Spectral Clustering and Downsampling-Based Model Selection For Functional Data

Maryam Al Alawi

Submitted in fulfilment of the requirements for the

Degree of Doctor of Philosophy

School of Mathematics and Statistics

College of Science and Engineering

University of Glasgow

November 2021

Abstract

Functional data analysis is a growing field of research and has been employed in a wide range of applications ranging from genetics in biology to stock markets in economics. A crucial but challenging problem is clustering of functional data. In this thesis, we review the main contributions in this field and discuss the strengthens and weaknesses of the different clustering functional data approaches. We propose a new framework for clustering functional data and a new paradigm for model selection that is specifically designed for functional data, which are designed to address many of the weaknesses of existing techniques. Our clustering framework is based on first reducing the infinite dimensional space of functional data to a finite dimensional space by smoothing and basis expansion. Then we implement the spectral clustering approach by designing a new distance measure which has the flexibility of using the distance between the original trajectories and/or their derivatives. In addition, we develop a new model selection criterion, by introducing a technique called 'downsampling', which allows us to create lower resolution replicates of the observed curves. These replicates can then be used to examine the clustering stability of the existing clustering functional data approaches and select the optimal number of clusters. Further, we combine the two proposed techniques to develop an integrated clustering framework to estimate the number of clusters inherently and accordingly cluster the functional data. An extensive simulation study with existing clustering functional data methods show a superior performance of our clustering framework and reliable results of the proposed model selection criteria. The usefulness of these new approaches is also illustrated through applications to real data.

Declaration

I declare that, all the work presents in this thesis has been done by myself under the supervision of Dr. Surajit Ray and Dr. Mayetri Gupta, except where otherwise stated. This thesis represent work completed, between 2017 and 2021, in the School of Mathematics and Statistics at the University of Glasgow. None of the work described has been submitted to any other university or institute.

Acknowledgements

Thank you Allah for giving me the strength, patience and blessings during this journey.

I would like to express my deepest gratitude to my supervisors Surajit Ray and Mayetri Gupta for providing me with invaluable guidance and continuous support throughout this study. Without them, this work would have never been accomplished. I would also like to acknowledge Sultan Qaboos University, Sultanate of Oman for funding my PhD.

I am grateful to all the people in the School of Mathematics and Statistics for their friendship and assistance during my time in Glasgow. I am also grateful to my family in Oman for their continuous encouragement and for believing in me.

A big thank you will go to my dear husband and my lovely kids. Thank you for being next to me all the time, thank you for always making me smile and keeping me positive, and thank you for your endless love.

To my husband Hamed and my children Mohammed and Reem.

Contents

List of Figures

List of Tables

Abbreviations

FDA	Functional Data Analysis
GCV	Generalized Cross Validation
FPCA	Functional Principal Component Analysis
CFD	Clustering Functional Data
SC	Spectral Clustering
CCR	Correct Classification Rate
ARI	Adjusted Rand Index
FunHDDC	Functional High Dimensional Data Clustering
FD-Kmeans	Non-parametric Functional Data Kmeans Clustering
B-splines-Km	Two-stage B-splines smoothing - Kmeans Clustering
FPCA-mbc	Functional Principal Component Analysis Model-Based Clustering
FSC-S(D_o)	Functional Spectral Clustering - Original Curves Distance-Based
FSC-S(D_1)	Functional Spectral Clustering - First Derivatives Distance-Based
FSC-S(D_2)	Functional Spectral Clustering - Second Derivatives Distance-Based
DSC	Downsampling Criteria
FSC-DSC	Functional Spectral Clustering - Downsampling Criteria
AHP	Average House Prices dataset

Chapter 1

Introduction

1.1 Overview

Functional data analysis is a branch of statistics that analyses information on functions or curves, widely known as FDA. The functions represent repeated observations of the same process taken over some continuum such as time or space and they take the form of smooth curves. The term FDA was first introduced by Ramsay and Dalzell (1991), then Ramsay and Silverman (2005) defined its varied contexts and provided techniques for analysing functional data with applications on real-life datasets, besides developing the R package `fda` (Ramsay et al., 2014). According to Ramsay and Silverman (2005), the definition of FDA is based on considering each curve as an individual entity instead of a sequence of individual observations along the curve. Therefore, it simplifies the representation of the important features of the data and the exploration of data pattern over time (or other continuum) with continuity, which in turn provides more information about the variation in the data. In addition, FDA can handle sparse data or irregularly spaced data, and it can estimate the derivatives of functions or other properties of curves for further analysis. Ramsay and Silverman (2005) have added that FDA techniques can be expanded to explore curve registration that transforms curves by transforming their arguments for functional data that display phase and amplitude variation. Due to the flexibility of FDA, it has been widely used in various applications and fields such as medical and biological sciences, environmental sciences, social sciences, and economics. Thus, it is of interest to expand FDA approaches with

the potential for many significant applications across varied research fields.

In general, functional data analysis can be viewed as a natural extension to multivariate data analysis, and therefore many standard multivariate analysis approaches have been expanded and redesigned to accommodate functional data. On the other hand, several new approaches have been developed specifically for functional data and are equivalent to some existing multivariate data analysis approaches. This thesis will primarily focus on clustering functional data. Along with exploring existing approaches, which are extensions of multivariate clustering approaches, we will be proposing new approaches which are specifically designed for functional data.

Clustering Functional Data (CFD) is a crucial step for data exploration with the aim of building homogeneous groups of curves that show similar features and patterns. Identifying particular clusters in the data will make further analysis more consistent. However due to the features of functional data, clustering functional data is in general a difficult task. The infinite dimensional space of functional data, and the lack of a clear definition of probability distributions on functional data are the main reasons multivariate data analysis techniques cannot be readily applied to clustering functional data. Nevertheless, several studies discussed CFD and proposed different approaches. Jacques and Preda (2014a) reviewed the main contributions to CFD, and found that the initial and the most popular approach is based on reducing the infinite dimensional space of the data to a finite dimensional space, then applying standard multivariate clustering approaches. A second approach is based on defining distance measures specific to functional data, then applying existing non-parametric clustering approaches such as k-means. A third approach is a model-based approach that assumes the functional data come from a mixture of distributions. Many of the proposed approaches obtained reasonable clustering results when applied to the specific application they were developed for. However, they are not generalizable. Further, many of these methods are computationally intensive.

On the other hand, there are several techniques for clustering multivariate data, that have showed good performance but have not been tested extensively on functional data. One of these

techniques is the spectral clustering algorithm that has been successfully used to cluster various data types including complex high dimensional data (Donath and Hoffman, 1973; Fiedler, 1973). The spectral clustering algorithm is considered a flexible method and does not require any strong distributional assumptions about the data. These features make it an ideal candidate for building a clustering framework that is specific for functional data. The newly developed framework will combine the flexibility of spectral clustering and the merits of functional data, which will allow us to develop a robust functional spectral clustering approach for clustering functional data.

In general a common problem that all clustering algorithms, even for multivariate or univariate data, face, is the choice of the appropriate number of clusters. A number of approaches have been developed to determine the number of clusters in CFD literature. Well known approaches in the context of multivariate data clustering, such as the AIC and BIC have been used for selecting the number of clusters for CFD methods (Bouveyron and Jacques, 2011; Same et al., 2011; Giacofci et al., 2013). Beyond the AIC and BIC, more specific criteria have also been introduced in the literature for selecting the number of clusters in the context of CFD. For instance, Sugar and James (2003) suggested to use the averaged Mahalanobis distance between the basis expansion coefficients and their closest cluster centre, to choose the number of clusters. Most Bayesian models for CFD define a framework in which the number of clusters can be directly estimated from the data based on the Dirichlet Process Prior or DPP, (Ray and Mallick, 2006; Suarez et al., 2016; Zhang et al., 2015; Scarpa and Dunson, 2009). Many of the proposed criteria cannot be generalized to existing CFD approaches. Thus, choosing the appropriate number of clusters in CFD is still an open research question. A technique that is gaining more attention recently is based on the clustering stability concept. It is a different philosophy from techniques such as the AIC and BIC in that it does not identify the clustering but it examines the stability of the corresponding clustering results (Von Luxburg et al., 2010). For instance, detecting the same clustering structure of several data sets that were generated from the same model would indicate that the used clustering method shows clustering stability. A number of non-parametric multivariate clustering approaches proposed the use of clustering stability as a model selection

technique (Ben-Hur et al., 2001; Lange et al., 2004; Ben-David et al., 2006). However, there exist very limited applications of clustering stability in the CFD context (Jacques and Preda, 2014a). Therefore, we aim to build a model selection criterion that is based on clustering stability for functional data. The purpose of the criterion is to provide a general procedure that is compatible with existing CFD approaches.

Two separate problems are tackled within this thesis; the first is to develop a framework for clustering functional data that uses the original curves as well as their derivatives, and the second is to develop a model selection criterion for functional data that relies on the clustering stability philosophy, and can be applied to different CFD approaches. In the first task, we aim to make the clustering framework flexible enough to exploit higher order features of curves including the derivatives, which can inherently cluster functional data with phase and amplitude variation without explicitly modelling these variations. In the second task, we introduce a new technique called 'downsampling', which is designed to create lower resolution replicates of a functional dataset based on a designed sampling scheme specific for functional data. The resulting lower resolution replicates will be used to evaluate the clustering stability of a number of CFD approaches. Subsequently, we will develop an integrated framework that will combine the downsampling technique with the proposed functional spectral clustering approach. The advantage of the integrated clustering framework over the initial proposed approach is the ability to estimate the number of clusters within the algorithm based on the downsampling technique. The three proposed approaches will be first explained and demonstrated using the Berkeley growth data (Ramsay and Silverman, 2005). Then, their effectiveness will be examined through comprehensive simulation studies and finally the approaches will be applied to a real-life dataset obtained from (http://statistics.gov.scot/data/house-sales-prices). The aims and objectives of this thesis can be summarized as follows;

- Main objectives:
  - to develop a new framework for clustering functional data based on spectral clustering.

  - to develop a new model selection criterion for choosing the appropriate number of clusters.

- Minor objectives:

  - to summarize the existing clustering functional data techniques.
  - to compare the performance of some chosen CFD methods among the existing methods with our proposed clustering approach.
  - to explore the advantage of using information from the first derivatives and the second derivatives in defining the clustering structure of the functional data.
  - to investigate the performance of the proposed methods on real-life datasets.
  - to build a comprehensive simulation scheme for functional data clustering.

1.2 Thesis Outline

This thesis is divided into a total of nine chapters. The remainder of the thesis is structured as follows:

Chapter 2 provides an overview of functional data analysis, and discusses its main techniques. The chapter starts with defining the general structure and statistics of FDA and its main features. It introduces some commonly used functional datasets, the Canadian weather data and the Berkeley growth data that will be used throughout the thesis to demonstrate our newly developed techniques. The important aspects of FDA, smoothing models and basis expansions are detailed through equations and examples. A brief overview of exploratory functional data analysis is also given, along with its applications. In addition, the chapter looks at different contexts of functional data, such as data with phase and amplitude variation, and derivatives of the functional data, which will be extensively considered in this work. Readers familiar with functional data analysis can skip this chapter.

Chapter 3 discusses challenges of clustering and reviews some of the techniques in this area. We especially focus on spectral clustering for multivariate data by describing the method and

referring to the most popular algorithm for implementing spectral clustering. The second half of the chapter provides an overview of clustering functional data and details the main categories of CFD and their approaches. In additions, it lists all the available R functions for CFD, and selects a few of them for further application later in the thesis. It discusses two evaluation measures that will be used to assess the clustering results: the correct classification rate (CCR) and the adjusted Rand index (ARI).

Chapter 4 introduces a new framework for clustering functional data based on the spectral clustering algorithm (FSC-S). It defines our general smoothing model that will be associated with the proposed clustering technique. This chapter also introduces the choice of distance measure and the need to extend the measure to the curves' derivatives. Following the definition of smoothing and distance, we define the functional spectral clustering algorithm. It demonstrates the application of the proposed algorithm on the Berkeley growth data. Finally, the chapter discusses the perturbation theory and its relation to spectral clustering and our proposed functional spectral clustering method.

Chapter 5 presents a new criterion for model selection, by introducing the downsampling approach (DSC). The criterion is based on the clustering stability philosophy, therefore we discuss and review a number of existing techniques that considers clustering stability in clustering multivariate data and functional data. The chapter then defines the downsampling model, and the criteria that must be satisfied. The downsampling approach requires a designed sampling technique which is illustrated in the chapter. The new criterion was applied with our proposed clustering method and the selected CFD methods that were used on the Berkeley growth data.

Chapter 6 extends the proposed functional spectral clustering approach to estimate the number of clusters within the algorithm (FSC-DSC). It discusses the importance of the scaling parameter σ in defining the structure of the data and subsequently the parameter k, the number of clusters. The extended algorithm uses the eigengap heuristic to estimate k with the aid of σ, while these estimates will be further examined through the stability concept. The chapter

illustrates the new approach of FSC-DSC for selecting the number of clusters and clustering the Berkeley growth data.

Chapter 7 presents an extensive simulation study with a range of scenarios that possess different levels of difficulty in terms of extracting the clusters and estimating the correct number of clusters. The simulated data will be used to investigate the effectiveness of the three proposed approaches FSC-S, DSC, and FSC-DSC.

Chapter 8 considers the application of our new methods on a real-life dataset, the average house prices in Scotland from 1993 to 2018. The chapter illustrates the application of DSC and FSC-DSC to the house prices data and compares and discusses the results obtained by these approaches.

Chapter 9 summarizes the main findings of the thesis. It highlights the advantages and limitations of the proposed methodology and suggests possible directions for future work.

Software and articles: Additionally, we have compiled the computational codes for implementing functional spectral clustering approach (FSC-S) as a github repository `FSC`. The downsampling approaches (DSC) and (FSC-DSC) will be added to the same repository in the near future. In addition, we published an article (Al Alawi et al., 2019) that discusses the proposed functional spectral clustering approach and its applications, and presented the work at the 34th IWSM workshop in 2019. Further, we are currently preparing another manuscript based on this thesis: Al Alawi,M., S.Ray, and M.Gupta. Downsampling based model selection for functional data clustering (2021).

Chapter 2

Review of Functional Data Analysis (FDA)

This chapter introduces functional data analysis and demonstrates some of the widely used techniques, based on (Ramsay and Silverman, 2005). The first section introduces FDA and its main model. Section 2.2 describes two widely used functional data sets. Section 2.3 explains the smoothing techniques and the basis expansions. Section 2.4 presents the exploratory functional data analysis tools, including functional principal component analysis. Section 2.5 discusses functional data that are defined by phase and amplitude variations and introduces derivatives of the functional data. Readers familiar with the literature on FDA can skip this chapter.

2.1 Introduction

Functional data are realizations of a smooth process that vary over a continuum, usually time, but it could take any other domain[1], and they take the shape of smooth curves. Functional data are considered to be infinite dimensional, and involves repeated measures of the same process. It is difficult to propose a statistical distribution for functional data, which makes their analysis and inference more challenging. The general model of functional data can be written as:

$$y_i = x(t_i) + \varepsilon_i, \tag{2.1}$$

[1]These continuum can be space, or weight, or probability, etc.

where:

- y_i represents the observed data vector over time t at $i \in [1,T]$.
- $x(t_i)$ represents the estimated curve that is usually expanded using basis functions $x(t_i) = \sum_{j=1}^{k} c_j \phi_j(t_i)$, where $\phi_j(t_i)$ represents the basis functions and c_j represents the coefficients.
- The error ε *is i.i.d* and distributed as $N(0,\sigma^2)$.

The term Functional Data Analysis (FDA) dates back to Ramsay and Dalzell (1991). To our knowledge, Ramsay (1982) was the first to discuss what happens if the data are considered as functions. A reader may ask how functional data is different from time-series data, or longitudinal data, or even high dimensional data. It depends on how the data is structured and viewed and what type of questions are asked. According to Ramsay and Silverman (2005), what makes FDA unique is the curve registration that is based on transforming curves, and the estimation of the curves' derivatives. Indeed, FDA can answer questions that other methods cannot, for example, predicting a scalar response from functional covariates.

In recent years functional data analysis has gained more attention from different scientific fields and has been applied to solve numerous real-life problems. Ramsay and Silverman (2005) provided many statistical techniques to handle this type of data, and created the software package '`fda`' in R to implement the techniques (Ramsay et al., 2014).

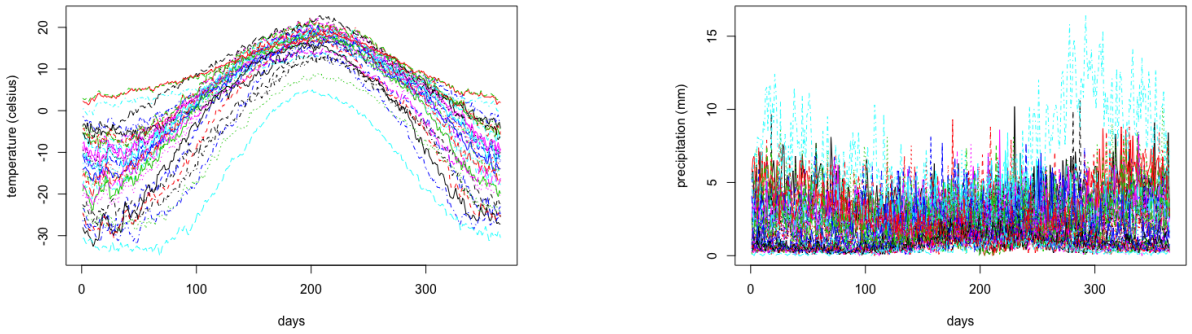
2.2 Examples of Real Life Data

In this section we introduce some data sets that have been frequently used in functional data analysis, to give the reader an insight about these data. The chosen data have been widely used by many researchers and they are also available in the R package `fda`. Both the Canadian weather data set (Section 2.2.1) and the Berkeley growth data set (Section 2.2.2) will be

used throughout the thesis, to illustrate some of their important features and to demonstrate our proposed techniques of clustering functional data.

2.2.1 The Canadian Weather Data

This dataset consists of the daily temperature and precipitation measures of 35 selected cities distributed across Canada. There are 365 records for each city, which makes this data very dense. There are also the monthly measures of temperature and precipitation, giving only 12 records for each city over the year and that is a relatively sparse data set. Figure 2.1 shows curves of the daily temperature and precipitation measures for the selected Canadian cities, where each curve represents a city. The temperature data usually show the shape of sinusoidal functions, as the coldest days are during December and January, while the warmest days are in June and July, which is seen in Figure 2.1a. We can also notice the noise over the curves specifically the precipitation data. To understand the underlying process and to accommodate the error, one proposal is to smooth the data (see Section 2.3) and then perform statistical analysis on the smooth data.



(a) Temperature curves

(b) Precipitation curves

Figure 2.1: The Canadian weather data (a) Daily temperature of 35 cities (b) Daily precipitation of 35 cities.

2.2.2 The Berkeley Growth Data

The Berkeley growth study data includes the heights of 39 boys and 54 girls. These heights have been measured from age 0 to 18 years old, and the time points are not evenly spaced.

The measurements were taken every quarter until the child is 1 year old, then annually from 2 to 8 years old, and every 6 months from 8 to 18 years old. Every curve represents a child and they are independent of each other and the noise in each curve is relatively small. The three panels in Figure 2.2, from left to right, show the heights (in cm) for the children, the first derivatives, which reflect the rate of change in the height functions (growth rates), and finally the second derivatives, which reflect the acceleration that happen to their heights over the years. The first derivative graph holds more information about puberty and it is clearly displayed as a bump between age 11 and 14 for most children. The acceleration graph also shows a strong positive acceleration in the age between 1 and 5 years old followed by a negative deceleration. This example demonstrates the importance of derivatives in some data, and that is why it is of interest to consider the data as functions instead of taking them as vectors of discrete values. FDA researchers have used this rich feature of the data and their derivatives to cluster this dataset into different subgroups.

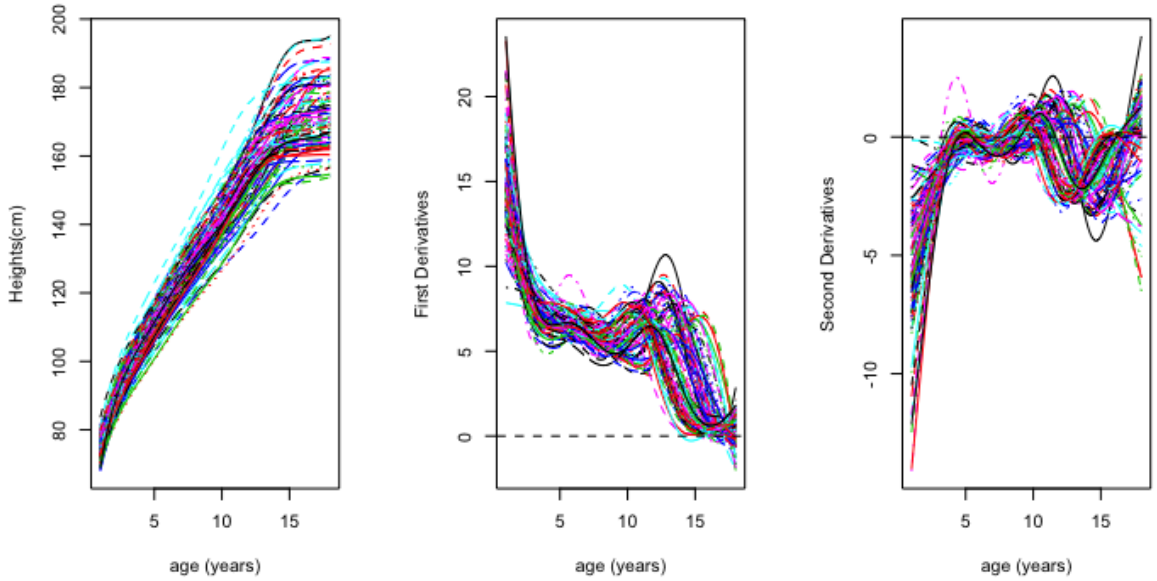


Figure 2.2: The Berkeley growth data in their original trajectories (left) and as first derivatives (middle) and as second derivatives (right).

2.3 Smoothing and Basis Expansions

Although functional data are initially observed as discrete data points, the concept of functional data is to assume every vector y_i is a single object. Thus, a vector y_i is the collection of all the points $\{y_1, y_2, ..., y_T\}$, while t_i represents time over which the data are recorded. The vector y_i can be smoothed by applying the basis functions.

A basis function system is a set of known functions ϕ_j that are linearly independent of each other and their linear combinations are able to effectively model the true structure of the data. There is a wide range of basis function systems to choose from, such as: polynomials, Fourier, splines, wavelets, and kernels. Ullah and Finch (2013) reviewed 84 studies with FDA applications and showed that B-spline smoothing was the most popular technique; about 30% of the studies used B-splines with a large number of knots. They hypothesized that this is probably because of their simplicity and flexibility in handling different situations. Ramsay and Silverman (2005) stated that the choice of the smoothing technique depends on the the behaviour of the data. For instance, Fourier functions are used for data with periodic or cyclic behaviour, while splines are typically used for data that do not show cyclical forms. Moreover, wavelets are used for data with discontinuities or rapid changes (Ullah and Finch, 2013).

The smoothing technique is considered as a key aspect of functional data analysis, because it is the process that moves the raw data at discrete times into continuous functions. This new representation of data allows the researcher to evaluate the records at any time point, which is helpful specially when the data are not equally spaced. A proper smoothing technique can reduce the noise and allows the evaluation of derivatives.

Looking back at equation (2.1), if we assume y_i represents the observed data vector, these data are converted into a curve or function $f(t_i)$ via expanding the basis $\sum_{j=1}^{k} c_j \phi_j(t_i)$. This expression gives the combination of the basis functions $\phi(t_i)$, and the coefficients c of k dimensions. Thus, $y_i = x(t_i) + \varepsilon_i$ will be $y_i = c_1\phi_1(t_i) + c_2\phi_2(t_i) + ... + c_k\phi_k(t_i) + \varepsilon_i$. In a matrix format,

this is written as $\boldsymbol{y} = \boldsymbol{x}(\boldsymbol{t}) + \boldsymbol{\varepsilon}$, where:

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix},$$

$$\boldsymbol{x}(\boldsymbol{t}) = \boldsymbol{\Phi}(\boldsymbol{t})^{\boldsymbol{T}}\boldsymbol{c} = \begin{bmatrix} \phi_1(t_1)\ \phi_2(t_1)\ \ldots\ \phi_k(t_1) \\ \phi_1(t_2)\ \phi_2(t_2)\ \ldots\ \phi_k(t_2) \\ \ddots \\ \phi_1(t_T)\ \phi_2(t_T)\ \ldots\ \phi_k(t_T) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix}, \text{ and}$$

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_T \end{bmatrix}.$$

Smoothing by least squares

In order to get a smoother version of the raw data vector y_i, we estimate the coefficients c_j of the expansion by minimizing the least squares criterion:

$$SSE(\boldsymbol{c}) = \sum_{i=1}^{T} \left[y_i - \sum_{j=1}^{k} c_j \phi_j(t_i) \right]^2. \tag{2.2}$$

Writing the above in a matrix format gives:

$$SSE(\boldsymbol{c}) = (\boldsymbol{y} - \boldsymbol{\Phi c})^T (\boldsymbol{y} - \boldsymbol{\Phi c}), \tag{2.3}$$

and solving for c:

$$\hat{\boldsymbol{c}} = \left(\boldsymbol{\Phi}^T \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^T \boldsymbol{y}. \tag{2.4}$$

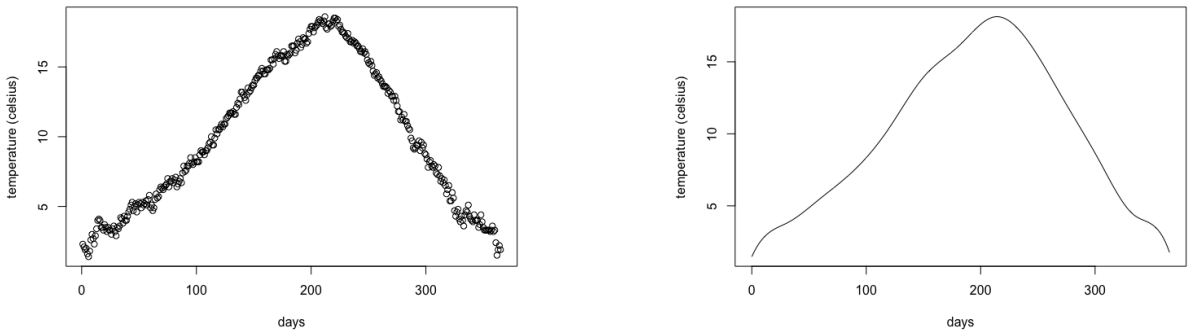
It is optimal to estimate c_j by the least squares fit when the errors are independently and

identically distributed, following a normal distribution with mean 0 and a constant variance. However, this situation is not always true, for instance the variance of y might vary over the observed time. In this case, we can add a weight matrix W to the least square equation (2.2), so c can be estimated as in equation (2.5), and the estimated data values $\hat{y}$ as in equation (2.6):

$$\hat{c} = \left(\Phi^T W \Phi\right)^{-1} \Phi^T W y, \tag{2.5}$$

$$\hat{y} = \Phi\left(\Phi^T W \Phi\right)^{-1} \Phi^T W y = S y. \tag{2.6}$$

After defining the basis function Φ, the researcher can estimate $\hat{c}$, and thus find the smoothing matrix S, which in turn will lead to the smoothed version of the raw data $\hat{y}$. Consider the Canadian weather data example, and for simplicity we will talk about only one vector (i.e. one city). The average daily temperature of Vancouver in Canada over a year is shown in Figure 2.3a. To smooth the data as shown in Figure 2.3b, we used a B-splines basis of order 4 with knots at the end of every month.

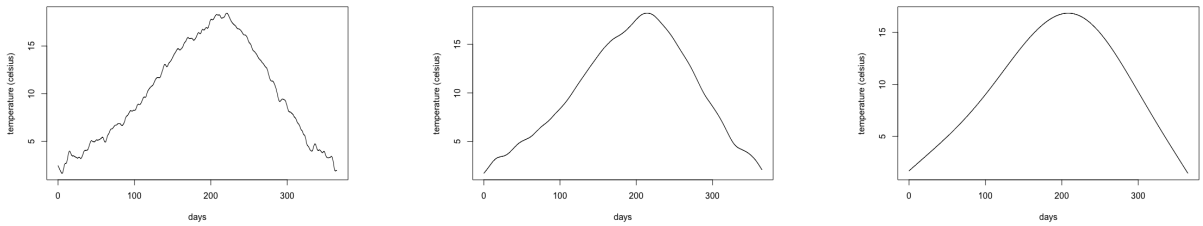


(a) from data (b) to functions

Figure 2.3: Vancouver temperature over a year: (a) as raw data, and (b) as smooth curves.

B-splines (De Boor et al., 1978) have been commonly used in FDA research. B-splines are polynomial segments on specific subintervals and zero otherwise. These segments are constrained to be smooth at the join (knot). Their functions are defined by the order of the spline

(n) and the number and location of the knots, consequently the number of basis functions = n + the number of interior knots. The total number of basis functions in the above example is 15, coming from 4 + 11 interior knots[2]. It should be mentioned that a B-spline of order n is equivalent to polynomials of degree $n-1$. B-splines of order 4 are the most popular choice that can give up to the second derivative[3] and allows controlling the smoothness, with the knots either being placed evenly or by placing more knots in sharp curves (where the values change fast). Another approach, however, is to use a saturated model, with applying a penalty term. A saturated model refers to placing a knot at every data point. Going back to our example, if we fit B-splines of order 4 again but with placing a knot at every day this time, we can control the smoothness of the curve by applying a smoothing parameter λ as shown in Figure 2.4. If the smoothing parameter is small, the curve will look wiggly (Figure2.4a), while if the parameter is large, the curve will look more flattened (Figure 2.4c). The optimal choice of λ must give a balanced form that shows the important features of the curve as in Figure 2.4b, which looks similar to the previous model (Figure 2.3b).



(a) $\lambda = 1$ (b) $\lambda = 1,000$ (c) $\lambda = 1,000,000$

Figure 2.4: The smoothed Vancouver temperature curve around the year when applying different smoothing parameters λ.

Smoothing with roughness penalty

The curves in Figure 2.4 were achieved by minimizing the Penalized Sum of Squared Error (PENSSE).

$$PENSSE_\lambda(x) = [y - x(t)]^T [y - x(t)] + \lambda \int [D^2 x(t)]^2 dt. \tag{2.7}$$

[2]Note there is one knot at $t = 0$, so total number of knots is 13.
[3]If the researcher wants to get a p derivative, then the number of order must be p + 2.

Equation (2.7) is similar to equation (2.3) but with adding the term $\lambda \int [D^2 x(t)]^2 dt$, where this additional term, λ, is a scalar that measures the smoothness, $D^2 x(t)$, is a scaler that measures the roughness of the curve x, and can be summarized by the penalty matrix R as below. Note that in equation (2.7), increasing λ will penalize the roughness and consequently gives a smooth fit. It should be also mentioned that in many studies, the roughness penalty and smoothing parameter are used for the same meaning. Now the roughness scalar value can be written as,

$$\begin{aligned} &\int [D^2 x(t)]^2 dt \\ &= \int [D^2 c\Phi(t)]^2 dt \\ &= \int c^T D^2\Phi(t) D^2\Phi(t) c.dt \\ &= c^T [\int D^2\Phi(t) D^2\Phi^T(t).dt].c, \\ &= c^T R c. \end{aligned} \tag{2.8}$$

Accordingly, we can rewrite equation (2.7) as in equation (2.9), while the estimated coefficients $\hat{c}$ and the estimated data values $\hat{y}$ are as in equation (2.10) and (2.11) respectively.

$$PENSSE_\lambda(x) = [y - x(t)]^T [y - x(t)] + \lambda c^T R c, \tag{2.9}$$

$$\hat{c} = \left(\Phi^T W \Phi + \lambda R\right)^{-1} \Phi^T W y, \tag{2.10}$$

$$\hat{y} = \Phi\left(\Phi^T W \Phi + \lambda R\right)^{-1} \Phi^T W y. \tag{2.11}$$

Regardless of the selected fitting/smoothing model, there is always a concern about getting a balance between over-fitting and over-smoothing. Over-fitted curves will possess high noise/variation, while over-smoothed curves will fail to project the true structure of the data. In other words, selecting the proper number of bases or choosing the optimal λ in the saturated model is a trade-off between bias and variance. This relationship can be represented by the

Mean Square Error (MSE), which can be written as:

$$MSE[\hat{x}(t)] = Bias^2[\hat{x}(t)] + Var[\hat{x}(t)], \tag{2.12}$$

where $\text{Bias}[\hat{x}(t)] = x(t) - E\hat{x}(t)$, and $\text{Var}[\hat{x}(t)] = E[\{\hat{x}(t) - E\hat{x}(t)\}^2]$. Figure 2.5 illustrates this relationship for the Vancouver temperature data example. This time the observation has been smoothed by a Fourier basis over different numbers of basis functions. We have applied 1000 simulations of the smoothing model, by randomly sampling and relocating the error. Then, for each instance, we fit the model and calculate the bias, variance, and SME. The graph shows using 14 basis functions might be a good choice for smoothing this observation.

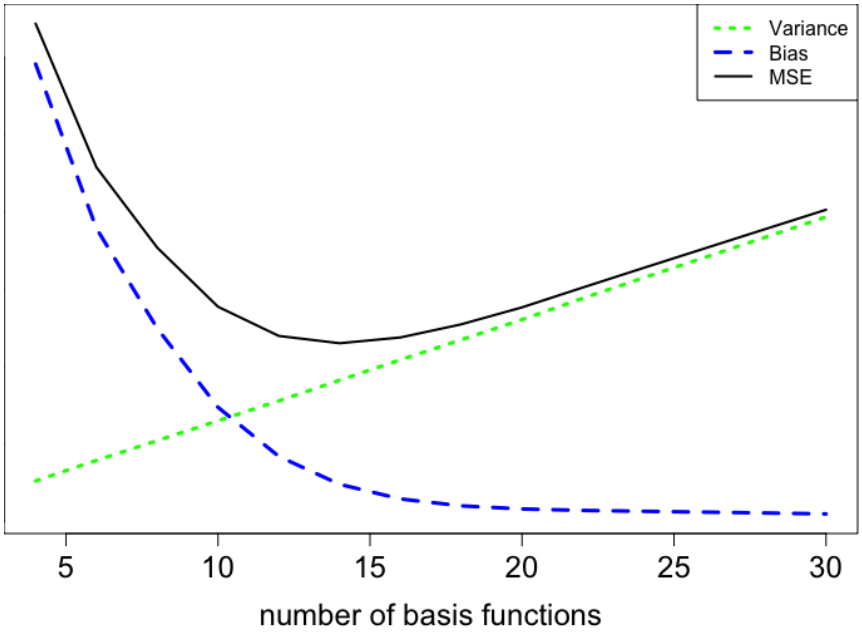


Figure 2.5: Simulating Bias and Variance for Vancouver city from the Canadian weather data.

Therefore to balance that bias and variance, it is appropriate to use a saturated model with a smoothing parameter λ, and use cross-validation to choose between the $\lambda's$. This can be done through minimizing the Generalized Cross Validation (GCV) criterion:

$$GCV(\lambda) = \left(\frac{n}{n - df(\lambda)}\right)\left(\frac{SSE}{n - df(\lambda)}\right). \tag{2.13}$$

In the example of Vancouver temperature, $\lambda = 1000$ gives low SSE and the curve looks smooth. However, note that sometimes a range of λ values can give similar GCV as in Figure 2.6. In this case we can check visually for the λ that gives a good-enough fit. Considering the 35 cities in the Canadian weather data, we can find the overall λ for all the observations by using the same approach (Figure 2.7).

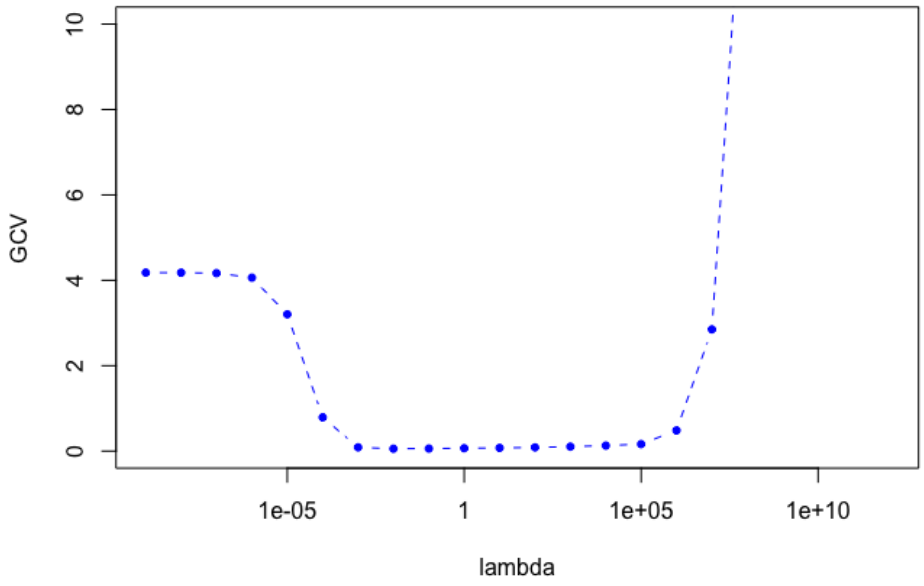


Figure 2.6: GCV curve shows the dip when λ values between 10^{-3} and 10^{5} for Vancouver temperature.

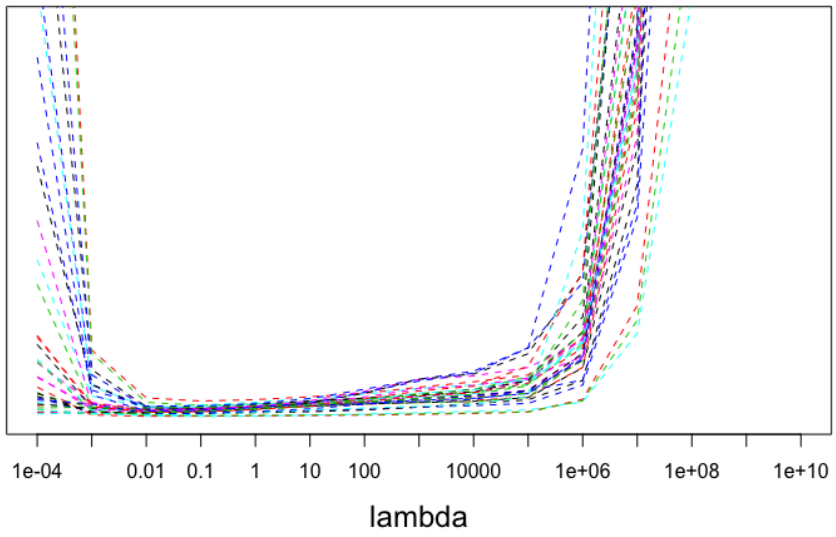


Figure 2.7: GCV curve shows the dip when λ values between 10^{-2} and 10^{2} for all the Canadian cities.

Instead of the B-spline basis the researcher can also use Fourier basis. Fourier basis are usually used to represent periodic curves, where these functions repeat themselves over a period of time. They are very popular basis functions, however, they are primarily used to fit periodic functions with no extreme changes or abrupt features (Ramsay and Silverman, 2005). In literature, there is a number of FDA applications that have used Fourier basis to smooth the data Ratcliffe et al. (2002); Guo (2004); Laukaitis and Račkauskas (2005).

2.4 Exploratory Functional Data Analysis

The summary statistics of any data set can give an impression about its general structure and help exploring the main features. Exploratory analysis for functional data can be carried out similarly to multivariate data. Consider a set of curves $y_j(t), j = 1,...,n$, where the mean function will be represented by one curve and is a result of calculating the average of values each at a time. Thus called the point-wise mean function, and is found by:

$$\bar{y}(t) = \frac{1}{n}\sum_{j=1}^{n} y_j(t). \tag{2.14}$$

Similarly the point-wise variance function is given by:

$$var_y(t) = \frac{1}{n-1}\sum_{j=1}^{n}[y_j(t) - \bar{y}(t)]^2. \tag{2.15}$$

Further, to explore the dependence between the curves at different time points (say t_s and t_r), we can find the covariance and the correlation functions by:

$$cov_y(t_r, t_s) = \frac{1}{n-1}\sum_{j=1}^{n}[y_j(t_r) - \bar{y}(t_r)][y_j(t_s) - \bar{y}(t_s)], \tag{2.16}$$

$$corr_y(t_r, t_s) = \frac{cov_y(t_r, t_s)}{\sqrt{var_y(t_r)var_y(t_s)}}. \tag{2.17}$$

Figure 2.8 displays the mean function and the covariance function as a set of level contours for the Canadian weather data. The red curve represents the mean temperature of the Canadian cities, while the contour plot shows high covariance between days with similar temperature and low covariance between days with different temperature.

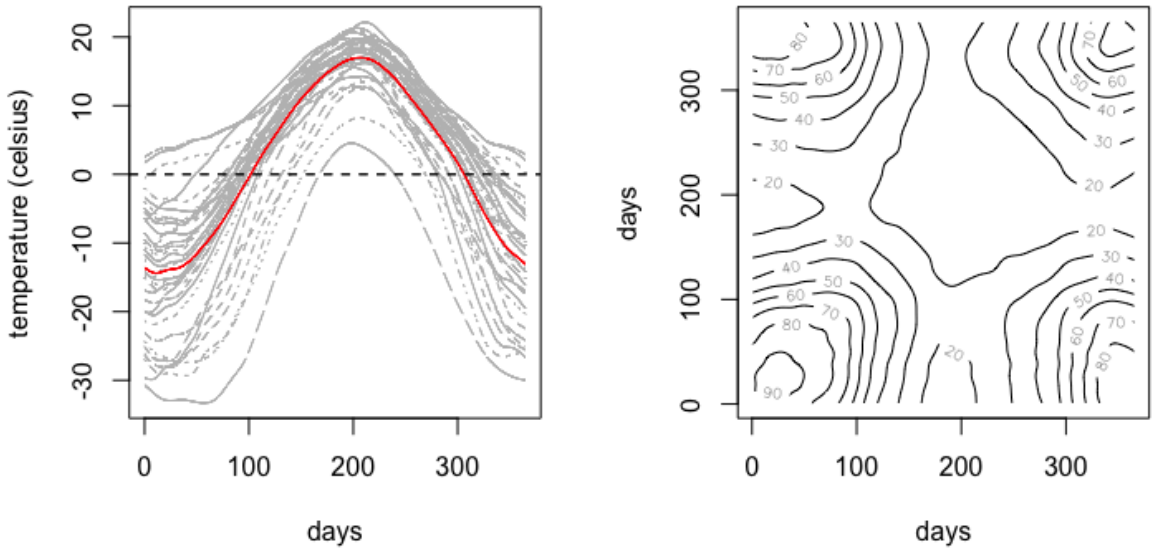


Figure 2.8: The mean function (red curve) of the Canadian temperature data (left), the contour plot of the correlation function for the same data (right).

Principal component analysis has been always a powerful tool to investigate the variation in a data set. Similarly functional principal component analysis (FPCA) can be used to explore the variability in functional data. It is considered as a dimension reduction technique, which recognizes the informative components that explain most of the variation in the functional data. In terms of computation, FPCA replaces the eigenvectors by eigenfunctions, matrices by linear operators, and summations by integrations, which makes the FPCA different from PCA.

Assuming again some continuous functions $y_j(t)$ with mean $\mu = \bar{y}(t)$ and covariance $G(t_s,t_r) = cov(y(t_s),y(t_r))$, the covariance can also be written as a decomposition of eigenvalues and eigenvectors;

$$G(t_s,t_r) = \sum_{k}^{\infty} \rho_k \xi_k(t_s)\xi_k(t_r), \tag{2.18}$$

where $\xi(t)$ represents the eigenfunctions of the variance-covariance function, and ρ are the

eigenvalues. The eigenfunctions can be calculated by solving the following eigen-equation:

$$\int G(t_s,t_r)\xi(t_r).dt_r = \rho\xi(t_s). \tag{2.19}$$

Considering the Canadian weather data again, Figure 2.9 shows the first 4 functional principal components for the data. These components represent most of the variation in the data.

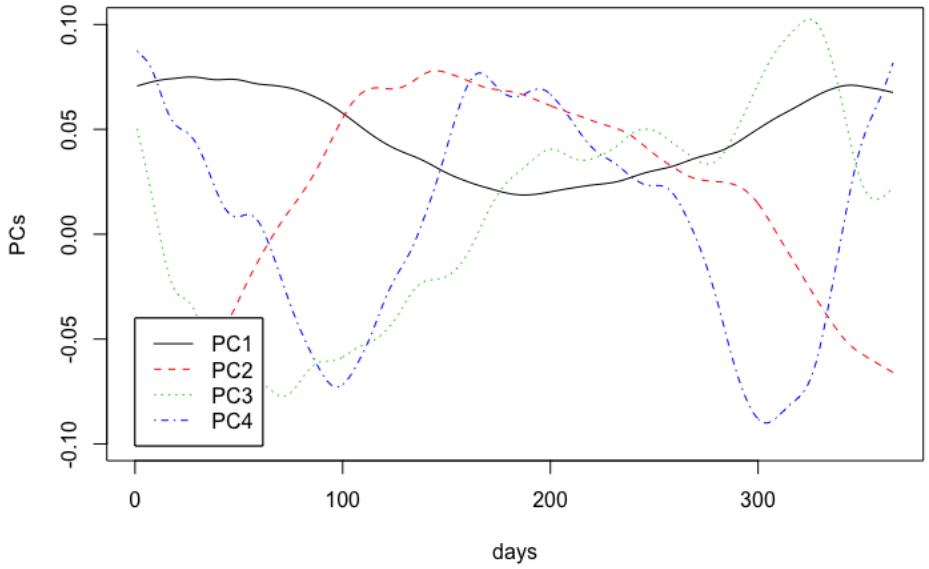


Figure 2.9: Principal components functions for the Canadian weather data. PC1 displays variation in the overall temperature, PC2 displays variation of the relative temperature in winter and summer, PC3 displays variation between fall and spring, and PC4 displays variation of the relative length of winter and summer.

2.5 More Complex Context of Functional Data

With the advance of modern technology, different types of data are being recorded over various regular/irregular time points. Thus, functional data come in different forms and each functional data set is unique and must be treated individually. For instance in a particular functional data set, some curves exhibit variations in amplitude or phase or both. Further, in some functional data the derivatives hold more information about the data than the original trajectories, thus it is of interest to estimate the derivatives of the functions in that case. In this section, we present

these forms of functional data, as they will be extensively discussed in this thesis.

2.5.1 Functional Data with Phase or Amplitude Variations

One of the situations motivating the development of important tools for FDA is when there are phase displacements and/or amplitude variations in curves. The presence of these variations often create challenges in analysing the data. According to Marron et al. (2015), their presence often inflates data variance, weakening the underlying structure of the functional data, and can lead to inaccurate analyses. There have been some studies that deal with this type of functional data, for instance, Sangalli et al. (2010), developed an algorithm that separate amplitude and phase variability and simultaneously cluster and align the functional data. Further, Srivastava et al. (2011) introduced a framework based on Fisher-Rao Riemannian metric to derive a proper distance for separating the phase and the amplitude variability in functional data. Marron et al. (2015) summarize several current ideas for separating phase and amplitude components, and motivate the importance of dealing with these variations in functional data.

In Ramsay and Silverman (2005), the authors defined the amplitude variability to be related to the size of a specific feature (usually a peak), while the phase variability is the shift in the timing of the specific feature regardless of their sizes. This definition can be illustrated by Figure 2.10. In order to get valid measures from the data, the phase and amplitude variabilities must be identified and separated. The authors suggested curve registration, which is transforming the curves by transforming their arguments. Several types of curve registration problems and examples have been discussed, for further details refer to Chapter 7 of Ramsay and Silverman (2005).

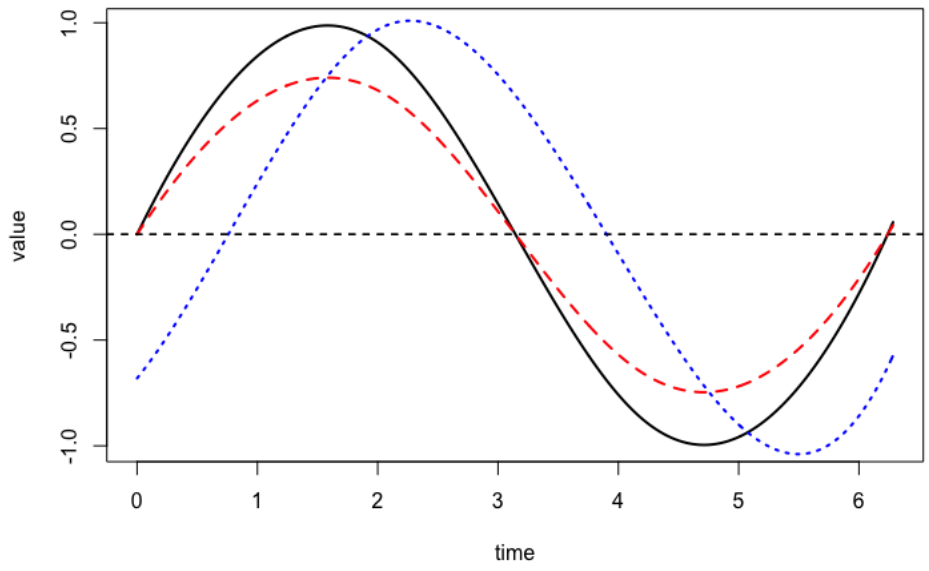


Figure 2.10: The original curve is represented as black, the red curve shows the effect of amplitude variation, while the blue curve shows the effect of phase variation when applied to the original curve.

