Clustering Multivariate and Functional Data Using Spatial Rank Functions

by

Mohammed Hussein Hassan Baragilly

A thesis submitted to
The University of Birmingham
for the degree of
Doctor of Philosophy

School of Mathematics
The University of Birmingham
June 2016

Abstract

A problem with cluster analysis is to determine the optimal number of clusters in the multivariate and functional data. In many of the clustering methods, the number of clusters is assumed to be fixed a priori. Moreover, in most exploratory applications, the number of clusters is unknown. That makes the determination of the number of clusters a very important problem in cluster analysis. Practically, determining the correct number of clusters depends on the experience of the investigator and the nature of the study. Statistically, many attempts and algorithms have been suggested in order to determine the optimal number of clusters. Over the last 40 years, a wealth of publications has been developed, introduced and discussed many graphical approaches and statistical algorithms in order to determine the optimal the number of clusters. In this work, we consider the problem of determining the number of clusters in the multivariate and functional data, where the data are represented by a mixture model in which each component corresponds to a different cluster without any prior knowledge of the number of clusters. For the multivariate case, the forward search algorithm is a graphical approach that helps us in this task. Three different forward search algorithms are considered in this study. The traditional forward search approach based on Mahalanobis distances has been introduced by Hadi (1992) and Atkinson (1994), while Atkinson et al (2004) used it as a clustering method. But like many other Mahalanobis distance based methods, it cannot be correctly applied to asymmetric distributions and more generally, to distributions which depart from

the elliptical symmetry assumption. We propose a new forward search methodology based on spatial ranks, where clusters are grown with one data point at a time sequentially, using spatial ranks with respect to the points already in the subsample. The algorithm starts from a randomly chosen initial subsample. We illustrate with simulated data that the proposed algorithm is robust to the choice of initial subsample and it performs well in different mixture multivariate distributions. We also propose a modified algorithm based on the volume of central rank regions. Our numerical examples show that it produces the best results under elliptic symmetry and it outperforms the traditional forward search based on Mahalanobis distances.

In addition, a second multivariate clustering method is proposed in this study. It is a new nonparametric clustering method based on different weighted spatial ranks (WSR) functions. They are completely data-driven and easy to compute without any need to parameter estimates of the underlying distributions, which make them robust against distributional assumptions. The WSR is more accurate in the purpose of intuitive visualization since we can easily determine the number of clusters from the weighted ranks contours for a low-dimensional input space, using dimension reduction. The main idea behind WSR is to define a dissimilarity measure locally based on a localized version of multivariate ranks. As a result, the proposed method can be used to determine the assumed number of clusters, and to assign each observation to its cluster. Selection of a proper weight function will lead to better identification of clusters when the data do not follow any standard parametric distribution. We have considered parametric and nonparametric weights for comparison. We have also introduced some WSR functions based on different robust weights like Mallow weight that has been introduced by Simpson et al. (1992) and Naranjo and Hettmansperger (1994). Moreover, many other different kernel weight functions have been considered. We give some numerical examples based on both simulated and real data sets to illustrate the performance of the proposed method.

In the age of technology, challenges of analysis, storage, and visualization of big-data have been become a very active topic in statistics, especially when the dimension d is large compared to the number of observations. Recently, functional data analysis received an attention in very diversified areas of scientific disciplines. In this study, there is a large body of work on using the ordinary and weighted spatial ranks as functional data clustering approaches. We propose two different clustering methods for functional data. The first method is an extension to the forward search based on spatial ranks that we proposed for the multivariate case, and it can initially be used to identify the number of clusters in the curves. This method can be considered as a new raw-data method since we do not use any preprocessing functional data steps, and we do not need to perform a data registration or a dimension reduction before clustering. In the second method, we extend the WSR method that has been introduced for the multivariate case to the functional data analysis. The proposed weighted functional spatial ranks (WFSR) method can be considered as one of the 2-stage methods, or the filtering methods, where it first approximate the curves into some basis functions and reduce the dimension using the functional principle components analysis (FPCA) and then perform the clustering using the basis expansion coefficients and the functional principle components scores. The WFSR method can be used to determine the assumed number of clusters, and assign each curve to its cluster. Both methods are completely data-driven and easy to compute without any need to estimate parameters of the underlying distributions, which make them robust against distributional assumptions. Different numerical examples from simulated and real data have been given in order to check the reliability of the proposed methods. Comparison between the existing methods, using the probability of misclassification error, has been considered as well. The results showed that the two proposed methods give a competitive and quite reasonable clustering analysis.

Dedication

To my beloved parents, my lovely sister and my brother. Your sincerest love, care, patience, and big hearts led me to where I am today and have given me courage to overcome the most difficult times in my life.

To you, I dedicate this thesis.

Acknowledgments

I would like to express my sincerest thanks to my supervisor, Dr. Biman Chakraborty, for his great guidance, support and patience throughout the course of this thesis. Without his helpful discussion, constructive guidance, warm encouragement and honest advice over four years I have pursued my education and research at the University of Birmingham, this work could not have been done. His unconditional support strengthened my abilities to move forward and to achieve my research goals.

I would like also to express my greatest appreciation to my sponsor: The Egyptian Government, and specially the Egyptian Cultural Centre and Educational Bureau in London for the continuous monitoring of my progress and supporting me over my research and the Department of Applied Statistics in Helwan University for the support and guidance.

A researcher cannot be successful without having a good environment to work in; I was lucky from the beginning of my work to get involved with kind and supportive Friends. I owe my deep appreciation and sincerest thanks to my dear friend Hend Gabr for her great support and giving me her endless and unconditional concern, also I extend my appreciation to my kind friend Olusola Makinde for providing a friendly and enjoyable environment during my time here.

Words are not enough to express my gratitude towards my parents, so I'm going to simply say "thank you" to my loving parents. I also give my most heart-felt thanks to my sister and brother for their kind encouragements to be the person I am today.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Recently, cluster analysis has become one of the most important statistical tools to study and clarify many scientific disciplines that require to cluster data in the aim to understand and interpret the studied phenomenon. It is a helpful exploratory tool for analysis of the low and high dimensional multivariate data and functional data as well. A particular attention is paid to the cluster analysis for the higher dimensional data. There have been many clustering methods, techniques and algorithms scattered in publications in very diversified areas such as pattern recognition, artificial intelligence, information technology, image processing, biology, psychology, market research, astronomy, psychiatry, weather classification, archaeology, bioinformatics and genetics (Gan et al., 2007; Everitt et al., 2011). However, the problem of determination the best number of clusters has been considered a kind of limitation in the most previous studies. Moreover, choosing a suitable clustering technique is another serious issue in cluster analysis. We postpone the discussion on the first problem to Section 1.6, where we present some of the common ways that try to address this problem of estimating the number of clusters, while we address the discussion on the second point to the next two Sections.

Examples of the first literature that discussed the clustering analysis have been given by Hartigan (1975), where the word clustering was known at this time as the numerical taxonomy. Sokal and Sneath (1963) has considered the first and most important book in numerical taxonomy. Some methods of measuring similarity are proposed in it, with particular attention given to category data. Clustering analysis is also known as Q-analysis, clumping, unsupervised learning, and typology. Many definitions for cluster analysis have been used in the literature. A simple one that has been considered by Hartigan (1975) is: "clustering is the grouping of similar objects". Jain and Dubes (1988) have defined the cluster analysis as "the formal study of algorithms and methods for grouping or classifying objects", while Kaufman and Rousseeuw (2005) defined it as "the art of finding groups in data". In pattern recognition and neural network studies, cluster analysis is known as unsupervised learning (training patterns with unknown category labels), and it is called segmentation analysis especially in the market studies.

It can be clearly seen that, all the previous definitions agree on an important point that the cluster analysis aims to group and partition a given set of data or objects into clusters, subsets, or groups, we need to know any properties should to be in this partitioning. The first property or assumption is the homogeneity within the clusters, i.e. points that belong to the same cluster should be as similar as possible. The second one is the heterogeneity between clusters, i.e. points that belong to different clusters should be as different as possible (Hoppner et al., 1999).

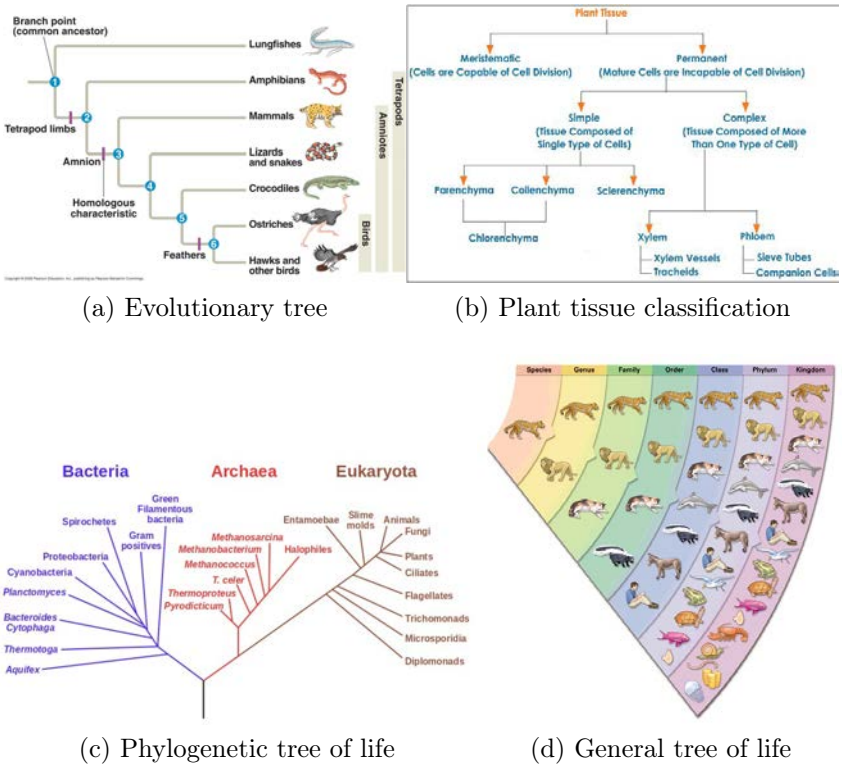
In order to illustrate the importance of cluster analysis, we provide some common examples that are usually used in the literature. The first example is the taxonomy of animals and plants. Taxonomy is the science of describing, naming, and classifying organisms which are described by their structure, appearance and hypothesized evolutionary relationships. The main interest of this science is that we need to cluster each animal, plant and species in the best way, which implies the homogeneity within the clusters and

the heterogeneity between clusters.

The same methodology can be considered for the phylogenetic tree of life, where there are three domains of life bacteria, archaea, eukarya and each of them consists of some components. For instance, the bacteria consist of 9 components while the animals and plants belong to the eukarya which contains 10 components as shown in Figure 1.1 (c). This tree gives a good example for the importance of the cluster analysis in our life, since it has provided physical significance in the evolutionary theories of Darwin. Figure 1.1 shows some examples of the taxonomy of animals and plants, where plots (a), (b), (c) and (d) give the evolutionary tree, classification of plant tissues, phylogenetic tree of life and general tree of life respectively.

A third example is the handwriting recognition. Recently, the handwriting recognition has become one of the important topics after the computerized information revolution has appeared. Many publications have been introduced and discussed some graphical approaches and statistical clustering algorithms that can be used to detect the handwriting numbers. Handwriting recognition concerns with the ability and performance of a computer to reed and interpret some handwritten inputs. These inputs may come from either paper documents or touchscreens. Cluster analysis and its novel algorithms that are introduced continuously, play an important role in this topic. Figure 1.2 shows some examples of the handwriting recognition, where plots (a) and (b) give the hierarchical clustering of handwritten numbers of a particular writer and some different handwritten numbers respectively.

Many other examples that show the importance of the cluster analysis could be given in this context, for example, one of the common cluster analysis examples is measuring the similarities of 11 languages, which has been introduced by Johnson and Wichern (2007). In this example, the main interest is clustering the most like European languages that use the Roman alphabet based on the numeral of these 11 languages.

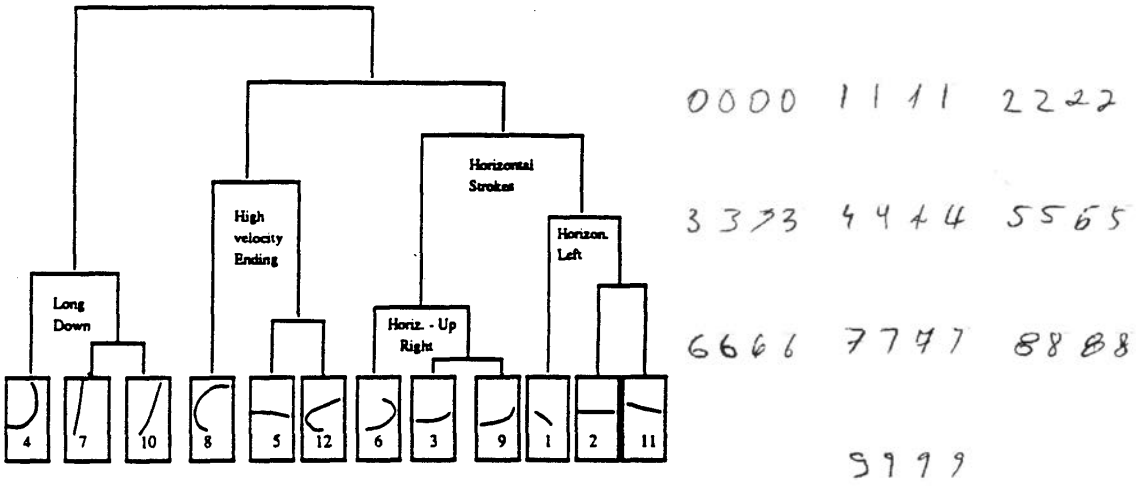


(a) Evolutionary tree (b) Plant tissue classification

(c) Phylogenetic tree of life (d) General tree of life

Figure 1.1: Examples of cluster analysis: The taxonomy of animals and plants (a) Evolutionary tree[64], (b) Classification of plant tissues[163], (c) Phylogenetic tree of life[162] and (d) General tree of life[75].

We cannot leave these examples of cluster analysis without mentioning the gene expression data topic, since cluster analysis is one of the most frequently approaches that are used to analyze the gene expression data. A particular attention is paid to the cluster analysis for the gene expression data, which is helpful to understand the functions of genes whose information has not been previously available (Eisen et al., 1998). As an example, a biologist would like to find out the clusters from the DNA microarray data on gene expressions, and consequently detecting the classes or subclasses of diseases. Full details about the cluster analysis for gene expression data can be found in (Gan et al., 2007). A good collection of the cluster analysis examples are available in (Hartigan, 1975; Gan et al., 2007; Everitt et al., 2011).



(a) Hierarchical clustering of handwritten numbers of a particular writer

(b) Different handwritten numbers

Figure 1.2: Examples of cluster analysis: The handwriting recognition (a) Hierarchical clustering of the 12 pen-strokes centroids of a particular writer[101], (b) Different handwritten numbers[53].

1.2 Hierarchical Clustering Methods

There are two major concerns in the existing literature regarding to the clustering techniques; hierarchical clustering techniques and non-hierarchical clustering techniques. However, the graphical approaches have been considered a third concern in literature. In this Section, we will discuss the similarity measures, the distance measures, the hierarchical clustering techniques and the graphical approaches.

In order to measure the homogeneity within or between clusters, we need to use some similarity measures. The similarity and dissimilarity measures in many cases are based on some measures of distance that help us to use some techniques in order to group the observations into clusters. Logically, after we use one of these measures, we need to apply some approach which can be considered as the initial clustering indicator. The graphical approaches are very useful approaches that give us initial view about if the data may contain clusters, and consequently we need to apply some formal clustering methods to

it. Two common types of techniques, known as hierarchical clustering techniques and non-hierarchical clustering techniques are usually used as attempt to find all grouping possibilities.

In order to measure the similarity between groups or samples, a measure of similarity or dissimilarity has to be defined. According to the data types, similarity measures can be used to determine either the similarity between entities or groups. For instance, one can use the association coefficients as similarity measure if the variable is binary, or use Gower coefficient (Gower, 1971) if the data is multinomial or quantitative data. Alternatively, dissimilarity (distance) measures can be used to determine the distance degree between entities or groups. A similarity coefficient indicates the strength of the relationship between two data points (Everitt, 1993). The various measures can be defined for several types of data such as: numerical, categorical, binary and mixed-typed data. Sneath and Sokal (1973), subdivided the similarity and dissimilarity coefficients into four groups; correlation coefficients, distance measures, association coefficients, and probabilistic similarity measures.

It is worth mentioning that there are two types of similarity or distance measures, the first is between entities (individuals) and the second is between groups. As we focus on measuring the similarity between entities, we give brief discussion about the similarity and dissimilarity measures between entities in Sections 1.2.1 and 1.2.2 respectively.

1.2.1 Similarity Measures

Similarity measures are used to describe how similar two data points (two clusters) are. There are many ways to measure the similarity or proximity between pairs of objects. One can consider the association coefficients as a similarity measures for the binary data, in order to cluster items, where each variable takes two cases in terms of counts of matches and mismatches (presence and absence) in each variable for two individuals. Suppose

that a represents the frequency of $1-1$ matches, b is the frequency of $1-0$ matches, c is the frequency of $0-1$ matches, and d is the frequency of $0-0$ matches, then we can put the frequencies of the matches and mismatches for item i and item k in the form of the contingency table 1.1:

Table 1.1: Counts of binary outcomes for two items.

	Item k			
	Outcome	1	0	Total
Item i	1	a	b	$a+b$
	0	c	d	$c+d$
	Total	$a+c$	$b+d$	$p=a+b+c+d$

There are many association coefficients that depend on the 2 by 2 association table like matching coefficient, Jaccard (1908) coefficient, Rogers and Tanimoto (1960) coefficient, Sneath and Sokal (1973) coefficient, and Gower and Legendre (1986) coefficient. More extensive coefficients can be found in Gower and Legendre (1986). Table 1.2 shows the common similarity measures for the binary data based on the frequencies in Table 1.1.

One of the important similarity measures that can be used to cluster variables is the correlation coefficient, which suggested by Karl Pearson (Pearson, 1920). Suppose that our variables are binary which can arranged in the contingency table 1.1, so we can get the usual product moment correlation coefficient, which is considered a common measure of the similarity between two binary variables and equivalent to the chi-square statistic for testing the independence of two categorical variables, as following:

$$r = \frac{ad-bc}{[(a+b)(c+d)(a+c)(b+d)]^{1/2}}. \tag{1.2.1}$$

Jukes and Cantor (1969), Tajima (1993) and Dyen et al. (1992) suggested some similarity measures for categorical data with more than two levels, while Gower (1971) proposed an important similarity measure that can apply for binary, multinomial, metric

Table 1.2: Common similarity measures for the binary data based on the frequencies.

Measure	**Coefficient Name**
$\frac{a+d}{a+b+c+d}$	Matching coefficient: Sokal and Michener
$\frac{a}{a+b+c}$	Jaccard coefficient
$\frac{a+d}{a+2(b+c)+d}$	Rogers and Tanimoto coefficient
$\frac{a}{a+2(b+c)}$	Sneath and Sokal coefficient
$\frac{a+d}{a+1/2(b+c)+d}$	Gower and Legendre coefficient
$\frac{a}{a+1/2(b+c)}$	Gower and Legendre coefficient
$\frac{2a}{2a+b+c}$	Czekanowski, Dice and Sorensen coefficient
$\frac{a}{a+b+c+d}$	Russell and Rao coefficient
$\frac{a}{b+c}$	Ratio of matches to mismatches coefficient

(quantitative), and mixed data. Suppose that SM_{ik} is the mean of similarity coefficient between the two points k, i ; x_{ij} is the $i-th$ observation for the variable j; and p is the number of variables, then Gower's coefficient for the mixed data takes the form:

$$SM_{ik} = \frac{\sum_{j=1}^{p} SM_{ikj}}{\sum_{j=1}^{p} W_{ikj}}; \qquad i \neq k = 1, ..., n \tag{1.2.2}$$

where W_{ikj} is a weight that takes two values, 1 when both entities i, k have the same property that we are interested in, and 0 otherwise; and SM_{ikj} is the similarity coefficient between the two entities i and k based on the variable j, which can be calculated in many ways depending on the type of data. More details can be found in Gower (1971).

1.2.2 Distance Measures

On the contrary of the similarity measures, dissimilarity (distance) measures determine the degree of distance between two entities, items or groups. Four standard criteria (mathematical properties), that can be used to judge whether a similarity measure is a true metric or not, are mentioned in Aldenderfer and Blashfield (1984), however not all the distance measures mentioned below are metrics.

The most popular distance measure, which is usually used for the numerical data, is the Euclidean distance (squared Euclidean distance, when the value of distance is squared). This distance is also known as the l_2 norm. Let d_{ik} be the distance between the two points k and i; x_{ij} is the $i-th$ observation for the variable j; and p is the number of variables, then the Euclidean distance can be written as:

$$d_{ik} = \left\{ \sum_{j=1}^{p} (x_{ij} - x_{kj})^2 \right\}^{1/2} . \tag{1.2.3}$$

However, it is well known that the Euclidean distance is affected by changes in the units of measurement, for example we would get two different values for the Euclidean distance between the weight and height if we considered the unit of measurement for the weight variable is kilograms instead of lbs. To address this problem, we need to standardize the variables by dividing by the standard deviation of each variable, such that:

$$d_{ik} = \left\{ \sum_{j=1}^{p} \left[\frac{(x_{ij} - x_{kj})^2}{\sigma_j^2} \right] \right\}^{1/2} . \tag{1.2.4}$$

this standardized form is not affected by the changes in the units of measurement. Another way to standardize the variables is based on the range R_j instead of the standard deviation. It is known as the maximum distance, which is defined to be the maximum value of the distances of the attributes, such that:

$$R_j = \max_{i,k} |x_{ij} - x_{kj}| . \tag{1.2.5}$$

Another well-known metric, which is used for the numerical data, is the city block or Manhattan distance (l_1 norm). The Manhattan distance was used in a cluster analysis context by Carmichael and Sneath (1969). It is usually used when the units of measurement are same for all the variables, and it takes the form:

$$d_{ik} = \sum_{j=1}^{p} W_j |x_{ij} - x_{kj}| . \tag{1.2.6}$$

It is worth mentioning in this context that, the Euclidean metric, Manhattan metric and maximum distance in (1.2.5) are special cases of the general Minkowski distance (l_r) at $r = 2, 1$ and ∞ respectively, where r is called the order of the Minkowski distance, and they satisfy the mathematical requirements of a distance function. The general form of Minkowski distance is:

$$d_{ik} = \left\{ \sum_{j=1}^{p} W_j |x_{ij} - x_{kj}|^r \right\}^{1/r} . \tag{1.2.7}$$

In literature Mahalanobis distance, also called generalized distance (Mahalanobis, 1936), has been considered as a very important metric that can be used for the numerical data. It takes the correlations among variables in the account by the inclusion of the variance-covariance matrix Σ, therefore, this distance applies a weight scheme to the data. The second feature is that Mahalanobis distance is invariant under all nonsingular transformations. When the correlation between variables is zero, Σ is the identity matrix; the squared Mahalanobis distance is same as the squared Euclidean distance. The Mahalanobis distance between two points $\underline{X}_i = (X_{i1}, ..., X_{id})^T$ and $\underline{X}_k = (X_{k1}, ..., X_{kd})^T$ in the d-dimensional space $\mathbb{R}^d$ is defined as:

$$d_{ik} = \sqrt{(X_i - X_k)^T \Sigma^{-1} (X_i - X_k)}. \tag{1.2.8}$$

A generalized Mahalanobis distance has been introduced by Morrison (1967), where the weight of each variable has been included in the measure by adding a diagonal matrix containing the p weights. At the same time, Gower (1967) suggested another distance measure for the numerical data which can be calculated by using the form:

$$d_{ik} = -\log_{10}\left(1 - \frac{1}{p}\sum_{j=1}^{p}\frac{|x_{ij} - x_{kj}|}{\beta_j - \alpha_j}\right). \tag{1.2.9}$$

where $\alpha_j = \min_{i \neq k}(x_{ij} - x_{kj})$ and $\beta_j = \max_{i \neq k}(x_{ij} - x_{kj})$. Many other non-metric distance measures for the numerical data that do not meet the metric conditions have been introduced over the literature. For instance, Sokal and Sneath (1963) proposed a distance function in order to measure the distance between two entities, such that:

$$d_{ik} = \sqrt{\frac{1}{p}\sum_{j=1}^{p}\left[\frac{(x_{ij} - x_{kj})}{(x_{ij} + x_{kj})}\right]^2}. \tag{1.2.10}$$

Canberra distance measure, introduced by Lance and Williams (1967), is often regarded as a generalization of the dissimilarity measure for binary data. They proposed the following two distance measures:

$$d_{ik}^{(L)} = \frac{\sum_{j=1}^{p}|x_{ij} - x_{kj}|}{\sum_{j=1}^{p}|x_{ij} + x_{kj}|}, \qquad d_{ik}^{(W)} = \frac{\sum_{j=1}^{p}|x_{ij} - x_{kj}|}{\sum_{j=1}^{p}|x_{ij}| + |x_{kj}|}. \tag{1.2.11}$$

Although all the above distance measures are used for the numerical data, there are many other distance measures that can be used for the categorical, binary and mixed data. For example, the simple matching dissimilarity measure and Harrison (1968) measure are usually used for the categorical data. Moreover, some studies suggested that we can use the relationship between the similarity and dissimilarity measures in order to get a distance

measure for the categorical data. So, one can use the transformation $d_{ik} = 1 - SM_{ik}$, if the data is binary, or Gower (1966,1967) transformation, $d_{ik} = \sqrt{2\,(1 - SM_{ik})}$, if the data is mixed data.

1.2.3 Agglomerative Hierarchical Methods

Two common types of methods, known as agglomerative hierarchical methods and divisive hierarchical methods, are usually used under the hierarchical clustering techniques. For the first type of methods, agglomerative hierarchical methods, they depend on the merging process so that we start with number of groups equals to the number of elements, in other words each element is in a cluster of its own. The most similar elements are first grouped, and then these groups are also merged, until all elements end up being in the same cluster. The result of the clustering can be displayed as a dendrogram, which illustrates the merger that have been made.

Linkage-Based Methods

In order to use one of the agglomerative hierarchical methods, we have to start by calculating the similarity or dissimilarity matrix between entities, and finish with the dendrogram. The most common and important agglomerative hierarchical methods are; single linkage, complete linkage, average linkage, coordinate centroid clustering, median clustering method, Ward's hierarchical clustering method and Lance and Williams flexible method. The only difference between the various agglomerative hierarchical methods, is that how can we calculate the similarity or distance measures between entities or subgroups. For instance, in the single linkage, complete linkage, and average linkage methods we start by searching about the most two similar elements in the distance (similarity) matrix, then we merge them to form the first cluster. The next step is calculating the distances between this cluster and the remaining elements by using specific distance function. After that, we continue in choosing the most two similar elements (clusters) in the

distance (similarity) matrix until all elements end up being in the same cluster. It is worth mentioning that, each of these three methods has its own distance function. The single linkage method has been considered one of the oldest methods of cluster analysis; it was suggested independently by Florek et al. (1951), McQuitty (1957) and Sneath (1957). Sibson (1973) proposed an important optimality efficient algorithm (SLINK) for the single linkage method, which can be applied on an unprecedented scale. Recently, many publications have developed this method. A useful overview is given in Muja and Lowe (2009), where they suggested an automatic algorithm to get fast approximate of the nearest neighbors and a system which takes any given dataset and desired degree of precision and use these to automatically determine the best algorithm and parameter values. On the other hand, the complete linkage method has been developed by Lance and Williams (1967) and Johnson (1967). Along the line of Sibson (1973), Defays (1977) introduced an efficient algorithm for the complete linkage method which is based like the (SLINK) algorithm and offers economy in computation. Following, we give brief discussion regarding to the three linkage-based methods, (single linkage, complete linkage, and average linkage methods).

We can conclude the main steps of the agglomerative hierarchical clustering algorithm as the following:

1. We start with number of groups equals to the number of elements, where each element is in a cluster of its own. Suppose we have n items, so we get firstly $n \times n$ distance (similarity) matrix by using any one of the measures that we have mentioned above.

2. In this step, we start by searching about the most two similar elements in the distance (similarity) matrix. Suppose that the most similar items (clusters) are i and k, and their distance $d_{i,k}$ are the smallest one, then we merge them to form the first cluster (i, k).

3. An update in the distance matrix should be done by omitting the rows and columns corresponding to items (clusters) i and k, and then we should add a new row and column giving the distances between cluster (i, k) and the other items (clusters) by using some distance function, where each method of them has its own distance function as we show below.

4. Continue in choosing the most two similar elements (clusters) in the distance (similarity) matrix and update it until all elements end up being in the same cluster.

Single Linkage Method

The single linkage method, which is also known as nearest neighbour clustering, depends on the nearest-neighbour rule. In this method, the distance between specific cluster (t) and another cluster (ik) that consists of the two entities i and k is calculated by using the distance function:

$$d_{t(ik)} = \min_{i \neq k} \left(d_{it}, d_{kt} \right), \tag{1.2.12}$$

or

$$SM_{t(ik)} = \max_{i \neq k} \left(SM_{it}, SM_{kt} \right), \tag{1.2.13}$$

if we use a similarity matrix instead of dissimilarity matrix.

Complete Linkage Method

On the other hand, the complete linkage method, which is also known as farthest neighbour clustering, depends on the furthest-neighbour rule. In this method, the distance between specific group (t) and another group (ik) is calculated by using the distance function:

$$d_{t(ik)} = \max_{i \neq k} \left(d_{it}, d_{kt} \right), \tag{1.2.14}$$

or

$$SM_{t(ik)} = \min_{i \neq k} \left(SM_{it}, SM_{kt} \right), \tag{1.2.15}$$

if we use a similarity matrix.

Average Linkage Method

In the third agglomerative hierarchical method, average linkage, which is also known as unweighted pair group method with arithmetic mean (UPGMA), the distance between two clusters can be considered as the average distance between all pairs of items where one member of a pair belongs to each cluster (Johnson and Wichern, 2007). Suppose that $n_{(ik)}$ and n_t are the number of items in clusters (ik) and t, respectively, then the distance, d_{uv}, between object u in the cluster t and object v in the cluster (ik) is obtained by:

$$d_{t(ik)} = \frac{\sum_u \sum_v d_{uv}}{n_{(ik)}.n_t}, \tag{1.2.16}$$

or

$$SM_{t(ik)} = \frac{\sum_u \sum_v SM_{uv}}{n_{(ik)}.n_t}, \tag{1.2.17}$$

if we use a similarity matrix.

Example 1.2.1 *Breakfast Cereals Data*

In this example, we consider the data on brands of breakfast cereals [Source: Data courtesy of Chad Dacus, Table 11.9 of Johnson and Wichern (2007)], which contains 43 types of breakfast cereals produced by three different American companies (General Mills (G), Kellogg's (K), and Quaker (Q)). It also contains eight variables, where eight numerical nutritional characteristics have been measured for each cereal. The variables are: Calories, Protein, Fat, Sodium, Fiber, Carbohydrates, Sugar, and Potassium. Also, the name of each cereal and the company that produced the cereal have been given.

Our target is clustering these 43 types of breakfast cereals based on the above 8 variables, by using the linkage-based methods (single, complete and average linkage methods). As we mentioned in the agglomerative hierarchical clustering algorithm, we start with number of groups equals to the number of breakfast cereals types (43), and we start clustering by merging the two cereals that have smallest nonzero distance in the distance matrix until all the cereals end up being in the same cluster. Figures 1.3, 1.4 and 1.5 show the single, complete and average linkage dendrograms for the distances between the 43 types of breakfast cereals respectively, where they illustrate the grouping and the partitions produced at each stage, for each method.

As we can see in Figure 1.3, the single linkage dendrogram does not give a clear clustering and it is very difficult visually to determine the number of clusters from the dendrogram. This is due to the distances between observations are not big enough to distinguish the separation between groups. For example, if we cut the tree at the height $h = 100$, then we get two clusters, one of them with only one cereals brand (AllBran) and the second cluster with the remaining 42 cereals brands. However, cutting the tree at $h = 85$ gives 3 clusters, which in fact means for each point x_{ij}, every other point x_{kj} in its cluster satisfies $d_{ik} \leqslant 85$, where d_{ik} is the Euclidean distances defined in (1.2.3). Compared to Figure 1.4, it can be clearly seen that the complete linkage dendrogram shows clearer clustering structure, where it is obviously gives an evidence about the existence of three clusters. The distances in this dendrogram are much wider, and that makes it easy visually to distinguish the number of clusters. For instance, cutting the complete linkage tree at $h = 250$ gives 3 clusters, which means for each point x_{ij}, every other point x_{kj} in its cluster satisfies $d_{ik} \leqslant 250$. For the average linkage method, we can see from Figure 1.5 that the dendrogram does not give reasonable result as the distances are not wide enough comparing to them in the complete linkage method. Like the single linkage dendrogram, the average one has assigned only one cereals brand (AllBran) in one cluster.

So, we can conclude that the complete linkage method outperforms both of the single and average linkage methods in this data. From this example, we can see that determining the cutting point in the linkage based methods is questionable, and may require some different methods to decide the right number of clusters.

Other Agglomerative Hierarchical Methods

The coordinate centroid clustering method, which is also known as unweighted pair-group method using the centroid approach (UPGMC), has been suggested by Sokal and Michener (1958), and then developed by King (1967) in order to cluster the variables. This method tries to merge each two subgroups to be fused together depending on the distance between their coordinate centroids. It starts also by searching about the most two similar elements in the distance matrix, then merging them to form the first cluster and calculate its corresponding centroid in order to adjust the distance matrix. We continue in choosing the most two similar elements (subgroups) in the distance matrix until all elements end up being in the same cluster like the linkage-based methods.

A disadvantage of this method is that, it is affected by the group size when one of the two subgroups to be fused has very different size than the other such that the centroid of two merged groups will be very close to that of the larger group and may remain within that group, and consequently the smaller group will lose its properties. For that reason, Gower (1967) has suggested the median clustering method, which assumes the equality of the size of each group to be fused.

Ward (1963) has introduced another hierarchical clustering procedure, which is depending on the error sum of squares (ESS) criterion. He pointed out that with each merging occur, there will be some loss of information, which can be measured by calculating the error sum of squares. According to Ward's method, one can consider that a merged group is acceptable if the increase in the total within-cluster error sum of squares (information loss) has been minimized as can as possible, and it is the reason to refer to

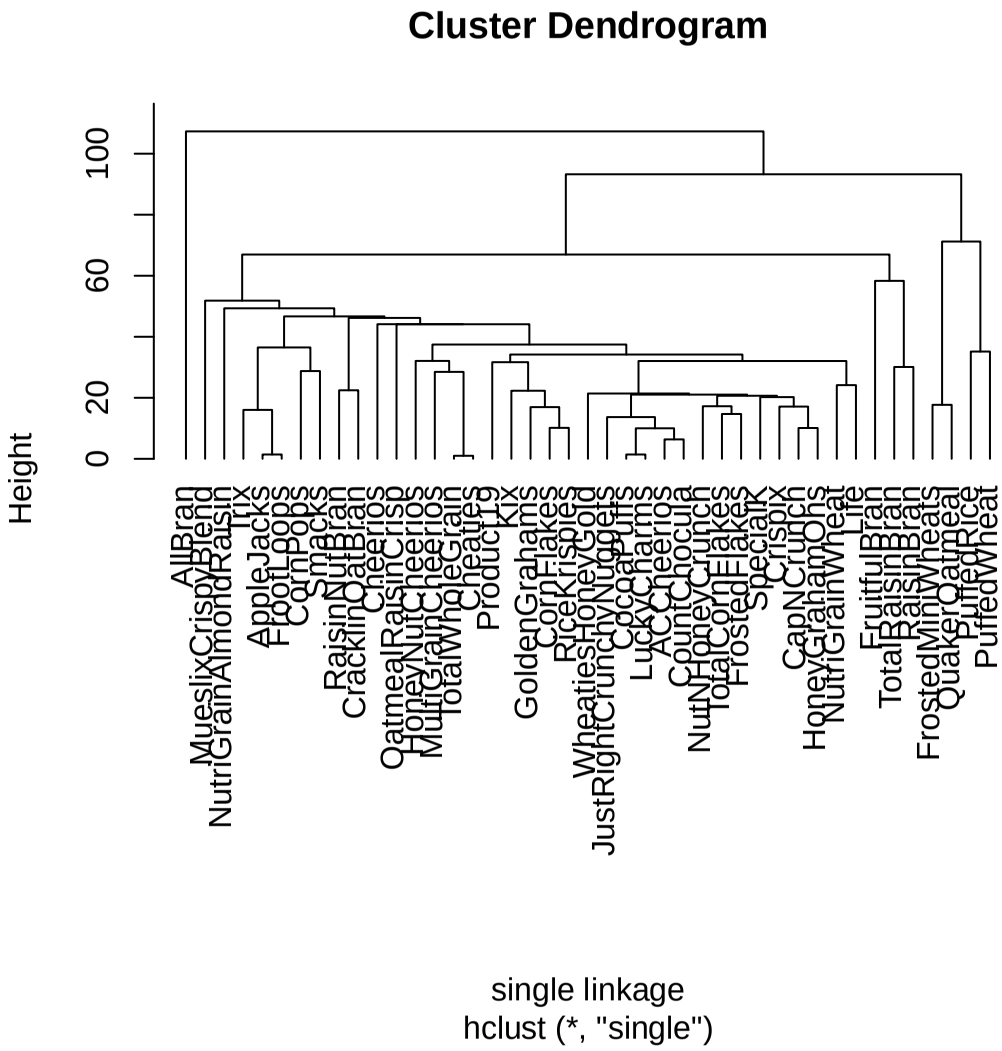


Figure 1.3: Example 1.2.1: Single linkage dendrogram for distances between 43 types of breakfast cereals.

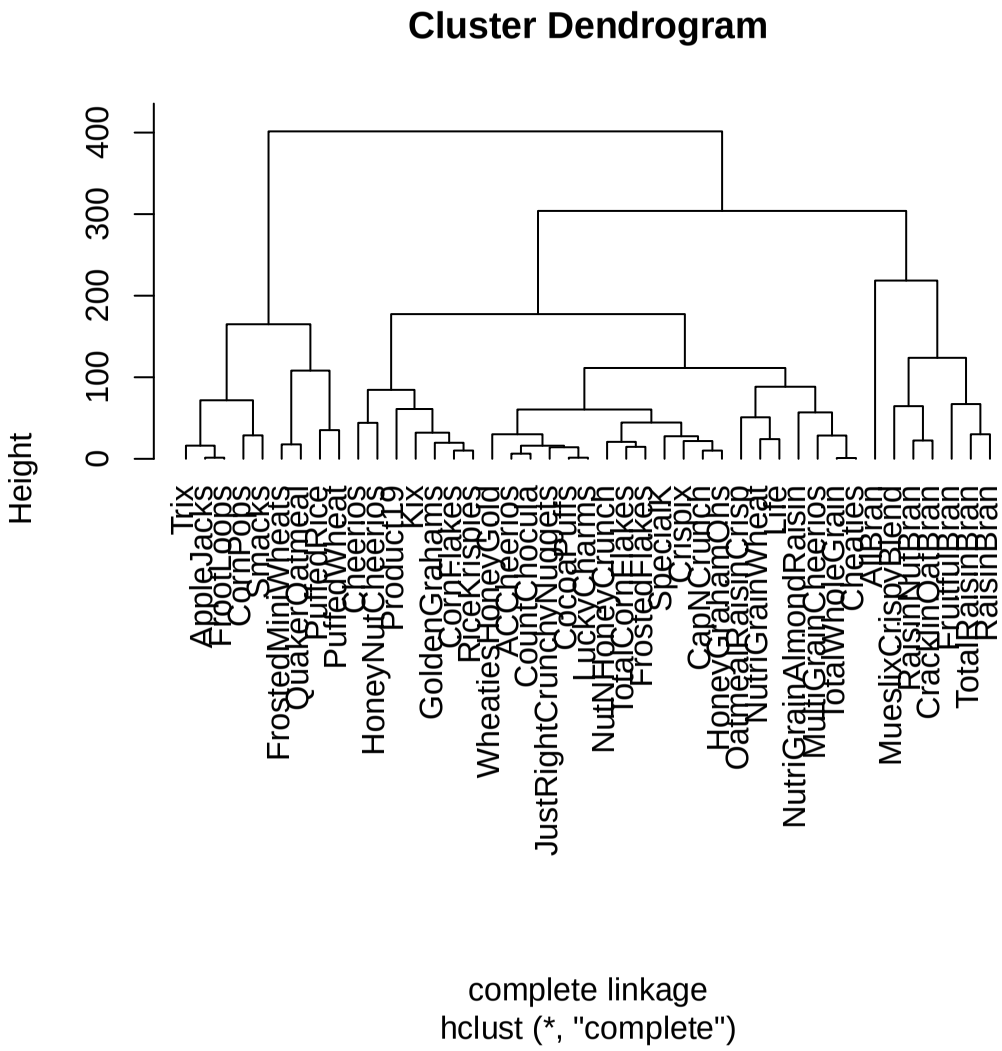


Figure 1.4: Example 1.2.1: Complete linkage dendrogram for distances between 43 types of breakfast cereals.

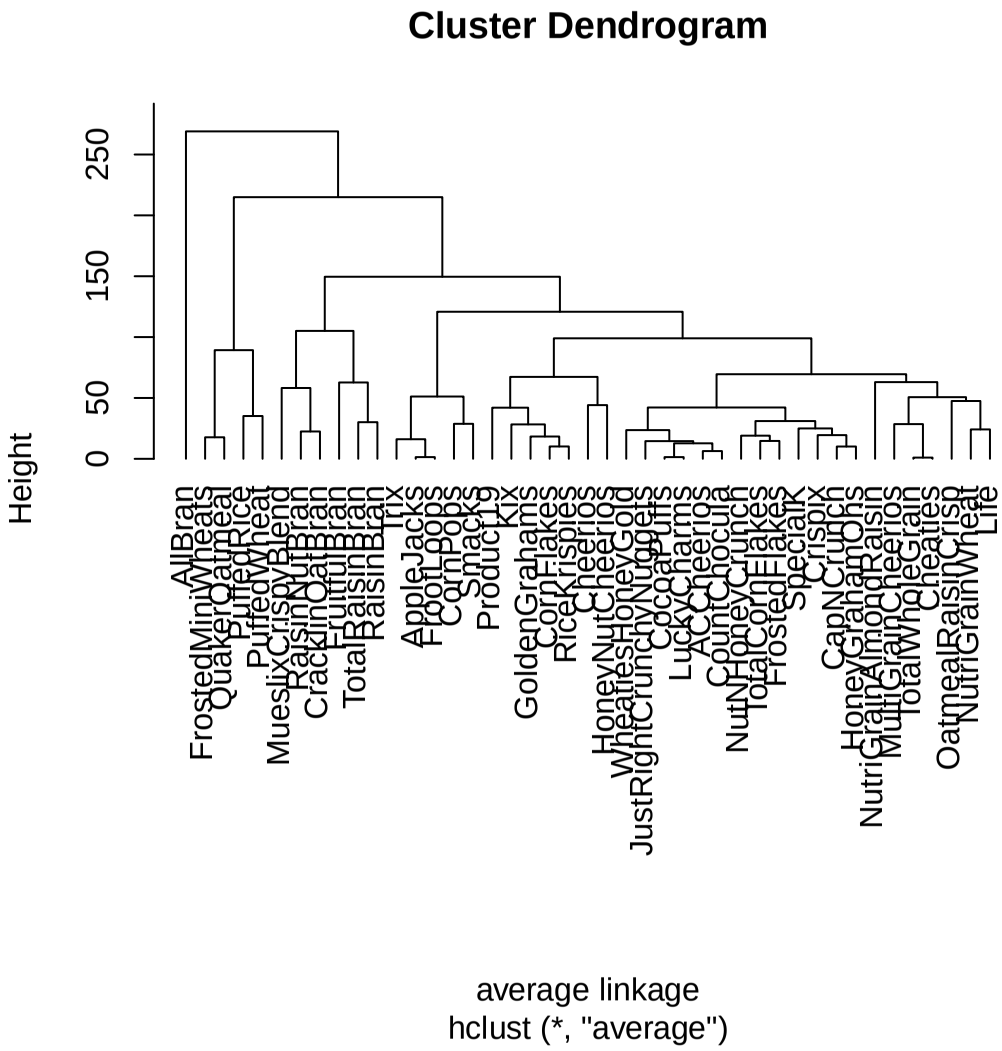


Figure 1.5: Example 1.2.1: Average linkage dendrogram for distances between 43 types of breakfast cereals.

this method as the minimum variance method. Lance and Williams (1967) have suggested a recurrence formula that gives the distance between two clusters t and (i, k), the formula is given by:

$$d_{t(i,k)} = \alpha_i d_{ti} + \alpha_k d_{tk} + \beta d_{ik} + \gamma \left| d_{ti} - d_{tk} \right|,$$

where d_{ik} is the distance between groups i and k while α, β and γ are parameters whose values are different for different methods above. It is worth mentioning that, all the distance measures between groups that are used by many cluster analysis methods satisfy Lance and Williams' recurrence formula by a suitable choice of the parameters α, β and γ. For instant, when $\alpha_i = \alpha_k = \frac{1}{2}, \beta = 0$ and $\gamma = -\frac{1}{2}$ the concept of single linkage is achieved. A table of parameters for standard methods can be found in Gan et al. (2007) and Everitt et al. (2011).

1.2.4 Divisive Hierarchical Methods

On the contrary of the agglomerative hierarchical methods, the divisive hierarchical methods start with a single cluster and then split it into two subclusters by using $2^{n-1} - 1$ possible divisions (Everitt, 1980), where n the number of elements, then successively splitting clusters. Divisive hierarchical clustering methods consist of two types of methods, monothetic divisive methods and polythetic divisive methods. A monothetic method divides the data on the basis of the possession or otherwise of a single specified attribute, while a polythetic method divides the data based on the values taken by all attributes.

MacNaughton-Smithe et al. (1964) have proposed the main features of the polythetic divisive methods. They pointed out that we have to find the element that is furthest away from the others within a group, and considering it as the seed for a splinter group. Then a splinter group is accumulated by sequential addition of the entity whose total

dissimilarity with the remainder less its total dissimilarity with the splinter group is a maximum (Everitt, 1980). Kaufman and Rousseeuw (1990) have developed a program for the polythetic divisive method which is known as DIANA (DIvisive ANAlysis clustering).

On the other hand, monothetic techniques are usually used when the data is binary, since they divide the data on the basis of the possession or otherwise of a single specified attribute. Basically, these monothetic techniques depend on optimizing a criterion reflecting either cluster homogeneity or association with other variables. Accordingly, two monothetic methods have been discussed in more details by Everitt (1980). The first one is the association analysis which creates division of a cluster into two sub-clusters in terms of the presence and absence of one of the binary characters for specific variable. By calculating the association coefficients (chi-square coefficient), we can divide the cluster according to the variable that has the maximum value of the calculated association coefficient. The second method is the automatic interaction detector method which determines the variables, and the categories within them, that are maximally different with respect to some dependent variable (Everitt, 1980). A good collection of the hierarchical methods applications is available in (Everitt et al., 2011), while many clustering algorithms have also been extensively studied in (Gan et al., 2007; Kaufman and Rousseeuw, 2005).

1.2.5 Graphical Approaches

A particular attention is paid to the graphical approaches over the existing literature, where they have become more and more popular in clustering analysis. They are considered an important tool that helps us in the clustering process, where some of them give initial view about if the data may contain clusters and consequently some formal clustering methods should be used. Other graphical approaches are used in order to determine the number of clusters in the multivariate and functional data. It is worth mentioning that, whether we use the hierarchical techniques or the non-hierarchical techniques, we

will desperately need to such that tool. However, some publications consider the graphical approaches are just helpful tool and others consider them as one of the clustering techniques. As part our study basically focuses on the forward plots as an indicator that gives us the expected number of clusters, we see that it is more reasonable to consider the graphical approaches as clustering techniques.

