0.997**	**0.999**
	K-means on B-spline Coefficients	0.940	0.942	0.933	0.814	0.827	0.802
	K-means on FPC Scores	0.937	0.930	0.946	0.806	0.789	0.839
	FunFEM	0.995	0.994	0.996	0.983	0.983	0.988
	FunHDDC	0.959	0.985	0.956	0.878	0.875	0.869

Table A.2: Summary Table of Average Accuracy and WCMSE from the Clustering Results over 200 Simulations in 4 Different Scenarios under $\sigma = 1.25$ and Number of Clusters $K = 4$.

		Accuracy			WCMSE		
		(Curves Number/Cluster)			(Curves Number/Cluster)		
Scenario	Methods	20	50	100	20	50	100
1	URF on Bspline Coefficients	**1**	**1**	**1**	**0.006**	**0.006**	**0.006**
	URF on FPC Scores	**0.999**	**1**	**1**	**0.006**	**0.006**	**0.006**
	K-means on Bspline Coefficients	0.919	0.921	0.920	0.034	0.034	0.033
	K-means on FPC Scores	0.916	0.919	0.922	0.034	0.031	0.031
	FunFEM	**0.997**	**0.999**	**0.997**	**0.007**	**0.006**	**0.007**
	FunHDDC	0.879	0.921	0.988	0.051	0.039	0.011
2	URF on Bspline Coefficients	**1**	**1**	**0.999**	**0.007**	**0.006**	**0.006**
	URF on FPC Scores	**0.999**	**1**	**1**	**0.007**	**0.006**	**0.006**
	K-means on Bspline Coefficients	0.919	0.905	0.910	0.036	0.039	0.038
	K-means on FPC Scores	0.916	0.902	0.906	0.034	0.040	0.038
	FunFEM	0.989	0.987	0.987	0.011	0.010	0.011
	FunHDDC	0.961	0.993	0.997	0.020	0.009	0.007
3	URF on Bspline Coefficients	**1**	**1**	**1**	**0.007**	**0.006**	**0.006**
	URF on FPC Scores	**0.998**	**0.999**	**1**	**0.007**	**0.006**	**0.006**
	K-means on Bspline Coefficients	0.922	0.919	0.926	0.033	0.033	0.029
	K-means on FPC Scores	0.927	0.927	0.938	0.026	0.025	0.021
	FunFEM	**0.997**	**0.995**	**0.991**	**0.008**	**0.007**	**0.008**
	FunHDDC	0.879	0.873	0.871	0.053	0.057	0.059
4	URF on Bspline Coefficients	**0.999**	**0.999**	**0.998**	**0.007**	**0.006**	**0.006**
	URF on FPC Scores	**0.997**	**0.998**	**0.999**	**0.007**	**0.006**	**0.006**
	K-means on Bspline Coefficients	0.920	0.926	0.915	0.034	0.031	0.033
	K-means on FPC Scores	0.919	0.907	0.928	0.032	0.037	0.028
	FunFEM	**0.993**	**0.993**	**0.995**	**0.010**	**0.009**	**0.007**
	FunHDDC	0.948	0.947	0.944	0.024	0.024	0.024