2.5.2 Derivatives of Functional Data

One of the advantages of moving data to functions is the possibility of computing the derivatives of the functions. Consider a function $y(t)$: the first derivative of the function is $Dy(t) = \frac{d}{dt}y(t)$, the second derivative is $D^2y(t) = \frac{d^2}{dt^2}y(t)$, and the m^{th} derivative is $D^my(t) = \frac{d^m}{dt^m}y(t)$ [4]. The first derivative of a function of time represents the rate of change in the observations over time, while the second derivative refers to the acceleration over time. These facts might be important in analysing some functional data. Looking back at Figure 2.2, the functional data are displayed in their original trajectories, their first derivatives and the second derivatives of the growth data. Also we have already seen the second derivative in equation (2.6) to measure the roughness of the curve.

The use of derivatives is of interest in extending the range of simple graphical exploratory methods, and in developing comprehensive approaches. This is a scheme that will be discussed in more detail in Chapter 4.

[4]Note: the derivatives of functions can be also written as $f_i^q(t)$, where q represents the order of derivative.

Chapter 3

Existing Multivariate and Functional Clustering Techniques

This chapter discusses clustering analysis and reviews the main contributions in this area. The first section defines clustering analysis in general. Section 3.2 considers the major clustering approaches for multivariate data and focuses on spectral clustering. Section 3.3 discusses clustering analysis for functional data. Section 3.4 defines two clustering evaluation metrics that will be of use throughout the thesis.

3.1 Introduction

Clustering analysis is the process of partitioning data to detect patterns. It is considered an unsupervised technique where there is no predefined labelled data. Cluster analysis groups data objects based only on information found in the data that describes the objects and their relationships. The aim is that objects in the same cluster must be similar as much as possible and objects in different clusters must be different as much as possible (Everitt et al., 2011). The similarity/dissimilarity measurements must be clearly defined and have practical meaning about the nature of the data.

Clustering analysis is performed through several steps. These steps can be summarized as

follows:

- Extracting some features that can best represent the data.
- Defining the proximity measure between objects.
- Selecting the clustering algorithm according to the problem.
- Evaluating the results of the clustering algorithm.
- Explaining the results in terms of the practical meaning of the data.

For years, research in clustering has led to a wide range of approaches and paradigms in different fields of application. An overview of these approaches will be given in the next sections.

3.2 Clustering Multivariate Data

This section discusses the different categories of cluster analysis methods and highlights the principles behind those approaches for clustering multivariate data (Hennig et al., 2015; Nagpal et al., 2013; Tan et al., 2005; Xu and Tian, 2015).

The traditional way of categorizing clustering is to determine whether it is done by partitions or hierarchical structure. Partition-based methods can find all the clusters simultaneously as divisions of the data and do not assume any hierarchical structure. Hierarchical clustering is based on locating nested clusters, thus the large clusters usually contain smaller clusters or sub-clusters (Johnson, 1967). There have been many hierarchy-based applications, for example, Guha et al. (2000); Karypis et al. (1999). The centroid-based clustering approach is the most common application of the partition category, some examples being K-means (MacQueen et al., 1967), Partitioning Around Medoids (PAM) (Kaufman and Rousseeuw, 1990), and K-medoids (Lucasius et al., 1993). Partition-based clustering can be hard or soft type. In hard clustering, each object x_i belongs to one and only one cluster, while in soft clustering, the object x_i is assigned a probability of membership to each of K clusters. This approach is more appropriate

when an object may be close to several clusters.

Apart from the above main divisions, a more modern categorization of clustering methods distinguishes between density-based and non-density-based approaches (Menardi, 2016). In the first category, one assumes a probability distribution to the data. The density-based concept takes two different directions: the parametric and the nonparametric approaches. The parametric refers to model-based clustering approach, and the nonparametric refers to mode-based clustering approach. The concept of model-based is selecting a specific model for each cluster and finding the data points which best fit that model (Wolfe, 1970). It is based on the assumption that data are generated from a mixture of probability distributions and it usually employs the expectation maximization algorithm (EM). One popular and traditional model-based approach is COBWEB developed by Fisher (1987). An alternative approach to estimating mixtures is to assume a Bayesian prior for the mixture parameters and the number of components (Richardson and Green, 1997). For more profound research of the model-based clustering approaches go to (Fraley and Raftery, 2002). Whereas, the original concept of modal-based is related to defining relatively densely regions and relatively empty regions (Carmichael and Julius, 1968). A taxonomy of more recent modal-based methods that have been proposed in the literature is provided by (Menardi, 2016). On the other hand, the non-density-based category mostly refers to the distance-based approaches, which defines the clusters based on some similarity measures between data objects. One specific approach that is considered under this category and will be discussed in more detail below is spectral clustering. Spectral clustering aims to define similarities between data points based on the variations in some eigenvectors, where the matrix of eigenvectors is derived from the data based on a Gaussian similarity function or based on a nearest neighbour graph. In addition to the above, there exist different approaches that might fall beyond these categories.

3.2.1 Spectral Clustering for Multivariate Data

A simple description of spectral clustering can be illustrated as; given some data points $x_1, \ldots, x_n$, the similarity between the data points can be written as $w(x_i, x_j)$, where w represents the edge

weight between point x_i and point x_j. These weights (or similarities) can help partitioning the data into groups so that edges within a group have high weights and edges between groups have low weights. The similarities between points are converted to similarity graph $\mathbb{A}$ (also can be defined as $\mathbb{W}$), which can be constructed using different functions. The similarity graphs that have been used in the algorithms of spectral clustering are the k-nearest neighbour graph, the ε-neighbourhood graph, and the fully connected graph. Based on the similarity graph, the degree matrix $\mathbb{D}$ is defined as the diagonal matrix of the sum of the degree of weights (similarities) $w(x_i, x_j)$ over each row. Then, to reduce the dimensions and filter the eigenvectors that capture most of the variation in the data of the similarity graph. The graph Laplacian matrix $\mathbb{L}$ is constructed from the spectrum of that similarity matrix using multiple ways that can be either considered as normalized Laplacian or as unnormalized Laplacian. One of the common unnormalized graph Laplacian matrix is found by $\mathbb{L} = \mathbb{D} - \mathbb{A}$. In the Laplacian matrix, the largest k-eigenvalues identify the eigenvectors that will represent the clusters, where k is known in advanced. The graph Laplacian matrix can be used to find many useful properties of a graph, thus it is considered as the key element of spectral clustering. However, there is no standard definition/format of the Laplacian matrix (Von Luxburg, 2007). Spectral graph theory focuses on studying graph Laplacian matrices and their properties (Chung and Graham, 1997).

The basic idea of spectral clustering is partitioning the graph into two groups by using the second eigenvector of the graph Laplacian matrix. This work is dated back to Donath and Hoffman (1973), who first proposed the use of eigenvectors of the adjacency matrices to find partitions in graphs. On the other side, Fiedler (1973) found that the bi-partitions are associated with the second eigenvector of the graph Laplacian and thus can be used for partitioning graphs. Note the use of the second eigenvector is because the first eigenvector consists of constant values and thus cannot reveal any grouping information. Later, spectral clustering was combined with linear programming (Barnes and Hoffman, 1984), and since then, spectral clustering gained more attention with more studies, extended algorithms and extensive applications.

Among the several approaches for spectral clustering, the recursive spectral method and the

multiway spectral method proposed by Shi and Malik (2000) and Ng et al. (2002) respectively are the most popular ones. The primary difference between the two is the type of normalized graph Laplacian matrix they use. Where the first used $\mathbb{L} = \mathbb{I} - \mathbb{D}^{-1}\mathbb{A}$ while the second used $\mathbb{L} = \mathbb{I} - \mathbb{D}^{-1/2}\mathbb{A}\mathbb{D}^{-1/2}$. To view the full comparisons between the two methods, go to Von Luxburg (2007), or Verma and Meila (2003).

In addition to these algorithms, there are others that considered as unnormalized spectral clustering (Barnard et al., 1995; Guattery and Miller, 1998). According to Von Luxburg et al. (2008), however, the eigenvectors of graph Laplacian matrices converge under very general conditions in normalized spectral clustering, while in the unnormalized case additional assumptions should be made for the algorithm to be consistent.

Verma and Meila (2003) conducted a comparisons between spectral clustering techniques and other clustering techniques and concluded that overall the spectral methods lead to competitive results compared to other clustering techniques. They also showed that multiway methods (Ng et al., 2002) perform slightly better than recursive methods (Shi and Malik, 2000) particularly when there is a clear structure in the data.

The advantages of using spectral clustering is the ability to cluster in high efficiency with high accuracy of clustering results without a need to make strong assumptions about the data. In addition, spectral clustering is simple to implement and can be solved efficiently by standard linear algebra methods. However, some disadvantages of spectral clustering could be time complexity increasing considerably with the increasing of graph complexity. Also, there is no one unique algorithm of spectral clustering, and there exist different choices of the mathematical objects involved in the process such as the similarity graph and the Laplacian graph.

3.3 Clustering Functional Data

This section reviews the main contributions to clustering functional data, and categorizes the existing methods into three different categories. Additionally, it presents recent algorithms for clustering functional data that are available in R.

Clustering Functional Data (CFD) has received more attention recently and has expanded rapidly in the last few years. In theory, standard clustering approaches for multivariate data can be applied for functional data by imposing some features to accommodate the structures of functional data. Yet, there have been numerous studies proposed and developed for functional data. Furthermore, some researchers categorized these proposed approaches into different classes. For instance, Jacques and Preda (2014a) divides CFD into three main techniques. The first one is the two-stage method, which reduces the dimensions of the data then applies standard clustering methods. The second approach is non-parametric clustering that uses specific distances and dissimilarities between the curves. The third technique is model-based clustering, it assumes the data come from a mixture of distributions. Bayesian model-based approaches assume probability distributions on some parameters that describe the curves. In addition, there is one method called the raw-data clustering that discretizes/regularizes the functions at some time points, so it does not consider the functional structure of the data.

Wang et al. (2016) followed a similar classification of CFD methods by dividing them into three categories. One category consists of model-based clustering approaches, the second category consists of centroid-based clustering approaches, while the last category contains subspace-based clustering approaches, the concept of this category is based on using combinations of the basis coefficients, the mean functions, and the set of eigenfunctions to identify the subspace, and to characterize the clusters. This approach has been proposed by Chiou and Li (2007).

More recently (Yassouridis and Leisch, 2017) gathered different CFD approaches in one R package named `funcy`. They used a simulation study to compare the different approaches. The

Rand index (Rand, 1971) was calculated as a performance measure between true cluster assignments and the resulting clusters from applying the algorithms. Further, a systematic review of FDA (Ullah and Finch, 2013) showed that the most common clustering technique for functional data is hierarchical cluster analysis, mostly applied to gene expression data.

Given the large amount of research on CFD, we follow the Jacques and Preda (2014a) classification, describing some significant studies, and detailing the challenges and drawbacks for each category. It should be mentioned that the following classification of CFD methods fall under the non-density-based category, since there is no definition of density for functional data.

Two-stage clustering methods for functional data are a natural extension of the multivariate clustering approaches and as such there exist a host of clustering approaches that fall under this category. They range from the initial proposal by Abraham et al. (2003) to a more recent method developed by Kayano et al. (2010). In this approach, the dimension reduction is done independent of the purpose of clustering. Thus, after the projection of the data into a finite dimensional space we assume the coefficients of the expanded basis are multivariate fixed values. Then, the regular multivariate clustering approaches can be applied. Some studies perform further reduction/filtering by applying the functional principal component analysis FPCA (Peng et al., 2008).

One of the potential problems with this method is the possible loss of any discriminative features between clusters during the process of dimension reduction. Specifically, if using the principal component scores, we can easily miss the component that best classifies the data. For instance, assume the fifth functional principal component gives the best clustering characteristics, but the researcher uses the first four components to do the clustering approach. This method also has the issue of choosing the best basis expansion system that can do clustering properly. One approach to overcome the issue is to choose a saturated model. Some researchers used a well-designed basis system to do the clustering stage that are usually related to their algorithm (Serban and Wasserman, 2005). Nevertheless, the two-stage approach might fail if applied on data with few values or data on an irregular grid, because the basis coefficients of the sparse data

will have very high variance leading to unreasonable estimates. For irregular grids, a weighted variance can be assigned to get accurate estimates, however it is computationally expensive (James and Sugar, 2003).

The non-parametric approach uses a similar clustering technique as with multivariate data, like k-means, or hierarchical clustering, but with additional features to suit the functional data. For instance, Febrero-Bande et al. (2012) created the `fda.usc` package that performs k-means clustering on functional data. This method locates the centre of curves in a grouped data for each k group, then measures the distances between the centre and the curves to assign the groups, the two steps iterating until convergence. Tokushige et al. (2007) defined the distance between functions as a function of time t, and applied k-means clustering and fuzzy k-means clustering. In addition, there are other methods that are dynamic programming-based algorithms (Hébrail et al., 2010; Yamamoto, 2012).

Another aspect of the non-parametric approach can depend highly on the distance measures between the functions. For instance, Ieva et al. (2013) created a designed distance to measure the distances between the curves. The distance d is defined as $d = \sqrt{d_o^2 + d_1^2}$, where d_o corresponds to the distances between the curves, while d_1 corresponds to the distances between the first derivatives of the curves. They find the centroids of the clusters randomly and assign the curves to the nearest centroids to form k clusters. By solving the optimization problem, they reassign the centroids (that happen to be the means of the clusters) at the end of the process. Another example of a designed distance-based clustering technique for functional data is Peng et al. (2008).

The non-parametric approach usually needs a predetermined number of clusters. Besides, in some proposed methods, it can be subjective to the application, or can be computationally intensive.

Model-based clustering assumes the data come from a distribution that is a mixture of some distributions (Banfield and Raftery, 1993). However, with functional data, it is not as straightforward as in the multivariate case. The curves are first projected on a finite dimensional space,

then the basis coefficients or the principal component scores can be used for clustering. However, model-based clustering is different from two-stage clustering because in the model-based the two tasks are done jointly. The first model-based approach that was proposed specifically for clustering functional data is James and Sugar (2003). They used a mixed effect of natural cubic splines model which implies:

$$Y_j = S_j(\lambda_o + \Lambda\alpha_{z_j} + \gamma_j) + \varepsilon_j, \qquad j = 1,...n \tag{3.1}$$

Where: S_j is the spline basis matrix for the jth curve, $\varepsilon_j \sim N(0,R)$, and $\gamma_j \sim N(0,\Gamma)$. The model is an amended version of $Y_j = S_j\eta_j + \varepsilon_j$, but it assumes the basis coefficients η_j are random effects and thus can be written as $\eta_j = \mu_{z_j} + \gamma_j$, with μ_{z_j} representing the mean of cluster z. This can be further parametrized to $\mu_{z_j} = \lambda_o + \Lambda\alpha_{z_j}$, where λ_o and α_{z_j} are p- and h- dimensional vectors and Λ is a $p \times h$ matrix. The model parameters can then be estimated through an EM algorithm. Sugar and James (2003) suggested using the distortion function to select the number of clusters. This function is the average Mahalanobus distance between each coefficient vector η_i and its closest cluster's centre c_{z_i}[1]. This model can be applied for any functional form of data including sparse data, and data with an irregular set of time points.

Alternately, some model-based approaches use the functional principal component scores, and assume they follow a Gaussian distribution (Bouveyron and Jacques, 2011; Jacques and Preda, 2013). Bouveyron and Jacques (2011) introduced the `funHDDC` package that models and clusters the curves through their eigenspace projection. This is based on the functional principal component analysis conditional to some model parameters, and the probabilities of curves to belong to specific group. The idea is similar to the multivariate HDDC method (Bouveyron et al., 2007), but for the funHDDC approach they used a functional metric for the eigenspace projection.

On the other hand, Ray and Mallick (2006) proposed the first model-based Bayesian approach for clustering curves. They proposed a discrete wavelet transformation on the white noise

[1]The package `fitfclust` in R is the application of (James and Sugar, 2003)

Gaussian model $\boldsymbol{Y}_i = \boldsymbol{X}\beta_i + \varepsilon_i$. They assumed the basis coefficients β_i and the error variance σ_i^2 are the clustering parameters that can be expressed by θ_i (where $\theta_i = (\beta_i, \sigma_i^2)$). They assumed a Dirichlet Process Prior (DPP), and used Gibbs Sampling to infer the posterior distribution. The full model is as follows:

$$\boldsymbol{Y_i}|\theta_i \sim N(\boldsymbol{X}\beta_i, \sigma_i^2 \boldsymbol{I_m}), \tag{3.2}$$

$$\theta_1, \theta_2, ..., \theta_n \sim F,$$

$$F \sim D(\alpha, \boldsymbol{H_\phi}),$$

$$\theta_n|\theta_{-n}, \alpha, \phi = \frac{\alpha}{\alpha + n - 1}\boldsymbol{H_\phi} + \sum_{i=1}^{d_{n-1}} \frac{n_i}{\alpha + n - 1}\delta_{\bar{\theta}_i}, \tag{3.3}$$

where $\boldsymbol{H_\phi}$ is the base prior with the parameter ϕ, and α is the concentration. d_{n-1} is the number of pre-existing clusters of tied samples in θ_{-n} at the nth draw. The ith cluster has n_i tied samples that can be expressed by $\bar{\theta}_i = (\bar{\beta}_i, \bar{\sigma}_i{}^2)$

The advantage of this model is allowing k (the number of clusters) to be unknown. Thus, it can be inherent in the process of clustering and estimated from the data.

Suarez et al. (2016) followed a very similar approach by applying the DPP on the wavelet coefficients but they modelled the coefficients independently instead of placing priors jointly, by adding a hierarchical parameter to the model. Zhang et al. (2015) developed a method to cluster curves using elastic shape metric, which is based on joint registration and comparisons of shapes of curves. The resulting elastic-inner product matrix is modelled using a Wishart distribution, where the prior come from a Dirichlet Process (DP), and the posterior is sampled through a Markov Chain Monte Carlo (MCMC) procedure to infer the number of clusters and the clustering configuration. Some Bayesian methods are suitable for a specific data set. For instance, Scarpa and Dunson (2009) proposed that the distribution of functions come from a mixture of a parametric and non-parametric model. The parametric part comes from a prior knowledge on the data while the non-parametric part is based on the DP. In general, most Bayesian approaches are based on Dirichlet process mixture models, and they frequently use wavelet bases. Wavelets

are good in detecting quick changes in the curves, also the wavelet basis that represent significant parts of the curve are prioritized over the least significant in the smoothing equation.

Technically the proposed methods work well for the selected data sets in the papers. But generally they have not been tested for other data sets or against each other. Although there are numerous studies on clustering functional data, few provide their code to allow further applications. After searching for all the available R packages, below are some of the main functional data clustering techniques:

- `fda.usc` (non-parametric approach): includes many utilities for functional data analysis, it also provides k-means clustering. The number of groups must be predetermined, and this method searches for the locations that consists of grouped data to locate the curves' centres. Then it measures the distances to assign the groups, with the two steps iterating until convergence.

- `fitfclust` (model-based approach): the package is not available currently, but the codes are accessible. It implements the James and Sugar (2003) approach.

- `funHDDC` (model-based approach): clusters the functional data by modelling each group within a specific functional subspace (Bouveyron and Jacques, 2011). It is a high dimensional data clustering method, but for functional data. It clusters the functional data into group-specific functional subspaces. The procedure is based on functional latent mixture models, and the parameter estimation is through an EM algorithm.

- `funFEM` (model-based approach): allows us to cluster functional data by modelling the curves within a common and discriminative functional subspace (Bouveyron et al., 2015). It is a functional EM-like clustering algorithm. It clusters the functional data into discriminative functional subspaces F. This algorithm assumes that the basis coefficients follow a

mixture of Gaussian distributions.

- `Funclustering` or `funclust` (model-based approach): a multivariate functional clustering. It allows clustering multivariate functional data, while considering the dependence between curves. The algorithm is based on (Jacques and Preda, 2013).

- `fdakma` (non-parametric approach): It is based on a k-means alignment algorithm that simultaneously clusters and aligns the functional data that show phase and amplitude variation. The method uses the original function and its first derivatives. For more information go to Sangalli et al. (2010), or Parodi et al. (2014).

- `fdasrvf` (non-parametric approach): It is an extension to the `fdakma` method, performs the alignments of curves using the square-root velocity framework. This framework introduces the elastic analysis of curves through phase and amplitude separation (Marron et al., 2015).

- `funcy`: combines seven model-based methods in one framework to cluster functional data (Yassouridis and Leisch, 2017). The methods are:

  - `fitfclust`: based on a functional mixed mixture model, allows irregular measurements. (James and Sugar, 2003)
  - `distFPCA`: based on a distance measure, allows irregular measurements. (Peng et al., 2008)
  - `iterSubspace`: based on a subspace projection, allows irregular measurements. (Chiou and Li, 2007).
  - `funclust`: based on a functional mixed mixture model. (Jacques and Preda, 2013)
  - `funHDDC`: based on a functional mixed mixture model. (Bouveyron and Jacques, 2011)

- `fscm`: based on a functional mixed mixture model. (Jiang and Serban, 2012)
- `waveclust`: based on a functional mixed mixture model. Wavelet basis is the only possible. (Giacofci et al., 2013)

3.3.1 Selected Functional Clustering Methods for Comparisons

For the purpose of comparing our proposed clustering approach (Chapter 4) with existing methods, we choose some of the above methods for the comparisons. Our choice is based on the availability of the algorithm, and the different categories of the CFD approaches. Thus, we consider `funHDDC` that belongs to the model-based category, and `fda.usc` that belongs to the nonparametric category. However, there is no convenient two-stage algorithm. Therefore, considering the above techniques, we have programmed an R function based on Abraham et al. (2003) and named it as `B-splines-Km`. The function applies a B-splines smoothing to the data, then uses the basis coefficients to carry out the regular k-means. It is a two-stage clustering technique where the basis expansion is done independently of the clustering purpose. Nevertheless, it can give good results if the smoothing technique was selected properly. The authors also mentioned the reason behind choosing B-splines rather than any other basis systems. They noted that B-splines can capture a lot of different shapes with only a few coefficients, and these bases have local support, thus every coefficient represents part of the time domain. Also the coefficients of B-splines are robust as their estimation is not affected by outliers in other parts of the time domain. These features are important in analysing some patterns of functional data.

Furthermore, we have developed another algorithm as an extension to the above approach. It is based on a dimension reduction through B-splines as a first step, then it further applies FPCA to use information that explains the variation in the data. We extract the first k eigenfunctions[2] to form a new k-dimensional multivariate dataset X. Finally, we use `mclust` to cluster the resulting data by the model-based clustering technique (VVV) [3]. We have named the algorithm as `FPCA-mbc`. This method combines the power of FPCA and the ability of model-based tech-

[2] k = predetermined number of clusters k

[3] The ellipsoidal model with varying volume, shape, and orientation VVV is the most general model among the other options and is used as a default for starting the EM algorithm.

nique in clustering any data set. The combination of these two clustering tools could be seen as a straightforward extension. However to our knowledge, there is no similar functional data clustering approach in the literature.

As mentioned above, these chosen methods cover the different aspects of clustering functional data, and their codes are available in R (or can be easily programmed). In addition, `fda.usc` and `B-splines-Km` are popular CFD techniques, while `funHDDC` and `FPCA-mbc` are considered to be robust methods. Thus we assume the selected methods are reasonably appropriate for a comparison scheme and simulations.

3.4 Clustering Evaluation Measures

Cluster validation is an integral part of any cluster analysis. Cluster validation refers to examining the quality and goodness of the clustering. As there is a vast collection of clustering methods to choose from, there is a need to evaluate them and arrive at the most appropriate clustering approach. According to Hennig et al. (2015), there are a variety of approaches to evaluate clustering results. The most common approach is calculating cluster evaluation criteria such as the Rand index, adjusted Rand index, Jaccard indicator, and accuracy rates from the confusion matrix. The evaluation indicators help the researcher to decide between the different clustering approaches and to assess to what extent the clustering results are informative and reliable. An alternative approach that has become more popular recently is the evaluation of the stability of a clustering. It is based on repeating the clustering several times and if they yield similar results, the clustering method is considered reliable. More details about clustering stability is in Section 5.1. In addition, data visualization is a method that is basically used for data exploration, which can also be used for cluster validation.

Among the wide range of available cluster validation indices, in this thesis we we will use the adjusted Rand index (ARI) and the correct classification rate (CCR) and.

The adjusted Rand index is based on the Rand index RI (Rand, 1971). Rand index is defined as the number of pairs of objects that are either in the same group or in different groups in both partitions divided by the total number of pairs of objects. It takes values between 0 and 1, where 1 indicates a perfect match between two partitions. An issue with the Rand index is that the expected value of two random partitions does not take a constant value. Thus, Hubert and Arabie (1985) proposed the ARI to correct this issue by assuming the generalized hyper-geometric distribution as the model of randomness. The adjusted Rand index is recommended for measuring agreement even when the partitions compared have different numbers of clusters.

The correct classification rate is often referred to as the accuracy rate, and can be calculated from the confusion matrix. A confusion matrix is a standard cross-tabulation table for summarizing the performance of a classification algorithm. It requires the user to have knowledge of the true clusters to figure out the differences between the clustering results and the true clusters. CCR refers to the total correct outcomes (predictions) among the total outcomes (predictions) made. It takes values between 0% and 100%, the larger the accuracy rate the better the clustering technique. From the confusion matrix, we can also calculate the misclassification rate or what is sometimes called the error rate and it is equal to 1 - CCR. To be able to use the confusion matrix correctly, the resulting clustering label assignment must match with the standard true cluster labels. However, a recent work done by Chacón (2021) has compared CCR and ARI in terms of their properties and differences. The author suggested that it is also possible to use CCR to compare any pair of partition of the same dataset without a need for true clusters, by permuting the cluster labels to find the optimal match between labels [4].

Considering the properties of each evaluation measure, we will use the correct classification rate (CCR) for assessing the clustering results in Chapter 4. While, we will use the adjusted Rand index (ARI) in Chapter 5 and 6. Since in Chapter 4 we consider the true clusters are known and they will be used to examine the performance of the clustering methods. Whereas, in Chapter 5 and 6 the approach is to compare two partitions without knowing the true clusters

[4]This note was added only after the final revision of the thesis.

of the data.

3.5 Chapter Summary

This chapter provided the basic definitions of clustering analysis and a discussion of clustering procedures. It gave a brief overview of the clustering approaches used for multivariate data. However, the spectral clustering approach was discussed in more detail, as it will be used as a basis for our proposed clustering method in the next chapter.

In addition, it focused on clustering functional data analysis and explained the most popular algorithms, listing the available R packages that provide clustering analysis for functional data. Accordingly, the chapter ended with some chosen functional data clustering methods that will be useful for clustering examinations and validations. The chosen clustering methods will take part in the simulation to compare the performance of the chosen methods against each other and to evaluate the performance of our approach among the other methods.

In this chapter we aimed to review clustering analysis and attempted to look at the major algorithms from different angles to explore their strengths and weaknesses. However, it is a challenging task to summarize all the clustering approaches due to the diversity of algorithms and applications in the different research areas.

Chapter 4

A Spectral Clustering Framework for Functional Data

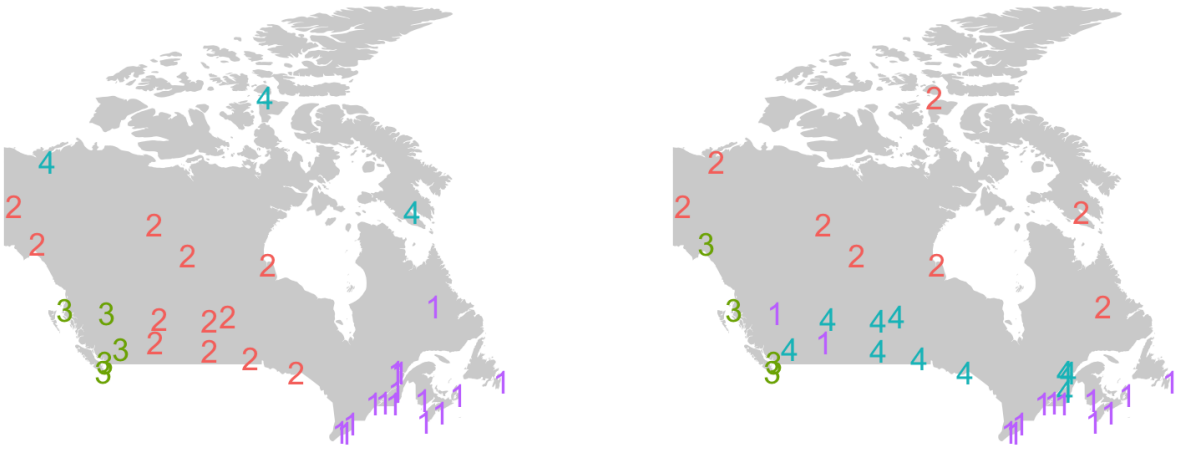
In this chapter we present a new framework for clustering functional data. Our clustering framework is built on the spectral clustering approach and is flexible enough to exploit higher order features of curves, including derivatives. The first section shows the motivation behind our proposal. Section 4.2 details our proposed technique starting with the smoothing technique, the distance measure and the clustering model. Section 4.3 evaluates the functional spectral clustering technique on the Berkeley growth data beside the chosen functional data clustering techniques in this study. Section 4.4 discusses the perturbation theory, the theoretical basis of our approach.

4.1 Motivation

Clustering functional data (CFD) has been an active area of research in recent years. One of the major challenges for clustering functional data is the lack of clear distributional theory for functional data. In addition, due to the high dimensionality of functional data, the clustering method becomes more challenging and computationally intensive. The structure and format of functional data are very diverse and complex.

Taking into account the complex structure of the data, the spectral clustering (Section 3.2.1) algorithm has been successfully applied to cluster high dimensional data embedded in a nonlinear manifold. The Swiss roll example is commonly used to illustrate the power of the spectral method which outperforms most other standard clustering methods such as k-means or model based clustering. The spectral clustering method has been shown to provide good results without the need to make strong assumptions about the data and its implementation is straightforward. In comparison, model-based clustering approaches are much more computationally expensive and rely on distributional assumptions, which are very hard to verify in the context of functional data.

However, applying spectral clustering directly on the Canadian weather data (Figure 4.1b), does not give reasonable results when compared to the geographical distribution of the Canadian cities as stated in (Ramsay and Silverman, 2005) (Figure 4.1a). The geographical distribution divides the cities into north, east south, west south, and inland cities based on mainly the location of the 35 cities, that might share similar characteristics such as annual temperature, and annual precipitation.



(a) (b)

Figure 4.1: The Canadian map shows locations of 35 cities involved in this study. Panel (a) shows the geographical distribution of these cities according to (Ramsay and Silverman, 2005), (b) shows the cities in 4 clusters when using the spectral clustering method.

Thus, considering the flexibility of the spectral algorithm, and its ability to resolve complex structures through a nonlinear dimension reduction procedure, we propose to develop a

framework to implement the spectral clustering method specifically for functional data. To our knowledge this is a completely new framework that utilizes the original trajectories as well as their derivatives.

4.2 Functional Spectral Clustering Approaches (FSC-S)

This section presents our two-stage functional spectral clustering approach (FSC-S) in its general form, which assumes the number of clusters in the data is given. It is considered as the default approach, while Chapter 6 will present the extended approach.

As was mentioned in Section 3.3, one of the methods of clustering functional data is the two-stage clustering approach. The two-stage approach is based on splitting CFD technique into two independent steps. The first step is applying smoothing and basis expansion to project the data into a functional space. The second step assumes that the evaluated coefficients of bases from the smoothing are multivariate fixed values. Thus, we have named the proposed algorithm as FSC-S, where 'FSC' stands for Functional Spectral Clustering, while '-S' shows that it is a two-stage approach where Smoothing is conducted first.

4.2.1 Smoothing

Suppose there is a data matrix S in $\mathbb{R}^l$. The matrix consists of n vectors (individuals), and the data are measured over a domain (usually time). Unlike the multivariate spectral clustering where we can directly apply the method on the data, in our case, we first project the data on a reduced dimensional space through a smoothing technique.

In this study, the default smoothing technique relied on B-splines. B-splines are very popular basis functions, which have been used earlier for univariate regression and for FDA. Their main advantage is the ability of performing fast computations and the flexibility in controlling the smoothness. The B-spline basis is often used for non-periodic functions, but can also be used for periodic functions. For more details about B-splines and more of their features refer to

Section 2.3 and Section 3.3.1 respectively.

We propose to use a general smoothing technique with a linear combination of B-splines of order 4 or above with a knot at every time point in the data (i.e. a saturated model). The reason behind choosing the basis to be at least of order 4, is for getting continuous first and second derivatives that are of interest in our clustering approach. The most popular choice is order 4, however in some data examples order 5 or order 6 might fit the data better.

In regards to the number and locations of knots, we can either place the knots evenly or place more knots where the data vary rapidly, however, these techniques are ideal for specific data sets and might not work in general. Thus, to reduce the computation time, we use a saturated model with fitting a smoothing parameter λ to penalize the representation of the data. Choosing the appropriate λ can be done through generalized cross-validation (GCV) (see Section 2.3). As was discussed in detail in Section 2.3, each smoothing problem must be treated individually. Indeed, smoothing is a crucial part of clustering and there are many choices of smoothing parameters. A saturated model is usually a preferred smoothing option for clustering in several applications. There are however, other appropriate smoothing options specific to a given data set. Note that all the functional data in this thesis have been smoothed based on a saturated B-splines model.

4.2.2 Distance Measure

Carrying out the above smoothing technique creates the curves $\mathbb{f} = \{f_1, f_2, ..., f_n\}$ over time. Then, the distance between the curve f_i and the curve f_j $(i \neq j)$ is calculated. A critical choice here, is how to determine the distance between curves in a set of functional data. Although, there have been many studies focused on calculating distances for functional data, we utilize the metric functions that are provided by the `fda.usc` package (Febrero-Bande et al., 2012). The package hosts several metric and semi-metric functions to measure the distances between curves. Considering the $\mathscr{L}_2$ spaces, we used `metric.lp` metric function that is based on Simpson's Rule. The metric function can be written as:

$$||d|| = \sqrt{\frac{1}{\int_T w(t).dt} \int_T (f_1(t) - f_2(t))^2 w(t).dt}, \tag{4.1}$$

where w represents weights (which by default are 1). The second metric function we used is `semimetric.deriv` that calculates distances between derivatives of order q; the metric function is written as:

$$||d_q|| = \sqrt{\frac{1}{T} \int_T \left[f_1^q(t) - f_2^q(t)\right]^2 .dt}. \tag{4.2}$$

The first reason behind this choice is the ease and the speed of the process. Second, it is flexible and can calculate the distances between the raw trajectories, equation (4.1), or their derivatives, equation (4.2), which are of interest. In addition, Tzeng et al. (2016) performed simulations to compare different distance measures for functional data based on true distances between the curves. The authors showed that the $\mathscr{L}_2$ distance metric is among the best measures and is unbiased in many situations as there are no missing data.

Taking into account the different ways of measuring the distance, and the importance of both the curves and their derivatives in the functional data analysis, we split our general approach into mainly: FSC-S(D_o) and FSC-S(D_1). The former, FSC-S(D_o), refers to the use of the original trajectories to create the distance matrix, while the latter, FSC-S(D_1), refers to the use of rate of change curves to create the distance matrix. Nevertheless, we have noticed from experience that in some cases when the distance matrix comes from the accelerations (second derivatives), the proposed clustering method can distinguish the clusters even better. Thus, we have also introduced FSC-S(D_2), to be used in some examples.

4.2.3 The Model

The pairwise distances, d_{ij}, help constructing the similarity graph (also called the affinity matrix or the weight matrix), which in turn models the neighbourhood relationships between the curves. There are three major approaches to transform pairwise distances into a graph (Section 3.2.1). In our model, we have chosen to use the fully connected graph which is based on using

the Gaussian Kernel estimation. The other two approaches assume that only neighbour data points are connected. The Gaussian Kernel estimation is a common and a more straightforward approach for constructing the similarity graph, besides we prefer assuming that all the curves are connected with each other in the data. According to Von Luxburg (2007), the ε-neighbourhood graph is more vulnerable to inappropriate choices of the parameter ε, and thus it is not a preferred similarity graph. The author also added, it has not been yet proved theoretically whether the choice of similarity graph will affect the results of spectral clustering. The Gaussian Kernel estimation is defined by:

$$A_{ij} = \exp(-||f_i - f_j||^2/2\sigma^2), \tag{4.3}$$

where $i \neq j, A_{ii} = 0$, while σ is a scaling parameter that is chosen by the researcher (most often $\sigma = 1$). However, the parameter σ in the Gaussian Kernel function plays an important role as it controls the width of the neighbourhoods between the curves. It is a reference distance, at which it defines the similar curves and the dissimilar curves. Thus, such parameter should come from the domain of the data instead of considering a rule of thumb value. In this model, we set σ to be the standard deviation of the elements of the distance matrix, while in Chapter 6, the model will consist of a flexible σ that can take a range of values.

In our model, we replace σ^2 by σ and represent the similarity graph $\mathbb{A} \in \mathbb{R}^{n\times n}$ as:

$$A_{ij} = \exp(-||f_i - f_j||^2/2\sigma), \tag{4.4}$$

The next step is constructing the diagonal matrix $\mathbb{D} \in \mathbb{R}^{n\times n}$, whose diagonal elements are calculated by, $D_{ii} = \sum_{j=1}^{n} A_{ij}$. Then, the Laplacian matrix $\mathbb{L} \in \mathbb{R}^{n\times n}$ is constructed by:

$$\mathbb{L} = \mathbb{D}^{-1/2}\mathbb{A}\mathbb{D}^{-1/2}. \tag{4.5}$$

The matrix $\mathbb{L}$ has n eigenvectors, but we are interested in only the first k eigenvectors $v_1, ..., v_k$, where k represents the number of clusters and is known in advance. The Laplacian matrix arrange the eigenvectors such that the eigenvectors that show most variation between the data come first. Stacking these k eigenvectors together forms the matrix $\mathbb{V}$ of size $n \times k$. Then,

the rows of matrix $\mathbb{V}$ are normalized to create the matrix $\mathbb{Y}$ by finding:

$$Y_{il} = V_{il}/(\sum_{l=1}^{k} V_{il}^2)^{1/2}. \tag{4.6}$$

This step shrinks the spread of points in a class to create more compact clusters. Assuming each row of $\mathbb{Y}$ is a point in $\mathbb{R}^k$, for $i = 1,...,n$, we cluster the points into k clusters by using the k-means algorithm. Finally, we assign the original curves to their corresponding clusters with $f_i = \{l|y_i \in C_l\}$, where every row y_i represent a curve f_i that belongs to cluster C_l.

Our functional spectral clustering algorithm borrows ideas from Ng et al. (2002) and it is based on the perturbation theory. A discussion of perturbation theory is given in Section 4.4. To the best of our knowledge, no previous research has used spectral clustering to cluster functional data. Thus, we want to examine the efficiency of our proposed functional spectral method and compare it to other existing methods. The main difference between classical spectral clustering and this version appears in the smoothing and the distance calculations.

4.3 Application of FSC-S on the Berkeley Growth Data

The Berkeley growth data (Section 2.2.2) has been widely used in research on clustering functional data, most often the data were clustered according to children gender. A common smoothing technique is using a B-spline basis of order 6 in a saturated model with $\lambda = 10^{-1/2}$. This smoothing model was also suggested by Ramsay and Silverman (2005). It is mainly designed for minimizing the noise and obtaining the best representation of the smoothed trajectories. However, after applying the generalized cross validation on the same model but with varying λ, we found out that the lowest GCV occurs when λ is between 10^{-10} and 10^{-1} as shown in Figure 4.2. Smoothing with $\lambda = 10^{-10}$ gives over-fitted derivatives that will possess high degrees of noise, while we need smoothed first and second derivatives as well as the original trajectories. Thus, we avoid $\lambda = 10^{-10}$ and consider $\lambda = 10^{-1}$ to be the appropriate choice that will project the proper structure of the data for both the trajectories and their derivatives. The final

choice of smoothed trajectories and their first and second derivatives are displayed in Figure 4.3. The figure consists of the height curves of children (original trajectories), the rate of change in height (or growth rates) (first derivatives), and the acceleration in height (second derivatives). For the purposes of clustering, we will proceed with our choice of smoothing, i.e. smoothing with $\lambda = 10^{-1}$. However, later we will also present the clustering results based on the common smoothing model that is with $\lambda = 10^{-1/2}$, mainly because we want to evaluate the impact of smoothing on the clustering results.

In the growth data, some individuals reach puberty earlier than others, which would create some phase variation in the trajectories. Thus, it is of interest to apply the functional spectral clustering methods to cluster the data and to investigate how the resulting clusters are associated with sex. Our main goal is to optimize the functional spectral clustering technique and assess its overall performance.

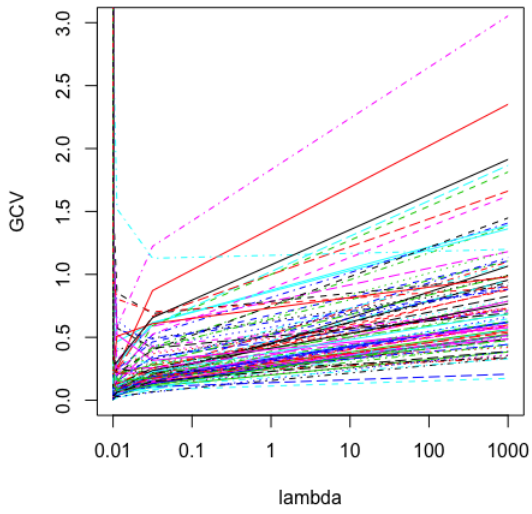


Figure 4.2: GCV curves for the Berkeley growth data show the dip when λ is between 10^{-10} and 10^{-1}.

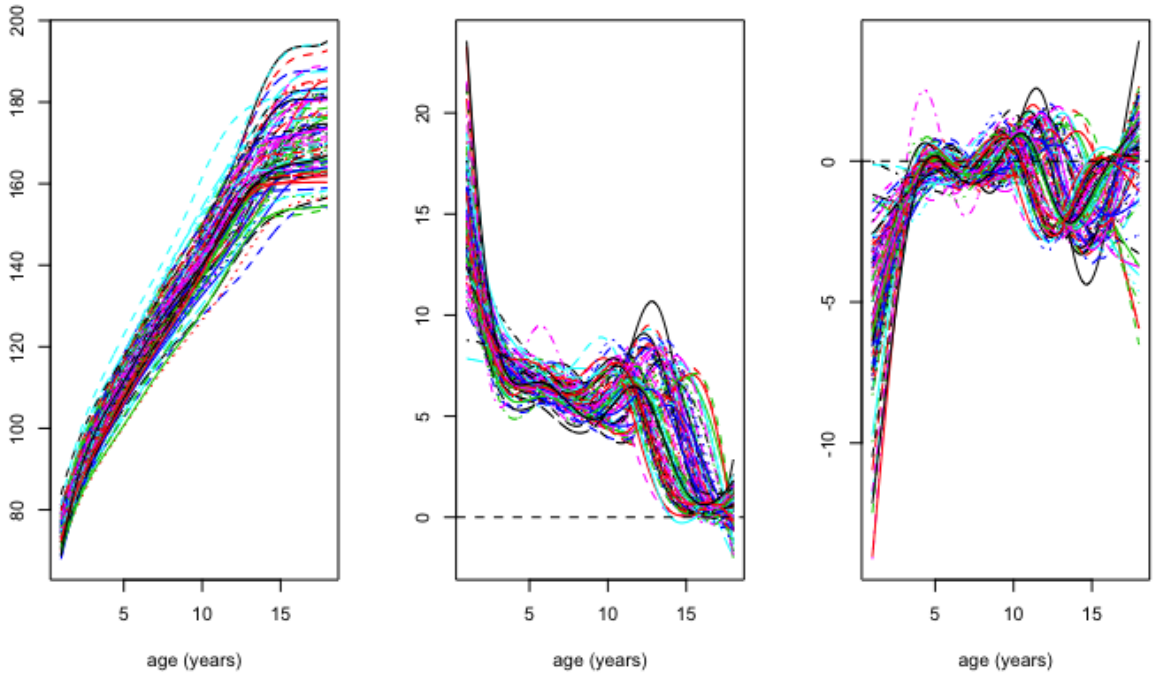


Figure 4.3: The Berkeley growth data as smoothed curves (left), the rate of change in heights (middle), and the accelerations in heights (right).

To assess the performance of our proposed method on the Berkeley growth data, we first apply FSC-S(D_o) with setting $k = 2$. The results are shown in Figure 4.4 on both the original height curves and the rates of change curves. The results visually look plausible and the misclassification rate is not high. Yet, applying FSC-S(D_1) leads to better results and the misclassification rate is lower (Figure 4.5).

The accuracy rates (CCR) are calculated for our proposed methods and the other CFD methods (mentioned in Section 3.3.1) by comparing the results of clustering against the natural grouping of girls and boys in the data. As per Table 4.1, the accuracy rates for FSC-S(D_o), FSC-S(D_1), and FSC-S(D_2) are 86%, 94%, and 85% respectively.

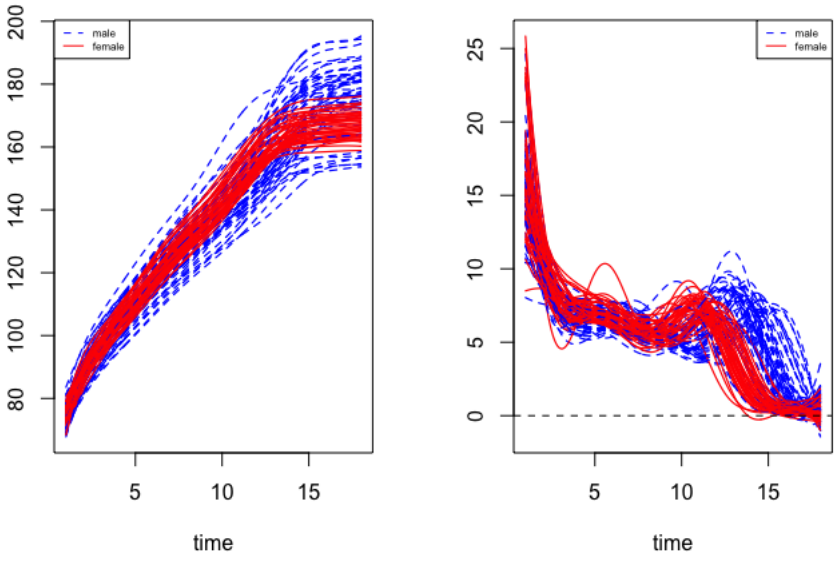


Figure 4.4: Clustered original trajectories (left), and first derivatives (right), when using FSC-S(D_o) with misclassification rate = 14%. Note the dashed blue curves and the red curves represent the clustering results and not the true male and female curves.

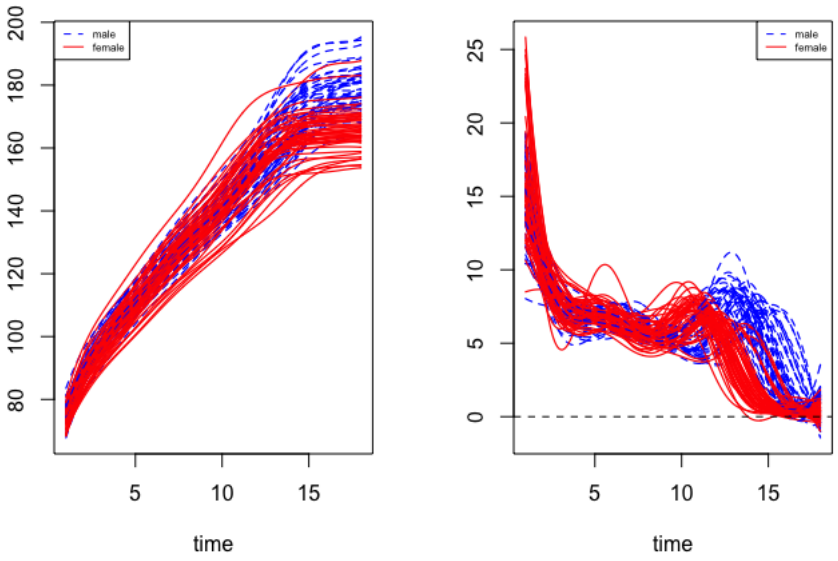


Figure 4.5: Clustered original trajectories (left), and first derivatives (right), when using FSC-S(D_1) with misclassification rate = 6%. Note the dashed blue curves and the red curves represent the clustering results and not the true male and female curves.

Looking into more details, Figure 4.6 illustrates why the first derivatives gives better results. In this figure there are some curves coloured in green and they represent the misclassified female curves to be considered as male when applying FSC-S(D_o). However, we can avoid some of this misclassification when applying FSC-S(D_1). This is because the rates of change can discriminate females from males more efficiently than the original curves by employing the puberty information. Both the height curves and the acceleration curves cannot reveal as much information as the growth rates.

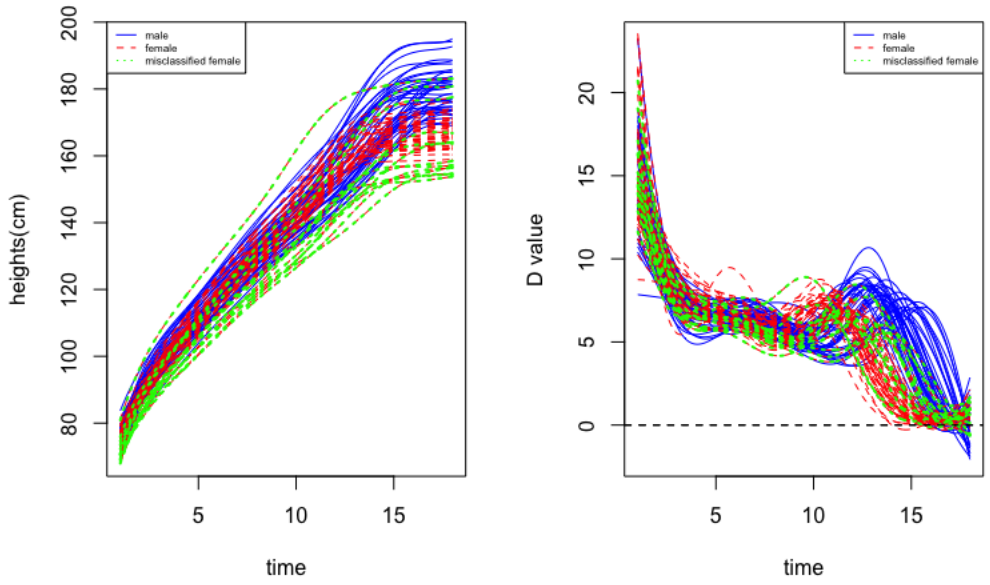


Figure 4.6: Resulting clusters of male (blue) and female (red) based on FSC-S(D_1), and additional green curves that represent the 13 misclassified female to be considered as male in FSC-S(D_o).