Thorndike (1953) has been considered one of the earlier graphical approaches to determine the suitable number of clusters. He pointed out that, with the increase in the number of clusters k, the average within-clusters distance will decrease, and the number of clusters will be acceptable when sudden fluctuation happen in the curve. Carmichael and Sneath (1969) also have used the graphical approaches in order to cluster the multivariate data by using the method of taxometric maps. Another graphical approach has been introduced by Sokal (1966). He have proposed a graphical representation of the dissimilarities matrix between the objects. After that, the distance graph was proposed and used by Chen et al. (1974). Hartigan (1975) showed how to plot the distances to detect clusters. Rousseeuw (1987) introduced a graphical representation a so-called silhouette (banner) plot that tell us how well each object lies within its cluster. He clarified that the height of the silhouette of a cluster gives information about the number of objects that lie within it, while its width tells us about the tightness of the cluster with respect to the other clusters. For more details about how to detect clusters graphically, see Everitt et al. (2011) and Kaufman and Rousseeuw (2005).

Many other publications used the scatterplot matrix and histograms that can be used as initial indicator to the presence of clusters. For instance, Atkinson (1994) and Riani and Cerioli (1999) have described variety of plots, such as scatterplot matrix and the stalactite plots, to detect the multivariate outliers which can be used as cluster technique. Many forward plots and entry plots are given by Atkinson et al. (2004), Atkinson et al. (2006), Atkinson and Riani (2007) and Riani et al. (2009) that are used to determine the

number of clusters in the multivariate data.

Basically, part of our methodology completely depends on the area of graph clustering, since the distinctive feature of this study is to introduce new algorithms and plots that can detect the right number of clusters in both multivariate and functional data. Our proposed forward plots based on either spatial ranks or volume of central rank regions and the weighted spatial ranks contours can be considered a new contribution to the graphical approaches group that can be used to determine the number of clusters graphically in both multivariate and functional data.

1.3 Non-hierarchical Clustering Methods

Non-hierarchical clustering methods, or partitional clustering methods, are usually used to group the elements into a collection of K clusters, rather than group the variables. The number of clusters, K, may either be specified in advance or determined as part of the clustering procedure. The non-hierarchical clustering methods operate in different direction to the hierarchical methods, since they find all the clusters simultaneously as a partition of the data unlike the hierarchical clustering methods that find the clusters in a hierarchical structure. In order to know this direction, three common types of non-hierarchical clustering techniques should be discussed here. They are known as optimization clustering techniques, density search techniques, and clumping techniques.

1.3.1 Optimization Clustering Techniques

For the first type of techniques, optimization clustering techniques, they are also known as iterative partitioning methods (Aldenderfer and Blashfield, 1984), and they can cluster the data by firstly assuming a specific number of clusters which can be determined by the investigator. An advantage of the optimization clustering techniques is that, they allow for the relocation of entities which can be considered as a correction step at every

clustering phase. It is worth mentioning that, there are two groups of the optimization clustering techniques that can be used. The first group is the methods that depending on the initial partition of the data, and the second group is the methods that depending on the clustering criterion.

The clustering methods based on the initial partition of entities usually start with an initial group of entities that known as seed points. These seed points are considered as initial estimates of the clusters center, and an entity can be allocated to the cluster with nearest center. Moreover, the seed point should be updated after every addition to the cluster. An important point should be noted in this context is the way of selection of these points, such that it has to be random way without any bias in the selection.

K-means Method

The most popular optimization clustering technique based on the initial partition of entities is known as K-means method. A particular attention is paid to the K-means over all the cluster analysis literature. In many publications, K-means has been considered a very important clustering algorithm that can be used given an important requirement. The initial number of clusters should be determined firstly. K-means has been independently introduced in different fields by Steinhaus (1957), Lloyd (1957), Ball and Hall (1965), and MacQueen (1967). However the huge number of clustering algorithms that have been published recently, it is still one of the most widely used algorithms for clustering for many reasons as ease of implementation, simplicity, efficiency, and empirical success (Jain, 2010). One of the distinctive features of K-means algorithm is that, it can be divided into two phases; the first one is the initialization phase where the elements can be randomly assigned into k clusters. The second one is the iteration phase, where the distance between each element and each cluster should be calculated and then the element can be assigned to the nearest cluster (Gan et al., 2007). The essential steps in the K-means algorithm are:

1. Start with an initial partition of the n items into K initial clusters.

2. Assign each item to the cluster whose mean is nearest by using a distance measure, and then calculate the change in the mean produced by moving each item to another group.

3. Repeat the previous step until no more reassignments (no improvement) take place.

Now, we consider the data on breakfast cereals introduced in Example 1.2.1. For the selection criterion, the CH index (Calinski and Harabasz, 1974) has been used. So we start with computing the CH index of K-means clustering assignments, over a range of the number of clusters K, in order to determine the initial suitable number of clusters. For a given number of clusters K, we can compute the CH index as following:

$$CH(K) = \frac{B(K)/(K-1)}{W(K)/(n-K)}, \tag{1.3.1}$$

where $W(K)$, $B(K)$ are the within and between cluster variations of $CH(K)$ respectively.

Table 1.3 gives the CH index for the breakfast cereals data based on the K-means clustering method, it can be clearly seen that the largest score $CH(K)$ existing when $K = 6$, the yellow shaded score in the table. That means $K = 6$ maximizes the CHindex. Figure 1.6 shows the CH index for the breakfast cereals data based on the K-means clustering method. Also, from this Figure, it can be clearly seen that the maximal point of $CH(K)$ occurs at $K = 6$, however the real number of clusters in this data is 3.

Table 1.3: Example 1.2.1: CH index for the breakfast cereals data based on the K-means clustering method.

K	2	3	4	5	6	7	8	9	10
CH(K)	26.12	34.76	38.28	42.69	48.10	46.07	45.83	42.82	47.48

Figure 1.7 is a scatterplot matrix which represents the K-means cluster centers and cluster assignments for the breakfast cereals data. The clustering assignments are marked

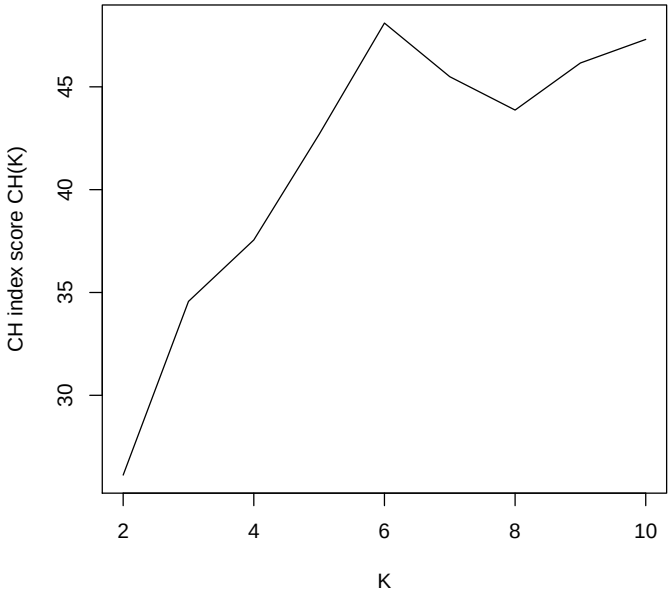


Figure 1.6: Example 1.2.1: CH index for the breakfast cereals data based on the K-means clustering method.

by colors. It can be clearly noticed from Figure 1.7 that we have six clusters, since $K = 6$ represents all the observations with the existence of homogeneity within the clusters and the heterogeneity between clusters. The K-means cluster centers are shown in Table 1.4. Clearly, the K-means behaves so poorly in this real dataset, where it failed to give the right clustering and it is difficult visually to see any clustering pattern in the scatterplot.

Table 1.4: Example 1.2.1: The K-means cluster centers for the breakfast cereals data.

Calories	Protein	Fat	Sodium	Fiber	Carbohydrates	Sugar	Potassium
114.44	3.11	1.67	171.11	2.78	15.00	6.56	123.89
107.14	2.71	0.71	282.86	0.79	18.21	4.43	56.43
111.43	1.93	1.00	199.29	0.68	14.68	8.57	51.43
110.00	1.60	0.60	110.00	0.80	11.40	13.20	29.00
112.50	3.25	0.75	225.00	5.75	12.50	10.75	245.00
75.00	2.75	0.50	0.00	1.68	9.50	2.00	68.75

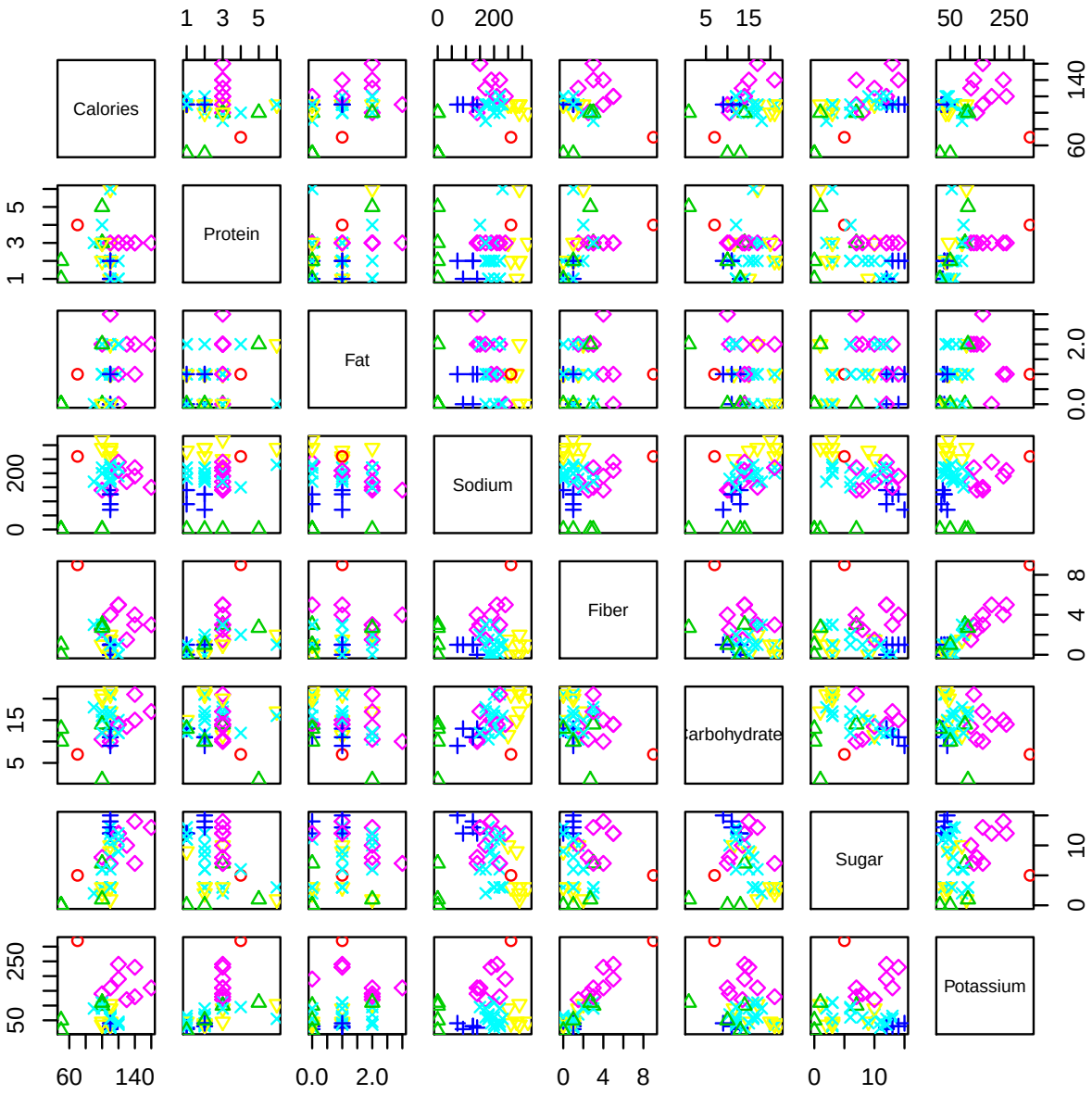


Figure 1.7: Example 1.2.1: Scatterplot matrix: The K-means cluster centers and cluster assignments for the breakfast cereals data.

A good overview of K-means method is given in Jain (2010) with a review for K-means algorithm, its historical developments, its parameters, and some of its extensions. Also Jain (2010) showed that K-means is one of the preferred algorithms in pattern recognition due to the nature of available data. Moreover, the author introduced a general review for various topics in cluster analysis, such that, data clustering and its historical developments, some major approaches to cluster analysis, the number of clusters, cluster validity, comparing clustering algorithms, trends in data clustering, semi-supervised clustering, large-scale clustering, and multi-way clustering.

Although K-means has been considered a very important clustering algorithm, some problems and properties should be mentioned in this context. The first problem is the computational complexity, where finding the optimal solution to the K-means clustering problem for n observations in d dimensions is computationally difficult (NP-hard), however, there are efficient heuristic algorithms (such as Lloyds algorithm) that are commonly employed and converge quickly to a local optimum. So, there is no guarantee that the solution will converge to the global optimum because it is a heuristic algorithm, and the result may depend on the initial clusters. Moreover, there is also no guarantee that the K-means algorithm gives clustering that globally minimizes the within-cluster variation. Third problem is that the final solution depends on the number and choice of the initial cluster centers, where different initial centers lead sometimes to different final clustering. So we may need to run K-means many times with random initial cluster centers and choose the one which gives the smallest within-cluster variation.

K-Medoids Method

A similar optimization clustering technique based on the initial partition of entities, is known as partitioning around medoids (PAM), has been proposed by (Reynolds et.al., 1992). The optimum average silhouette width (Kaufman and Rousseeuw, 1987) is usually used to estimate the number of clusters in the PAM algorithm. The method is also know

as K-median algorithms (Brusco and Kohn, 2009; Kohn et al., 2010). Instead of using the mean in the K-means method, K-medoids uses the median of the data to represent the clusters. It is similar to the K-means algorithm, except when fitting the centers we use the points themselves as centers without using their means. One of the important advantages of using K-medoids is that, it is more robust to noise and outliers compared to K-means. This is due to that the medoid is less influenced by outliers and extreme values than a mean. Moreover, it works efficiently for small datasets, but does not scale well for large data.

The essential steps in the partitioning around medoids (PAM) algorithm are:

1. Randomly Partition the items into K initial clusters by selecting K of the n data points as the medoids.

2. Associate each data point to the closest medoid by using some distance measure.

3. Compute the total cost of the configuration each medoid with each non-medoid data point.

4. Select the configuration with the lowest cost.

5. Repeat steps 2 to 4 until there is no change in the medoid.

The second group of the optimization clustering techniques is the methods that depend on the clustering criterion. These methods search about the entity that needs to relocate to another group, such that an improvement in a particular clustering criterion should be happen as a result of this reallocation. Many clustering criteria have been used for this purpose (Everitt, 1980). Three of them depend on the matrix $T = W + B$, where T is the total scatter or dispersion matrix, W is the matrix of within-groups dispersion, and B is the between-groups dispersion matrix. These clustering criteria are: minimization of $trace(W)$, minimization of the determinant of W, and maximization of $trace(BW^{-1})$.

It should be noticed that minimization of $trace(W)$ is equivalent to maximization of $trace(B)$, where $trace(T) = trace(W) + trace(B)$, and minimization of the determinant of W is equivalent to maximization of the ratio $|T|/|W|$.

Rubin (1967) have proposed another clustering criterion, which is known as average entity stability. It is based on the average similarity between the entity and the members of the group which is known as the attraction of an entity to specific group. An entity is said to be unstable when it is attracted to another group more than to the group it is in. In addition, there is another criterion proposed which is known as the information measure (Everitt, 1980). It is a function of the data, measurement accuracies and parameters of certain distributions. This criterion is applicable for the variables whether they are continuous and normally distributed or they are multistate and follow a multinomial distribution.

1.3.2 Density Search Techniques

In the second type of non-hierarchical clustering techniques (density search techniques), it is assumed that the entities are depicted as points in a metric space suggests that there should be parts of the space in which the points are very dense, separated by parts of low density. This could form the basis for the definition of a natural cluster (Everitt, 1980). Many clustering methods have been proposed for this purpose, like the TAXMAP method of Carmichael and Sneath, Gitman and Levine's methods for detecting unimodal fuzzy sets, Cartet count method, mode analysis, and method of mixtures.

TAXMAP method, it is also called the method of taxometric maps, is proposed by Carmichael and Sneath (1969), which has been considered the most popular density search technique. It represents a partition using a diagram in which the groups are displayed as circles and each diameter of a circle can be considered as the corresponding diameter of the cluster. If a cluster contains only one element, then it is represented by one point. In this

method, the clusters can be initially formed in a similar way to the single linkage method using some criteria to help us to know when we should stop adding elements to the cluster. Then it attempt to place the clusters in the map in such a way that the distances between them are proportional to their actual distance. Gitman and Levine (1970) proposed another method also starts in a similar way to the single linkage clustering technique with using a particular order to allocate each element to a suitable cluster. Mode analysis method also is a derivative of single linkage clustering that searches for natural subclusters by estimating disjoint density surfaces in the sample distribution. A disadvantage of mode analysis is that it cannot identify the large and small clusters simultaneously (Everitt et al., 2011). For more comprehensive details of this subject, Cartet count method, and Method of mixtures, the reader is referred to Everitt (1980) and Everitt et al (2011).

1.3.3 Clumping Techniques

Regarding to the third type of non-hierarchical clustering techniques (clumping techniques), they are usually used when there is an overlap between the clusters. In many studies and fields there is a correlation between the variables and groups which makes an overlap between the clusters, and it could be addressed by using one of the clumping techniques. These techniques depend on minimizing some function, which called a cohesion function, between each pair of groups. Needham (1967) proposed a mathematical form in order to define a symmetric cohesion function while Parker-Rhodes and Jackson (1969) modified it. Extensive review for the clumping techniques and their algorithms can be found in (Everitt et al., 2011).

1.4 Model-based Clustering

All the hierarchical and non-hierarchical clustering methods discussed earlier are intuitively reasonable procedures but they do not have a model to explain the way that the

observations were produced and their probabilistic distribution(Johnson and Wichern, 2007). Model-based clustering is a popular tool in clustering analysis due to its probabilistic foundations and its flexibility (Bouveyron and Brunet-Saumard, 2014). In the model-based clustering, the clusters are defined in a probabilistic framework, and this in fact helps to formalize the clusters' notion based on their probability distribution to interpret the obtained partition from a statistical concept. So, in model-based clustering, the data are represented by a mixture model in which each component corresponds to a different cluster, and each component is described by a density function and has an associated probability or weight in the mixture. In principle, any probability model for the components can be considered, but usually it is assumed that the components have p-variate normal distributions. Thus, the probability model for clustering will often be a mixture of multivariate normal distributions and each component in the mixture represents a cluster. In edition, Gaussian components with different parametrizations and cross-cluster constraints are usually used to formalize the models with different geometric properties.

The first works on finite mixture models were from Wolfe (1963) and Scott and Symons (1971). The classic reference for the model-based cluster analysis is Banfield and Raftery (1993) where they proposed the model-based Gaussian and non-Gaussian clustering based a reparameterization of the covariance matrix. They also proposed an approximate Bayesian method for choosing the number of clusters based on the Bayesian information criterion(BIC). Model-based clustering has been extensively studied in McLachlan and Basford (1988), McLachlan and Peel (2000), Banfield and Raftery (1993), Fraley (1998), Fraley and Raftery (1999) and Fraley and Raftery (2002), and it has become then a popular and reference technique. More comprehensive details and review are given in (Melnykov and Maitra, 2010; Bouveyron and Brunet-Saumard, 2014).

In a model based clustering approach, one may assume that the d-dimensional data

$\mathbf{X}_1, \ldots, \mathbf{X}_n$ are coming from a mixture probability density function

$$f(\mathbf{x}) = p_1 f_1(\mathbf{x}) + \cdots + p_k f_k(\mathbf{x}), \tag{1.4.1}$$

where $f_1, \ldots, f_k$ are d-dimensional unimodal density functions and $p_1, \cdots, p_k$ are the mixing proportions with $p_1 + \cdots + p_k = 1$. The clusters are often modeled by the same parametric density function with the finite mixture model:

$$f(\mathbf{x}) = \sum_{k=1}^{K} p_k f(\mathbf{x}; \theta_k), \tag{1.4.2}$$

where θ_k is the vector of parameters for the k-th mixture component. Accordingly, the log-likelihood of the above mixture model is:

$$\ell(\theta; \mathbf{x}) = \sum_{i=1}^{n} \log(\sum_{k=1}^{K} p_k f(\mathbf{x}_i; \theta_k)). \tag{1.4.3}$$

Since the group labels $z_1, \cdots, z_n$ of the observations are unknown, the inference of this model cannot be directly done through the maximization of the likelihood. This is due to the exponential number of solutions to explore, the maximization of the previous equation is unfortunately intractable, even for limited numbers of observations and groups (Bouveyron and Brunet-Saumard, 2014). As a result, we have to use some other inference algorithm. The expectation-maximization (EM) algorithm is the most popular inference algorithm in this case. The motivation behind using EM algorithm is to make inference for the models that not all their variables are observed, and this happens in some complicated applications where we directly observe some variavles $x_1, \cdots, x_n$, but some other variables (a set of unobserved latent data) $z_1, \cdots, z_n$ are unobserved. This situation will be complicated, because of the missing variables $z_1, \cdots, z_n$, we cannot estimate the parameter θ and without estimating θ we cannot consequently infer what the value of z

may be. The EM algorithm depends on the complete log-likelihood:

$$\ell_c(\theta; \mathbf{x}, z) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log(p_k f(\mathbf{x}_i; \theta_k)), \tag{1.4.4}$$

where $z_{ik} = 1$ if the i-th observation belongs to the k-th cluster and $z_{ik} = 0$ otherwise. So, in the EM algorithm we start with an initial guess of the model parameters θ and derive the expected values of the missing variables $z_1, \cdots, z_n$. Based on this initial expected values, we can maximize the likelihood w.r.t. θ, and again based on this initial estimated value of θ, we can derive the new expected values of $z_1, \cdots, z_n$ and so on. So, we assume that one of both z and θ is known in each iteration, and we repeat this iterative process until the likelihood cannot be increased anymore.

The expectation-maximization algorithm is considered one of the basic tools in model-based clustering. It has been extensively used in the literature for inferring the mixture models and determining the partitions. The EM algorithm iteratively maximizes the conditional expectation of the complete log-likelihood through two steps. The first step is the expectation step (E-step), which computes the expectation of the complete log-likelihood conditionally to the current value of the parameter set. In other words, we calculate the expected value of the log likelihood function, with respect to the conditional distribution of $\mathbf{z}$ given $\mathbf{x}$ under the current estimate of the parameters θ. The second step is the maximization step (M-step), which maximizes the expectation of the complete log-likelihood over the parameter to provide an update for the parameter set. Once a stopping criterion is satisfied, the algorithm would stop after many iterations, and the partition of the data can be deduced from the posterior probabilities by using the maximum a posteriori (MAP) rule, where we can assign the observation x_i to the group with the highest posterior probability.

Model-based clustering of high-dimensional data is considered one of the active topics

in clustering analysis. Many approaches related to the clustering of high-dimensional have been introduced such as, the dimension reduction methods, regularization methods, constrained and parsimonious methods and subspace clustering methods. For more details about these methods, the reader is referred to (Bouveyron and Brunet-Saumard, 2014).

We apply now the model-based clustering algorithm of Fraley and Raftery (2002), which was implemented in the R package `mclust`, on the breakfast cereals data. Fraley and Raftery (2003) assumed a family of different models of the parameterization of the Gaussian mixture models (parsimonious models), introduced earlier by Banfield and Raftery (1993), to be fitted in the EM phase of clustering. The models' names and details of the number of free parameters to estimate for the parsimonious Gaussian mixture models with K components and p variables, are available in Table 2 of Bouveyron and Brunet-Saumard (2014) and in the function `mclustModelNames(model)` in the package `mclust`. For example, EII refers to a spherical equal volume model, which assumes that the covariance matrices of each class are equal and spherical such that $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}_p$, for $k = 1, \ldots, K$ and with $\sigma^2 \in \mathbb{R}$, while VEI refers to a diagonal varying volume and equal shape model which assumes that the covariance matrices of each class are different.

Figure 1.8 shows the optimal model according to BIC for EM initialized by hierarchical clustering for parameterized Gaussian mixture models. As we can see, the best model according to BIC values is a diagonal equal shape model (VEI) with 4 clusters, where the maximum BIC value was for this model with (BIC= -2158.89). However, the algorithm failed to give the right number of clusters based on the Bayesian information criterion(BIC).

1.5 Functional Data Clustering

Recently, functional data analysis (FDA) has become one of the most important and active topics in statistics. This is due to its ability to represent the sequences of individual

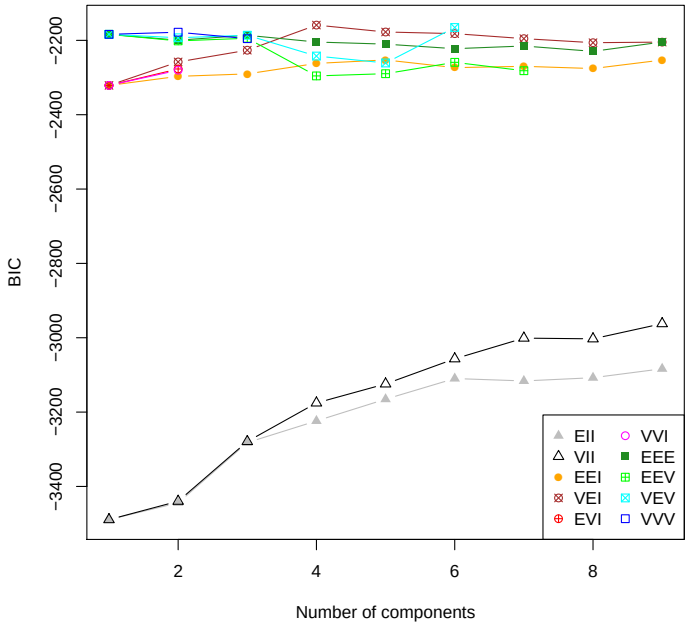


Figure 1.8: Example 1.2.1: BIC plot based on model-based clustering (mclust) for the breakfast cereals data.

discretely observed data as functions and analyze them as single entities. The greatest advantage of FDA is the additional information that can be extracted from the underlying functions, derivatives and primitives. In the functional data, the data are usually sampled along a continuum, and the random variables take values into an infinite dimensional space such as a space of functions defined on some set $\mathcal{T}$, where $\mathcal{T} \subset \mathbb{R}$ could be time interval. For example, the stochastic process $X = \{X(t); t \in \mathcal{T}\}$; where $\mathcal{T} \subset \mathbb{R}$ is a good example of the functional data. Functional data can exist as univariate or multivariate. Ramsay and Silverman (2005) and Ferraty and Vieu (2006) have been considered the classic and popular references for the functional data analysis. They proposed different definitions and examples for the functional data.

In the FDA, it is important to distinguish between two different kinds of the sampled curves. The first one is the regularly sampled curves, where the evaluation points

$t \subset \mathcal{T}$ are supposed to be fixed for each curve with the same length and knots. For example the discrete observations X_{ij} of each sample path $\mathbf{X}_i(t)$ at a finite set of knots $\{t_{ij} : j = 1, \ldots, m\}; i = 1, \ldots, n$ is considered a regularly sampled curves, where the curves' length (m) is fixed among the different functions. The second one is the irregularly sampled curves, where the evaluation points are assumed to be different and each curve has its own length based on the number of knots that represent the discrete observations in the sampled curve. In other words, we have discrete observations X_{ij} of each sample path $\mathbf{X}_i(t)$ at a finite set of knots $\{t_{ij} : j = 1, \ldots, m_i\}; i = 1, \ldots, n$, where the curves' length (m_i) changes for each function.

Functional data are observed discretely. Suppose we have discrete observations X_{ij} of each sample path $X_i(t)$ at a finite set of knots $\{t_{ij} : j = 1, \ldots, m_i\}$, then each functional observation of a single curve $X_i(t)$ consists of m_i pairs (X_{ij}, t_{ij}), where X_{ij} is the i-th observation of the function $X_i(t)$ at time t_{ij}. Modeling of functional data depends on whether sample curves are observed without error or with error. For the first case, the functional predictor will be,

$$X_{ij} = X_i(t_{ij}); \qquad j = 1, \ldots, m_i, \tag{1.5.1}$$

and for the second case, when the sample curves are observed with error e_{ij}, we formally write:

$$X_{ij} = X_i(t_{ij}) + e_{ij}; \qquad j = 1, \ldots, m_i. \tag{1.5.2}$$

Nowadays, many researchers from diversified areas use FDA in order to study different applications in many scientific fields such as medical studies, weather researches, phonetics, economics, social science and stock market. The most widely used example in FDA is the Berkeley Growth Study (Tuddenham and Snyder, 1954). It has been extensively studied in Ramsay and Silverman (2005). We give more extensive and detailed analysis

of growth data in Chapter 5 in the context of functional data clustering. Other examples for using the clustering analysis in the functional data is clustering gene expression time series, electrocardiograms (ECG) for the cardiac pathology, and the weather forecast.

Many functional data clustering methods have been proposed in the literature. An classification of the different functional data clustering approaches is given by Jacques and Preda (2014). It classifies the functional data clustering approaches into four groups, the raw-data methods, filtering methods, adaptive methods and distance-based methods. An example of the raw-data methods is the work by Boullé (2012). Moreover, many other methods that have been introduced for the multivariate case, have been considered as raw-data methods, more details are given in Chapter 5. On the other hand, examples of the filtering methods based on the functional principle components analysis (FPCA) and the spline coefficients in literature are the work that have been introduced by Peng and Müller (2008), Abraham et al. (2003), Rossi et al. (2004) and Kayano et al. (2010). Different adaptive methods based on the probabilistic model of the basis expansion coefficients have been introduced by James and Sugar (2003), Heard et al. (2006), Ray and Mallick (2006), Samé et al. (2011) and Giacofci et al. (2012). In addition, other adaptive methods based on the probabilistic model of the FPCA scores have been proposed by Chiou and Li (2007), Delaige and Hall (2010), Bouveyron and Jacques (2011), and Jacques and Preda (2013). For the distance-based methods, many publications introduced different methods like Cuesta-Albertos and Fraiman (2000), Tarpey and Kinateder (2003), Ferraty and Vieu (2006), Tokushige et al. (2007), and Ieva et al. (2012). More extensive review of the functional data clustering methods is given in Chapter 5.

1.6 Determination of the Number of Clusters

A problem with cluster analysis is to determine the optimal number of clusters in the multivariate data. Most of the previous methods that have been discussed above, require

the number of clusters to be fixed a priori and suppose that it is known. In fact, determining the suitable number of clusters has been considered one of the important clustering analysis topics, and it has been investigated extensively over the last several decades by many publications.

Some clustering methods; like K-means, K-medoids, linkage-based methods and expectation - maximization algorithm; assume an initial number of clusters depending only on the investigator's experience. However, many hierarchical methods assume that the number of clusters K is implicitly defined by cutting a hierarchical clustering tree at a given height, and in the most of exploratory applications the number of clusters K is unknown. So, particular attention should paid to the methods that can determine the number of clusters. The main message of this study is to propose novel methods and algorithms that can be utilized to determine the suitable number of clusters. Practically, determining the correct number of clusters depends on the experience of the investigator and the nature of the study. Statistically, there are many attempts and algorithms have been suggested in order to determine the optimal number of clusters.

While a comprehensive review of these methods is difficult simply because of the huge number of the literature involved, we try to review the most important and related literature. Over the last 40 years, a wealth of publications in this topic has been developed. Many of them have been introduced and discussed different graphical approaches and statistical algorithms to detect the number of clusters in the multivariate data. In this Section, we discuss the most important related methods.

The early works on cluster number determination methods were from Thorndike (1953), where he has introduced a graphical approach, which can determine the suitable number of clusters in the multivariate data. It depends on choosing a number of clusters which assumes that adding different cluster will not improve the modeling of the data. This method is known as the "elbow method", and it concerns with the percentage

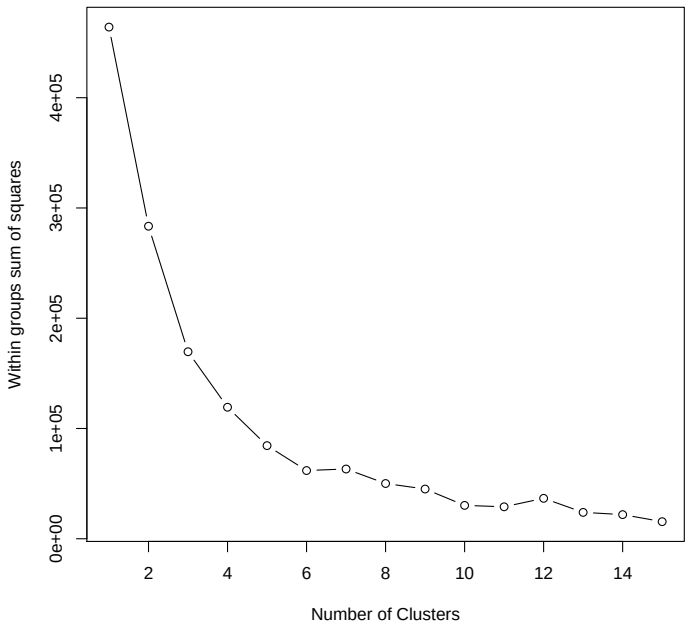


Figure 1.9: Example 1.2.1: Elbow plot for the breakfast cereals data.

of variance explained as a function of the number of clusters. Thorndike (1953) pointed out that with the increase in the number of clusters K, the average within-clusters distance will decrease. Accordingly, the number of clusters will be acceptable when sudden fluctuation happen in the curve of the percentage of variance explained by the clusters against the number of clusters (elbow criterion). Figure 1.9 shows the elbow plot for the breakfast cereals data, which wrongly suggests 2 clusters.

On the other hand, a different method has been proposed by Beale (1969). He has proposed an F statistic to test if the number of clusters k_2 is better than a different number of clusters k_1 or not, where $k_2 > k_1$. This F statistic is:

$$F(k_1, k_2) = \frac{\frac{R_{k_1} - R_{k_2}}{R_{k_2}}}{\left(\frac{n-k_1}{n-k_2}\right)\left(\left(\frac{k_2}{k_1}\right)^{\frac{2}{p}} - 1\right)}, \tag{1.6.1}$$

where $R_k = (n-k)S_k^2$, and S_k^2 is the means squared deviations for the clusters mean in the sample. Beale (1969) illustrated that we have to compare between the above F statistic and the tabulated F value at $p(n-k_2), p(k_2-k_1)$ and level of significance α. If the result of the test is significant, we would say that the number of clusters k_2 is better than the number of clusters k_1.

An important attempt that has been introduced in the same context, was for Marriott (1971). He supposed that under the equality of the variances-covariances matrices, the number of clusters can be determined when $K^2\,|W|$ becomes a minimum value, where W is the matrix of within-groups dispersion. Everitt (1977) confirmed that Marriott's method is considered one of the better ways to determine the number of clusters K.

One of the most popular techniques that can be used to determine the number of clusters has been introduced by Calinski and Harabasz (1974), and it is known as the CH index that we mentioned in (1.3.1). Alternative form can be used instead of (1.3.1) by using the trace of B and W. For a given number of clusters K, we can compute the CH index as following:

$$CH(K) = \frac{trace(B)/(K-1)}{trace(W)/(n-K)}, \tag{1.6.2}$$

where W, B are the within and between cluster variations matrices of $CH(K)$ respectively. Figure 1.10 gives the K-means partitions cascade comparison using a range of values of K based on Calinski index, for the breakfast cereals data, which wrongly suggests 10 clusters. The algorithm is implemented in the R package `vegan`.

There are also many rules and ways that can be used in this context when the mixture models with unknown number of components are considered. The most important one is the log-likelihood ratio test statistics, where we test the significance of k_1 against the significance of k_2 by using the statistic, $-2ln\lambda$, where $\lambda = L_{k_1}/L_{k_2}$, L_{k_1} is the likelihood function using k_1, and L_{k_2} is the likelihood function using k_2. Wilks (1938) proved that, under some assumptions, the test statistic $-2ln\lambda$ becomes asymptotically and probabilis-

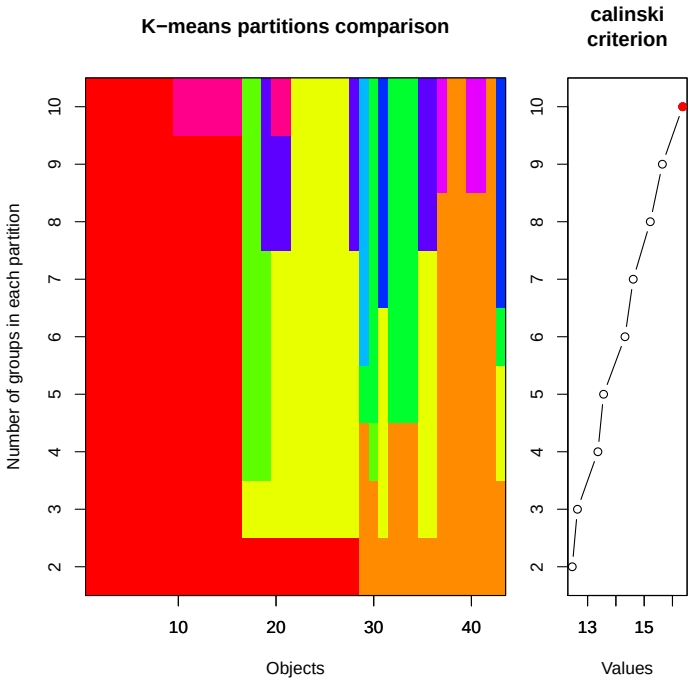


Figure 1.10: Example 1.2.1: K-means partitions cascade comparison using a range of values of K based on Calinski index for the breakfast cereals data.

tically close from χ^2 distribution with degrees of freedom equal to the difference between the number of parameters for the two tested hypotheses. Moreover, Wolfe (1971) used this test to design his algorithms (NORMIX, NORMAX), to fit the multivariate mixture normal distributions by adjusting the test statistic to become, $-2(\frac{1}{n})(n-1-p-\frac{k_2}{2})ln\lambda$. This statistic becomes also probabilistically close from χ^2 distribution with degree of freedom $2p(k_1-k_2)$. For more comprehensive details about the other methods, the reader is referred to Lennington and Flake (1975); Engelman and Hartigan (1969) and Everitt et al. (2011).

Along the line of all the previous methods, there are many other attempts and algorithms have been suggested in order to determine the optimal number of clusters. For instance, Duda and Hart (1973), Baker and Hubert (1975), Mojena (1977), Davies and Bouldin (1979), Milligan (1980, 1981), Milligan and Cooper (1985), Overall and Magee

(1992) and Gordon (1998) have suggested many of indices and criteria in order to determine the suitable number of clusters.

Gap statistic is considered one of the popular methods that can be used to determine the number of clusters. It has been introduced by Tibshirani et al. (2001). The idea of this statistic is to compare the observed within-cluster variation $W(K)$ with the within-cluster variation for points distributed uniformly, $W_{unif}(K)$, which is computed by simulation. After that we have to compute the standard error $s(K)$ of $\log W_{unif}(K)$ over the simulations. The gap for K clusters is defined as,

$$Gap(K) = \log W(K) - \log W_{unif}(K). \tag{1.6.3}$$

Then we choose K as following,

$$\widehat{K} = \min \{K \in \{1, ..., K_{max}\} : Gap(K) \geqslant Gap(K+1) - s(K+1)\}. \tag{1.6.4}$$

Figure 1.11 shows the gap plot for the breakfast cereals data. In the gap plot, the highest point of gap value refers to the suggested number of clusters. It can be clearly seen that, the plot suggests 10 clusters, however the real number is 3.

Sugar and James (2003) introduced an information theoretic approach which can find K in a dataset by applying the rate distortion theory. This method chooses the number of clusters that maximizes efficiency while minimizing error by information theoretic standards.

Another important clustering concept that has been introduced by Rousseuw (1987) is the silhouette. Silhouette is a method of interpretation and validation of the clusters. Moreover, it can be utilized in choosing the number of clusters K. This method tells us how well each object lies within its cluster by using some graphical tool. Let $a(i)$ be the average dissimilarity of the item i with all other data within the same cluster, and let $b(i)$

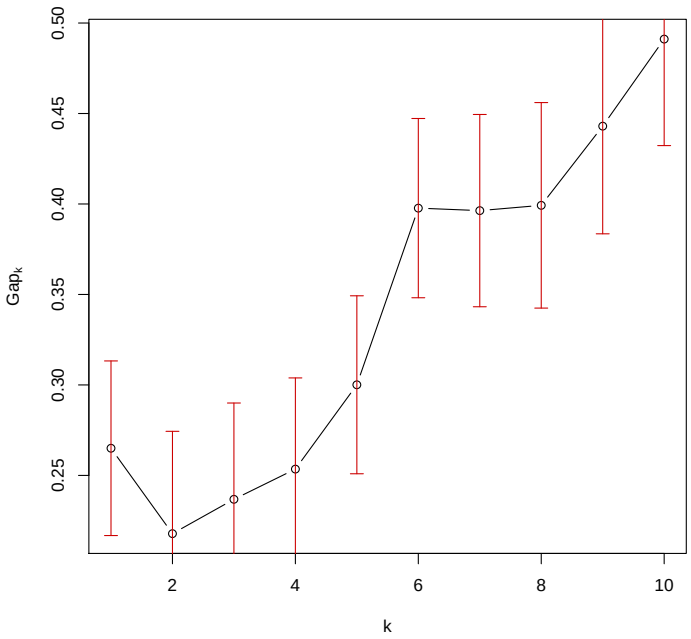


Figure 1.11: Example 1.2.1: Gap plot for the breakfast cereals data.

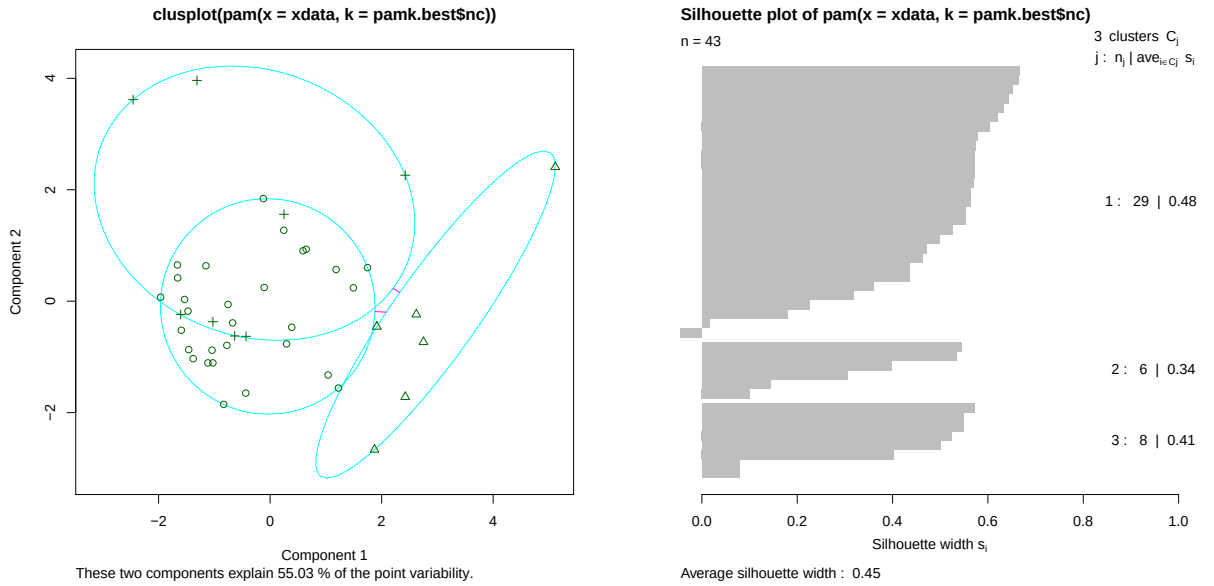
be the lowest average dissimilarity of i to any other cluster which i is not a member, then we can write the silhouette function as following:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{1.6.5}$$

which can be written as:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases} \tag{1.6.6}$$

It can be clearly seen that, $-1 \leq s(i) \leq 1$, which means when the value of silhouette is close to 1 that implies the point (datum) is in an appropriate cluster, while when the



(a) Bivariate cluster plot (clusplot)

(b) Silhouette plot

Figure 1.12: Example 1.2.1: Bivariate cluster plot (clusplot) and Silhouette plot based on the partitioning around medoids clustering (PAM) for the breakfast cereals data.

silhouette value is close to -1 that implies the datum is in the wrong cluster. Figure 1.12 gives the bivariate cluster plot (clusplot) and Silhouette plot based on the partitioning around medoids clustering (PAM) for the breakfast cereals data, suggesting the right number of clusters. In the clusplot, a bivariate plot visualizing the clustering assignments of the data is given. All observation assignments are marked by symbols in the plot, and around each cluster an ellipse is drawn. On the other hand, the silhouette plot displays how close each point in some cluster is to other points in the neighboring clusters. Moreover, the silhouette plot shows the number of horizontal lines for each cluster. In the right hand column of the plot we can see the mean similarity of each cluster to its own cluster minus the mean similarity to the next most similar cluster, and the average silhouette width.

The previous example and plots are in fact a good example to show two important

facts in clustering analysis. The first one is there is no completely acceptable solution to the optimal number of clusters due to high complexity of real data sets. The second fact is that different methods give different numbers of clusters.

On the other hand, there is another set of methods for determining the number of clusters that depend on the information criterion approach. In literature the information criteria; such as the Akaike information criterion (AIC), Bayesian information criterion (BIC), integrated complete-data likelihood (ICL), or the deviance information criterion (DIC); have been considered very important tools to determine the number of clusters. For instance, in the model-based clustering, BIC is considered an important criterion as shown in Figure 1.8. Moreover, we can also use the cross-validation as a tool that helps to analyze the number of clusters. The idea of using the cross-validation is that, we partition the data into two types of sets, the first one is the test set and the second is the training set. For each set, some goal function (like the sum of the squared distances) should be calculated and averaged for each alternative number of clusters. The suitable number of clusters then is that minimizes the test set errors. Other methods like kernel matrix and the detection of multivariate outliers can also be utilized in order to determine the optimal number of clusters.

One of the common ways to determine the number of clusters is to detect the presence of outliers in a sample of multivariate data. Many of the outlier detection methods have been later considered as clustering methods. Recently, many publications have been introduced and discussed some graphical approaches and statistical algorithms to detect the outliers and consequently determining the expected number of clusters. The use of robust methods for the detection of outliers was introduced by Rousseeuw and Leroy (1987). This was followed by other publications such Rousseeuw and van Zomeren (1990), Cook and Hawkins (1990), Hadi (1992), Woodruff and Rocke (1993), Hawkins and Simonoff (1993), Atkinson and Mulira (1993), Atkinson (1993) and Riani and Cerioli

(1999) who described variety of algorithms and plots for the detection of multivariate outliers. It is well known that the detection of outliers is usually used in two common cases, the first is for the regression problem and the second is for the multivariate data. We focus our interest in the last one since our methodology is related to the clustering of multivariate data. The minimum volume ellipsoid (MVE) method is usually used for the detection of masked multivariate outliers. Many publications have been proposed regarding to MVE. For instance, Cook and Hawkins (1990) showed that the MVE-based method can detect many of outliers, but the changes in the algorithm are effected on the detection. Woodruff and Rocke (1993) compared a variety of algorithms for calculation of the MVE. Along the line of Woodruff and Rocke (1993), Hadi (1992) introduced a similar search for the MVE and used a robustly estimated starting point for his single search. Stalactite plot that can be used as a detector of multivariate outliers has been proposed by Atkinson and Mulira (1993), where the pattern of multivariate outliers can be detected during each search. Atkinson (1993) described the fast forward algorithm for multivariate outliers, while Atkinson (1994) improved it to get a fast very robust method for the detection of multiple outliers for both regression and multivariate case. In our paper (Baragilly and Chakraborty, 2016) we have proposed a new forward search methodology based on nonparametric multivariate spatial rank functions and it is robust in terms of determining the number of clusters by the data. We illustrated in this paper that the proposed algorithm is robust to the choice of initial subsample and it performs well in different mixture multivariate distributions. We also proposed a modified algorithm based on the volume of central rank regions. Our numerical examples show that it produces the best results under heavy tailed mixture distributions with elliptic symmetry and it outperforms the forward search based on Mahalanobis distances for non-normal mixture distributions.