Considering the other CFD methods, we used the same smoothing model and set $k = 2$ for all. The applications of the competing methods were not limited to the original trajectories, but were also carried out on the first derivatives $D(F)$. There are a few reasons behind this procedure. First, we have noticed that the majority of researchers make use of the first derivatives to cluster the Berkeley growth data. Second, to compare the methods fairly, we consider using the first derivatives in this example as we also use the first derivative in FSC-S(D_1). As shown in Table 4.1, the first column displays the accuracy rates of the original method and the second

column displays the accuracy rates when using the first derivatives $D(F)$. Also note that Table 4.1 consists of two main columns where $\lambda = 10^{-1}$ and $\lambda = 10^{-1/2}$. The first one represents the smoothing parameter obtained from GCV, while the second is a standard smoothing parameter used by other researcher. In fact, we have explored a range of λ values lies between 10^{-10} and 10^{1}. The clustering results based on the other smoothing parameters are not very different from the results that are displayed in Table 4.1.

According to Table 4.1 at $\lambda = 10^{-1}$, in general FSC-S(D_1) outperforms all the other clustering methods with a CCR of 94%, while FSC-S(D_o) comes in second. FunHDDC and FPCA-mbc perform similarly, while FD-Kmeans and B-splines-km show the lowest CCR. Moving to the other side of Table 4.1, FSC-S(D_1) again shows the highest CCR followed by FSC-S(D_o) and FPCA-mbc, while the accuracy rates of FunHDDC are similar under the two λ values. Once again, FD-Kmeans and B-splines-km are performing similarly and slightly better at $\lambda = 10^{-1/2}$. Also note that apart from FunHDDC, the competing methods do better when using the first derivatives $D(F)$ under the two smoothing models.

Based on the standard smoothing model with $\lambda = 10^{-1/2}$, FSC-S(D_1) obtains good accuracy rates but lower than our optimal choice (when the accuracy rate is 94%). This difference in accuracy rates demonstrates the effect of smoothing on the clustering results of some methods. Note that FSC-S(D_o) is not affected by changing the smoothing parameter, because the distance matrix remains the same in the two smoothing models, since the coefficients of the original curves are less vulnerable to change under the two smoothing models. The smoothing choice also plays an important role in the clustering results of FPCA-mbc; the reason behind its better performance when $\lambda = 10^{-1/2}$ relates to the functional principal components scores which give better representation of the variation between the curves in this smoothing model. As was discussed in Section 3.3.1, we set the number of functional principal components to be equal to number of clusters, k. We have also observed that setting functional principal components $= k$ in this example leads to the best results for FPCA-mbc, while choosing more or fewer components gives lower accuracy rates in both the original method and the first derivatives.

The accuracy rates of FD-Kmeans and B-splines-km show that they perform similarly in this example with B-splines-km performing slightly better. Both clustering methods involve k-means in their procedure. Whereas FD-Kmeans carries out the smoothing and the k-means clustering simultaneously, B-splines-km smooths the data first, then clusters the coefficients of the smoothed curves through k-means.

Finally, in both smoothing models, the accuracy rate for FunHDDC is just 75%, which is relatively low. Although Jacques and Preda (2014b) mentioned that for the Berkeley growth data the accuracy rate of FunHDDC is 96.77%, we could not replicate their high accuracy. It should be mentioned that we have used the default model '$A_{kj}B_kQ_kD_k$' of FunHDDC [1], while their approach involves a number of models, and it was not mentioned which model led to the high accuracy rate. It is always possible to choose multiple models that lead to several results, but it is computationally intensive. Another possible reason behind the difference in the accuracy rates is the effect of the smoothing model that was used in their application and in our application.

In this example we attempted to cluster the first derivatives besides the original curves for the chosen CFD methods. However, we will not consider using the first derivatives for these methods in the next chapters. In general, if the derivatives are important and more informative, then clustering the derivatives using the same technique would lead to better accuracy rates. But that means the user chooses to cluster the derivatives directly instead of the original curves. In all the competing CFD methods, clustering the derivatives is not built-in in their techniques. Therefore, choosing to cluster the derivatives besides the original methods generalizes their clustering algorithms beyond their original design and adds more computational load to the comparisons and simulations that we will implement in Chapter 7.

[1] In the FunHDDC there exist a number of submodels that are defined based on different parameters.

Method	CCR			
	$\lambda = 10^{-1}$		$\lambda = 10^{-1/2}$	
	original method	using $D(F)$	original method	using $D(F)$
FunHDDC	0.75	0.51	0.75	0.52
FD-Kmeans	0.63	0.87	0.66	0.81
B-splines-km	0.64	0.89	0.67	0.88
FPCA-mbc	0.76	0.80	0.86	0.85
FSC-S(D_o)	0.86	-	0.86	-
FSC-S(D_1)	0.94	-	0.90	-
FSC-S(D_2)	0.85	-	0.88	-

Table 4.1: Accuracy rates for clustering the Berkeley growth data according to two different smoothing parameters.The left side shows CCR results when $\lambda = 10^{-1}$ and the right side shows CCR results when $\lambda = 10^{-1/2}$.

4.4 Perturbation Theory

Perturbation theory considers the behaviour of the ideal case and the affect of adding perturbation to the ideal case. The ideal case refers to the data set that consists of k clear clusters, in which all points in one cluster are very similar and they are very different to points in other clusters. Consider the matrix $A \in R^{n\times n}$ to be a symmetrical ideal matrix, and its perturbed version is $\tilde{A}$, where $\tilde{A} = A + H$. The matrix H represents the perturbation added to the ideal case, however, we usually don't know what A is. Therefore we raise the question: how much does H affect $\tilde{A}$? To answer this question we will consider two theorems.

The first theorem is the Weyl inequality in matrix theory, developed by Weyl (1912). For more recent literature we refer to Fan (1950), Bhatia (1987), Kolotilina (2000), and Tao (2010). It addresses the changes that occurs in the eigenvalues of a perturbed matrix, and it is stated as follows:

Theorem 1 *Weyl (1912) For $i = 1, ..., n$:*

$$\lambda_i(A) + \lambda_n(H) \leq \lambda_i(\tilde{A}) \leq \lambda_i(A) + \lambda_1(H).$$

Note that, $\lambda_1(H)$ represents the largest eigenvalue of H and $\lambda_n(H)$ represents the smallest eigenvalue of H.

The second theorem is the Davis-Kahan Theorem (Davis and Kahan, 1970). It is considered to be the fundamental theory of the perturbation approach to spectral clustering. This theorem is commonly used in statistical procedures to bound the distances between sets of subspaces spanned by eigenvectors. It has been widely discussed in literature and for further readings, we refer to Stewart and Sun (1990) and Bhatia (1997), and it is stated as follows:

Theorem 2 *Davis and Kahan (1970) Consider the symmetric matrix A, and the perturbation matrix $H \in R^{n \times n}$. Let $\tilde{A} = A + H$ be a perturbed version of A, where $A = E_o A_o E_o^T + E_1 A_1 E_1^T$, and $\tilde{A} = F_o \tilde{A}_o F_o^T + F_1 \tilde{A}_1 F_1^T$, with $[E_o, E_1]$ and $[F_o, F_1]$ orthogonal. Also, assume the eigenvalues of A_o are contained in $[a, b]$, while the eigenvalues of $\tilde{A}_1$ are contained in $(-\infty, a - \delta) \cup (b + \delta, +\infty)$ for some $\delta > 0$, then:*

$$||F_1^T E_o|| \leq \frac{||F_1^T H E_o||}{\delta}.$$

Following arguments discussed by Von Luxburg (2007), and Xie (1997), to interpret spectral clustering techniques from a perturbation theory point of view, we answer the above question by looking at two aspects: How are the eigenvalues of $\tilde{A}$ affected by H? And how is the eigenspace of $\tilde{A}$ affected by H?

Theorem (1) shows that the ordered eigenvalues of the matrix $\tilde{A}$ are fairly stable under small perturbation. Now considering the effect of the perturbation on the eigenspaces, an eigenspace of a matrix is a span of some eigenvectors of that matrix. If we decompose the matrix A into its action on an eigenspace Φ and its orthogonal complement Φ^c, and following the spectral theorem, it states that for any matrix M, $M = v \lambda v^T$, with $v_1, ..., v_n$ orthonormal basis (Xie, 1997).

Then, we can write A as:

$$A = \underbrace{E_o A_o E_o^T}_{\Phi} + \underbrace{E_1 A_1 E_1^T}_{\Phi^c}. \tag{4.7}$$

Similarly, we can decompose $\tilde{A}$ into its action on an eigenspace $\tilde{\Phi}$ and its orthogonal complement $\tilde{\Phi}^c$, so that:

$$\tilde{A} = \underbrace{F_o \tilde{A}_o F_o^T}_{\tilde{\Phi}} + \underbrace{F_1 \tilde{A}_1 F_1^T}_{\tilde{\Phi}^c}, \tag{4.8}$$

where E_o and E_1 are orthonormal bases for Φ and Φ^c respectively. Likewise, F_o and F_1 are orthonormal basis for $\tilde{\Phi}$ and $\tilde{\Phi}^c$ respectively. Also note the eigenspaces Φ for A and $\tilde{\Phi}$ for $\tilde{A}$ represent the eigenvectors that are of interest[2]. If we assume that any vector in Φ can be written as $E_o\alpha$ (where α has same dimension as Φ), then the projection of this vector on the perturbed eigenspace $\tilde{\Phi}$ is $F_o F_o^T E_o \alpha$, and the distance between the two vectors is:

$$||E_o\alpha - F_o F_o^T E_o \alpha|| = ||(I - F_o F_o^T)E_o\alpha|| = ||F_1 F_1^T E_o \alpha|| = ||F_1^T E_o \alpha||. \tag{4.9}$$

Therefore the distance between the eigenspaces Φ and Φ^c can be calculated by the Frobenius norm of $F_1^T E_o$. The Frobenius of any matrix M is: $||M|| = \sqrt{\sum_i \sum_j m_{ij}^2}$. Yet, there is a condition that needs to be satisfied. The eigenvalues of Φ and the eigenvalues of $\tilde{\Phi}^c$ must be well separated by a value δ. For instance, if the eigenvalues of Φ are all contained in the interval $[a,b]$, then we need the eigenvalues of $\tilde{\Phi}^c$ to come from the interval $(-\infty, a-\delta) \cup (b+\delta, +\infty)$. In fact, we want the eigenspaces $\tilde{\Phi}^c$ and Φ to be as far as possible from each other, so that when the perturbation H occurs the δ value is still large enough relative to H. Looking at Theorem (2), it states that the distance between the two eigenspaces will be bounded by $||F_1^T H E_o||/\delta$. From here we notice that the larger δ or the smaller H, the closer the eigenspaces are from each other. If this is the case, then we can use $\tilde{\Phi}$ to approximate Φ.

Reflecting the above discussion on our spectral clustering approach, consider A to be the

[2] eigenvectors that correspond to the first few very large eigenvalues compared to the rest of eigenvalues.

ideal matrix of size $n \times n$ that consists of k clear components (clusters). In this case, all entries between-clusters are very low (say $= 0$), and all entries within clusters are very high. Now, consider the perturbed version $\tilde{A}$ that consists of the same k components, but these are not totally disconnected. Due to noise the between-cluster values will no longer be 0's and thus create some edges between the k clusters. In the ideal case, L will consist of k orthogonal eigenvectors $v_1, v_2, ..., v_k$ that hold all the information about the clusters in the data. Next, these vectors are renormalized to have a unit length, and k-means will trivially find the true clusters. Going to the real case, $\tilde{L}$ will consist of some $\tilde{v_1}, \tilde{v_2}, ..., \tilde{v_k}$ that are supposed to reveal the same information about the clusters in the data. The eigenspace $\tilde{\Phi}$ does not coincide fully with Φ but they must be similar "enough" to get plausible results. To measure this similarity between Φ and $\tilde{\Phi}$, we look at dissimilarity between Φ and $\tilde{\Phi}^c$ based on Davis-Kahan theorem. Thus, we want the eigenvalues of A_o and $\tilde{A}_1$ to be as far as possible (large δ). Then, we bound the distance between the eigenspaces $||F_1^T E_o||$ by their perturbed version $||F_1^T H E_o||$ over δ. Note that, if $\tilde{\Phi}$ fully coincides with Φ, then δ will turn out to be the eigengap $|\lambda_{k+1} - \lambda_k|$ (The eigengap will be discussed in more detail in Chapter 6).

4.5 Chapter Summary

In this chapter we have introduced FSC-S, a two-stage functional spectral clustering technique for clustering functional data. This clustering technique can be categorized as FSC-S(D_o), FSC-S(D_1), and FSC-S(D_2) based on the distance metric of original trajectories, first derivatives and second derivatives respectively. We have given a simple example initially to demonstrate that the regular spectral clustering technique cannot perform well when directly applied to functional data. Thus, we require spectral clustering to accommodate the curvature structure of the functional data. This is mainly done by smoothing and basis expansion of the data. Smoothing techniques and distance measures are two critical choices and they play an important role in the FSC-S methods.

We have examined the performance of our proposed method on the Berkeley growth data. We found that the first derivatives that reflect the rate of growth in children cluster the boys' and girls' groups better than the original curves (heights of children). As is already known, the first derivatives of the growth data hold information about puberty that is more separable for boys and girls. In addition, we compared the performance of our methods with other clustering functional data techniques. In general, the accuracy rates of the three proposed methods are higher than the accuracy rates of the competing methods. Indeed, FSC-S(D_1) was favoured among the CFD methods.

Further, we looked at the clustering results with two smoothing techniques (not differing greatly) based on two smoothing parameters (λ). Both models represent adequately smoothed curves of the growth data. The overall clustering performance was not significantly different between the two smoothing models. We found that a good smoothing model would lead to plausible clustering results, on the other hand a poor smoothing model would lead to inadequate clustering results. In addition, a particular smoothing model that fits the original trajectories properly might show more noise with the first and second derivatives. Thus, if the user is interested in the derivatives more than the original curves, then it is important to choose a smoothing model that will display reasonably smoothed curves as well as smoothed derivatives.

We have additionally attempted to cluster the first derivatives directly using the competing methods. We assumed that will be a fair comparison in this particular data as it was the routine in most clustering functional data research. However, we cannot continue with this implementation throughout the study, due to the huge computational cost. In the end, it should be mentioned that choosing the derivatives for clustering the data instead of the original curves will often result in different outcomes.

Chapter 5

A New Framework for Model Selection in Clustering Functional Data

In this chapter we present a new paradigm for model selection, by introducing the technique of downsampling, which allows us to create lower resolution replicates of the observed curves. This procedure aims to provide inference into the number of clusters for any functional clustering method, and it is based on the concept of stability of clusters. The first section presents an overview of model selection criteria in general and discusses clustering stability in more detail. Section 5.2 introduces our downsampling criterion and the sampling scheme used for creating replicates of the original functional data. The use of this new criterion is illustrated in Section 5.3 through application to the functional clustering of the Berkeley growth data.

5.1 Clustering Stability

In this section, we review the model selection criteria that are widely used in the clustering functional data literature. Additionally, we discuss the clustering stability approach for defining the appropriate number of clusters in a data set.

Clustering methods always raises the crucial question about how many potential homogeneous groups are there in a given data set. In most cases, the number of clusters k is unknown

and must be estimated from the data. In the literature a wide variety of approaches have been proposed to determine the number of clusters for different clustering methods. Commonly, for model based clustering techniques, the parameter k is inherent in the model and can be estimated. The best k can then be chosen by one of the popular model selection criteria such as the AIC (Akaike, 1974) and the BIC (Schwarz et al., 1978).

In particular, under the functional data framework, similar approaches have been proposed for model selection. For instance, for the frequentist functional model-based clustering methods Bouveyron and Jacques (2011); Giacofci et al. (2013); Same et al. (2011) used the regular AIC and BIC to perform model selection. On the other hand, most Bayesian model-based functional data clustering approaches are based on the Dirichlet Process Prior (DPP) model. Thus k is a model parameter and can be directly estimated from the data, for instance, see Ray and Mallick (2006); Scarpa and Dunson (2009); Suarez et al. (2016); Zhang et al. (2015). Further, Sugar and James (2003) suggested to use the distortion function to select the number of clusters. They defined the distortion function as the Mahalonobis distance between each coefficient vector and its closest cluster's centre. On the other hand, there are other model free approaches that determine the number of clusters before the clustering process starts such as Ieva et al. (2013), using the silhouette plot.

However, in non-parametric clustering, model selection is a more difficult problem. Therefore such methods tend to use other techniques to choose the optimal k. A technique that is widely used for non-parametric methods is testing clustering stability. The basic principle is that a true cluster is a stable structure in the data set. That is, if there are several data sets created from the same distribution, a good clustering technique must detect the same structuring (grouping) in all these sets. Von Luxburg et al. (2010) discussed the different ways in which clustering stability can be computed and used for model selection. They also reviewed some theoretical results for clustering stability that were based on k-means algorithms but can be extended to other clustering methods like spectral clustering. Several studies have suggested using the stability approach and proved consistency results and compared the stability approaches to

well known methods through simulations and real world data, for instance see Ben-David et al. (2006); Ben-Hur et al. (2001); Hennig (2007); Lange et al. (2004).

In practice, various methods have been proposed to validate the stability scores and use them for model selection. One of these approaches is based on the perturbation scheme. To be able to evaluate the stability of a fixed clustering algorithm, the clustering algorithm must run a number of times on slightly different data sets. Thus, we need to generate perturbed versions of the original data set. The most common perturbation schemes that have been used in literature are: (1) drawing random subsamples from the original data without replacement, or (2) adding different levels of random noise to the original data points. However, the researcher must be cautious when creating the perturbed versions, because high perturbation will destroy the structure of the original data, while low perturbation will create datasets too similar to the original data, which might be useless in examining clustering stability. It is difficult to achieve this balance in practice, so studying the data set carefully, and trying different perturbation schemes is crucial to selecting the right k.

The literature on the use of cluster stability to determine the number of clusters in the context of functional data is very sparse. To date we have only found two references. Kayano et al. (2010) suggested performing repeated clustering for different number of clusters, then choosing the cluster count that shows more stability by comparing the sum of squared error criterion for each clustering. Chiou and Li (2007) followed a more heuristic approach by setting different k values and retaining the clustering that produces the best physical interpretation.

In the following section we will introduce a new framework of defining cluster stability especially designed to work in the context of clustering functional data.

5.2 General Downsampling Criteria (DSC)

This section introduces the downsampling criterion (DSC) in its general form, while in Chapter 6 we will show another downsampling approach specific to the functional spectral clustering framework.

The framework of downsampling is exclusive to the functional data as it starts with splitting the curve into two non-overlapping low resolution copies, each copy containing 50% of the original functional data. Downsampling can create replicates of the original data set without losing the important features about the curve, which in turn helps in validating techniques for functional data analysis.

As mentioned in equation (2.1), the general model of functional data can be written as:

$$y_i = x(t_i) + \varepsilon_i, i = 1,2,...,T.$$

In its simplest implementation splitting the data into two non-overlapping replicates with odd and even time points, gives:

- Replicate one:

$$y_{i_1} = x(t_{i_1}) + \varepsilon_{i_1}, i_1 = 1,3,5,...,T-1. \tag{5.1}$$

- Replicate two:

$$y_{i_2} = x(t_{i_2}) + \varepsilon_{i_2}, i_2 = 2,4,6,...,T. \tag{5.2}$$

Note that we assume the error ε's are $i.i.d \sim N(0,\xi^2)$, and $corr(\varepsilon_{i_1},\varepsilon_{i_2}) = 0$ for all i_1, i_2 which implies $corr(y_{i_1}, y_{i_2}) = 0$. Thus, the two replicates are uncorrelated and they represent two different realizations of the original data over the same timeline.

The procedure depends on the type of functional data. Making replicates of curves that are dense and have a regular timeline is easier and more straightforward than creating replicates of

sparse and dynamic with irregular timeline functions. For instance, in the first case, take every odd time point and its corresponding value to make the first replicate, and take every even time point and its corresponding value to make the second replicate, as shown in equation (5.1) and equation (5.2) respectively. While for the second case, dividing the curves into odd and even might give very different replicates and thus we should be more cautious when applying the procedure. For instance, if the data are generated by the user, then it might be a good idea to move the data from sparse to dense by increasing the time points and their corresponding arguments. Even if the data include an irregular timeline or dynamic features, downsampling still can create two similar replicates. However, downsampling may not be able to create similar replicates when the given functional data is sparse, as in the case illustrated in Figure 5.2.

After preparing the two replicates, they go through the same smoothing and basis expansion. The bases choice is similar to the one that would be applied for the original data, but we need to consider the reduced dimensions of the two replicates. This will preserve the functional structure of the high dimensional data set. Figure 5.1 illustrates the method using the sparse case of the Canadian weather data (where there are 12 temperature measures) as an example. Clearly, we can visually see that the odd and even replicates preserve most of the important features of the 35 curves. Also note the dense cases are less affected by the split and the loss will be minimal. On the other hand, Figure 5.2 shows that splitting the very sparse curves will result in different replicates. In this example, we assumed the available temperature records are only for January, March, May, July, September, and November, resulting in only three time points for each of the two downsampled replicates

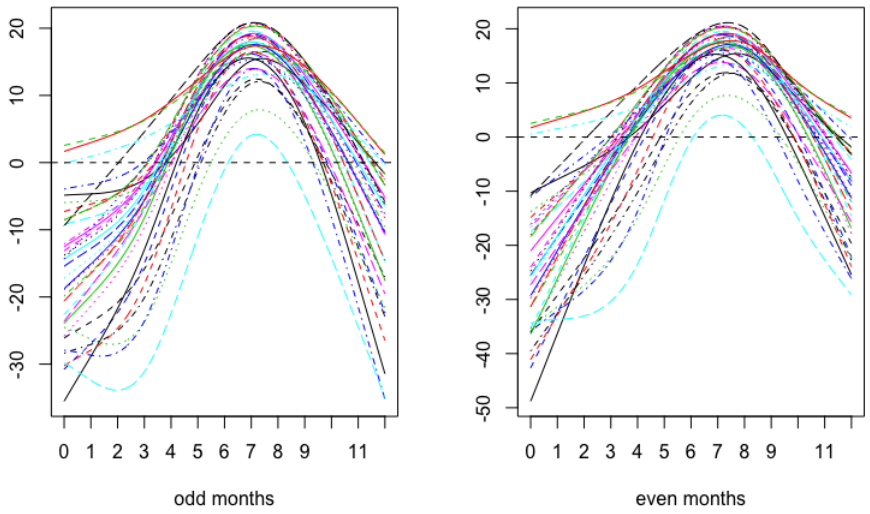


Figure 5.1: Smoothed curves of the Canadian weather data are split into two low resolutions replicates by the downsampling method, where (left) shows temperature of the odd months and (right) shows temperature of the even months.

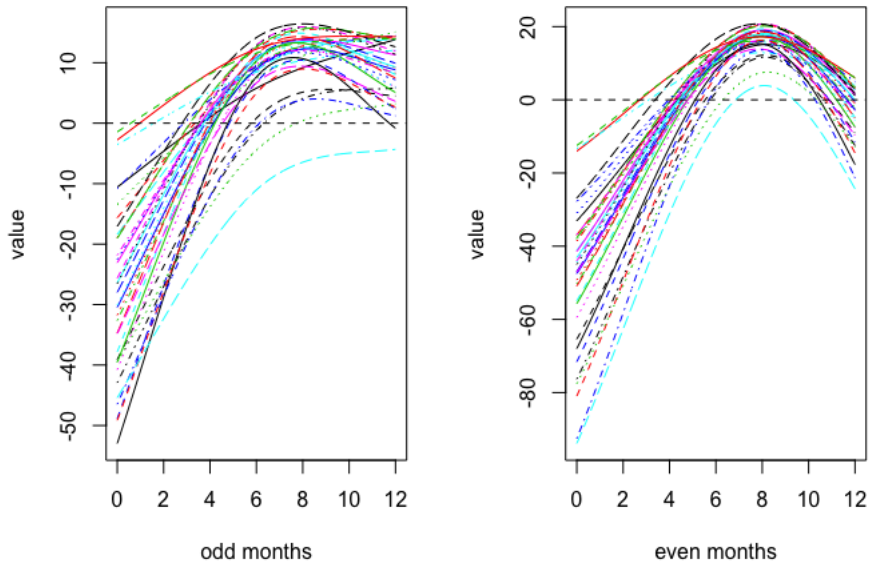


Figure 5.2: Smoothed curves of the Canadian weather data are split into two low resolutions replicates by the downsampling method, where (left) shows temperature of only January, May, and September, while (right) shows temperature of only March, July, and November.

The downsampling criterion is considered as a stability-based model selection approach for

selecting the number of clusters. Its fundamental idea is explained as follows. Given two low-resolution replicates, the functional clustering method will be performed individually on the two replicates. Then, the clustering results of replicate 1 and replicate 2 will be compared by the adjusted Rand index (ARI) (Hubert and Arabie, 1985). Recall, ARI measures the agreement between two partitions without a need for a standard true cluster. Thus, it is convenient to use ARI values and they are considered as the stability scores in our approach. This step will be repeated for a range of number of clusters, $K = \{k_{min}, ..., k_{max}\}$. By default we will use $k_{min} = 2$ and $k_{max} = 15$. However, to speed up the algorithm we can shrink the range by studying the data first. Finally, there will be one stability score, which is the ARI index for each value of K from k_{min} to k_{max}. The appropriate number of clusters will achieve the highest stability score. However, this is not sufficient to choose the best number of clusters. Thus, to control the uncertainty of the downsampling results, we create more replicates than just one pair of odd and even copies. The general procedure we have proposed for generating more low-resolution copies of the original functional data is through borrowing ideas from systematic sampling, which is described in the next section.

5.2.1 The Sampling Scheme

One of the ways to evaluate the stability of a fixed clustering algorithm is by performing the clustering algorithm several times on slightly different data sets. As mentioned above the new sets can be created through sampling the original data. Our sampling scheme is based on projecting the original high-dimensional data to low-dimensional spaces. To keep the sampling scheme consistent over the new low-dimension (low-resolution) copies we proposed a semi-systematic sampling scheme.

Based on the well-known systematic sampling, we have developed a sampling scheme to create different samples. Before describing our sampling scheme, we will briefly review the systematic sampling scheme. In the standard systematic sampling the first element is selected randomly, then the remaining are selected automatically according to a predetermined pattern. Suppose there are N elements in the population and the required sample size is n. Then we define some p integer such that $N = np$. Assume all the N elements of the population can be

arranged in a list. The first element i is selected randomly, where $1 \leq i \leq p$, then include every p^{th} element of the population in the sample. For instance, see Figure 5.3 for easy illustration of this sampling procedure.

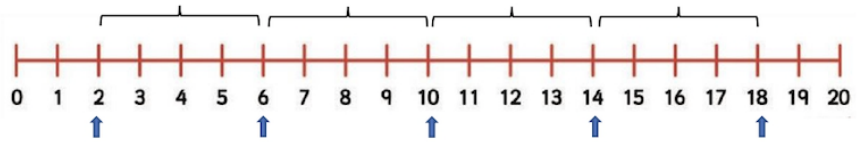


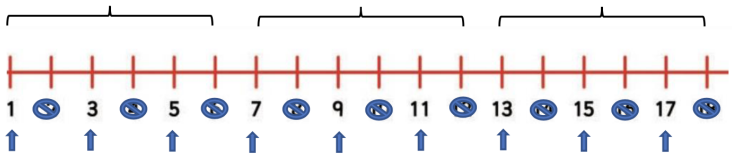
Figure 5.3: This chart illustrates a toy example of the systematic sampling procedure. When $N = 20$ and $n = 5$, then $p = 4$. Starting randomly with 2 will include $\{2, 6, 10, 14, 18\}$ in the sample.

However, in downsampling we aim to include 50% (for a discussion, see below) of the full functional data. Thus we will only borrow the idea of having a predetermined pattern for the sample selection by defining integer p. The procedure will divide the original timeline into subintervals each of size p. Then from each subinterval, we will choose 50% of the time points and their corresponding values. Selecting the time points in the first subinterval will follow a predetermined pattern and will be repeated over the rest of the subintervals. To simplify the procedure and generalize it so that it can be applied in different functional data scenarios, we fixed $p = 6$ and defined the selection scheme in terms of logical sets. All the possible sets are listed in Table 5.1. The number of possible logical sets can be calculated using the combination of p points taken $\frac{p}{2}$ at a time. Thus, $\binom{p}{p/2} = \binom{6}{3} = 20$, and these 20 logical sets will be listed as pairs of opposite sets. The key idea is to create two non-overlapping sets at every pair. For instance, the first line shows the sampling of odd values versus the sampling of even values to create the first downsampled functional data as mentioned above. Since we need more copies in a low-dimensional space of the original functional data, we generated different sets of odd and even pairs. Note that set 1 always starts with T thus we call it the odd set, while set 2 always starts with F thus we call it the even set, for simplicity. Based on this selection scheme each generated downsampled functional data set will be unique. Also see Figure 5.4 for an illustration of the our sampling scheme when applied to a line of data points using some of the

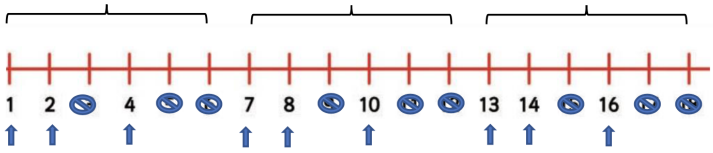
patterns.

pair	set 1 (odd)	set 2 (even)
1	T,F,T,F,T,F	F,T,F,T,F,T
2	T,T,T,F,F,F	F,F,F,T,T,T
3	T,T,F,T,F,F	F,F,T,F,T,T
4	T,T,F,F,T,F	F,F,T,T,F,T
5	T,T,F,F,F,T	F,F,T,T,T,F
6	T,F,T,T,F,F	F,T,F,F,T,T
7	T,F,F,T,F,T	F,T,T,F,T,F
8	T,F,F,F,T,T	F,T,T,T,F,F
9	T,F,T,F,F,T	F,T,F,T,T,F
10	T,F,F,T,T,F	F,T,T,F,F,T

Table 5.1: Sampling scheme for generating pairs of downsampled functional data based on logical sets of True and False. T stands for selected values from the original time points, while F are the omitted values. Set 1 and set 2 are opposite to each other and non-overlapping.



(a) pattern {T,F,T,F,T,F}



(b) pattern {T,T,F,T,F,F}

Figure 5.4: These charts illustrate a toy example of our sampling scheme in different patterns. When $N = 18$, $p = 6$, and $n = 9$. Then (a) subset will include $\{1,3,5,7,9,11,13,15,17\}$, and (b) subset will include $\{1,2,4,7,8,10,13,14,16\}$.

After explaining the sampling procedure, a few points must be clarified. First we discuss the

reason behind specifically choosing 50% when sampling the original curves. The simplest way to obtain similar but non-overlapping sets of curves is by dividing them into odd and even, thus every set contains 50% of the time points and their corresponding values from the full curves. Hence this percentage preserves the equality between the downsampled curves in holding the important features of the functional data set. A higher percentage will create overlapping pairs of odd and even sets, which in turn will result in dependency between the sets, while a lower percentage might fail to catch some information related to clustering. In fact, Lange et al. (2004) have proposed a stability approach that is based on splitting the data into two disjoint and equal size subsets because the overlap could already determine the clustering structure. This in turn will create dependence and would lead to artificial stability. Further, Hennig (2007) stated that choosing a large subset will not generate enough variation to be informative, while choosing a small subset might obtain poor clustering results. Thus, the author proposed a subsetting scheme that uses half of the original dataset for evaluating the clustering stability.

As mentioned above, one pair of odd and even sets is not sufficient for choosing the number of clusters, more pairs must be sampled. Dividing the full timeline into subintervals and restricting the procedure to sample 50% of the points within each subintervals avoids missing the main structure and does not lead to a confused copy of the original functional data.

One might question why $p = 6$ and why the sampling is systematic and not random. Since we want to include 50% of the data at every sample, we are restricted to an even p integer, to have equal values of True and False. Thus the potential values are $p = 4$, $p = 6$, and $p = 8$. Looking at $p = 4$, the possible odd combinations are only {T,F,T,F}, {T,T,F,F}, and {T,F,F,T}, and their even pairs. This limited number of samples is not enough to compute the overall stability scores. On the other hand, setting $p = 8$ can give 35 pairs of odd and even sets. However, most timelines of the functional data are not divisible by 8, which might lead to omitting more time points to adjust the timeline. Therefore, we assumed obtaining 10 pairs of odd and even when $p = 6$, is adequate to carry out comparisons and draw conclusions. Besides, it is more likely to get timelines divisible by 6 than 8. Nevertheless, in some examples, we will have to

omit a few values to make the timeline divisible by 6. The common procedure is to omit the very last time points. Despite that, it is recommended to study the data in advance to insure the omitted values are not playing a major role in defining the structure of the curves.

We now briefly discuss why we choose systematic sampling over random sampling. Random sampling of 50% of the data points in a curve usually disorders the main structure of the functional data and leads to correlated sub-sampled sets. For instance, sampling randomly from the original time points and their corresponding values might omit the first 10 points, or it can omit all the points that reflect a major peak or a sudden shift. Further, the downsampling approach attempts to create non-overlapping copies of the original functional data to keep the two replicates uncorrelated, which is unattainable with random sampling with replacement. Finally, random sampling will create different sampled curves within the sampled set, and it will be challenging and computationally expensive to smooth every curve individually. In fact, we have attempted to apply downsampling based on the random sampling. The resulting curves lose the original structure of the data. Further, every curve needs a different smoothing model, which is time consuming.

5.2.2 The Criteria

Now we describe the procedure of applying the downsampling criteria and the sampling scheme, as explained above. First the functional data will go through the sampling scheme to create 10 pairs of odd and even replicates. Each of these replicates will be smoothed and clustered by applying a functional clustering method. Then, for each pair of opposite sets, the clustering results of the odd and the even sets will be compared by ARI to compute the stability scores. This step will be repeated over a range of K values, where $k_{min} \leq K \leq k_{max}$. Thus, there will be 10 stability scores for each k value, which can be represented graphically by boxplots. Then, the number of clusters k with the highest stability is chosen. Algorithm 1 (Figure 5.5) explains our proposed paradigm for choosing the number of clusters based on the stability arguments for functional data. The algorithm is written in the general format, where the user can specify p and accordingly m.

In our proposal we set $p = 6$ which gives $m = 10$ pairs. Also, we find letting $2 \leq K \leq 15$ is appropriate to examine the optimal k value among the given range.

Algorithm 1 General Downsampling Criterion

procedure CLUSTER & ESTIMATE(k)
  for $i \leftarrow 1, m$ **do**
    Split the data into odd and even ▷ based on p
    for each replicate **do**
      Smooth the data
      for $K \leftarrow k_{min}, k_{max}$ **do**
        Cluster the data ▷ with any CFD method
      end for
      return clustering results $\forall K$
    end for
    Compute ARI: $ari(odd, even)$ ▷ one stability score for each k
  end for
  Record final results ▷ m stability scores for each k
  Create boxplots of ARI vs K
  Boxplot with largest medium will lead to optimal k
end procedure

Figure 5.5: The general downsampling criterion algorithm.

5.3 Application of DSC on the Berkeley Growth Data

Considering the Berkeley growth data again, we can apply the general downsampling criterion to estimate the number of clusters. Recall in Section 4.3 we set $k = 2$ that refers to the male and female clusters in the data. However, we are also interested in looking at the performance of DSC on this data and what will be the resulting k. This functional data set is a hard example

as it consists of an unequally-spaced timeline. Implementing our sampling scheme gives 10 pairs of odd and even sets of the growth data. Figure 5.6 displays the odd subsample sets from the original curves. The overall structure of the curves is maintained, despite the fact that each individual curve has changed.

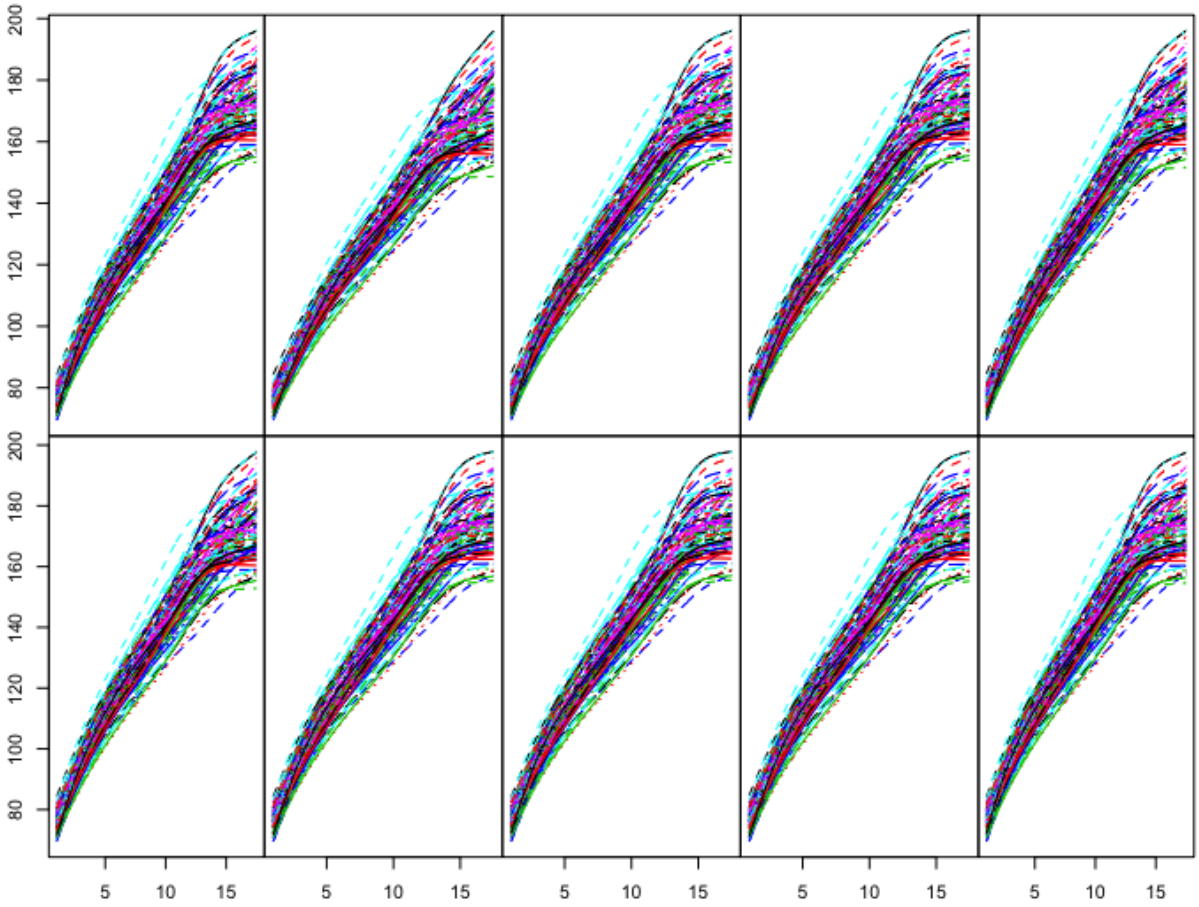


Figure 5.6: The odd subsamples of the growth data

Considering the first pair of odd and even copies, we can apply the permutation T-test using `tperm.fd` in the `fda` package to test for a difference between the two sampled copies. The null hypothesis assumes that there is no difference between the odd set and the even set in terms of children heights over the timeline. The resulting p-value= 0.92, and the graph with the T-test pointwise critical values is shown in Figure 5.7. The results suggest that there is no significant evidence to reject the null hypothesis, thus there is no significant difference between the two sets of curves.

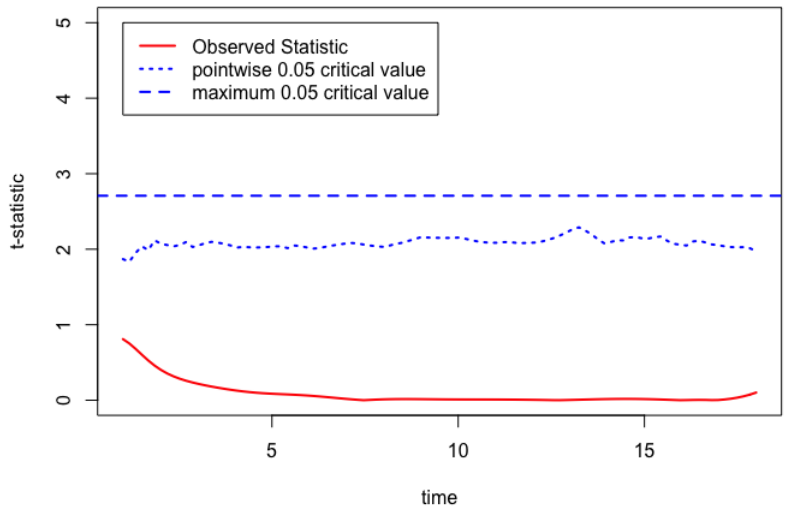


Figure 5.7: Permutation t-test for odd copy and even copy of the original functional data.

In our first trial, we applied the downsampling approach on only one pair of the regular odd and even sets. Including all the functional data clustering methods that have been explained in this thesis, we examined its performance for $2 \leq K \leq 6$. The results are shown in Figure 5.8; in general, there is no clear trend of the ARI lines to conclude the optimal k value. Thus, it is not enough to judge on the optimal number of clusters from only one pair of replicates. However, looking at each clustering method separately, there are some peaks at $k = 2$. Therefore, it would be of interest to carry out the general DSC on each of the clustering methods.

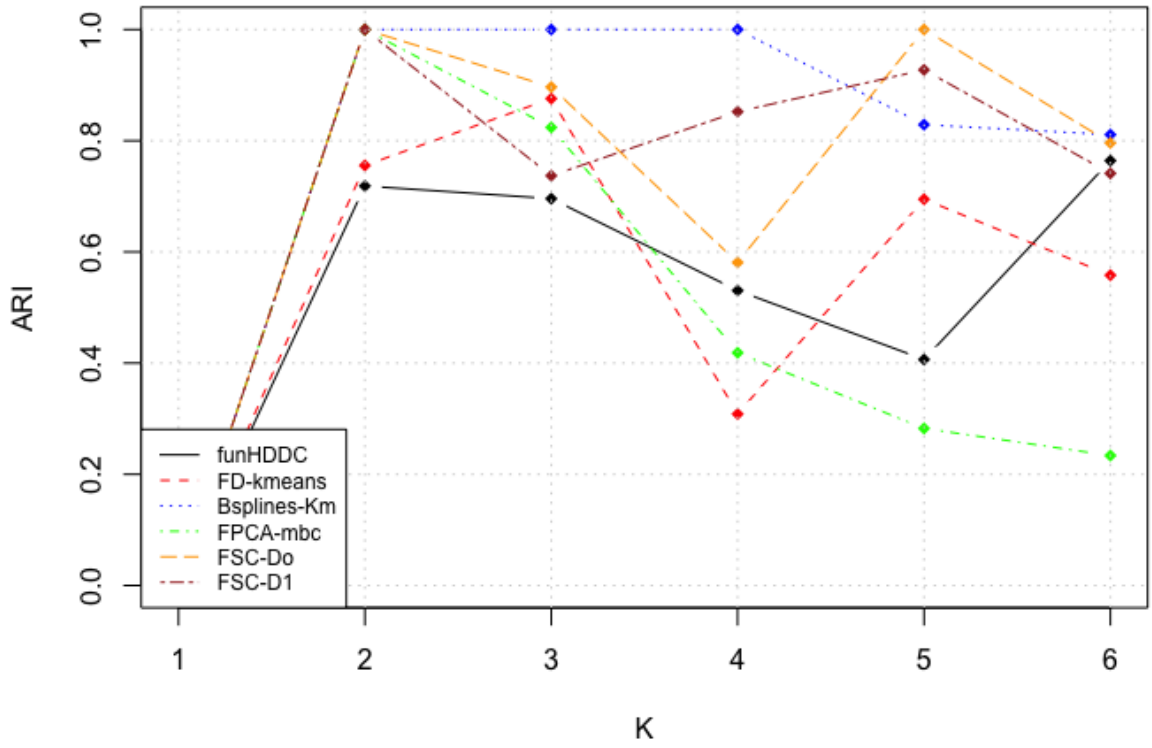


Figure 5.8: Results of ARI for each K when using the downsampling criteria with different clustering techniques

The general DSC was applied on the following clustering methods: FunHDDC, FD-Kmeans, B-splines-km, FPCA-mbc, FSC-S(D_o), and FSC-S(D_1), for $2 \leq K \leq 9$. As often the basis expansion is done before clustering, to examine the performance of the criteria we have fixed the smoothing model for all the methods. The appropriate smoothing technique is B-splines of order 6 in a saturated model (depending on the selected time points) with $\lambda = 10^1$. Comparing this smoothing choice to the one used for the original Berkeley growth data in Section 4.3, we can notice that the smoothing parameter λ is slightly bigger which will impose more smoothing constraints to the downsampled curves. This is because applying smaller λ was noticed to create some dips between a data point and another which is unrealistic with children's heights.

The results of the general downsampling criterion are shown in form of boxplots. The boxplots represent the clusters stability based on the adjusted Rand index for each k. The following

results relate to FunHDDC (Figure 5.9), FD-Kmeans (Figure 5.10), B-splines-km (Figure 5.11), FPCA-mbc (Figure 5.12), FSC-S(D_o) (Figure 5.13), and FSC-S(D_1) (Figure 5.14).

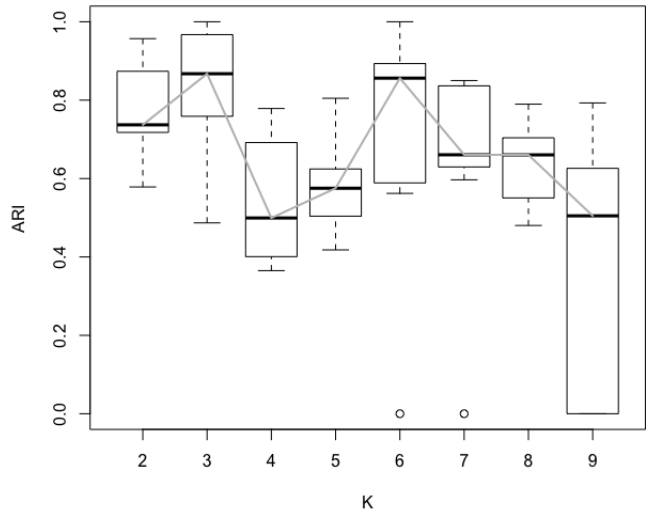


Figure 5.9: Boxplots of the ARI over k values when applying the general DSC with FunHDDC on the growth data. The approach suggests there are 3 clusters in the data.

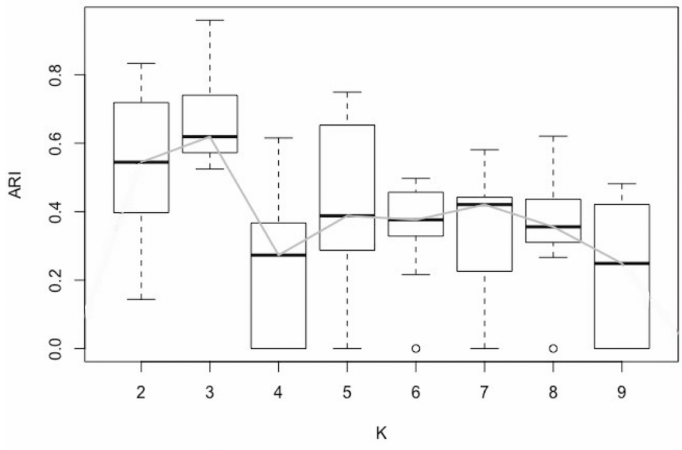


Figure 5.10: Boxplots of the ARI over k values when applying the general DSC with FD-Kmeans on the growth data. The approach suggests there are 3 clusters in the data.

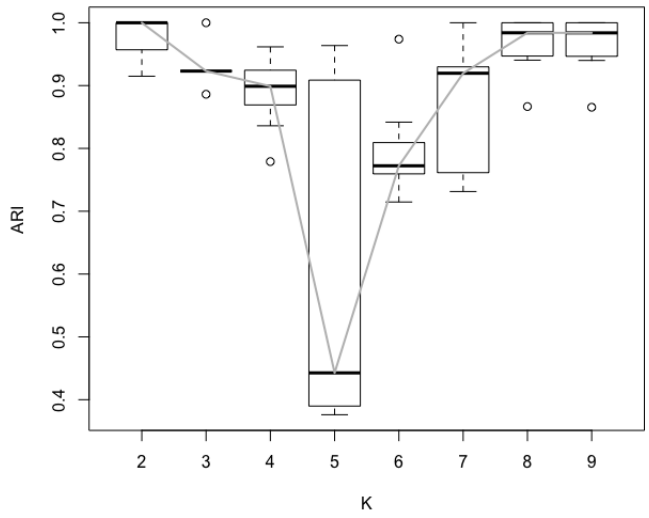


Figure 5.11: Boxplots of the ARI over k values when applying the general DSC with B-splines-km on the growth data. The approach suggests there are 2 clusters in the data.

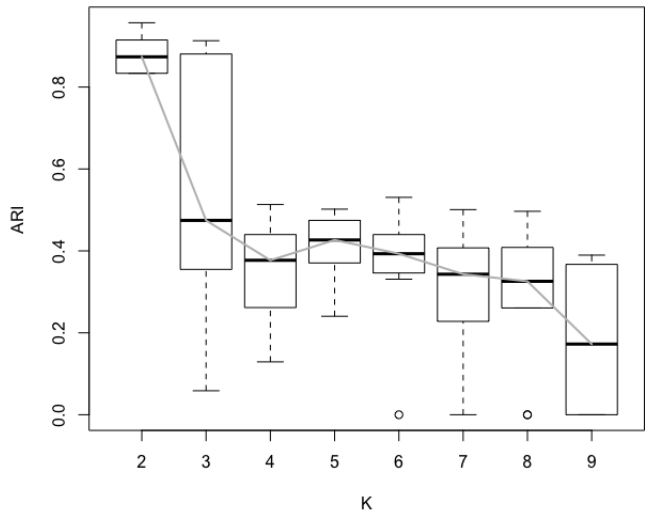


Figure 5.12: Boxplots of the ARI over k values when applying the general DSC with FPCA-mbc on the growth data. The approach suggests there are 2 clusters in the data.

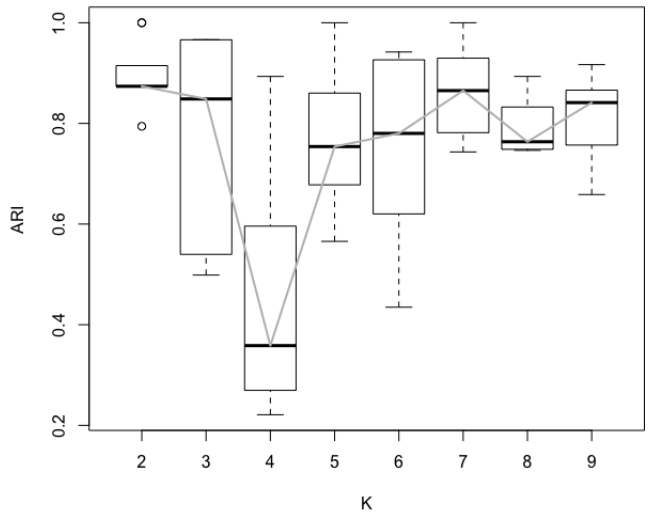


Figure 5.13: Boxplots of the ARI over k values when applying the general DSC with FSC-S(D_o) on the growth data. The approach suggests there are 2 clusters in the data.

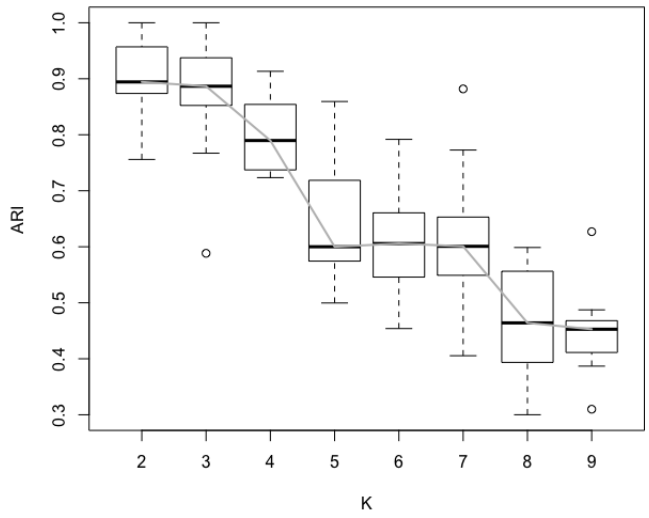


Figure 5.14: Boxplots of the ARI over k values when applying the general DSC with FSC-S(D_1) on the growth data. The approach suggests there are 2 clusters in the data.

In general, there is a clear preference to choose $k = 2$ as the optimal number of clusters in the growth data. However, $k = 3$ looks to be a reasonable choice too. Apart from FunHDDC and FD-Kmeans, all the other methods give the highest boxplot at $k = 2$. Also, in FPCA-mbc, FSC-S(D_o), and FSC-S(D_1) the second highest boxplot is at $k = 3$. While, in FunHDDC and

FD-Kmeans $k = 3$ comes before $k = 2$. Figure 5.15 summarizes the resulting means of the ARI for each clustering techniques which confirms the results of the individual boxplots.

From experience we noticed changing the smoothing model through varying λ can change the outcome of the downsampling criterion. For instance, if we set $\lambda = 10^{-0.5}$ we get the highest stability score at $k = 2$ in both FunHDDC and FD-Kmeans. On the other hand, the same λ will lead to different results in our proposed methods FSC-S(D_o), and FSC-S(D_1), where the highest boxplot will point to $k = 7$ in FSC-S(D_o) and in FSC-S(D_1) the highest stability is at $k = 2$ and $k = 3$ equally. Thus, it is crucial to choose the appropriate smoothing model for the sampled data with care. As the number of clusters is usually unknown in most of the clustering problems, an inappropriate smoothing model will give misleading results. It should be also mentioned that the clustering technique did not converge in all the iterations in FunHDDC, FD-Kmeans, and FPCA-mbc, which resulted in a few missing ARI values.

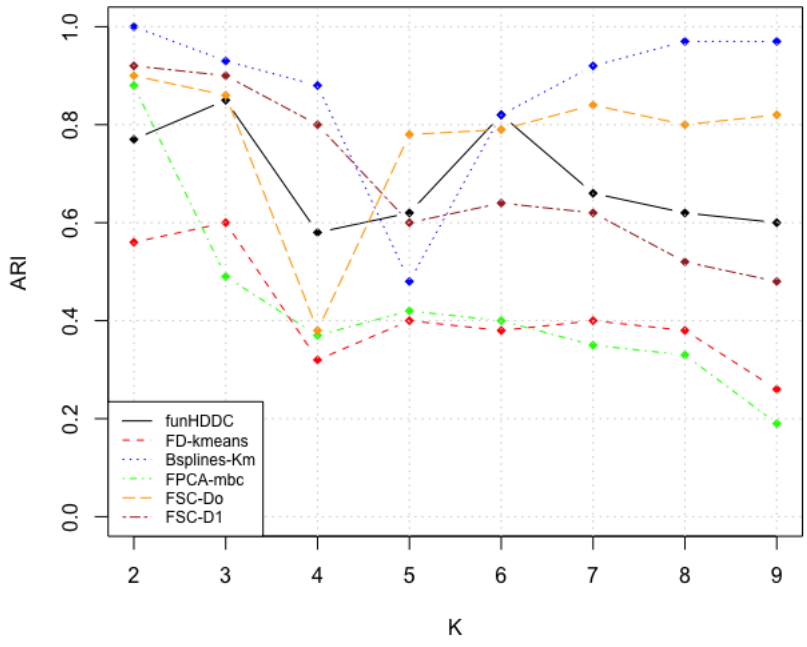


Figure 5.15: Results of the ARI **mean** values for each K when using the downsampling criteria with different clustering techniques.

5.4 Chapter Summary

In this chapter we have introduced the general downsampling criterion (DSC), a model selection technique based on clustering stability. It is a new paradigm designed for functional data that attempts to identify the number of clusters in the data. The paradigm is based on first creating low-resolution replicates of the original data by using a designed sampling scheme. This sampling scheme creates 10 pairs of two non-overlapping sampled sets (odd replicates and even replicates). The downsampled sets go through smoothing and clustering over a range of K values using any of the functional clustering methods. At each k, the clustering results of the two replicates are compared by the ARI which refers to the stability score in our paradigm. This process is repeated for the 10 pairs of odd and even replicates, and the stability score at every iteration is calculated. Finally, the stability scores are represented as boxplots for the specified range of K, and the highest boxplot will indicate the optimal k value.

Performing the downsampling criterion in 10 pairs of odd and even replicates instead of just one pair is more informative and reliable. As every sampled set consists of different data points, some of the sampled sets might miss important information relating to the clustering structure of the data. Through applications on the Berkeley growth data, the downsampling criterion showed acceptable results in most functional data clustering methods.

It should be mentioned that due to the possibility of some lost in the quality of the replicates and accordingly the clustering results, the DSC is only used as a model selection criterion and is not preferred for finalizing the clustering results. The basic concept is to identify the optimal number of clusters k, and not clustering the data into groups. The resulting k will be used to cluster the original functional data. This is because the fact that if a clustering structure is stable in low-resolution space, it will be stable in high-resolution space (original data), but it does not necessarily reflect the optimal assignment of curves into clusters. It is also worth mentioning that in case the data consists of only one cluster, then the approach should give very low ARI for all $k > 1$ without a peak at a specific k. However, in real datasets it might be hard to achieve this

and the criterion might provide some misleading k rather than suggesting that the data cannot be clustered, which is in fact a common issue with distance-based clustering approaches.

Downsampling is best used for dense functional data, and not recommended for sparse functional data, since in the latter case the sampling scheme will create replicates that will fail to retain the original structure of the functional data, which in turn will give misleading results. Also it is important that the smoothing technique is selected thoughtfully. As is always the case in any functional data analysis, smoothing and basis expansion play a critical role in determining the final outcomes. Finally, we should consider the fact that the success of the downsampling criterion depends on the performance of the chosen clustering method.

Chapter 6

Downsampling Criterion with Functional Spectral Clustering Approach

In this chapter we introduce the integrated functional spectral clustering-downsampling approach. As discussed in the previous chapter, downsampling allows us to create lower resolution replicates of the observed curves. These replicates will now be used to provide insight into the parameters k and σ for the FSC-S approach. The first section defines the eigengap heuristic and explores the potentials of using it for examining the clustering stability, and briefly reviews some corresponding studies. Section 6.2 details the extended approach of functional spectral clustering based on downsampling criteria. The use of the integrated approach is illustrated in Section 6.3 through application of our approach on the Berkeley growth data.

Recall, in Chapter 4, we have explained our proposed framework FSC-S. The approach showed favourable results compared to other methods when applied to the Berkeley growth data. Later, in Chapter 5, we have introduced the general DSC as a model selection criterion for identifying the number of clusters in a functional data set. Again we evaluated the criterion on the Berkeley growth data and it showed encouraging results. One limitation of the FSC-S approach is the need to specify the number of clusters a-priori. In this chapter we will address this limitation by employing the downsampling criterion in selecting the number of clusters. Based on the concept of stability clustering we aim to estimate the parameter k, the number of

clusters, within the functional spectral clustering technique.

6.1 Conceptual Understanding and Motivation

Spectral clustering is one of the most popular clustering approaches that falls under the broad category of non-parametric clustering. In this category, the clustering techniques do not rely on an underlying model. Therefore, the well established techniques for choosing the number of clusters that work well for model-based clustering approaches, would not be valid in the non-parametric approaches. Instead, a wide range of criteria and techniques have been developed in the literature to choose the number of clusters in non-parametric settings. Since our plan is to employ the cluster stability concept for spectral clustering, we briefly list some of the techniques that have been used particularly in this context in the literature. For instance, Wang (2010) developed a scheme for choosing the number of cluster that minimizes the algorithm's instability based on cross validation. Hess and Duivesteijn (2019) suggested a method that is based on determining whether two clusters are likely to come from a single distribution, and accordingly proposed a probability bound specified only by the sample mean and variance. More recently, Andreotti et al. (2020) introduced a stability measure for spectral clustering by computing the structured distance to ambiguity, which refers to the minimal distance of the Laplacian to Laplacians of graphs with the same vertices and edges but with weights that are perturbed such that there is no clear k^{th} spectral gap. In addition to the above, a traditional tool that is linked to clustering stability and has been frequently used in spectral clustering is the eigengap heuristic (Chung and Graham, 1997). This approach has been particularly designed for spectral clustering and its concept is to compute the difference between each eigenvalue λ_j and eigenvalue λ_{j+1}, and the eigengap will be the maximum value which can be written as: eigengap $= max(\lambda_{j+1} - \lambda_j)$. Then, all eigenvalues that come before the maximum eigengap will indicate how many clusters k_i would be in the data. This procedure can be justified by the perturbation theory (Section 4.4). According to the perturbation theory, in the ideal case there will be k very small eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_k\}$ compared to the rest of the eigenvalues; the gap between λ_k and λ_{k+1} represents the eigengap. Besides, the Davis-Kahan theorem (Section 4.4) illustrates

that the distance between the ideal eigenspace and the perturbed eigenspace is bounded by some value which is proportional to the perturbation size and inversely proportional to the eigengap. Thus, it has been suggested in the literature that the eigengap can be used as a stability indicator of the clustering (Von Luxburg, 2007). In a similar direction, our proposed method is motivated by the perturbation theory and Davis-Kahan theorem, as it will be explained in more details throughout this chapter.

Now, we will be presenting the eigengap heuristic using a toy example to illustrate the concept. Consider some functional data over the time $0 \leq t \leq 2\pi$ that come from the 3 functions:

- $\frac{1}{2}t + u_1 + 2 - \cos(t) + \varepsilon$, with $u_1 \sim \mathcal{N}(\mu = 1, \xi^2)$,
- $\frac{1}{2}t + u_2 + 1 + \sin(t) + \varepsilon$, with $u_2 \sim \mathcal{N}(\mu = 0, \xi^2)$,
- $\frac{1}{2}t + u_3 + \cos(\frac{2t}{\pi}) + \varepsilon$, with $u_3 \sim \mathcal{N}(\mu = -1, \xi^2)$,

where $\varepsilon \sim \mathcal{N}(\mu = 0, \xi^2)$, while the standard deviation ξ takes the values $0.1, 0.25, 0.45$, and 0.8 to move from low-noise functional data to high-noise functional data in 4 different scenarios. In other words, we want to vary the difficulty of the clustering and observe the change in the eigengap for each scenario. Figure 6.1 displays the created functional data in four levels according to the noise along with the eigenvalues graphs after clustering the data with FSC-S(D_o). Before explaining the results, it should be mentioned that σ of the similarity matrix A_{ij} (defined in Chapter 4) is fixed and equal to 1. In our proposed clustering approach (Section 4.2) we set $\sigma =$ the standard deviation of the elements of the distance matrix. However, in this example we fixed σ to be 1. This is because we want to observe the change in the eigengap heuristic only by increasing the noise (perturbation) in the data, keeping all other parameters e.g. σ and the smoothing parameter λ fixed.

The first set of functional data displayed in the first row of Figure 6.1 consists of 3 well separated clusters, and we can see the first 3 eigenvalues are very small compared to the rest of the eigenvalues. The gap between the third and the fourth eigenvalues represents the eigengap. The same behaviour can be observed from the second set of functional data (second row of Figure 6.1). In the third set, although the separation between clusters is not very visible, the eigengap heuristic still indicates 3 clusters in the data (third row of Figure 6.1). But clearly the

eigengap shrinks as the noise increases. In the last functional data set, there is no clear eigengap. In fact, the noise in this data is very high and thus there is no clear pattern of clusters, which makes it a hard clustering problem. This toy example illustrates that the eigengap heuristic often works well if the data consists of well defined clusters, but would not give optimal results when the clusters in the data overlap very much, which is expected in most criteria for choosing the number of clusters.

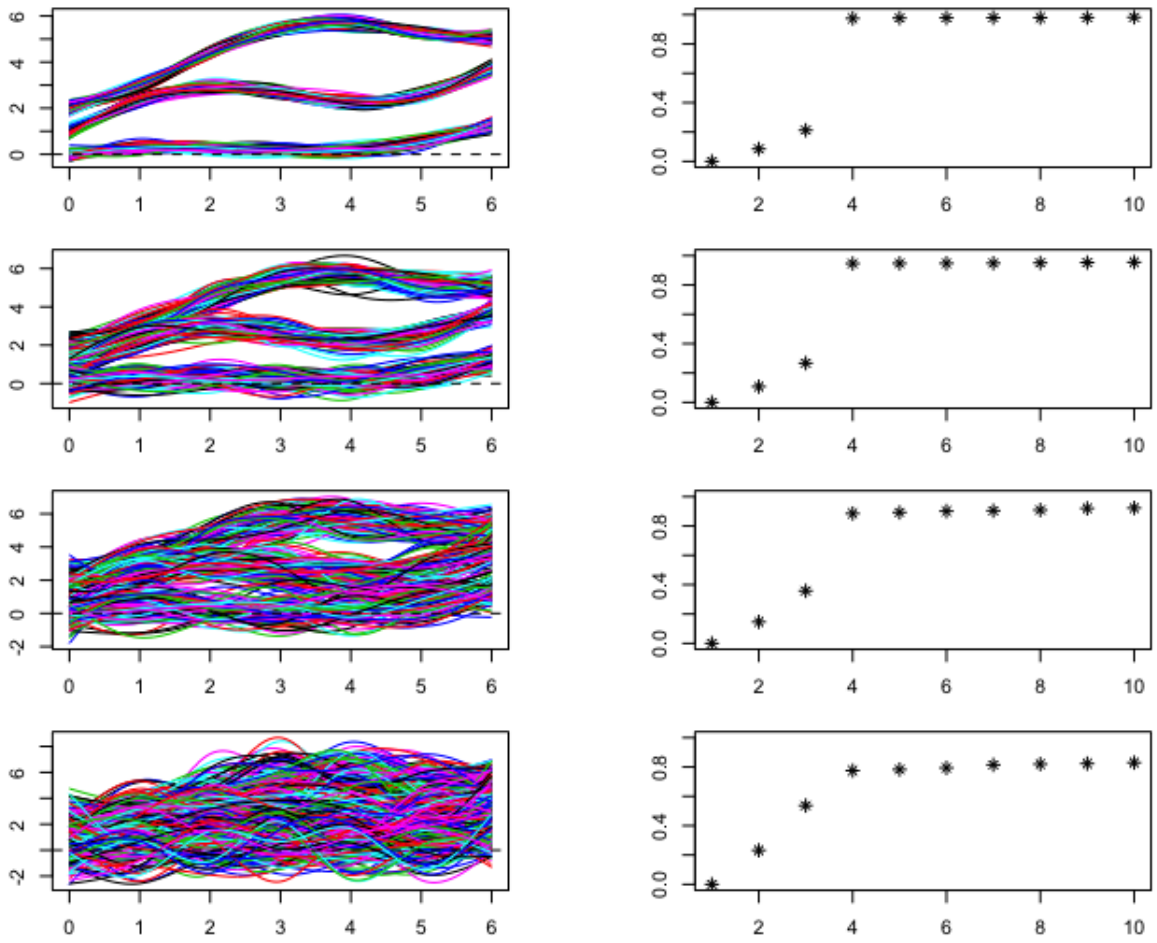


Figure 6.1: Different functional data sets with the smallest 10 eigenvalues according to FSC-S(D_o). From top to bottom: low-noise to high-noise functional data.

Note that in this example, we have fixed σ to be 1, however, there is a strong relation between the eigengap heuristic and the parameter σ. The effect of σ on the eigengap heuristic is illustrated in Figure 6.2. Consider again the toy example explained above, specifically the scenario when $\xi = 0.25$. According to the graphs of the eigenvalues, FSC-S(D_o) can indicate

that there are 3 clusters based on the eigengap for $\sigma = 0.5, 0.75, 1$ and 1.25. However, when $\sigma \geq 1.5$, the eigengap comes after the first eigenvalue which gives only one cluster. On the other hand, at $\sigma = 0.25$ all eigenvalues will be stacked in a line with no eigengap, which assumes there are as many clusters as number of curves in the data. Overall, this example illustrates that the parameter σ influences the eigenvalues and accordingly the eigengap. In more details, choosing a smaller σ tends to stack all eigenvalues together while a large σ tends to create a big gap between the first eigenvalues and the rest. In between, some σ values will support the appropriate eigengap which in turn will reveal the correct number of clusters in the data.

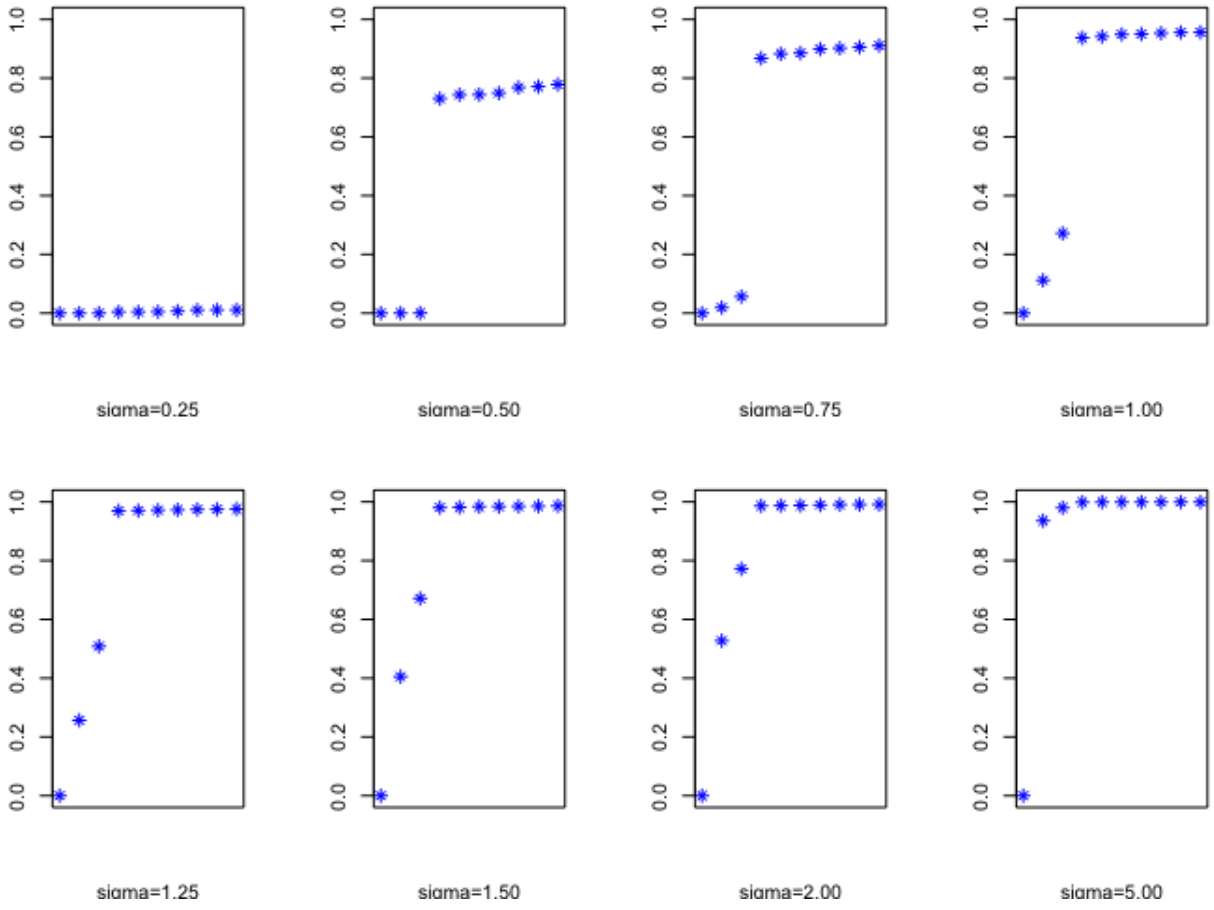


Figure 6.2: Graphs of the 10 smallest eigenvalues when applying FSC-S(D_o) with a range of σ values for the toy functional data.

Another point of discussion is associated with the choice of optimal values of σ and the proper eigengap if the functional toy data is slightly altered. Consider the above toy example obtained by multiplying all data values by 100. Then, the optimal value(s) of σ that will support

the eigengap to reveal the right number of groups will change too. Figure 6.3 shows the 10 smallest eigenvalues when applying FSC-S(D_o). In this case, the new optimal σ values are different and 10 times higher than the previous values.

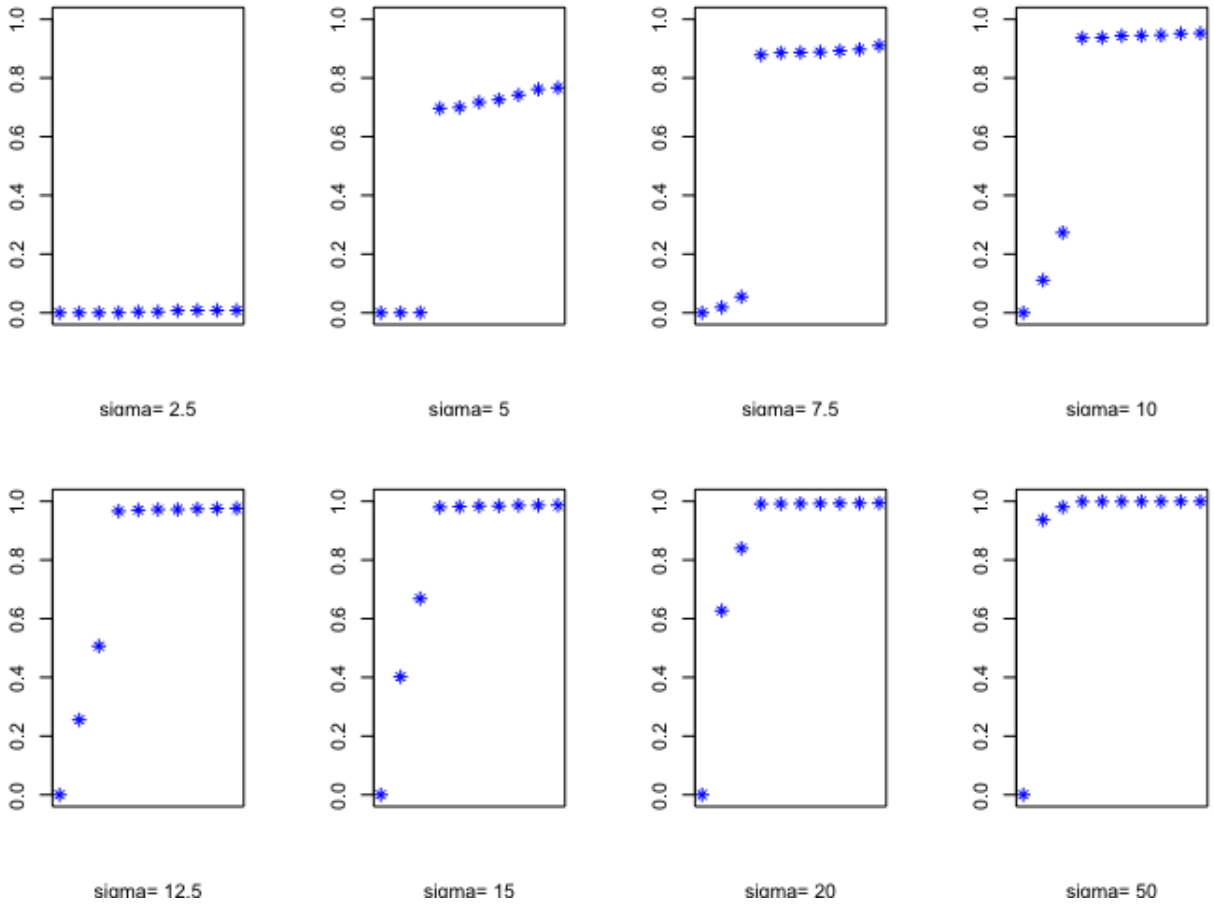


Figure 6.3: Graphs of the 10 smallest eigenvalues when applying FSC-S(D_o) with a range of σ values for the toy functional data multiplied by 10.

From above, we attempted to explore three different forms of relationship as follows:

- The relation between the eigengap and the perturbation in the data,
- The relation between the eigengap and the values of σ,
- The relation between the values of σ and the domain of the data.

The outcomes contribute to our understanding of the role of σ in finding k, and the effects of perturbation on these parameters. Although our main interest is not σ itself, the value(s)

of σ will help in revealing the optimal eigengap and accordingly the optimal k. Based on the above facts, we are interested to move forward and introduce our new proposal in the following section.

6.2 Specific Downsampling Criterion (FSC-DSC)

In this section, we design a downsampling criterion that is specific for the functional spectral clustering approach by considering σ as a flexible parameter in the process. This proposed criterion leads to extending the default FSC-S method to FSC-DSC, that additionally estimates k when clustering the data.

In Section 4.2, we assumed k is known a-priori and σ is fixed. However, from the toy example we found that the parameter σ plays an important role in spectral clustering as it controls the width of the neighbourhoods. Von Luxburg (2007) stated that the choice of σ depends on the domain of the data, and no general advice is given. The author also added that σ has direct influence on the choice of number of clusters in the data. In the context of the standard spectral clustering, there exist some studies that show the significant influence of σ in changing the clustering results and proposed self-tuning methods (Afzalan and Jazizadeh, 2019; Bruneau et al., 2014; Zelnik-Manor and Perona, 2005).

To our knowledge, there has been no prior application of spectral clustering on functional data, and there does not exist any method for finding the best σ or k when employing spectral techniques to cluster functional data. In fact, the tasks of finding the right number of clusters k and the optimal σ are still an active area of research for multivariate spectral clustering.

We propose the use of the downsampling criterion to estimate both σ and k by modifying the general approach of downsampling explained in Section 5.2. This criterion can only be applied to functional data, and thus it gives our functional spectral clustering approach an advantage over the standard multivariate spectral clustering methods.

How does downsampling help in estimating the optimal σ and k for clustering the data? From the toy example, we observed that a large σ combines all eigenvectors (based on the eigengap heuristic approach) in one big cluster, while a very small σ forces each eigenvector in its own cluster. Hence, the number of clusters in the data can vary from 1 to n over a specified range of $\{\sigma_1,...,\sigma_t\}$ values. We can then find $K = \{k_1,...,k_t\}$ by using the eigengap heuristic approach for every σ. Initially, we will explore pre-fixed ranges of σ values, consisting of short equispaced discrete intervals such as: $\{0.05,...,0.50\}$, $\{1,...,5\}$, $\{10,...,15\}$, $\{40,...,60\}$, and $\{100,...,120\}$. These ranges will depend on the calculated standard deviation of the elements of the distance matrix and should not exceed the calculated standard deviation. For instance, if the standard deviation is 44, then we will explore the ranges of σ that are below this value. Later, we will only select the range that shows variations among the 15 smallest eigenvalues [1] for further and more detailed search, and will avoid any range of σ that result in only $k = 1$ or $k = 15$.

Based on our downsampling approach, the original data will be split into low resolution replicates as odd and even. It is possible to use all 10 pairs of odd and even replicates in the specific downsampling criterion as it was the case with the general downsampling criterion (Section 5.2). However, considering the computational cost of this practice we can rely on just the simplest form of odd and even pairs (i.e. the first pair). Each replicate will go through the same smoothing and then will be clustered by FSC-S technique over a pre-determined range of σ. Each value of σ will yield some k value resulting from the eigengap heuristic, and the clustering process will be repeated over the range of σ until K hits 1. Then, the clustering results of the odd replicate and the even replicate will be compared using the ARI to compute the stability score. Based on the concept of clustering stability, the highest stability score reflects the best choice of σ and k. However, usually a range of σ values will give the optimal results instead of only one specific value, indicating robustness in the choice of σ. For instance, if the range $[\sigma_i, \sigma_j]$ gives the same value of k in both replicates along with high ARI values, it would indicate clustering stability for those σ values and would support the choice of k. Obviously,

[1] We assume $1 < k < 15$, while if the number of curves n is less than 15, then $1 < k < n$.

high ARI values that are associated with $k = 1$ and a very large k ($k \approx$ number of curves in the data) will not be counted as clustering stability and will be excluded from the choices. Our extended algorithm is detailed in Figure 6.4:

Algorithm 2 FSC-S with downsampling approach

procedure CLUSTER FD & ESTIMATE(K, σ)
    Split the data into odd and even
    for each replicate **do**
        Smooth the data
        Compute the Distance matrix
        Examine global σ values
        Zoom in a specific range of σ
        for $i \leftarrow 1, z$ **do**
            $\sigma = i/d$ ▷ where d take any value: $\{1, 2, ..., 100\}$
            Compute matrices: $\mathbb{A}, \mathbb{D}$, and $\mathbb{L}$ ▷ regular steps of FSC-S
            Find k: $k_i = arg\{max(\lambda_{j+1} - \lambda_j)\}$
            Create $\mathbb{V}$ based on the k eigenvectors
            Normalize $\mathbb{V} \leftarrow \mathbb{Y}$
            Find k clusters by applying k-means on Y
        end for
        return K, σ,and the k clusters ▷ $K = \{k_1, k_2, ..., k_z, 1\}$
    end for
    for $i \leftarrow 1, z$ **do**
        Compute ARI: $ari(odd, even)$
    end for
end procedure

Figure 6.4: The specific downsampling criterion algorithm.

Applying the algorithm leads to results shown in Table 6.1. The table displays a list of σ

values besides their corresponding k for each copy, and the adjusted Rand index (ARI). Apart from $k = 1$ and $k = n$, the table suggests the highest ARI is 1 for $k = 4$ from both data sets at $\sigma = \sigma_i$. This is a simple example to show how the concept of our algorithm works, however, in real world examples it might not be as straightforward as this example. We might never get a match of k between the two copies, which usually occurs if the two sets are not alike, or if the data can be explained by more than one k. Either way, the specific downsampling criteria can still provide some information about the optimal clustering structure. Recall from Section 5.2, we observe that the downsampling criterion is not appropriate for sparse functional data analysis, which is also true while using downsampling for selection of σ and k. We have also observed that the optimal σ in the lower resolution replicates is slightly smaller than the optimal σ of the original functional data, but they reveal the same k. Thus, the resulting k from FSC-DSC is appropriate and in most clustering problems the clustering assignments of curves are appropriate. FSC-DSC works best when the functional data are dense, with a regular timeline.

σ	K (odd set)	K (even set)	ARI
$\sigma_1 \approx 0$	n	n	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
σ_5	6	7	0.32
$\vdots$	$\vdots$	$\vdots$	$\vdots$
σ_{i-1}	4	4	0.91
σ_i	4	4	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
σ_{z-6}	3	2	0.40
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\sigma_z \approx 1000$	1	1	1

Table 6.1: Simulated results of FSC-DSC algorithm to indicate the optimal σ and k for the functional data. The table suggests the optimal k is 4 with $\sigma = \sigma_i$.

We have initially attempted to use additional criteria to aid our choice of k such as the

AIC and BIC. However, since our approach is not a maximum likelihood estimation, we used a commonly presented version of AIC and BIC for K-means as presented in Towers (2013). Based on `kmeans` function in R and using the information of the residual sums of squares (RSS) after fitting the data. The formulas of AIC and BIC can be written as follows:

$$AIC = RSS + 2mk, \tag{6.1}$$

$$BIC = RSS + \ln(n)mk, \tag{6.2}$$

where:

- $n =$ number of observations,
- $k =$ number of clusters, and
- $m =$ number of dimensions.

In FSC-DSC, $n =$ number of curves, and $m =$ number of eigenvectors of the Laplacian graph chosen by the eigengap heuristic. Since in our approach the number of clusters is also determined by the eigengap heuristic, then m always equals k in equation (6.1) and equation (6.2). However we have observed that as k decreases and σ increases, the calculated AIC and BIC values decrease. Therefore, we could not rely on these values to confirm the results of FSC-DSC.

6.3 Application of FSC-DSC on the Berkeley Growth Data

We will now revisit the Berkeley growth data. We will apply FSC-S(D_1) to cluster the data based on its superior performance over FSC-S(D_o), as illustrated in Section 4.3. Unlike in Section 4.3 where we assumed $k = 2$ based on the natural grouping of gender among the children, here we will be estimating k by performing FSC-DSC.

The outcome of applying the algorithm on the first pair of odd and even replicates of the growth data is shown in Table 6.2. The two replicates give similar results of k with high ARI

from $\sigma = 0.5$ to $\sigma = 0.69$. Although we have mentioned that in general we will limit the application of FSC-DSC on only one pair of odd and even replicates, we have used all the pairs in the growth data, mainly because we want to make sure that the different pairs will perform similarly. We found that all pairs yield more or less similar results, as is evident from Table 6.2. There is only one set, where the replicates coming from the logical set {T,T,T,F,F,F} and {F,F,F,T,T,T} for odd and even, respectively, showed different results. In this case, the odd replicate was able to detect the 2 clusters in the curves, while the even set gave a further split for one of the clusters to be in total 3 clusters. A summary of the results for this case is shown in Table 6.3.

For more detailed outcomes of the algorithm, Figure 6.5 displays some selected results of the first replicate for the odd set at $\sigma = 0.32, 0.5$, and 1, shown in Figure 6.5a, 6.5b, and 6.5c respectively. The figures explicitly show the effect of σ on the eigenvalues which in turn will lead to the choice of k through the eigengap heuristic approach. In addition, Figure 6.6 summarizes the ARI of comparing the two replicates over a range of σ values. It shows that when k of the odd set (set 1) fully matches with k of the even set (set 2), the ARI achieves a value of 100%. On the other hand, when there is a mismatch between the two k's, the ARI drops to lower values. The figure also illustrates that the estimated k values from the algorithm take only a few unique values: 92, 84, 42, 3, and 2. The algorithm assigns very high k for both sets over $\sigma = \{0.01, ..., 0.33\}$ and then drops quickly to smaller k until the two sets settled at $k = 2$ before they finally reach $k = 1$. Considering the clustering results of $k = 2$ at the optimal range of σ gives an accuracy rate of 91% when compared to the gender grouping of the data.

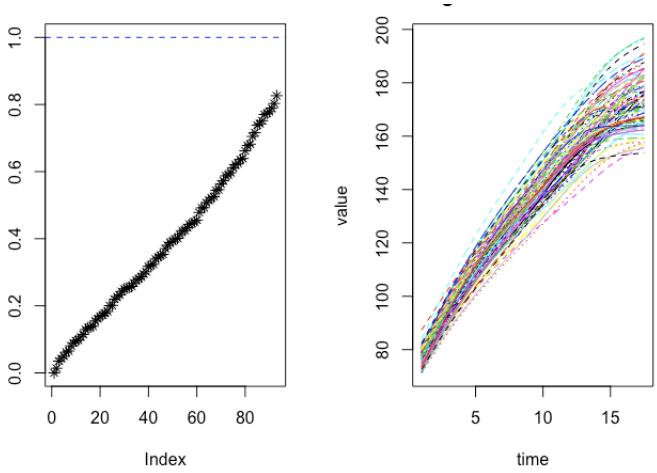
σ	K (odd set)	K (even set)	ARI
0.01	92	92	1
0.30	92	84	0.27
0.33	92	42	0.022
0.34	2	42	0.04
0.36	3	42	0.064
0.38	3	2	0.70
0.5	2	2	0.96
0.54	2	2	1.00
⋮	⋮	⋮	⋮
0.69	2	2	1.00
0.70	2	1	0
0.71	1	1	1
1.00	1	1	1

Table 6.2: Selected results of FSC-S(D_1) algorithm with FSC-DSC on the growth data. The table suggests $k = 2$ according to the highest ARI. The shaded area shows the highest ARI reflected from a match of the two K's over the optimal σ values.

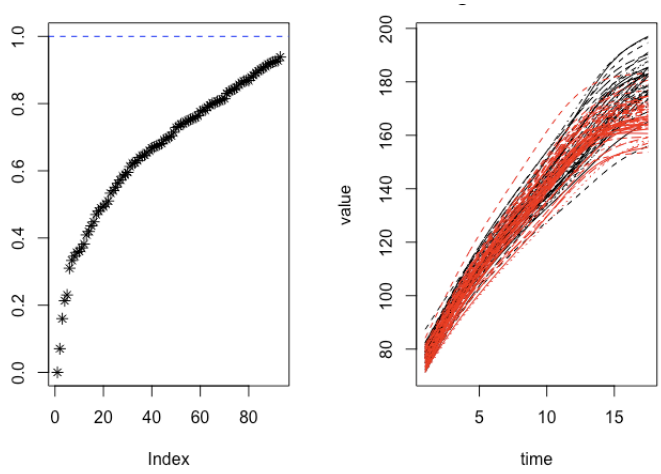
σ	K (odd set)	K (even set)	ARI
0.01	92	92	1
0.40	2	29	0.073
0.41	2	3	0.63
0.50	2	1	0
0.99	2	1	0
1.00	1	1	1

Table 6.3: Selected results of FSC-S(D_1) algorithm with FSC-DSC on the growth data. Where there is no match for a pair of odd and even sets in terms of k. Note that using the ARI criterion we arrive at 2 or 3 clusters for ARI = 0.63.

(a) $\sigma = 0.32, k = 92$



(b) $\sigma = 0.5, k = 2$



(c) $\sigma = 1, k = 1$

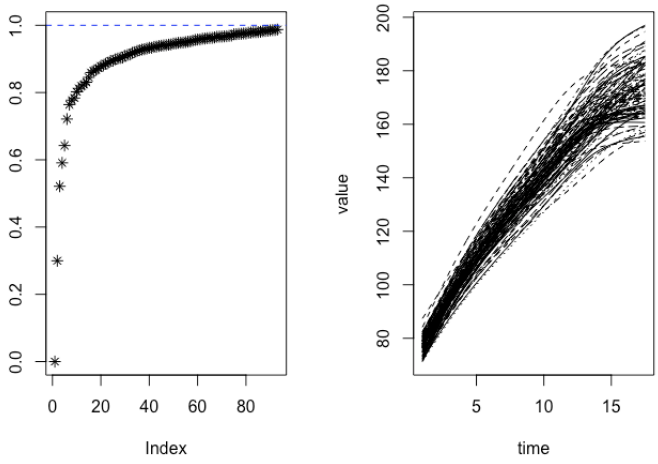


Figure 6.5: Some of the resulting graphs of the application of FSC-DSC to the Berkeley growth data (the odd replicate). The left panel shows the eigenvalues based on the given σ while the right panel shows the clustered curves based on the chosen k.

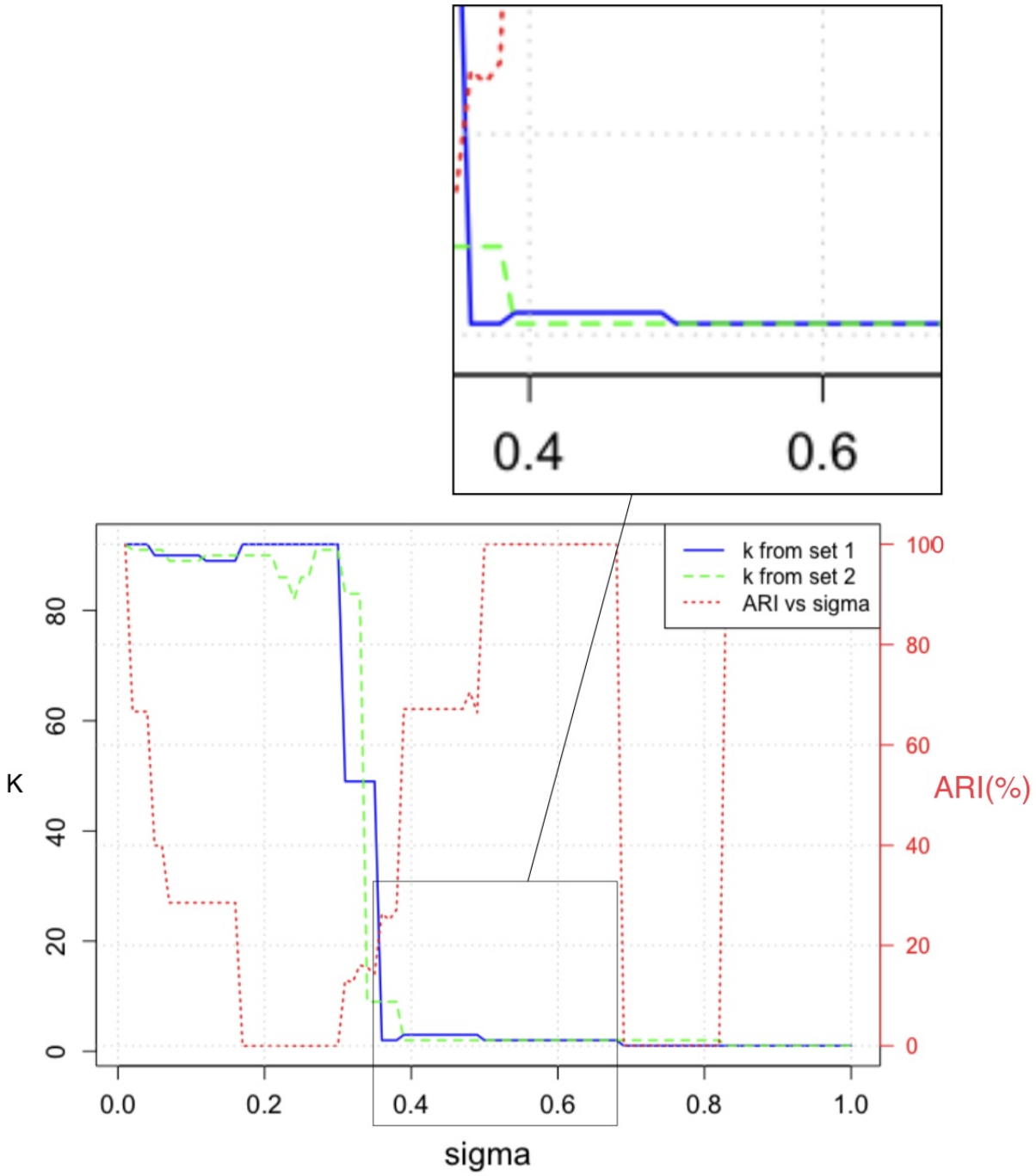


Figure 6.6: A diagram of ARI shows the overall results of comparing k of the odd (set 1) and even (set 2) over a range of σ. Initially at $\sigma \simeq 0$, the ARI starts as 100% since both sets give $k = 92$ but it immediately drops to very low values as σ increases. Starting from around 0.5 the two sets start to coincide in clustering the curves which is reflected in high ARI and this ends when σ hits 0.7 where first $k = 1$ and by that the ARI drops to 0. The zoomed-in picture shows the match between the two sets at $\sigma = [0.5, 0.69]$.

On the other hand, if we consider the specific downsampling criterion with FSC-S(D_o), the optimal number of clusters is different. The original curves of the growth data represent the heights of children, thus these curves will normally group the data into different heights categories regardless of their gender. The results of applying the algorithm of the first pair of odd and even replicates is shown in Table 6.4 and in Figure 6.7. According to the results, the value for k is 5 clusters, which is clear from the significant jump from $k > 90$ for both sets to $k = 5$ before the two sets settle at $k = 1$. We notice that the ARI fluctuates around 87%, due to a few curves moving between clusters at different σ values. Figure 6.8 represents the clusters for $k = 5$. The bigger group consists of 46% of the curves including both boys and girls, which could be considered as a middle category (displayed as green curves in Figure 6.8), while the children with relatively lower heights are only 8 and are all girls (the red group). Whereas, children who are relatively tall are only 5 boys and 1 girl (the black group).

σ	K (odd set)	K (even set)	ARI
0.1	92	92	1
0.2	92	92	1
0.3	92	91	0.67
⋮	⋮	⋮	⋮
1.5	92	91	0.67
1.6	91	91	1
1.7	5	5	0.84
1.8	5	5	0.89
⋮	⋮	⋮	⋮
3.5	5	5	0.87
3.6	1	1	1

Table 6.4: Some selected results of FSC-S(D_o) algorithm with FSC-DSC on the growth data. The table suggests $k = 5$ according to the highest ARI. The shaded area shows the highest ARI reflected from a match of the two K's over the optimal σ values.

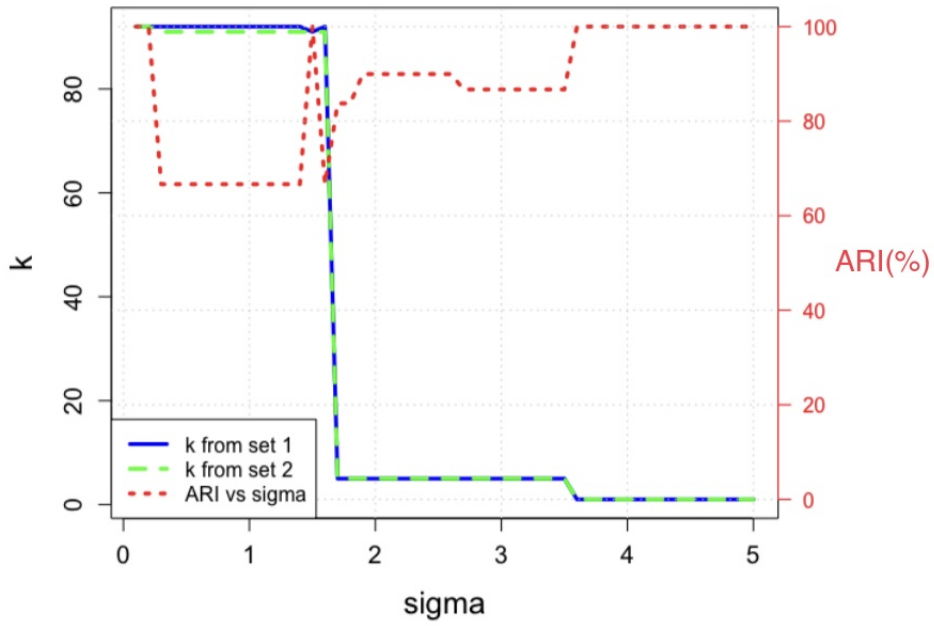


Figure 6.7: A diagram of ARI shows the overall results of comparing k of the odd and even sets over a range of σ. Both sets start at $k = 92$ and $k = 91$, then at $\sigma = 1.7$ they give $k = 5$ with ARI averaged 87%.

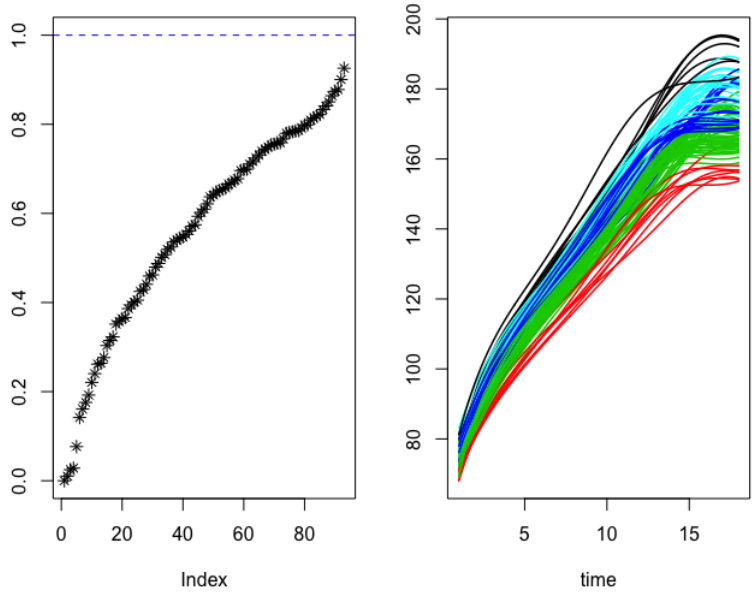


Figure 6.8: The left graph shows the eigenvalues with clear eigengap at 5, while the right graph shows the 5 clusters of the Berkeley growth data. Results from the odd set.