Mahalanobis distance has an important role in the detection of multivariate outliers'

methods, since such that methods based on the Mahalanobis distance has been extensively studied in many publications. They assumed that the data follow a multivariate normal distribution and consequently the distribution of the Mahalanobis distance behaves as a Chi-Square distribution for a large number of instances. Examples of the outlier detection methods based on the Mahalanobis distance are Rousseeuw and Leroy (1987), Hadi (1992), Atkinson (1993, 1994); and Woodruff and Rocke (1993).

It is very important to mention in this context that, some of these methods have been later considered as clustering methods. For instance, the forward search approach introduced by Hadi (1992) and Atkinson (1994) has been used as a clustering method in Atkinson et al. (2004), where they introduced many applications of the forward search in the analysis of multivariate data. Moreover, Atkinson et al. (2006) used the forward search based on Mahalanobis distances in combination with envelopes, to give tests for multiple outliers in multivariate data. Atkinson and Riani (2007) presented using of the forward search as exploratory tools for clustering multivariate data. Another example is Jörnsten (2004), where she proposed an important clustering algorithm based on the L1 data depth which depends on an outlier detection method. Her algorithm, which is known as DDclust, finds and selects the number of clusters which maximizes a combination between average silhouette width and average relative data depth. The non-parametric method that she proposed is based on the simple concept of data depth. She also discussed the relative data depth plot which is considered a convenient visualization and validation tool. Third example is given in Chen et al. (2009), where they introduced another important outlier detection method based on the kernelized spatial depth function.

1.7 Outline of the Thesis

Firstly, we illustrate the methodology and objectives of study, and then we discuss the outline of the thesis. Our objective is to propose some methods based on different notions

of multivariate ranks in order to propose new algorithms that can be helpful in determining the number of clusters in the high dimensional data. One of our main interests is to propose a novel forward search algorithm based on some rank functions, which can be used to determine the number of clusters in multivariate data. The traditional forward search method is the subject of Chapter 2, where we discuss the performance of the forward search algorithm based on Mahalanobis distances, and its problems and deficiencies with some numerical examples.

Like many other Mahalanobis distance based methods, the forward search based on Mahalanobis distances cannot be correctly applied to heavy tailed mixture distributions, asymmetric distributions and more generally, to distributions that depart from the elliptical symmetry assumption. In Chapter 3 we propose a new forward search methodology based on spatial ranks, where clusters are grown with one data point at a time sequentially, using spatial ranks with respect to the points already in the subsample. The algorithm starts from a randomly chosen initial subsample. We illustrate with simulated data that the proposed algorithm is robust to the choice of initial subsample and it performs well in different mixture multivariate distributions. We also propose a modified algorithm based on the volume of central rank regions. Our numerical examples show that it produces the best results under elliptic symmetry. Chapter 3 gives more details about the forward search method based on the spatial ranks and the volume of central rank regions with different numerical examples. The notion of central rank regions and volume of central rank regions are discussed in Section 3.4. Section 3.5 reveals the forward search based on volume of central rank regions. Section 3.6 gives the simulation envelope algorithm and the entry plot based on the multivariate ranks. In Section 3.7, we demonstrate the results of some real data sets compared to some standard methods.

The methodology of Chapter 4 is to propose new clustering method based on different weighted spatial rank (WSR) functions. In Section 4.2, we give a brief review of the

parametric and nonparametric weights functions, where we consider different kernel and robust weights. Section 4.3 introduces the proposed weighted spatial rank functions with some numerical examples and comparisons with other standard parametric and nonparametric methods. In Section 4.4, we propose a confirmatory classifier based on weighted spatial ranks that can be used to properly assign the observations to specific cluster. Section 4.5 demonstrates the weighted rank based clustering algorithm. Finally, in Section 4.6, we give some numerical examples based on both simulated and real datasets to show the performance of the proposed algorithm.

In Chapter 5, there is a large body of work on using the ordinary and weighted spatial ranks as functional data clustering approaches. We propose two different clustering methods for functional data. The first method is an extension to the forward search based on spatial ranks that has been introduced in Chapter three, and the second method is considered an extension to the weighted spatial ranks WSR method that has been introduced in Chapter 4. The proposed methods can be used to determine the number of clusters in the functional data, and to assign each curve to its cluster. Chapter 5 is organized as follows. Section 5.2 gives a review of the important existing literature on the functional data clustering methods giving numerical examples and comparisons. In Section 5.3, we discuss the curse of dimensionality in the traditional forward search method and the ability of using the forward search based on functional spatial ranks. In Section 5.4, we propose the functional data clustering based on spatial ranks. Numerical results based on simulation and real data, and other relevant discussions are contained in succeeding subsections. Finally, in Section 5.5, we propose the functional data clustering based on weighted spatial ranks with some numerical examples and comparisons with the other functional data clustering methods. The results show that the two proposed methods give a quite reasonable clustering analysis.

Chapter 2

The Forward Search Algorithm

2.1 Introduction

Determining the optimal number of clusters has become one of the most important topics in cluster analysis. Over the last 40 years, a wealth of publications has been developed for this point. As shown in Section 1.6, there are many methods to use for determining the number of clusters. The forward search method is one of these methods that depends on detecting the presence of outliers in a sample of multivariate data and consequently determining the expected number of clusters. It is worth mentioning here that the forward search method depends on the graphical presentations to provide plots not only to detect the clusters but also to determine the membership of the observations. We can define the forward search approach as a graphics rich approach that leads to the formal detection of outliers and consequently determining the expected number of clusters in the multivariate data.

There are two major concerns in the existing forward search literature; using the Mahalanobis distances as distance measure to grow the cluster size starting from a randomly chosen initial subsample, and the robust parameter estimates based on increasing the subset size during the search. The traditional forward search approach based on Maha-

lanobis distances have been introduced by Hadi (1992), and Hadi and Simonoff (1993). They considered a forward search, which terminates when the subset size m is the median of the number of observations, while a similar method used by Atkinson and Mulira (1993), Atkinson (1994) continues until $m = n$, the sample size. Cerioli and Riani (1999) proposed an unified approach to the exploratory analysis of spatial data which is based on a forward search algorithm. They pointed out that the search is made up of four steps, the first one is estimating the parameters, the second step is the choice of an initial subset, the third step is the progress in the forward search by monitoring the search and the fourth step is ordering the spatial data by monitoring of the statistics during the progress of the search. Atkinson et al. (2004) introduced many applications of the forward search in the analysis of multivariate data. In Atkinson et al. (2004), the forward search has been used as a clustering method. Moreover, Atkinson et al. (2006) used the forward search based on Mahalanobis distances in combination with envelopes, to give tests for multiple outliers in multivariate data, while Atkinson and Riani (2006) provided some distributional results for testing multiple outliers in regression by using the forward search. Atkinson and Riani (2007) used the forward search as an exploratory tools for clustering multivariate data. A good overview of the forward search and its applications is available in (Atkinson et al., 2004; Atkinson et al., 2010).

We use the forward plot in order to detect the clusters in the data under study. The forward plot is a plot of number of subsets with incremental size (m) versus a specific distance measure. The forward plot has been introduced by Atkinson et al. (2004). Along the line of Atkinson et al. (2004), Atkinson et al. (2006), Atkinson and Riani (2006), Atkinson and Riani (2007), Riani et al. (2009), and Atkinson et al. (2010) introduced many forward plots and shed the light on its importance in the detection of outliers and clusters. To highlight the importance of the forward plot, two important points should be noticed here. The first is that the forward plot is an important exploratory tool that

can be used in order to detect the observations that could be outliers, by plotting the distance values for each subset size during the search. The second one is that the forward plot gives more detailed information during the search, where we can know the number of clusters and the membership of each observation by plotting minimum Mahalanobis distance amongst units not included in the subset. An important feature of the forward plot is that the forward plot of the largest distance will show a sharp peak when the first outlier is included which leads to the identification of the outliers and consequently determining the number of clusters.

The main idea of a forward search algorithm is to grow the cluster size starting from an initial subset of observations based on a some kind of distance measure. All the previous literature assumed Mahalanobis distance as the distance measure to be used in the forward search procedure. It is well known that Mahalanobis distance is invariant under all nonsingular linear transformations and it also performs well with the Gaussian mixture models (GMM).

To show that Mahalanobis distance is invariant under all nonsingular linear transformations, suppose that $(x_1, x_2, ..., x_n)$ is an original data set with covariance matrix defined as, $\Sigma_x = \frac{1}{n}X^T X$, where X is an $(n \times d)$ matrix with the (i, j)th element $(x_{ij} - \overline{x}_j)$, and let A be any nonsingular $(d \times d)$ matrix applied to the original data set $(x_1, x_2, ..., x_n)$, such that $y_i = Ax_i$. Then, the Mahalanobis distance between the two points y_i and y_j equals to the Mahalanobis distance between the original two points x_i and x_j, i.e., $d_{Mahalanobis}(y_i, y_j) = d_{Mahalanobis}(x_i, x_j)$, which means that Mahalanobis distance is invariant under all nonsingular linear transformations. To prove that, recall the Mahalanobis distance function that is defined in (1.2.8), the Mahalanobis distance between y_i and y_j is obtained as,

$$d_{Mahalanobis}(y_i, y_j) = \sqrt{(y_i - y_j)\Sigma_y^{-1}(y_i - y_j)^T}$$

where the covariance matrix, Σ_y, can be obtained as, $\Sigma_y = \frac{1}{n}Y^TY = \frac{1}{n}(XA^T)^T(XA^T)$ where Y is an $(n \times d)$ matrix with the (i,j)th element $(y_{ij} - \overline{y}_j)$, now the Mahalanobis distance between y_i and y_j can be written as:

$$d_{Mahalanobis}(y_i, y_j) = \sqrt{(y_i - y_j)(\frac{1}{n}Y^TY)^{-1}(y_i - y_j)^T}$$

$$= \sqrt{(x_i - x_j)A^T(\frac{1}{n}(XA^T)^T(XA^T))^{-1}A(x_i - x_j)^T}$$

$$= \sqrt{(x_i - x_j)(\frac{1}{n}X^TX)^{-1}(x_i - x_j)^T} = \sqrt{(x_i - x_j)\Sigma_x^{-1}(x_i - x_j)^T} = d_{Mahalanobis}(x_i, x_j).$$

which shows that Mahalanobis distance is invariant under nonsingular linear transformations.

However, Mahalanobis distances cannot be correctly applied to asymmetric distributions and more generally to distributions, which depart from the elliptical symmetry assumptions. In this Chapter, we illustrate with numerical examples some of these cases, considering some heavy tailed distributions like Laplace and student's t-distributions.

2.2 Forward Search Algorithm

Suppose that $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ is a random sample with distribution F, then the squared Mahalanobis distance for the d-dimensional i-th observation is defined as:

$$d_i^2 = \{\mathbf{X}_i - \widehat{\boldsymbol{\mu}}\}^T \widehat{\boldsymbol{\Sigma}}^{-1} \{\mathbf{X}_i - \widehat{\boldsymbol{\mu}}\}, \tag{2.2.1}$$

where $\widehat{\boldsymbol{\mu}}$ and $\boldsymbol{\Sigma}$ are the unbiased moment estimators of the mean vector and covariance matrix of the d-dimensional observations $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ respectively. Atkinson et al. (2004) showed that in the forward search the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are estimated from a subset $S(m)$ from the n observations. The parameters estimates can be given by:

$$\widehat{\boldsymbol{\mu}}\,(m) = \frac{1}{m} \sum_{i \in S(m)} \mathbf{X}_i, \tag{2.2.2}$$

and

$$\widehat{\boldsymbol{\Sigma}}(m) = \frac{1}{m-1} \sum_{i \in S(m)} \left(\mathbf{X}_i - \widehat{\boldsymbol{\mu}}(m)\right) \left(\mathbf{X}_i - \widehat{\boldsymbol{\mu}}(m)\right)^{\top}. \tag{2.2.3}$$

So for every subset $S(m)$ we get n squared Mahalanobis distances $d_i^2(m)$ such that,

$$d_i^2(m) = \left\{\mathbf{X}_i - \widehat{\boldsymbol{\mu}}(m)\right\}^{\top} \widehat{\boldsymbol{\Sigma}}^{-1}(m) \left\{\mathbf{X}_i - \widehat{\boldsymbol{\mu}}(m)\right\}; \qquad i = 1, \ldots, n. \tag{2.2.4}$$

The steps of the forward search algorithm (Atkinson et al., 2004; Atkinson and Riani, 2007) are:

1. In order to start the search, we need to choose an initial subset. Suppose that $S^*(m)$ is the initial subset with $m = d + 1$, then one search can be run from this starting point.

2. Now we need to add observations such that the subset $S(m)$ grows in size during the search. We can obtain n squared Mahalanobis distances $d_i^2(m)$ as shown in the last equation from the subset $S(m)$ that consists of m observations, $m_0 \leqslant m \leqslant n - 1$. To determine the next subset, we need to order the squared distances and take the observations corresponding to the $m + 1$ smallest as the new subset $S(m + 1)$. Usually this process augments the subset by one observation, but sometimes two or more observations enter as one or more leave. This is the same case when one cluster is completely fitted and observations from a second cluster have to be included as the search progresses and m increases.

3. In this step we detect the outliers by examining the minimum Mahalanobis distance amongst observations not in the subset. In the case that the observation is an

outlier relative to the other m observations, the distance will be large compared to the maximum Mahalanobis distance of observations in the subset. Consequently, all other observations not in the subset will have distances greater than $d_{min}(m)$, where $d_{min}(m) = \min d_i(m);\ i \notin S(m)$, and will therefore also be outliers.

4. A forward plot of the Mahalanobis distances can be obtained by plotting the minimum Mahalanobis distances $d_{min}(m)$ against the corresponding subset sizes (m).

2.3 Some Numerical Examples

In this Section, we applied a series of simulated cases which can show the performance of the forward search based on Mahalanobis distances especially when we consider the Laplace and Student's t distributions. We generated data from either spherically or elliptically symmetric data (correlated or uncorrelated variables). Full analysis of the performance of the forward search based on Mahalanobis distances is given in this Section.

2.3.1 Example 1: Bivariate Mixture Distributions with Uncorrelated Variables

In the first example, we consider three bivariate mixture distributions with spherical symmetry. We suppose that the scale matrix is an identity matrix i.e. there is no correlation among the variables. The three bivariate mixture distributions, namely, multivariate normal, multivariate Laplace and multivariate t with 3 degrees of freedom. The mixing proportion p is taken to be 0.3. In all three cases considered, we generate samples of $n = 100$ observations and produce forward search plots with $k = 100$ randomly chosen initial subsets for each as considered in Atkinson and Riani (2007).

For mixture normal distribution, we take $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ as a random sample from bivariate mixture normal distribution, $p.N_2(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + (1-p).N_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}_1 = (0,0)^\top$, $\boldsymbol{\mu}_2 = (5,5)^\top$, $\boldsymbol{\Sigma} = \boldsymbol{I}_2$ and $p = 0.3$.

For the second case, we consider multivariate Laplace distribution, $L_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with the probability density function (2.3.1), and consider a random sample from the bivariate mixture Laplace distribution, $p.L_2(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + (1-p).L_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$ with $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\boldsymbol{\Sigma}$ and p as before, where a multivariate Laplace distribution is defined as follows:

Definition 2.3.1 *: Multivariate Laplace Distribution:*
Suppose $\mathbf{X} = (x_1, ..., x_n)^T$ *be an* $n * d$ *dataset, where* d *is the number of variables and* n *is the number of observations, i.e.* $\mathbf{X}$ *is a d-variate random variable distributed as multivariate Laplace with mean* $\boldsymbol{\mu}$ *and covariance matrix* $\boldsymbol{\Sigma}$*, then the density function* $f(x)$ *of* $\mathbf{X}$ *is given by:*

$$f(\mathbf{X}) \propto \frac{1}{|\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{-\sqrt{(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}} \tag{2.3.1}$$

For the third case, we consider the multivariate Student's t-distribution with ν degrees of freedom, $t_d(\nu; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with the probability density function (2.3.2), and consider a random sample from bivariate mixture t-distribution with $\nu = 3$ degrees of freedom, $p.t_2(3; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + (1-p).t_2(3; \boldsymbol{\mu}_2, \boldsymbol{\Sigma})$, with $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}$ as before, and multivariate t distribution as defined below:

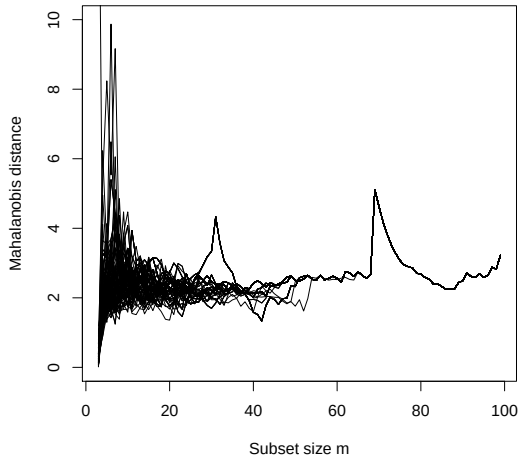
Definition 2.3.2 *: Multivariate Student's t-Distribution:*
Suppose that y *and* u *are two independent variables and distributed as* $N(0, \boldsymbol{\Sigma})$ *and* χ^2_ν *(i.e. multivariate normal and chi-squared distributions) respectively, where* ν *is the degree of freedom of chi square distribution, the covariance* $\boldsymbol{\Sigma}$*, is a* $d \times d$ *matrix, and* $y\sqrt{\nu/u} = x - \boldsymbol{\mu}$*, then* x *is said to be distributed as a multivariate t-distribution with degree of freedom* ν *and parameters* $\boldsymbol{\Sigma}$*,* $\boldsymbol{\mu}$ *and has the density function:*

$$f(x) = \frac{\Gamma\left[(\nu+d)/2\right]}{\Gamma(\nu/2)\nu^{d/2}\pi^{d/2}\left|\boldsymbol{\Sigma}\right|^{1/2}\left[1 + \frac{1}{\nu}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]^{(\nu+d)/2}}. \tag{2.3.2}$$

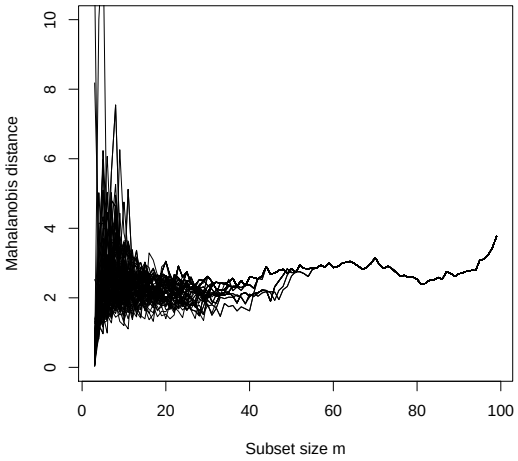
Our objective is to determine the subsets for the trajectories where there is evidence of a cluster structure. Since our generated data coming from mixture models, with mixture proportions ($p = 0.3, 1 - p = 0.7$), we expect to get a clearly common structure around subsets with size 30 and 70 respectively. Figure 2.1 is a forward plot of minimum Mahalanobis distances from 100 randomly chosen initial subsets for samples size $n = 100$ from bivariate mixture normal, Laplace and t distributions. In general the plots in Figure 2.1 initially show that there are many different values of $d_{min}(m)$ presented in many trajectories. We are interested in the subsets $S(m)$ for these trajectories where there is evidence of a cluster structure. As we can see, only for the normal distribution, there is a common structure around subsets with size 30 and 70 respectively. However, the forward plot based on Mahalanobis distance failed to give us a reasonable result for both Laplace and Student's t distributions.

In plot (a), there are two clear maxima (sharp peaks), one at $m = 30$ and the other at $m = 70$, suggesting the existence of two clusters with sizes 30 and 70. So, this plot leads to the division of the data into two clusters, such that the first one includes 30 observations and the second cluster includes 70 observations. Actually it is a good result, which shows that the forward search based on Mahalanobis distance performs well with the normal distribution, and can detect the clusters in the data.

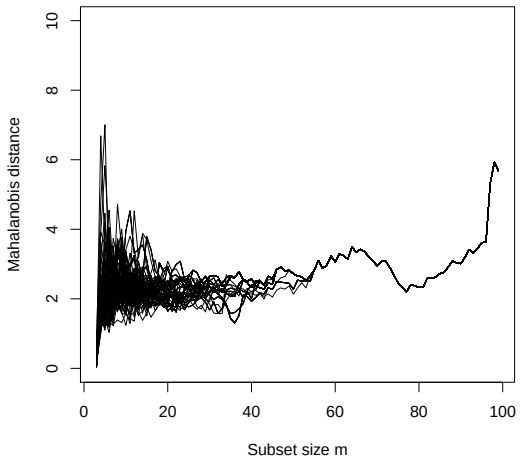
However, from plots (b) and (c) we can see that the forward plots failed to give two clear peaks around the subsets with sizes 30 and 70. Nevertheless, in plot (c) there is slight pattern around subset with size 70. Thus, we can conclude that the forward search based on Mahalanobis distances works better for the data from bivariate normal distribution with uncorrelated variables, but it poorly behaves with the data from bivariate mixture Laplace and t distributions.



(a) Bivariate normal



(b) Bivariate Laplace



(c) Bivariate t

Figure 2.1: Forward plot of minimum Mahalanobis distances from 100 randomly chosen initial subsets for sample size $n = 100$ from bivariate mixture (a) normal, (b) Laplace and (c) t distributions with uncorrelated variables.

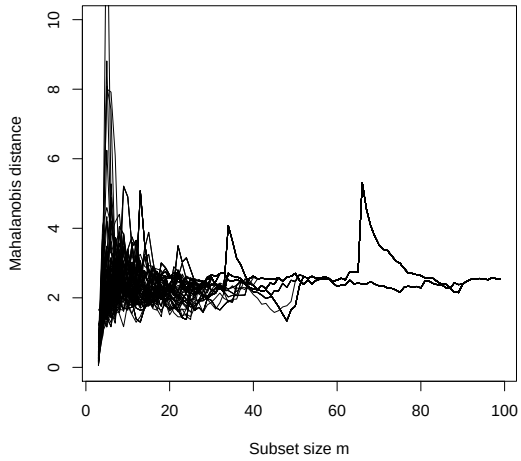
2.3.2 Example 2: Bivariate Mixture Distributions with Correlated Variables

In this example we assume the elliptic symmetry case such that there is correlation among the variables and the scale matrix is not the identity matrix ($\mathbf{\Sigma} \neq I$). The generated data, as before, are from three bivariate mixture distributions (normal, Laplace and Student's t) with ($p = 0.3$), ($k = 100$), and ($n = 100$). In the previous set-up, consider

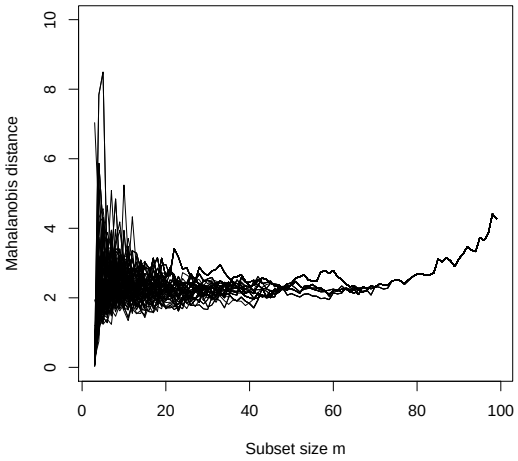
$$\mathbf{\Sigma} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}.$$

Figure 2.2 is a forward plot of minimum Mahalanobis distances from 100 random starts for sample size $n = 100$ from bivariate mixture normal, Laplace and t distributions with correlated variables. Since our target is to determine the subsets $S(m)$ for these trajectories where there is evidence of a cluster structure, only for the normal distribution, there is a common structure around subsets with size 30 and 70 respectively. Form plot (a) we can see that there is clearly common structure around subsets with size 30 and 70 respectively, where there are two clear sharp peaks, one at $m = 30$ and the other at $m = 70$, suggesting the existence of two clusters with sizes 30 and 70. This concludes that the forward search based on Mahalanobis distances also performs well with the bivariate mixture normal distribution when the variables are correlated.

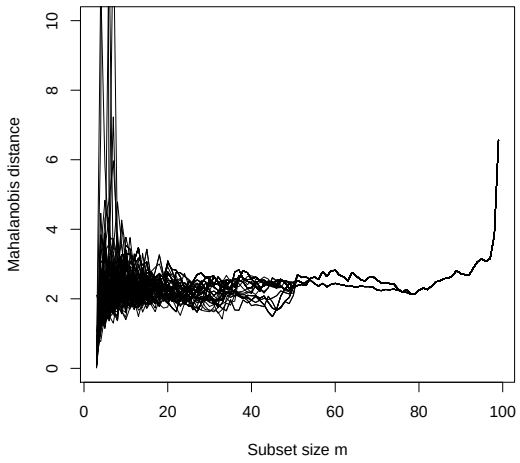
On the other hand, the forward plot based on Mahalanobis distance failed again to detect the two clusters in the data from both bivariate mixture Laplace and t distributions, where it does not give us common structure or clear peaks around subsets with size 30 and 70. Thus, we can conclude that the forward search based on Mahalanobis distances does not correctly work with data from bivariate mixture Laplace and t distributions when the variables are correlated as well.



(a) Bivariate normal



(b) Bivariate Laplace



(c) Bivariate t

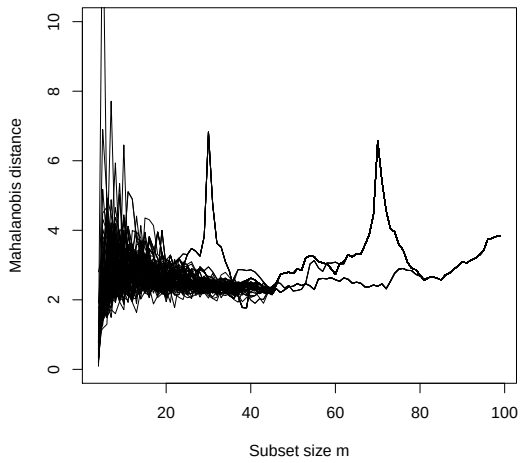
Figure 2.2: Forward plot of minimum Mahalanobis distances from 100 randomly chosen initial subsets for sample size $n = 100$ from bivariate mixture (a) normal, (b) Laplace and (c) t distributions with correlated variables.

2.3.3 Example 3: Trivariate Mixture Distributions with Uncorrelated Variables

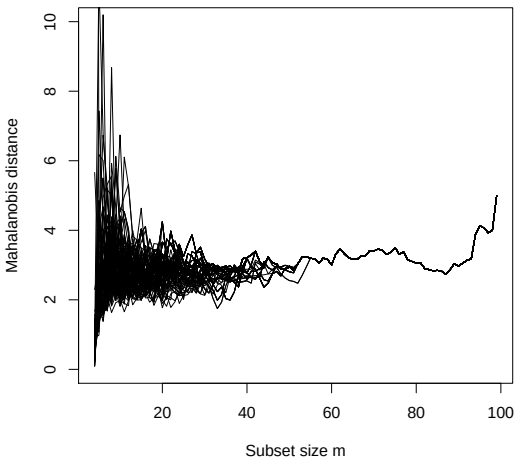
In order to check the stability of the forward search approach, we consider here a higher dimensional data. We assume in this example that the number of variables is three $d = 3$, such that the generated data come from trivariate distributions. We suppose that the data come from trivariate mixture normal, Laplace and Student's t with 3 degrees of freedom, as before, and the scale matrix is an identity matrix, i.e. there is no correlation among the variables, $(p = 0.3)$, $(k = 100)$, and $(n = 100)$.

We suppose that $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ is a random sample from trivariate mixture normal Laplace and t distributions with spherical symmetry as in the bivariate case with $\boldsymbol{\mu}_1 = (0, 0, 0)^\top$, $\boldsymbol{\mu}_2 = (5, 5, 5)^\top$, $\boldsymbol{\Sigma} = \boldsymbol{I}_3$ and $p = 0.3$.

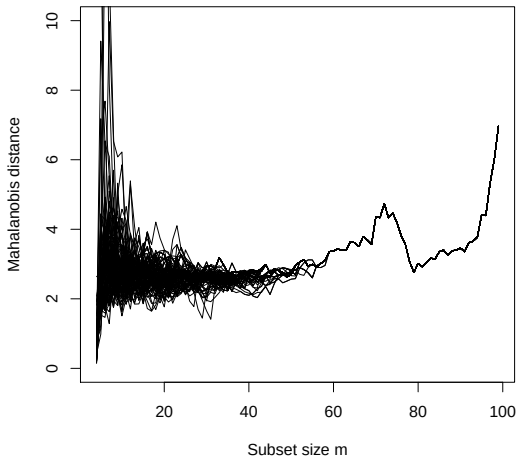
Figure 2.3 is a forward plot of minimum Mahalanobis distances from 100 random starts for sample size $n = 100$ from trivariate mixture normal, Laplace and t distributions with uncorrelated variables. Only the forward plot (a) gives an evidence for the existence of the two clusters, with two clear peaks at $m = 30$ and 70. Increasing the dimension of the data did not improve the performance of the forward plot for both trivariate mixture Laplace and t distributions. However, in plot (c) there is slight pattern around subset with size 70, which wrongly suggest the number of clusters. Thus, we can conclude that the forward search based on Mahalanobis distances performs well for the data from trivariate normal distribution with uncorrelated variables, but it poorly behaves with the data from trivariate mixture Laplace and t distributions.



(a) Trivariate normal



(b) Trivariate Laplace



(c) Trivariate t

Figure 2.3: Forward plot of minimum Mahalanobis distances from 100 randomly chosen initial subsets for sample size $n = 100$ from trivariate mixture (a) normal, (b) Laplace and (c) t distributions with uncorrelated variables.

2.3.4 Example 4: Trivariate Mixture Distributions with Correlated Variables

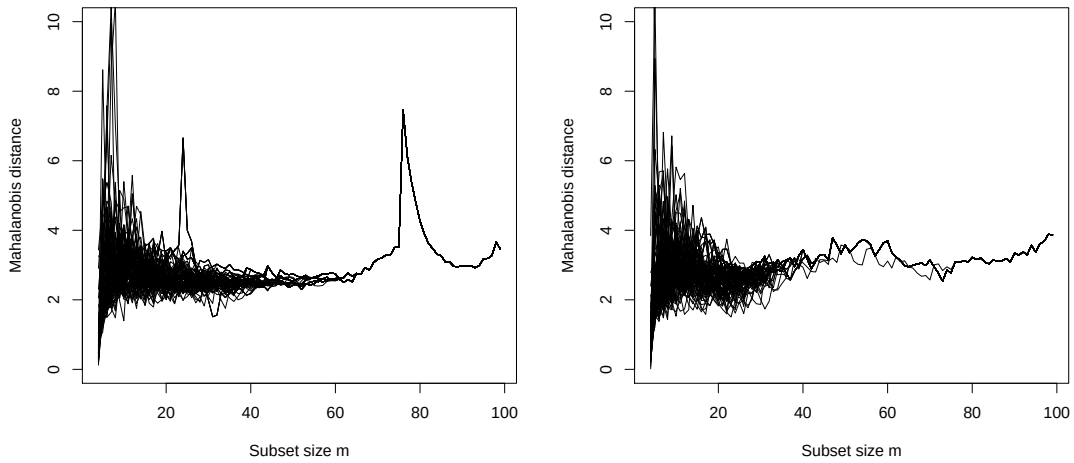
We suppose in this example that the variables are correlated with ($\rho = 0.5$). For the normal distribution, we suppose that $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ is a random sample from trivariate mixture normal, Laplace and t distributions with elliptic symmetry. In the previous set-up, consider

$$\mathbf{\Sigma} = \begin{pmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 1 \end{pmatrix}.$$

Figure 2.4 is a forward plot of minimum Mahalanobis distances from 100 random starts for samples size $n = 100$ from trivariate mixture normal, Laplace and t distributions with correlated variables. As usual, the forward plot (a) gives two clear peaks at $m = 30$ and 70, suggesting two clusters with sizes 30 and 70. Both plot (b) and (c) did not detect the clusters in the data. So, we conclude that the forward search based on Mahalanobis distances performs well for the data from trivariate normal distribution with correlated variables, but it poorly performs with the data from trivariate mixture Laplace and t distributions.

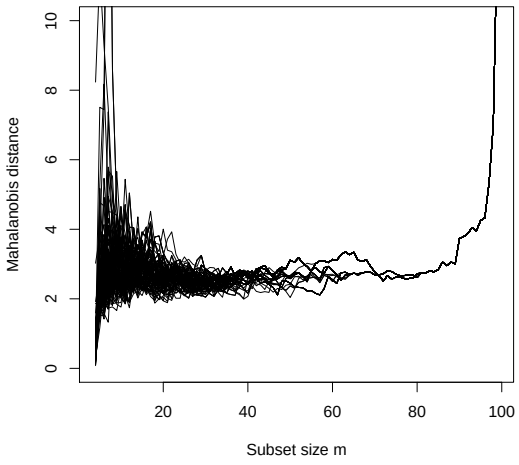
2.4 Simulation Envelope

Using the envelopes in the forward plot is helpful to provide a guide as to what kind of fluctuations are to be expected in such plots. These envelopes can be found by simulation. The simulation envelope has been studied and used in Atkinson et al. (2004), Atkinson et al. (2006), and Atkinson and Riani (2007) for establishing cluster membership in their procedure. They used the random start forward searches combined with envelope plots of forward Mahalanobis distances to detect the clusters in the data. Atkinson and Riani



(a) Trivariate normal

(b) Trivariate Laplace



(c) Trivariate t

Figure 2.4: Forward plot of minimum Mahalanobis distances from 100 randomly chosen initial subsets for sample size $n = 100$ from trivariate mixture (a) normal, (b) Laplace and (c) t distributions with correlated variables.

(2007) proposed a flexible polynomial approximations to the envelopes in the forward search based on Mahalanobis distances. The envelope is an indicator to judge if the point is lying on, or outside the normal pattern of the data and consequently it is an outlier. The envelope's line can be considered as a lower/upper limit to the points in the forward plots, such that if the point lyes outside the simulation envelope's line then it should be classified as outlier. Atkinson and Riani (2007) described the structure of the forward plot of simulation envelopes for minimum Mahalanobis distances from 1000 simulations. They pointed out that it is virtually horizontal in the center of the plot and the plot looks like a series of superimposed prows of viking long ships. The steps of the simulation envelope's algorithm are:

1. Simulate data from the empirical standard underlying distribution.

2. Choose an initial subset $S(m)$ with $m = d+1$, and start the search from this starting point.

3. Calculate the Mahalanobis distance $d_i(m)$ based on the observations in the subset $S(m)$.

4. Compute $d_{min}(m)$, where $d_{min}(m) = \min d_i(m)$; $i \notin S(m)$.

5. Grow the subset $S(m)$ to $S(m+1)$ by taking $m+1$ observations $\mathbf{X}_i$'s, which correspond to smallest $m+1$ $d_i(m)$'s. Set $m = m+1$.

6. Iterate $3-5$ until $m = n-1$

7. Iterate $1-6$ 1000 times, so for each subset size m we have 1000 values of $d_{min}(m)$.

8. Take 99% percentiles of these $d_{min}(m)$ and plot it against m to get the 99% envelope.

Figure 2.5 is a forward plot of minimum Mahalanobis distances from 100 random starts with $1\%, 50\%$ and 99% envelopes for sample size $n = 100$ from bivariate mixture normal

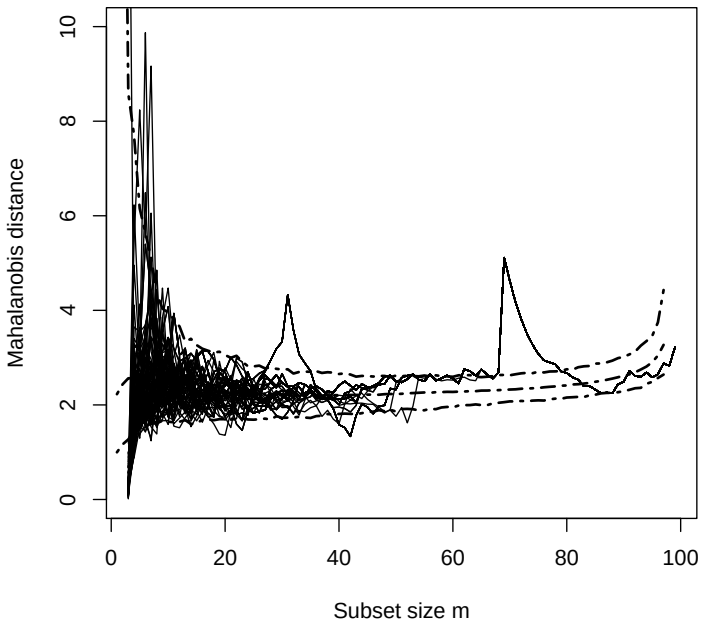


Figure 2.5: Forward plot of minimum Mahalanobis distances from 100 random starts with $1\%, 50\%$ and 99% envelopes for sample size $n = 100$ from bivariate mixture normal distribution with uncorrelated variables.

distribution with uncorrelated variables. The figure shows that the smallest and largest observations all lie on or within the simulation envelopes except those around subset with sizes 30 and 70, as they are outliers. Figure 2.6 is a forward plot of simulation envelopes for minimum Mahalanobis distances from 1000 simulations for samples sizes $n = 200, 500, 700$ and 1000. The envelope given is the 99 point of the empirical distribution of the minimum Mahalanobis distance amongst observations that are not in the subset for $d = 2, 5$ and 10. There is clearly some common structure as n and d vary, where small d at the bottom of the plot.

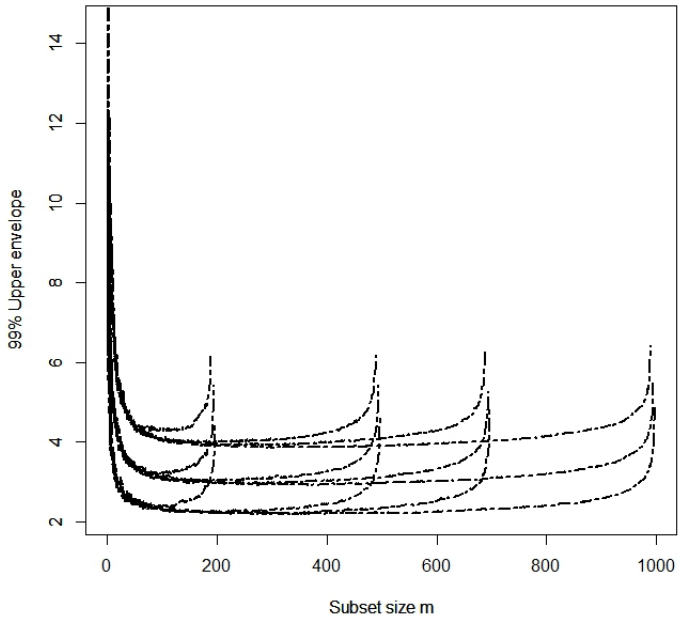


Figure 2.6: Minimum Mahalanobis distances: 99% point for $n = 200$, 500, 700 and 1000 and $d = 2$, 5 and 10. Small d at the bottom of the plot.

2.5 Entry Plot

The entry plot essentially shows which groups of observations are in the subset. The main idea of the entry plot is to order the observations based on Mahalanobis distances $d_i(m); i = 1, \ldots, n$, then plot the observations in the subset against the subset size. Thus, the dots in the plot indicate to the presence of an observation in specific subset. Accordingly, the number of dots increases towards the right of the graph when the subset size increase consequently. Atkinson et al. (2004), Atkinson et al. (2006), and Atkinson and Riani (2007) have used the entry plot as a confirmatory stage in the forward search method.

Figure 2.7 is an entry plot based on Mahalanobis distances from $m_0 = 3$ with 100 randomly chosen initial subsets for a mixture bivariate normal data set with 3 mixing

densities. The data is simulated from a mixture of 3 bivariate normal distributions,

$$p_1 N_2(\boldsymbol{\mu}_1, \boldsymbol{I}) + p_2 N_2(\boldsymbol{\mu}_2, \boldsymbol{I}) + (1 - p_1 - p_2) N_2(\boldsymbol{\mu}_3, \boldsymbol{I}), \tag{2.5.1}$$

where $\boldsymbol{\mu}_1 = (0, 4)^\top$, $\boldsymbol{\mu}_2 = (-4, -4)^\top$, $\boldsymbol{\mu}_3 = (4, -4)^\top$ and $p_1 = 0.2$, $p_2 = 0.3$. As we can see in the entry plot, the observations are ordered so that 1-20 are those from the small group with the 20 observations, 21-50 are those from the second group with the 30 observations and 51-100 coming from the big group with 50 observations. The plot at $m = 3$ shows that the initial subset includes observations from the three groups, where the first subset includes the three observations 60, 15 and 76. In the second subset of the forward search, at $m = 4$, a large amount of interchange has been happened, where the observations in this subset started to enter in the third group. Thereafter, until $m = 50$ the subset consists solely of observations from the big group. From $m = 61$ only observations from the second group started to join the subset. From $m = 81$ the observations from the small group started to join the subset. At the end of the search ($m = 100$), all the observations entered the search.

2.6 Problems

From the previous numerical examples, we can conclude that however Mahalanobis distance is invariant under all nonsingular linear transformations and it also performs well with the Gaussian mixture models (GMM), it cannot be correctly applied to asymmetric distributions and more generally to distributions, which depart from the elliptical symmetry assumptions.

According to the previous results, it was noticed that using the forward search based on Mahalanobis distances does not give an effective performance with the heavy tailed distributions. In other words, the forward plots based on Mahalanobis distances did not

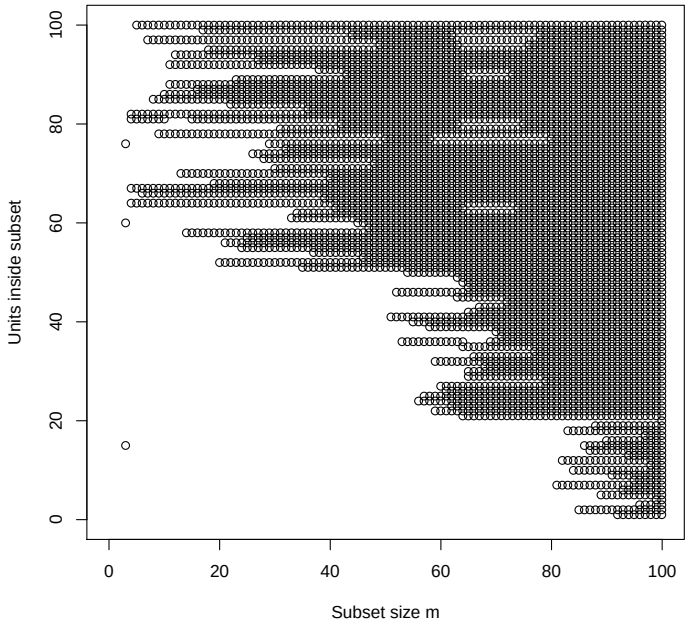


Figure 2.7: Entry plot based on Mahalanobis distances from $m_0 = 3$ with 100 randomly chosen initial subsets for a mixture bivariate normal data set with 3 mixing densities.

give us reasonable results, where they did not detect the clusters in the data coming from either bivariate or trivariate Laplace and t mixture distributions with either correlated or uncorrelated variables.

Moreover, for large number of clusters, the forward search plots may produce too many peaks and makes it very difficult visually to determine the number of clusters and the cluster sizes (Baragilly and Chakraborty, 2016). In Figure 2.8, we present an example of a forward plot based on Mahalanobis distances, where the data is simulated from a mixture of 3 bivariate normal distributions that defined in (2.5.1). With trajectories from 100 randomly chosen initial subsets, we see a clear pattern of 4 cluster sizes here, however the simulated data includes 3 clusters .

As a conclusion, we recap that, there are some deficiencies with the forward search based on Mahalanobis distances algorithm, where it is not suitable for the heavy tailed

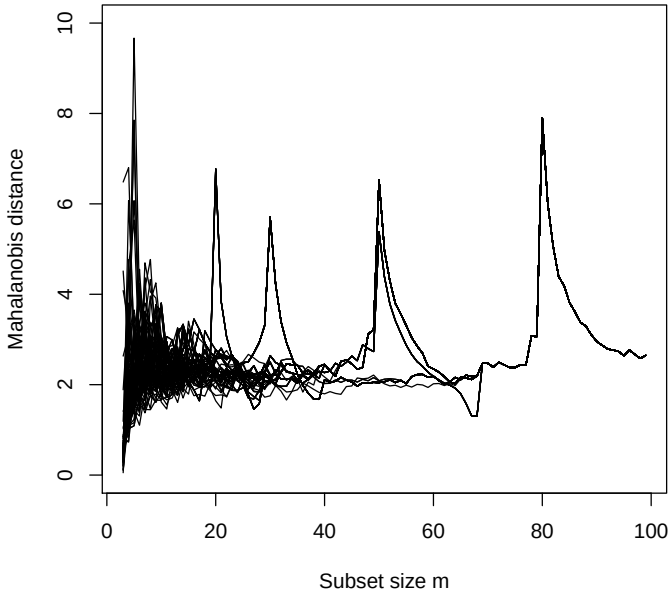


Figure 2.8: Forward plot based on Mahalanobis distances with 100 randomly chosen initial subsets for a mixture bivariate normal data set with 3 mixing densities.

distributions like multivariate Laplace, Student's t, Cauchy and Log-normal distributions under either spherical or elliptical case, where most of the previous literature assumed the multivariate normal case. Moreover, its performance is getting worse when higher dimensional data with elliptic symmetry problems are considered as we noticed in the trivariate distributions case.

In order to address this limitation, in this study, we propose a new forward search methodology based on some nonparametric methods like spatial ranks and volume of central rank regions (Chaudhuri, 1996; Serfling, 2002) to tackle the problem of heavy tailed mixture distributions with higher dimensional data. In the next Chapter we consider the forward search based on spatial ranks and the volume of central rank regions algorithms in order to improve the algorithm of the forward search.

Chapter 3

Multivariate Signs and Ranks

3.1 Introduction

In the previous Chapter, we pointed out that there are some deficiencies with the traditional forward search algorithm that does not make it suitable to be used in different types of multivariate data. According to the results in Chapter 2, the forward plots based on Mahalanobis distances failed to detect the right number of clusters in the simulated data coming from either multivariate mixture Laplace and t distributions under either spherical or elliptic symmetry.

In many statistical analyses some nonparametric multivariate methods, as spatial signs and ranks, are usually used to solve and tackle the problems in analyzing the multivariate data parameterically, and to get techniques which are less sensitive to the statistical model assumptions. For last two decades, spatial ranks are being used in analyzing multivariate data nonparametrically. They are easy to compute, and do not depend on parameter estimates of the underlying distributions, which make them robust against distributional assumptions. Koltchinskii (1997) also proved that the spatial ranks characterize a multivariate distribution. More robust results can be obtained by using the ranks instead of the original values. The novelty of this Chapter is to develop the forward search technique

by using some methods based on notions of multivariate ranks.

The remainder of the Chapter is organized as follows. In Section 3.2, we give some definitions of multivariate signs and ranks with some properties. In Section 3.3, we propose the forward search method based on spatial ranks and we give some numerical examples based on simulated data sets to show the performance of the proposed algorithm when some heavy tailed mixture distributions under the elliptic symmetry case are considered. We define the concepts of central rank regions and volume of central rank regions in Section 3.4, while the forward search method based on volume of central rank regions with some numerical examples based on simulated data are proposed in Section 3.5. Section 3.6 gives the simulation envelope algorithm and the entry plot based on the multivariate ranks. Finally, we demonstrate the results of some real data sets compared to some standard methods in Section 3.7.

3.2 Multivariate Signs and Ranks

Signs and ranks functions have been considered as important nonparametric tools in the statistical multivariate analysis. In this Section we start with discussing the notions of sign and ranks functions and their properties, then we introduce the forward search algorithm based on spatial ranks with some numerical examples.

3.2.1 Multivariate Signs

The concept of sign function is related with the possibility to order the data. We start with the univariate sign function definition, and then the multivariate sign definition can be generalized after that. Basically, the sign function of a real number x is defined as follows;

Definition 3.2.1 *: The univariate sign function:*
For a real number x, the univariate sign function can be obtained by:

$$sign(\mathbf{x}) = \begin{cases} -1, & if\ x < 0 \\ 0, & if\ x = 0 \\ 1, & if\ x > 0, \end{cases} \tag{3.2.1}$$

equivalently, we can use the form:

$$sign(x) = \begin{cases} \frac{x}{|x|}, & if\ x \neq 0 \\ 0, & if\ x = 0. \end{cases} \tag{3.2.2}$$

Definition 3.2.2 *: The multivariate sign function:*

For $\mathbf{x} \in \mathbb{R}^d$*, the multivariate spatial sign function is defined as:*

$$sign(\mathbf{x}) = \begin{cases} \frac{\mathbf{x}}{\|\mathbf{x}\|} & if\ \mathbf{x} \neq 0 \\ 0 & if\ \mathbf{x} = 0, \end{cases} \tag{3.2.3}$$

where $\|\mathbf{x}\|$ is the Euclidean norm such that; $\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \ldots + x_d^2}$. Note that this is nothing but the direction of the d-dimensional vector $\mathbf{x}$.

3.2.2 Multivariate Ranks

Using the ranks instead of the original observations can provide us with more information for each observation about how central it is, and in which direction it is moving from the center. This is due to that length of rank (x) tells us how far away this point is from the center and the direction of rank (x) tells us about the direction of x from the center of the data. We start with the univariate centered rank definition for both sample and population, and then a generalization form with the multivariate rank definition can be considered.

Definition 3.2.3 *: The univariate spatial ranks functions:*

The usual univariate rank can be defined as $\sum_{i=1}^{n} I(X_i \leqslant x)$*, and a sample version for*

the univariate centered rank of x *w.r.t.* $X_1, X_2, \ldots, X_n$ *can be defined as,*

$$Rank(x) = R_X(x) = \frac{1}{n}\sum_{i=1}^{n} sign\,(x - X_i)\,. \tag{3.2.4}$$

It can be noticed that $Rank(x)$ satisfies:

$$-1 \leqslant Rank(x) \leqslant 1, \tag{3.2.5}$$

and this property makes the $Rank(x)$ a useful quantity for measuring both the direction and length of each observation from the center. For instance, when $Rank(x) = -1$ this implies that x is smaller or equal to the minimum ordered statistics such that $x \leqslant \min(x_1, x_2, \ldots, x_n)$. Conversely, when $Rank(x) = +1$ this implies that x is larger or equal to the maximum ordered statistics such that $x \leqslant \max(x_1, x_2, ..., x_n)$, and when $Rank(x) = 0$ this implies that x is the *median*. Furthermore, we can show that,

$$E[Rank(x)] = 2F(x) - 1, \tag{3.2.6}$$

where $F(x)$ is the distribution function of X_i. It is worth mentioning in this context that there is an important relationship between both $Rank(x)$ and the corresponding quantiles, such that $Rank(x) = u$ implies that x is the $\frac{u+1}{2}th$ quantile, and it is very important property that can be utilized in order to address the problem of ordering the high dimensional data.

Definition 3.2.4 *: The population and sample multivariate spatial ranks functions: Suppose that* $\mathbf{X} \in \mathbb{R}^d$ *has a* d*-dimensional distribution* F*, which is assumed to be absolutely continuous throughout this study, then the multivariate spatial rank function of the point* $\mathbf{x} \in \mathbb{R}^d$ *with respect to* F *can be defined as:*

$$Rank_F(\mathbf{x}) = E_F\left(\frac{\mathbf{x} - \mathbf{X}}{\|\mathbf{x} - \mathbf{X}\|}\right). \tag{3.2.7}$$

Now suppose that $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ *is a random sample with distribution* F*, then the sample version of the multivariate spatial rank function of* $\mathbf{x} \in \mathbb{R}^d$ *with respect to* $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ *is given by:*

$$Rank_{F_n}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} Sign(\mathbf{x} - \mathbf{X}_i) = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathbf{x} - \mathbf{X}_i}{\|\mathbf{x} - \mathbf{X}_i\|}. \tag{3.2.8}$$

Some important properties of the spatial rank function are:

1. If $Rank_F(\mathbf{x}) = \mathbf{0}$, then $\mathbf{x}$ is the spatial median.

2. $Rank_F(\mathbf{x}) = \mathbf{u}$ implies that $\mathbf{x}$ is the $\mathbf{u}$-th geometric quantile (Chaudhuri, 1996) of F.

3. $\|Rank_F(\mathbf{x})\|$ are bounded by 1, i.e. $\|Rank_F(\mathbf{x})\| \leqslant 1$ for all $\mathbf{x} \in \mathbb{R}^d$.

4. The distribution function F is a $1-1$ function of $Rank_F(\mathbf{x})$.

5. The spatial ranks are invariant under orthogonal transformations, but they are not invariant under general affine transformations of the data.

It is well known that both of spatial depth and spatial ranks are completely depended on each other, since $SD(\mathbf{x}) = 1 - \|Rank(\mathbf{x})\|$, where $SD(\mathbf{x})$ is the spatial depth function and hence $SD(\mathbf{x}) = 1$ implies that $\mathbf{x}$ is the spatial median. The origins of the spatial approach date back to Brown (1983), when he introduced the idea of spatial median considering the problem of robust location estimation for two-dimensional spatial data. After that, the geometry notions of the data started to be used in different important nonparametric functions such that the multivariate spatial quantiles by Chaudhuri (1996) and the multivariate spatial depth function by Serfling (2002).

3.3 Forward Search Based on Spatial Ranks

In this Section we propose a novel forward search algorithm which can be used as a clustering tool. As we discussed earlier, the forward search based on Mahalanobis distances cannot be correctly applied to asymmetric distributions and more generally to distributions, which depart from the elliptical symmetry assumptions, so an adjustment in the forward search technique will be helpful in order to address theses problems. We develop the forward search technique by using some methods based on notions of multivariate ranks. As a first step, we started with using the spatial ranks instead of Mahalanobis distances. To highlight the effect of this adjustment on the efficiency of the forward search, a comparison between the proposed and traditional algorithms has been considered in this Section considering either the spherically or elliptically symmetric cases.

3.3.1 Forward Search Based on Spatial Ranks Algorithm

In the forward search algorithm, let $S(m)$ be a subset of size m at a particular stage. Then define the spatial ranks of individual observations corresponding to the subset $S(m)$ as

$$r_i(m) = \frac{1}{m} \sum_{j \in S(m)} \frac{\mathbf{X}_i - \mathbf{X}_j}{\|\mathbf{X}_i - \mathbf{X}_j\|}, \tag{3.3.1}$$

for $i = 1, \ldots, n$. Let us now introduce the forward search procedure based on the multivariate spatial ranks (Baragilly and Chakraborty, 2016).

Forward search algorithm with spatial ranks:

1. In order to start the search, we need to choose an initial subset. Suppose that $S(m)$ is the initial subset with $m = d + 1$, then one search can be run from this starting point.

2. Calculate the spatial ranks $r_i(m)$ depending on the observations in the subset $S(m)$.

3. Compute $r_{min}(m)$, where $r_{min}(m) = \min \|r_i(m)\|$; $i \notin S(m)$.

4. Grow the subset $S(m)$ to $S(m+1)$ by taking $m+1$ observations $\mathbf{X}_i$'s, which correspond to smallest $m+1$ $\|r_i(m)\|$'s. Set $m = m+1$.

5. Iterate $2-4$ until $m = n-1$.

6. The forward plot of the spatial ranks can be obtained by plotting the $r_{min}(m)$ against the corresponding subset sizes m.

The previous algorithm is computationally easy and straightforward. It can be noticed that, when the points in $S(m)$ belong to the same cluster, $\|r_i(m)\|$ for a point $\mathbf{X}_i$ belonging to the same cluster is expected to be smaller than that of point from a different cluster. Even if our initial subset contain points from different clusters, the algorithm will ensure that $S(m)$ will move to a single cluster as it grows in size and is constructed by taking points with smallest ranks. So whenever $S(m)$ grows bigger than the cluster it originally belonged to, we expected to see a jump in the magnitude of the rank function as the nearest point to $S(m)$ is then from a different cluster. However, we can observe that $\|rank_F(\mathbf{x})\| < 1$ for all $\mathbf{x} \in \mathbb{R}^d$. Thus, all $\|r_i(m)\|$'s are bounded by 1. Hence, even if a particular point $\mathbf{X}_i$, is far from the cluster $S(m)$, the corresponding $\|r_i(m)\|$ may not be very large compared to an observation $\mathbf{X}_j$, which is an extreme observation in $S(m)$. For this reason, the plot of $r_{min}(m)$ against m may not show any sharp increase even when we include a point from a different cluster, and it becomes visually difficult to detect the clusters (Baragilly and Chakraborty, 2016). To enhance the visual detection of clusters, we modify the algorithm by using central rank regions determined by $r_{min}(m)$. We postpone the discussion on the forward search based on volume of central rank regions to Section 3.5, where we need firstly to know how the spatial rank's algorithm behaves and why we need to improve it by using the volume of central ranks region.

3.3.2 Some Numerical Examples

Example 1: Bivariate Mixture Distributions with Uncorrelated Variables

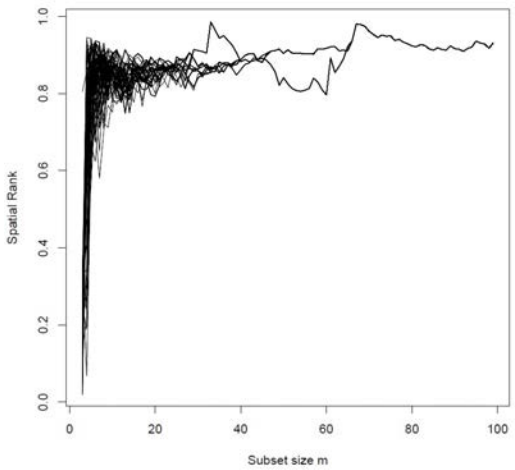
Similarly, as we did in Section 2.4.1, we now consider three bivariate mixture spherically symmetric distributions, where the variables are uncorrelated. The three bivariate mixture distributions are normal, Laplace and Student's t distributions with the same mixing proportion ($p = 0.3$), number of random starts ($k = 100$), and samples size ($n = 100$).

Figure 3.1 is a forward plot of minimum spatial ranks from 100 randomly chosen initial subsets for samples size $n = 100$ from bivariate mixture normal, Laplace and Student's t distributions with uncorrelated variables. Both Figure 2.1 and 3.1 give similar results. As we mentioned before, we are interested in the subsets $S(m)$ for these trajectories where there is evidence of a cluster structure.

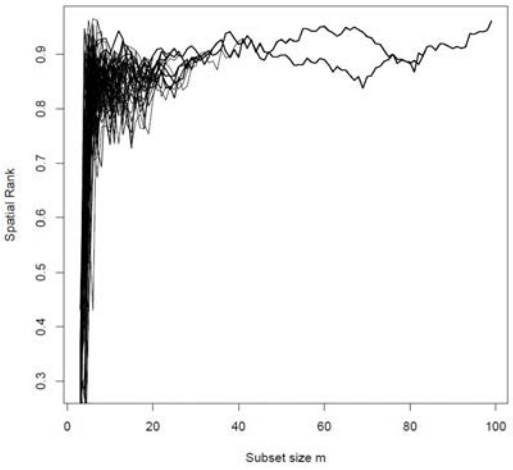
It can be clearly noticed that there are many different values of $r_{min}(m)$ presented in many trajectories. Moreover, the three plots in Figure 3.1 show that there is clearly common structure around subsets with size 30 and 70 respectively, where there are two clear maxima in these plots, one at $m = 30$ and the other at $m = 70$, suggesting the existence of two clusters. So these plots lead to the division of the data into two clusters, which means that the forward search based on spatial ranks performs well with the three spherically symmetric distributions, and it outperforms the one based on Mahalanobis distances for Laplace and t distributions.

Example 2: Bivariate Mixture Distributions with Correlated Variables

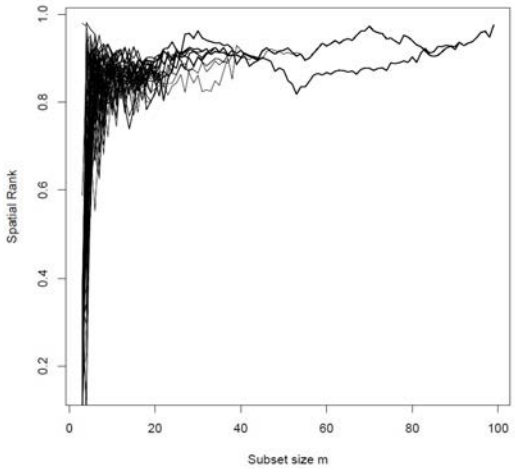
Now we consider that the data are generated from the same three bivariate mixture distributions but with correlated variables with same mixing proportion ($p = 0.3$), number of random starts ($k = 100$), and samples size ($n = 100$). Figure 3.2 is a forward plot of minimum spatial ranks from 100 randomly chosen initial subsets for samples size $n = 100$ from bivariate mixture normal, Laplace and t distributions with elliptic symmetry. From



(a) Bivariate normal



(b) Bivariate Laplace



(c) Bivariate t

Figure 3.1: Forward plot of minimum spatial ranks from 100 randomly chosen initial subsets for sample size $n = 100$ from bivariate mixture (a) normal, (b) Laplace and (c) t distributions with uncorrelated variables.

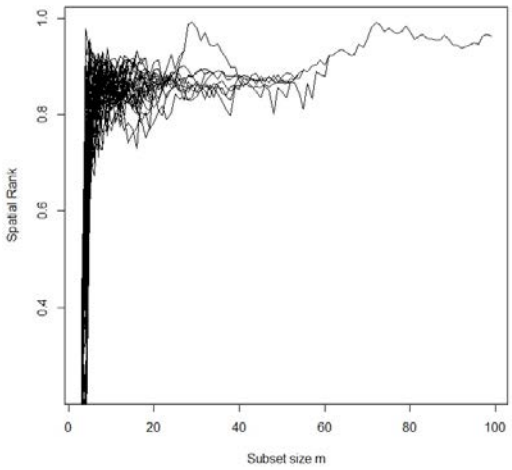
plots (a), (b) and (c) we see that there is again clearly common structure around subsets with size 30 and 70 respectively, where there are two clear maxima around $m = 30$ and $m = 70$, suggesting the existence of two clusters. So these plots also lead to the division of the data into two clusters. Compared to the forward plot based on Mahalanobis distances we can say that the forward search based on spatial ranks performs well with the three elliptically symmetric distributions, and it outperforms the one based on Mahalanobis distances for Laplace and t distributions.

Example 3: Trivariate Mixture Distributions with Uncorrelated Variables

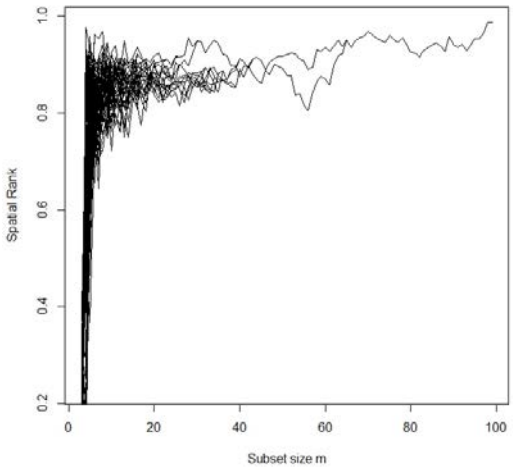
Here, we consider trivariate mixture distributions of normal, Laplace and Student's t with 3 degrees of freedom, as before with same mixing proportion ($p = 0.3$), number of random starts ($k = 100$), and samples size ($n = 100$). Figure 3.3 is a forward plot of minimum spatial ranks. There are two clear maxima one around $m = 30$ and the other around $m = 70$, which can be considered as indicator of the existence of two clusters. Compared to Figure 2.3, we can see that Figure 3.3 gives better results, where it gives plots with a clearer structure around the subsets with size 30 and 70, which means that the forward search based on spatial ranks gives better result for the data with higher dimensions. Moreover, it outperforms the one based on Mahalanobis distances for Laplace and t distributions.

Example 4: Trivariate Mixture Distributions with Correlated Variables

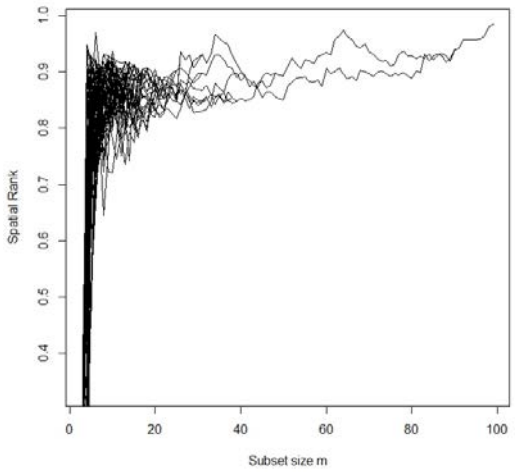
Now consider the same data but under elliptic symmetry. Figure 3.4 is a forward plot of minimum spatial ranks from 100 randomly chosen initial subsets for samples size $n = 100$ from trivariate mixture normal, Laplace and t distributions. Similarly, this Figure gives a reasonable structure like Figure 3.3, and it gives better results than Figure 2.4, where there are also two clear maxima one around $m = 30$ and the other around $m = 70$, which suggests two clusters. So, the forward search based on spatial ranks outperforms forward



(a) Bivariate normal

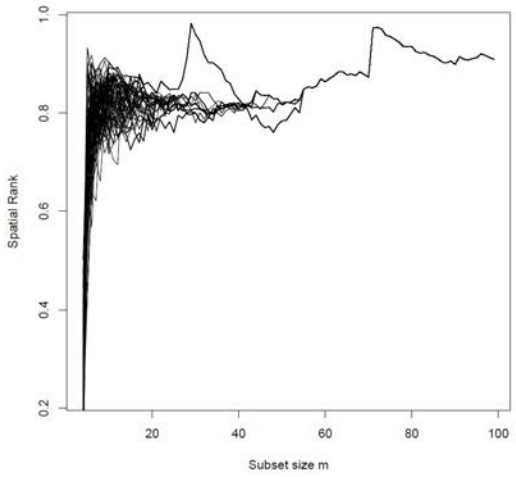


(b) Bivariate Laplace

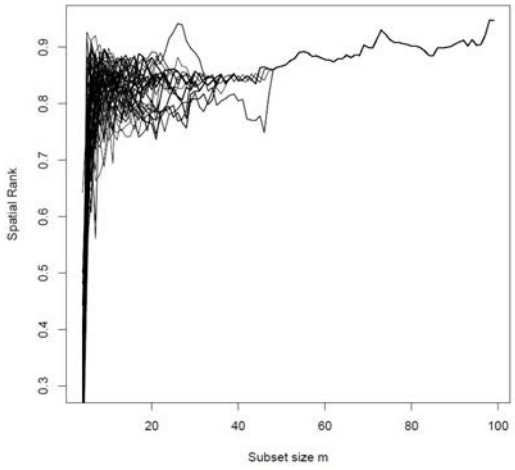


(c) Bivariate t

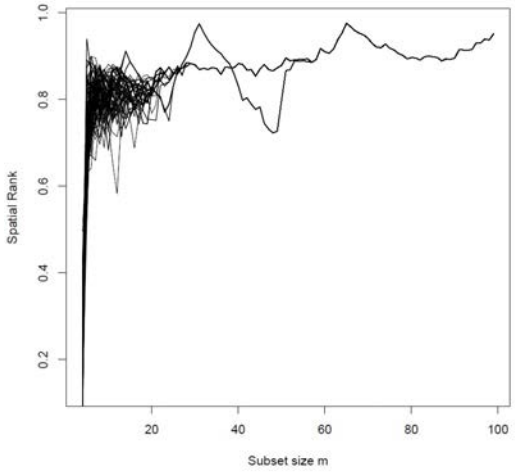
Figure 3.2: Forward plot of minimum spatial ranks from 100 randomly chosen initial subsets for sample size $n = 100$ from bivariate mixture (a) normal, (b) Laplace and (c) t distributions with correlated variables.



(a) Trivariate normal



(b) Trivariate Laplace



(c) Trivariate t

Figure 3.3: Forward plot of minimum spatial ranks from 100 randomly chosen initial subsets for sample size $n = 100$ from trivariate mixture (a) normal, (b) Laplace and (c) t distributions with uncorrelated variables.

search based on Mahalanobis distances for the three trivariate mixture distributions under the elliptic symmetry case.

So, we conclude that the forward search algorithm based on spatial ranks outperforms the traditional one based on Mahalanobis distances under either spherical or elliptic symmetry, especially for Laplace and t distributions. However, as mentioned earlier, the spatial ranks are bounded by 1 and hence do not produce a good visual effect to detect clusters in an easier way. In order to enhance the visual detection of clusters, we modify the algorithm by using central rank regions determined by $r_{min}(m)$.

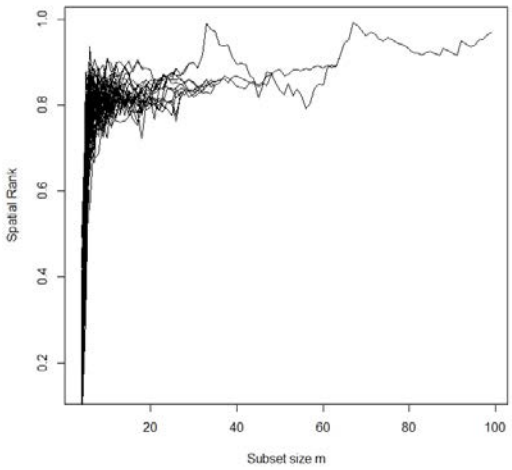
3.4 Central Rank Regions and Volume of Central Rank Regions

3.4.1 Geometric Quantiles for Multivariate Data

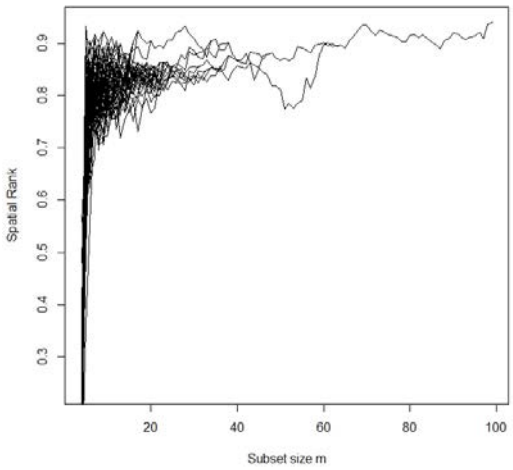
To calculate the central rank regions, we may use the geometric quantiles (Chaudhuri, 1996). Serfling (2002) proposed the concept of volume functional based on spatial central regions. He considered the spatial quantiles, introduced by Chaudhuri (1996) and Koltchinskii (1997) as a certain form of generalization of the univariate case based on the L_1 norm.

According to Chaudhuri (1996), the spatial quantiles can be defined as vectors in $\mathbb{R}^d$ that are indexed by a vector $\mathbf{u}$ in d-dimensional unit ball. Let $B^{(d)} = \{\mathbf{u} \mid \mathbf{u} \in \mathbb{R}^d, \|\mathbf{u}\| < 1\}$, be an open ball, for any $\mathbf{u} \in B^{(d)}$ and $\mathbf{t} \in \mathbb{R}^d$, $\Phi(\mathbf{u}, \mathbf{t}) = \|\mathbf{t}\| + < \mathbf{u}, \mathbf{t} >$ where $< ., . >$ denotes the usual Euclidean inner product. So, the spatial quantile corresponding to $\mathbf{u}$ and based on $\mathbf{X}_1, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ can be defined as,

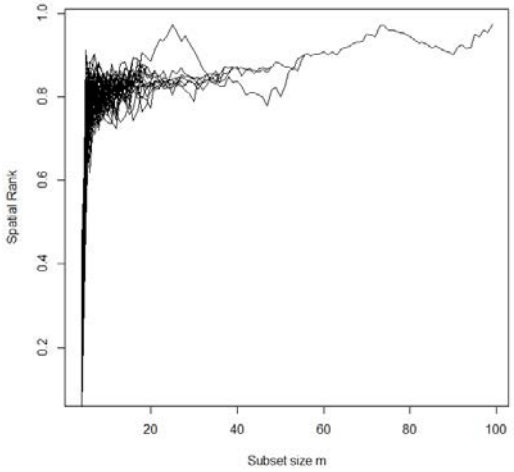
$$\widehat{\mathbf{Q}}_n(\mathbf{u}) = \arg\min_{\mathbf{Q} \in \mathbb{R}^d} \sum_{i=1}^{n} \Phi(\mathbf{u}, \mathbf{X}_i - \mathbf{Q}).$$



(a) Trivariate normal



(b) Trivariate Laplace



(c) Trivariate t

Figure 3.4: Forward plot of minimum spatial ranks from 100 randomly chosen initial subsets for sample size $n = 100$ from trivariate mixture (a) normal, (b) Laplace and (c) t distributions with correlated variables.

And from Theorem 1.1.2 of Chaudhuri (1996) we observe that,

$$\sum_{i=1}^{n} \frac{\mathbf{X}_i - \widehat{\mathbf{Q}}_n(\mathbf{u})}{\|\mathbf{X}_i - \widehat{\mathbf{Q}}_n(\mathbf{u})\|} + n\mathbf{u} = \mathbf{0},$$

if $\widehat{\mathbf{Q}}_n(\mathbf{u}) \neq \mathbf{X}_i$ for all $1 \leqslant i \leqslant n$, which implies,

$$\mathbf{u} = \frac{1}{n} \sum_{i=1}^{n} \frac{\widehat{\mathbf{Q}}_n(\mathbf{u}) - \mathbf{X}_i}{\|\widehat{\mathbf{Q}}_n(\mathbf{u}) - \mathbf{X}_i\|}. \tag{3.4.1}$$

In addition, Serfling (2004) has defined $Rank_{F_n}(\mathbf{x})$ as the inverse function of the geometric quantile, $\widehat{\mathbf{Q}}_n(\mathbf{u})$. Consequently, we can write the previous equation as,

$$\mathbf{u} = Rank_{F_n}(\widehat{\mathbf{Q}}_n(\mathbf{u})) = Rank_{F_n}(\mathbf{x}),$$

hence,

$$\widehat{\mathbf{Q}}_n(\mathbf{u}) = \mathbf{x} \quad implies \quad Rank_{F_n}(\mathbf{x}) = \mathbf{u}.$$

Serfling (2002) has considered the volume functional as a spatial scale curve that provides a convenient two-dimensional characterization of the spread of a multivariate distribution of any dimension. Suppose that $Q_F(\mathbf{u})$ is the $u-th$ spatial quantile (Chaudhuri, 1996) corresponding to the underlying distribution function F for $\mathbf{X}$ on $\mathbb{R}^d$, and for $\mathbf{u} \in B^{(d-1)}(0)$, a $Q_F(\mathbf{u})$ having both direction and magnitude. By Koltchinskii (1997), the quantile $Q_F(\mathbf{u})$ may be represented as the solution $\mathbf{X} = \mathbf{X}_u$ of:

$$-E\left\{\frac{\mathbf{X} - \mathbf{x}}{\|\mathbf{X} - \mathbf{x}\|}\right\} = \mathbf{u}, \tag{3.4.2}$$

namely, it is that spatial quantile $Q_F(\mathbf{u}_x)$ indexed by the average unit vector $\mathbf{u}_x$ pointing to $\mathbf{x}$ from a random point having distribution F. we interpret $\mathbf{u}_x$ as the inverse at x of

the spatial quantile function Q_F and denote it by $Q_F^{(-1)}(\mathbf{x})$.

Definition 3.4.1 *: The sample spatial quantile function:*
For a dataset $\mathbf{X}_1, \ldots, \mathbf{X}_n$ *,computation of the sample spatial quantile function can be obtained by:*

$$-\frac{1}{n}\sum_{i=1}^{n}\frac{\mathbf{X}_i - \mathbf{x}}{\|\mathbf{X}_i - \mathbf{x}\|} = \mathbf{u}, \tag{3.4.3}$$

where the left-hand side of (3.4.2) is the sample version of the centered rank function.

3.4.2 Volume of Central Rank Regions

Definition 3.4.2 *: Central rank regions:*
Corresponding to the spatial quantile function Q_F*, we call the* $r-th$ *central region:*

$$C_F(r) = \{Q_F(\mathbf{u}) : \|\mathbf{u}\| \leqslant r\}, \tag{3.4.4}$$

and the central rank regions:

$$C_F(r) = \{\mathbf{x} : \|rank_F(\mathbf{x})\| \leqslant r\}, \qquad 0 < r < 1. \tag{3.4.5}$$

Definition 3.4.3 *: Volume functional of multivariate central ranks region:*
According to the central rank regions' definition, one can define the (real-valued) volume functional of the multivariate central ranks region as:

$$V_F(r) = volume(C_F(r)), \qquad 0 \leqslant r < 1. \tag{3.4.6}$$

Serfling (2002) pointed out that as an increasing function of r, $V_F(r)$ characterizes the spread of F in terms of expansion of the central regions $C_F(r)$. For each r, $V_F(r)$ is invariant under shift and orthogonal transformations, and $V_F(r)^{1/d}$ is equivariant un-

der homogeneous scale transformations. Serfling (2006b) proposed applications of depth functions by using the central regions and the volume functional, and showed that depth-based central regions $C_F(r)$ are affine equivariant, nested, connected, and compact. Now we define the sample central rank regions and the sample volume of central rank regions:

Definition 3.4.4 *: Sample central ranks region and sample volume functional:*
The Sample central rank regions can be defined as:

$$C_{Fn}(r) = \{\mathbf{x} : \|rank_{Fn}(\mathbf{x})\| \leqslant r\}, \qquad 0 < r < 1. \tag{3.4.7}$$

in this case we can define the sample volume functional of multivariate central ranks region as:

$$V_{Fn}(r) = volume(C_{Fn}(r)), \qquad 0 \leqslant r < 1. \tag{3.4.8}$$

Theorem 3.4.1 *: The spatial rank vector, spherically symmetric case:*
For the random vector $\mathbf{X}$ *with the distribution function* F*, suppose that* F *is spherically symmetric about* θ*, then the spatial rank vector of* $\mathbf{x}$ *can be written as a multiplication of an increasing function* $h(.)$ *and the basic term in the spatial rank function* $\frac{\mathbf{x}-\theta}{\|\mathbf{x}-\theta\|}$*, such that:*

$$Rank_F(\mathbf{x}) = h(\|\mathbf{x} - \theta\|)\frac{\mathbf{x} - \theta}{\|\mathbf{x} - \theta\|}. \tag{3.4.9}$$

Moreover, from the previous Theorem, the central rank region $C_F(p)$ for the spherically symmetric case can be written as:

$$C_F(p) = \{\mathbf{x} : \|\mathbf{x}\| \leqslant r_F(p)\}, \tag{3.4.10}$$

where $r_F(p)$ is the $p-$th quantile of $\|\mathbf{X} - \theta\|$.

Theorem 3.4.2 *: The spatial rank vector, spherically symmetric case:*
For the random vector $\mathbf{X}$ *with the distribution function* F*, suppose that* F *is spherically*

symmetric about θ, $C_F(p)$ is the central rank region, $r_F(p)$ is the $p-$th quantile of $\|\mathbf{X}-\theta\|$, and $V_F(p)$ is the volume of the central rank region $C_F(p)$, then $V_F(p)$ can be written as:

$$V_F(p) = \frac{\pi^{\frac{d}{2}}(r_F(p))^d}{\Gamma(\frac{d}{2}+1)}. \tag{3.4.11}$$

For more comprehensive details and proof of this theorem, the reader is referred to(Guha, 2012).

An important point has been considered by Guha (2012) regarding to the $p-$th quantile $r_F(p)$. She pointed out that if the underlying distribution is the standard multivariate normal distribution, then $r_F^2(p)$ will be the $p-$th quantile of the χ_d^2 distribution. On the other hand, if the distribution is the standard multivariate Laplace distribution, then $r_F(p)$ will be the $p-$th quantile of the $\Gamma(d,1)$ distribution and for the standard multivariate t distribution with ν degrees of freedom, $r_F^2(p)/d$ will be the $p-$th quantile of the $F_{d,\nu}$ distribution.

3.4.3 Spherically and Elliptically Symmetric Distributions

Oja (2010) proposed a good review of the construction of the multivariate models. He illustrated the concepts of the spherically symmetrical, marginally symmetrical, centrally symmetrical and exchangeable cases. Moreover, he discussed the multivariate elliptical distributions and their properties. Maxwell (1860) is considered one of the earlier publications that discussed the spherically symmetric distributions and their properties. Hartman and Wintner (1940) gave a discussion about the spherical approach to the normal distribution. Chmielewski (1981) introduced an extensive review for the spherically and elliptically symmetric distributions. Some notions of multivariate symmetry and asymmetry have been considered by Serfling (2006a).

Firstly, we start with the definition of spherically symmetric distributions, then the concept of elliptically symmetric distribution can be easily discussed. One can say that

the random vector $\mathbf{X}$ has a spherically symmetric distribution about the point θ if the distribution of $(\mathbf{X}-\theta)$ remains unchanged under any orthogonal transformation. A mathematical expression can be written in this context as,

$$\mathbf{X} - \theta = A(\mathbf{X} - \theta), \tag{3.4.12}$$

where A is an orthogonal $d \times d$ transformation matrix. Lord (1954) derived the characteristic function of the random vector $\mathbf{X}$ and he illustrated that it has a density function which is a function of the form $(x-\theta)^T(x-\theta)$ if it exists.

It is worth mentioning in this context that the spherically symmetric distributions have an important property which is both $\| \mathbf{X}-\theta\|$ and the random unit vector $(\mathbf{X}-\theta)/\| \mathbf{X}-\theta\|$ which is distributed uniformally are independent.

Now, we consider the concept of elliptically symmetric distribution. One can say that the random vector $\mathbf{X}$ has an elliptically symmetric distribution with parameters θ and Σ if there exists an orthogonal $d \times d$ transformation matrix A such that $A(\mathbf{X} - \theta)$ has a spherically symmetric distribution about 0. Alternative definition can be introduced such that the random vector $\mathbf{X}$ has an elliptically symmetric distribution if it can be written in the following mathematical expression,

$$\mathbf{X} = \mathbf{A}\mathbf{Y} + \theta, \tag{3.4.13}$$

where $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^T$, and $\mathbf{Y}$ has a spherically symmetric distribution about 0. It can be noticed that when $\theta = \mathbf{0}$ and $\mathbf{\Sigma} = \mathbf{I}_d$, then $\mathbf{X}$ is said to have spherically symmetric distribution centered at zero. An extensive discussion of the properties of the elliptically symmetric distributions can be found in Fang et al. (1990).

3.5 Forward Search Based on Volume of Central Rank Regions

As a second improvement in the forward search method, we modify the previous algorithm by using central rank regions determined by $r_{min}(m)$ to enhance the visual detection of clusters. We modify Step 6 of the above algorithm and produce a forward plot of the volume functional, $vol(m)$ against the subset size m, where $vol(m)$ is the volume of the central rank region determined by $r_{min}(m)$, i.e. $vol(m) = V_{S(m)}(r_{min}(m))$ based on the subset $S(m)$. Note that, as soon as we include a point from a different cluster in the subset, the volume of the central rank region increases substantially and then it may remain around that large volume as it includes more and more points from that cluster and we may see a sharp decrease in volume after some time if the subset $S(m)$ moves to the new cluster completely. However that depends on the relative cluster sizes and how far they are from each other. Eventually, points from all clusters will be in $S(m)$ and the volume of the central rank regions will grow with m.

According to Baragilly and Chakraborty (2016), in order to compute the volume of the central rank regions, we first compute a discretized boundary of $C_F(r)$ by computing geometric quantiles corresponding to index vector $\mathbf{u}$ with $\|\mathbf{u}\| = r$ following Chaudhuri (1996). We construct a convex hull of this discretized boundary of quantiles to obtain a convex polyhedra and then compute the volume of that convex polyhedra using the quickhull algorithm of Barber et al. (1996). So, the volume of the discretized central rank region $C_F(r)$ is computed using the quickhull algorithm of Barber et al. (1996), which was implemented in the R package `geometry`. The computation of volumes may be computationally expensive in very high dimensions. This computational simplification produces an estimate of the volume of $C_F(r)$, however, the precision of the estimate increases with the increase in the number of points chosen on the boundary. We may

need to choose the level of discretization sensibly to balance between the computational time and accuracy in estimation. As this is a visualization tool, even if our estimate of volume is not too precise, we are still able to see the distinct jumps for the clusters when they are well separated.

In principle the initial subset size can be anything more than 1 as the rank of any $\mathbf{x} \in \mathbb{R}^d$ with respect to a single data point is always 1 and we cannot proceed in our algorithm. Also, note that in the modified version of the algorithm, we are computing volumes of central rank regions and as we mentioned earlier that the volume provides a measure of scale, the computation of volumes are meaningful only when the number of observations are at least $d+1$. Thus, purely for more stability in the algorithm, we choose a initial subset size of $d+1$. If there are large number of clusters and all are with sizes smaller than $d+1$, then our algorithm will not be able to estimate the number of clusters efficiently, but that is a rarity for large sample size n (Baragilly and Chakraborty, 2016).

3.5.1 Algorithm for the Forward Search Based on Volume of Central Rank Regions

1. In order to start the search, we need to choose an initial subset. Suppose that $S(m)$ is the initial subset with $m = d+1$, then one search can be run from this starting point.

2. Calculate the spatial ranks $r_i(m)$ depending on the observations in the subset $S(m)$.

3. Compute $r_{min}(m)$, where $r_{min}(m) = \min \|r_i(m)\|$; $i \notin S(m)$.

4. Grow the subset $S(m)$ to $S(m+1)$ by taking $m+1$ observations $\mathbf{X}_i$'s, which correspond to smallest $m+1$ $\|r_i(m)\|$'s. Set $m = m+1$.

5. Iterate $2-4$ until $m = n-1$.

6. Compute $vol(m)$ which is the volume of the central rank region determined by $r_{min}(m)$, i.e. $vol(m) = V_{S(m)}(r_{min}(m))$ based on the subset $S(m)$.

7. The forward plot of the volume of central rank region can be obtained by plotting the $vol(m)$ against the corresponding subset sizes m.

3.5.2 Some Numerical Examples

In this Section, we apply a series of simulated cases which can show the efficiency of the forward search based on the volume functional of central rank regions especially when we consider the Laplace and t distributions under elliptical symmetry. Again, we generated data from bivariate and trivariate mixture normal, Laplace and t distributions, when the variables are either correlated or uncorrelated, with $(p = 0.3)$, $(k = 100)$, and $(n = 100)$.

Example 1: Bivariate Mixture Distributions with Uncorrelated Variables

Figure 3.5 is a forward plot of the volume functional of central rank regions from 100 randomly chosen initial subsets for samples size $n = 100$ from bivariate mixture normal, Laplace and t distributions with uncorrelated variables. For the three plots (a), (b) and (c) there is clearly some common structure around subsets with size 30 and 70 respectively. There are two clear maxima in the three plots, one at $m = 30$ and the other at $m = 70$, suggesting the existence of two clusters. Compared to Figure 3.1, we can notice that both the forward plot based on the spatial ranks and volume of central rank regions were able to detect the clusters in the generated data, however, the forward plots based on the volume of central rank regions give better results, specially in Laplace and t distributions, where it gives plots with a clearer structure around subsets with size 30 and 70. Moreover, it is more accurate in the purpose of visualization since we can easily determine the number of clusters from the plot based on volume of central rank regions. Thus, it should be concluded that the forward search based on volume of central rank regions outperforms

forward search based on Mahalanobis distances and spatial ranks for the bivariate mixture distributions under spherical symmetry.

Example 2: Bivariate Mixture Distributions with Correlated Variables

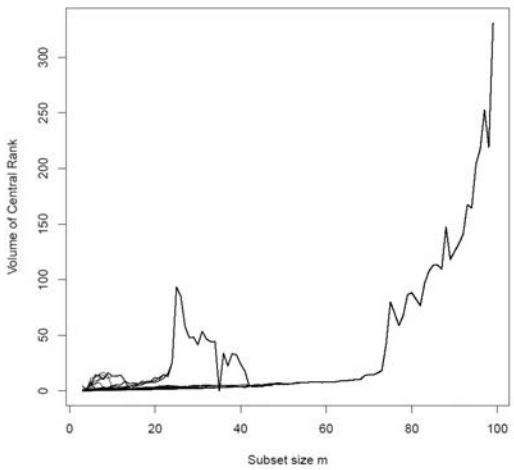
Now, we assume that there is a correlation among the variables in the previous model. Figure 3.6 is a forward plot of volume of central rank regions from 100 randomly chosen initial subsets for samples size $n = 100$ from bivariate mixture normal, Laplace and t distributions with correlated variables. Similarly, the three plots (a), (b) and (c) in Figure 3.6 show that there is clearly common structure around subsets with size 30 and 70 respectively, where there are two clear maxima in these plots, one at $m = 30$ and the other at $m = 70$, suggesting the existence of two clusters. The peaks in the three plots are clearer than those in Figure 3.2 which conclude that the forward search based on volume of central rank regions outperforms forward search based on Mahalanobis distances and spatial ranks for the bivariate mixture distributions under elliptic symmetry.

Example 3: Trivariate Mixture Distributions with Uncorrelated Variables

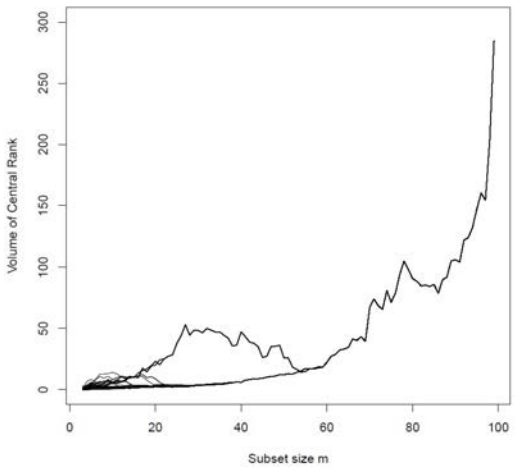
Here we consider trivariate mixture distributions of normal, Laplace and Student's t with 3 degrees of freedom, as before with $(p = 0.3)$, $(k = 100)$, and $(n = 100)$. Figure 3.7 is a forward plot of volume of central rank regions for samples size $n = 100$ from trivariate mixture normal, Laplace and t distributions with uncorrelated variables. It is very clear that the three plots (a), (b) and (c) give sharp peaks clearer than them in the plots of the bivariate distributions, which indicate that the algorithm performs well with the higher dimension.

Example 4: Trivariate Mixture Distributions with Correlated Variables

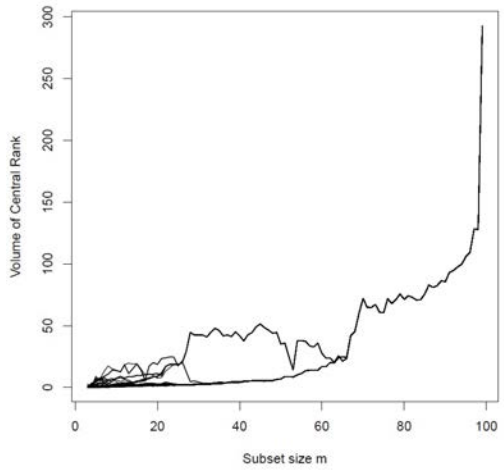
Now, we assume that there is a correlation among the variables in the previous model. As we can see in Figure 3.8, which is a forward plot of volume of central rank regions for samples size $n = 100$ from trivariate mixture normal, Laplace and t distributions with



(a) Bivariate normal

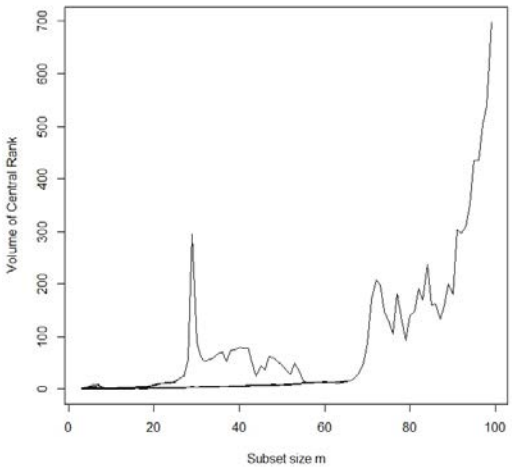


(b) Bivariate Laplace

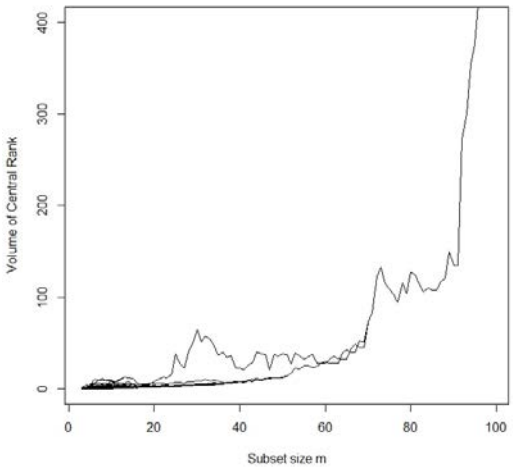


(c) Bivariate t

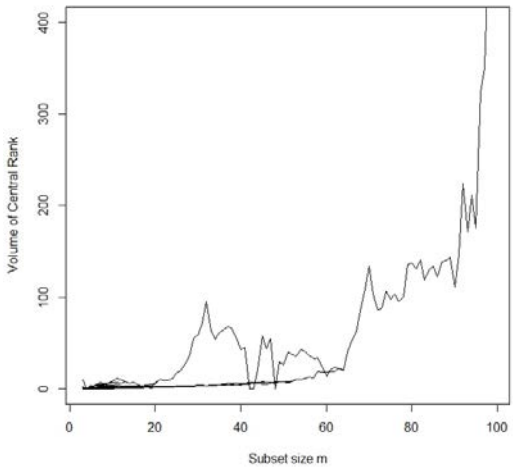
Figure 3.5: Forward plot of volume of central rank regions from 100 randomly chosen initial subsets for sample size $n = 100$ from bivariate mixture (a) normal, (b) Laplace and (c) t distributions with uncorrelated variables.



(a) Bivariate normal

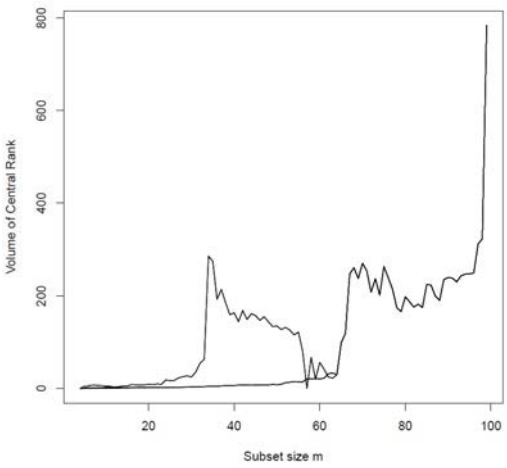


(b) Bivariate Laplace

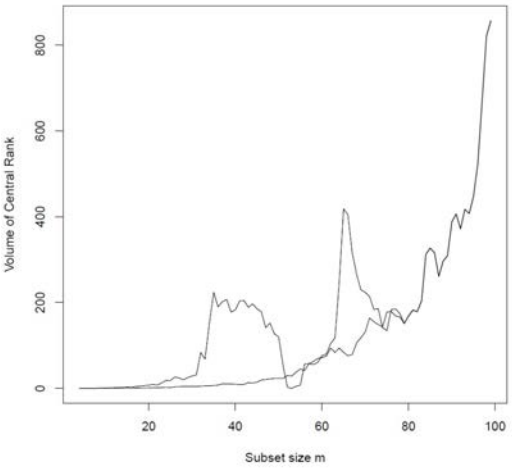


(c) Bivariate t

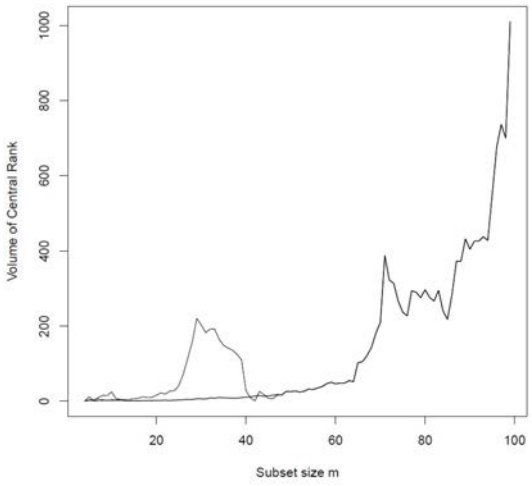
Figure 3.6: Forward plot of volume of central rank regions from 100 randomly chosen initial subsets for sample size $n = 100$ from bivariate mixture (a) normal, (b) Laplace and (c) t distributions with correlated variables.