It is well known that the original scale will often demonstrate a different interpretation of the data than the first derivatives. This concept is clear with the growth data, where the original curves hold information about the heights of children while the first derivatives hold information about the rate of change (growth rates) in children's heights. The rate of change in heights is linked to puberty, thus it can often discriminate boys from girls. Therefore, if we are interested in clustering the data based on gender, then using FSC-DSC with FSC-S(D_1) is more informative than using FSC-S(D_o).

It should be mentioned that the results of FSC-S(D_1) in the general DSC (Section 5.3) are compatible with its results in the specific FSC-DSC. However, FSC-S(D_o) in the general DSC suggests $k = 2$, but it does not support $k = 5$. There are two main reasons for this apparent mismatch. First, while FSC-DSC estimates the number of clusters from the domain of the data, general DSC clusters the data according to pre-determined k values. Second, and most importantly, σ in the general DSC is fixed and equal to the standard deviation of the elements of the distance matrix as explained in Section 4.2, whereas σ is a variable parameter and plays an important role in the final clustering results in FSC-DSC. For instance using FSC-S(D_o), $\sigma = 18$ in the general DSC, which is far bigger than $\sigma = [1.7, 3.5]$ in the specific DSC. While for FSC-S(D_1), $\sigma = 2.8$ in the general DSC and $\sigma = [0.5, 0.69]$ in the specific DSC. Based on the Berkeley growth data, we noticed that even if the specific and general DSC do not fully agree in the final clusters, they both lead to reasonable results. More details will follow in Chapter 7 and Chapter 9.

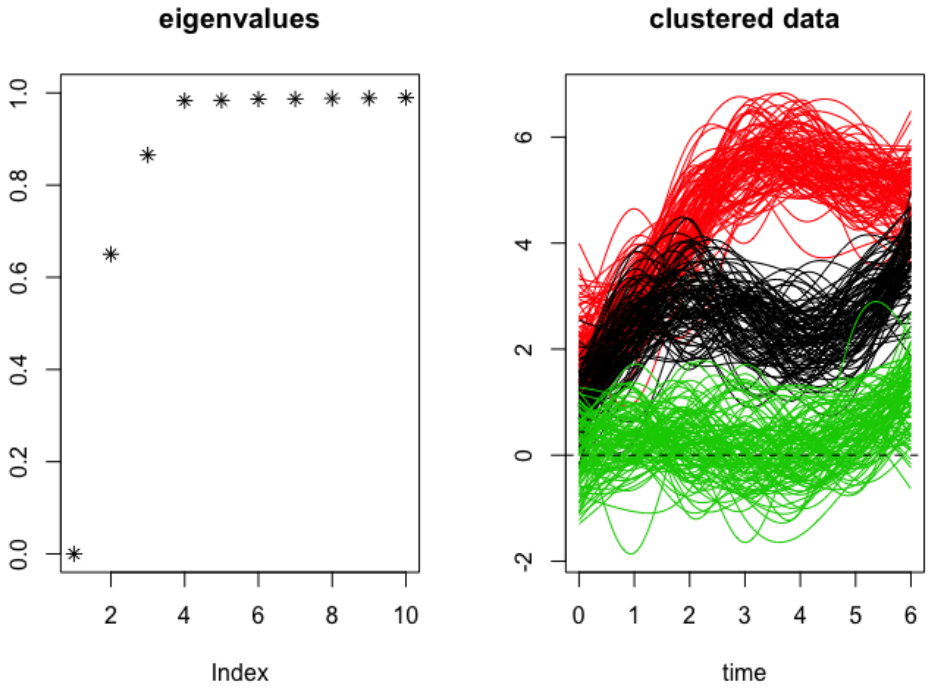
6.4 Chapter Summary

In this chapter, we have addressed the limitations of the FSC-S approach with respect to the number of clusters, and proposed a criterion to optimally estimate the number of clusters. In particular we have proposed the specific downsampling criterion to estimate the optimal number of clusters k at the optimal range of σ. The addition of the downsampling method added a distinctive feature to our approach. The choice of k and σ is an open-ended question in spectral

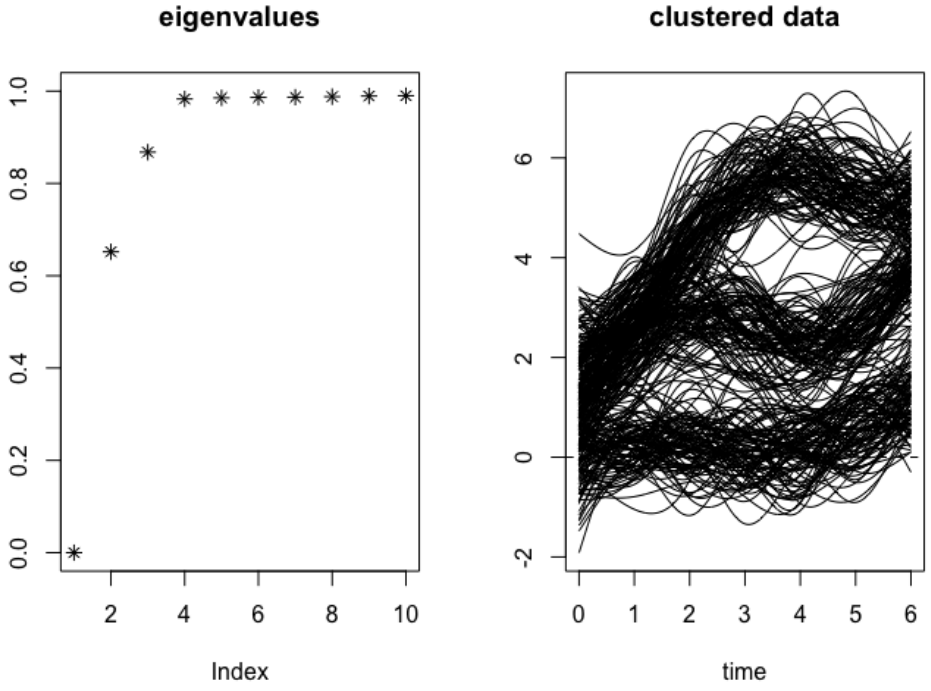
clustering and our FSC-DSC approach provides promising results in answering the question in the context of clustering functional data.

The success of our approach can be explained by the behaviour of σ and the Laplacian graph. For instance, moving from low σ to high σ, we move from the scenario where each curve is a cluster to the scenario where all curves in one big cluster. During this process k does not change continuously but abruptly, which suggests that the k values tend to identify inherent clustering in the data structure. We can presume that σ creates a natural threshold which in turn divide the data into groups. The Laplacians graph makes sure that the block diagonal of eigenvectors are in a meaningful order, where the eigenvectors that hold more information about the variation in the data come first. Hence, it does not miss a cluster nor duplicate a cluster, and thus preserve all the information about the clusters in the first k eigenvectors.

Based on several applications we have found that the parameter σ may change the eigenvalues but does not alter the eigenvectors. This explains why the functional spectral clustering algorithm is capable of performing efficiently if k is known a-priori and was provided to run the algorithm, irrespective of the chosen σ value. To illustrate the concept we will reuse the toy example that was introduced in Section 6.1 with relatively high noise $\xi = 0.45$. First, we will assume that k is known and equal to 3 clusters, FSC-S(D_o) will perform nicely and cluster the curves properly given that $\sigma = 3$ which is calculated from the standard deviation of the elements of the distance matrix (Figure 6.9a). Second, we will assume k is unknown, and the algorithm will attempt to estimate k from the eigengap heuristic given $\sigma = 3$. In this case, k will be estimated as 1 cluster, due to the way σ determines the width of the neighbourhoods (Figure 6.9b). However, implementing FSC-DSC in this example will give the right k at the proper range of σ as shown in Figure 6.10. Since the clustering structure will be stable at the optimal range of σ, the eigengap heuristic approach will repeatedly give $k = 3$ in both the odd and the even sets. The final clustering results from applying FSC-S(D_o) providing $k = 3$, and from applying FSC-DSC, fully agree with the correct grouping of the data.



(a) FSC-S(D_o) with $\sigma = 3$ and $k = 3$



(b) FSC-S(D_o) with $\sigma = 3$ and k is estimated by the eigengap to be 1

Figure 6.9: Results of applying FSC-S(D_o) with any random choice of σ on the toy example (a) when k is known priori and supplied, and (b) when k is unknown and is estimated by the eigengap heuristic.

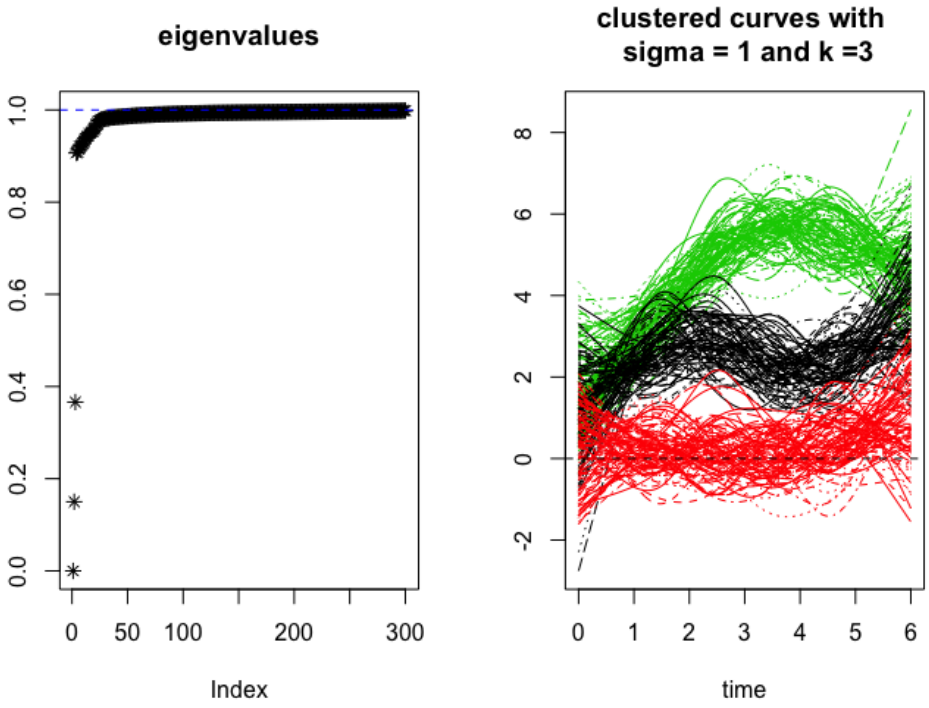


Figure 6.10: Results of applying FSC-DSC approach on the toy example suggests $k = 3$ at $\sigma = \{0.2, ..., 1.4\}$ for both the odd and the even replicates.

Based on this work we can suggest that if k is given, the FSC-S technique is an appropriate choice to cluster the data and computationally very fast. Whereas if k is unknown, it is recommended to use FSC-DSC to estimate the optimal parameters k and σ and use their corresponding clustering results. It should be mentioned that although the algorithm cluster several replicates and compare the pairs repeatedly, the computational time is still relatively low. However, the computational time increases as the range of σ values to be explored increases. Besides, the algorithm runs slower when k is very large ($k \approx$ number of curves in the data) but runs fast at small k values.

Chapter 7

Simulation Studies and Comparisons with Existing Methods

In this chapter we set up a host of different simulation scenarios to compare the performance of the chosen clustering functional data approaches, to our newly proposed functional spectral clustering techniques. In addition, we examine the performance of the general and the specific downsampling criteria for a wider set of scenarios beyond the examples demonstrated in Chapter 5 and 6. The first section outlines our aims and objectives and highlights the main points of interest behind the study. Section 7.2 presents several examples of functional data that consist of variations primarily in phase and amplitude. Using these examples we also demonstrate how downsampling is effective in choosing the number of clusters. Section 7.3, presents another simulation scheme, obtained by perturbation of the Canadian weather data, together with the comparison of clustering techniques on these perturbations.

7.1 Introduction

A common strategy used to evaluate the performance of a new statistical approach is through well designed simulations. To examine the performance of our algorithms, we developed a comprehensive simulation scheme that cover different scenarios and formats of functional data. In this study our primary interest is to investigate the performance of the proposed algorithms on

specific scenarios, such as; (1) functional data with phase and amplitude variations, (2) dense functional data, and (3) sparse functional data. Further, we will attempt to create different levels of difficulty within the above mentioned scenarios by varying the inherent noise in the data, creating scenarios which are hard to cluster. We will also demonstrate a procedure that uses a real functional data set to create simulated data based on perturbation. In addition, we will evaluate the model selection ability of the downsampling criteria which are designed to select the optimal number of clusters. In some cases we may not get a unique answer and which case we will provide the interpretation of super-clusters and sub-clusters.

Despite the wide range of clustering functional data methods, there exists no comprehensive study to compare their performances beyond their own contexts and specific examples they have been applied on. Therefore, based on the simulations we aim to compare our clustering algorithms with the chosen CFD methods (as detailed in Section 3.3). Recall that among the different CFD approaches, FunHDDC is a model-based clustering technique, FD-Kmeans is a nonparametric clustering technique, and B-splines-Km and FPCA-mbc are two-stage clustering techniques.

The main objective is to investigate the strengths and weakness of our functional spectral clustering techniques (FSC-S(D_o), FSC-S(D_1), FSC-S(D_2)) and the downsampling based model selection approaches (the general DSC, and the specific FSC-DSC). We aim to gain more insight and knowledge about their use in the different scenarios. This chapter provides key contributions to the field of clustering functional data in terms of simulation schemes, clustering analysis, and model selection.

7.2 Functional Data with Phase/Amplitude Variations

In this section, we build a framework to develop a simulation study to show the performance of the functional spectral clustering approaches on functional data, that involves shifts in either phase, or amplitude or both. A similar simulation scheme was previously introduced by Sangalli

et al. (2010), but we have expanded and made a few modifications to the scheme to cover more scenarios. Sangalli et al. (2010) clustered the data by a k-means alignment algorithm which aligns and clusters the curves simultaneously, and is based on detecting the amplitude cluster and the phase clusters in curves. The primary goal of our investigating is to determine whether FSC-S techniques can inherently find the clusters in the functional data without explicitly modeling the phase and amplitude variations. In addition, we will apply the downsampling-based approaches to investigate if the proposed approaches are capable of detecting the true clusters in the data. We have previously worked with this simulation in slightly different settings of the scenarios as discussed in Al Alawi et al. (2019).

7.2.1 Simulation Scheme

The simulation was generated as aperiodic data spanning the range from 0 to 2π. The initial model of the data is coming from prototype (7.1), that is, a simple function with neither phase shift nor amplitude shift. Introducing an amplitude shift in the data can be done through prototype (7.2) which gives a slightly different shape. Further, we add to (7.2) a phase shift, and we stretch the function over a larger period to form prototype (7.3). In a similar manner to (7.3), we create prototype (7.4) that displays a different phase shift and a more stretched function.

$$f(t) = \sin(t) + \sin\left(\frac{t^2}{2\pi}\right), \tag{7.1}$$

$$f(t) = 2\sin(t) - \sin\left(\frac{t^2}{2\pi}\right), \tag{7.2}$$

$$f(t) = 2\sin\left(-\frac{1}{3} + \frac{3}{4}t\right) - \sin\left(\frac{(-\frac{1}{3} + \frac{3}{4}t)^2}{2\pi}\right), \tag{7.3}$$

$$f(t) = 2\sin\left(-\frac{1}{3} + \frac{1}{2}t\right) - \sin\left(\frac{(-\frac{1}{3} + \frac{1}{2}t)^2}{2\pi}\right). \tag{7.4}$$

A simple illustration of the above mentioned functions is displayed in Figure 7.1. The figures show 5 smoothed curves from each prototype with very small error. Since some of our proposed

clustering techniques involve the functions' derivatives, the figures also show the first derivative (second row) and the second derivative (third row) of each prototype. From the figure, we notice that there is some similarity between prototype 1 and prototype 2, because they only vary in their amplitudes, and this is even more clear from their second derivatives. In contrast, prototype 3 and prototype 4 each led to a distinct curvature structure and this is also reflected in their first and second derivatives. To develop the full simulation scheme we will generate a series of curves along with a set of noise variance to generate the raw data. Thus, we set a general equation for each prototype to simulate functional data with observational error ε's distributed normally with mean = 0 and standard deviation = 0.05. Equations: (7.5), (7.6), (7.7), and (7.8) represent an extended format of the prototypes (7.1), (7.2), (7.3), and (7.4) respectively.

$$\begin{aligned} f(t) &= (1+\varepsilon_{1i}) * \sin(\varepsilon_{3i} + (1+\varepsilon_{4i}) * t) \\ &\quad + (1+\varepsilon_{2i}) * \sin\left(\frac{(\varepsilon_{3i} + (1+\varepsilon_{4i}) * t)^2}{2\pi}\right), \end{aligned} \tag{7.5}$$

$$\begin{aligned} f(t) &= (2+\varepsilon_{1i}) * \sin(\varepsilon_{3i} + (1+\varepsilon_{4i}) * t) \\ &\quad - (1+\varepsilon_{2i}) * \sin\left(\frac{(\varepsilon_{3i} + (1+\varepsilon_{4i}) * t)^2}{2\pi}\right), \end{aligned} \tag{7.6}$$

$$\begin{aligned} f(t) &= (2+\varepsilon_{1i}) * \sin\left(\varepsilon_{3i} + (1+\varepsilon_{4i}) * (-\frac{1}{3} + \frac{3}{4}t)\right) \\ &\quad - (1+\varepsilon_{2i}) * \sin\left(\frac{\left(\varepsilon_{3i} + (1+\varepsilon_{4i}) * (-\frac{1}{3} + \frac{3}{4}t)\right)^2}{2\pi}\right), \end{aligned} \tag{7.7}$$

$$\begin{aligned} f(t) &= (2+\varepsilon_{1i}) * \sin\left(\varepsilon_{3i} + (1+\varepsilon_{4i}) * (-\frac{1}{3} + \frac{1}{2}t)\right) \\ &\quad - (1+\varepsilon_{2i}) * \sin\left(\frac{\left(\varepsilon_{3i} + (1+\varepsilon_{4i}) * (-\frac{1}{3} + \frac{1}{2}t)\right)^2}{2\pi}\right). \end{aligned} \tag{7.8}$$

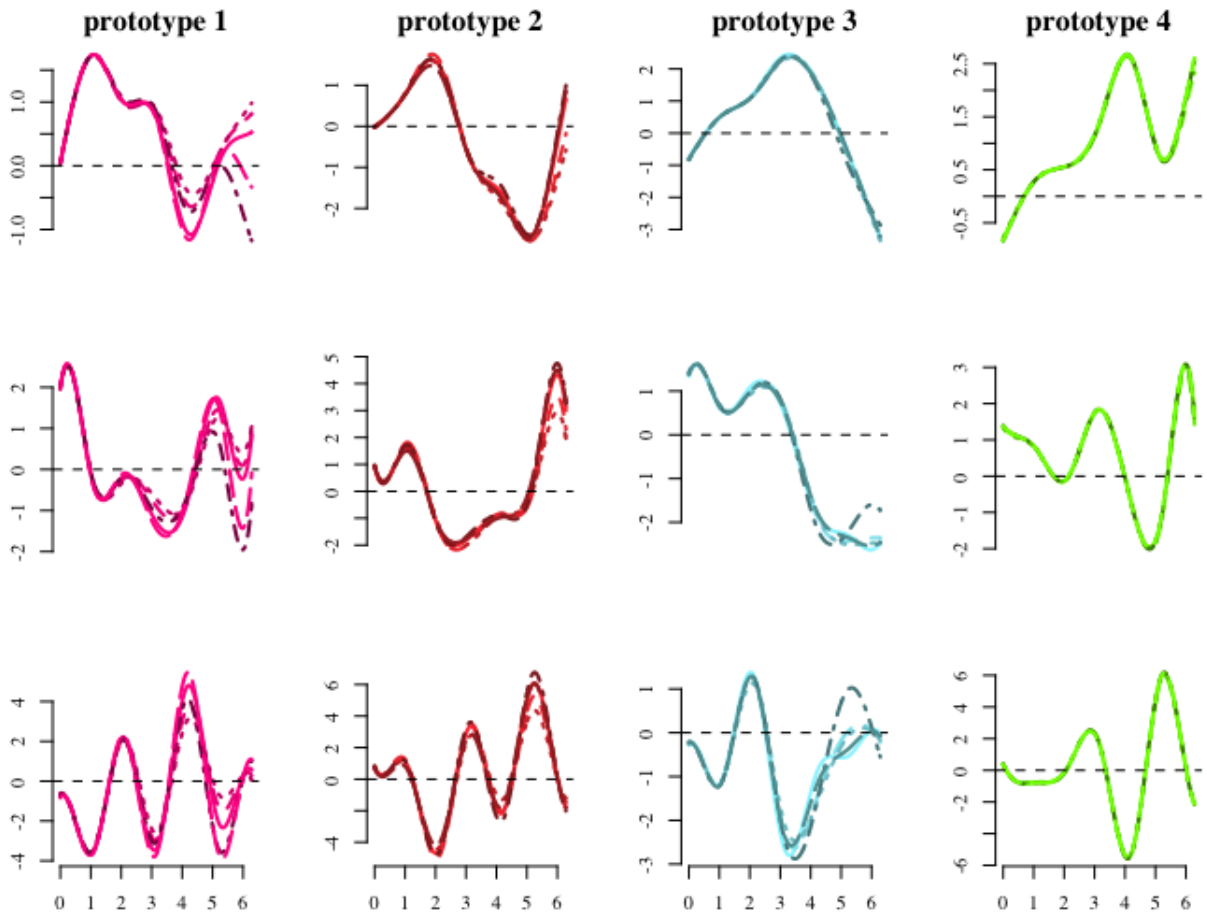


Figure 7.1: Curves simulated from prototype 1, 2, 3, and 4 are displayed in first row. Second row displays the first derivatives of each prototype while third row displays their second derivatives.

While fitting the curve from the raw data we go through our general smoothing approach, which is using B-splines of order 4 with a saturated model and a penalty term that best fits the data, which is $\lambda = 10^{-3}$. Based on the above, we created 4 different scenarios of functional data sets as shown in Figure 7.2. The description of how each case was obtained is detailed below:

- The first scenario is **case A** that consists of 90 simulated curves from equation (7.5). Case A consists of only 1 group, thus it would not be included in the clustering process later, but it is considered as a template to build the other scenarios.

- The second scenario is **case B**. In this case there are two groups where the first group consists of 45 curves (i=1, ..., 45) that come from equation (7.5), while the other group consists of another 45 curves (i=46, ...,90) that come from equation (7.6).

- The third scenario is **case C**. In this case there are 3 groups; the first 30 curves (i=1, ... , 30) come from equation (7.5), the second 30 curves (i=31, ... , 60) come from equation (7.6), and the last 30 curves (i=61, ... , 90) come from equation (7.7).

- The last scenario (and the most difficult for clustering) is **case D**. In this case, the curves are obtained as follows: 20 curves (i= 1, ... , 20) come from equation (7.5), 20 curves (i=21, ... , 40) come from equation (7.6), 20 curves (i=41, ... , 60) come from equation (7.7), and finally 30 curves (i=61, ... , 90) come from equation (7.8). This pattern suggests the existence of 4 groups. However, we also attempted to create a higher level of grouping in the curves, where, the first and second sets of curves form the first super-cluster, and the third and fourth sets of curves form the second super-cluster. This clustering comes from the fact that in the first super-cluster there is no phase shift and the curves span over a range of 2π. On the other hand, the second super-cluster consists of curves that display phase shift and span over a more stretched period, since the coefficient of t^2 is not 1 in equations (7.3) and (7.4).

It is of interest to test the performance of our proposed functional spectral clustering method FSC-S on these simulated data, and comparing that with the performance of the other chosen functional clustering approaches. In addition, we intend to examine the general downsampling criteria DSC and the specific downsampling criteria FSC-DSC in finding the optimal k of the simulated data. For this reason, we have created case D, which is a more challenging scenario in terms of number of clusters. A visual representation of the different scenarios with their true clusters is shown in Figure 7.3.

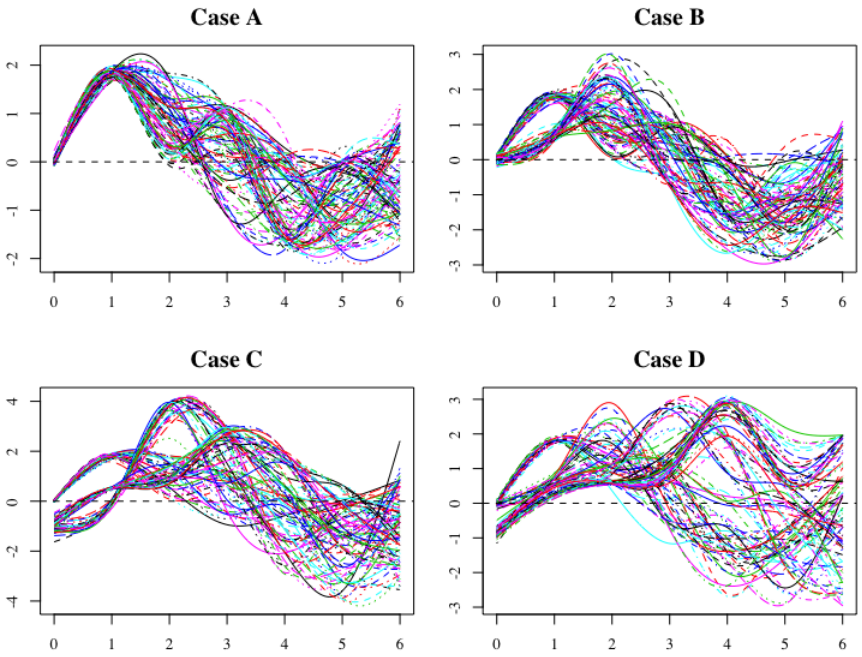


Figure 7.2: Smoothed curves simulated in case A, case B, case C, and case D. Note the colours are generated by the `fda` package and have no specific meaning.

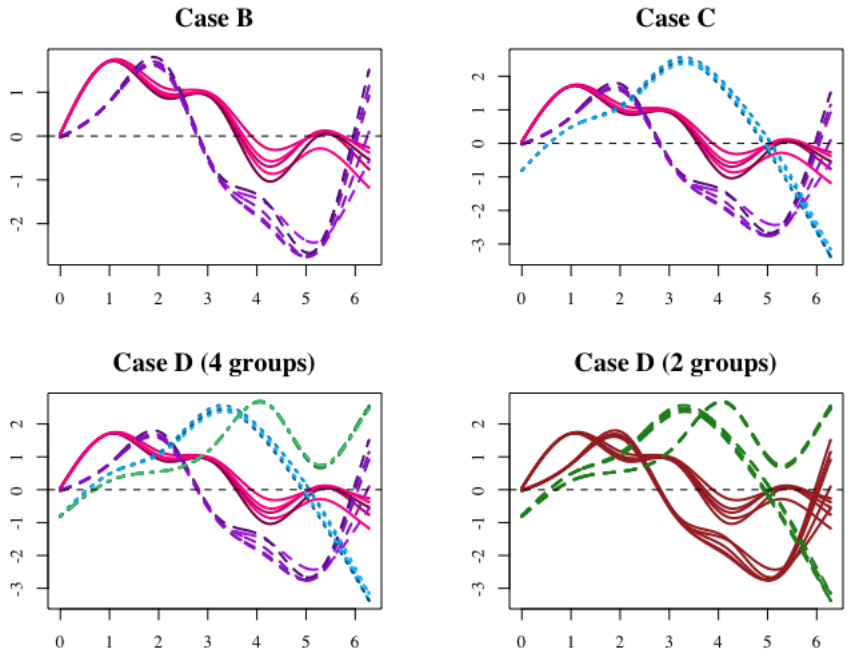


Figure 7.3: Sampled curves with low noise to show the clusters in the different scenarios. Case B displays 2 groups, case C displays 3 groups, while the groups in case D can be considered as 4 groups or 2 groups.

7.2.2 Application of FSC-S Approaches

In this section we present the results of clustering the simulated functional data with phase and amplitude variation using the FSC-S techniques and the competing CFD approaches.

We created 100 sets of data for each scenario and applied every clustering algorithm on these 100 sets. Initially, we provided correct number of clusters k for all the clustering approaches. At each iteration we calculated the correct classification rate (CCR) based on the true clusters for each scenario. The final results will be the average of the correct classification rate for the 100 iterations. The results are summarized in Table 7.1, while more details of CCR are shown in terms of boxplots in Figures 7.4, 7.5, 7.6, and 7.7 for case B, case C, case D (4 groups), and case D (2 groups) respectively.

Mean of Correct Classification Rate (CCR)				
Method	Case B (2 groups)	Case C (3 groups)	Case D (4 groups)	Case D (2 groups)
FunHDDC	0.86 (0.055)	0.83 (0.215)	0.85 (0.109)	0.75 (0.042)
FD-Kmeans	0.83 (0.071)	0.80 (0.150)	0.76 (0.128)	0.95 (0.035)
B-splines-km	0.90 (0.044)	0.92 (0.170)	0.95 (0.036)	0.80 (0.051)
FPCA-mbc	0.73 (0.123)	0.97 (0.032)	0.90 (0.109)	0.78 (0.023)
FSC-S(D_o)	**0.93** (0.028)	**0.99** (0.014)	**0.96** (0.022)	0.92 (0.046)
FSC-S(D_1)	0.72 (0.046)	0.91 (0.024)	0.84 (0.047)	**0.98** (0.018)
FSC-S(D_2)	0.72 (0.080)	0.81 (0.079)	0.65 (0.066)	**0.98** (0.019)

Table 7.1: Mean CCR of the clustering methods when applied on the simulated data. Note: Bold digits represent the best value within a column, and values in brackets represent standard deviation of the CCR.

According to the average accuracy rates, the clustering methods perform differently in the different scenarios. For instance, FSC-S(D_o) achieves high accuracy rates in all scenarios except in case D if we assume that the truth is 2 clusters. On the other hand, FSC-S(D_2) is performing

poorly in all scenarios except in case D (2 groups). Whereas, FSC-S(D_1) is performing well in specific scenarios, which are case C and case D (2 groups). In general, there is a tendency to achieve high CCR in case C compared to the other cases, which suggests it is relatively easy scenario. However, FunHDDC, FD-Kmeans, and B-splines-Km consists of many iterations that give very low accuracy rates and thus lower average CCR in this case. It is also noticed that FPCA-mbc performs worst in the scenarios that assume only 2 clusters. In addition, the clustering methods that show good performance in detecting the super-clusters of case D, show poor performance in detecting the 4 sub-clusters (true clusters). To consider the overall performance of the clustering methods at each scenario individually, we refer to the boxplots of CCR in Figures 7.5, 7.6, 7.7, and 7.8. In general, the results of the median CCR support the mean CCR in Table 7.1. There are only a few cases when the median CCR is very different from the mean CCR, for instance, in case C of FunHDDC the median CCR (Figure 7.6) is much higher and approximately 98% compared to 83% for mean CCR.

Further, we compare only the FSC-S techniques against each other. We notice that FSC-S(D_o) can detect the true clusters properly and locate the super-clusters with good accuracy rates as well. While the use of first derivatives (FSC-S(D_1)) can achieve good accuracy rates, the use of second derivatives (FSC-S(D_2)) does not in general add any advantages to the clustering results, in fact it gives lower accuracy rates, except for that case D (2 groups) where the derivatives can reveal the distinct structures in the super-clusters more efficiently, see Figure 7.1. Based on this, we can say that if there are phase and amplitude variations on the original curves scale, then FSC-S(D_o) is able to identify the hierarchical clustering structure that is embedded in these variations. On the other hand, the first and second derivatives would not reflect the phase and amplitude variations as in their original form, thus it would be harder to detect the clustering structure when using the derivatives to measure the distances between the functions in order to apply spectral clustering. However, FSC-S(D_1) and FSC-S(D_2) were able to find the 2 super-clusters in case D with very high accuracy rates. This is mainly because the first super-cluster spans over a different period than the second super-cluster and this fact can be more obvious in the derivatives than in the original curves. Figure 7.4 displays the clustered curves of data based on the FSC-S(D_o) results.

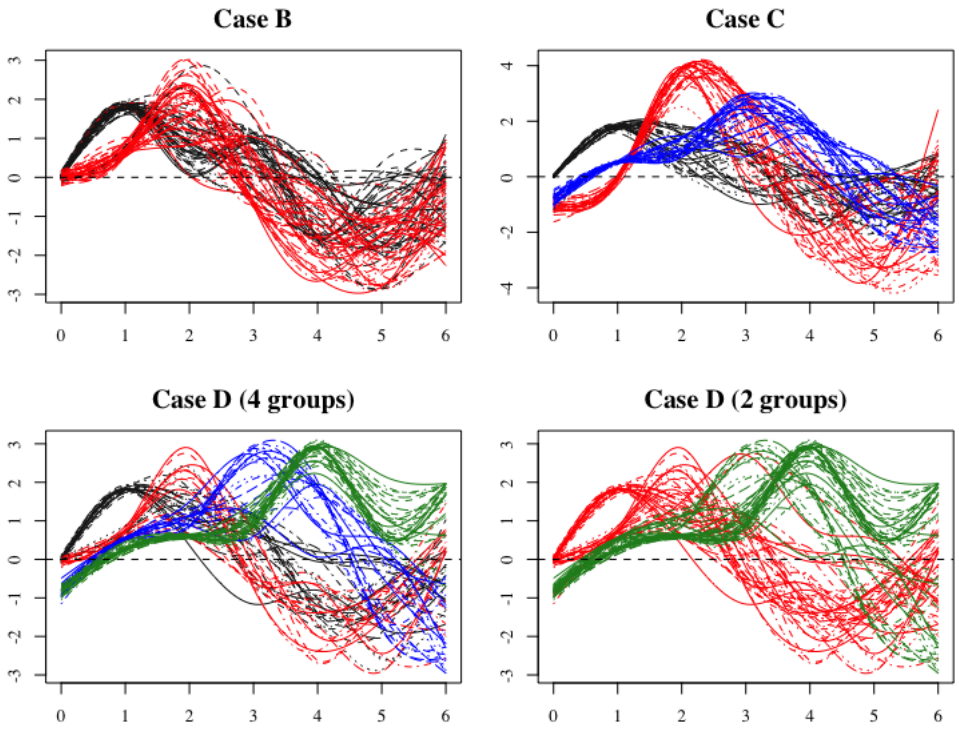


Figure 7.4: The clustering results of FSC-S(D_o) on the simulated data.

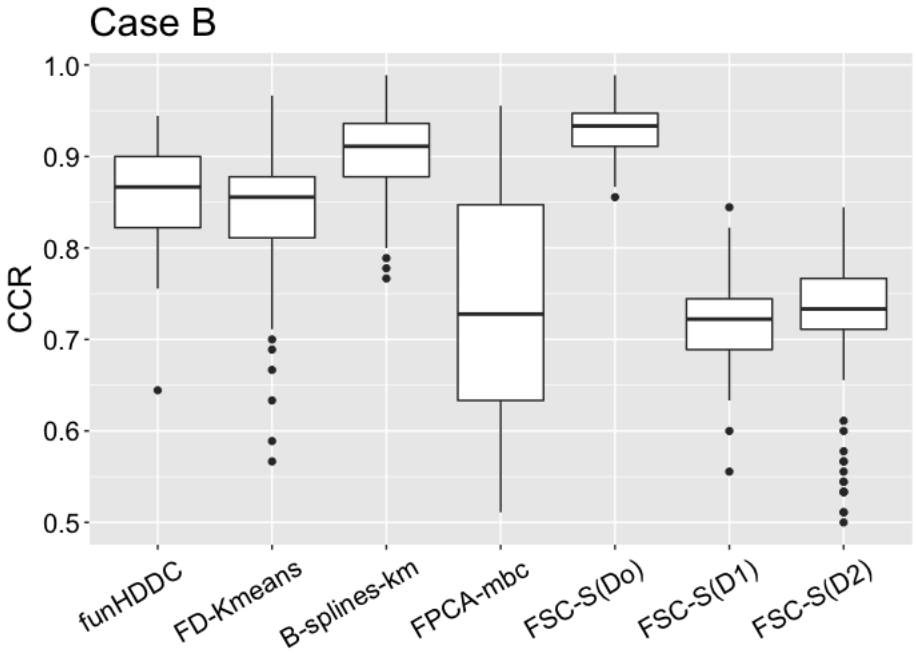


Figure 7.5: Mean CCR for the clustering methods when applied to the simulated data case B.

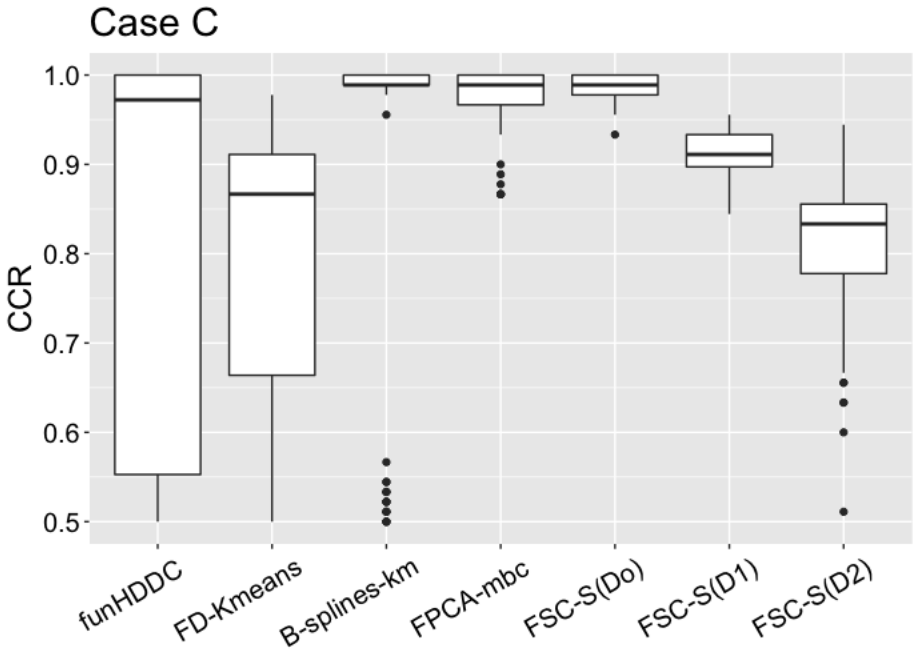


Figure 7.6: Mean CCR for the clustering methods when applied to the simulated data case C.

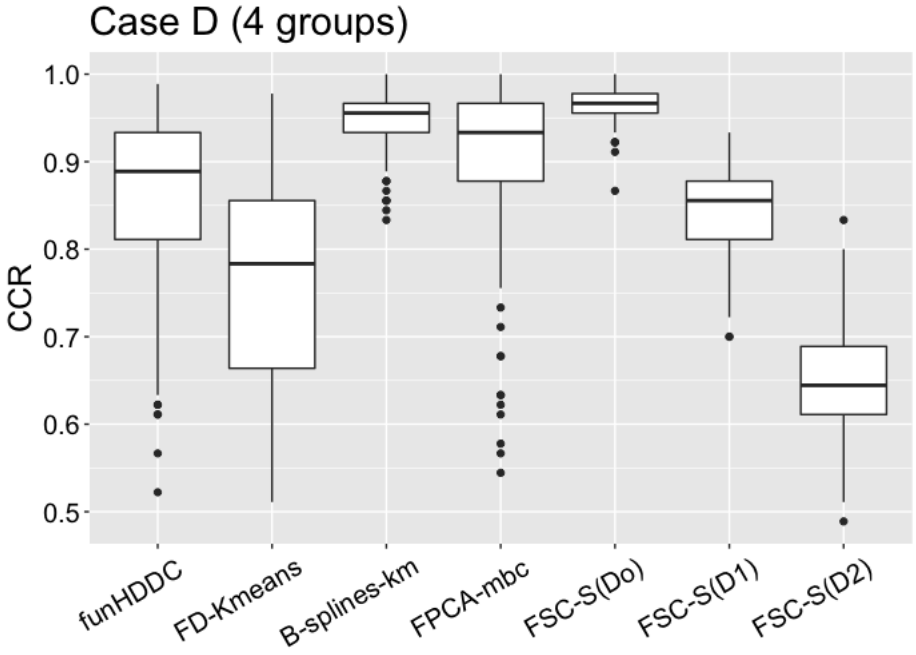


Figure 7.7: Mean CCR for the clustering methods when applied to the simulated data case D, assuming there are 4 groups.

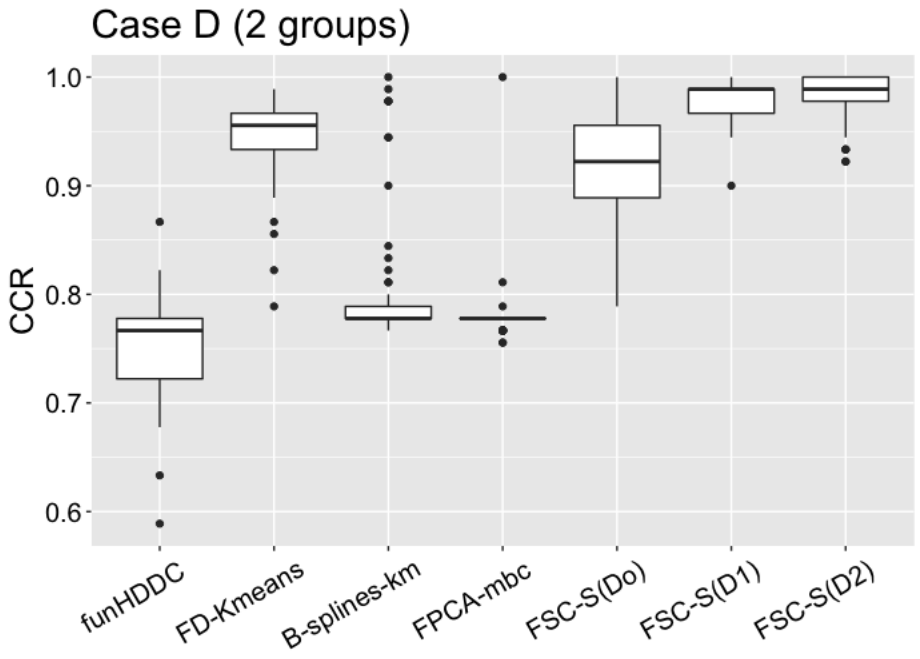


Figure 7.8: Mean CCR for the clustering methods when applied to the simulated data case D, assuming there are 2 groups.

7.2.3 Application of Specific Downsampling Criterion

In this section we will again consider the above simulated functional data and now apply the specific downsampling criterion. However, as these data are very sparse and dynamic, splitting them up directly into odd and even replicates will not create similar copies. In fact, the split replicates will sometimes lose the original structure of the generated curves, see Figure 7.9. Taking into account these limitations, we simulated the data set by adding more time points within the same range $[0, 2\pi]$ to make it dense. Now, dividing the adjusted simulated data into odd and even replicates keeps the structure of the original curves and creates similar copies, see Figure 7.10. Note that we followed a similar smoothing model to the previous one but changed the smoothing parameter to $\lambda = 10^{-2}$ for each replicate based on the updated value of the GCV.

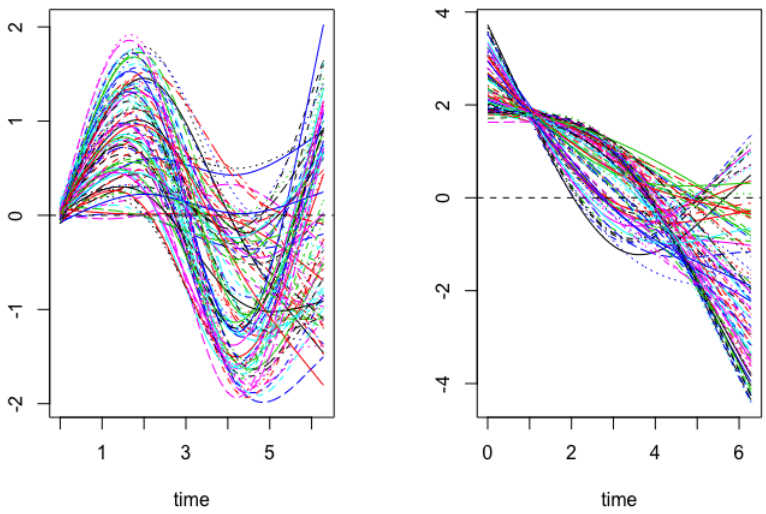


Figure 7.9: Example of downsampling the sparse aperiodic data case A into 2 replicates. The new functional data sets diverged from the original curves.

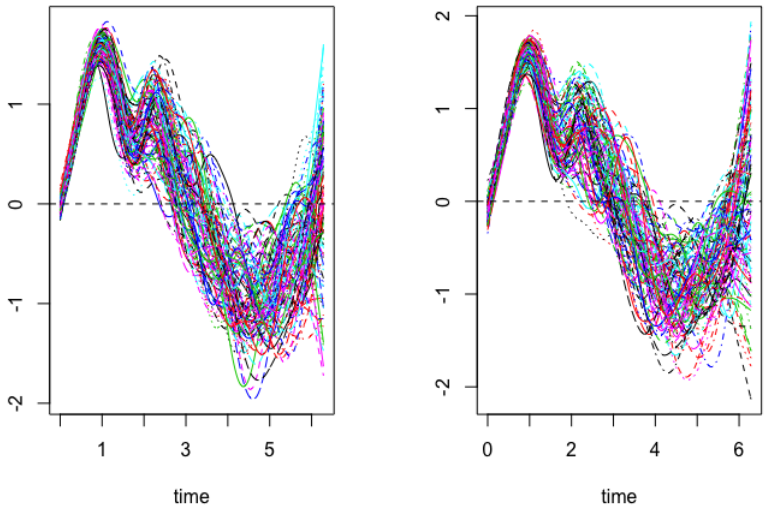


Figure 7.10: Example of downsampling the dense aperiodic data case A into 2 replicates. The two copies retains the structure of the original curves.

Proceeding with the newly simulated data we present the results of applying the FSC-DSC approach on the simulations of case B, case C, and case D. We will use FSC-S(D_o) to cluster the data based on its superior performance in Section 7.2.2. The simulations were run on 100

data sets, but first we will explain the results on one data set (as shown in Tables 7.2, 7.3, and 7.4). In these tables we only show the important outcomes (rows) of applying FSC-DSC over the range of $\sigma = \{0.1, ..., 3.0\}$. Considering Table 7.2, we notice that in case B the highest ARI is for $k = 2$. While in Table 7.3 for case C, the highest ARI is for $k = 3$, and finally in Table 7.4 for case D the highest ARI is for $k = 4$ and $k = 2$. The approach is showing a clear jump from very large k values to the true k value in each scenario. Further, the approach can detect both the super-clusters and the sub-clusters in case D at different ranges of σ. The above initial results give a good insight of the approach when applied to one set of the simulated functional data. However, to examine the robustness of the approach and the stability of the outcomes, we show the overall results in Figures 7.11, 7.12, and 7.13 for case B, case C, and case D respectively. The percentages that are displayed in the figures are results of 100 simulations (i.e 100 comparisons tables) over 30 σ values each, which gives a total of 3000 outcomes. However, we only consider the outcomes that satisfy the following: (1) show a match between k_{odd} and k_{even}, and (2) only if k is within the range $[2, 15]$. Let the outcomes that satisfy the two points be given as $K = \{k_1, ..., k_i, ..., k_l\}$, then we calculate the percentage of $K = k_i$ over the total outcomes and consider the associated ARI.

In case B (Figure 7.11) the majority of the matched outcomes found $k = 2$ to be the optimal number of clusters with ARI=1, and ARI=0.95. In case C (Figure 7.12) the majority of the outcomes found $k = 3$ with ARI ranges from 0.86 to 1, while there were a few results choosing $k = 2$ with low ARI and even fewer cases choosing $k = 4$. In case D (Figure 7.13) almost 67% of the cases detected the sub-clusters in the data and thus gave $k = 4$, while about 31% of the results detected the super-clusters and so gave $k = 2$. It should be mentioned that the results in case B looks better than the results in case C and case D, due to the fact that stability tends to increase for decreasing k, besides that there is more chance for FSC-DSC to cluster the 3-group data (case C) into 2 while it is meaningless to cluster the 2-group data (case B) into 3 clusters. Note that the percentages displayed in the figures must sum up to 100%, apart from missing error.

Case B			
σ	K (odd set)	K (even set)	ARI
0.1	88	88	1
0.2	88	88	1
0.3	2	2	1
⋮	⋮	⋮	⋮
0.9	2	2	1
1.0	1	2	0
1.1	1	1	1

Table 7.2: Some selected results of FSC-DSC from a random iteration of case B. The shaded area shows the highest ARI reflected from a match of the two K's, which gives $k = 2$.

Case C			
σ	K (odd set)	K (even set)	ARI
0.1	88	89	0.76
0.2	84	89	0.43
0.3	32	37	0.22
0.4	3	32	0.28
0.6	3	3	0.97
⋮	⋮	⋮	⋮
1.1	3	3	1
1.2	3	1	0
1.3	1	1	1

Table 7.3: Some selected results of FSC-DSC from a random iteration of case C. The shaded area shows the highest ARI reflected from a match of the two K's, which gives $k = 3$.

Case D			
σ	K (odd set)	K (even set)	ARI
0.1	88	89	0.76
0.2	59	63	0.012
0.3	4	4	1
⋮	⋮	⋮	⋮
1.4	4	4	1
1.5	2	4	0.43
1.6	2	2	1
⋮	⋮	⋮	⋮
2.1	2	2	1
2.2	1	1	1

Table 7.4: Some selected results of FSC-DSC from a random iteration of case D. The shaded area shows the highest ARI reflected from a match of the two K's, which gives $k = \{2,4\}$.

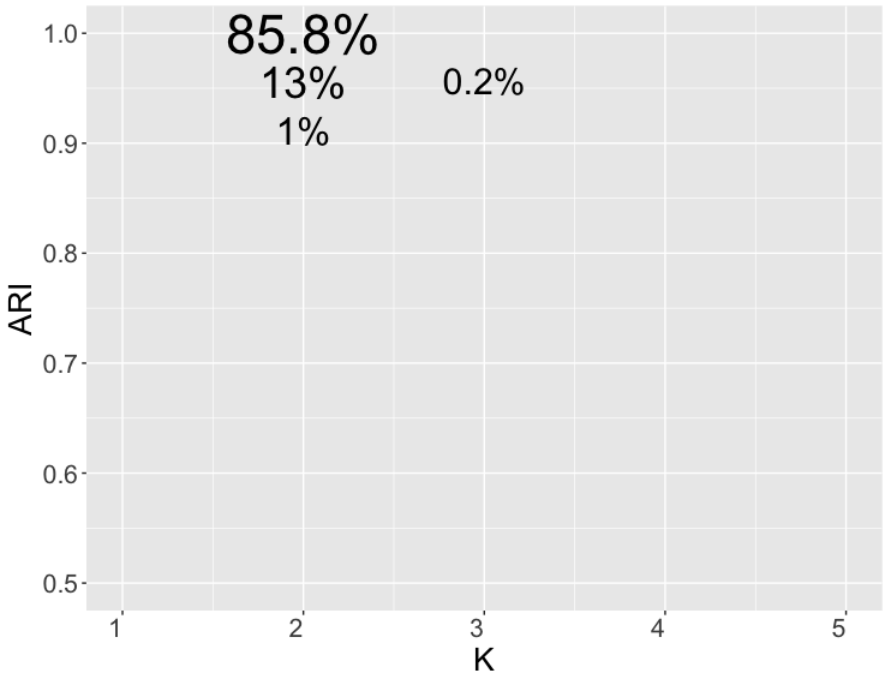


Figure 7.11: The graph displays percentages of choosing $K = k_i$ from the total outcomes along with the associated ARI based on the FSC-DSC. For case B, it is clear that $k = 2$ is favoured over the other k values with showing high ARI.

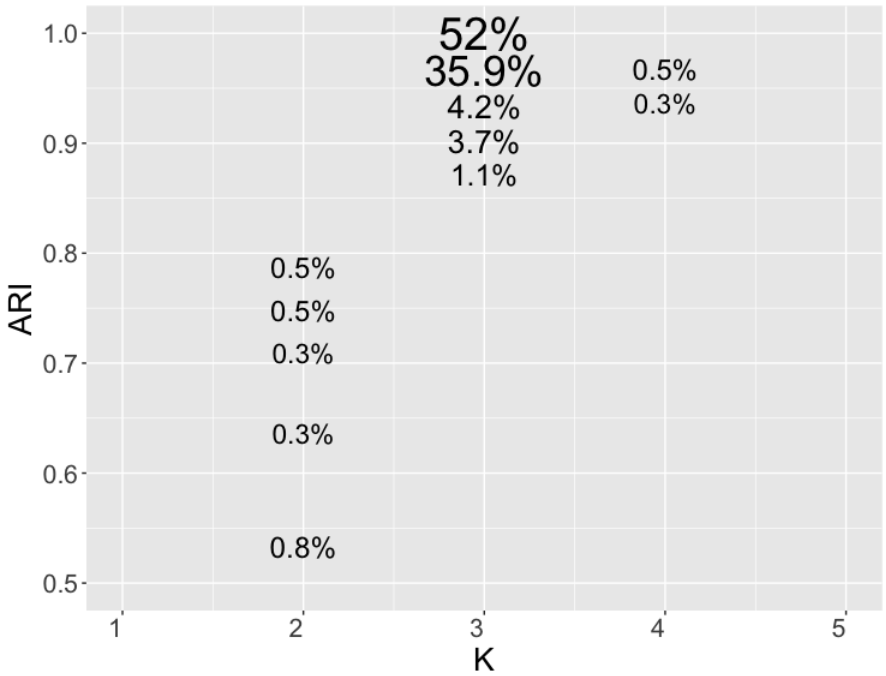


Figure 7.12: The graph displays percentages of choosing $K = k_i$ from the total outcomes along with the associated ARI based on the FSC-DSC. For case C, it is clear that $k = 3$ is favoured over the other k values with showing high ARI.

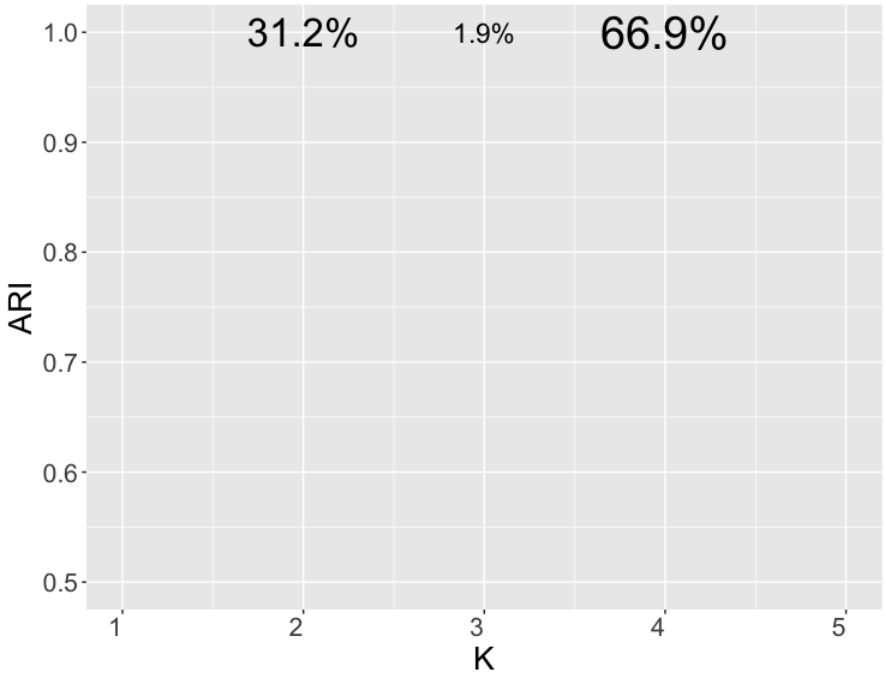


Figure 7.13: The graph displays percentages of choosing $K = k_i$ from the total outcomes along with the associated ARI based on the FSC-DSC. For case D, it is clear that $k = 4$ is favoured over the other k values with showing high ARI.

Despite the fact that it is a better practice to employ FSC-S(D_o) in the specific downsampling criterion since it showed better performance, in our experience using FSC-S(D_1) and FSC-S(D_2) can still lead to good estimate of k. For instance, both the derivative-based FSC-S approaches are able to find $k = 2$ in case B at smaller values of σ. In case C, it was not as straightforward as in case B, because the smoothing parameter λ that permits finding the optimal k is slightly different from the one used originally for smoothing the data. Likewise in case D, we had to use different values of λ to find the optimal k, yet both FSC-S(D_1), and FSC-S(D_2) could not detect $k = 4$ and only suggested $k = 2$. The procedure of changing λ in order to obtain the optimal k will complicate the algorithm. In fact, our proposed approaches are two-stage based approaches, which means that the smoothing stage is independent of the clustering stage. However, we attempted to show here how the smoothing is crucial in changing the clustering. The ranges of the smoothing parameter λ that can be used to smooth the data (i.e appropriate λ for smoothing) and at the same time lead to optimal k in each approach are summarized in Table 7.5. It is clear that across FSC-S(D_o) cases, the range of λ is almost the same, while we need finer intervals of λ at FSC-S(D_1) and FSC-S(D_2) to get the best clustering results. It should be mentioned that the values of σ that lead to optimal k are unlikely to be similar in these examples. As it was explained in Section 6.2, the parameter σ is obtained based on the domain of the data. Since the domain of the derivatives is different from the domain of the original trajectories, the values of σ will be different in each dataset.

Case	optimal k	FSC-S(D_o)	FSC-S(D_1)	FSC-S(D_2)
Case B	2	$10^{-2} \leq \lambda \leq 10^{2}$	$10^{-2} \leq \lambda \leq 10^{2}$	$10^{-2} \leq \lambda \leq 10^{2}$
Case C	3	$10^{-2} \leq \lambda \leq 10^{1}$	$10^{0} \leq \lambda \leq 10^{1}$	$10^{0} \leq \lambda \leq 10^{2}$
Case D	2	$10^{-2} \leq \lambda \leq 10^{1}$	$10^{1} \leq \lambda \leq 10^{2}$	$10^{-2} \leq \lambda \leq 10^{0}$
	4	$10^{-2} \leq \lambda \leq 10^{-1}$	-	-

Table 7.5: A summary of the smoothing parameter values that are appropriate for smoothing the data, also can support the FSC-DSC algorithm to detect the optimal k.

7.2.4 Application of General Downsampling Criterion

In this section we demonstrate the application of the proposed general downsampling criterion on the simulated data. Considering the modified simulated data as in Section 7.2.3 we will apply the approach on all the chosen clustering functional data methods. We will start with FSC-S(D_o) and will give more details of the application, as it showed better performance when compared to the other clustering techniques on this simulated functional data set.

The results of applying the general downsampling criterion using FSC-S(D_o) is displayed in Figures 7.14, 7.15, and 7.16 for case B, case C, case D respectively. The algorithm runs twice on each pair, one for the odd set and one for the even set. Thus, the algorithm will run 20 times for every simulated data set, since there are 10 pairs and 2 opposite replicates for each pair. Therefore for the 100 simulated data sets, there will be 2000 iterations at K that will build the boxplots. We examined the approach at $K = \{2,3,...,9\}$; selecting this smaller range of K to save computational time, since we know the true clusters.

Considering case B first, Figure 7.14 illustrates that the optimal $k = 2$ is achieved with a very high ARI (almost 1) compared to any other values of k. In case C (Figure 7.15) the highest boxplot corresponds to $k = 3$, which reflects the true clusters in the data. Whereas in case D (Figure 7.16) the algorithm leads to choosing 3 optimal numbers of clusters $k = 2,3$, and 4. As explained above, case D contains 2 super-clusters and 4 sub-clusters, thus we have $k = 2$ and $k = 4$. However, $k = 3$ might arise if curves generated from equation (7.5) and equation (7.6) are put in one cluster, and curves arising from equation (7.7) and equation (7.8) stay in their own distinct clusters. This is because the first two equations show no phase variations and span over the same period, while the other 2 equations each show a different phase variation and spans over a different period. Thus, when the algorithm examines $k = 3$, the FSC-S(D_o) approach will attempt to locate the 3 clusters by defining the 3 levels of variation in the data as we have explained. We notice that the general DSC results confirm the specific FSC-DSC results, except in case D where the general criteria results in a different clustering structure.

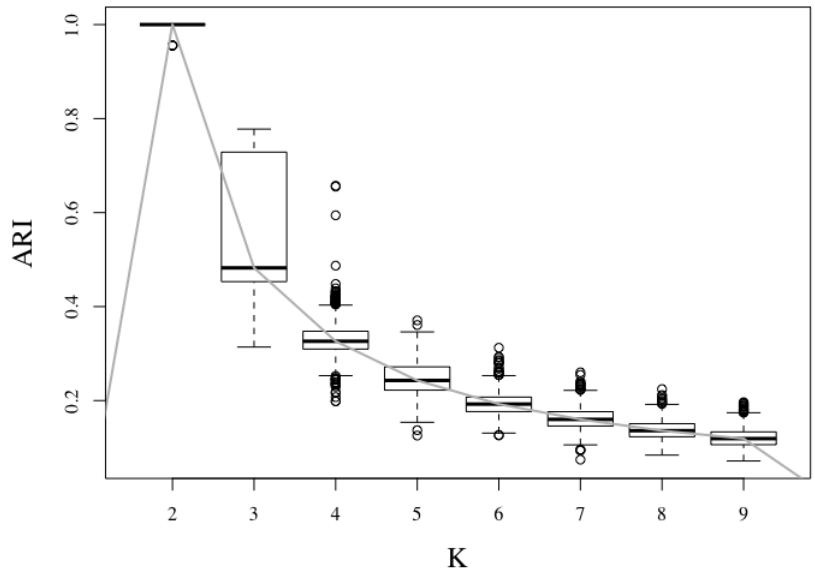


Figure 7.14: Boxplots of the ARI over k values when applying the general downsampling criteria with FSC-S(D_o) on case B. The approach suggests there are 2 clusters in the data.

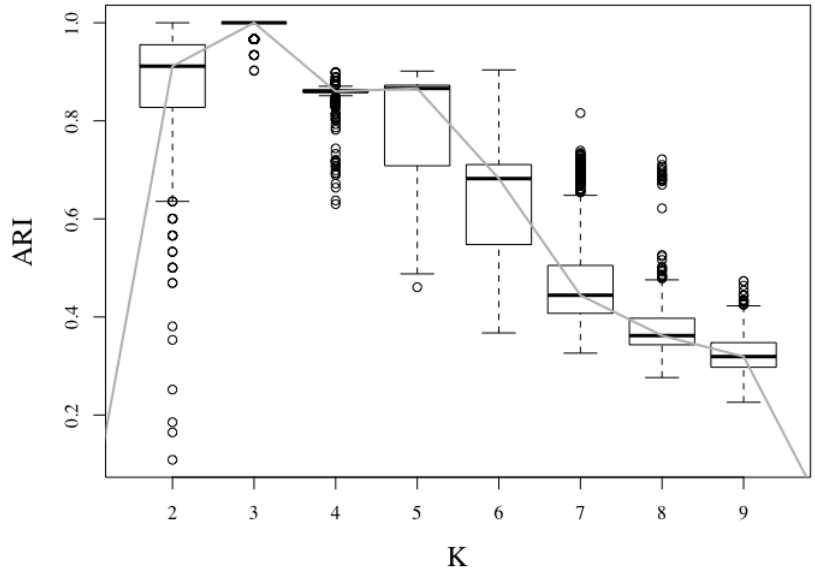


Figure 7.15: Boxplots of the ARI over k values when applying the general downsampling criteria with FSC-S(D_o) on case C. The approach suggests there are 3 clusters in the data.

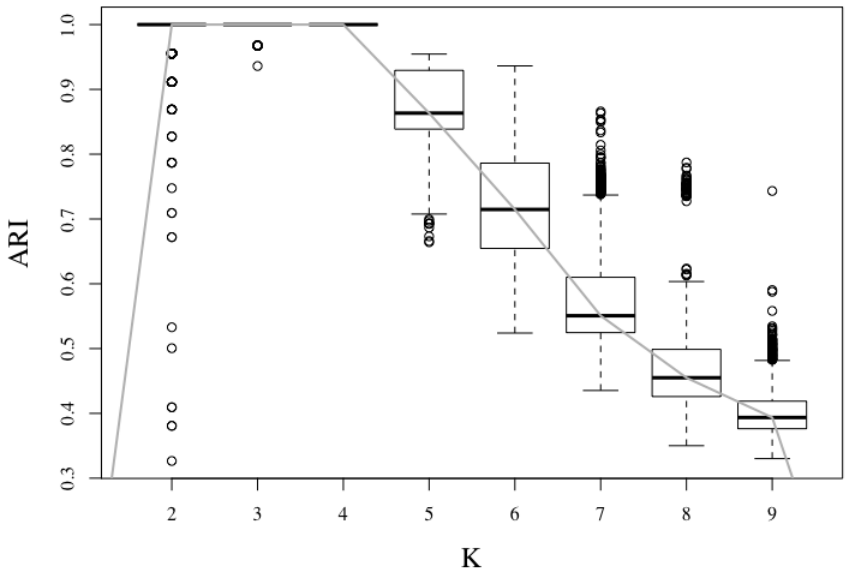


Figure 7.16: Boxplots of the ARI over k values when applying the general downsampling criteria with FSC-S(D_o) on case D. The approach suggests there are 2, 3, or 4 clusters in the data.

It was discussed in Chapter 5 that the general downsampling criterion was designed to work with any CFD method. Hence, we will now expand the use of the criterion for selecting the optimal k for the other CFD methods. The results are illustrated in Figure 7.17, 7.18, and 7.19 for case B, case C, and case D, respectively. The graphs summarize the mean ARI at each k based on applying the algorithm 200 times for each method, since there are 20 downsampled sets and the process is repeated 10 times. In case B, all the clustering methods can easily detect the 2 clusters without giving any other suggestions. In case C, FunHDDC, B-splines-Km, FPCA-mbc, and FSC-S(D_o) are all able to detect the true number of clusters $k = 3$, while FD-Kmeans gives $k = 4$ as well as $k = 3$. FSC-S(D_1) gives $k = 2$, while FSC-S(D_2) fails to identify any k value, as the ARI is very low. In case D, FSC-S(D_o) and Bsplines-Km propose $k = \{2,3,4\}$, but Bsplines-Km prefers $k = \{2,4\}$ over $k = 3$ based on the ARI. While FPCA-mbc only identifies the sub-clusters and gives $k = 4$, FunHDDC, and FSC-S(D_1) can only identify the super-clusters, thus preferring $k = 2$. FD-kmeans is different from the other approaches and suggests $k = \{2,3\}$. Finally, FSC-S(D_2) once again fails to detect any k value. It should be mentioned that both FSC-S(D_1) and FSC-S(D_2) will give better results if the smoothing parameter λ was adjusted to

support the clustering structure better (see Table 7.5).

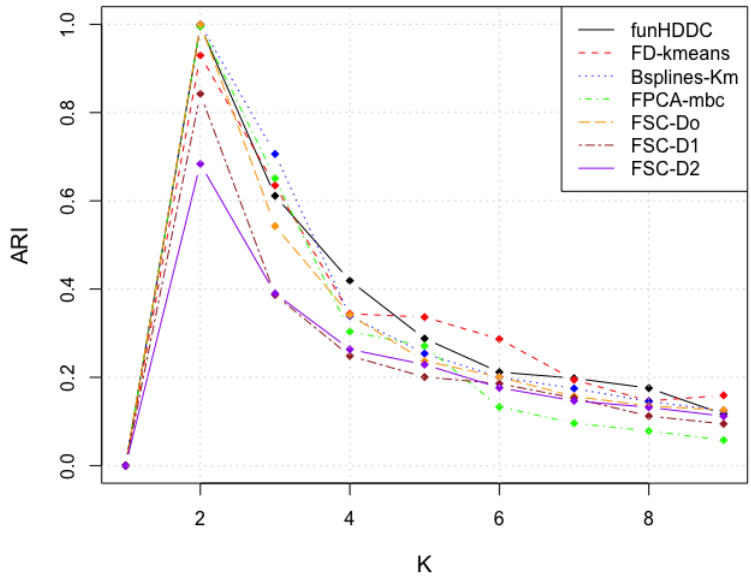


Figure 7.17: Results of the **mean** ARI for each K when using the general downsampling criteria with different clustering approaches on case B functional data.

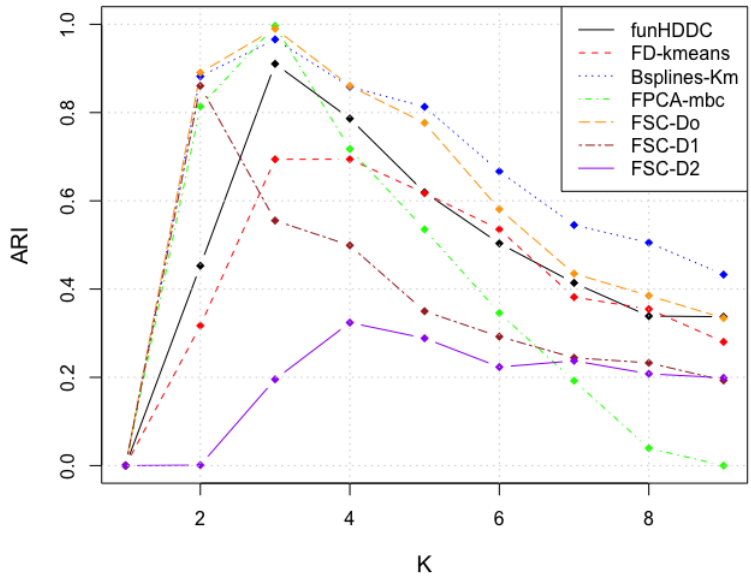


Figure 7.18: Results of the **mean** ARI for each K when using the general downsampling criteria with different clustering approaches on case C functional data.

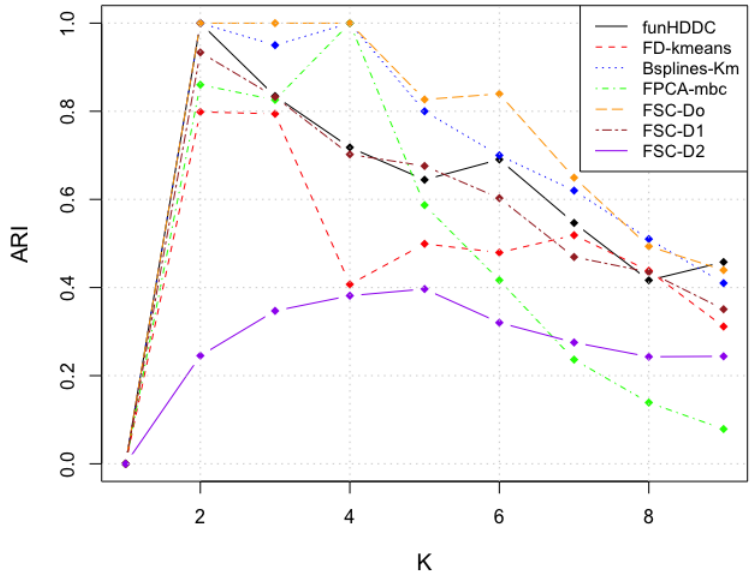


Figure 7.19: Results of the **mean** ARI for each K when using the general downsampling criteria with different clustering approaches on case D functional data.

7.3 The Canadian Weather Data

In this section, we set up a simulation study based on a real-world data set, to examine the performance of our proposed clustering techniques and the model selection techniques in functional dense data sets as well as sparse data sets.

The Canadian weather data consists of temperature and precipitation measures of 35 selected cities distributed across Canada. We introduced this data briefly in Section 2.2.1, as a data set that has been widely used by FDA researchers. However, in this section we will only consider the temperature data in its two forms; the dense daily measures (365 time points), and the sparse monthly measures (12 time points). Figure 7.20 shows the raw observations of the daily and monthly temperature data for the 35 Canadian cities. This data set was selected for setting up the simulations because it has been widely employed in FDA research, and it can be used as sparse or dense data. Also, Ramsay and Silverman (2005) have clustered the temperature data according to the geographical distribution of the Canadian cities as 4 groups (see Figure 4.1a and Section 4.1). We will assume this is the true clustering of the data when carrying out the simulation and comparisons between the CFD methods in Section 7.3.2.

Representing this data set as functional data can be done through smoothing and basis expansion. A good fit and commonly used smoothing model for the Canadian temperature data is given by B-splines of order 6, placing knots at the end of every month for the daily data and a knot at every quarter for the monthly data. Figure 7.21 shows the smoothed curves of the daily and monthly temperature data over a year. After fitting the smoothing model, we have estimated the error values (residuals) as shown in Figure 7.22. The error values will be used to set up the simulation scheme, which we also call 'the perturbation scheme'.

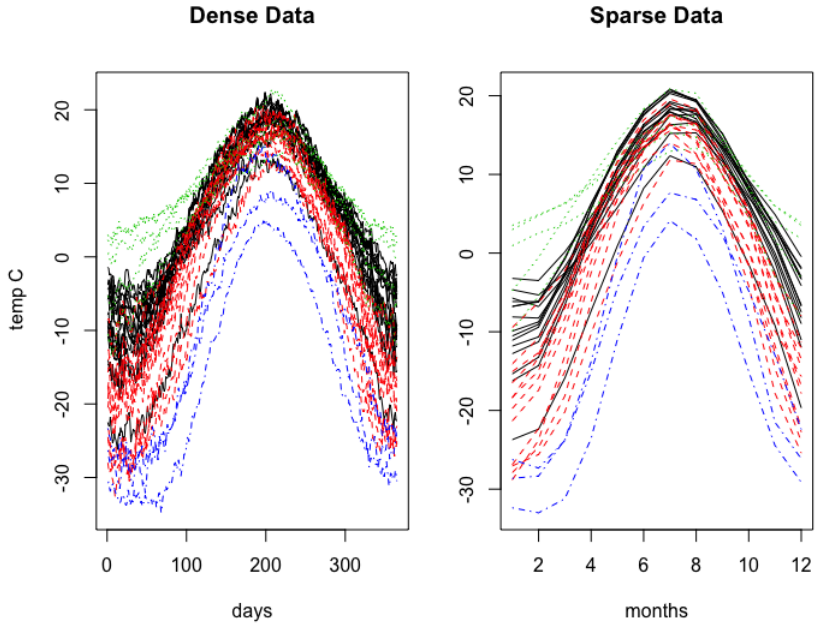


Figure 7.20: Raw data of the daily temperature (left), and the monthly temperature (right) for a year. Note the colours represent the 4 clusters according to the geographical distribution of the cities.

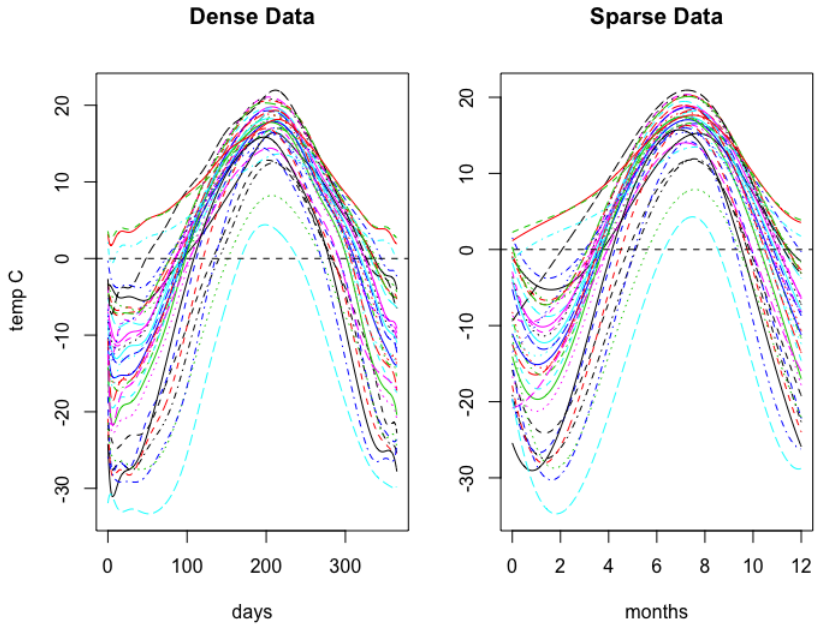


Figure 7.21: Smoothed curves of the daily temperature (left), and the monthly temperature (right) for a year.

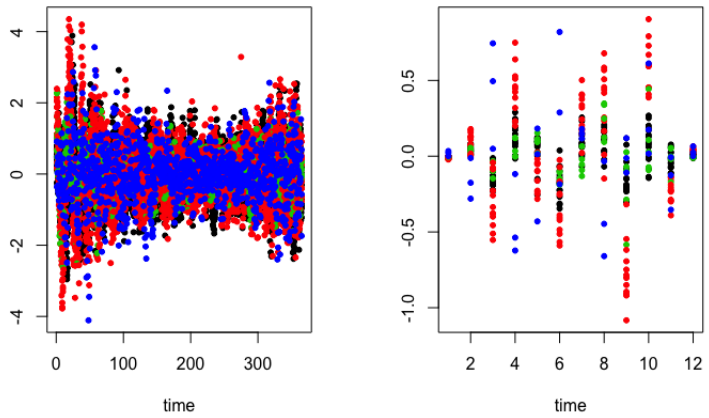


Figure 7.22: Estimated error for both the daily temperature data (left), and the monthly temperature data (right). Note the colours represent the 4 true clusters of the data.

7.3.1 Simulation Scheme

The simulation scheme is based on adding perturbation to the estimated error (residuals) of the functional data. The residual vector for each curve was evaluated by $\varepsilon_i = y_i - x(t_i)$. As there are 35 temperature curves, there is an associated error vector for each smoothed curve. Thus, there will be an error matrix of size 35×365 related to the dense data, and another error matrix of size 35×12 related to the sparse data. The perturbation to the error matrices can be divided to two main categories, where the first is nonparametric-based and the second is parametric-based. In the nonparametric-based perturbation, we mix and relocate the error randomly to perturb the original error vectors within each group. Whereas in the parametric-based perturbation, we first estimate the standard deviation $\hat{\xi}$ of the error in every group. Then for each group, we create new error values that are normally distributed with mean $(\mu = 0)$ and standard deviation $(\xi = \hat{\xi})$. To perturb the data more, we created a varied scale of the estimated standard deviations, so that we can move from simple scenarios to more complicated scenarios. The simulation scheme through adding perturbation to the error is summarized below in Table 7.6.

Scenario	New Error ε^*	iteration
1	randomly mixing the error and relocating them	100 times
2	creating error, $\varepsilon \sim N(\mu = 0, \xi^2 = \hat{\xi}^2)$	100 times
3	creating error, $\varepsilon \sim N(\mu = 0, \xi^2 = (2\hat{\xi})^2)$	100 times
4	creating error, $\varepsilon \sim N(\mu = 0, \xi^2 = (5\hat{\xi})^2)$	100 times
5	creating error, $\varepsilon \sim N(\mu = 0, \xi^2 = (10\hat{\xi})^2)$	100 times

Table 7.6: Simulation setup for creating perturbed sets of the original data set.

Considering the new error values (ε^*), we create new data sets y^* by adding the evaluated $x(t_i)$ from the basis expansion of the initial smoothing model to the simulated error values ε^*. As a consequence we will obtain new raw observations of the Canadian weather data. Through bootstrapping each scenario 100 times, there will be 500 data sets as dense data, and another 500 as sparse data $\{y_1^*, y_2^*, ..., y_{1000}^*\}$. Finally, the created data sets will be smoothed by a different smoothing choice from the initial one. The basis expansion will be B-splines again but with order 4, also the model will be saturated (i.e knots at every data point), while the smoothness will be controlled by the penalty term λ. Based on the GCV and by checking the effect of different λ values on the smoothed curves visually, we used $\lambda = 10^4$ for the dense case, while we used $\lambda = 10^{-2}$ for the sparse case. Note that we have used the same penalty term (smoothing parameter) for all the scenarios, mainly because we attempt to examine the performance of the clustering approaches on more noisy functional data while fixing the effect of the smoothing model. Figure 7.23 displays some perturbed data of scenarios 3, 4, and 5 after smoothing. It is clear that the varied scale of error created varied sets, while scenarios 1 and 2 look more similar to the original data.

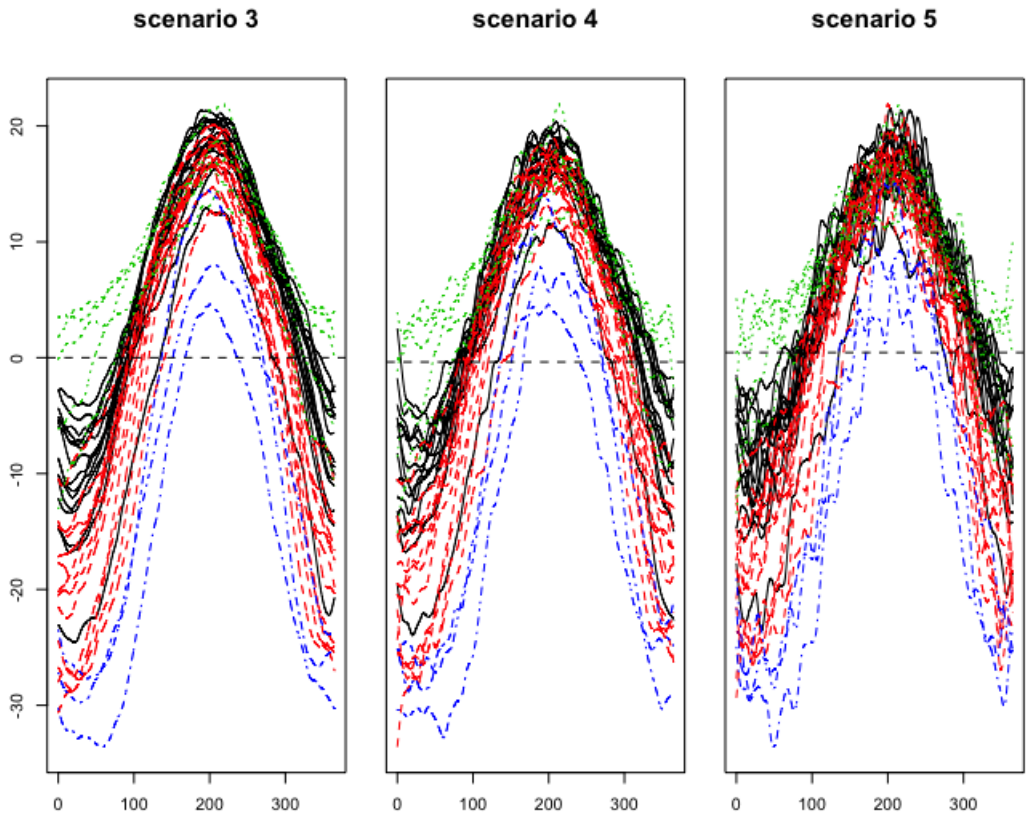


Figure 7.23: Examples of perturbed data for scenarios 3, 4, and 5. Note the colours represent the 4 true clusters of the data.

7.3.2 Application of FSC-S Approaches

In this section we present the results of clustering the simulated Canadian weather data using the FSC-S techniques and the chosen CFD approaches.

For all the clustering approaches, we fixed the smoothing model and we provided the number of clusters as $k = 4$. At each iteration we calculate CCR based on the true clusters, then we compute the mean and the standard deviation of the CCR for 100 simulated data of each scenario for the dense and the sparse case. The final results are shown in Table 7.7 below.

Mean of Correct Classification Rate (CCR)

Dense Data

Methods	Nonparametric	Parametric			
		$1\hat{\xi}$	$2\hat{\xi}$	$5\hat{\xi}$	$10\hat{\xi}$
FunHDDC	0.66 (0.074)	0.68 (0.087)	0.68 (0.076)	0.67 (0.075)	0.62 (0.102)
FD-Kmeans	0.65 (0.093)	0.65 (0.085)	0.66 (0.086)	0.63 (0.072)	0.63 (0.069)
B-splines-Km	0.55 (0.030)	0.55 (0.026)	0.53 (0.031)	0.55 (0.034)	0.55 (0.040)
FPCA-mbc	0.71 (0.018)	0.70 (0.076)	0.66 (0.103)	0.68 (0.048)	0.56 (0.088)
FSC-S(D_o)	0.54 (0.000)	0.54 (0.000)	0.54 (0.000)	0.55 (0.018)	0.58 (0.044)
FSC-S(D_1)	**0.82** (0.015)	**0.83** (0.000)	**0.83** (0.013)	**0.76** (0.072)	**0.67** (0.123)
FSC-S(D_2)	0.76 (0.075)	0.75 (0.061)	0.67 (0.082)	0.50 (0.057)	0.42 (0.058)

Sparse Data

Methods	Nonparametric	Parametric			
		$1\hat{\xi}$	$2\hat{\xi}$	$5\hat{\xi}$	$10\hat{\xi}$
FunHDDC	0.65 (0.087)	0.73 (0.076)	0.60 (0.091)	0.49 (0.088)	0.44 (0.072)
FD-Kmeans	0.72 (0.151)	0.64 (0.085)	0.64 (0.099)	0.44 (0.049)	0.46 (0.061)
B-splines-Km	0.60 (0.040)	0.59 (0.038)	0.58 (0.039)	0.56 (0.103)	0.51 (0.083)
FPCA-mbc	0.62 (0.100)	0.58 (0.079)	0.55 (0.095)	0.53 (0.075)	**0.55** (0.070)
FSC-S(D_o)	0.54 (0.010)	0.54 (0.023)	0.59 (0.047)	0.59 (0.051)	**0.55** (0.042)
FSC-S(D_1)	0.73 (0.031)	0.73 (0.047)	**0.74** (0.061)	**0.67** (0.100)	0.50 (0.075)
FSC-S(D_2)	**0.87** (0.057)	**0.86** (0.038)	0.70 (0.069)	0.48 (0.080)	0.41 (0.033)

Table 7.7: Mean CCR for the clustering methods when applied to the Canadian weather perturbed data sets. Note: Bold digits represent the best value within a column and values in brackets represent standard deviation of CCR.

Examining the performance of the functional clustering approaches on the dense data set, we noticed that FSC-S(D_1) shows relatively high performance in all scenarios. It should be noted that as the noise increases, the accuracy rates of FSC-S(D_1) decrease. The decrease in CCR with the increase of noise can also be observed for the other clustering methods. FSC-S(D_2) is also performing reasonably in general but the accuracy rates quickly drop to low values when the noise increases. Also, FPCA-mbc gave reasonable results and again its performance is affected by the level of noise in the data. In contrast, for all noise levels, FunHDDC, and FD-Kmeans gave lower accuracy rates than the approaches mentioned above. B-splines-Km, and FSC-S(D_o) performed poorly and uniformly in all scenarios. It should be mentioned that in the dense case, FunHDDC, FD-Kmeans, and FPCA-mbc could not converge in all iterations and created some missing accuracy values (less than 15%). In our calculations we accounted for the the missing value issue by creating more iterations for these clustering approaches, then we calculated the mean CCR out of 100 that give clustering results. We acknowledge this might leave out some poorly performing cases and give a higher CCR for those methods.

On the other hand, the performance of the clustering methods changed in the sparse data. In this case, FSC-S(D_2) gave high CCR in the simplest scenarios (1 and 2). Further, FSC-S(D_1) showed reasonable results in scenarios 3 and 4. However, in the highest level of perturbation (scenario 5), both FSC-S(D_1) and FSC-S(D_2) did not give good results, while FSC(D_o) constantly gave low accuracy rates in all scenarios. The accuracy rates of FPCA-mbc were lower in this case than in the dense case. FunHDDC, FD-Kmeans, and B-splines-Km in general performed better in the sparse case, yet they still gave low accuracy rates in the more noisy data.

To understand the behaviour of the functional spectral clustering approaches we illustrate their performance on one perturbed data set of scenario 2 as shown in Figure 7.24. On the left hand side of the figure there are the dense data, and on the right hand side there are the sparse data. While the top row displays the original curves clustered by FSC-S(D_o), the middle row displays the first derivatives clustered by FSC-S(D_1), and the last row displays the second

derivatives clustered by FSC-S(D_2).

According to the accuracy rates, FSC-S(D_1) gave the best results in the dense data case, which is shown in the second row, left panel of Figure 7.24, while, FSC-S(D_2) gave the best results in the sparse data case, which is shown in the third row, right panel of Figure 7.24. Visually the two examples show clear amplitude variations and some phase shifts. The FSC-S techniques can often detect these variations (if they exist), which in turn will support identifying the clusters. These two sub-figures demonstrate the clearest appearance of the 4 clusters among the other sub-figures. In the dense case, the first derivatives (the rate of change in temperature) hold more information about the data than the original trajectories and at the same time they are less disordered than the second derivatives (the acceleration in temperature), therefore FSC-S(D_1) achieves high accuracy rates. Similarly in the sparse case, the accelerations in monthly temperatures can reveal more information about the different clusters in the data. Hence, FSC-S(D_2) showed better performance than FSC-S(D_o) and FSC-S(D_1). Note that we use the graphs of the derivatives for illustration of how the derivative-based distance metric (refer to Section 4.2.2) can support the FSC-S algorithm to detect the groups more efficiently in some situations.

The clustering of the Canadian cities according to the best results of the dense case and the sparse case is shown in Figure 7.25. In general, the allocation of groups based on FSC-S(D_1) for the daily data and FSC-S(D_2) for the monthly data are similar to some extent and not very different from the geographical distribution of the cities. Overall, the northern cities appear in one cluster as they are colder than the other cities throughout the year, while the southern cities located on the Atlantic shore can be in one cluster. Likewise, the southern cities located on the Pacific shore can be in another cluster as they are warmer and more affected by the seasonal climate changes. Further, the south inland cities could have similar temperatures over the year thus will be in the same cluster.

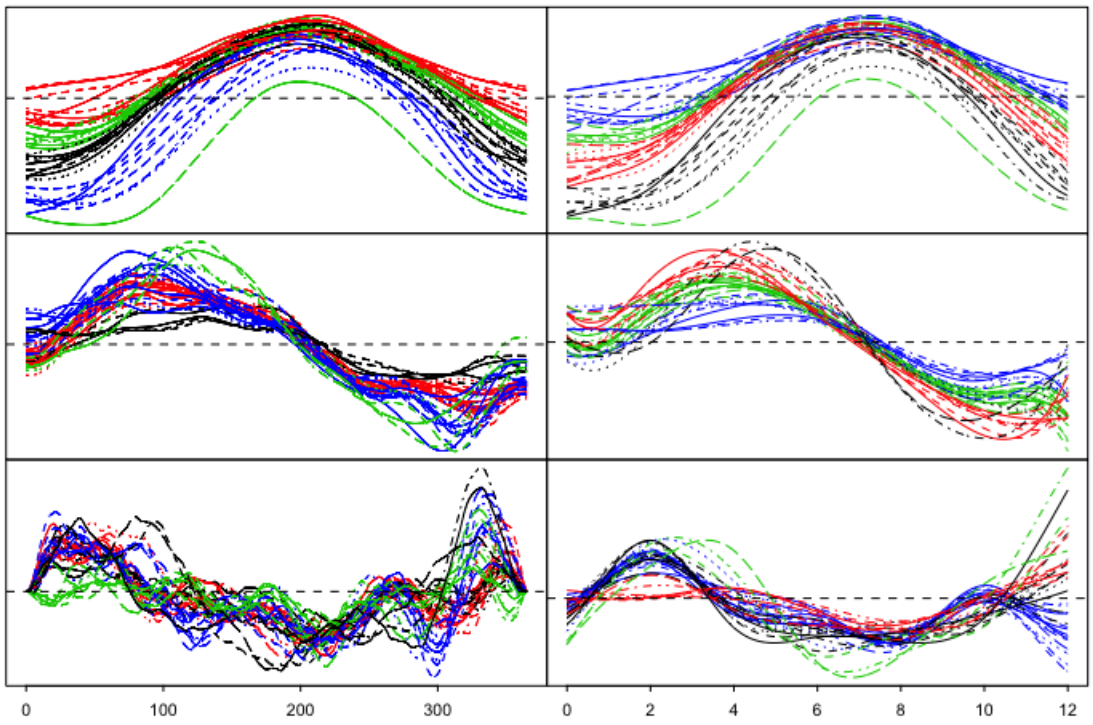
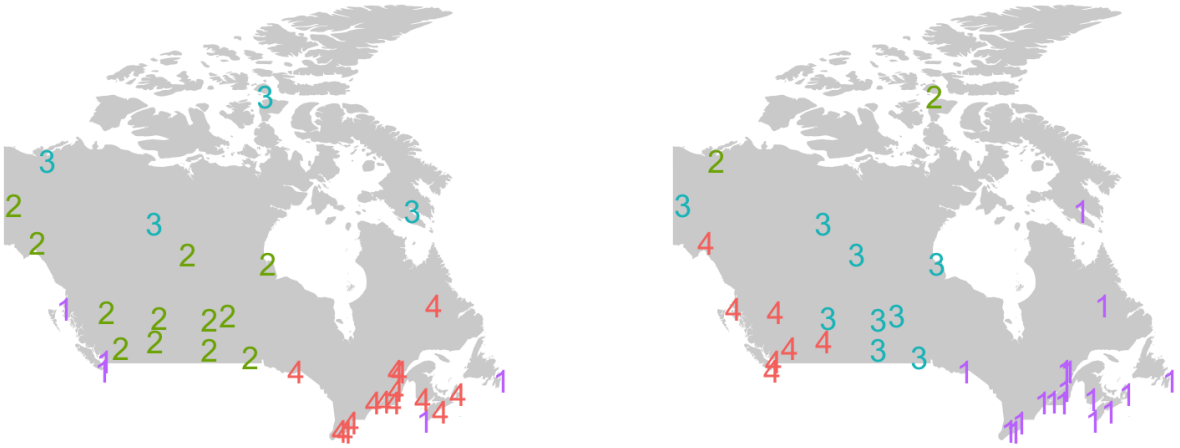


Figure 7.24: Resulted clusters using FSC-S(D_o), FSC-S(D_1), and FSC-S(D_2), are displayed in first row, second row and third row respectively, for both the daily data (left panel), and the monthly data (right panel). The above curves come from the second scenario of simulation when the error $\varepsilon \sim N(\mu = 0, \xi^2 = \hat{\xi}^2)$.



(a) (b)

Figure 7.25: The Canadian maps show results of clustering the cities according to the FSC-S approaches, where (a) displays the clusters according to FSC-S(D_1) when applied to the daily temperature curves and (b) displays the clusters according to FSC-S(D_2) when applied to the monthly temperature curves.

It is worth mentioning that the smoothing choice plays a critical role in clustering functional data, thus it is of importance to select the best smoothing technique for the data before moving to the clustering stage. The number and order of bases along with the smoothing parameter λ all define the smoothness of the data. Further, the selected smoothing model for the original curves will influence the smoothness of the derivatives.

Finally, we want to highlight an observation on the standard deviations of the CCR for FSC-S(D_o) and FSC-S(D_1). Note the 0 standard deviations in a few cases in the dense data in Table 7.7. The reason behind that can be explained by the behaviour of the functional spectral clustering approach, specifically the Laplacian matrix $\mathbb{L}$, which generates the first k eigenvectors. In every iteration, these k eigenvectors are very similar and they lead to the same clusters when applying the k-means as a last step. While the more perturbed errors introduce slight changes to the k eigenvectors, which in some iterations give different results than the common clustering results.

7.3.3 Application of Specific Downsampling Criterion

In this section we will examine the performance of the downsampling criteria on both the dense data and the sparse data. We will only consider the simulated data of scenario 2 when the errors are perturbed based on a normal distribution $\varepsilon \sim N(\mu = 0, \xi^2 = \hat{\xi}^2)$. Limiting the application to the least perturbed data (scenario 2) is to minimize the uncertainty of the results that will come from the more noisy data (scenarios 3, 4, and 5). Downsampling the dense curves will still maintain all the features of the original data, since there are 365 time points, while downsampling the sparse data can maintain most but not all the important features of the Canadian weather data. The downsampled curves will go through the same smoothing model as mentioned above but with reduced smoothing parameter λ. For instance, we set $\lambda = 10^2$ for the dense case and $\lambda = 10^{-3}$ for the sparse case.

Despite previously assuming that the geographical distribution of the Canadian cities de-

fine the true clusters of the data, we will not rely on this fact when examining the performance of FSC-DSC on the Canadian weather data. To discuss the performance of the algorithm, we will first present one sample out of the 100 total results for each FSC-S technique on both the dense case and the sparse case. Starting with FSC-S(D_o), the algorithms detects two k values for the dense case as shown in Table 7.8. According to the results, k could be 8 or 3 clusters at $\sigma = [8, 20]$ showing high ARI. Likewise, Table 7.9 shows the results of the sparse case and again suggests $k = 3$ with high ARI at $\sigma = [1.6, 3.6]$. Although K decreases more gradually in the sparse case than in the dense case, the sparse case does not show any other match of K over the range of σ. Note that the values of σ in the dense case are bigger than the values of σ in the sparse case, this selection is based on the domain of the data.

σ	K (odd set)	K (even set)	ARI
1	34	34	1
⋮	⋮	⋮	⋮
4	33	34	0.67
5	33	24	0.25
6	17	24	0.42
7	8	17	0.44
8	8	8	0.99
9	8	8	0.99
10	8	8	1
11	3	3	1
⋮	⋮	⋮	⋮
19	3	3	1
20	3	3	0.89
21	1	1	1

Table 7.8: Some selected results of FSC-S(D_o) algorithm with downsampling criteria on the Canadian weather **dense** data. The shaded area shows the highest ARI reflected from a match of the two K's over the optimal σ values, which gives $k = \{3, 8\}$.

σ	K (odd set)	K (even set)	ARI
0.1	34	34	1
0.2	34	34	1
0.3	34	34	1
0.4	27	34	0.2
0.5	27	21	0.37
0.6	27	21	0.37
0.7	27	21	0.37
0.8	16	7	0.48
0.9	16	7	0.48
1.0	8	7	0.78
1.1	6	7	0.67
1.2	6	7	0.67
1.3	6	7	0.67
1.4	6	7	0.67
1.5	6	3	0.21
1.6	3	3	1
⋮	⋮	⋮	⋮
2.1	3	3	1
2.2	3	3	0.88
⋮	⋮	⋮	⋮
3.6	3	3	0.88
3.7	1	1	1

Table 7.9: Some selected results of FSC-S(D_o) algorithm with downsampling criteria on the Canadian weather **sparse** data. The shaded area shows the highest ARI reflected from a match of the two K's over the optimal σ values, which gives $k = 3$.

On the other hand, the results of applying the algorithm using FSC-S(D_1) are displayed in Table 7.10 and Table 7.11 for the dense and the sparse case, respectively. The results of the algorithm on the dense data suggests $k = 5$ for $\sigma = [0.13, 0.15]$, yet $k = 13$ is also a possible

option as it shows high ARI. Moving to the sparse case, the results in Table 7.11 are not as encouraging as the previous results. For instance, there exist more k values on each replicate that do not often match, while the match at $k = 5$ gave low ARI and occurs at a small range of σ, which means the clustering structure is not very stable at $k = 5$. Also the algorithm cannot detect any other clustering structure at different range of σ values.

Further, we attempted to apply the algorithm using FSC-S(D_2) to see if the second derivatives scale would be able to detect any clustering structure in the data. The results are displayed in Table 7.12 and Table 7.13 for the dense case and the sparse case, respectively. It is clear that for the dense case the algorithm moves instantaneously from $k \simeq n$ to $k = 1$ even at small values of σ. The even set occasionally can detect $k = 3$ but only at one value of σ, which reflects no clustering stability, probably due to the high noise in the second derivatives. Whereas, in the sparse case, there appear a few k values in each set that did not match, which suggests that each replicate consists of a different clustering structure.