(a) Trivariate normal



(b) Trivariate Laplace



(c) Trivariate t

Figure 3.7: Forward plot of volume of central rank regions from 100 randomly chosen initial subsets for sample size $n = 100$ from trivariate mixture (a) normal, (b) Laplace and (c) t distributions with uncorrelated variables.

correlated variables, there are two sharp peaks at $m = 30$ and $m = 70$. Clearly, they are clearer than them in the plots of the bivariate distributions as well, which indicate that the algorithm performs well with the higher dimension unlike the traditional algorithm based on Mahalanobis distances, where its performance is getting worse when higher dimensional data under elliptic symmetry.

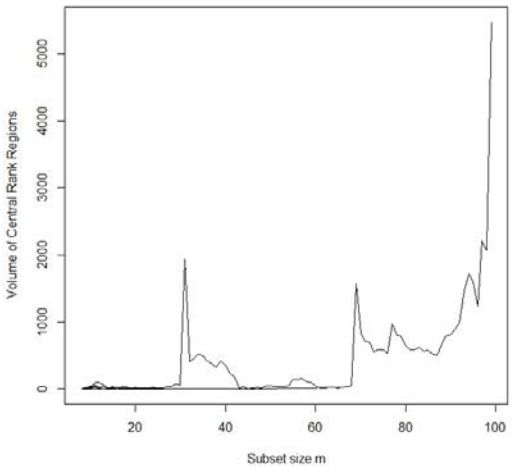
It should be concluded that the forward search based on volume functional of central rank regions outperforms forward search based on Mahalanobis distances for either bivariate or trivariate Laplace and t distributions under either spherical or elliptical symmetry. Moreover, the algorithm based on the volume of central rank regions gives better visualization with sharp peaks than the algorithm based on spatial ranks.

Mixture of 3 bivariate normal distributions

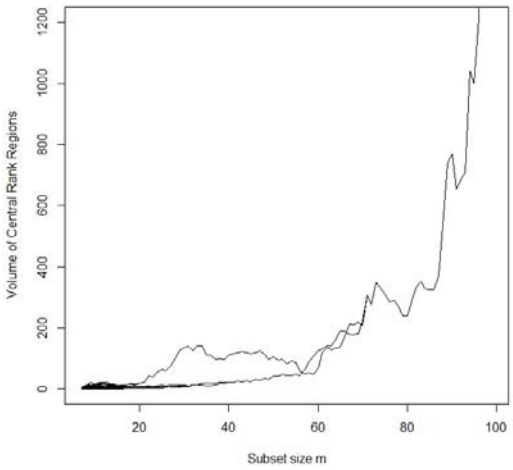
In Figure 3.9, we present an example of a forward plot based on both spatial ranks and volume of central rank regions, where the data is simulated from a mixture of 3 bivariate normal distributions that defined in (2.5.1):

$$p_1 N_2(\boldsymbol{\mu}_1, \boldsymbol{I}) + p_2 N_2(\boldsymbol{\mu}_2, \boldsymbol{I}) + (1 - p_1 - p_2) N_2(\boldsymbol{\mu}_3, \boldsymbol{I}),$$

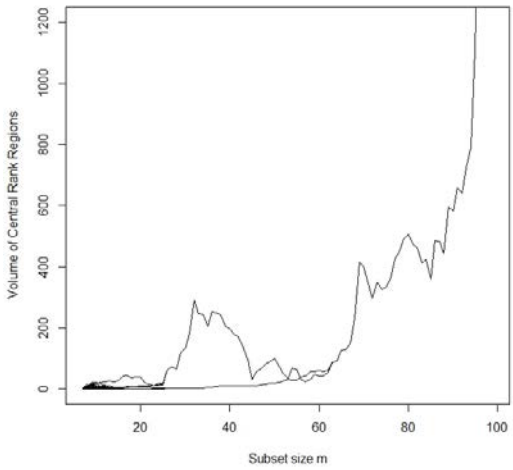
where $\boldsymbol{\mu}_1 = (0, 4)^\top$, $\boldsymbol{\mu}_2 = (-4, -4)^\top$, $\boldsymbol{\mu}_3 = (4, -4)^\top$ and $p_1 = 0.2$, $p_2 = 0.3$. With trajectories from 100 randomly chosen initial subsets, we see a clear pattern of 3 cluster sizes here in both of the forward plots based on spatial ranks and volume of central rank regions unlike the forward plot based on Mahalanobis distances in Figure (2.8), which gives a clear pattern of 4 cluster sizes here, however the simulated data includes 3 clusters.



(a) Trivariate normal

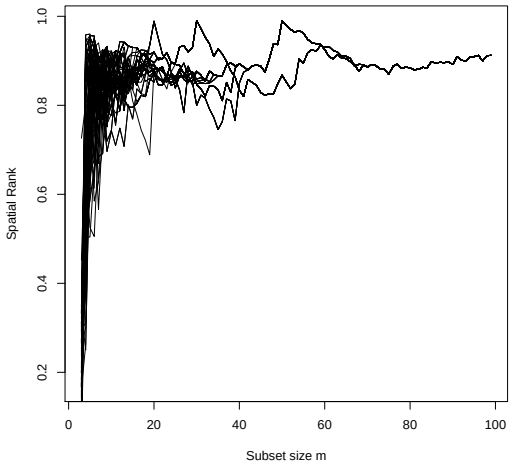


(b) Trivariate Laplace

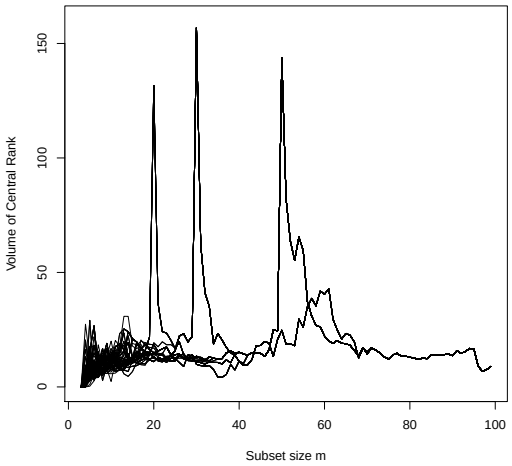


(c) Trivariate t

Figure 3.8: Forward plot of volume of central rank regions from 100 randomly chosen initial subsets for sample size $n = 100$ from trivariate mixture (a) normal, (b) Laplace and (c) t distributions with correlated variables.



(a) Forward plot based on spatial ranks



(b) Forward plot based on volume of central rank regions

Figure 3.9: Forward plot based on (a) spatial ranks and (b) volume of central rank regions, from 100 randomly chosen initial subsets for a mixture bivariate normal data set with 3 mixing densities.

3.6 Simulation Envelope and Entry Plot Based on Spatial Ranks

As we we mentioned earlier, using the envelopes in the forward plot is helpful to know the level of fluctuations to be expected in the forward plots. In this Section, we gives the steps of the simulation envelope's algorithm based on the spatial ranks.

1. Simulate data from the empirical standard underlying distribution.

2. Choose an initial subset $S(m)$ with $m = d+1$, and start the search from this starting point.

3. Calculate the spatial ranks $r_i(m)$ based on the observations in the subset $S(m)$.

4. Compute $vol(m)$, where $vol(m) = V_{S(m)}(r_{min}(m))$ based on the subset $S(m)$.

5. Grow the subset $S(m)$ to $S(m+1)$ by taking $m+1$ observations $\mathbf{X}_i$'s, which correspond to smallest $m+1$ $\|r_i(m)\|$'s. Set $m = m+1$.

6. Iterate $3-5$ until $m = n-1$

7. Iterate $1-6$ 1000 times, so for each subset size m we have 1000 values of $vol(m)$.

8. Take 99% percentiles of these $vol(m)$ and plot it against m to get the 99% envelope.

Figure 3.10 is a forward plot of volume of central rank regions from 100 random starts with 1% and 99% envelopes for sample size $n = 100$ from bivariate mixture normal distribution with uncorrelated variables. The figure shows that the volume values at the beginning of the search all lie on or within the simulation envelopes until $m = 16$ where the volume values have been increased to reach to the first peak at $m = 30$ and then they started to decrease again. Following that, the values started to exit from the simulation envelopes again as a preface to reach to the second peak at $m = 70$ and naturally they started to increase gradually until the end of the search.

Now, we propose the entry plot based on spatial ranks. The entry plot essentially shows which groups of observations are in the subset. To get the entry plot based on spatial ranks we have to order the observations based on the spatial ranks $r_i(m); i = 1, \ldots, n$, then plot the observations in the subset against the subset size. Figure 3.11 is an entry plot based on spatial ranks from $m_0 = 3$ with 100 randomly chosen initial subsets for a mixture bivariate normal data set with 3 mixing densities that defined in (2.5.1). From the Figure we can see that the observations are ordered so that 1-20 are those from the small group with the 20 observations, 21-50 are those from the second group with the 30 observations and 51-100 coming from the big group with 50 observations. The plot at $m = 3$ shows that the initial subset includes observations from the three groups. In the second subset of the forward search, at $m = 4$, a large amount of interchange has been happened, where the observations in this subset started to enter in the second group.

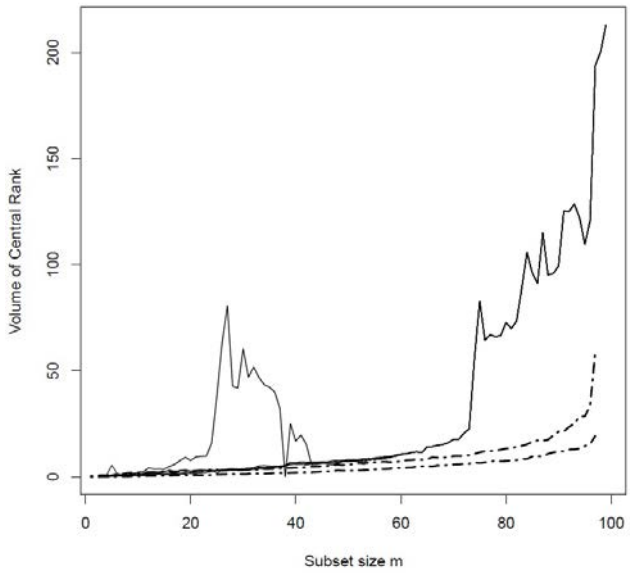


Figure 3.10: Forward plot of volume of central rank regions from 100 random starts with 1% and 99% envelopes for sample size $n = 100$ from bivariate mixture normal distribution with uncorrelated variables.

Thereafter, from $m = 32$ to $m = 44$ the subset consists solely of observations from the small group. From $m = 36$ some observations from the third group started to join the subset. From $m = 63$ the observations from the small group started to join again the subset. At the end of the search ($m = 100$), all the observations entered the search.

3.7 Real Data Examples

In this Section, we check the performance of the forward search algorithm based on volume of central rank regions in different real datasets. We compare the performance of the proposed forward search method for three different real datasets with two popular clustering methods: mclust approach (Fraley and Raftery, 2003) where the best number of groups is chosen according to BIC and K-means where the best number of groups is chosen according to CH index.

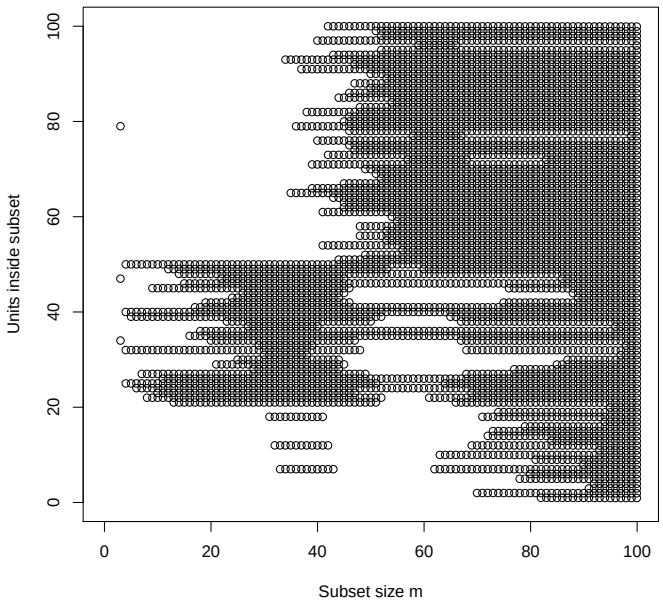


Figure 3.11: Entry plot based on spatial ranks from $m_0 = 3$ with 100 randomly chosen initial subsets for a mixture bivariate normal data set with 3 mixing densities.

3.7.1 Financial Data

The first real dataset we consider in this Chapter is a financial data, which stems from the Italian financial journal Il Sole - 24 Ore for May 7th, 1999 and has been analyzed in Atkinson et al. (2004). It is a real data contains measurements on three variables monitoring the performance of 103 investment funds operating in Italy since April 1996 [Table A.16 of Atkinson et al. (2004)]. These three variables are, y_1: short term (12 month) performance, y_2: medium term (36 month) performance, and y_3: medium term (36 month) volatility. Additionally, this data include two different kinds of fund, since the units 1- 56 are all stock funds whereas units 57- 103 are balanced funds.

Atkinson et al. (2004) and Atkinson et al. (2006) applied their forward search method based on Mahalanobis distances to the clustering of these financial data and introduced detailed analysis of it. Table A.16 of Atkinson et al. (2004) shows the Italian investment

funds since April 1996 in the short and medium term performance (y_1, y_2) and medium term volatility (y_3).

As a preliminary analysis, we prepare a scatter-plot matrix of the data in Figure 3.12, and 3D scatter-plot in Figure 3.13. Initially, from either the scatter-plot matrix or the 3D scatter-plot, it can be clearly seen that there are two clusters with some observations between them, it is clearer in the y_1 vs y_3 panels, and these observations are not close to either cluster centre.

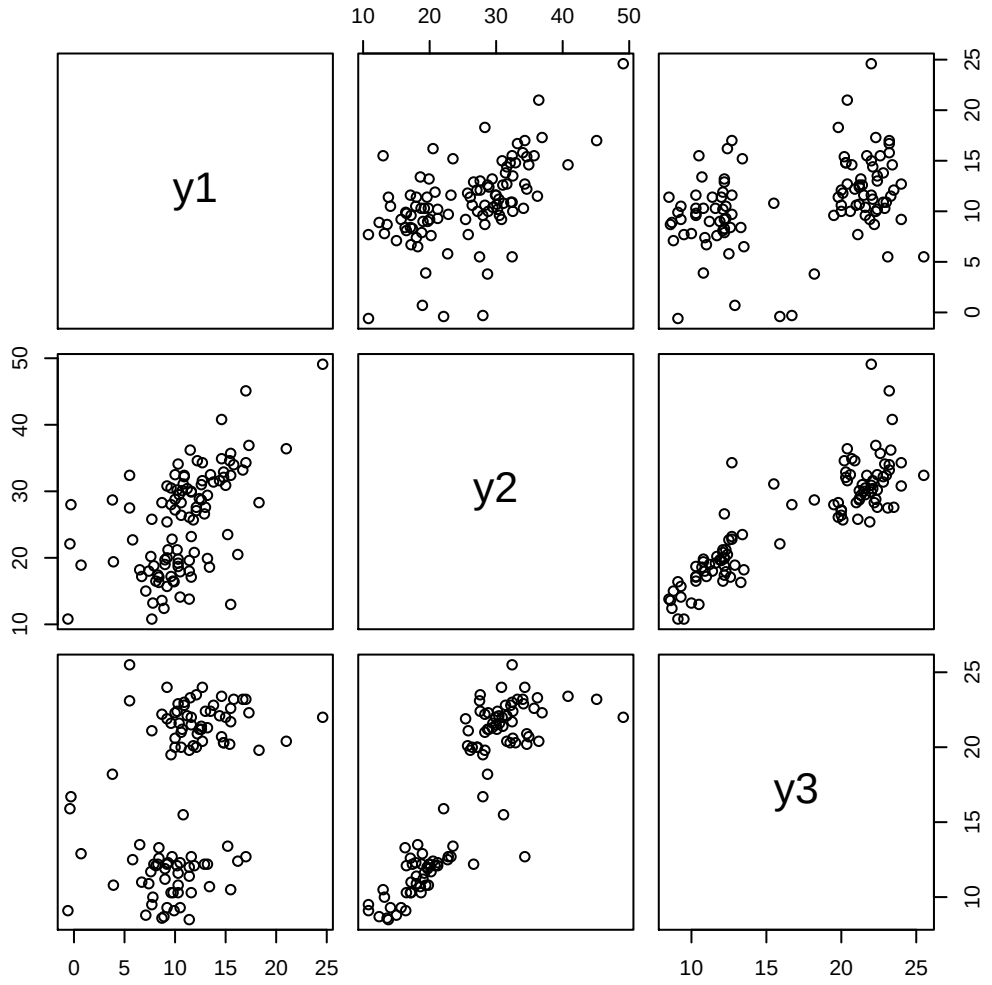


Figure 3.12: Financial data: scatter-plot matrix.

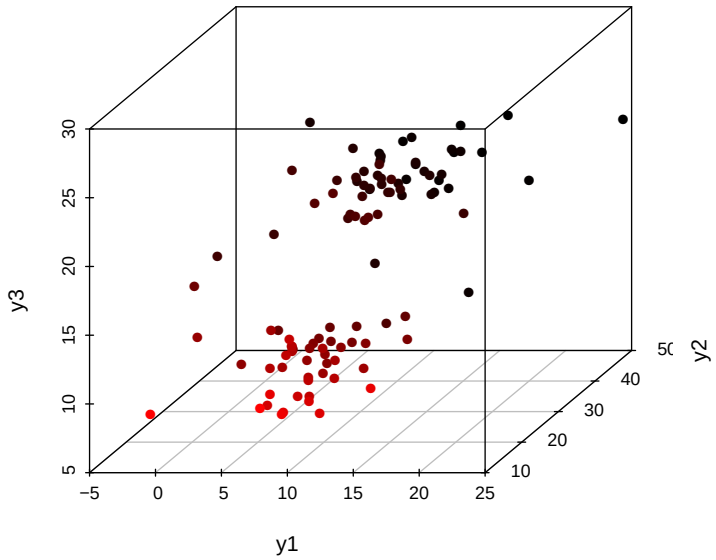


Figure 3.13: Financial data: 3D scatter-plot.

Similarly, we can note from Figure 3.14, which is a forward plot of minimum Mahalanobis distances among units not in the subset from 100 random starting points, that there are two clusters. Clearly, we can see two trajectories with high values (outliers) around $m = 50$, one of them with twin peaks and the other has one only.

Now we consider the forward search method based on the volume of central rank regions. Figure 3.15 is a forward plot of volume of central rank regions among units not in the subset from 100 random starting points with 1% and 99% envelopes. It can be clearly seen that there are two clear peaks in the plot at $m = 44$ and 56 suggesting two clusters.

For more investigation, we tried to analyze every cluster separately. We started with the first trajectory which represents the first cluster, and stopped at the maximum value of the volume of central rank regions ($vol(m)$) corresponds to each subset (m), as shown in the right panel of Figure 3.16. In this trajectory we can see that during the search, the

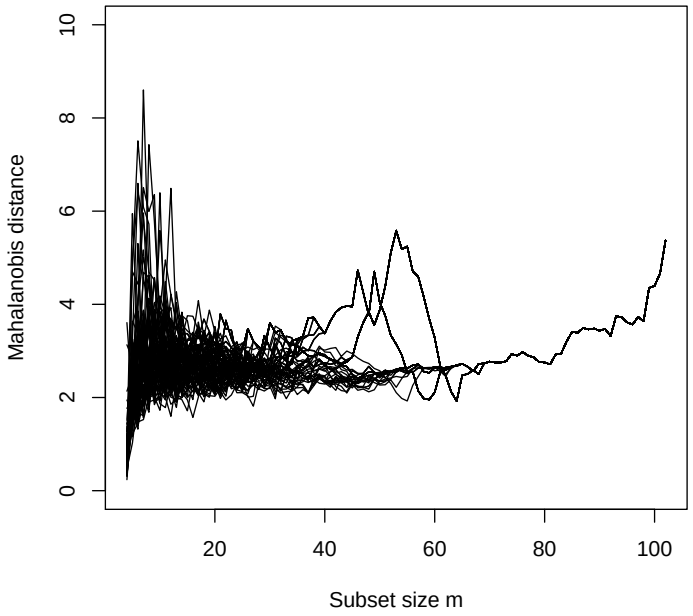


Figure 3.14: Financial data: forward plot of minimum Mahalanobis distances among units not in the subset from 100 random starts. Two clusters are evident around $m = 50$.

highest value of the volume of central rank regions occurs when the subset size equal to 56, we can check this from Table 3.1 which gives the volume of central rank regions for each subset (m) regarding to the first cluster, where the yellow shaded value, $vol(56) = 1904.01$, is the highest one. Moreover, if we started with different points and consequently with different subsets, most of trajectories for this second cluster will take the same pass starting from subset number 33 which has a volume value, $vol(33) = 161.544$. This appears as the blue shaded distance in Table 3.1. The left panel of Figure 3.16 is a scatterplot of y_1 against y_2, where the red points refer to the units in cluster 1, and green points are unassigned units.

The left-hand panel of Figure 3.17 is a scatterplot of y_1 against y_2, also the red points in this case mean that all the units entered the search and it is the end of it. The right-hand panel of the Figure shows the forward plot of volume of central rank regions amongst

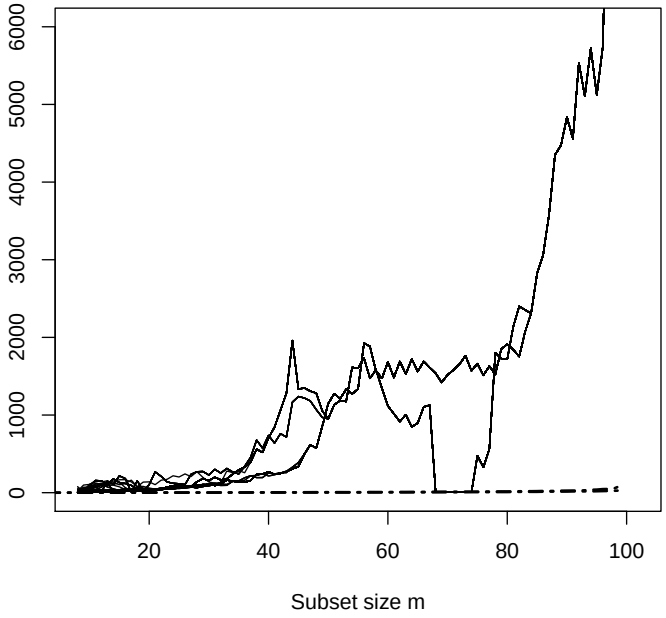


Figure 3.15: Financial data: forward plot of volume of central rank regions among units not in the subset from 100 random starts with 1% and 99% envelopes. Two clusters are evident at $m = 44$ and 56.

units not in the subset at the end of search ($m = 102$) which has a peak at $m = 56$. If we take the 55 units before the peak in this trajectory, we can get the first cluster. Table 3.2 gives the units that have been grouped in the first cluster. The units from 1 to 56 are stock funds, which means we have roughly found the reasonable clustering, where it can be clearly seen from Table 3.2 that the first cluster is including all the units from 1 to 55 except the units number 10, 20, 37, 39, 50, 52 and 54. These units are undecided units and remain to be clustered. Initially, it is acceptable and reasonable result, but we still need to investigate the units that are belonging to the second cluster.

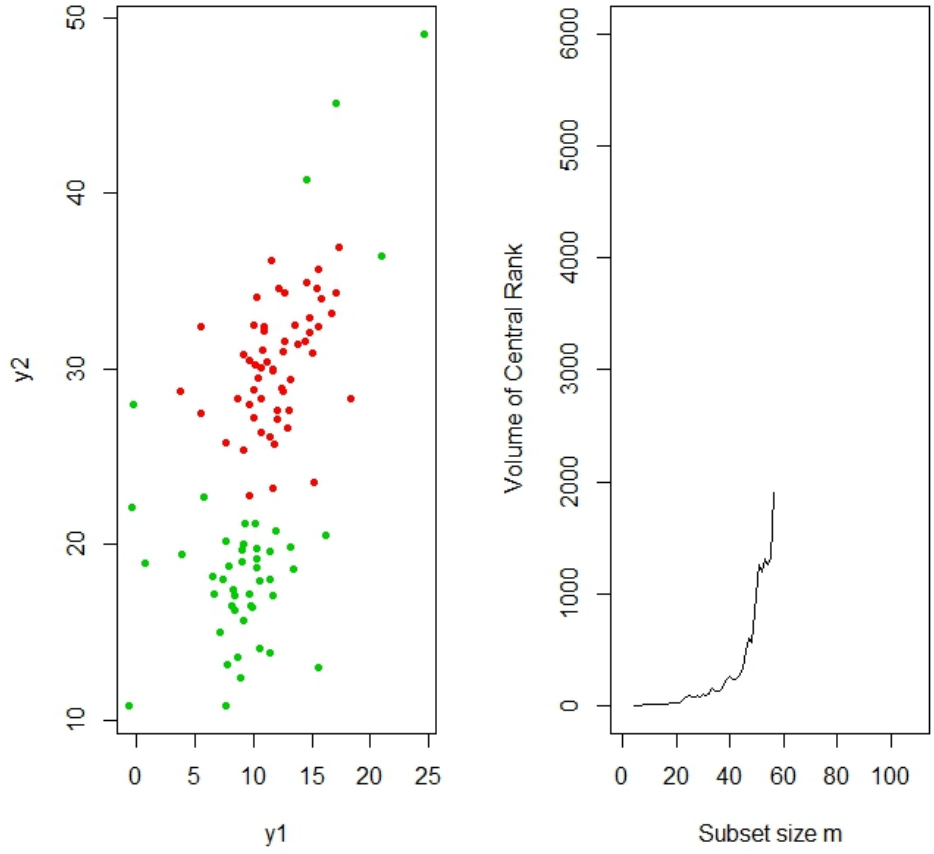


Figure 3.16: Financial data: Cluster 1: left panel, scatterplot y_1 vs y_2, red points are the units in cluster 1, and green points are unassigned units; right panel, forward plot of volume functional of central rank regions at the first peak $m = 56$.

Table 3.1: Financial data: The volume of central rank regions for each subset (m) for Cluster 1

m	$vol(m)$	m	$vol(m)$	m	$vol(m)$	m	$vol(m)$	m	$vol(m)$
4	0.0719617	26	87.05440	48	567.6713	70	1497.523	92	5461.964
5	135.8755000	27	83.87361	49	841.5898	71	1551.585	93	5036.461
6	110.8953000	28	97.71099	50	1130.6070	72	1625.693	94	5643.386

Continued

Table 3.1 – *Continued*

m	$vol(m)$	m	$vol(m)$	m	$vol(m)$	m	$vol(m)$	m	$vol(m)$
7	85.5589800	29	88.33577	51	1259.4540	73	1736.155	95	5051.731
8	92.6583000	30	111.29430	52	1194.9950	74	1533.900	96	5657.606
9	78.3433100	31	91.08209	53	1315.4720	75	1642.381	97	8434.414
10	24.9538500	32	111.73520	54	1256.2410	76	1482.590	98	10226.270
11	14.8005700	33	161.54400	55	1319.6020	77	1607.370	99	11550.110
12	14.5966000	34	146.12570	56	1904.0100	78	1503.613	100	14206.020
13	12.9266700	35	138.56190	57	1861.9580	79	1826.217	101	32669.900
14	11.4619900	36	133.77460	58	1551.5990	80	1887.760	102	54312.470
15	18.3696500	37	154.92370	59	1456.3800	81	1817.943		
16	20.7959200	38	230.95390	60	1664.0910	82	1730.348		
17	21.5905700	39	236.48800	61	1466.5730	83	2046.090		
18	23.7899700	40	264.54790	62	1671.7770	84	2276.736		
19	25.0441700	41	234.39340	63	1510.9080	85	2784.036		
20	26.2108100	42	245.10590	64	1703.2300	86	3011.890		
21	36.0838900	43	269.66800	65	1543.9850	87	3531.806		
22	49.2971700	44	301.87910	66	1671.7620	88	4290.074		
23	70.6884200	45	332.02540	67	1592.9330	89	4411.096		
24	88.4231800	46	490.70320	68	1520.4910	90	4770.362		
25	93.5353400	47	607.27360	69	1399.6860	91	4495.690		

Now, we consider the second trajectory which represents the second cluster. We stopped again at the maximum value of the volume of central rank regions ($vol(m)$) as shown in the right-hand panel of Figure 3.18. From this Figure we can see that there is a first peak at $m = 44$, where the volume of central rank regions at this time is very high,

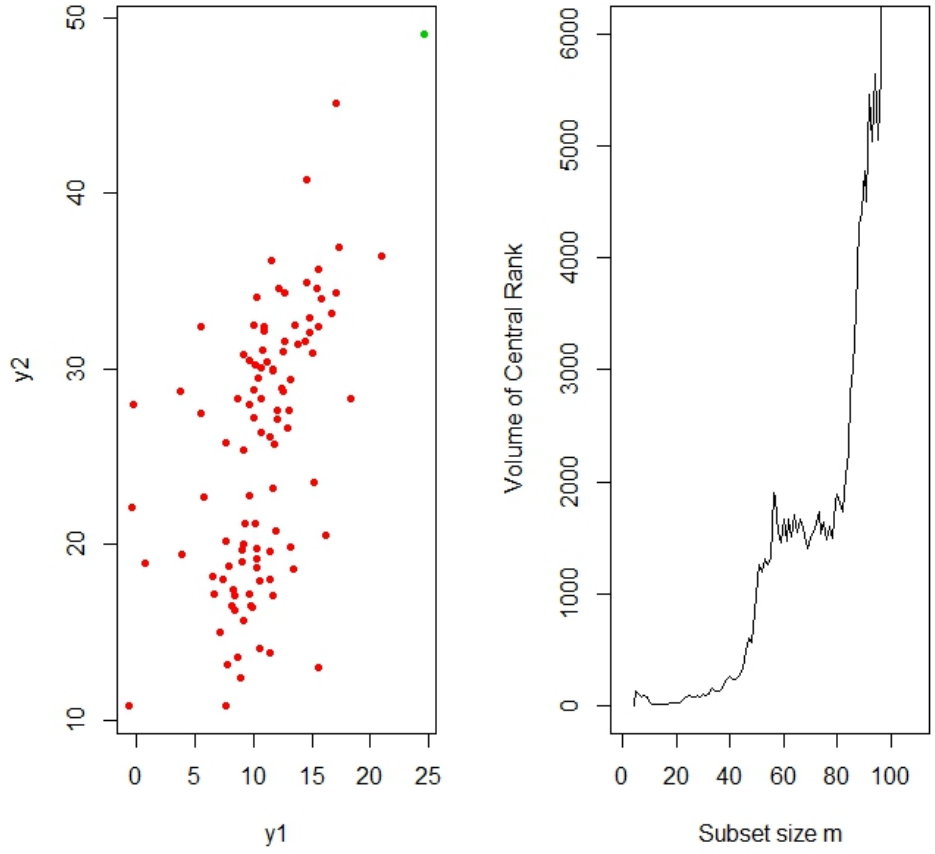


Figure 3.17: Financial data: Cluster 1: left panel, scatterplot y_1 vs y_2; right panel, forward plot of volume of central rank regions at $m = 102$.

$vol(m) = 1931.516$, as shown in Table 3.3 which gives the volume of central rank regions for each subset (m) for the second cluster, where the first yellow shaded distance represents the first peak. Also, if we started with different points and consequently with different subsets, most of trajectories for this second cluster will take the same pass starting from subset number 26 which has a volume, $vol(26) = 87.08442$. This appears as the blue shaded value in Table 3.3. The left panel of Figure 3.18 is a scatterplot of y_1 against y_2, where the red points refer to the units in cluster 2, and green points are unassigned units.

We considered the end of search for this trajectory at $m = 102$, as shown in Figure

Table 3.2: Financial data: The assigned units in the first cluster by using forward search based on volume of central rank regions (55 units).

47	45	17	36	23	5	28	24	1	31	4	40	13	27
51	7	42	48	2	56	15	22	19	26	46	9	18	3
49	8	53	12	25	34	32	29	6	33	89	16	55	38
43	11	35	30	41	90	44	21	14	68	70	77	71	

3.19, which has a peak at $m = 44$, followed by a second similar sized peak at $m = 56$ which means also that some remote observations are entering in this region. The right panel of Figure 3.19 is a forward plot of volume of central rank regions amongst units not in the subset at the end of search ($m = 102$). It clearly shows that there is a second peak at $m = 56$ again, where the volume of central rank regions at this subset is a maximum, $vol(56) = 1704.919$, as shown in Table 3.3, the second yellow shaded distance represents this second peak. The left panel is a scatterplot of y_1 against y_2, also the red points in this case mean that all the units entered the search and it is the end of it. Similarly, if we take the 43 units before the first peak in this trajectory, we can get the second cluster. Table 3.4 gives the units that have been grouped in the second cluster. The units from 57 to 103 are balanced funds, which means we have roughly again found the reasonable clustering, where it can be clearly seen from Table 3.4 that the second cluster is including all the units from 57 to 103 except only the unit number 80 that remains to be clustered.

Unlike Table 3.2, Table 3.4 includes the units 77 and 89 which means that they have been clustered at this time. We took the 55 units before the peak in the first trajectory and likewise the 43 units before the first peak in the second trajectory, with three units have been incorporated in both clusters, they are 68, 70 and 71 and they could belong to either cluster, grey shaded units in Tables 3.2 and 3.4. If we considered that these three clusters are belonging to the second cluster according to their positions in the scatterplot, and that agrees with the nature of data since the units from 57 to 103 are balanced funds as we mentioned before, thus, out of 103 units, 95 are clustered and 8 are remote from

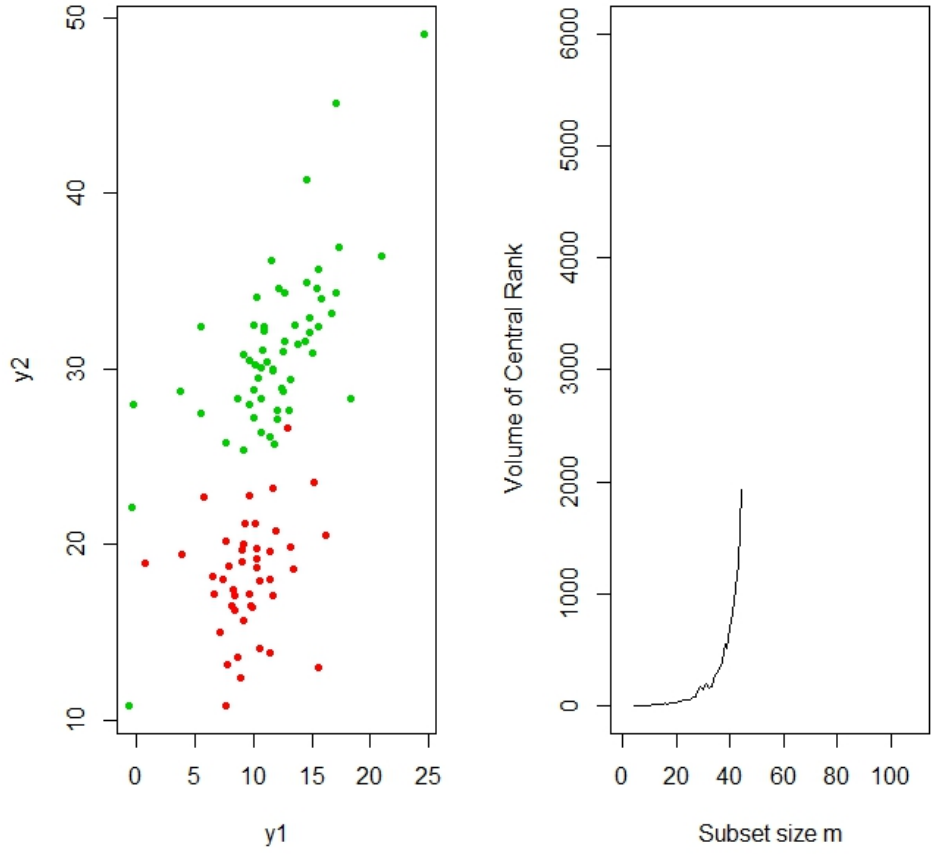


Figure 3.18: Financial data: Cluster 2: left panel, scatterplot y_1 vs y_2, red points are the units in cluster 2, and green points are unassigned units; right panel, forward plot of volume functional of central rank regions at the first peak $m = 44$.

either cluster. In other words, first group consists of 55 observations, from unit 1 to unit 56 less seven undecided, and the second group consists of 43 observations, from unit 57 to unit 103, less three belonging to cluster 1. So we now know which units belong to the first and second clusters.

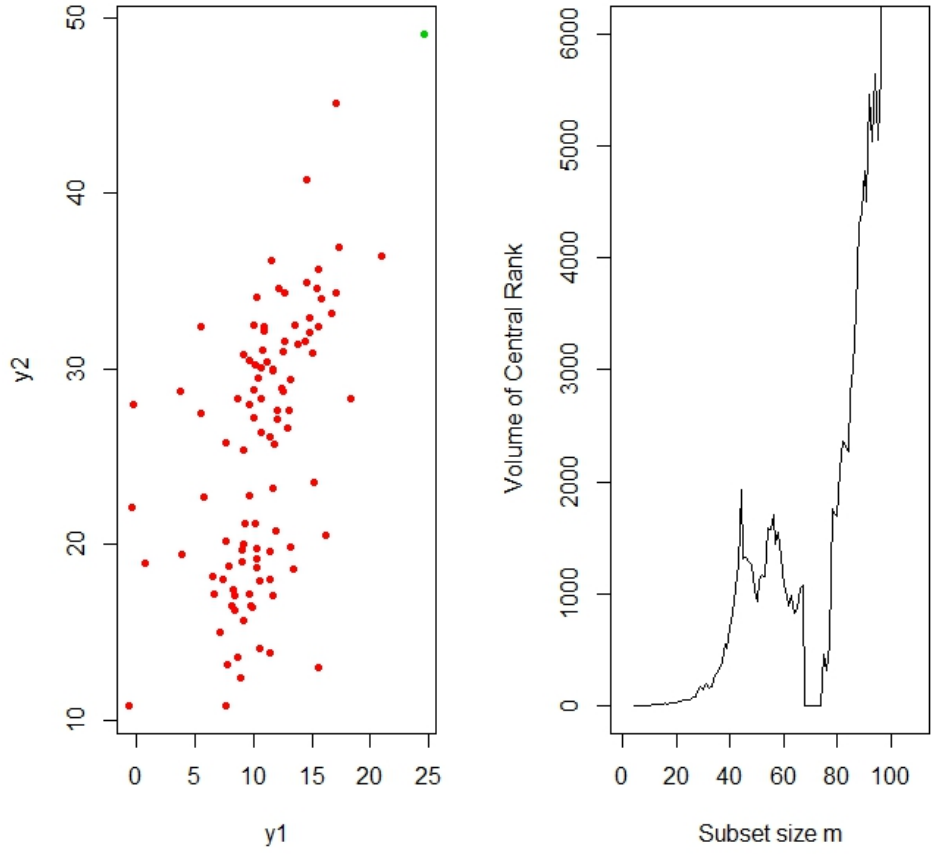


Figure 3.19: Financial data: Cluster 2: left panel, scatterplot y_1 vs y_2; right panel, forward plot of volume functional of central rank regions at $m = 102$.

Table 3.3: Financial data: The volume of central rank regions for each subset (m) for Cluster 2

m	$vol(m)$	m	$vol(m)$	m	$vol(m)$	m	$vol(m)$	m	$vol(m)$
4	0.4772787	26	87.08442	48	1260.830000	70	2.808459	92	5461.964
5	1.8012360	27	77.72739	49	1036.742000	71	2.426188	93	5036.461
6	2.1085420	28	138.37920	50	933.862500	72	2.416222	94	5643.386
7	3.4626790	29	179.41430	51	1112.098000	73	2.602631	95	5051.731

Continued

Table 3.3 – *Continued*

m	$vol(m)$	m	$vol(m)$	m	$vol(m)$	m	$vol(m)$	m	$vol(m)$
8	8.5510760	30	152.53420	52	1166.376000	74	4.462318	96	5657.606
9	9.1133460	31	197.73910	53	1153.281000	75	465.450200	97	8434.414
10	9.4057640	32	162.90650	54	1586.840000	76	324.447300	98	10226.270
11	11.4020600	33	176.16300	55	1577.416000	77	555.739500	99	11550.110
12	20.0848000	34	236.03560	56	1704.919000	78	1763.363000	100	14206.020
13	19.5499300	35	288.04900	57	1447.157000	79	1702.002000	101	32669.900
14	21.5188100	36	325.72630	58	1556.648000	80	1693.037000	102	54312.470
15	20.7002000	37	390.37050	59	1322.775000	81	2105.913000		
16	23.6334400	38	553.06720	60	1075.426000	82	2366.689000		
17	21.9519200	39	511.84190	61	989.384000	83	2321.202000		
18	32.2182700	40	695.82850	62	892.998100	84	2276.736000		
19	31.1849000	41	832.20280	63	985.487900	85	2784.036000		
20	33.4025300	42	1048.10500	64	829.677300	86	3011.890000		
21	46.7386500	43	1264.99000	65	867.188300	87	3531.806000		
22	44.4261900	44	1931.51600	66	1057.223000	88	4290.074000		
23	49.6965700	45	1310.49600	67	1077.354000	89	4411.096000		
24	53.9726500	46	1325.88100	68	4.612333	90	4770.362000		
25	57.3836100	47	1286.75100	69	2.826030	91	4495.690000		

Table 3.4: Financial data: The assigned units in the second cluster by using forward search based on volume of central rank regions (43 units).

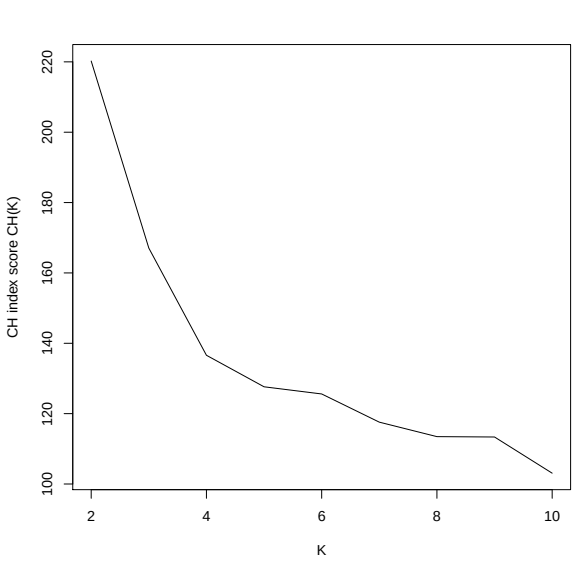
76	61	63	102	81	92	99	75	74	91	65
69	86	94	79	66	98	87	84	103	85	101
62	93	100	58	88	83	72	82	64	71	78
60	70	97	67	95	73	59	68	57	96	

Atkinson et al. (2004) clarified the economic interpretation of the data. They mentioned that the undecided units show different features, which can explain their behavior along the forward search. They pointed out that the Italian Stock Exchange experienced a remarkable increase in most stock prices at that time. Also there were positive short and medium term performances of many funds, especially so for stock funds. Stock funds also exhibited higher volatility, which is synonymous with higher risk.

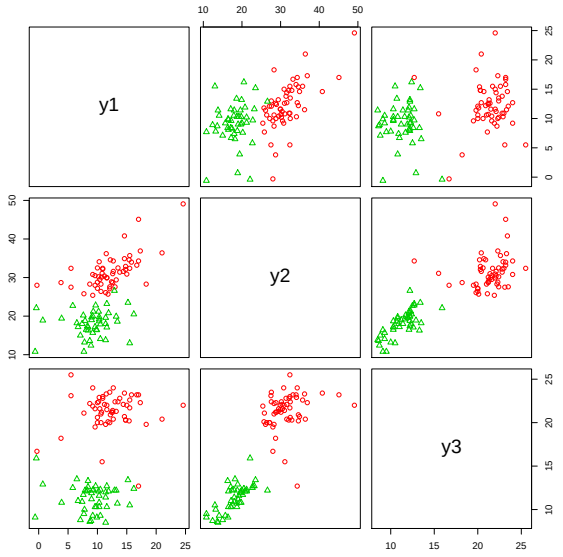
For the K-means, the selection criterion that we use is again the CH-index (Calinski and Harabasz, 1974), where we use it to estimate the number of clusters that K-means algorithm should start with it, and for the BIC criterion we use the mclust library (Fraley and Raftery, 2003), where Fraley and Raftery (2003) assumed 10 models of the parameterization of the Gaussian mixture models (parsimonious models) introduced earlier by Banfield and Raftery (1993). According to Figure 3.20, like our method, K-means indicated two clusters, while the mclust approach based on BIC again failed to give the true number of clusters, where the maximum value of the BIC criterion was for the EEE model (BIC=-1664.278).

3.7.2 Old Faithful data

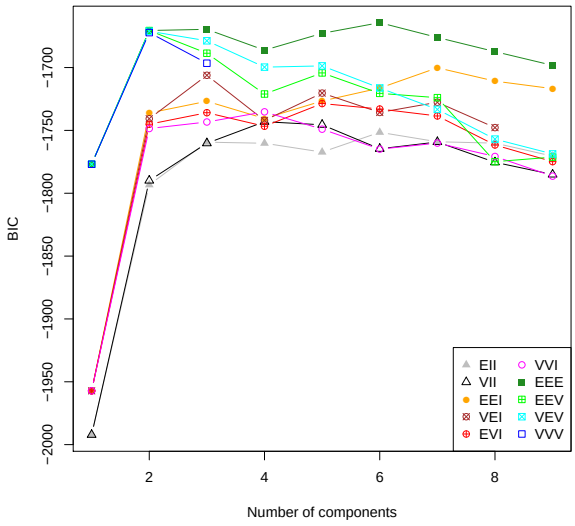
The second dataset we consider is known as the Old Faithful Geyser Data, which are taken from Azzalini and Bowman (1990) and the MASS library Venables and Ripley (2002). This data gives the waiting time between eruptions and the duration of the eruption in minutes for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA, with two apparent groups in the data. The analysis of this data using the standard forward approach based on Mahalanobis distances had been done in Atkinson and Riani (2012). It includes 272 observations with two variables, x_{1i}: the duration of the $i-th$ eruption and x_{2i}: the waiting time to the start of that eruption from the start of eruption $i-1$. Figure 3.21 gives the scatter-plot of old faithful data. Initially, from the scatter-plot we



(a) CH index



(b) K-means clustering



(c) BIC based on mclust

Figure 3.20: Financial data: (a) CH index suggests $K = 2$, (b) K-means with 2 clusters, and (c) BIC plot suggesting 6 clusters with best BIC values for EEE model.

can clearly see that there are two clusters with some observations between them.

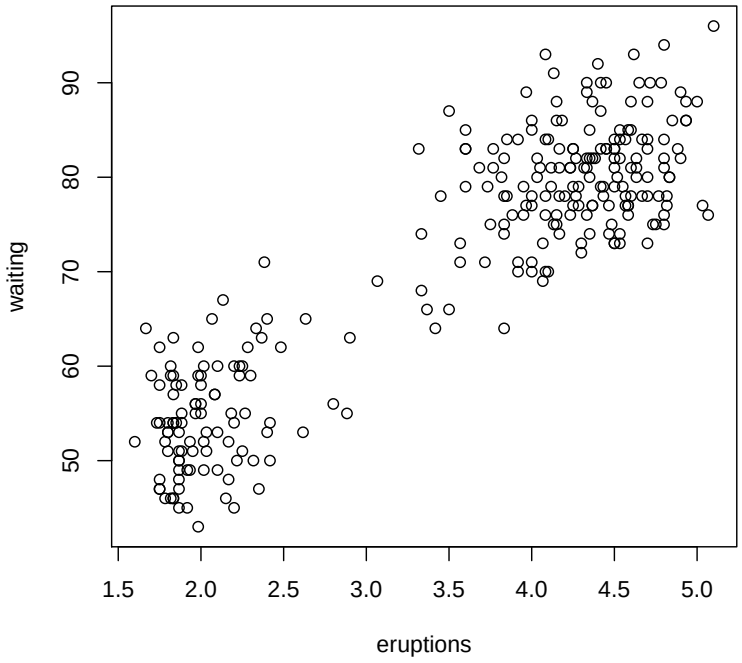


Figure 3.21: Old faithful data: Scatter-plot

Figure 3.22 shows the forward plot of volume of central rank regions among units not in the subset from 100 random starts for old faithful data. As we can see, there are two clear maxima in this plot, one at $m = 105$ and the other at $m = 179$, suggesting the existence of two clusters, which gives the right number of groups in the data with their sizes.

Figure 3.23 shows the behavior of the CH-index, K-means, and BIC. Panels (a) and (b) of Figure 3.23 are the CH-index plot which indicates ten clusters and the clustering with K-means respectively. Clearly, the K-means behaves so poorly in this real dataset, where it failed to give us the right clustering. On the other hand, from panel (c) we can see that the best model according to BIC is an equal-covariance model with three clusters, where the maximum value of the BIC criterion among the 10 parsimonious models was

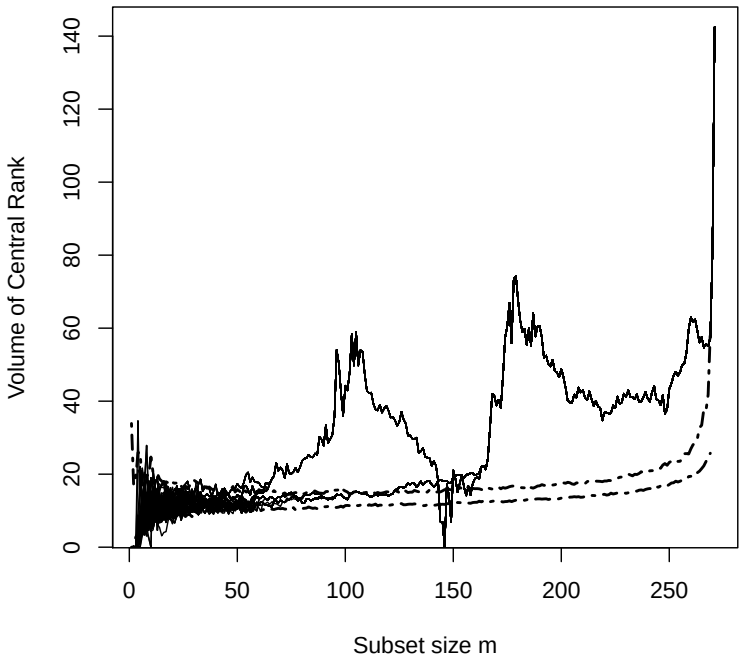
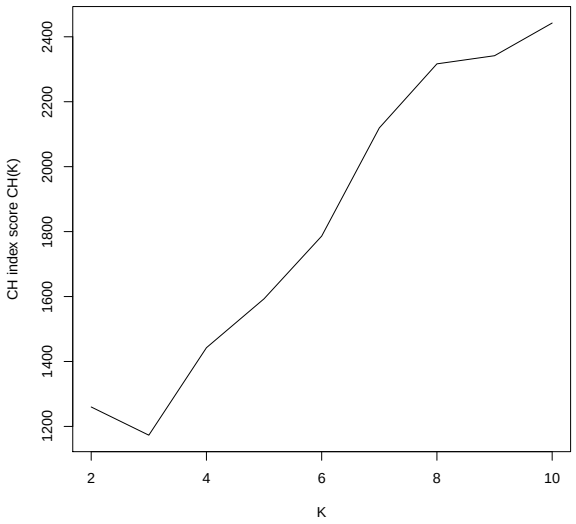


Figure 3.22: Old faithful data: forward plot of volume of central rank regions among units not in the subset from 100 random starts with 1% and 99% envelopes. Two clusters are evident at $m = 105$ and 179.

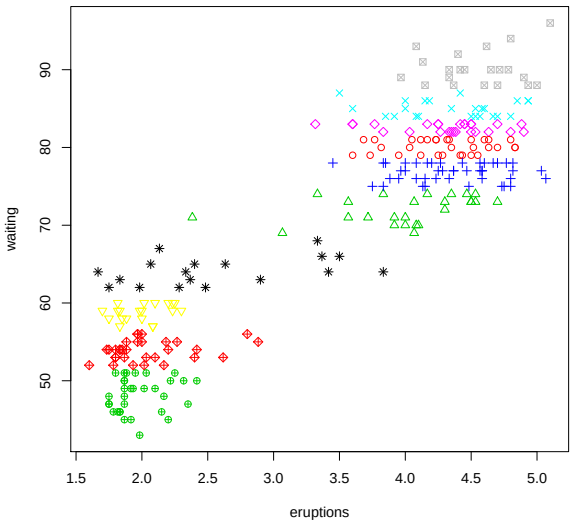
for the EEE model (BIC=-2314.386) with similarly shaped covariance matrices, and the next best model had four clusters (BIC=-2320.207) with the same covariance structure. This indicates that the mclust approach based on BIC criterion failed to give the right number of clusters as well as K-means method. The forward plot in Figure 3.22 in fact outperforms K-means and mclust approach in this data, where it gave the right number of groups.

3.7.3 Iris data

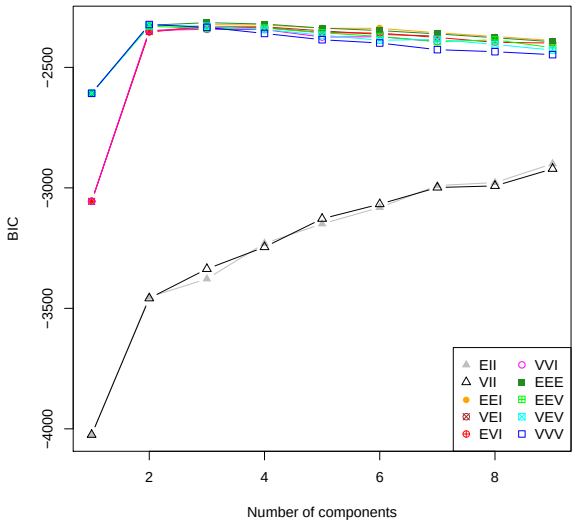
The third real dataset used in this Chapter, is the iris data (Fisher, 1936). In iris data, one of the clusters contains iris setosa, while the other cluster contains both iris virginica and iris versicolor. In clustering analysis, most of the clustering techniques consider this



(a) CH index



(b) K-means clustering



(c) BIC based on mclust

Figure 3.23: Old faithful data: (a) CH index suggests $K = 10$, (b) K-means with 10 clusters, and (c) BIC plot suggesting 3 clusters with best BIC values for EEE model.

data includes 2 groups, this is due to both iris virginica and iris versicolor is not separable without the species information that Fisher used, which consider a good example to explain the difference between supervised and unsupervised methods. Figure 3.24 is a scatter-plot matrix of the iris data which shows that there are two clusters with some observations between them.

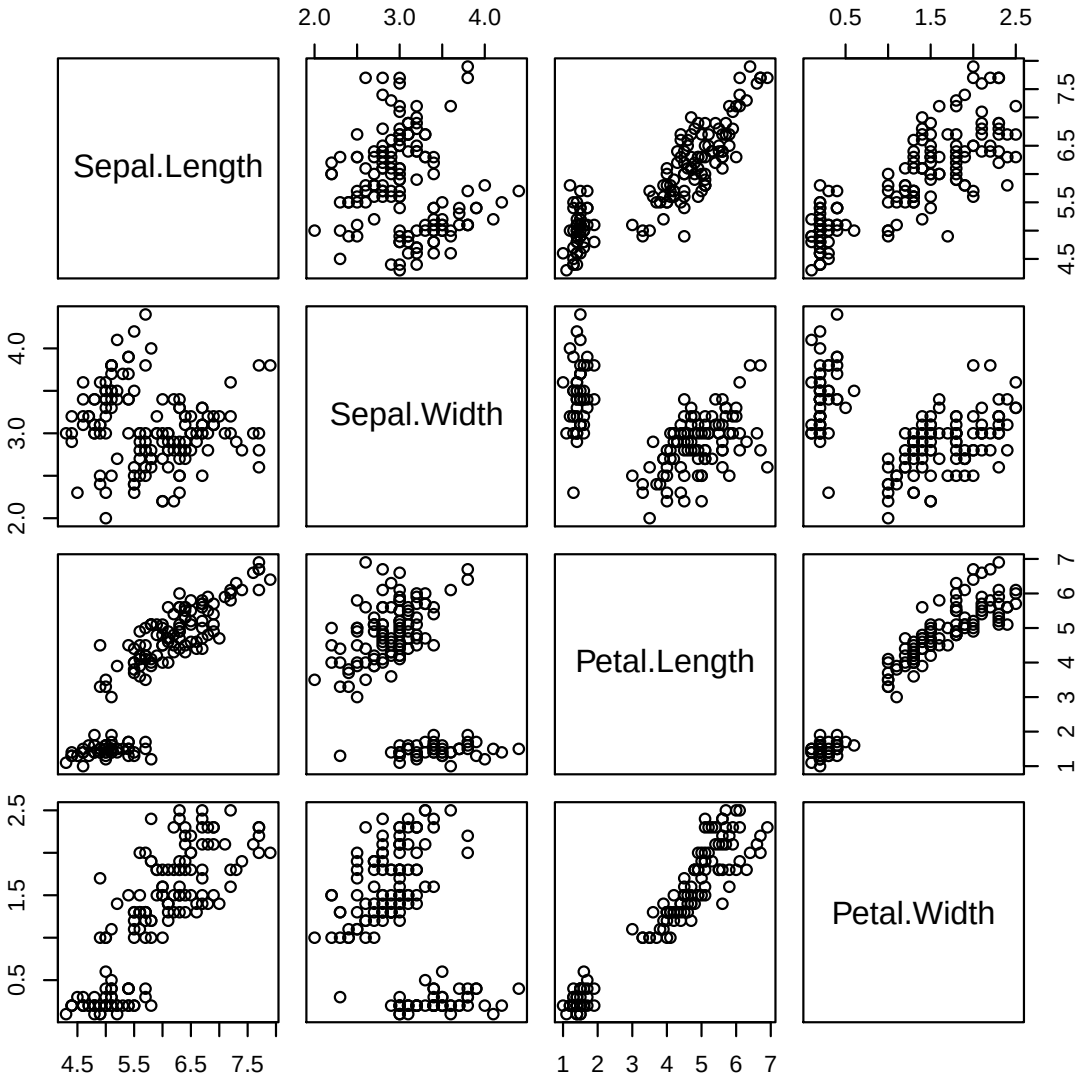


Figure 3.24: Iris data: Scatter-plot matrix

Figure 3.25 shows the forward plot of volume of central rank regions among units not in the subset from 100 random starts for the iris data. As we can see, there are two clear maxima in this plot, one at $m = 50$ and the other at $m = 100$, suggesting the existence of two clusters, with sizes 50 and 100.

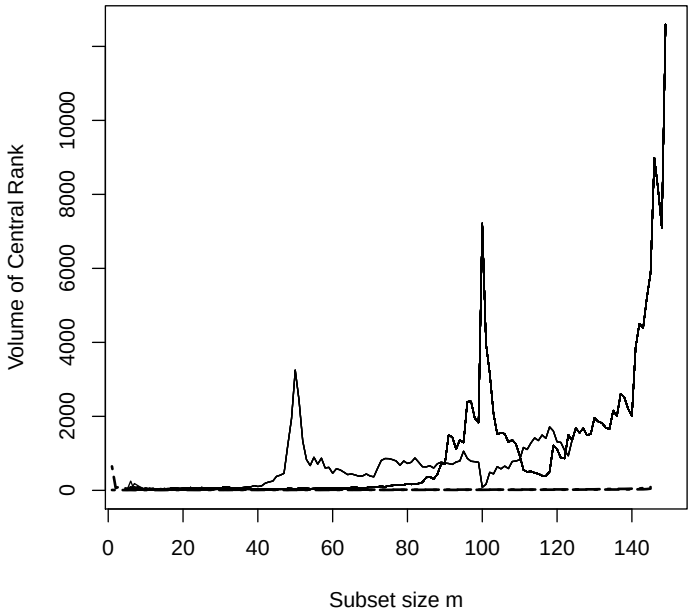
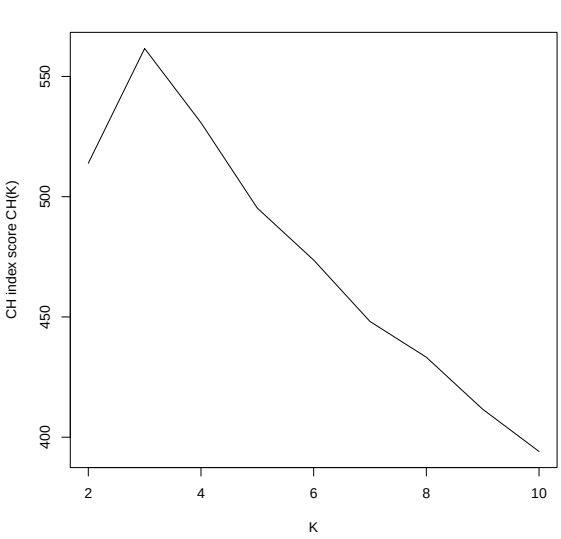
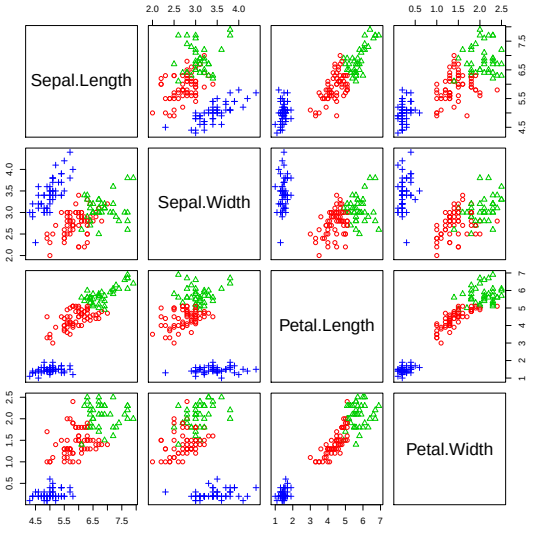


Figure 3.25: Iris data: forward plot of volume of central rank regions among units not in the subset from 100 random starts with 1% and 99% envelopes. Two clusters are evident at $m = 50$ and 100.

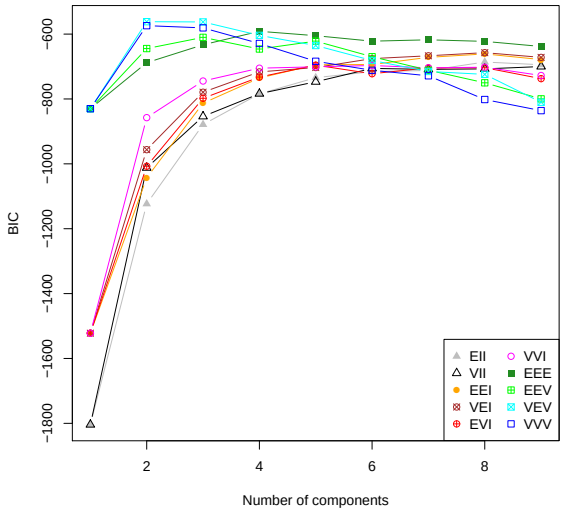
Figure 3.26 gives the clustering of iris data based on the K-means and mclust. Panel (a) of Figure 3.26 gives the CH-index plot which indicates three clusters, while panel (b) gives the K-means clustering with the three groups. Panel (c) shows that the best model according to BIC is an ellipsoidal equal-shape model with two clusters, where the maximum value of the BIC criterion among the 10 parsimonious models was for the VEV model (BIC=-561.7285), which agree with our method.



(a) CH index



(b) K-means clustering



(c) BIC based on mclust

Figure 3.26: Iris data: (a) CH index suggests $K = 3$, (b) K-means with 3 clusters, and (c) BIC plot suggesting 2 clusters with best BIC values for VEV model.

Chapter 4

Clustering Multivariate Data Using Weighted Spatial Ranks

4.1 Introduction

One of the most important properties of the traditional spatial ranks function that have been discussed earlier is that, they tell us how central each observation is and in which direction it is moving from the centre. However they do not consider the distances between the observations and each other. On the other hand, the weighted spatial rank functions that are discussed in this chapter take in consideration the distances between the observations and each other as weights, which makes it easier to group and partition a given set of data or objects into specific number of clusters based on the homogeneity assumption such that the objects belong to the same cluster should be as similar as possible.

In this chapter, we propose a new nonparametric cluster detection methodology based on different weighted spatial rank (WSR) functions. Our proposed weighted spatial ranks are completely data-driven and easy to compute without any need to parameter estimates of the underlying distributions, which make them robust against distributional assumptions. Moreover, the weighted spatial ranks are more accurate in the purpose of intuitive

visualization since we can easily determine the number of clusters from the weighted ranks contours for a low-dimensional input space, using dimension reduction that allow to map the data cloud. This is due to the weighted spatial ranks can capture the clusters structure because of considering the weights as a similarity (dissimilarity) measure, where the weight value of each observation presents the distance between it to the others. The main idea behind WSR is to define a dissimilarity measure locally based on a localized version of multivariate ranks. As a result, the proposed method can be used to determine the assumed number of clusters, and to assign each observation to its cluster consequently by using the weighted ranks as a classifier and confirmatory tool.

Selection of a proper weight function will lead to better identification of clusters when the data do not follow any standard parametric distribution. We have considered parametric and nonparametric weights for comparison. One of the most popular weights functions is the kernel weights. In pattern analysis, over the last decade a particular attention is paid to the use of kernels in classification, cluster analysis, machine learning and support vector machines (Hofmann et al., 2008). Many different kernel weight functions have been considered in this study. Moreover, we have introduced some weighted multivariate spatial ranks based on different robust weights such as the generalized Mallow weights function that has been introduced by Simpson et al. (1992) and adjusted by Naranjo and Hettmansperger (1994).

This chapter is organized as follows. Section 4.2 gives a brief review of the parametric and nonparametric weights functions, where we consider some kernel and robust weights. Section 4.3 introduces the proposed weighted spatial rank functions with some numerical examples and comparisons with other standard parametric and nonparametric methods. In Section 4.4, we propose a confirmatory classifier based on weighted ranks that can be used to properly assign the observations to specific cluster. Section 4.5 demonstrates the weighted rank based clustering algorithm. Finally, in Section 4.6, we give some numerical

examples based on both simulated and real datasets to show the performance of the proposed algorithm.

4.2 Parametric and Nonparametric Weights Functions

Two different types of weights functions have been considered in this study. The first one is a nonparametric, which depends on many different kernel functions like Gaussian, triangular, uniform, Laplacian, and logistic kernels. The second is a parametric, and it is related to the weights functions to be used for some robust statistics such as the generalized Mallow weights (Simpson et al., 1992) and Naranjo and Hettmanspergers' weights (Naranjo and Hettmansperger, 1994).

4.2.1 Kernel Weights Functions

Recently, the kernel functions and methods have raised researchers' concerns regarding their helpful use, especially the positive definite kernels of a reproducing kernel Hilbert space. They are usually used to improve many different algorithms like support vector machines, Bayes point machines, kernel principal component analysis, and Gaussian processes to solve problems of classification, regression, density estimation, and clustering. A good overview of the usage of kernel is given in (Genton, 2001; Vapnik, 1995; Cristianini and Shawe-Taylor, 2000). It is worth mentioning in this context that, there are many different kernel functions, like Gaussian, exponential, Laplacian, logistic, triangular, uniform, Epanechnikov, linear, polynomial, rational quadratic, multiquadric, spherical, Cauchy, and chi-square kernels. For more comprehensive details about the other kernels, the reader is referred to Souza (2010). In this study, we considered the following 6 kernel functions. The first one is the Gaussian kernel function,

$$k(x) = e^{-x^2/2}, \tag{4.2.1}$$

the second function is the Laplacian kernel function,

$$k(x) = e^{-|x|}, \tag{4.2.2}$$

the third function is the logistic kernel function,

$$k(x) = \frac{1}{e^x + 2 + e^{-x}}, \tag{4.2.3}$$

the fourth function is the triangular kernel function,

$$k(x) = \begin{cases} 1 - |x| & \text{if } |x| \leqslant 1 \\ 0 & \text{otherwise ,} \end{cases} \tag{4.2.4}$$

the fifth function is the uniform kernel function,

$$k(x) = \begin{cases} \frac{1}{2} & \text{if } |x| \leqslant 1 \\ 0 & \text{otherwise ,} \end{cases} \tag{4.2.5}$$

and the last function is the Epanechnikov kernel function (Epanechnikov, 1969),

$$k(x) = \begin{cases} \frac{3}{4}(1 - x^2) & \text{if } |x| \leqslant 1 \\ 0 & \text{otherwise ,} \end{cases} \tag{4.2.6}$$

where x in the above kernel functions has been considered as the Euclidean norm of the distance between the observation x and all the other observations $\mathbf{X}_i$, i.e. $\|\mathbf{x} - \mathbf{X}_i\|$, where $\|\mathbf{x}\|$ is the Euclidean norm such that; $\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + ... + x_d^2}$, the direction of the d-dimensional vector $\mathbf{x}$.

4.2.2 Robust weight Functions

One of the popular weights functions to be used for robust statistics is the generalized Mallow weights which was proposed by Simpson et al. (1992) and adopted by many authors like Naranjo and Hettmansperger (1994) and Chang et al. (1999). An important feature of this function is that it reduces the sensitivity of the estimate to leverage points compared to other weights. The generalized Mallow weights can be written as:

$$k(\mathbf{x}) = \min\left\{1, \left(\frac{c}{(\mathbf{x}-\widehat{\mu})^t\widehat{\Sigma}^{-1}(\mathbf{x}-\widehat{\mu})}\right)^{r/2}\right\}, \tag{4.2.7}$$

where $\widehat{\mu}$ and $\widehat{\Sigma}$ are some robust estimates of location and scatter of the d-dimensional vector x. Such a choice for $\widehat{\mu}$ and $\widehat{\Sigma}$ might be the minimum volume ellipsoid (MVE) estimator (Rousseeuw and van Zomeren, 1990) or S estimator. In order to determine the value of c, one can use the fact that $(\mathbf{x}-\widehat{\mu})^t\widehat{\Sigma}^{-1}(\mathbf{x}-\widehat{\mu})$ has the distribution $\chi^2(d)$, so we can consider c as the 95th percentile of $\chi^2(d)$, and it is clear that with $r = 0$ we move to the unweighted case. For a discussion on the choice of r, the reader is referred to section 1 of Simpson et al. (1992).

On the other hand, Naranjo and Hettmansperger (1994) proposed their optimal robust weights,

$$k(\mathbf{x}) = \min\left\{1, \frac{c}{\left((\mathbf{x}-\widehat{\mu})^t\widehat{\Sigma}^{-1}(\mathbf{x}-\widehat{\mu})\right)^{1/2}}\right\}, \tag{4.2.8}$$

we used both of generalized Mallow weights and Naranjo and Hettmanspergers' weights in the weighted ranks function, to compare them with the proposed kernel weights, in order to determine the proper weight function that can improve the performance of the spatial ranks. In the next section, we present a numerical example to compare the performance of the weighted spatial ranks based on both the kernel and robust weights functions.

4.3 Weighted Spatial Rank Functions

In this section, we propose two different functions of the weighted spatial ranks based on the weight functions that we discussed in the previous Section. We start with giving definition of the WSR functions, then we compare them with other standard parametric distance methods.

Definition 4.3.1 *: The weighted spatial rank function:*
Suppose that $\mathbf{X} \in \mathbb{R}^d$ *has a d-dimensional distribution* F*, which is assumed to be absolutely continuous throughout this chapter, then the first weighted spatial rank function of the point* $\mathbf{x} \in \mathbb{R}^d$ *with respect to* F *can be defined as:*

$$WSR_{F_n}(\mathbf{x}) = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i Sign(\mathbf{x} - \mathbf{X}_i) = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i \frac{\mathbf{x} - \mathbf{X}_i}{\|\mathbf{x} - \mathbf{X}_i\|}, \tag{4.3.1}$$

and the second weighted spatial rank function can be defined as:

$$WSR_{F_n}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} w_i Sign(\mathbf{x} - \mathbf{X}_i) = \frac{1}{n} \sum_{i=1}^{n} w_i \frac{\mathbf{x} - \mathbf{X}_i}{\|\mathbf{x} - \mathbf{X}_i\|}. \tag{4.3.2}$$

and their L_2 *norm is:*

$$WSRN_{F_n}(\mathbf{x}) = \|WSR_{F_n}(\mathbf{x})\|. \tag{4.3.3}$$

On one hand, we can observe that, the weighted spatial rank function (4.3.1) is a local function, where the denominator $\sum_{i=1}^{n} w_i$ is a function of $\mathbf{x}$, such that $w_i = w_i(\mathbf{x})$, and this makes it a data-dependent function. This in fact explains why its contour plot has more than one centre as we will see in the next numerical example. Moreover, we observe that the unweighted function (traditional spatial ranks function (3.2.8)) is a global function, where the denominator n does not depend on $\mathbf{x}$, which makes it a data-independent function. On the other hand, the weighted spatial rank function (4.3.2) is a trade-off between the local function (4.3.1) and the global function (3.2.8), where its denominator

n is data-independent and its numerator $\sum_{i=1}^{n} w_i$ is data-dependent making it a trade-off between the local and global ones. This in fact explains why its contour plot has only one centre as we will see in the next numerical example, which makes the WSR in (4.3.2) better than WSR in (4.3.1) in terms of the clustering detection and visualization.

Using the weights functions that are defined in equations (4.2.1) to (4.2.8), we can write the weights (w_i) as following: if we consider the Gaussian kernel weights,

$$w_i = e^{-\|\mathbf{x}-\mathbf{X}_i\|^2/2}, \tag{4.3.4}$$

if we consider the Laplacian kernel weights,

$$w_i = e^{-\|\mathbf{x}-\mathbf{X}_i\|}, \tag{4.3.5}$$

if we consider the logistic kernel weights,

$$w_i = \frac{1}{e^{\|\mathbf{x}-\mathbf{X}_i\|} + 2 + e^{-\|\mathbf{x}-\mathbf{X}_i\|}}, \tag{4.3.6}$$

if we consider the triangular kernel weights,

$$w_i = \begin{cases} 1 - \|\mathbf{x} - \mathbf{X}_i\| & \text{if } \|\mathbf{x} - \mathbf{X}_i\| \leqslant 1 \\ 0 & \text{otherwise ,} \end{cases} \tag{4.3.7}$$

if we consider the uniform kernel weights,

$$w_i = \begin{cases} \frac{1}{2} & \text{if } \|\mathbf{x} - \mathbf{X}_i\| \leqslant 1 \\ 0 & \text{otherwise ,} \end{cases} \tag{4.3.8}$$

if we consider the Epanechnikov kernel weights,

$$w_i = \begin{cases} \frac{3}{4}(1 - \|\mathbf{x} - \mathbf{X}_i\|^2) & \text{if } \|\mathbf{x} - \mathbf{X}_i\| \leqslant 1 \\ 0 & \text{otherwise ,} \end{cases} \tag{4.3.9}$$

if we consider the generalized Mallow weights,

$$k(\mathbf{x}) = \min \left\{ 1, \left(\frac{c}{(\mathbf{x} - \mathbf{X}_i)^t \widehat{\Sigma}^{-1} (\mathbf{x} - \mathbf{X}_i)} \right)^{r/2} \right\}, \tag{4.3.10}$$

and if we consider Naranjo and Hettmansperger (1994) robust weights,

$$k(\mathbf{x}) = \min \left\{ 1, \frac{c}{\left((\mathbf{x} - \mathbf{X}_i)^t \widehat{\Sigma}^{-1} (\mathbf{x} - \mathbf{X}_i) \right)^{1/2}} \right\}. \tag{4.3.11}$$

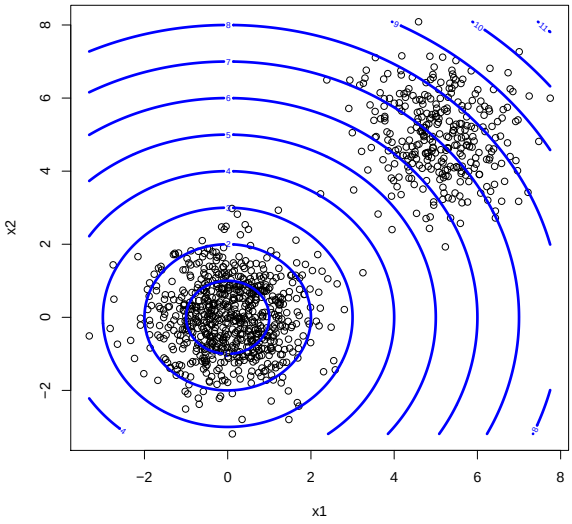
4.3.1 Numerical Examples and Comparison with Other Standard Methods

In order to highlight the efficiency of our proposed weighted ranks based clustering in terms of fitting the shape of clusters in the data cloud; we give this numerical example based on simulated data. We consider in this example a bivariate Gaussian mixture distribution with two clusters, mixture proportion ($p = 0.3$), and sample size ($n = 1000$), such that $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ is a random sample from bivariate mixture normal distribution:

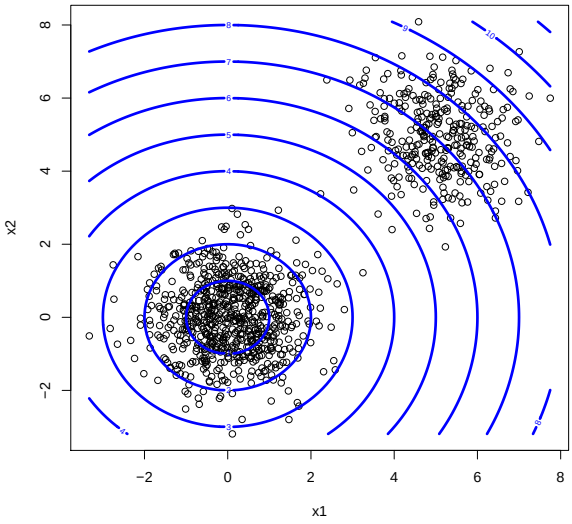
$$pN_2(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + (1 - p)N_2(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}), \tag{4.3.12}$$

where $\boldsymbol{\mu}_1 = (0, 0)^\top$, $\boldsymbol{\mu}_2 = (5, 5)^\top$ and $\boldsymbol{\Sigma} = \boldsymbol{I}$.

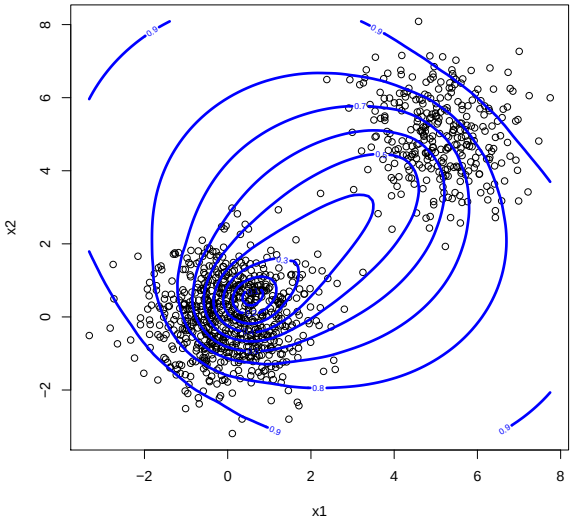
Figure 4.1 shows the contour plots of the Euclidean distances, Mahalanobis distances, spatial ranks, and spatial depth based on 1000 random observations from bivariate mixture



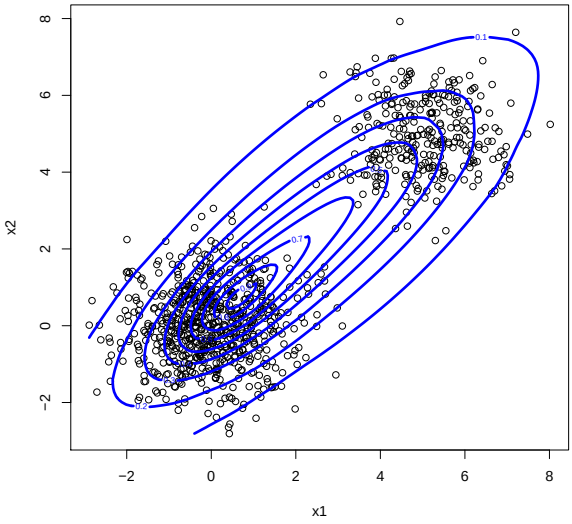
(a) Euclidean distances



(b) Mahalanobis distances



(c) Spatial ranks



(d) Spatial depth

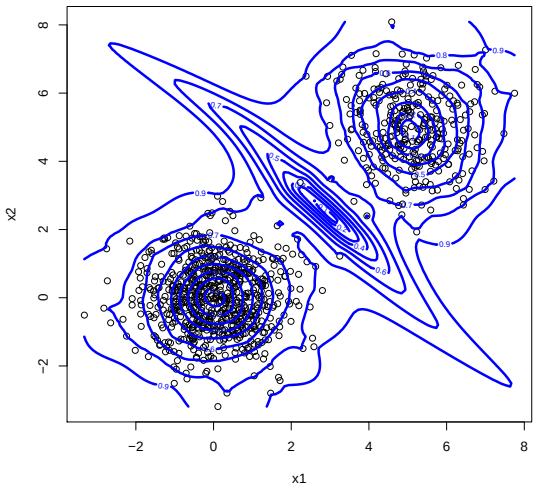
Figure 4.1: Simulated data example: Contour plots of (a) Euclidean distances, (b) Mahalanobis distances, (c) spatial ranks, and (d) spatial depth based on 1000 random observations from bivariate mixture normal distribution with two groups.

normal distribution with two groups. However the contours produced from the different four methods fit nicely to the shape of the data cloud, they failed to map the shape of the two present clusters structure.

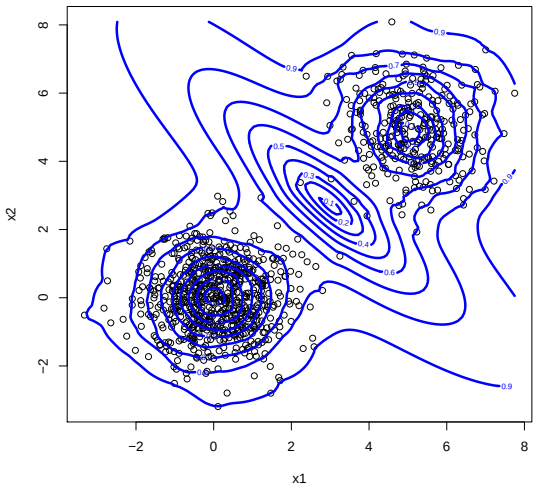
As we mentioned before, one of the advantages of the weighted spatial ranks is that, they are completely data-driven without any need to parameter estimates. Moreover, we can easily determine the number of clusters from the weighted ranks contours that allow to map the data cloud. We consider now the weighted ranks contours produced from the weighted rank function that is defined in (4.3.1).

Figure 4.2 gives the contour plots of the weighted spatial ranks, using Gaussian, Laplacian, logistic, triangular, uniform, and Epanechnikov kernel weights, based on the same simulated data that have been considered above. It can be clearly seen that on each contour, the weighted spatial rank function is constant with the indicated value. The weighted ranks values increase outward from the center (i.e., the spatial median) of the cloud, which indicates that a point with a high weighted rank value is easily to be classified in one group than a point with a low weighted rank value, which can be assigned in any of the two clusters. This in fact sheds the light on the possibility of using the weighted ranks to identify the expected number of clusters.

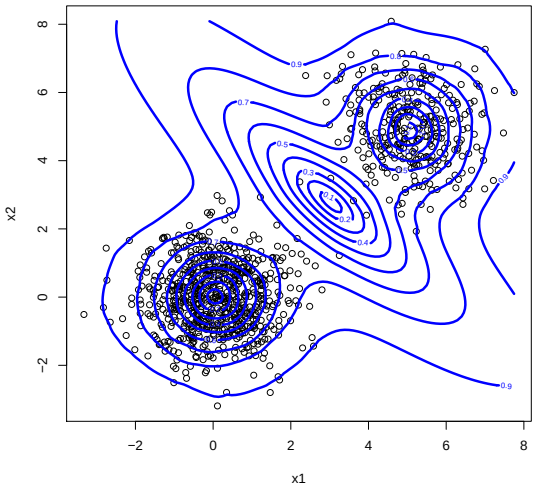
Compared to Figure 4.1, we can notice from Figure 4.2 that, the weighted ranks based on the different kernel functions fit nicely to the shape of the data cloud and to the shape of the two present clusters as well. However, the weighted ranks contours based on Gaussian, Laplacian and logistic kernels, gave some lines between the two clusters, suggesting the potential presence of a third cluster, due to the presence of some observations that are not close to either cluster centre, and can be classified either in third cluster or in one of the two groups. On the other hand, the weighted ranks contours based on triangular, uniform and Epanechnikov kernels gave an indicator of the right number of clusters, however the two groups' contours are very close to each other.



(a) Gaussian

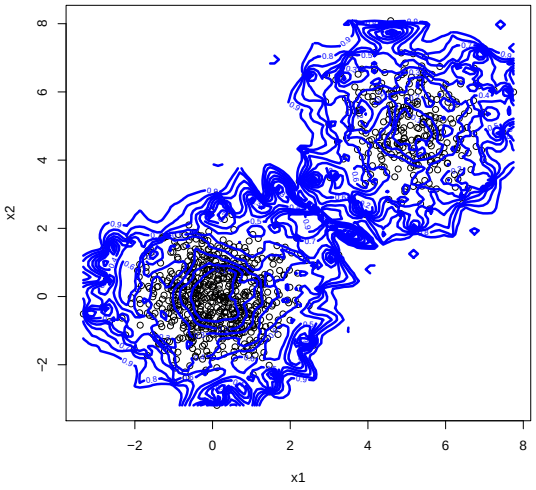


(b) Laplacian

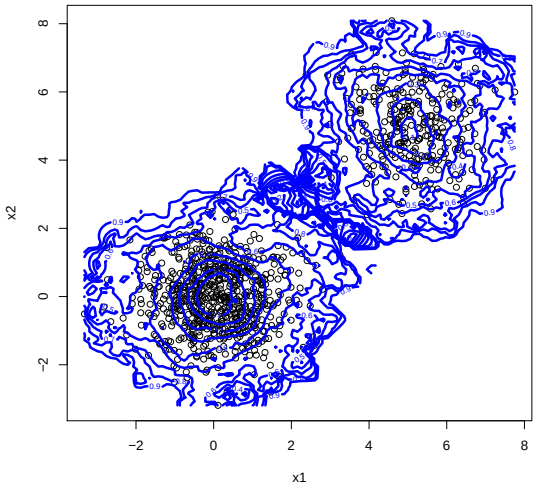


(c) Logistic

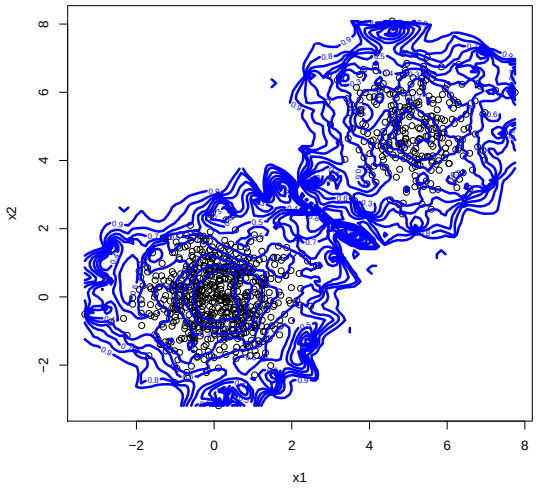
Figure 4.2: Contour plots of the weighted spatial rank function (4.3.1) using: (a) Gaussian, (b) Laplacian, (c) logistic, (d) triangular, (e) uniform, and (f) Epanechnikov kernel weights based on 1000 random observations from bivariate mixture normal distribution with 2 groups.



(d) Triangular



(e) Uniform



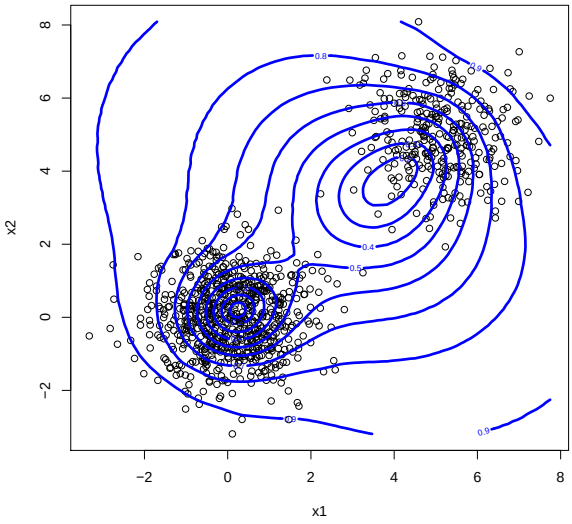
(f) Epanechnikov

Figure 4.2: Continued.

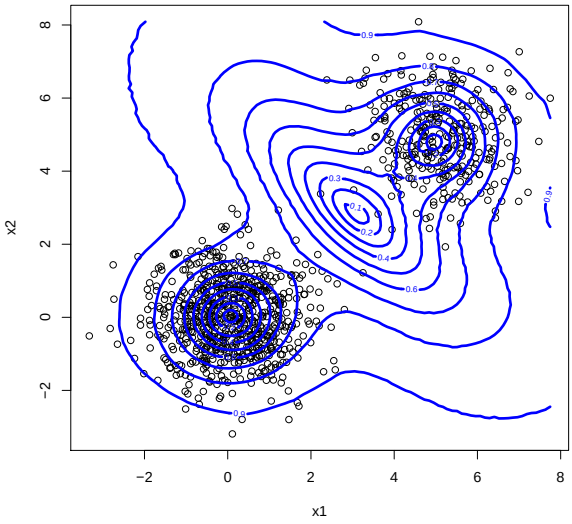
Figure 4.3 gives the contour plots of the weighted spatial ranks based on generalized Mallow weights, at $r = 1, r = 3, r = 5$ and Naranjo and Hettmanspergers' weights. As we can see, the weighted spatial ranks based on generalized Mallow weights do not fit nicely to the shape of the clusters when $r = 1$, and they give similar contour shapes to the weighted ranks contours based on Gaussian kernels when r=3 and 5 with some lines between the two clusters. From the last panel in this figure, it can be clearly noticed that the weighted spatial ranks based on Naranjo and Hettmanspergers' weights failed to fit the shape of clusters and gave similar contour shape to the generalized Mallow weights at $r = 1$. Finally, it should be mentioned that their codes take longer time because of the MVE estimations and the calculations of the denominator of functions (4.3.10) and (4.3.11) for each observations of the data.

In order to choose the weighted rank function that performs well and its contours that completely fit to the shape of the data cloud and accurately map the clusters structure, we use the same simulated data to apply on the second weighted spatial rank function that defined in (4.3.2), and compare its performance with the other weighted ranks function.

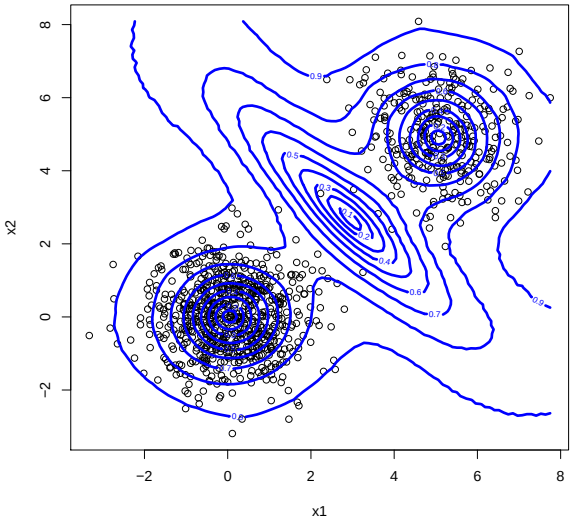
We consider now the weighted ranks contours produced from the weighted spatial rank function that is defined in (4.3.2). Figure 4.4 gives the weighted spatial ranks contours based on the Gaussian, Laplacian, logistic, triangular, uniform, and Epanechnikov kernel weights. Compared to Figure 4.2, we can see that the contour plots that presented in Figure 4.3 are more accurate and can fit better to the shape of the clusters structure. Moreover, the weighted ranks contours based on Gaussian, Laplacian and logistic kernels, did not give many lines between the two clusters, which suggesting the presence of two groups only. Explicitly, the weighted ranks contours based on the Gaussian kernel weights gave the best result, as they captured each observation carefully and assigned it in the true group without any misclassifications. Only one observation that has not been captured due to its location since it is not close to either cluster centers, and can be classified



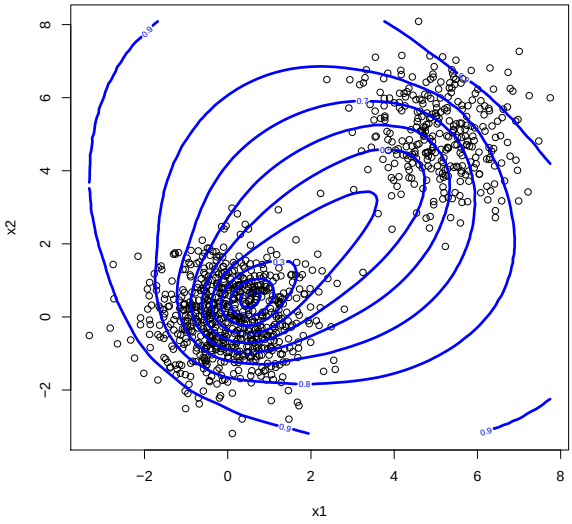
(a) Mallow weights at $r = 1$



(b) Mallow weights at $r = 3$



(c) Mallow weights at $r = 5$



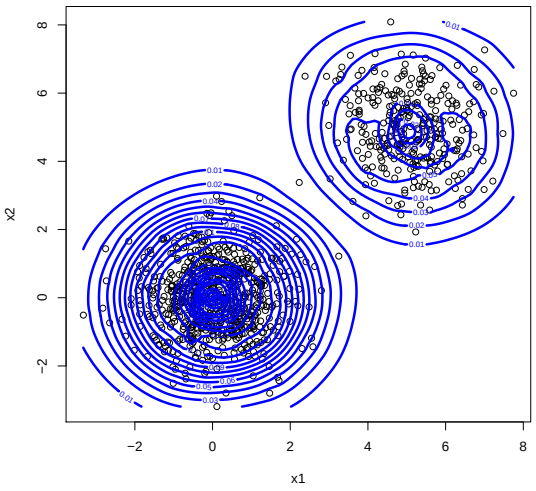
(d) Naranjo and Hettmansperger weights

Figure 4.3: Contour plots of the weighted spatial rank function (4.3.1) using generalized Mallow at (a) $r = 1$, (b) $r = 3$, (c) $r = 5$ and (d) Naranjo weights based on 1000 random observations from bivariate mixture normal distribution with 2 groups.

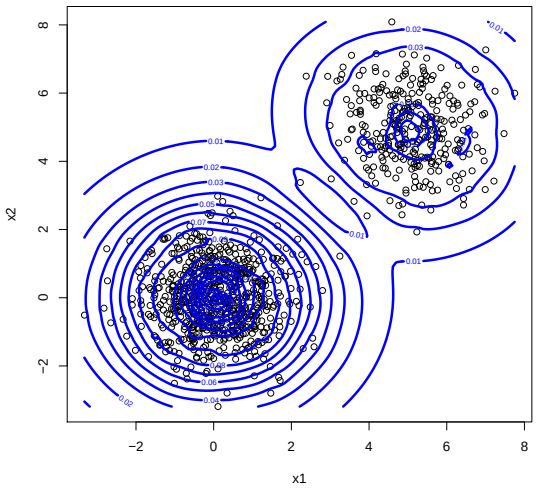
in any of the two groups. Here, the confirmatory weighted spatial rank classifier, which introduced in the next Section of this Chapter, plays its rule and easily determines the suitable cluster for this observation to be assigned. More explanation about this point has been considered in the numerical examples Section, where we provide some similar cases from real dataset.