σ	K (odd set)	K (even set)	ARI
0.01	34	34	1
⋮	⋮	⋮	⋮
0.08	30	34	0.22
0.09	13	13	0.91
0.10	13	13	0.91
0.11	13	9	0.73
0.12	13	5	0.44
0.13	5	5	1
0.14	5	5	1
0.15	5	5	1
0.16	1	1	1

Table 7.10: Some selected results of FSC-S(D_1) algorithm with the specific downsampling criteria on the Canadian weather **dense** data. The shaded area shows the highest ARI reflected from a match of the two K's over the optimal σ values, which gives $k = 5$.

σ	K (odd set)	K (even set)	ARI
0.01	34	34	1
0.02	31	24	0.08
0.03	31	24	0.08
0.04	31	24	0.08
0.05	23	24	0.03
0.06	23	24	0.03
0.07	23	24	0.03
0.08	23	5	0.04
0.09	9	5	0.09
⋮	⋮	⋮	⋮
0.17	9	5	0.08
0.18	5	5	0.14
0.19	5	5	0.19
0.20	5	5	0.19
0.21	5	3	0.10
⋮	⋮	⋮	⋮
0.28	5	3	0.08
0.29	4	3	0.14
0.30	4	1	0
⋮	⋮	⋮	⋮
0.35	4	1	0
0.36	2	1	0
⋮	⋮	⋮	⋮
0.51	2	1	0
0.52	1	1	1

Table 7.11: Some selected results of FSC-S(D_1) algorithm with the specific downsampling criteria on the Canadian weather **sparse** data. The shaded area shows the highest ARI reflected from a match of the two K's over the optimal σ values, which gives $k = 5$.

σ	K (odd set)	K (even set)	ARI
0.01	34	34	1
0.02	34	34	1
0.03	1	3	0
0.04	1	1	1

Table 7.12: Some selected results of FSC-S(D_2) algorithm with the specific downsampling criteria on the Canadian weather **dense** data. The table does not suggest any k.

σ	K (odd set)	K (even set)	ARI
0.01	30	31	0.26
0.02	30	31	0.26
0.03	22	21	0.25
0.04	22	6	0.19
0.05	10	6	0.38
⋮	⋮	⋮	⋮
0.08	10	6	0.34
0.09	3	6	0.31
⋮	⋮	⋮	⋮
0.13	3	6	0.31
0.14	1	4	0
0.15	1	4	0
0.16	1	1	1

Table 7.13: Some selected results of FSC-S(D_2) algorithm with the specific downsampling criteria on the Canadian weather **sparse** data. The table does not suggest any k.

In order to confirm the selected k of the above results, we examine the approach on the 100 simulated data sets of scenario 2 over the associated optimal range of σ. This means there will be 100 comparison tables for each algorithm. Then, we only consider the results when there is a match between k_{odd} and k_{even} for $2 \leq k \leq 15$ (as discussed in Section 7.2.3).

Figures 7.26, 7.27, 7.28, and 7.29 summarize the possible k values and their associated ARI based on the percentage of matches. For instance, the percentage of $k = 3$ with ARI=1 is 69% in the dense case and 73% in the sparse case when using FSC-S(D_o), see Figure 7.26 and Figure 7.27. In addition, using FSC-S(D_1) showed that 41% of the results gave $k = 5$ with ARI=1 in the dense case (Figure 7.28). Similarly in the sparse case, 52% of the results gave $k = 5$ but with very low ARI (Figure 7.29). Also, note that there are other k values with high ARI in Figure 7.26 and Figure 7.28. For instance, FSC-S(D_o) can also detect $k = 8$ with high ARI, but this occurs only 16% of the total matches in the algorithm. Besides, FSC-S(D_1) can detect $k = 13$ with high ARI, but their percentage of occurring is much lower than the optimal k. In addition, Figure 7.29 displays more k values scattered over the range $[2, 15]$, but they all show low ARI and occur less often than $k = 5$. Note that there is no rule of identifying whether ARI is high or low, however, in our study we set the minimal acceptable ARI to be 0.6.

Finally, we should discuss the situations where FSC-DSC could not give results (i.e. suggest any k). The first was using FSC-S(D_1) and FSC-S(D_2) in the sparse case. This can be explained by the nature of the data and the derivative-based FSC-DSC approaches, and to clarify this point we will recall Figure 7.24. We have already mentioned that downsampling the original curves of the sparse data will still maintain most of the curves' structure. Therefore, if the specific downsampling algorithm is based on FSC-S(D_o) we expect to get reasonable results similar to the results of the dense case. However, this is not true when FSC-DSC is based on FSC-S(D_1) or FSC-S(D_2), because the lower resolution replicates of the original curves will lead to slightly different derivatives' formats of each replicate and that will be more obvious in the second derivatives. Therefore, the derivative-based distance metrics will be different in each replicate which in turn will give varied results, thus less or no match between the two K's. The second situation was using FSC-S(D_2) in the dense case. Due to the pattern of the second derivatives and the associated noise (Figure 7.24), the eigengap heuristic was not able to detect a clustering structure except $k = 1$.

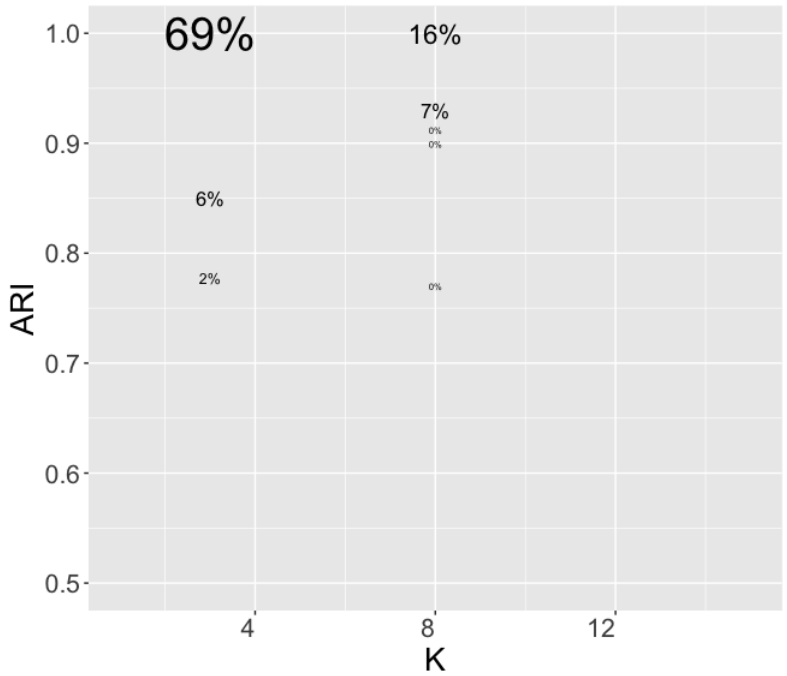


Figure 7.26: The graph displays percentages of choosing $K = k_i$ from the total outcomes along with the associated ARI based on the FSC-DSC in scenario 2 of the simulated **dense** data. Based on FSC-S(D_o), the chosen number of clusters is $k = 3$.

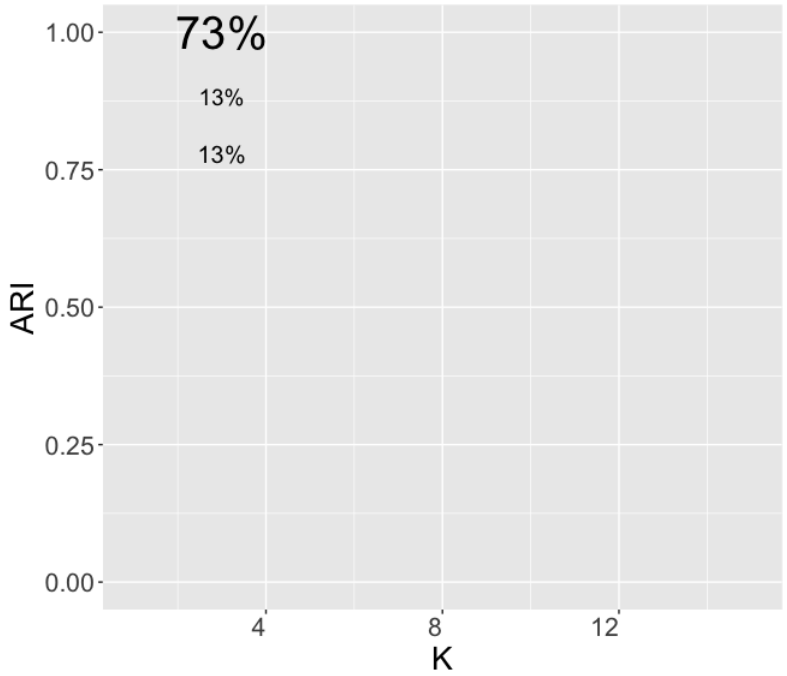


Figure 7.27: The graph displays percentages of choosing $K = k_i$ from the total outcomes along with the associated ARI based on the FSC-DSC in scenario 2 of the simulated **sparse** data. Based on FSC-S(D_o), the chosen number of clusters is $k = 3$.

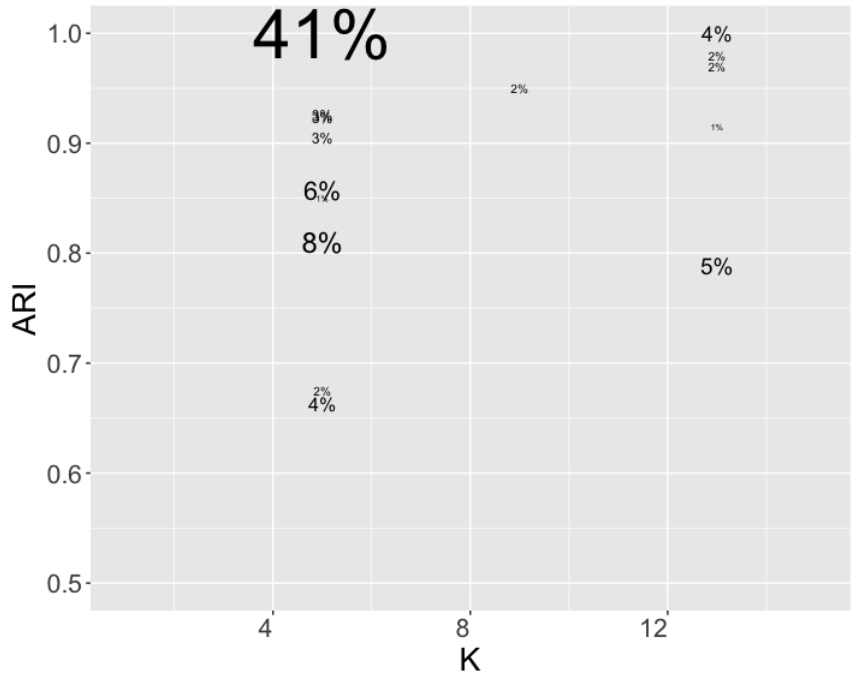


Figure 7.28: The graph displays percentages of choosing $K = k_i$ from the total outcomes along with the associated ARI based on the FSC-DSC in scenario 2 of the simulated **dense** data. Based on FSC-S(D_1), the chosen number of clusters is $k = 5$.

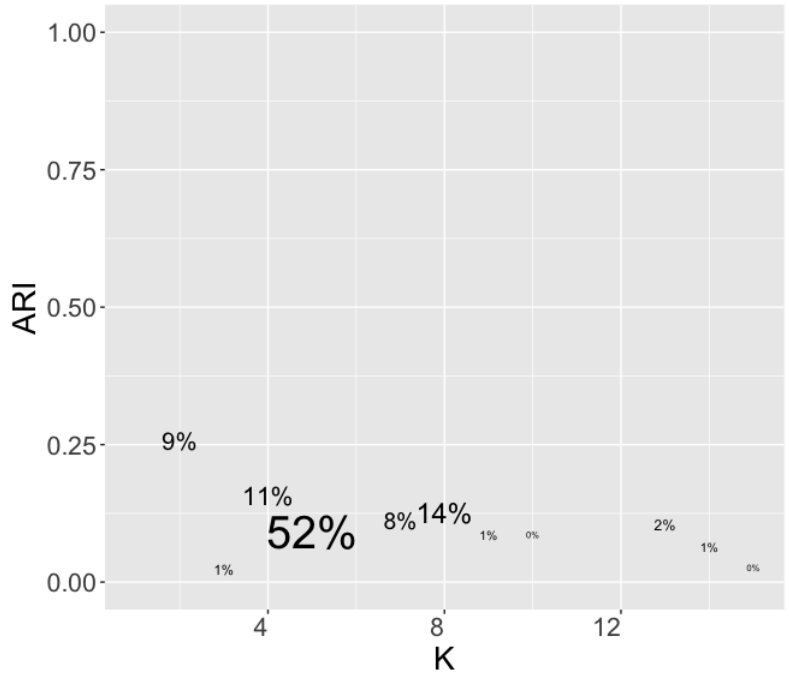
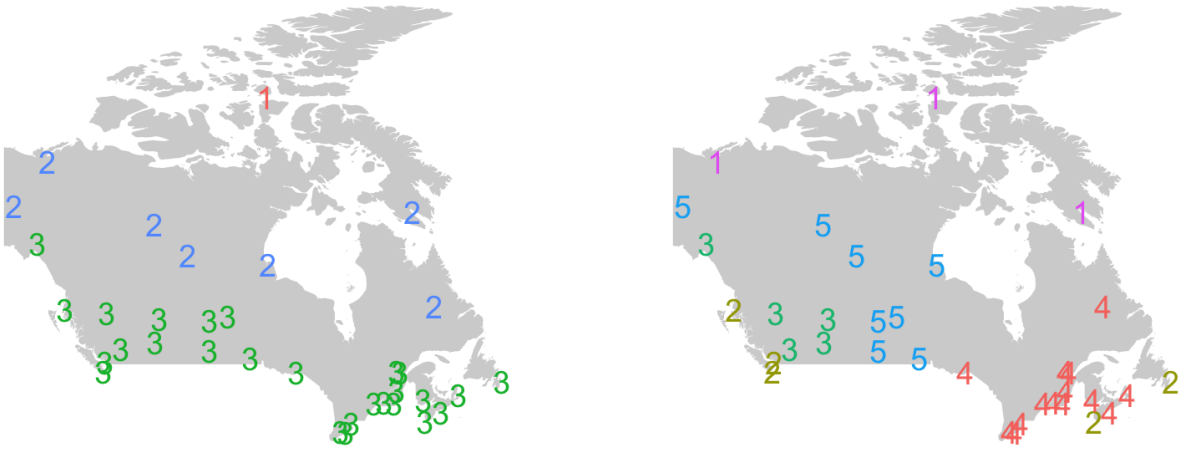


Figure 7.29: The graph displays percentages of choosing $K = k_i$ from the total outcomes along with the associated ARI based on the FSC-DSC in scenario 2 of the simulated **sparse** data. FSC-S(D_1) is unable to detect a specific k as the ARI values are very low.

According to the results, we can conclude that if we consider the original curves scale, which refers to clustering and locating the Canadian cities into temperatures categories, then there are 3 clusters in the data as displayed in Figure 7.30a. The figure shows the Canadian map with the cluster labels, the algorithm places Resolute city in its own cluster (cluster 1), while the other northern cities in another cluster (cluster 2), and all the southern cities in a cluster (cluster 3). This classification is reasonable, as Resolute is much colder than the other cities around the year, while the southern part is often warmer than the other parts of Canada. The clustered curves are also displayed in Figure 7.31 (left).

If we consider the first derivative scale, which refers to the rate of change in the temperatures over the year, then there are 5 clusters in the data as displayed in Figure 7.30b. The algorithm places the northern cities Resolute, Inuvik, and Iqaluit in one cluster (cluster 1) as per the geographical distribution. Most of the south-eastern cities are tied up in a cluster (cluster 4), while it places the warmest coastal cities Vancouver, Victoria, Prince Rupert, St. Jones, and Yarmouth in one cluster (cluster 2). Most of the remaining cities are inland and are divided into two clusters (cluster 3 and cluster 5). This clustering looks reasonable when considering the cities that are most and least affected by the seasonal climate changes. Also, Figure 7.31 (right) shows the clustered first derivatives of the data, where the algorithm can detect the phase and amplitude variations in the curves.



(a) $k = 3$ (b) $k = 5$

Figure 7.30: The Canadian cities clustered according to the FSC-DSC using (a) FSC-S(D_o) and (b) FSC-S(D_1).

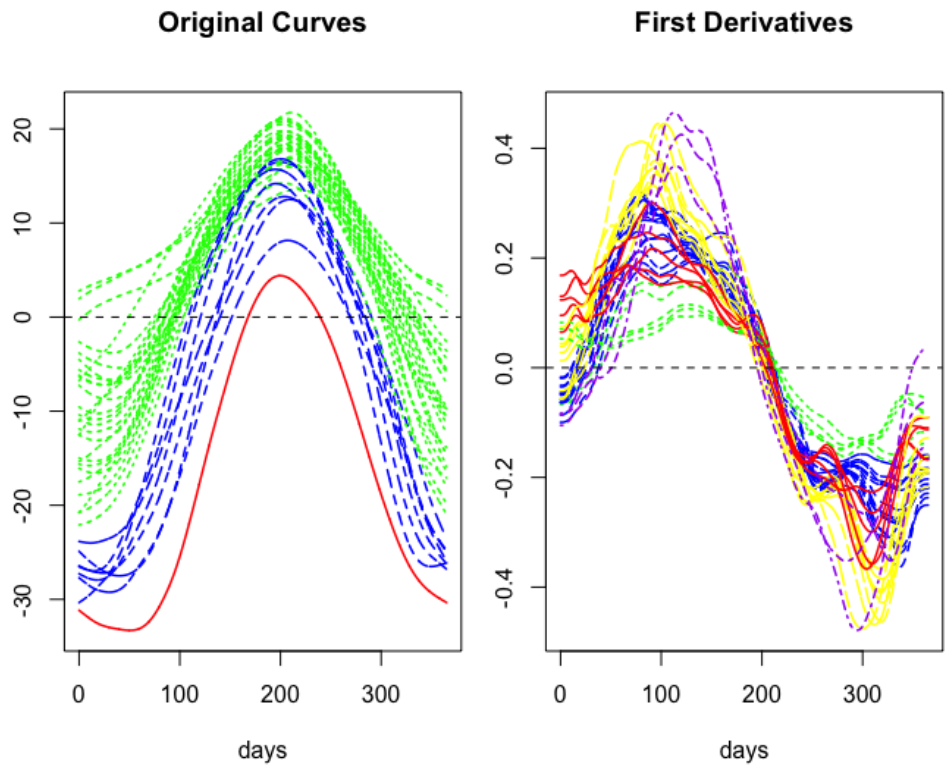


Figure 7.31: Clusters of the Canadian temperature curves using FSC-DSC with FSC-S(D_o) (left), that shows 3 clusters, and FSC-S(D_1) (right) that shows 5 clusters.

7.3.4 Application of General Downsampling Criterion

In this section we demonstrate the application of the proposed general downsampling criterion on the perturbed Canadian weather data of scenario 2. We will examine the performance of the general DSC at $K = \{2,3,...,15\}$ by compiling 100 simulated data each 20 times to build the ARI boxplots.

First, applying FSC-S(D_o) with general DSC on the dense and the sparse case gave the results as shown in Figure 7.32 and Figure 7.33 respectively. Consider the ARI in Figure 7.32, we notice there is no clear peak at specific k. Based on the simulation, the clustering results at $k = \{2,3,4,8\}$ are relatively more stable than at the other k values. On the other hand, the same algorithm applied to the sparse case results in first $k = 2$, then $k = 3$, based on the highest ARI (Figure 7.33).

Second, applying FSC-S(D_1) with general DSC on the dense and the sparse case gave the results as shown in Figure 7.34 and Figure 7.35 respectively. In the dense case, the highest boxplot is at $k = 4$, however we cannot rely on this answer as the median ARI for all k values are below 0.6 which suggests a low degree of match between the odd and the even replicates in most of the simulations. Furthermore, applying the same algorithm on the sparse case finds $k = 2$ with high ARI as it was the case in FSC-S(D_o).

Third, the application of FSC-S(D_2) on the dense and the sparse case are displayed in Figure 7.36 and Figure 7.37 respectively. The results demonstrate the poor performance of FSC-S(D_2) in detecting any k in both cases since the ARI values are very low.

The technique of the general DSC is based on the assumption that the data will achieve some clustering stability at the optimal k, which can be confirmed by obtaining a match of the odd/even replicates in every pair at that k. However, this is not the case in the daily Canadian weather data, where the clustering stability occurs at most k values when using FSC-S(D_o), while, this clustering stability is never achieved when using FSC-S(D_1). The artefact of not getting a unique k in these scenarios is related to the denseness and the smoothness of the data. For further explanation, we should refer again to Table 7.7 that displayed some 0 standard deviations in the dense case. We have explained the reason behind the 0 values in Section 7.3.2 by the phenomenon of the Laplacian matrix generating very similar k eigenvectors at each iteration. For the same reason FSC-S techniques repeatedly locate the same curves to the same cluster in every odd set/pair and in every even set/pair. Whether the odd set will always match the even set (as in FSC-S(D_o) application) or it will not (as in FSC-S(D_1) application), the generated results from $\{pair1, pair2, ..., pair10\}$ will be the same, and this is applied to all K. However, this issue did not occur in any example except in the dense case of the Canadian weather data. We assumed that the used smoothing model created very condensed functional data, and reducing the number of bases in the model by using a few knots instead of a saturated model might avoid this issue. However, we will not explore this option as it is beyond our designed smoothing model. Therefore, we will limit the application of this data on our FSC-S techniques and will not examine the other clustering techniques.

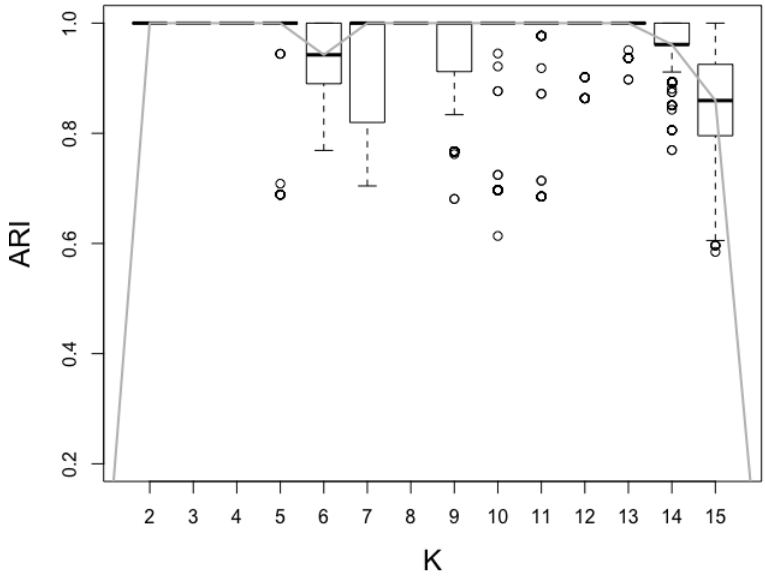


Figure 7.32: Boxplots of the ARI over k values when applying the general downsampling criteria with FSC-S(D_o) on scenario 2 simulations of the **dense** data. The approach cannot detect a unique k.

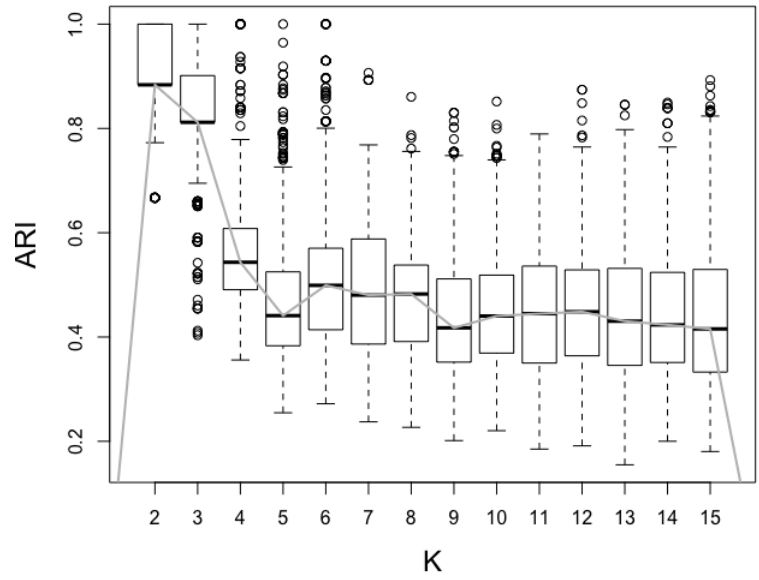


Figure 7.33: Boxplots of the ARI over k values when applying the general downsampling criteria with FSC-S(D_o) on scenario 2 simulations of the **sparse** data. The approach suggests there are 2 clusters in the data.

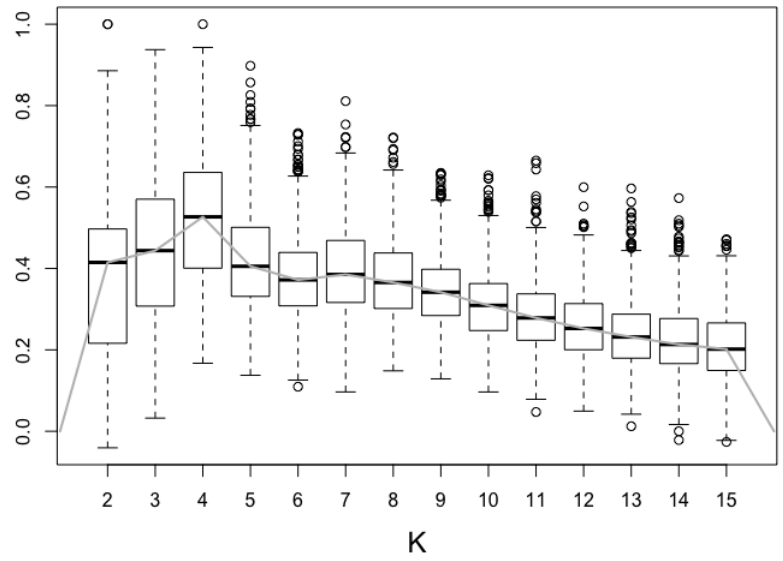


Figure 7.34: Boxplots of the ARI over k values when applying the general downsampling criteria with FSC-S(D_1) on scenario 2 simulations of the **dense** data. The approach suggests there are 4 clusters in the data with low ARI.

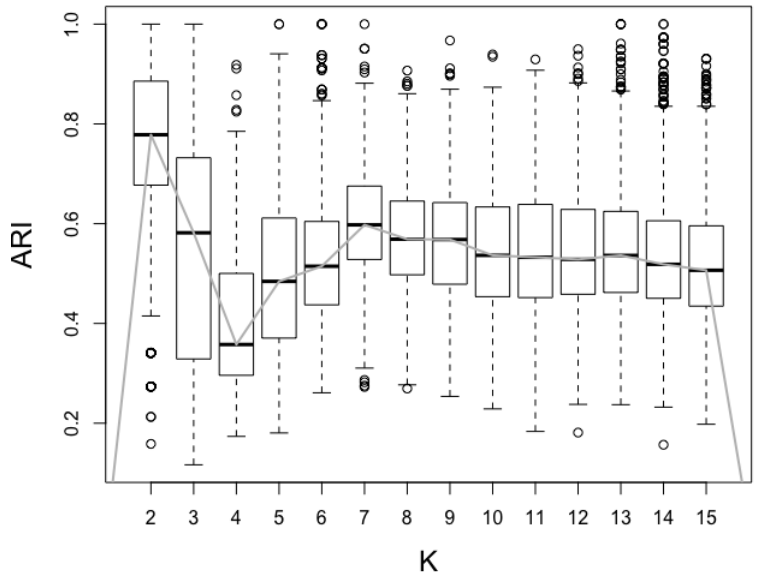


Figure 7.35: Boxplots of the ARI over k values when applying the general downsampling criteria with FSC-S(D_1) on scenario 2 simulations of the **sparse** data. The approach suggests there are 2 clusters in the data.

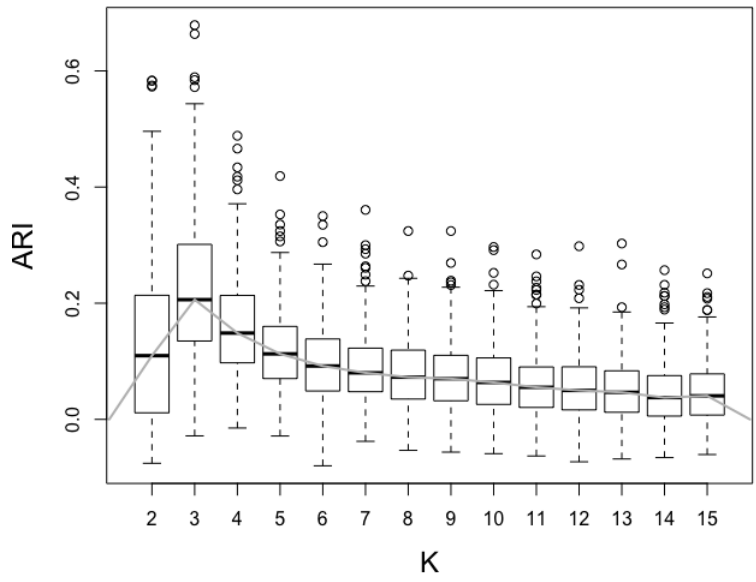


Figure 7.36: Boxplots of the ARI over k values when applying the general downsampling criteria with FSC-S(D_2) on scenario 2 simulations of the **dense** data. The approach cannot find any k.

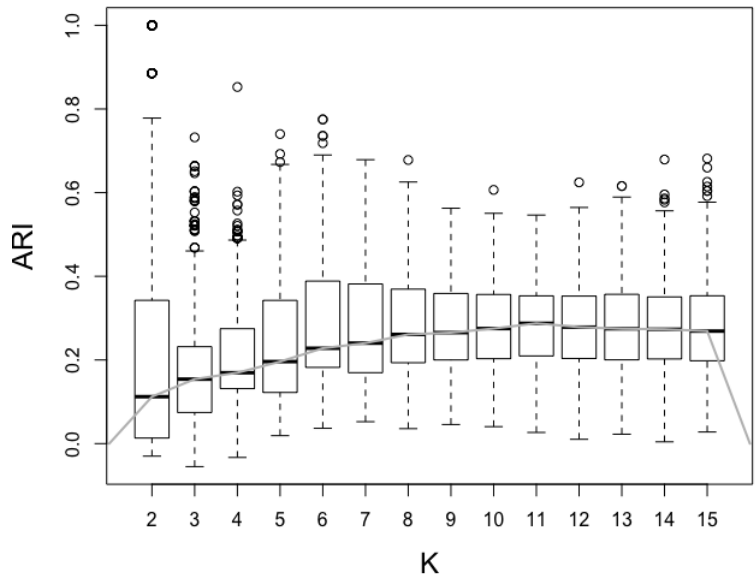
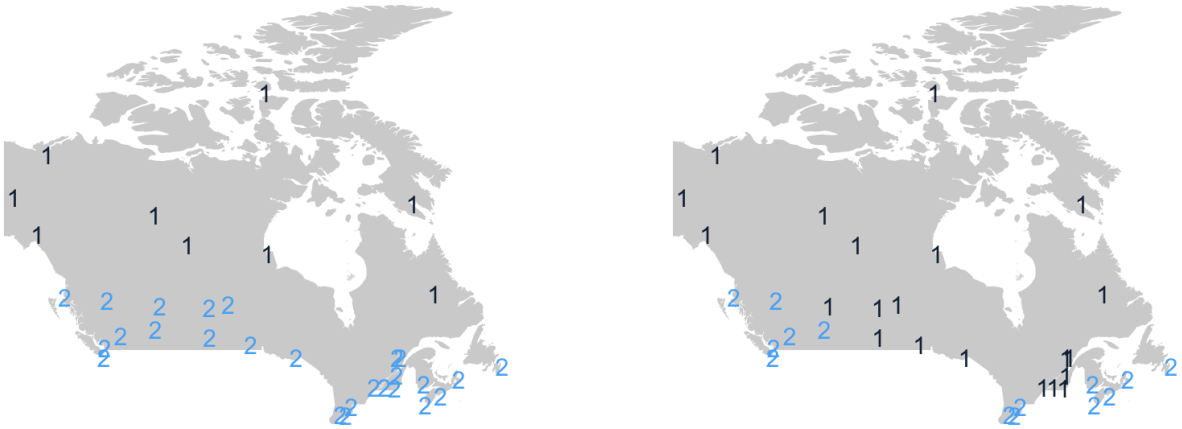


Figure 7.37: Boxplots of the ARI over k values when applying the general downsampling criteria with FSC-S(D_2) on scenario 2 simulations of the **sparse** data. The approach cannot find any k.

Considering the clustering results of the sparse case, FSC-S(D_o) simply divided the curves into 2 big clusters, one consisting of all the northern cities and the other consisting of all the southern cities in Canada (see Figures 7.38a and 7.39 (left)). FSC-S(D_1) however, is related to the rate of change in temperatures, therefore the allocation of the two clusters' members is different and based on the amplitude variations as shown in Figures 7.38b and 7.39 (right).



(a) FSC-S(D_o) (b) FSC-S(D_1)

Figure 7.38: The Canadian cities clustered according to the general DSC using (a) FSC-S(D_o) and (b) FSC-S(D_1) when applied to the sparse case.

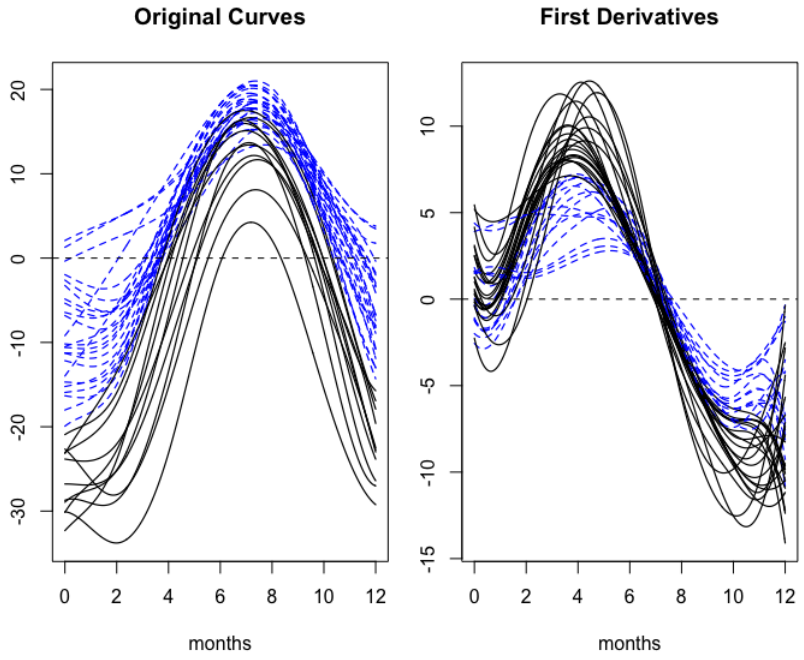


Figure 7.39: Clusters of the monthly Canadian temperature curves using the general DSC with FSC-S(D_o) (left), and FSC-S(D_1) (right).

7.4 Chapter Summary

In this chapter we have presented two simulation studies that cover different aspects of functional data formats. In each of the simulated data sets, we applied our proposed FSC-S techniques beside the other chosen CFD approaches. In addition, we investigated the performance of the model selection based criteria; the general DSC and the specific FSC-DSC.

According to the simulations, our proposed FSC-S techniques outperform the other clustering techniques in most scenarios. We noticed that the FSC-S techniques are very good in detecting phase and amplitude variations in the functional data. Whether these variations exist in the original trajectories scale or in the derivatives scale, FSC-S technique will cluster the data based on them. Several data sets can be made more informative by their first derivatives such as the daily Canadian weather data and the Berkeley growth data, therefore it is ideal to consider the first derivatives when clustering functional data. Thus, we prefer to plot the first and second derivatives beside the original curves to visually check for phase and amplitude variations. However, in case the derivatives display high noise and the variation is not clear, then it is better to avoid them because they will lead to poorer results. Based on our studies, we noticed that the second derivatives are not as informative as the first derivatives, and the only example where it showed better performance was the sparse case of the Canadian weather data. However, as a general procedure and to avoid uncertainty in the clustering results, FSC-S(D_o) and FSC-S(D_1) are favoured over FSC-S(D_2).

In addition, this chapter demonstrated the remarkable success of the specific downsampling approach in determining the appropriate number of clusters in functional data. Based on the simulations, FSC-DSC was able to detect the true number of clusters with a high success rate. Note that, it is better to avoid using FSC-S(D_2) within this approach and limit it to FSC-S(D_o) and FSC-S(D_1). However, deciding between FSC-S(D_o) and FSC-S(D_1) is a key point, and the ideal practice is to apply both. They will agree in some examples but most often they will provide different results. This is related to the nature of the data and the interpretation each technique

will provide. For instance, the expected clusters might be hidden in the first derivatives as in the Canadian weather data and the Berkeley growth data, yet using FSC-S(D_o) in these examples resulted in plausible and meaningful clusters.

On the other hand, the general downsampling criterion showed a fluctuating performance in identifying the appropriate number of clusters. In general, the approach attained very good results in the functional data with phase and amplitude variations using most of the chosen CFD methods. Whereas in the Canadian weather data, the approach showed poor performance in the dense case, and acceptable results in the sparse case.

We have noticed that the general DSC and the specific FSC-DSC cannot agree in all clustering problems, yet they often lead to sensible results. The general criterion depends on a fixed σ that is always much larger than the range of σ we explore in the specific criterion. As we have stated above, σ is a crucial parameter that can change the clustering structure of the data. However, we find that the specific FSC-DSC is more accurate than the general DSC. This may be because the specific criterion estimates k from the domain of the data that reflects the clustering structure based on the evaluated σ, while the general criterion requires providing a set of k values in order to examine each k, with keeping σ fixed at all K. In addition, the general DSC could be more sensitive to the noise and the sparseness of the data. We should implement more simulations to understand the reasons and to address the limitations of the general DSC in order to improve the approach (Chapter 9).

Chapter 8

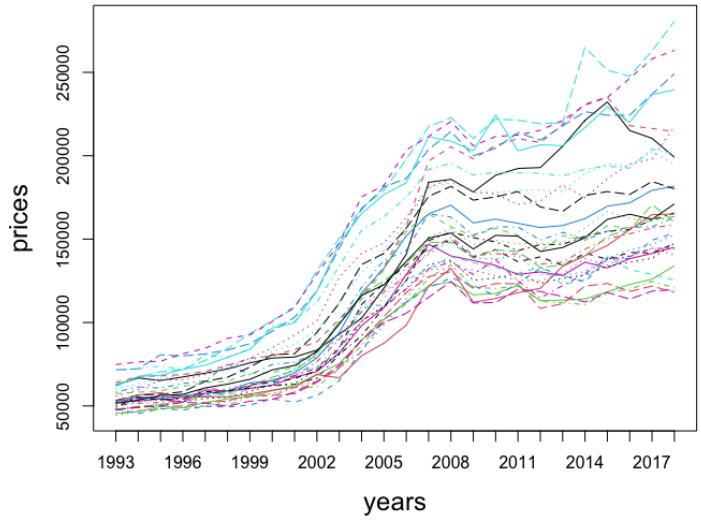
Application: House Prices in Scotland

In this chapter we will consider a real-life dataset to examine the performance of our proposed methods. The first section introduces the original dataset, then in Section 8.2 we convert it to a functional dataset and perform exploratory analysis to summarise the functional data. Section 8.3 illustrates the use of the model selection downsampling criteria (general DSC and specific FSC-DSC) on the data. The last section summarizes the outcomes of the cluster analysis.

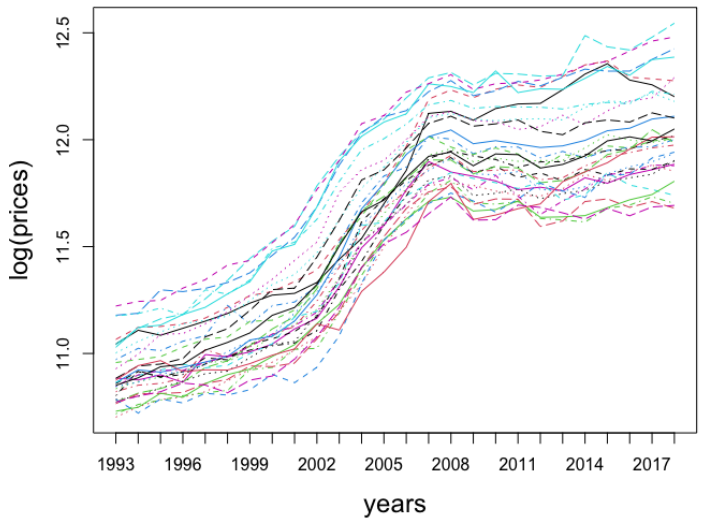
8.1 Data Description

The house prices data consists of the average value of house-sale transactions within each council area per year for 32 areas across Scotland from 1993 to 2018. The council areas are responsible for the provision of a range of public services, and they reflect the geographical, economical and population diversity in different parts of Scotland. The data were obtained from the Scottish Statistics website Scottish Government (2021). The data from each council area consists of 26 discrete points that represent the average sale price of houses every year in that area. Note that we have excluded one council area 'Na h-Eileanan Siar' as it contains some missing values. Figure 8.1a shows the average house prices (in GBP) for the council areas from 1993 to 2018. In general there is a clear increase in the prices over this time range. The prices from 1993 to 2002 were stable in all areas and ranged between £50,000 and £120,000. However a sharp increase in the prices started from 2003, which inflated the prices by more than double by the

end of 2008. After that, the council areas showed varied figures and the prices were clearly fluctuating in some areas. We have also demonstrated the data in a logarithmic scale as shown in Figure 8.1b. The logarithmic scale reveals the percentage change of the average house prices per year. Considering the time component and the temporal smoothness of the data, we can use FDA techniques to represent the data as functions in time. Note that we will refer to the data by AHP that stands for average house prices.



(a) Raw data



(b) Logarithmic scale

Figure 8.1: The average houses prices for the council areas in Scotland from 1993 to 2018 in (GBP) as raw data (a), and as price in logarithmic scale (b).

8.2 Smoothing Techniques

A first step in functional data analysis is to convert the observed data from discrete points to continuous functions through smoothing techniques. We will apply our general smoothing model that was used throughout the thesis, which is a B-splines of order 4 with a basis at every time point. Then the smoothing level is controlled by the smoothing parameter λ, which is chosen by the GCV. Based on the GCV (Figure 8.2) the dip of λ occurs within $[0.01, 1]$ and the minimum is 0.1. In addition, we examine the effect of using the other smoothing parameters on the estimated smoothed trajectories by visual inspection. We found that $\lambda = 0.1$ gives the best fit to the data. The resulting smoothed curves of the AHP data are displayed in Figure 8.3. We have also applied the same smoothing model on the logarithmic scale of AHP as displayed in Figure 8.4.

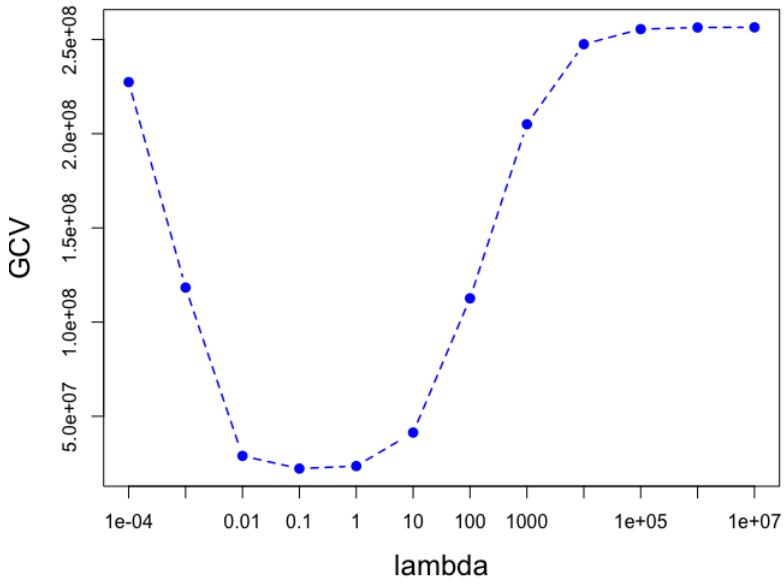


Figure 8.2: GCV curve shows the dip when $\lambda = 10^{-1}$ for the AHP data.

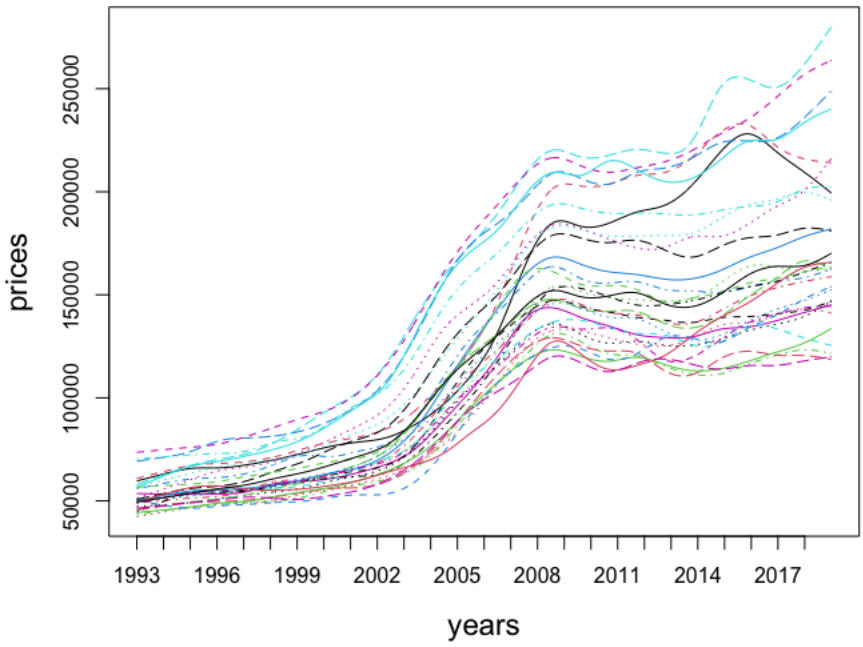


Figure 8.3: Smoothed curves of the AHP for the council areas in Scotland from 1993 to 2018.

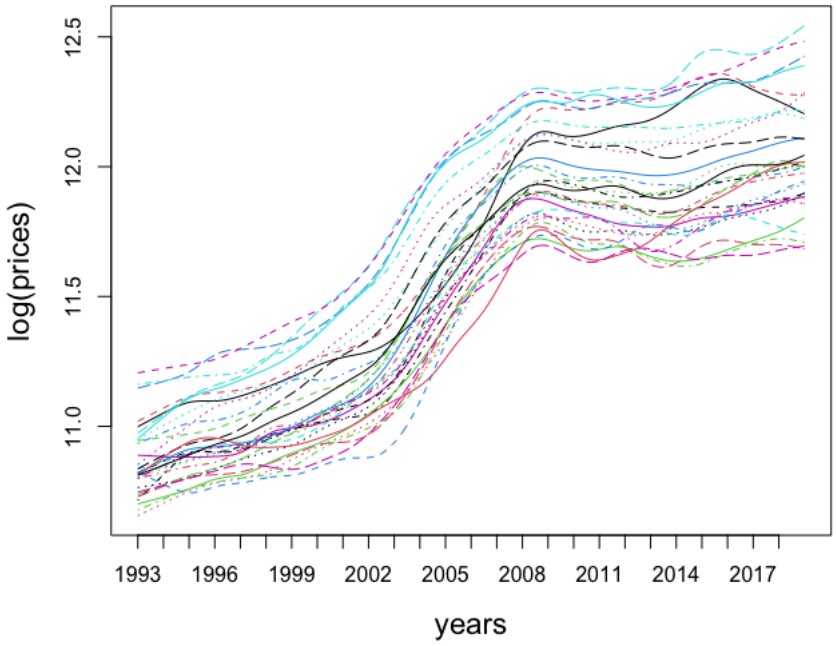


Figure 8.4: Smoothed curves of the log(AHP) for the council areas in Scotland from 1993 to 2018.

In this application we will focus more on the log(AHP), to deal with the logarithmic scale instead of the natural scale. Figure 8.5 displays some summary statistics of the logarithmic scale to give us some insight into the data. The mean and correlation functions calculate the mean and correlation of the data values at every pair of time points along the curves. In Figure 8.5 (left) the red curve represents the mean of the data, while Figure 8.5 (right) shows that there is high correlation between the years.

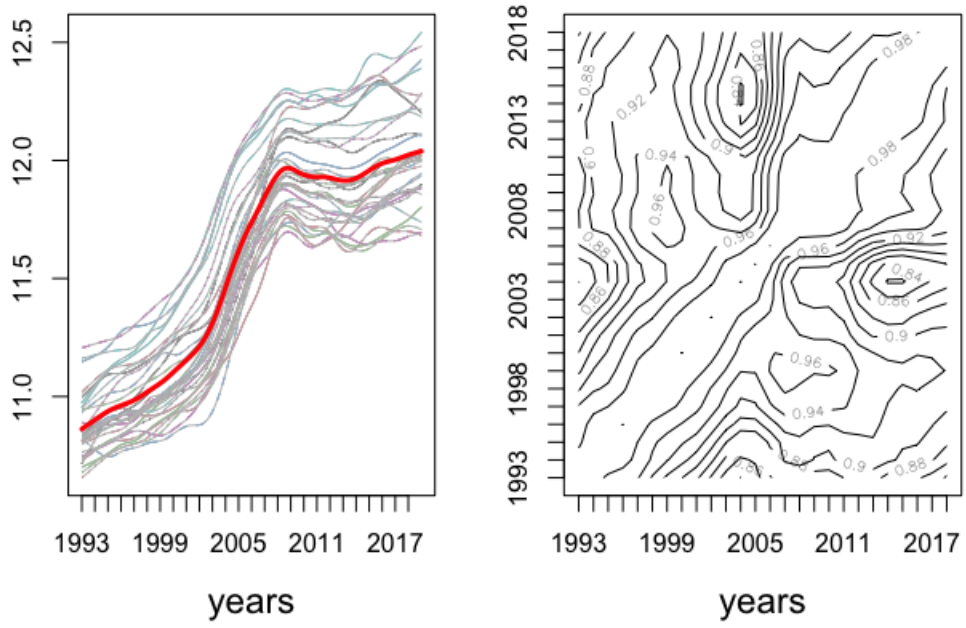


Figure 8.5: The mean function (red curve) of the log(AHP) data (left), and the contour plot of the correlation function of the same data (right).