On the other hand, the weighted ranks contours based on the triangular, uniform and Epanechnikov kernel weights gave clearer visualization than before, where it can be easily seen that the number of clusters depending on the contour lines. As we saw before, the weighted spatial ranks contours using function (4.3.1) and based on the triangular, uniform and Epanechnikov kernel weights produced the groups' contours that are very close to each other, and may it be difficult in other dataset to differentiate the observations in each groups. Conversely, the contours produced by using the weighted ranks in (4.3.2) and the same kernel weights, are very clear and we can easily see the true number of clusters that has been captured. Hence, we strongly recommend the efficiency of using the weighted spatial rank function that defined in (4.3.2), and considering the Gaussian kernel weights as the best weight function for it until now.

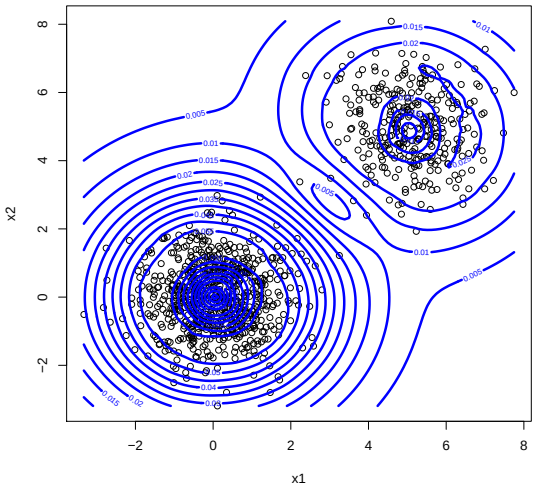
Figure 4.5 shows the weighted spatial ranks contours using function (4.3.2) and based on generalized Mallow weights, at $r = 1, r = 3, r = 5$ and Naranjo and Hettmanspergers' weights. Compared to Figure 4.3, it can be noticed that using the weighted spatial ranks that are defined in (4.3.2) based on generalized Mallow weights gives better result, where their contour fit the shape of the clusters whether $r = 1$, 3 or 5. Clearly, when r=5, we get more accurate contour lines, suggesting that the bigger value of r takes, the better result we get. On the other hand, the weighted spatial ranks based on Naranjo and Hettmanspergers' weights failed to fit the shape of clusters again and did not give any evidence of the existence of two clusters in the simulated data. Finally, it should be noticed from the above contour plots that, the weighted spatial rank function that defined



(a) Gaussian

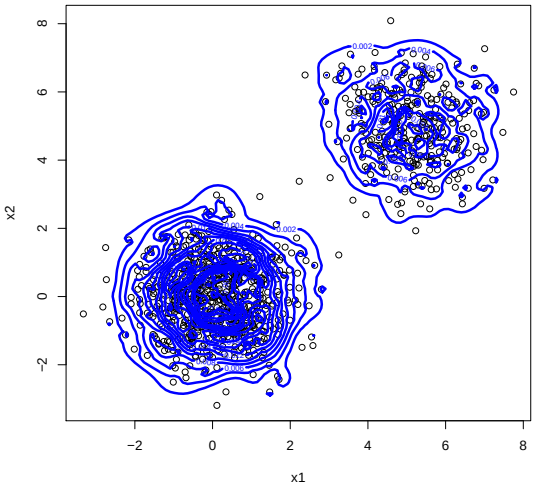


(b) Laplacian

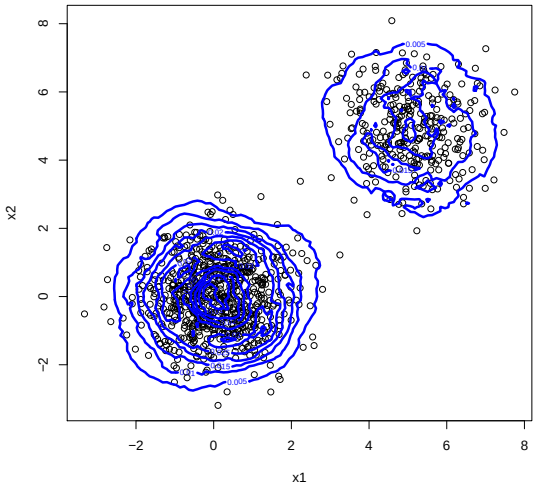


(c) Logistic

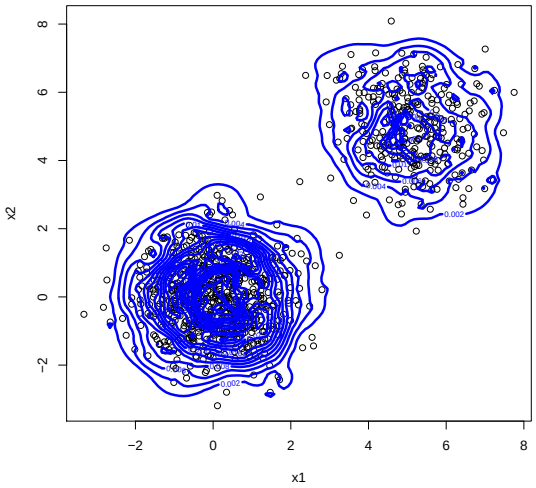
Figure 4.4: Contour plots of the weighted spatial rank function (4.3.2) using: (a) Gaussian, (b) Laplacian, (c) logistic, (d) triangular, (e) uniform, and (f) Epanechnikov kernel weights based on 1000 random observations from bivariate mixture normal distribution with 2 groups.



(d) Triangular



(e) Uniform



(f) Epanechnikov

Figure 4.4: Continued.

in (4.3.2) based on Gaussian kernel weights gives the best result.

4.4 Confirmatory Analysis Based on Weighted Spatial Ranks Classifier

In this section, we propose a confirmatory nonparametric classifier that can be used to check and confirm if the observations' assignment based on the weighted spatial rank contours are right or not. The proposed confirmatory classifier is based on the weighted spatial ranks, and it is simple and easy to compute without any need to parameter estimates of the underlying distributions.

Two clusters case:

Suppose that we have two groups, with distributions F and G respectively, then based on the weighted spatial ranks classifier (WSRC) rule, we can assign the d-dimensional observation vector $\mathbf{x}$ to the first group if the Euclidean norm of the weighted spatial ranks of the observation $\mathbf{x}$ based on F is less than the Euclidean norm of the weighted spatial ranks of the observation $\mathbf{x}$ based on G such that assign $\mathbf{x}$ to the group with distribution F if:

$$WSRN_F(\mathbf{x}) < WSRN_G(\mathbf{x}), \tag{4.4.1}$$

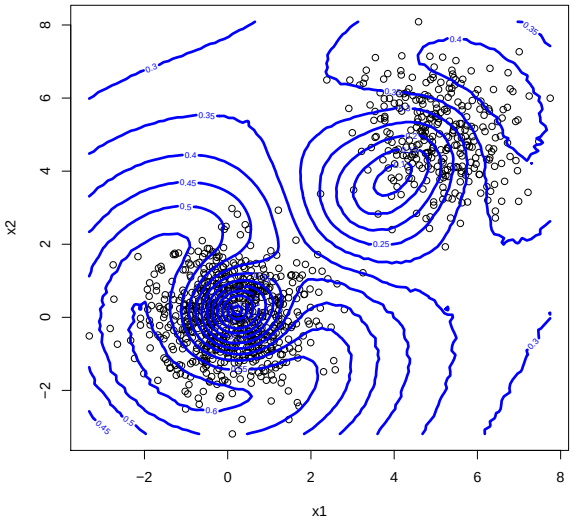
and assign it to the second group with distribution G otherwise, where $WSRN_F(\mathbf{x}) = \|WSR_F(\mathbf{x})\|$ and $WSRN_G(\mathbf{x}) = \|WSR_G(\mathbf{x})\|$ as defined in equation (4.3.3).

More than two clusters:

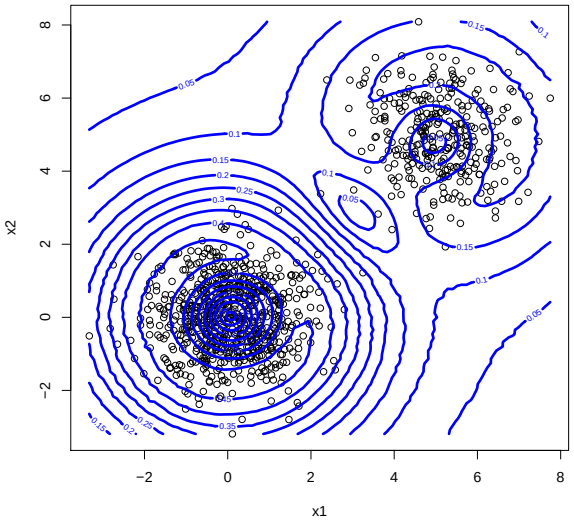
Suppose that we have K groups, with distributions $F_1, F_2, \ldots, F_k$, then we can assign the d-dimensional observation vector $\mathbf{x}$ to the i-th group if:

$$WSRN_{F_i}(\mathbf{x}) = \min_{1 \leqslant j \leqslant k} WSRN_{F_j}(\mathbf{x}), \tag{4.4.2}$$

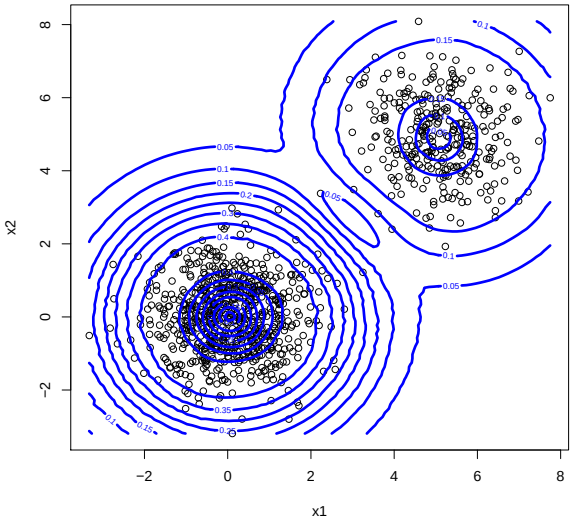
where $i \neq j, 1 \leqslant i \leqslant k$.



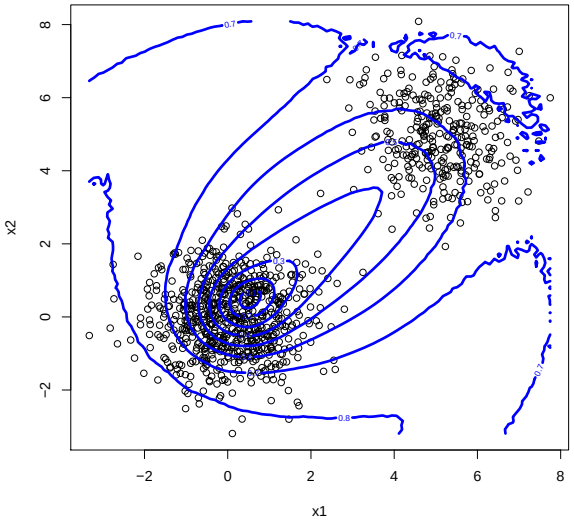
(a) Mallow weights at $r = 1$



(b) Mallow weights at $r = 3$



(c) Mallow weights at $r = 5$



(d) Naranjo and Hettmansperger weights

Figure 4.5: Contour plots of the weighted spatial rank function (4.3.2) using generalized Mallow at (a) $r = 1$, (b) $r = 3$, (c) $r = 5$ and (d) Naranjo weights based on 1000 random observations from bivariate mixture normal distribution with 2 groups.

So, after determining the number of clusters using the weighted spatial ranks contour, we can now use the above weighted spatial ranks classifier as a confirmatory analysis in order to assigning each observation to the most suitable clusters. Moreover, we introduce a confirmatory plot based on the weighted spatial ranks, which can be used easily to now the assignment of each observation as we present in the numerical examples section.

4.5 Weighted Spatial Ranks Clustering Algorithm

Now, we present the weighted spatial ranks clustering algorithm. We start with the bivariate case ($d = 2$), and then we consider the higher dimensional case $d > 2$.

Weighted spatial ranks clustering algorithm for bivariate case ($d = 2$)

1. Let $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ is a 2-dimensional random sample with the two variables x_1 and x_2, and let S_{x_1} and S_{x_2} are two vectors of the generated regular sequences from the minimum to maximum value of x_1 and x_2 respectively (grid points) with length l, then construct the matrix $\mathbf{S}_x$ which is a grid matrix consists of the two vectors S_{x_1} and S_{x_2}.

2. For each $\mathbf{s} \in \mathbf{S}_x$, calculate $WSRN(\mathbf{s})$ with respect to $\mathbf{X}_i$:

$$WSRN(\mathbf{s}) = \|WSR(\mathbf{s})\| = \|\frac{1}{n}\sum_{i=1}^{n} w_i \frac{\mathbf{s} - \mathbf{X}_i}{\|\mathbf{s} - \mathbf{X}_i\|}\|, \tag{4.5.1}$$

   where $i = 1, \ldots, n$ and $w_i = exp(-\|\mathbf{s} - \mathbf{X}_i\|^2/2)$.

3. Get the outer product Z of the arrays S_{x_1} and S_{x_2} based on the $WSRN(\mathbf{s})$ with respect to $\mathbf{X}_i$ from step 2.

4. Plot the weighted functional spatial ranks contour based on S_{x_1}, S_{x_2} and Z, and determine the number of clusters K from the contour lines.

5. Based on the contour lines, specify the observations that are allocated in each cluster. We can use a lower contour level for better visualization.

6. Use the confirmatory weighted spatial rank classifier's rule that is shown in (4.4.1) and (4.4.2) to confirm the assignment of each observation, assign the unassigned observations to the proper cluster, and get the confirmatory plot.

Weighted spatial ranks based clustering algorithm for higher dimensions ($d > 2$)

If the dimension of the data is higher than 2, then we need to use the principle component analysis PCA in order to reduce the dimension and get the contour plot. In this case, the proposed method can be classified as a filtering method based on PCA. The high dimensional data are often transformed into lower dimensional data via the PCA or singular value decomposition where coherent patterns can be detected more clearly (Jolliffe, 2002). The main idea of using PCA as a dimension reduction tool is that PCA tries to choose the dimensions with the largest explained variances, or in mathematical words it is equivalent to find the best low rank approximation in L_2 norm of the data via the singular value decomposition SVD (Eckart and Young, 1936). So, PCA constructs a set of uncorrelated directions that are ordered by their variance. The unsupervised dimension reduction is used in very diversified areas of scientific disciplines such as image processing, meteorology and information retrieval.

In clustering analysis, different filtering methods have been proposed using the PCA and kernel principal component analysis (KPCA). For example, Ng et al. (2001) proposed a spectral clustering algorithm based on eigenvectors and KPCA with a Gaussian kernel, and they pointed out that K-means can be applied after the data embedded in a low-dimensional space. Moreover, in Zha et al. (2002) the PCA has been used to project data to a lower dimensional subspace and then K-means has been applied in the subspace. On the other hand, Ding and He (2004) proved that principal components are the continuous

solutions to the discrete cluster membership indicators for K-means clustering, as the subspace spanned by the cluster centroids are given by spectral expansion of the data covariance matrix truncated at K-1 terms. They also applied their technique on DNA gene expression and Internet newsgroups data. Ben-Hur and Guyon (2003) discussed a method based on PCA, which can be used in detecting the stable clusters.

In order to get the k-th component based on PCA, we have first to remove the mean effect by substracting the mean vector from each d-dimensional observations vector. After getting the mean-corrected version, we can define the d-dimensional vectors of weights or loadings $\mathbf{w}_{(k)} = (w_1, \ldots, w_d)_{(k)}$ that constrained to be a unit vector, and they may be chosen to be the eigenvectors of $\mathbf{X}^\top\mathbf{X}$, where $\mathbf{X} \in \mathbb{R}^d$ has a d-dimensional, in order to satisfy the condition below. These loadings map each row vector $\mathbf{x}_{(i)}$ of $\mathbf{X}$ to a new vector of principal component scores $\mathbf{C}_{(i)} = (C_1, \ldots, C_k)_{(i)}$. Then we can get a vector of principal component scores:

$$C_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(k)}, \tag{4.5.2}$$

The previous set-up should satisfy that each individual variables of C considered over the data set successively inherit the maximum possible variance from $\mathbf{x}$. The first loading vector $w_{(1)}$ should satisfy:

$$\mathbf{w}_{(1)} = \arg\max \left\{ \frac{\mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w}}{\mathbf{w}^\top\mathbf{w}} \right\}, \tag{4.5.3}$$

where $\mathbf{w}$ is a matrix of the d-dimensional vectors of the loadings $\mathbf{w}_{(k)}$, and for the symmetric matrix $\mathbf{X}^\top\mathbf{X}$ the maximum possible value is the largest eigenvalue of the matrix, which occurs when $\mathbf{w}$ is the corresponding eigenvector. Now, the k-th component can be found by subtracting the first $k-1$ principal components from $\mathbf{X}$, such that:

$$\widehat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} \mathbf{X}\mathbf{w}_{(s)}\mathbf{w}_{(s)}^\top, \tag{4.5.4}$$

so the loading vector which extracts the maximum variance from this new data matrix satisfy:

$$\mathbf{w}_{(k)} = \arg\max \left\{ \frac{\mathbf{w}^\top \widehat{\mathbf{X}}_k^\top \widehat{\mathbf{X}}_k \mathbf{w}}{\mathbf{w}^\top \mathbf{w}} \right\}. \tag{4.5.5}$$

Finally, we can get the full principal components decomposition of $\mathbf{X}$:

$$\mathbf{C} = \mathbf{X}\mathbf{W}, \tag{4.5.6}$$

where $\mathbf{W}$ is a d-by-d matrix whose columns are the eigenvectors of $\mathbf{X}^\top\mathbf{X}$.

The steps of the weighted spatial ranks clustering algorithm when $d > 2$, are as the following:

1. Let $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ is a d-dimensional random sample, then use the PCA to get the two components C_1 and C_2 of that sample, and construct the matrix $\mathbf{C}$, which is a matrix consists of the two components C_1 and C_2.

2. Now let S_{C_1} and S_{C_2} are two vectors of the generated regular sequences from the minimum to maximum value of C_1 and C_2 respectively with length l, then construct the matrix $\mathbf{S}_c$, which is a matrix consists of the two vectors S_{C_1} and S_{C_2}.

3. For each $\mathbf{s} \in \mathbf{S}_c$, calculate $WSRN(\mathbf{s})$ with respect to $\mathbf{C}_i$:

$$WSRN(\mathbf{s}) = \|WSR(\mathbf{s})\| = \|\frac{1}{n}\sum_{i=1}^{n} w_i \frac{\mathbf{s} - \mathbf{C}_i}{\|\mathbf{s} - \mathbf{C}_i\|}\|, \tag{4.5.7}$$

   where $i = 1, \ldots, n$ and $w_i = exp(-\|\mathbf{s} - \mathbf{C}_i\|^2/2)$.

4. Get the outer product Z of the arrays S_{C_1} and S_{C_2} based on the $WSRN(\mathbf{s})$ with respect to $\mathbf{C}_i$ from step 3.

5. Plot the weighted functional spatial ranks contour based on S_{C_1}, S_{C_2} and Z, and determine the number of clusters K from the contour lines.

6. Based on the contour lines, specify the observations that are allocated in each cluster. We can use a lower contour level for better visualization.

7. Use the confirmatory weighted spatial rank classifier's rule that is shown in (4.4.1) and (4.4.2) to confirm the assignment of each observation, assign the unassigned observations to the proper cluster, and get the confirmatory plot.

Figure 4.6 shows the flowchart of the weighted spatial ranks based clustering algorithm.

4.6 Numerical Examples

We present some systematic evaluations of the proposed weighted spatial ranks based clustering algorithm. In the first example, we test the performance of the weighted spatial ranks on simulated data includes four variables and three clusters from mixture normal distribution. Next, we consider another simulated data includes six variables and four groups. Finally, on three different real datasets, we check the performance of the weighted spatial ranks based clustering algorithm.

4.6.1 Simulated Data

In the first simulated data example, we consider a mixture of three 4-dimensional Gaussian distributions. The mixing proportions p are taken to be $p_1 = 0.3$, $p_2 = 0.4$, and the sample size ($n = 100$), such that $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ is a random sample from 4-dimensional mixture normal distribution:

$$p_1 N_4(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + p_2 N_4(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}) + (1 - p_1 - p_2) N_4(\boldsymbol{\mu}_3, \boldsymbol{\Sigma}), \tag{4.6.1}$$

where $\boldsymbol{\mu}_1 = (4, 4, 4, 4)^\top$, $\boldsymbol{\mu}_2 = (-4, 4, -4, 4)^\top$, $\boldsymbol{\mu}_3 = (-4, -4, -4, -4)^\top$ and $\boldsymbol{\Sigma} = \boldsymbol{I}$. The problem of this data is that, it is a misleading data, where it can be possible to view it as a data includes 2 clusters; however we simulated it to include 3 groups. For example,

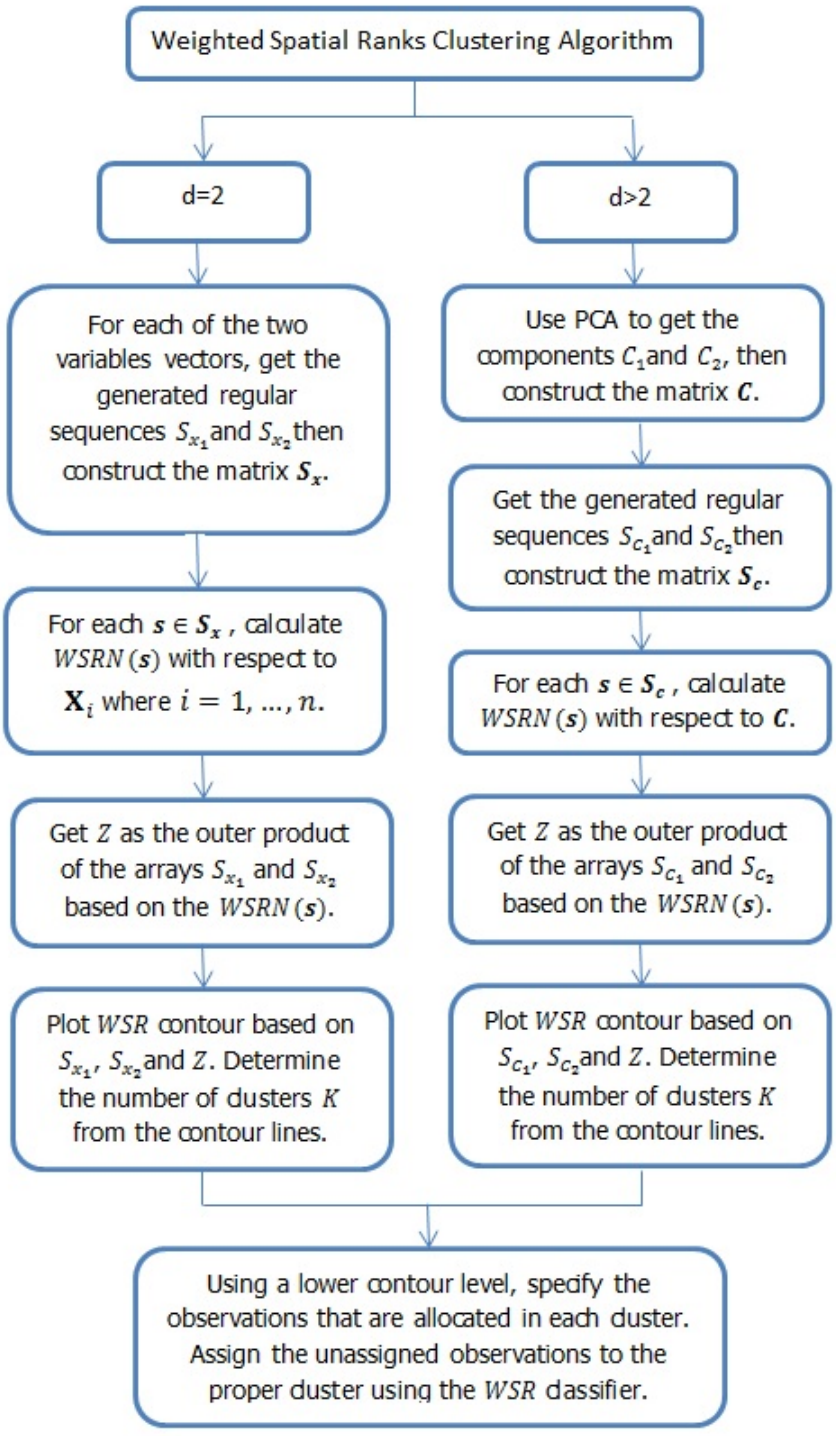


Figure 4.6: The steps of the weighted spatial ranks based clustering algorithm

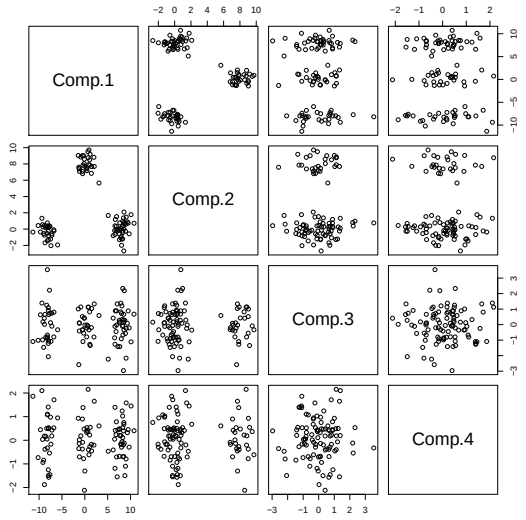
it is clear from the var1 vs var2 panels, in the scatterplot matrix of the data in Figure 4.7, that we have 3 clusters, but if we look to the var1 vs var3 or var2 vs var4 panels, we definitely decide that we have only 2 clusters. The upper left and right panels of Figure 4.7 are the scatterplot matrix of the PCA components, and the total variance explained by each component respectively. Clearly, we can see that the first two components give very strong evidence about the number of clusters which suggests 3 clusters, and they explain 97% (61.75) of the total variances explained by the four components (63.66). Panels (c) and (d) are the weighted spatial ranks contour of the first two components and the contour at level 0.005 respectively. It can be clearly seen that the weighted spatial ranks contour accurately fit to the shape of the three clusters, without any misclassification. Panels (e) and (f) show the confirmatory plots based on weighted ranks classifier for the first 2 components and the original data respectively. Successfully, we got the right observations' assignment compared to the simulated three clusters data. Based on this result, we can say that we have a full clustering methodology that gives us the right number of clusters and accurate assignment.

To check the stability of our proposed method against the dimensions and the number of clusters, in this example, we consider a higher dimensional data with four clusters. In this example, we consider a mixture of four 6-dimensional Gaussian distributions, with equal proportions weights ($p = 0.25$), and the sample size is ($n = 100$) again. So, $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^d$ is a random sample from 6-dimensional mixture normal distribution:

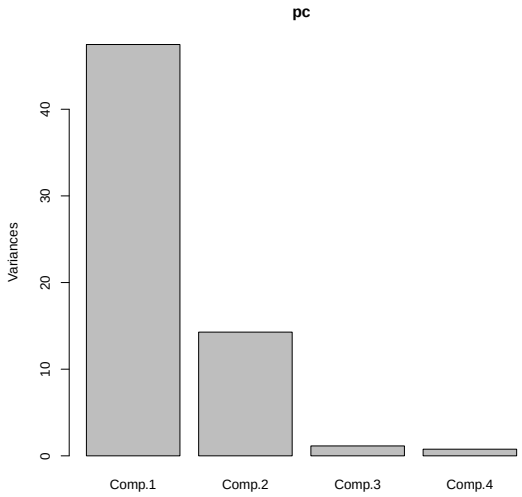
$$pN_6(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + pN_6(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}) + pN_6(\boldsymbol{\mu}_3, \boldsymbol{\Sigma}) + pN_6(\boldsymbol{\mu}_4, \boldsymbol{\Sigma}), \tag{4.6.2}$$

where $\boldsymbol{\mu}_1 = (4, 4, 4, 4, 4, 4)^\top$, $\boldsymbol{\mu}_2 = (-4, 4, -4, 4, -4, 4)^\top$, $\boldsymbol{\mu}_3 = (-4, -4, -4, -4, -4, -4)^\top$, $\boldsymbol{\mu}_4 = (4, -4, 4, -4, 4, -4)^\top$ and $\boldsymbol{\Sigma} = \boldsymbol{I}$.

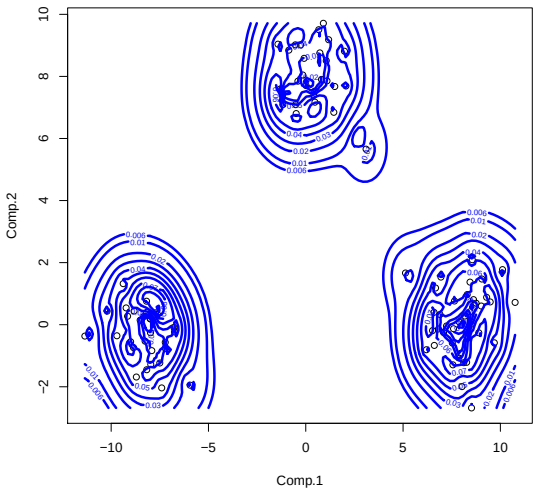
We simulated this data in the same way of the previous one as well, where it can be possible to view it as a data includes either 2 or 4 clusters; however we simulated it to



(a) Scatterplot matrix

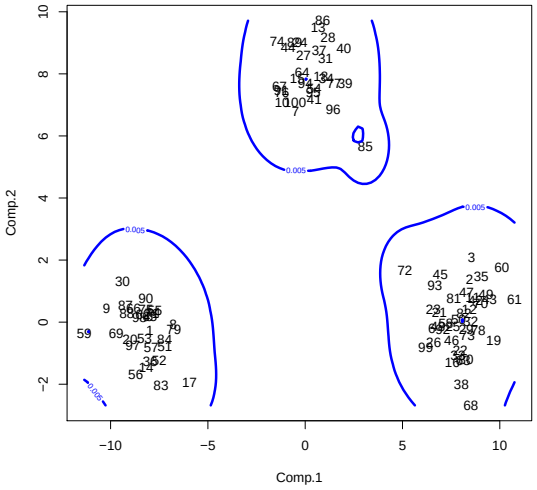


(b) Scree plot

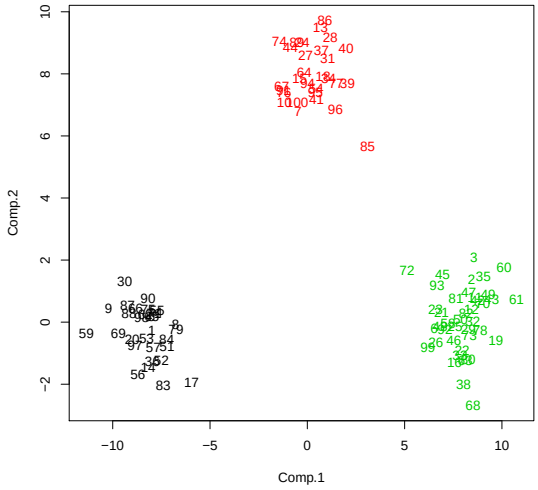


(c) WSR contour

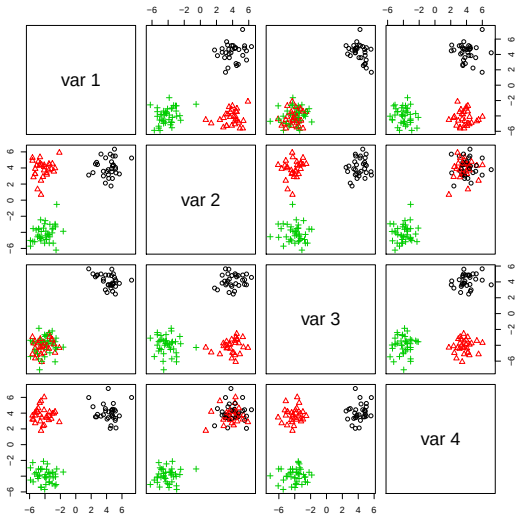
Figure 4.7: Simulated data 1: (a) scatterplot matrix of the PCA components, (b) the total variance explained by each component, (c) the weighted spatial ranks contour, (d) the contour at level 0.005 and confirmatory plots based on weighted ranks classifier for (e) the first 2 components and (f) the original data.



(d) Contour at level 0.005



(e) Confirmatory plot: first 2 components



(f) Confirmatory plot: original data

Figure 4.7: Continued.

include 4 groups. From the scatterplot matrix of the data in Figure 4.8, we can see that we have 4 clusters based on the var1 vs var2 panels. Conversely, it seems that we have only 2 clusters if we look to the var1 vs var3 or var2 vs var4 panels.

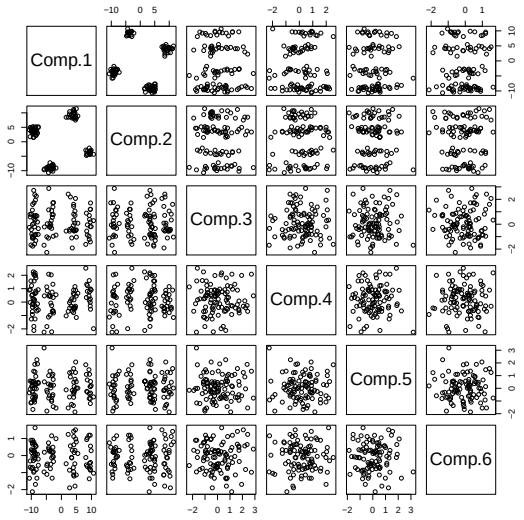
According to the scatterplot matrix of the PCA components that shown in the upper left panel of Figure 4.8, we can see that the first two components strongly suggest the right number of clusters $K = 4$. Moreover, these two components explain 98% (96.22) of the total variances explained by the six components (98.16), as shown in the scree plot for each component in panel (b) of the same figure.

From the weighted spatial ranks contour of the first two components, this is given in panel (c) of the figure, it can be clearly noticed that the weighted spatial ranks contour completely fit to the shape of the four clusters. For better visualization, we used a lower contour level at level 0.001, as shown in panel (d), in terms of viewing the observations that belong to each cluster. Finally, we give the confirmatory plots based on the weighted spatial ranks classifier for the first 2 components and the original data. From panels (e) and (f), we can get information about which observation belongs to which cluster. Again, we got the true assignment which is consistent with the simulated observations.

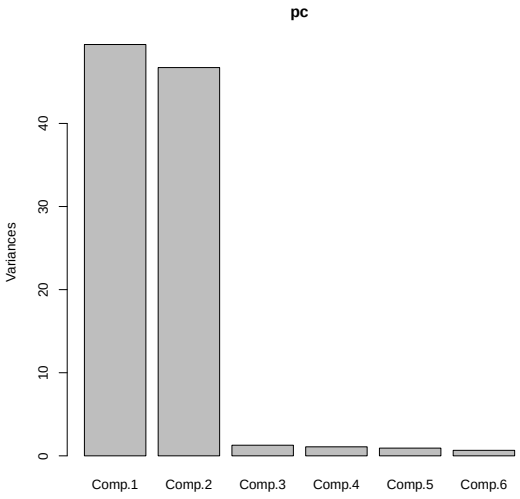
4.6.2 Real Data

In this section, we check the performance of the weighted spatial ranks based clustering algorithm when we consider some real data. The three datasets that we consider here are the iris data (Fisher, 1936), financial data (Atkinson et al., 2004), and old faithful geyser data (Azzalini and Bowman, 1990; Venables and Ripley, 2002).

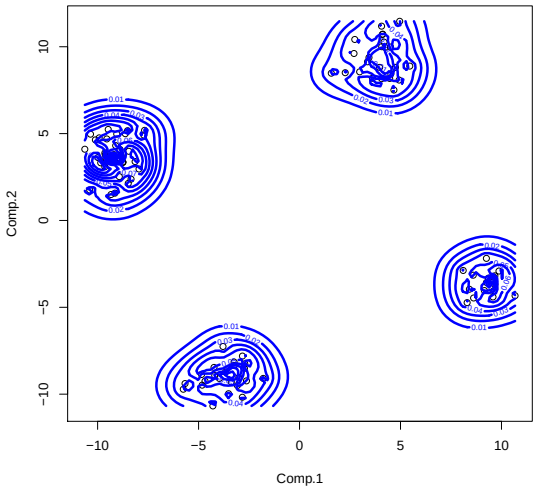
In iris data, as we mentioned before, one of the clusters contains iris setosa, while the other cluster contains both iris virginica and iris versicolor. In clustering analysis, most of the clustering techniques consider this data includes 2 groups, this is due to both iris virginica and iris versicolor is not separable without the species information that



(a) Scatterplot matrix

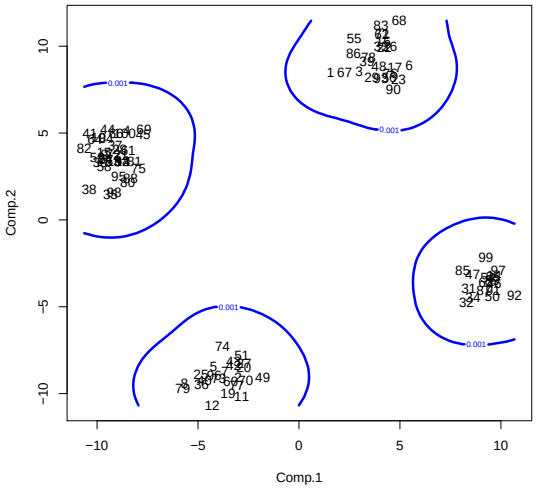


(b) Scree plot

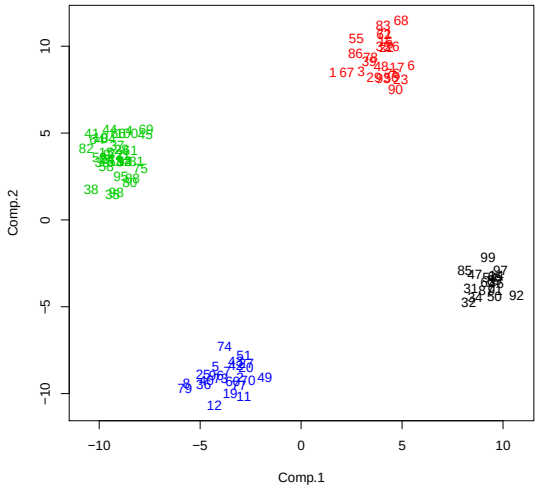


(c) WSR contour

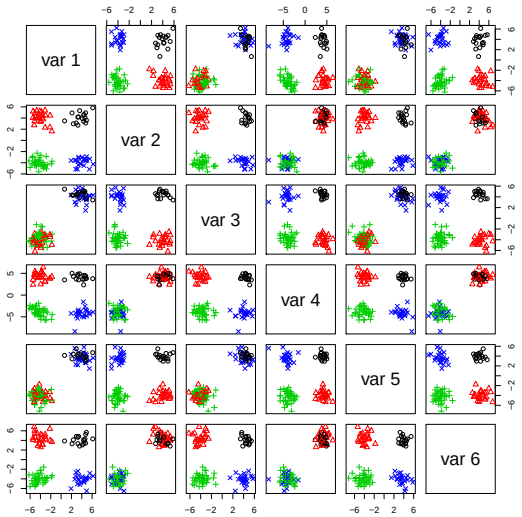
Figure 4.8: Simulated data 2: (a) scatterplot matrix of the PCA components, (b) the total variance explained by each component, (c) the weighted spatial ranks contour, (d) the contour at level 0.001 and confirmatory plots based on weighted ranks classifier for (e) the first 2 components and (f) the original data.



(d) Contour at level 0.001



(e) Confirmatory plot: first 2 components



(f) Confirmatory plot: original data

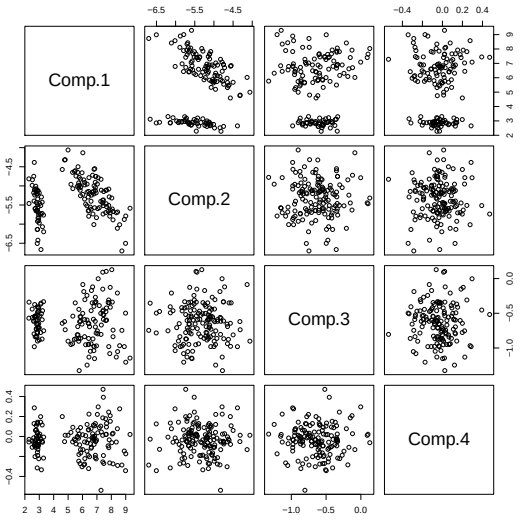
Figure 4.8: Continued.

Fisher used, which consider a good example to explain the difference between supervised and unsupervised methods. As we can see from Figure 4.9, the weighted ranks contours based on the first two components, which explain 97.8% of the total variances, indicate two clusters. Successfully, the confirmatory plots based on the weighted spatial ranks classifier gave the exact assignment in Iris setosa group, and both iris virginica and iris versicolor group.

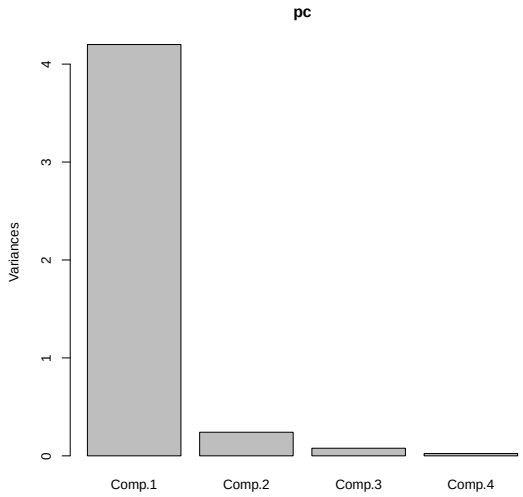
The second real dataset is the financial data (Atkinson et al., 2004) that has been analyzed in Chapter 3 and as we mentioned before it includes 103 observations, 3 variables and 2 clusters. From Figure 4.10, the weighted ranks contours of components 1 and 3, which explain 96.4% of the total variances and give initial idea about the number of clusters, indicate two clusters. Moreover, the confirmatory plots based on the weighted spatial ranks classifier gave the true observations' assignment which is consistent with the two types of fund.

For the Old Faithful data, as we mentioned before it includes 272 observations with 2 variables and 2 clusters. From Figure 4.11, the weighted ranks contours of the data indicate 2 clusters with one observation (number 174) located between them. Using the confirmatory weighted spatial ranks classifier is helpful in such a case, as it assigned this observation in the second cluster, as shown in the confirmatory plots that gave the true observations' assignment which is consistent with the two types of eruptions (the short and long eruptions).

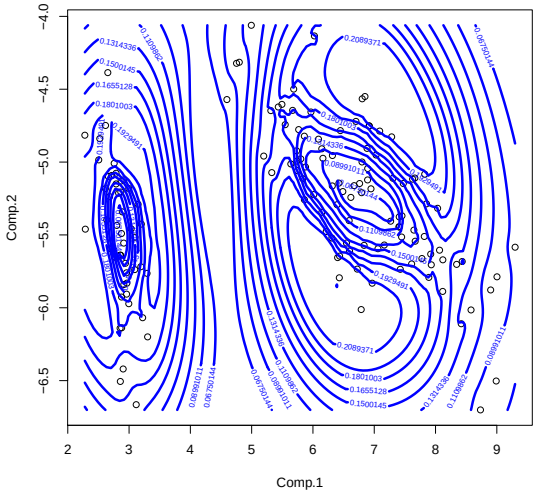
We implement ten of the clustering methods on the three real datasets that we discussed above, in order to compare them with the proposed WSR method. These methods are: K-means (Lloyd, 1957), K-Centroids (Leisch, 2006), linkage based methods (hclust), model-based clustering "mclust" (Fraley and Raftery, 2002), high dimensional data clustering "HDDC" (Bouveyron et al., 2007), mixtures of probabilistic principle component analysers "MixtPPCA" (Tipping and Bishop, 1999), partitioning around medoids cluster-



(a) Scatterplot matrix

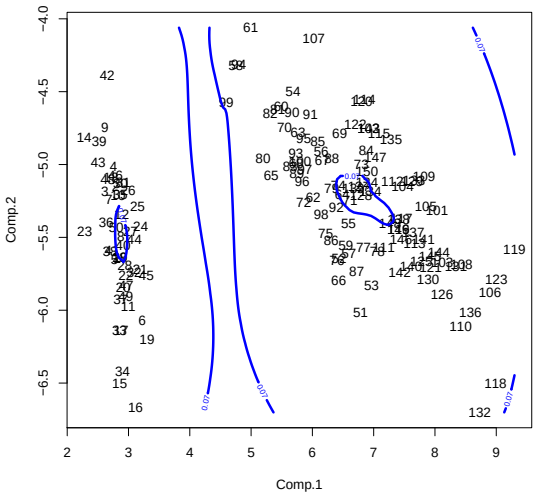


(b) Scree plot

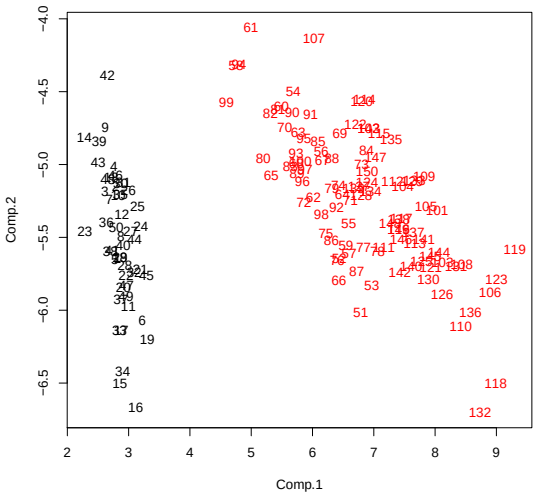


(c) WSR contour

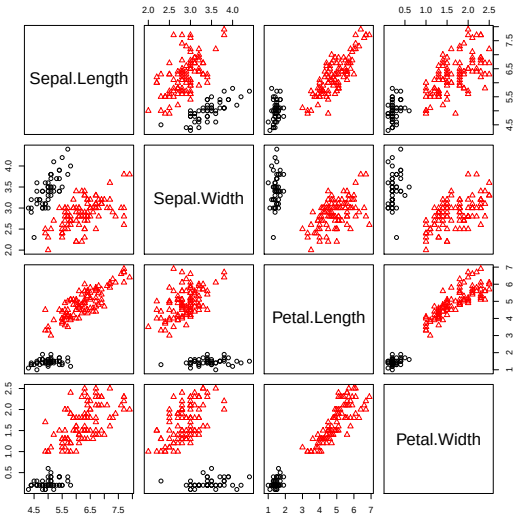
Figure 4.9: Iris data: (a) scatterplot matrix of the PCA components, (b) the total variance explained by each component, (c) the weighted spatial ranks contour, (d) the contour at level 0.07 and confirmatory plots based on weighted ranks classifier for (e) the first 2 components and (f) the original data.



(d) Contour at level 0.07

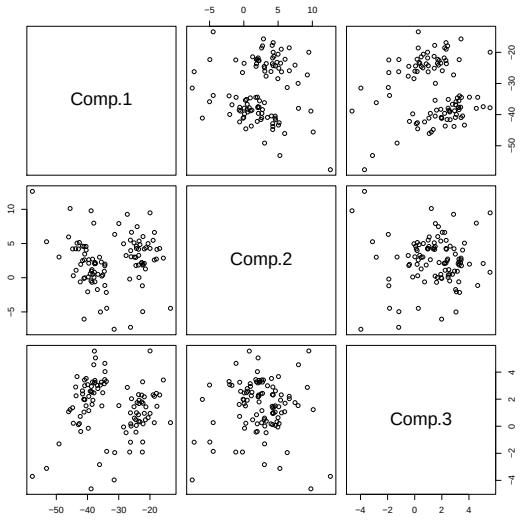


(e) Confirmatory plot: first 2 components

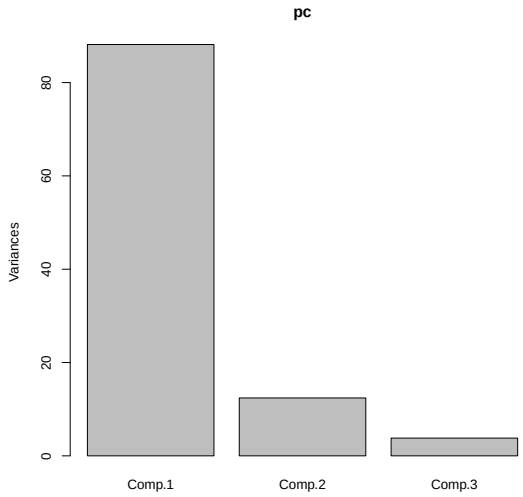


(f) Confirmatory plot: original data

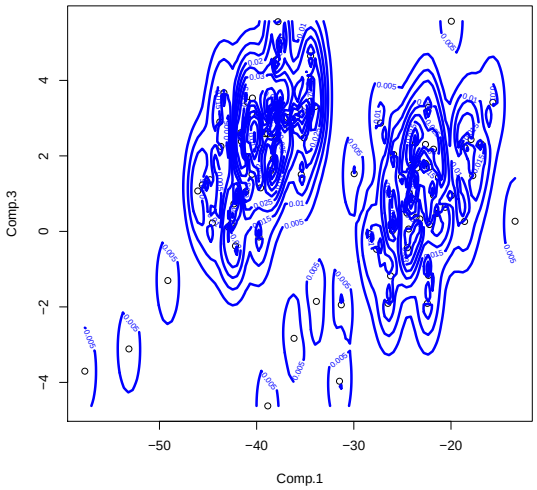
Figure 4.9: Continued.