Beyond the original smoothed data we inspect the first and second derivatives of the curves to check for any forms of amplitude and phase variation in the data. Figure 8.6 shows the first derivatives (left) and the second derivatives (right) of the log(AHP) data. The derivatives do not present any clear variation in phase and/or amplitude, note the derivatives of the original data show similar patterns in a different scale. Therefore, we will only rely on FSC-S(D_o) for clustering the data and we will not use FSC-S(D_1) or FSC-S(D_2), since their distance metrics

are based on the derivatives.

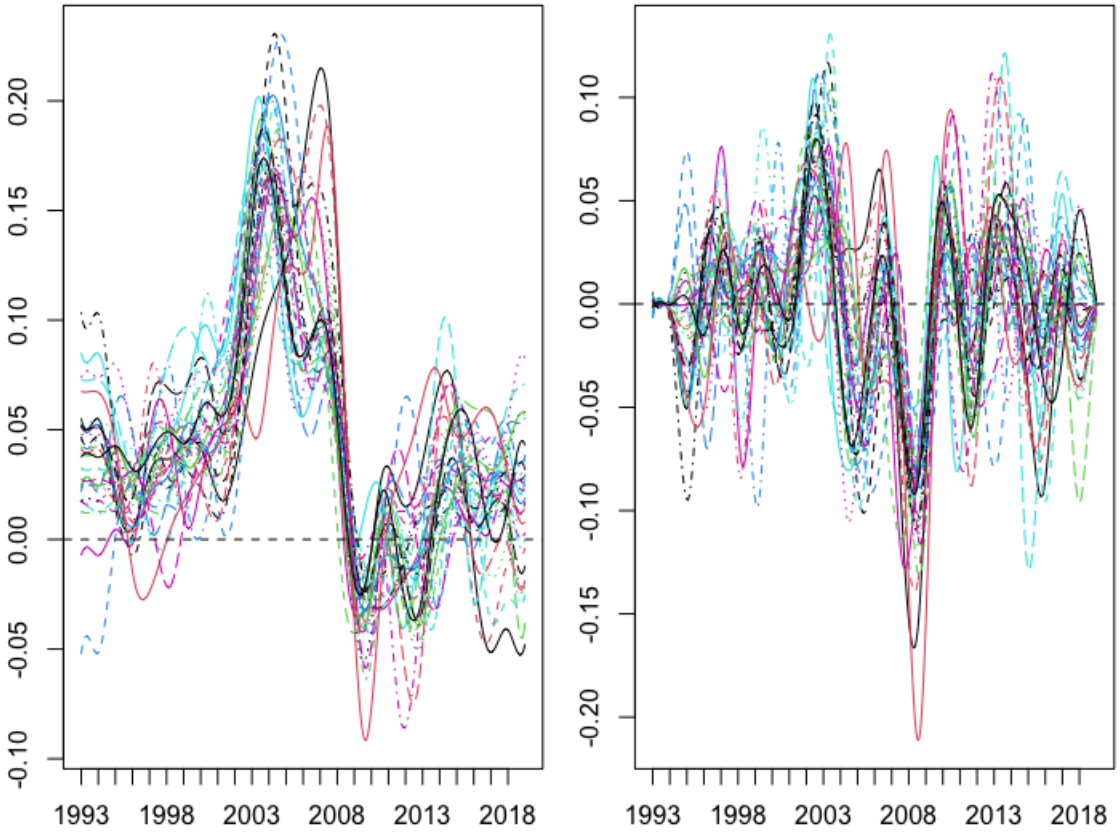


Figure 8.6: First derivatives (left), and second derivatives (right) of the log(AHP) data.

8.3 Data Clustering

In this section we apply our proposed downsampling approaches on the AHP data. The algorithm is based on splitting the functional data into two replicates as displayed in Figure 8.7. The odd replicate considers the values from 1993 and then takes every second value to 2017, while the even replicate considers the values from 1994 and then takes every second value to 2018, and thus they show some variation. Considering the downsampled data sets we will apply the specific FSC-DSC in Section 8.3.1 , then we will apply the general DSC in Section 8.3.2. The aim of these applications is to explore the clustering structure of the AHP dataset.

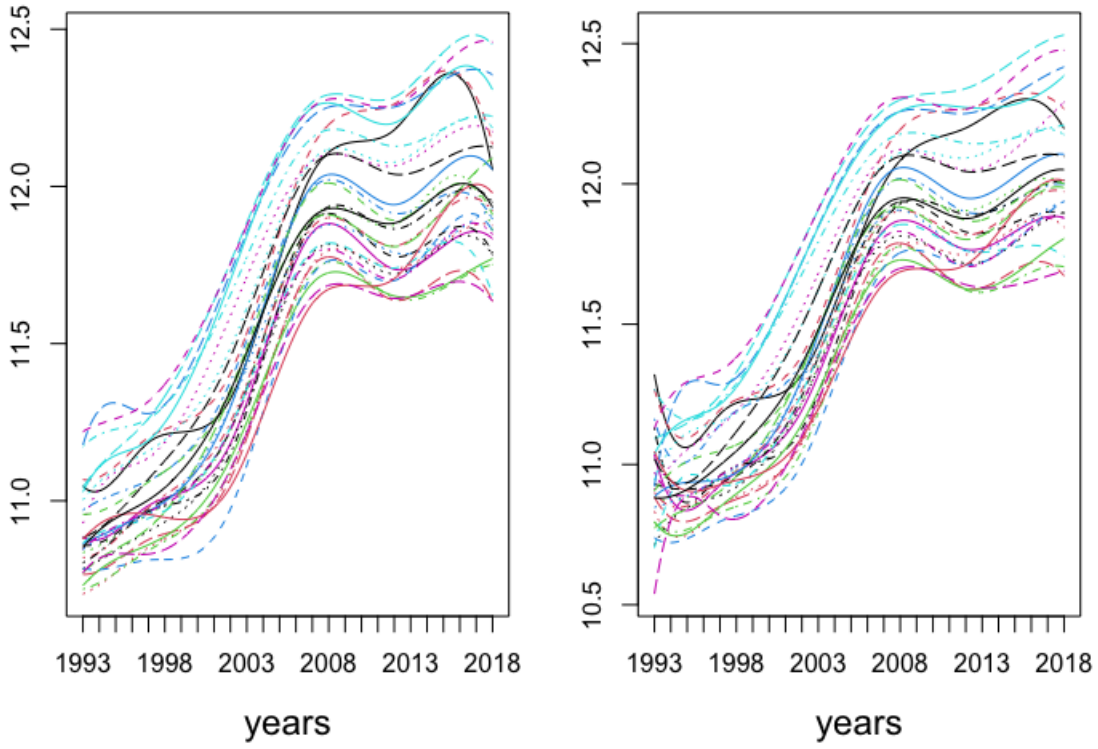


Figure 8.7: The downsampled log(AHP) data into 2 replicates odd (left) and even (right). Each replicate consists of 13 time points.

8.3.1 Specific Downsampling Criteria (FSC-DSC)

In this section we present the results of applying the specific downsampling criteria FSC-DSC to the AHP data. Although we have mentioned previously that we will focus on the logarithmic scale, we also show the results of the approach on the original data. Carrying out FSC-S(D_o) within the FSC-DSC approach on the log(AHP) data gave the results as displayed in Table 8.1. The results displayed two matches of K_{odd} and K_{even}, that are $k = 2$ and $k = 5$. Based on the highest ARI we would prefer 2 clusters over 5 clusters, yet $k = 5$ also gave a relatively high ARI.

To consider the results of the algorithm when applied to the original data of AHP, see Table 8.2. The only match of K_{odd} and K_{even} is at $k = 2$ with ARI=1, while the odd set gave some $k = 5$, the even set gave $k = 3$ at that range of σ. It is important to notice a large value of the

scale parameter σ is used in Table 8.2, due to the domain of the original data.

In addition, we attempted to cluster both the original data and the data on the logarithmic scale based on FSC-S(D_1) and FSC-S(D_2) within FSC-DSC. However, neither of these approaches was able to detect a clustering structure in the data.

σ	K (odd set)	K (even set)	ARI
0.01	29	30	0.67
0.02	29	30	0.67
0.03	18	30	0.22
0.04	18	30	0.22
0.05	18	5	0.26
0.06	5	5	0.89
0.07	5	5	0.89
0.08	2	5	0.46
0.09	2	2	1
0.10	2	2	1
⋮	⋮	⋮	⋮
0.21	2	2	1
0.22	2	1	0
0.23	1	1	1

Table 8.1: Results of FSC-S(D_o) with FSC-DSC on the log(AHP) data. The shaded area shows the highest ARI reflected from a match of the two K's over the optimal σ values. The table suggest $k = 2$ according to the highest ARI. In addition, based on the match of K_{odd} and K_{even} we can also arrive to $k = 5$ clusters with ARI=0.89.

σ	K (odd set)	K (even set)	ARI
1000	30	29	0.67
2000	30	21	0.11
3000	29	21	0.29
4000	29	17	0.21
5000	18	17	0.89
6000	7	17	0.29
7000	5	3	0.49
⋮	⋮	⋮	⋮
11000	5	3	0.49
12000	2	3	0.89
13000	2	3	0.79
14000	2	2	1
⋮	⋮	⋮	⋮
31000	2	2	1
32000	2	1	0
33000	1	1	1

Table 8.2: Results of FSC-S(D_o) with FSC-DSC of the AHP data. The shaded area shows the highest ARI reflected from a match of the two K's over the optimal σ values. The table suggest $k = 2$ according to the highest ARI. Note, the large values of σ in this application.

8.3.2 General Downsampling Criteria (DSC)

In this section we present the results of applying the general downsampling criteria to the AHP data. As we have mentioned in Chapter 5, the general DSC was designed to work with all CFD approaches. However, we will look into using FSC-S(D_o) in more detail and will interpret the outcomes based on FSC-S(D_o). Consider the results on the logarithmic scale as shown in Figure 8.8. The highest median ARI is at $k = 5$ and then $k = 2$, which suggests similar clustering structure to the results of the specific FSC-DSC, while applying the algorithm on the original data gave the results as shown in Figure 8.9. There is a clear peak at $k = 2$, which is consistent

with the results of the specific FSC-DSC when applied to the original AHP data.

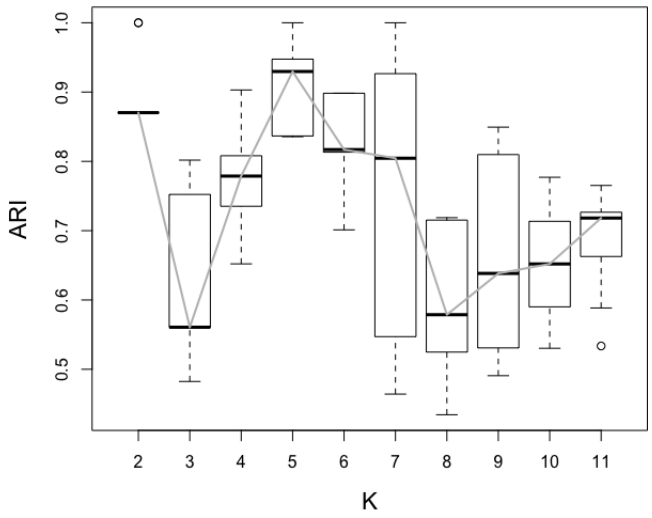


Figure 8.8: Boxplots of the ARI over K based on the general DSC with FSC-S(D_o) on the log(AHP) data. The approach suggests there are 2 and 5 clusters in the dataset.

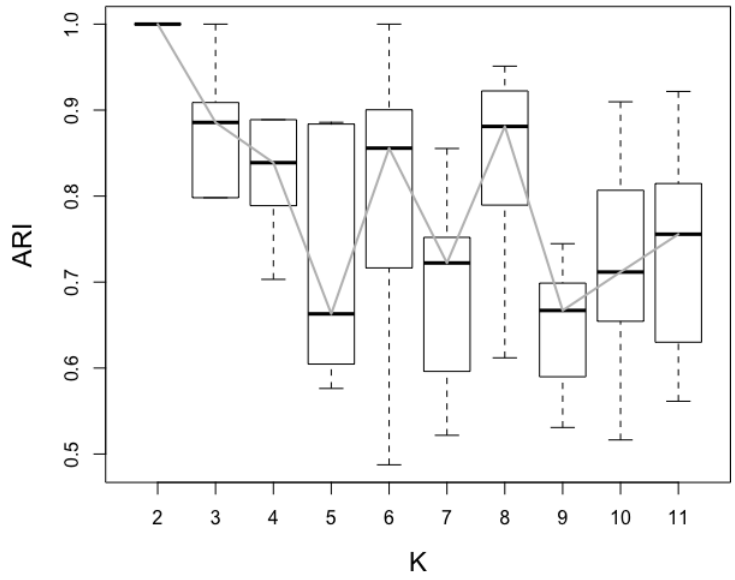


Figure 8.9: Boxplots of the ARI over K based on the general DSC with FSC-S(D_o) on the AHP data. The approach suggests there are 2 clusters in the dataset.

Further, applying the approach with the chosen CFD methods gave the results as displayed in Figure 8.10 and Figure 8.11 for the logarithmic scale and the original data respectively. In Figure 8.10, we notice that FunHDDC, FD-kmeans, Bsplines-Km, and FPCA-mbc all detect $k = 2$ in the data. However, both FD-kmeans and FPCA-mbc could not converge in all replicates from $k = 5$ and above, hence leading to missing ARI values. FSC-S(D_1) and FSC-S(D_2) could not detect any clustering structure in the data. Although there is a peak at $k = 3$ in the FSC-S(D_1) curve, we cannot rely on this value as the ARI is low and below 0.4. The only approach that suggests k could be 2 or 5 is FSC-S(D_o).

In Figure 8.11, we see that all the methods gave $k = 2$, apart from FSC-S(D_2) that showed no peak at any k, while FD-kmeans and FPCA-mbc could not converge in most iterations at $k > 4$ values. Again FSC-S(D_1) suggested $k = 3$ besides $k = 2$ but both at low ARI values.

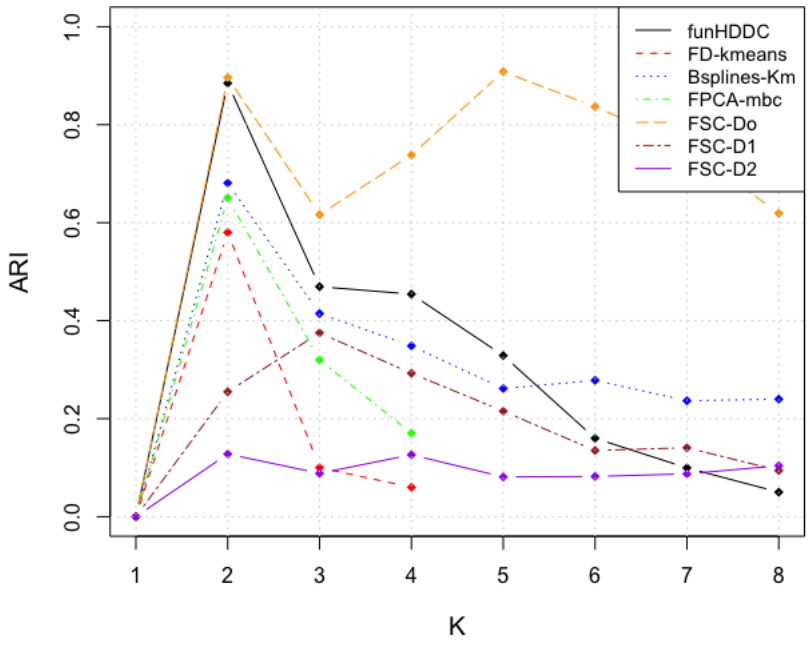


Figure 8.10: Results of the **mean** ARI for each K based on the general DSC with different CFD approaches on the log(AHP) data.

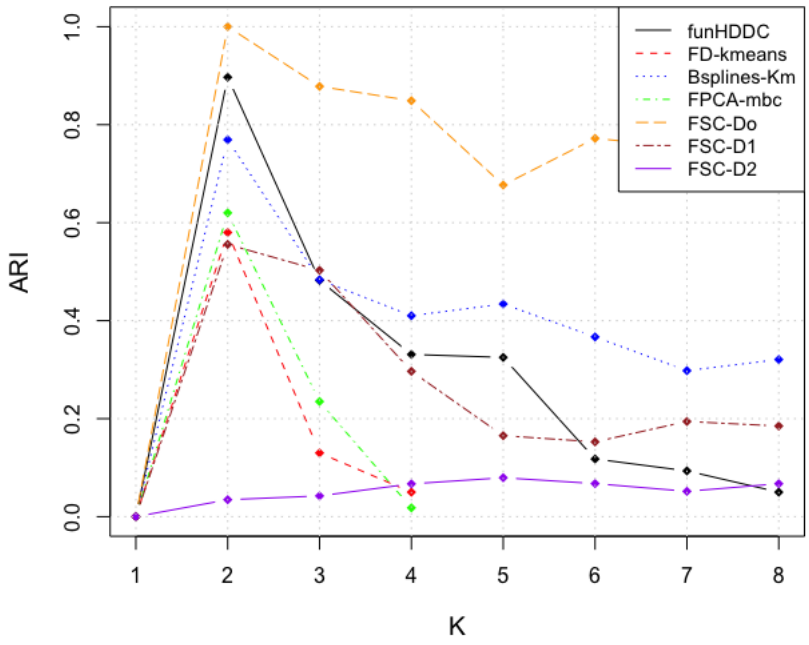


Figure 8.11: Results of the **mean** ARI for each K based on the general DSC with different CFD approaches on the AHP data.

8.4 Results

The house sale market is influenced by many factors such as location and property-based characteristics, and these factors vary from one council area to another. In this chapter, we attempted to identify the clustering structure of the house prices in Scotland based on the annual average house prices recorded within a council area from 1993 to 2018. There are 32 council areas, however 'Na h-Eileanan Siar' consists of some missing values, thus it was excluded from the study. According to the model selection downsampling approaches, the council areas could be categorized as 2 clusters or as 5 clusters based on clustering the logarithmic scale. The 2 clusters of AHP curves are shown in Figure 8.12, and in general it divides the curves into "low-price" and "high-price" ranges across the time range. Although the discrimination between the two groups might not be very clear at the beginning of the timeline, it is more obvious after 2008. The 5 clusters of the data curves are shown in Figure 8.13. This clustering structure is a further

division of the 2 super-clusters, where the "low-price" category is clustered into 2 sub-clusters, and the "high-price" category is clustered into 3 sub-clusters.

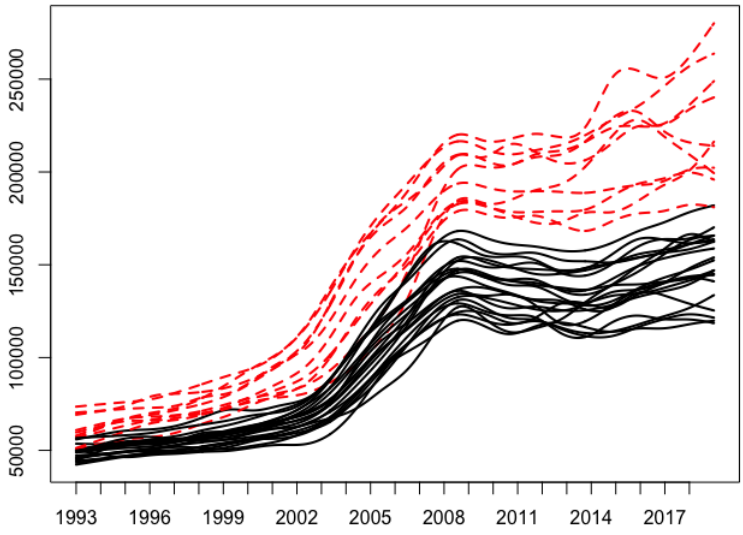


Figure 8.12: Smoothed curves of the AHP data clustered based on FSC-S(D_o) for $k = 2$.

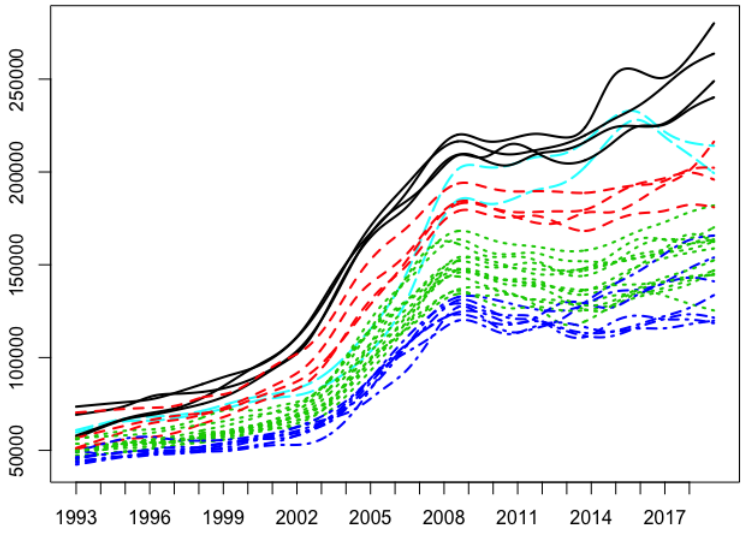
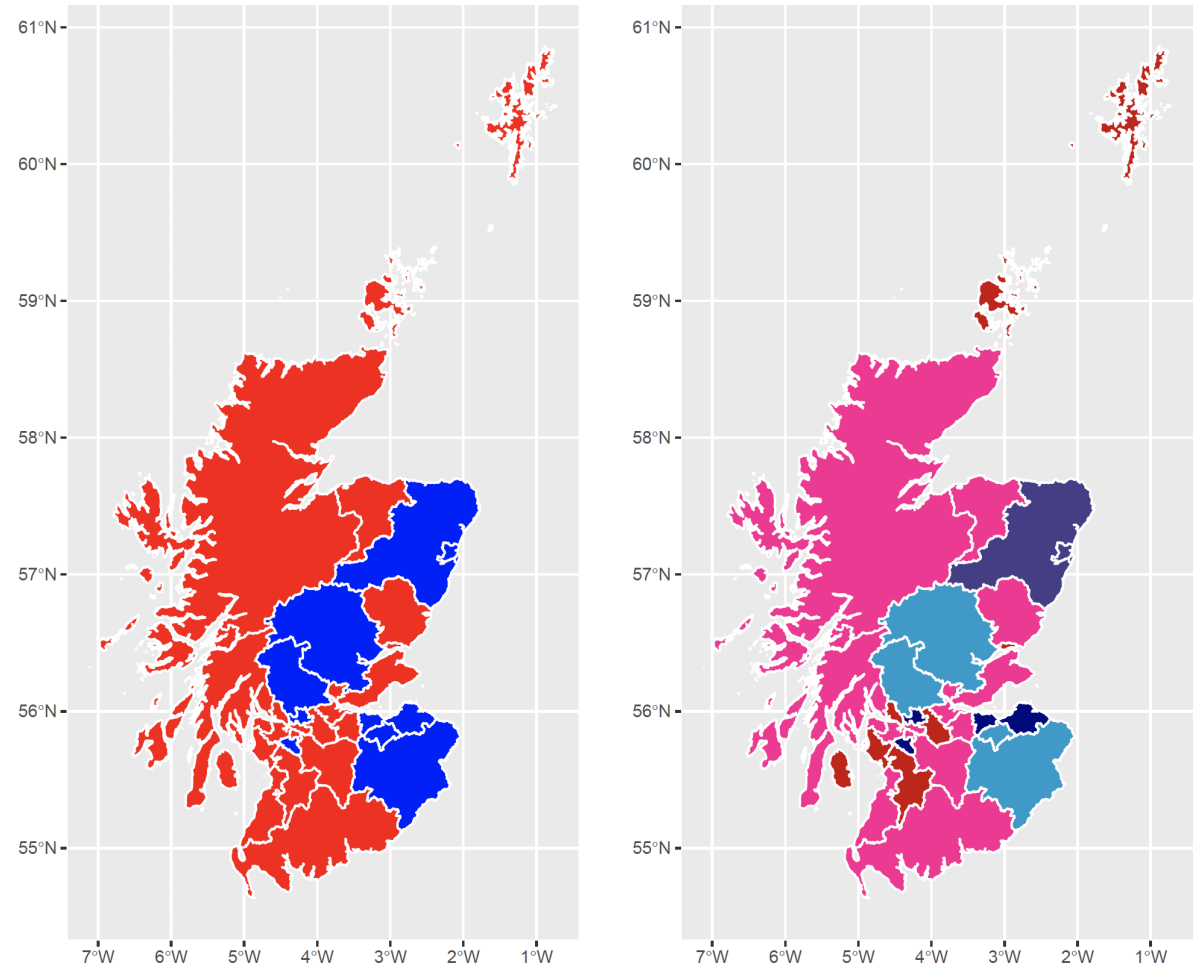


Figure 8.13: Smoothed curves of the AHP data clustered based on FSC-S(D_o) for $k = 5$.

We allocate the council areas into super-clusters and sub-clusters as shown in Table 8.3. In addition, we projected the resulted clusters on the Scottish map to see if any spatially-related clusters appear. Figure 8.14a shows the 2 super-clusters, and Figure 8.14b shows the 5 sub-clusters on the Scottish map.

Super-clusters	Sub-clusters	Council Areas
One	1	City of Edinburgh East Dunbartonshire East Lothian East Renfrewshire
	2	Aberdeen City Aberdeenshire
	3	Midlothian Perth and Kinross Scottish Borders Stirling
Two	4	Angus Argyll and Bute Clackmannanshire Dumfries and Galloway Falkirk Fife Glasgow City Highland Inverclyde Moray Renfrewshire South Ayrshire South Lanarkshire West Lothian
	5	Dundee City East Ayrshire North Ayrshire North Lanarkshire Orkney Islands Shetland Islands West Dunbartonshire

Table 8.3: The Scottish council areas categorized as 2 super clusters and 5 sub clusters based on the DSC approaches. The sub-clusters from 1 to 5 are arranged in ascending order from highest to lowest AHP.



(a) $k = 2$ (b) $k = 5$

Figure 8.14: The council areas clustered according to the DSC approaches with FSC-S(D_o).

In Figure 8.14a we notice that there are some specially-related clusters which appear in the two super-clusters. This is because neighbouring areas are more likely to share similar socio-economic characteristics in terms of deprivation and population behaviour. Although that might not be very clear in Figure 8.14b, yet to some extent there are some neighbouring areas that fall in the same sub-clusters.

Chapter 9

Conclusion

This research focused on the cluster analysis of functional data and tackled two separate problems; proposing a new two-stage clustering approach for clustering functional data, and proposing a new model selection criteria for choosing the appropriate number of clusters in clustering functional data. We started by defining the statistical aspect of functional data analysis and we reviewed the main contributions of functional data clustering methods. There have been a wide range of approaches proposed for clustering functional data in the literature, mentioned in Chapter 3. Based on the literature, we found that spectral clustering has not been used previously as a functional data clustering method. In addition, we reviewed the proposed approaches for choosing the appropriate number of clusters in the context of clustering techniques for functional data as well as multivariate data. We found that the clustering stability concept was used frequently in the non-parametric methods of multivariate clustering analysis, but there were very limited applications of clustering stability in the context of clustering functional data.

In this thesis, we developed the functional spectral clustering framework (FSC-S) by employing spectral clustering and using the features of functional data. We also introduced the downsampling approach and initially proposed the general downsampling criteria (DSC) that can be used with any functional data clustering methods. Later, we combined the proposed functional spectral clustering approach and the downsampling criteria to build the integrated FSC-DSC, that can inherently estimate the number of clusters in the data. The three proposed

approaches were first illustrated on the Berkeley growth data. Then, they have been extensively examined through a comprehensive simulation study and an application of Scottish house price data. In general, the proposed approaches showed promising results in clustering functional data and in identifying the appropriate number of clusters, both in simulations and in real-life datasets. However, there are a number of limitations and a range of unresolved questions that can be addressed by extending the current work.

9.1 Discussion and Limitations

Here we briefly review the proposed techniques to highlight some points and discuss some of the limitations.

9.1.1 FSC-S

The developed FSC-S approach falls within the two-stage clustering category that applies the dimension reduction first and then moves to clustering. One of the potential problems with this type of approach is the possibility of losing any discriminative features between clusters during the process of dimension reduction. One way to overcome this issue is by applying a saturated smoothing model. Thus we defined the general smoothing model to be a saturated model of B-splines of order 4 or above that can give up to the second derivatives, while the smoothness is controlled by the smoothing parameter λ which is chosen based on the GCV. We found that a good smoothing model that reflects a good data fit will lead to reasonable clustering results.

The resulting smoothed curves are used in the clustering process. Then, the distance between the smoothed curves is measured by Simpson's Rule. In addition, we created more distance measures that are based on the first derivatives and the second derivatives. We found that measuring the distances between the derivative functions gives new information about the structure of the data. Based on the possible different ways of measuring the distances in functional data, we created the FSC-S(D_o), FSC-S(D_1), and FSC-S(D_2) approaches. Beyond this step, comes the application of the spectral clustering technique and projecting the resulting clusters on the orig-

inal trajectories of the functional data.

The three FSC techniques proposed in this thesis did not always give the same accuracy rates. In the data sets where the original functional data show clear structure of phase and/or amplitude variations, FSC-S(D_o) can detect the clusters better than FSC-S(D_1) and FSC-S(D_2). Whereas, if the first derivatives of the functional data provide more information about the data, FSC-S(D_1) will be more accurate in defining the clustering structure of the expected groups in the data. In limited datasets, the second derivatives of the curves will be informative in terms of the clustering structure, in which FSC-S(D_2) will give better clustering results. In fact, we noticed that the FSC-S technique that outperformed the other two techniques is most often associated with curves that show clear pattern of phase and/or amplitude variation. For instance, if the original data consists of these variations, then FSC-S(D_o) would outperform the derivative-based FSC-S techniques. Beyond the original curves, if the first or second derivatives show phase and amplitude variation, then FSC-S(D_1) or FSC-S(D_2) will perform better.

One of the limitations of the FSC-S technique is the need to provide the parameters k (number of clusters) and σ (scaling parameter in the similarity matrix). Another limitation is that we usually cannot know in advance which FSC-S technique will perform best. A good practice is to plot the original curves, the first derivatives and the second derivatives and visually inspect their patterns. Also, we found that we can exclude FSC-S(D_2) in most applications. Therefore, we can limit the implementation to FSC-S(D_o) and FSC-S(D_1). In most situations, the two techniques are able to define an appropriate clustering structure based on their scale, but often lead to different inferences about the data.

9.1.2 DSC

The developed paradigm for model selection is based on the downsampling technique, which divides the original data into two non-overlapping replicates (odd and even). The newly created low-resolution replicates represent 50% of the original data. However, to be able to examine the clustering stability, we needed to create more than 2 replicates. Therefore, we developed a

semi-systematic sampling scheme that is based on splitting the timeline into subintervals each consisting of 6 time points (i.e. $p = 6$). There are 20 possible combinations of 6 taken 3 at a time, thus we defined 20 different sampling patterns based on logical sets of T and F. We paired every 2 opposite sets to make 1 pair of odd and even replicates to avoid any correlation between the two replicates. As a result, there were 10 different pairs each consisting of 2 unique logical sets. According to this sampling scheme, the number of replicates for examining clustering stability is limited and cannot exceed 20 replicates. However, we can explore different sampling patterns beyond $p = 6$, for instance, it would be of interest to set $p = 8$ or 10 in dense functional data. On the other hand, we can set $p = 4$ in sparse data that do not show dynamic curvature structure.

Based on this sampling scheme, the original observed data y_i was downsampled into 10 y_{odd_i} and 10 y_{even_i} unique replicates each consisting of half of y_i, and the same smoothing model was applied to all replicates. Then, a set of number of clusters, K was provided, where generally we set $K = \{k = 2, ..., k = 15\}$, and the clustering was applied to each downsampled replicate for each number of clusters. The two opposite replicates in the same pair were compared by ARI, finally the number of clusters k that led to the highest stability of the partition was retained.

In this thesis we proposed that the DSC techniques could be used with any method for clustering functional data. However, in some situations the clustering approaches that involve model-based clustering such as Fun-HDDC, FD-Kmeans, and FPCA-mbc could not converge in all iterations specifically when K was large.

The main limitation of this criterion is that it does not work on sparse data and might fail in data that show high levels of noise. Another limitation of this approach is that it depends on the success of the used CFD method, and if the method performs poorly then it is not possible to retain any information about the number of clusters in the data, as was the case when using FSC-S(D_2).

9.1.3 FSC-DSC

We have addressed the limitation of FSC-S by employing the downsampling criteria to select the number of clusters k over a range of σ values and accordingly cluster the functional data. Using one pair of odd and even replicates, each replicate was smoothed and then at every value of σ, spectral clustering was applied and k was estimated by the eigengap heuristic. As a consequence, the functional data was clustered based on the resulting k at each σ. Based on the stability clustering, the stable clustering structure was reflected by a range of σ that in turn led to an optimal k. In addition, this outcome was obtained by the two replicates, which were compared by the ARI.

The additional of downsampling criteria added a distinctive feature to our integrated approach. However, as was the case for the general DSC, the specific FSC-DSC cannot be applied to sparse data or functional data with high noise. Yet, usually the noise can be controlled by the smoothing parameter. Another limitation is related to the appropriate range of σ that must be inspected for exploring the k values. We have mentioned that we set pre-fixed ranges of σ consisting of short equispaced intervals. However, so far, these intervals are checked manually, and the interval that shows variations among the 15 smallest eigenvalues (since largest possible $k = 15$) is selected for further more detailed exploration of k. We are currently working on enhancing the FSC-DSC algorithm so that it can estimate the appropriate range of σ inherently.

Some further notes of interest

Further, we have noticed a few interesting aspects while working on this thesis, which could benefit from further exploration. These are:

- A designed distance measure that comes from the original curves and first derivatives can sometimes lead to better accuracy rates when clustering in some scenarios of simulated data. In contrast, the same designed measure, in other scenarios, gave low accuracy rates. For this reason, we didn't proceed with a measure that combines both the original curves and the first derivatives, since we could not explain that behaviour.

- FSC-S(D_2) cannot support estimating k in both the general DSC and the specific FSC-DSC approaches. The main reason is that the second derivative functions are critically affected by the downsampling process that takes place in the original data scale.

- In some examples it is possible to use multivariate information resulting from FSC-S(D_o) and FSC-S(D_1) within the specific FSC-DSC to interpret the final clustering results.

9.2 Future Work

There are several potential extensions to the proposed algorithms in this thesis in terms of both methodology and application. From a methodological perspective, the FSC-S techniques can be applied with different smoothing models. In addition, it is possible to enhance the FSC-DSC approach to be able to simultaneously apply the smoothing and the clustering. The new model-based FSC-DSC approach will avoid the issue of choosing the appropriate smoothing parameter λ and the effect of dimension reduction on the clustering results. In addition, it is of interest to try different distance measures for estimating the distance matrix. Then, the remaining of the FSC-S algorithm can be applied as suggested in the thesis. Some suggestions for alternative distance metrics are based on Marron et al. (2015) or Tzeng et al. (2016). Further, the similarity graph can be replaced by another measure. In this thesis, we used the fully connected graph, however the k-nearest neighbour graph, or the ε-neighbourhood graph could also be used. Although Von Luxburg (2007) stated that it has not been yet proved theoretically whether the choice of similarity graph will affect the results of spectral clustering, it is of interest to study the effect of the similarity graph practically on functional data.

From an application perspective, in this thesis, we have focused on one-dimensional functional data, the axis being time (t). However, many applications involving functional data come as multi-dimensional output such as time-space data (t,s). It is of interest to expand the applications of our proposed FSC-S techniques and FSC-DSC approach to multi-dimensional functional data. Moreover, all the applications of our proposed approaches were examined on functional data that consists of less than 100 curves (data objects). It would be interesting to examine the

approaches on functional data that consists of a larger number of curves. We believe that the performance of FSC-S and DSC are likely to be similar in larger data sets. However, for FSC-DSC, we may need to adjust the method of calculating the eigengap, because including all the possible eigenvalues n (where $n =$ number of curves) could obscure the optimal eigengap. Finally, since we applied the general DSC on a limited number of CFD techniques, it is of interest to consider different clustering techniques beyond the ones chosen in this thesis.

Bibliography

Abraham, C., P.-A. Cornillon, E. Matzner-Løber, and N. Molinari (2003). Unsupervised curve clustering using B-splines. *Scandinavian Journal of Statistics 30*(3), 581–595.

Afzalan, M. and F. Jazizadeh (2019). An automated spectral clustering for multi-scale data. *Neurocomputing 347*, 94–108.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control 19*(6), 716–723.

Al Alawi, M., S. Ray, and M. Gupta (2019). A new framework for distance-based functional clustering. In *34th International Workshop on Statistical Modelling, Portugal, July 2019*.

Andreotti, E., D. Edelmann, N. Guglielmi, and C. Lubich (2020). Measuring the stability of spectral clustering. *Linear Algebra and Its Applications 610*, 673–697.

Banfield, J. D. and A. E. Raftery (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics 49*, 803–821.

Barnard, S. T., A. Pothen, and H. Simon (1995). A spectral algorithm for envelope reduction of sparse matrices. *Numerical Linear Algebra with Applications 2*(4), 317–334.

Barnes, E. R. and A. J. Hoffman (1984). Partitioning, spectra and linear programming. In *Progress in Combinatorial Optimization*, pp. 13–25. Elsevier.

Ben-David, S., U. Von Luxburg, and D. Pál (2006). A sober look at clustering stability. In *International Conference on Computational Learning Theory*, pp. 5–19. Springer.

Ben-Hur, A., A. Elisseeff, and I. Guyon (2001). A stability based method for discovering structure in clustered data. In *Biocomputing 2002*, pp. 6–17. World Scientific.

Bhatia, R. (1987). *Perturbation bounds for matrix eigenvalues*, Volume 53. Siam.

Bhatia, R. (1997). Matrix analysis. *Graduate Texts in Mathematics 169*.

Bouveyron, C., E. Côme, J. Jacques, et al. (2015). The discriminative functional mixture model for a comparative analysis of bike sharing systems. *Annals of Applied Statistics 9*(4), 1726–1760.

Bouveyron, C., S. Girard, and C. Schmid (2007). High-dimensional data clustering. *Computational Statistics & Data Analysis 52*(1), 502–519.

Bouveyron, C. and J. Jacques (2011). Model-based clustering of time series in group-specific functional subspaces. *Advances in Data Analysis and Classification 5*(4), 281–300.

Bruneau, P., O. Parisot, and B. Otjacques (2014). A heuristic for the automatic parametrization of the spectral clustering algorithm. In *2014 22nd International Conference on Pattern Recognition*, pp. 1313–1318. IEEE.

Carmichael, J. and R. Julius (1968). Finding natural clusters. *Systematic Biology 17*(2), 144–150.

Chacón, J. E. (2021). A close-up comparison of the misclassification error distance and the adjusted rand index for external clustering evaluation. *British Journal of Mathematical and Statistical Psychology 74*(2), 203–231.

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

Chung, F. R. and F. C. Graham (1997). *Spectral graph theory*. Number 92. American Mathematical Soc.

Davis, C. and W. M. Kahan (1970). The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis 7*(1), 1–46.

De Boor, C., C. De Boor, E.-U. Mathematicien, C. De Boor, and C. De Boor (1978). *A practical guide to splines*, Volume 27. Springer-Verlag New York.

Donath, W. E. and A. J. Hoffman (1973). Lower bounds for the partitioning of graphs. In *Selected Papers Of Alan J Hoffman: With Commentary*, pp. 420–425. IBM J. Res. Develop.

Everitt, B. S., S. Landau, M. Leese, and D. Stahl (2011). *Cluster analysis 5th ed.* John Wiley.

Fan, K. (1950). On a theorem of Weyl concerning eigenvalues of linear transformations: Ii. *Proceedings of the National Academy of Sciences of the United States of America 36*(1), 31–35.

Febrero-Bande, M., M. O. de la Fuente, et al. (2012). Statistical computing in functional data analysis: The R package fda.usc. *Journal of Statistical Software 51*(4), 1–28.

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal 23*(2), 298–305.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning 2*(2), 139–172.

Fraley, C. and A. E. Raftery (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association 97*(458), 611–631.

Giacofci, M., S. Lambert-Lacroix, G. Marot, and F. Picard (2013). Wavelet-based clustering for mixed-effects functional models in high dimension. *Biometrics 69*(1), 31–40.

Guattery, S. and G. L. Miller (1998). On the quality of spectral separators. *SIAM Journal on Matrix Analysis and Applications 19*(3), 701–719.

Guha, S., R. Rastogi, and K. Shim (2000). ROCK: A robust clustering algorithm for categorical attributes. *Information Systems 25*(5), 345–366.

Guo, W. (2004). Functional data analysis in longitudinal settings using smoothing splines. *Statistical methods in medical research 13*(1), 49–62.

Hébrail, G., B. Hugueney, Y. Lechevallier, and F. Rossi (2010). Exploratory analysis of functional data via clustering and optimal segmentation. *Neurocomputing 73*(7-9), 1125–1141.

Hennig, C. (2007). Cluster-wise assessment of cluster stability. *Computational Statistics & Data Analysis 52*(1), 258–271.

Hennig, C., M. Meila, F. Murtagh, and R. Rocci (2015). *Handbook of cluster analysis*. CRC Press.

Hess, S. and W. Duivesteijn (2019). K is the magic number—inferring the number of clusters through nonparametric concentration inequalities. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 257–273. Springer.

Hubert, L. and P. Arabie (1985). Comparing partitions. *Journal of Classification 2*(1), 193–218.

Ieva, F., A. M. Paganoni, D. Pigoli, and V. Vitelli (2013). Multivariate functional clustering for the morphological analysis of electrocardiograph curves. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 62*(3), 401–418.

Jacques, J. and C. Preda (2013). Funclust: A curves clustering method using functional random variables density approximation. *Neurocomputing 112*, 164–171.

Jacques, J. and C. Preda (2014a). Functional data clustering: A survey. *Advances in Data Analysis and Classification 8*(3), 231–255.

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

James, G. M. and C. A. Sugar (2003). Clustering for sparsely sampled functional data. *Journal of the American Statistical Association 98*(462), 397–408.

Jiang, H. and N. Serban (2012). Clustering random curves under spatial interdependence with application to service accessibility. *Technometrics 54*(2), 108–119.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika 32*(3), 241–254.

Karypis, G., E.-H. Han, and V. Kumar (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer 32*(8), 68–75.

Kaufman, L. and P. J. Rousseeuw (1990). Partitioning around medoids (program PAM). *Finding Groups in Data: an Introduction to Cluster Analysis 344*, 68–125.

Kayano, M., K. Dozono, and S. Konishi (2010). Functional cluster analysis via orthonormalized Gaussian basis expansions and its application. *Journal of Classification 27*(2), 211–230.

Kolotilina, L. Y. (2000). A generalization of Weyl's inequalities with implications. *Journal of Mathematical Sciences 101*(4), 3255–3260.

Lange, T., V. Roth, M. L. Braun, and J. M. Buhmann (2004). Stability-based validation of clustering solutions. *Neural Computation 16*(6), 1299–1323.

Laukaitis, A. and A. Račkauskas (2005). Functional data analysis for clients segmentation tasks. *European journal of operational research 163*, 210–216.

Lucasius, C. B., A. D. Dane, and G. Kateman (1993). On K-medoid clustering of large data sets with the aid of a genetic algorithm: background, feasiblity and comparison. *Analytica Chimica Acta 282*(3), 647–669.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, Volume 1, pp. 281–297. Oakland, CA, USA.

Marron, J. S., J. O. Ramsay, L. M. Sangalli, and A. Srivastava (2015). Functional data analysis of amplitude and phase variation. *Statistical Science 30*(4), 468–484.

Menardi, G. (2016). A review on modal clustering. *International Statistical Review 84*(3), 413–433.

Nagpal, A., A. Jatain, and D. Gaur (2013). Review based on data clustering algorithms. In *2013 IEEE Conference on Information & Communication Technologies*, pp. 298–303. IEEE.

Ng, A. Y., M. I. Jordan, and Y. Weiss (2002). On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pp. 849–856.

Parodi, A., M. Patriarca, L. Sangalli, P. Secchi, S. Vantini, and V. Vitelli (2014). fdakma: Functional data analysis: K-mean alignment. *R Package Version 1*(1).

Peng, J., H.-G. Müller, et al. (2008). Distance-based clustering of sparsely observed stochastic processes, with applications to online auctions. *Annals of Applied Statistics 2*(3), 1056–1077.

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

Ramsay, J. and B. Silverman (2005). *Functional Data Analysis*. Springer, c1997. New York.

Ramsay, J. O. and C. Dalzell (1991). Some tools for functional data analysis. *Journal of the Royal Statistical Society: Series B Methodological 53*(3), 539–561.

Ramsay, J. O., H. Wickham, S. Graves, and G. Hooker (2014). fda: Functional data analysis. *R Package Version 2*(4), 142.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association 66*(336), 846–850.

Ratcliffe, S. J., L. R. Leader, and G. Z. Heller (2002). Functional data analysis with application to periodically stimulated foetal heart rate data. I: Functional regression. *Statistics in Medicine 21*(8), 1103–1114.

Ray, S. and B. Mallick (2006). Functional clustering by Bayesian wavelet methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*(2), 305–332.

Richardson, S. and P. J. Green (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 59*(4), 731–792.

Same, A., F. Chamroukhi, G. Govaert, and P. Aknin (2011). Model-based clustering and segmentation of time series with changes in regime. *Advances in Data Analysis and Classification 5*(4), 301–321.

Sangalli, L. M., P. Secchi, S. Vantini, and V. Vitelli (2010). K-mean alignment for curve clustering. *Computational Statistics & Data Analysis 54*(5), 1219–1233.

Scarpa, B. and D. B. Dunson (2009). Bayesian hierarchical functional data analysis via contaminated informative priors. *Biometrics 65*(3), 772–780.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *Annals of Statistics 6*(2), 461–464.

Scottish Government (2021). House Prices Data. http://statistics.gov.scot/data/house-sales-prices.

Serban, N. and L. Wasserman (2005). Cats: clustering after transformation and smoothing. *Journal of the American Statistical Association 100*(471), 990–999.

Shi, J. and J. Malik (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*(8), 888–905.

Srivastava, A., W. Wu, S. Kurtek, E. Klassen, and J. S. Marron (2011). Registration of functional data using Fisher-Rao metric. *arXiv preprint arXiv:1103.3817*.

Stewart, G. and J. Sun (1990). *Matrix perturbation theory: Computer Science and Scientific Computing*. Academic press New York.

Suarez, A. J., S. Ghosal, et al. (2016). Bayesian clustering of functional data using local features. *Bayesian Analysis 11*(1), 71–98.

Sugar, C. A. and G. M. James (2003). Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association 98*(463), 750–763.

Tan, P.-N., M. Steinbach, and V. Kumar (2005, 01). Cluster analysis: Basic concepts and algorithms. *Introduction to data mining*, 487–568.

Tao, T. (2010). 254a, notes 3a: Eigenvalues and sums of hermitian matrices. Terence Tao's Blog.

Tokushige, S., H. Yadohisa, and K. Inada (2007). Crisp and fuzzy K-means clustering algorithms for multivariate functional data. *Computational Statistics 22*(1), 1–16.

Towers, S. (2013). ASU AML 610 Fall 2013 lecture series. Notes on K-means-clustering.

Tzeng, S., C. Hennig, Y.-F. Li, and C.-J. Lin (2016). Distance for functional data clustering based on smoothing parameter commutation. *arXiv preprint arXiv:1604.02668*.

Ullah, S. and C. F. Finch (2013). Applications of functional data analysis: A systematic review. *BMC Medical Research Methodology 13*(1), 43.

Verma, D. and M. Meila (2003). A comparison of spectral clustering algorithms. *University of Washington Tech Rep UWCSE030501 1*, 1–18.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing 17*(4), 395–416.

Von Luxburg, U. et al. (2010). Clustering stability: An overview. *Foundations and Trends® in Machine Learning 2*(3), 235–274.

Von Luxburg, U., M. Belkin, and O. Bousquet (2008). Consistency of spectral clustering. *Annals of Statistics 36*(2), 555–586.

Wang, J. (2010). Consistent selection of the number of clusters via crossvalidation. *Biometrika 97*(4), 893–904.

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

Weyl, H. (1912). Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). *Mathematische Annalen 71*(4), 441–479.

Wolfe, J. H. (1970). Pattern clustering by multivariate mixture analysis. *Multivariate behavioral research 5*(3), 329–350.

Xie, J. (1997). A note on the Davis-Kahan theorem. *Linear Algebra and Its Applications 258*, 129–135.

Xu, D. and Y. Tian (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science 2*(2), 165–193.

Yamamoto, M. (2012). Clustering of functional data in a low-dimensional subspace. *Advances in Data Analysis and Classification 6*(3), 219–247.

Yassouridis, C. and F. Leisch (2017). Benchmarking different clustering algorithms on functional data. *Advances in Data Analysis and Classification 11*(3), 467–492.

Zelnik-Manor, L. and P. Perona (2005). Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, pp. 1601–1608.

Zhang, Z., D. Pati, and A. Srivastava (2015). Bayesian clustering of shapes of curves. *Journal of Statistical Planning and Inference 166*, 171–186.