(a) Scatterplot matrix

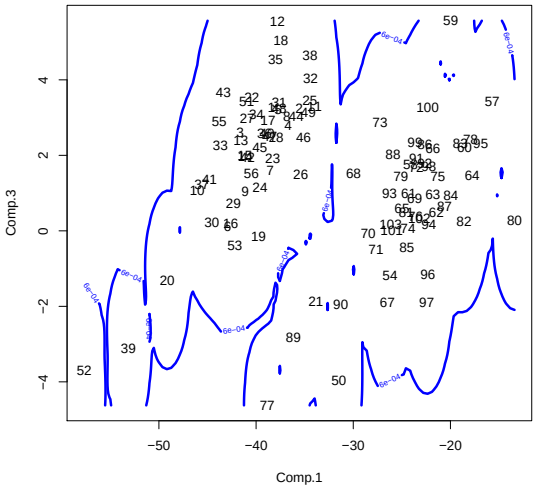


(b) Scree plot

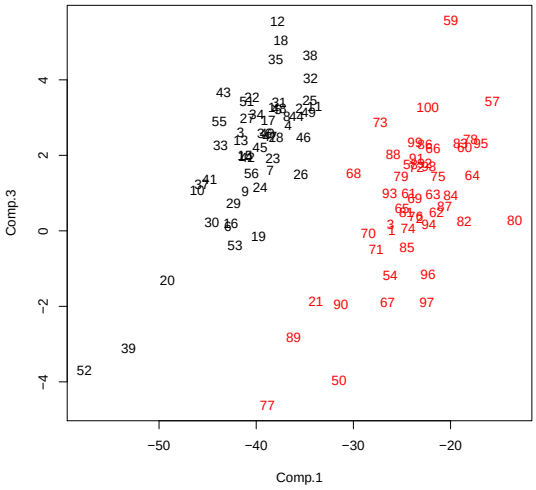


(c) WSR contour

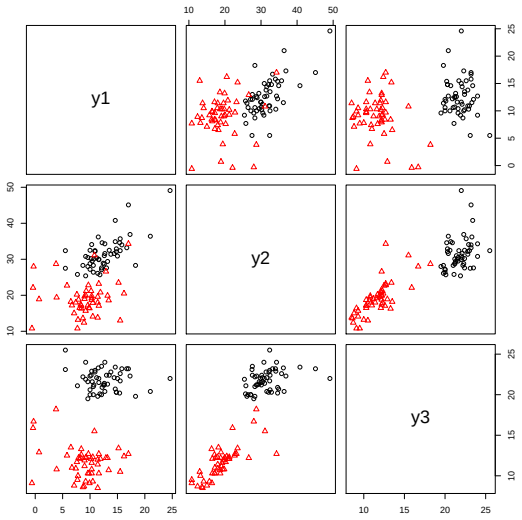
Figure 4.10: Financial data: (a) scatterplot matrix of the PCA components, (b) the total variance explained by each component, (c) the weighted spatial ranks contour, (d) the contour at level 0.0006 and confirmatory plots based on weighted ranks classifier for (e) the first 2 components and (f) the original data.



(d) Contour at level 0.0006

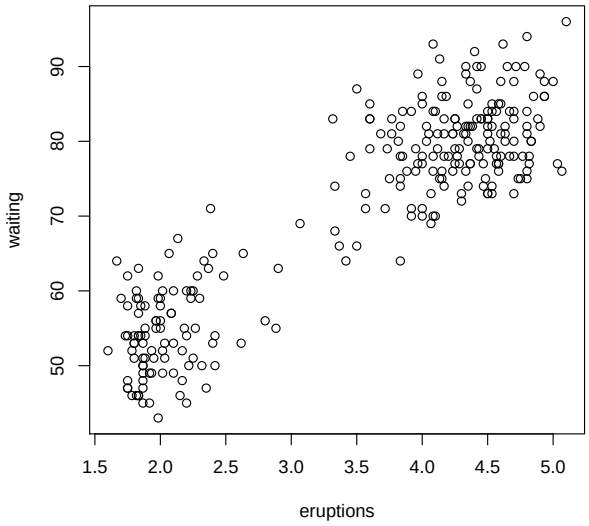


(e) Confirmatory plot: first 2 components

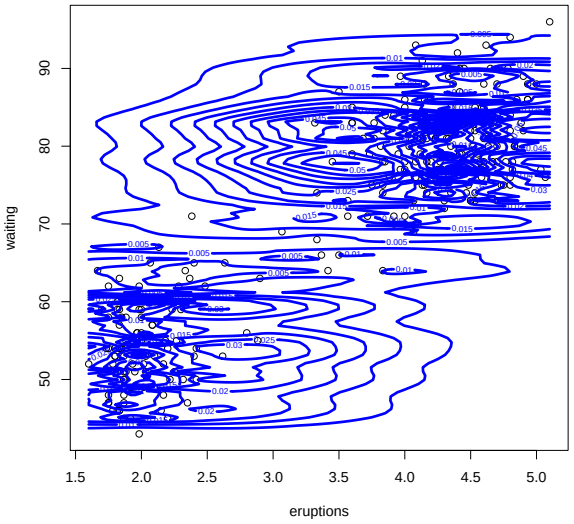


(f) Confirmatory plot: original data

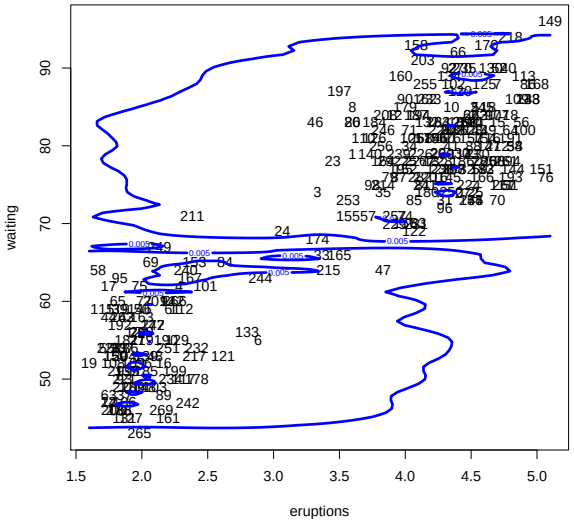
Figure 4.10: Continued.



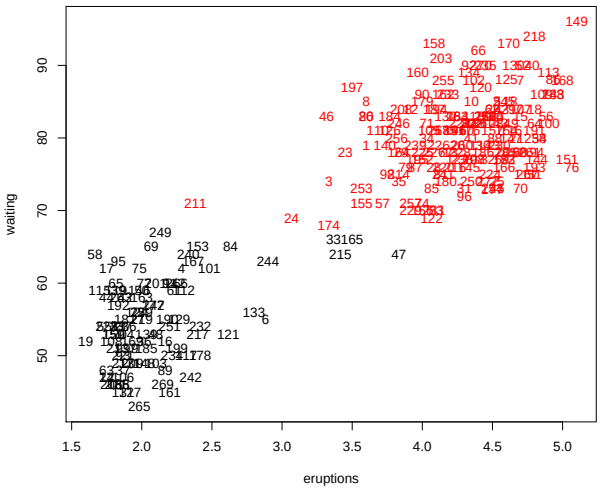
(a) Old faithful data



(b) WSR contour



(c) Contour at level 0.005



(d) Confirmatory plot

Figure 4.11: Old faithful data: (a) scatterplot of the faithful data, (b) the weighted spatial ranks contour, (c) the contour at level 0.005 and (d) confirmatory plots based on weighted ranks classifier for original data.

ing "PAM" (Reynolds et al., 1992), convex clustering "cclust" based on the hard competitive learning method (Ripley, 1996), clustering with divisive analysis "DIANA" (Kaufman and Rousseeuw, 1990), and clustering large applications "CLARA" (Kaufman and Rousseeuw, 1990).

In order to perform this comparison, we need to use some cluster validity index. The cluster validity index measures how accurate the clustering results are, after a clustering algorithm produces its result. This is done by determining how many points have been correctly associated with their class labels accurately and how many belong to a class label with which they should not be associated. In other words, the cluster validity measure can be used to test the performance and accuracy of various algorithms or accuracy of one algorithm for various parameter values or number of clusters. Different validity measures have been proposed in the literature, and they are classified into three types. The first type is the external indexes that can be used to measure the extent to which cluster labels match externally supplied class labels. The second one is the internal indexes that measure the goodness of a clustering structure without respect to external information. The third type is the sum of squared error (SSE) relative indexes that compare two different clusterings or clusters.

The cluster validation has been considered one of the questionable topics in the clustering analysis. This is due to the fact that there is no completely acceptable solution for the different measures of the cluster validity. For instance, we may get different validity results using the same measure if we get different clustering from the same clustering method. A good example for this situation is the kmeans algorithm, where we may get different clustering every time we run the algorithm for the same data. This is because the algorithm starts with different initial points (partitioning) in every run, and this will definitely cause getting different validity result. Certainly, the problem will be getting worse and worse when we get different number of clusters every run. For this reason,

there is no preferable cluster validity measure to tackle this problem. In the next part, we use the probability of misclassification error as a cluster validity tool, however it is still suffering from the previously mentioned problem.

Table 4.1 gives the probabilities of misclassification error based on the different clustering approaches for the three real datasets. Some of these methods give different rate of misclassification error in every time we run the algorithm due to the difference in the initial partitioning of the data points in every time. To address this problem and to calculate a fair version of the probability of misclassification error, we have repeated the algorithms 1000 times, and then we took the average of their correspondence probabilities. From Table 4.1, we can see that the proposed WSR method gives optimal and competitive results for the three real datasets, where it gives the smallest probability of misclassification error amongst all the other methods. As a conclusion, we recap that:

1. The main idea behind WSR is to define a dissimilarity measure locally based on a localized version of multivariate ranks.

2. The WSR is completely data-driven.

3. The WSR is Easy to compute without any need to parameter estimates of the underlying distributions.

4. The WSR is Robust against distributional assumptions.

5. The WSR is more accurate in the purpose of intuitive visualization since we can easily determine the number of clusters from the weighted ranks contours for a low-dimensional input space, using dimension reduction.

6. The WSR can be used to determine the assumed number of clusters, and to assign each observation to its cluster.

Table 4.1: The probabilities of misclassification error based on the different clustering methods for faithful, financial and iris datasets.

Method	Old Faithful	Financial	Iris
WSR	**0.0184**	**0.0291**	**0.0000**
K-means	0.5116	0.5000	0.4856
K-Centroids	0.4701	0.5217	0.4943
Single linkage	0.3603	0.4660	**0.0000**
Complete linkage	0.0515	0.0583	0.1867
Average linkage	**0.0184**	**0.0291**	**0.0000**
Mclust	0.1654	0.6214	**0.0000**
HDDC	-	0.5083	0.4680
MixtPPCA	-	**0.0291**	**0.0000**
PAM	**0.0184**	**0.0291**	0.0067
Cclust	0.4884	0.4755	0.5038
DIANA	0.0037	0.0388	**0.0000**
CLARA	**0.0184**	**0.0291**	0.0133

Chapter 5

Clustering of Functional Data

5.1 Introduction

In the age of technology, challenges of analysis, storage, and visualization of big-data have been become a very active topic in statistics, especially when the dimension d is large compared to the number of observations. A particular attention is paid, in the literature, to the case that the random variables taking values into an infinite dimensional space such as a space of functions defined on some set $\mathcal{T}$, where $\mathcal{T} \subset \mathbb{R}$ could be time interval. In these circumstances, this type of data can be explained well by curves, and then it is called functional data.

Recently, functional data analysis received an attention in diversified areas such as medical studies, weather researches, phonetics, economics, social science and stock market. Many definitions for functional data have been used in the literature. A simple one that was considered by Ferraty and Vieu (2006) is: "functional data is the data that is represented by a set of curves belonging to an infinite dimensional space". Ramsay and Silverman (2005) pointed out that functional data can be a function of time and it consist of observed functions or curves.

Ferraty and Vieu (2006) defined the functional random variable $\mathcal{X}$ as a random variable

with values in an infinite dimensional space or functional space, such that the functional data represents a set of observations $\mathcal{X}_1, \ldots, \mathcal{X}_n$ of $\mathcal{X}$. For instance, the stochastic process $\mathcal{X} = \{\mathcal{X}(t); t \in \mathcal{T}\}$; where $\mathcal{T} \subset \mathbb{R}$ is a good example for the functional data which taking values in some Hilbert space $\mathbb{H}$ of functions defined on some set $\mathcal{T}$ which represents an interval of time, of wavelengths or any other subset of $\mathbb{R}$ (Jacques and Preda, 2014). So, one can say that:

$X_i(t_ij) = \{\{X_1(t_{11}), X_1(t_{12}), \ldots, X_1(t_{1d_1})\}, \ldots, \{X_n(t_{n1}), X_n(t_{n2}), \ldots, X_n(t_{nd_n})\}\}$ is a finitely observed functional dataset, where $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, d_i\}$, if $X_1(t), X_2(t), \ldots, X_n(t)$ is a functional dataset generated by the functional random variables $\mathcal{X}_1(t), \mathcal{X}_2(t), \ldots, \mathcal{X}_n(t)$. Functional data may take univariate or multivariate version. For the multivariate functional data, the elements of $\mathbb{H}$ are $\mathbb{R}^d$-valued functions ($d \geqslant 2$). For more comprehensive details of this subject, the reader is referred to Ramsay and Silverman (2005), Jacques and Preda (2013b), Claeskens et al. (2014) and Ferraty and Vieu (2006).

Many publications have been introduced and discussed different statistical topics in order to analyze the functional data. One of the most important topics that are usually used to explain the functional data is the clustering analysis. Over the last decade, a wealth of publications has been developed for the clustering of functional data. We review many of these publications in the next section. In this chapter, there is a large body of work on using the ordinary and weighted spatial ranks as functional data clustering approaches. Here, we propose two different clustering methods for functional data. The first method is an extension to the forward search based on spatial ranks that has been introduced in Chapter 3, where we extend it to the functional data case. This method can be considered as a new raw-data method without any use of a dimension reduction, since it performs the clustering directly to the discrete observation of the curves or functions, which can initially be used to identify the number of clusters in the curves.

In the second method, we extend the weighted spatial ranks (WSR) method that has been introduced in Chapter 4 to the functional data. This method can be considered as one of the 2-stage methods, or the filtering methods, where it first approximate the curves into some basis functions and then perform the clustering using the basis expansion coefficients and the functional principle components scores. As discussed earlier, the main idea behind WSR is to define a dissimilarity measure locally based on a localized version of spatial ranks. As a result, the proposed method can be used to determine the assumed number of clusters, and to assign each curve to its cluster. Both methods are completely data-driven and easy to compute without any need to estimate parameters of the underlying distributions, which make them robust against distributional assumptions. Different numerical examples from simulated and real data have been introduced in order to check the reliability of the proposed methods. Comparison between the existing methods, using the probability of misclassification, has been considered as well. The results showed that the two proposed methods give a quite reasonable clustering analysis.

The chapter is organized as follows. Section 5.2 gives a review of the important existing literature on the functional data clustering methods giving numerical examples and comparisons. Along the line of Jacques and Preda (2014), we classify them into four groups, raw-data methods, filtering methods, adaptive methods and distance-based methods. In Section 5.3, we discuss the curse of dimensionality in the traditional forward search method and the ability of using the forward search based on functional spatial ranks. In Section 5.4, we propose the functional data clustering based on spatial ranks. Numerical results based on simulation and real data, and other relevant discussions are contained in succeeding subsections. Finally, in Section 5.5, we propose the functional data clustering based on weighted spatial ranks with some numerical examples and comparisons with the other functional data clustering methods, which shows the good behavior of our methods.

5.2 Functional Data Clustering Methods

Many functional data clustering methods have been reviewed in this Section. We follow the same classification of the different approaches that Jacques and Preda (2014) presented. Actually, it classifies the functional data clustering approaches into four groups. The first group is the raw-data methods that consist of the methods that are directly working on the evaluation points of the curves without any use of approximation or dimension reduction. The second group is the filtering methods that start to approximate the curves into a known finite basis of functions and then perform the clustering approach using the basis expansion coefficients or the functional principle components scores. The third group is the adaptive methods that consist of the methods that are simultaneously performing dimensionality reduction of the curves and clustering leading to functional representation of data depending on clusters. Finally, the fourth group which is known as the distance-based methods and it is clearly obvious from the name that these methods use some clustering algorithms based on specific distances for functional data, like K-means based on the distance d_0 (the L_2-metric) or with the semi-metric d_2.

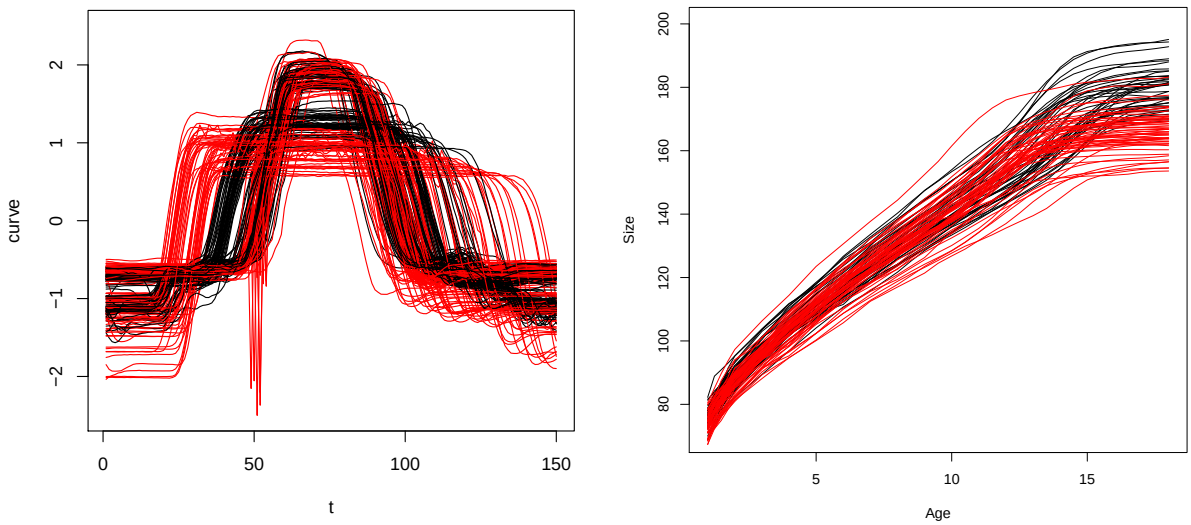
For more illustration and explanation of these different methods, we have chosen two case studies to cover a comparison between these methods throughout this Section. The first case study is the Berkeley Growth Study (Tuddenham and Snyder, 1954). It is extensively used in the literature related to the functional data analysis. It gives the heights of 54 girls and 39 boys in centimeters measured at a set of 31 ages from 1 to 18 years old. Our aim is to determine whether the resulting clusters reflect gender differences or not. The data is available in the `fda` package of **R**. For more comprehensive details of growth data, the reader is referred to Ramsay and Silverman (2005).

The second case study is the Gun-Point data (Ratanamahatana and Keogh, 2004). It is a content-based video retrieval, as it is coming from the video surveillance domain. It

consists of 200 instances and two clusters, each containing 100 instances that were created using one female actor and one male actor in a single session and for each instance there are 150 data points. The first cluster is known as Gun-Draw class, where in this one the actors have their hands by their sides, and draw a replicate gun from a hip-mounted holster, then they point it at a specific target for approximately one second, after that they return the gun to the holster, and their hands to their sides. The second cluster is known as Point class, where the actors have their gun by their sides, and point with their index fingers to a target for approximately one second, and then return their hands to their sides. The centroid of the actor's right hands in both X and Y axes has been tracked in the two classes and the similarity between the two different moves, makes a subtle distinction between the clusters. The overlapping between the classes makes it hard to discriminate the two groups in this data, even for supervised classifiers. The data can be found in UCR Time Series Classification Archive. For more details of gun-point data, the reader is referred to Ratanamahatana and Keogh (2004). Figure 5.1 shows both of gun-point and growth curves. The figure shows clearly the two groups in each dataset, where the black color refers to curves in first cluster while the red color refers to the curves in second cluster. From now on, we will use the black color to refer to curves in first cluster and red color to refer to the curves in second cluster. In this Section, the two case studies are comprehensively reviewed. Further analysis of the data sets is given in the next two Sections where our proposed methods are considered in full.

5.2.1 Raw Data Methods

The raw-data methods are considered the simplest methods among the other functional data clustering methods, as they are directly clustering the curves based on the basis of their evaluation points or based on the discretization of the underlying data (Jacques and Preda, 2014). In other words, when the functions are considered as discrete observation



(a) Gun-point data

(b) Growth data

Figure 5.1: Gun-point and growth data curves with two groups for both datasets. Black curves are the curves in cluster 1, and red curves are the curves in cluster 2.

points, then the raw-data methods can be directly applied by performing the clustering on the discretized points. These methods neither need to reconstruct the functional form of the data nor any filtering techniques like using the spline coefficients or the functional principle component analysis.

Some of the disadvantages related to these methods when the evaluation points of the curves are considered are that, they do not take into account the functional feature of data like the continuity and derivatives. In addition, when the curves observed at different evaluation points then such a case cannot be taken into account by these methods. Moreover a loss of information can be happened if the researcher extract the true signal from noisy data and use some specific metric related to the functional feature of the data (Jacques and Preda, 2014). The previous deficiencies make the using of such methods is not preferable, however using them leading sometimes to interesting results. Bouveyron

and Brunet (2013) introduced a survey on the clustering techniques for high-dimensional vectorial data that have to be used when the number of evaluation points being usually large. For more details about these techniques, see Bouveyron and Brunet (2013).

On the other hand, when the raw-data clustering methods are based on the discrete observation of the curves, in this case they will not suffer from the previous disadvantages and deficiencies that have been mentioned above. If we do not use the basis expansion of the evaluation points, with their constraints on the nature of curves, this will lead to getting better use of the original curves without losing any information. From now on, we will use the raw-data methods concept to denote to these methods that depending on the discrete observation of the curves (discretized case), where we will use them throughout this chapter.

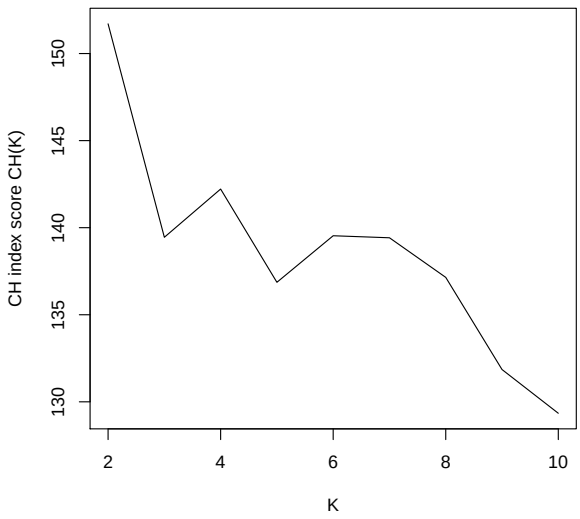
An example of the raw-data method is the work that has been introduced by Boullé (2012), who introduced a clustering algorithm based on the probability distribution of the number of discrete observations of the curves. Other methods can be well reviewed from the survey on the functional data clustering by Jacques and Preda (2014), where they introduced some numerical illustration which includes implementation of some well-known standard clustering methods. In the next part, we implement ten of the raw-data methods on the two case studies that we discussed above, and give some analysis regarding the clustering results. These methods are: K-means, K-Centroids, linkage based methods (hclust), model-based clustering (mclust), high dimensional data clustering (HDDC), mixtures of probabilistic principle component analysers (MixtPPCA), partitioning around medoids clustering (PAM), convex clustering (cclust) based on the hard competitive learning method, clustering with divisive analysis (diana), and clustering large applications (clara). The words in brackets refer to well-known functions and packages in **R**. Figure 5.2 shows the CH index scores (Calinski and Harabasz, 1974) and k-means clustering, based on the algorithm of Lloyd (1957), for the discretized gun-point and growth datasets.

For the gun-point data, CH-index suggests 2 clusters and accordingly K-means assigned the different curves in 2 clusters with 50% misclassification rate. Table 5.1 and 5.2 give the probability of misclassification based on the different functional data clustering approaches for the two datasets. Comparing the curves in panel (a) of Figure 5.1 and panel (b) of Figure 5.2 shows the misclassification clearly. For the growth data, CH index suggests 7 clusters; clearly, the K-means behaves so poorly in this real dataset, where it failed to give the right clustering.

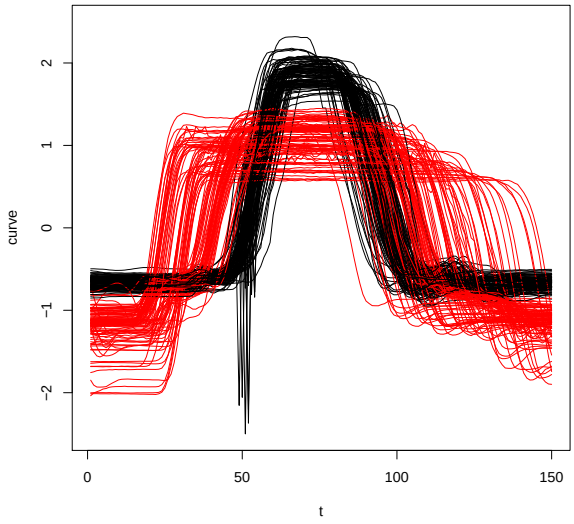
Compared to K-means, the K-centroids clustering based on the algorithm of Leisch (2006) gives the same result for gun-point data. It is required to specify the number of clusters in the used algorithm. Figure 5.3 shows the K-centroids clustering for the two functional datasets, with misclassification rate 50% and 49.62% for gun-point and growth data respectively.

Next, the linkage based methods (hclust) has been considered, as a raw-data method as well, based on the three hierarchical methods; single, complete and average linkage methods. Figure 5.4 shows their dendrogram for the discretized gun point and growth datasets.

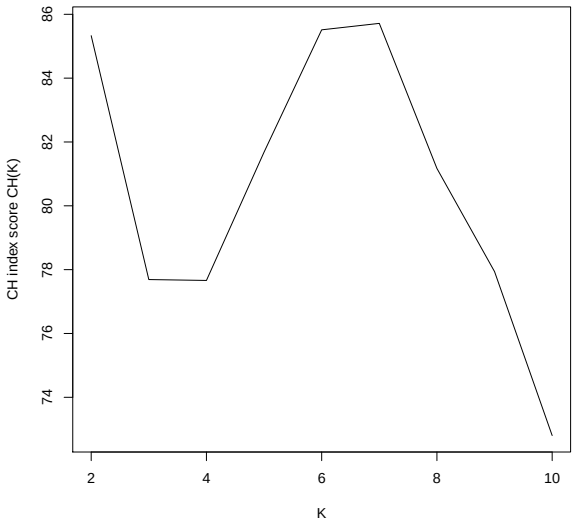
It can be clearly seen that, the three hierarchical methods give an evidence about the existence of two clusters in gun-point data. However, the complete linkage dendrogram shows clearer clustering structure, where it is obviously gives an evidence about the existence of two clusters as the distances in this dendrogram are much wider, and that makes it much easier visually to distinguish the number of clusters. On the other hand, the single linkage dendrogram for growth data does not give a clear clustering and it is very difficult visually to determine the number of clusters from the dendrogram. Obviously, the complete and average linkage dendrograms give an evidence about the existence of two clusters. For instance, cutting the complete and average linkage trees at $h = 120$ and $h = 70$ respectively gives 2 clusters for growth data. According to table 5.1 and 5.2, the



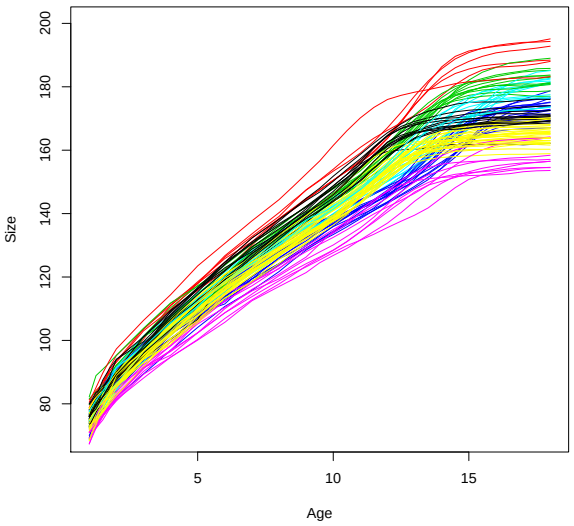
(a) Gun-point data: CH index



(b) Gun-point data: K-means

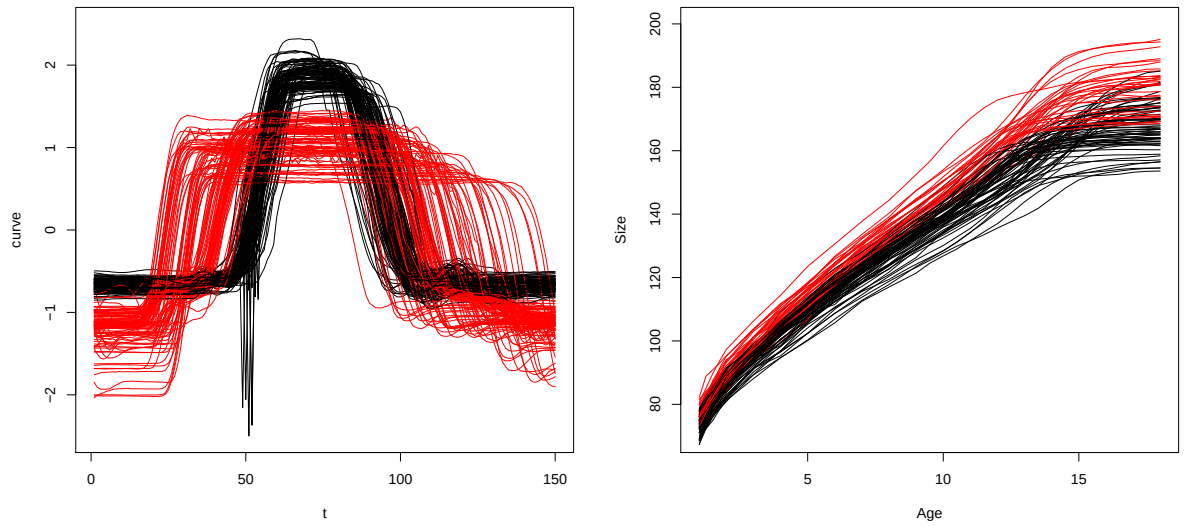


(c) Growth data: CH index



(d) Growth data: K-means

Figure 5.2: CH index scores and K-means clustering for the discretized gun-point and growth datasets. CH index suggests 2 and 7 clusters for gun point and growth data respectively.



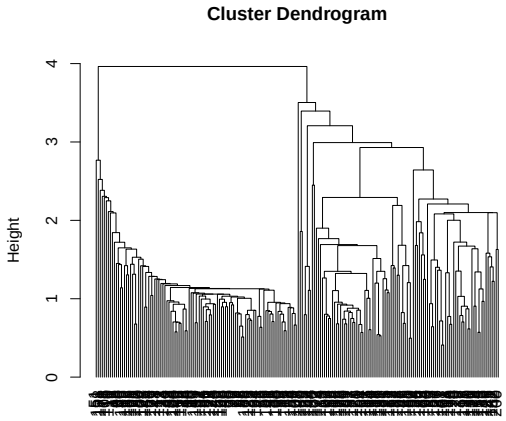
(a) Gun-point data

(b) Growth data

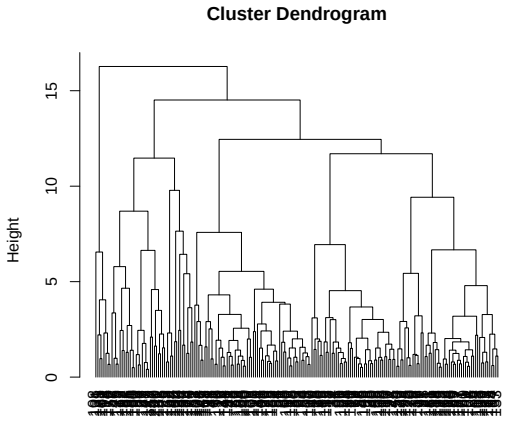
Figure 5.3: K-centroids clustering (kcca) for the discretized (a) gun-point data and (b) growth data.

average linkage method outperforms both of the single and complete linkage methods in the two datasets, where it gave smaller misclassification rate, which is 44% and 37.63% for gun-point and growth data respectively.

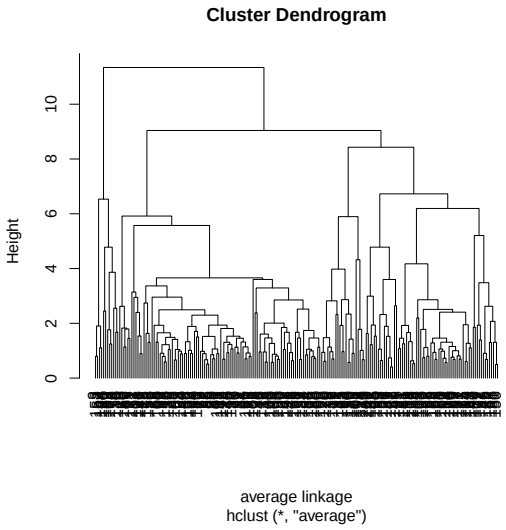
Regarding to the model-based clustering (mclust) that introduced by Fraley and Raftery (2002), Figure 5.5 shows the optimal model according to BIC and ICL for EM initialized by hierarchical clustering for parameterized Gaussian mixture models. For Gun point data, the best model according to BIC values is an equal-covariance model with 1 cluster only, where the maximum BIC value (BIC=150522.8), was for the EEE model (elliposidal multivariate normal model). For Growth data, the best model according to the maximum BIC values (BIC=-6091.822), was for the EEE model again with 1 cluster only, which behaves poorly in the two dataset because of giving wrong number of clusters. In order to calculate the misclassification rates, we assumed that $K = 2$ in mclust and



(a) Gun-point data: Single linkage

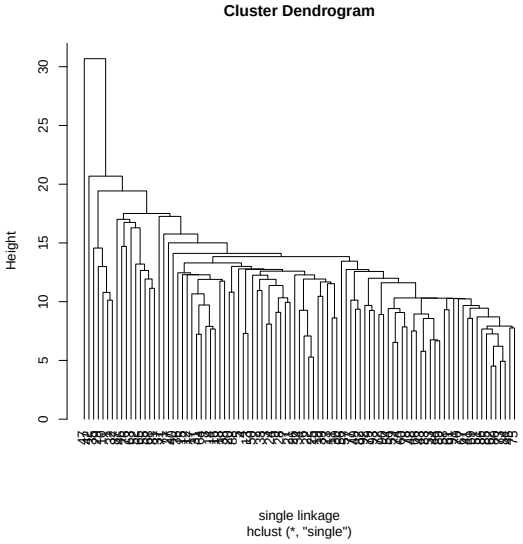


(b) Gun-point data: Complete linkage

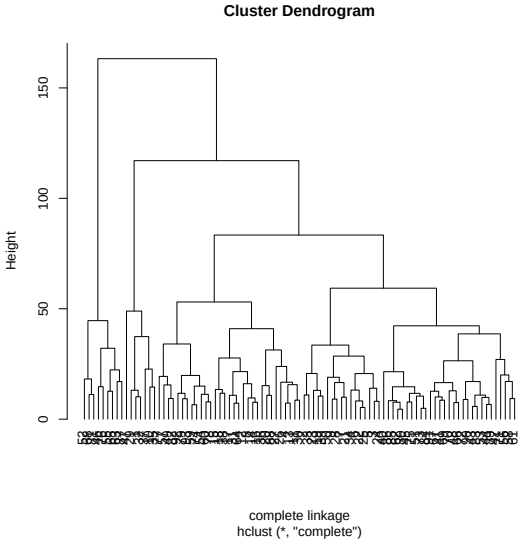


(c) Gun-point data: Average linkage

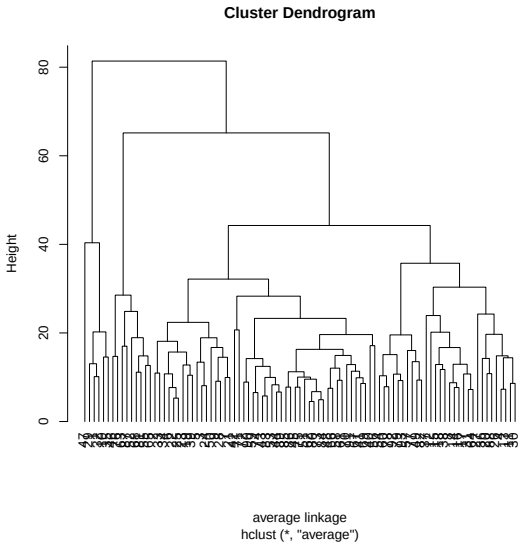
Figure 5.4: Linkage based methods for the discretized gun-point and growth datasets.



(d) Growth data: Single linkage



(e) Growth data: Complete linkage



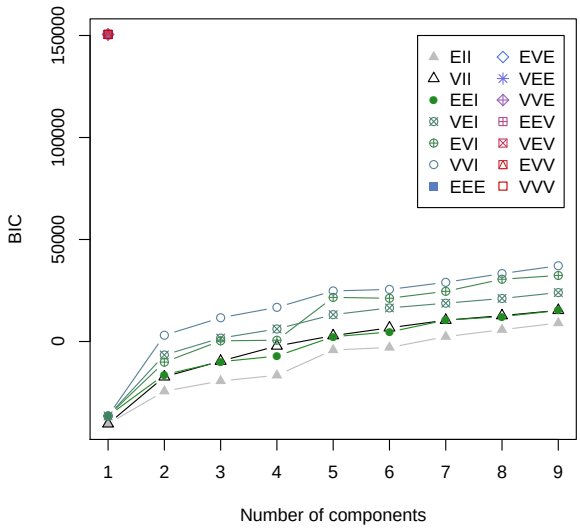
(f) Growth data: Average linkage

Figure 5.4: Continued.

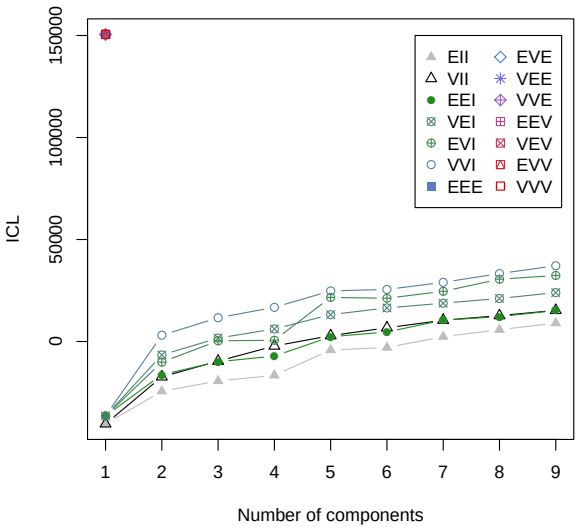
the rates were 50% for the gun-point data and 34.41% for growth data.

The high dimensional data clustering (HDDC) method has been introduced by Bouveyron, Girard, and Schmid (2007). It is a model-based clustering method based on the Gaussian mixture model when the data lives in subspaces with a lower dimension than the dimension of the original space. Like the other model-based clustering methods, it uses the expectation-maximization algorithm to estimate the parameters of the model. For gun-point data, if K is unknown, the best model is the most general model AKJBKQKDK (Berge et al., 2012) with 10 clusters and maximum BIC=46465.62, and for growth data, if K is unknown, the selected model is also AKJBKQKDK with 3 clusters and maximum BIC=-12037.43. The AKJBKQKDK model refers to that each class has its parameters and its proper noise, there is one parameter for each dimension, all classes have its proper orientation matrix and the dimensions are free and proper to each class. Figure 5.6 shows the Cattell's scree-test and BIC criterion based on HDDC when $K = 2$. The Cattell's scree-test gives the intrinsic dimension of each class which can be selected also by the BIC criterion. When $K = 2$, the rates of the misclassification error were 50% for the gun-point data and 51.61% for growth data.

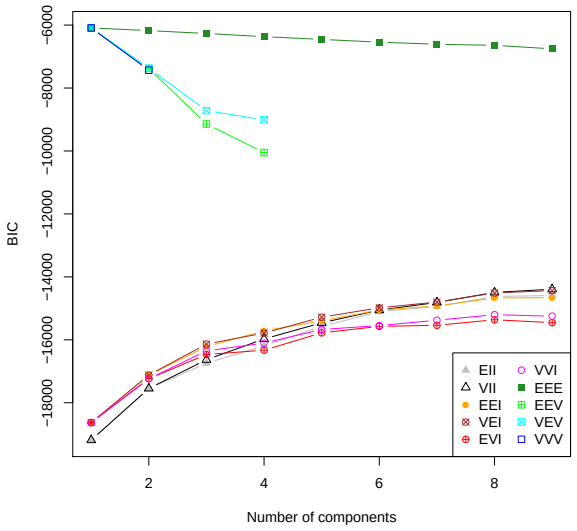
The mixtures of probabilistic principle component analysers method (MixtPPCA) has been introduced by Tipping and Bishop (1999). They used the PCA within a mixture-likelihood framework, based on a specific form of Gaussian latent variable model and the parameters can be estimated using an EM algorithm. We used the function (mppca.metabol) in **R**, by the authors Nyamundanda et al.(2010) in order to perform the clustering based on MixtPPCA. Figure 5.7 shows the best models with the number of groups and PCs based on the BIC criterion based on the MixtPPCA for the discretized gun point and growth datasets. For gun-point data, the plot of BIC suggests 4 PCs and 4 groups and for growth data it suggests 4 PCs and 2 groups. So it gave a wrong number of clusters for gun-point data. Based on MixtPPCA, when $K = 2$, then the rates of the



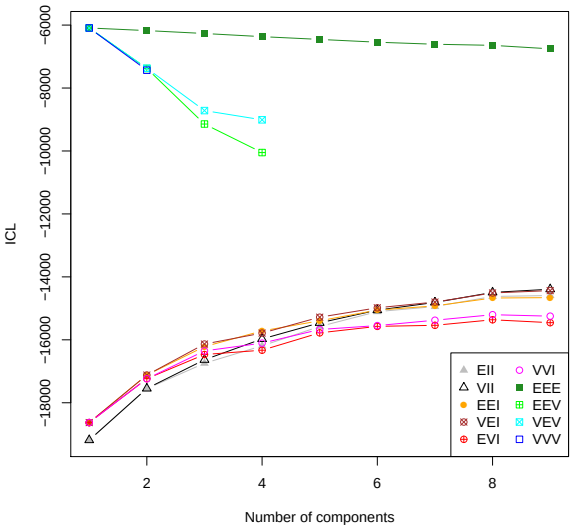
(a) Gun-point data: BIC



(b) Gun-point data: ICL

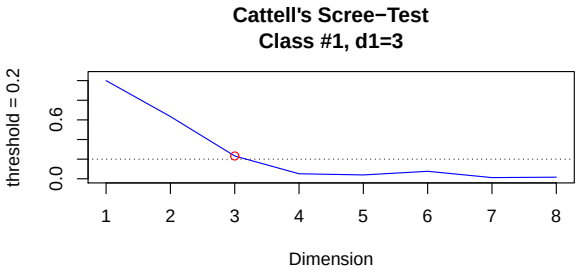


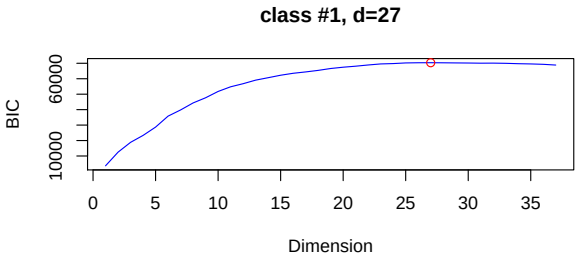
(c) Growth data: BIC

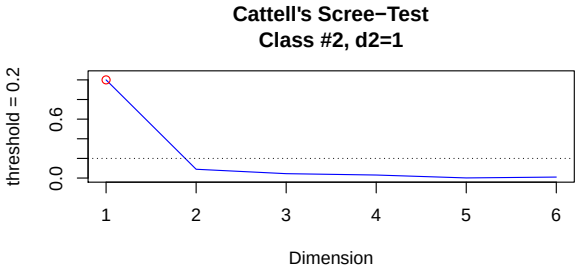


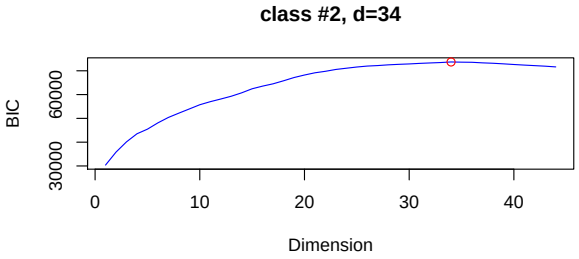
(d) Growth data: ICL

Figure 5.5: BIC and ICL plots based on Gaussian Mixture Models (GMM) for the discretized gun-point and growth datasets. One cluster is evident for both data.



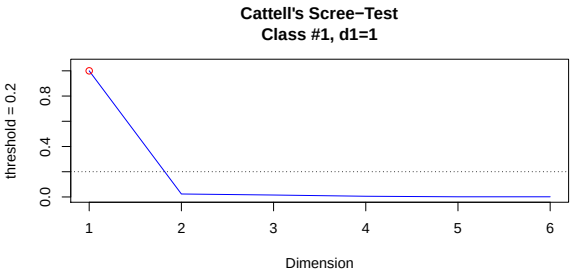


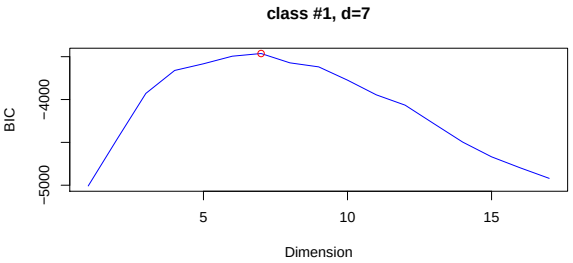


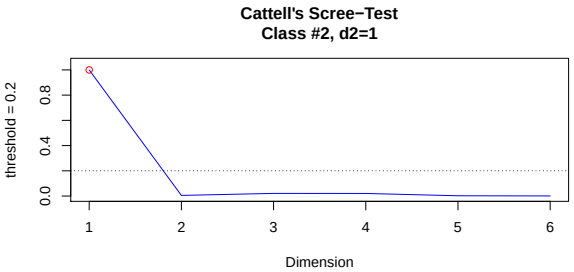


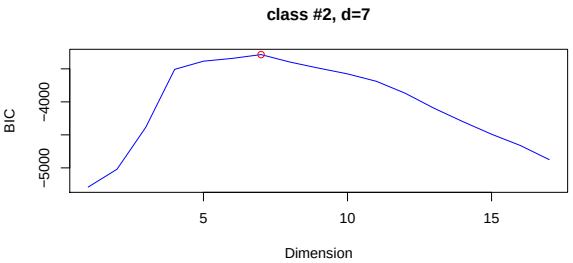
(a) Gun-point data: Cattell's test

(b) Gun-point data: BIC





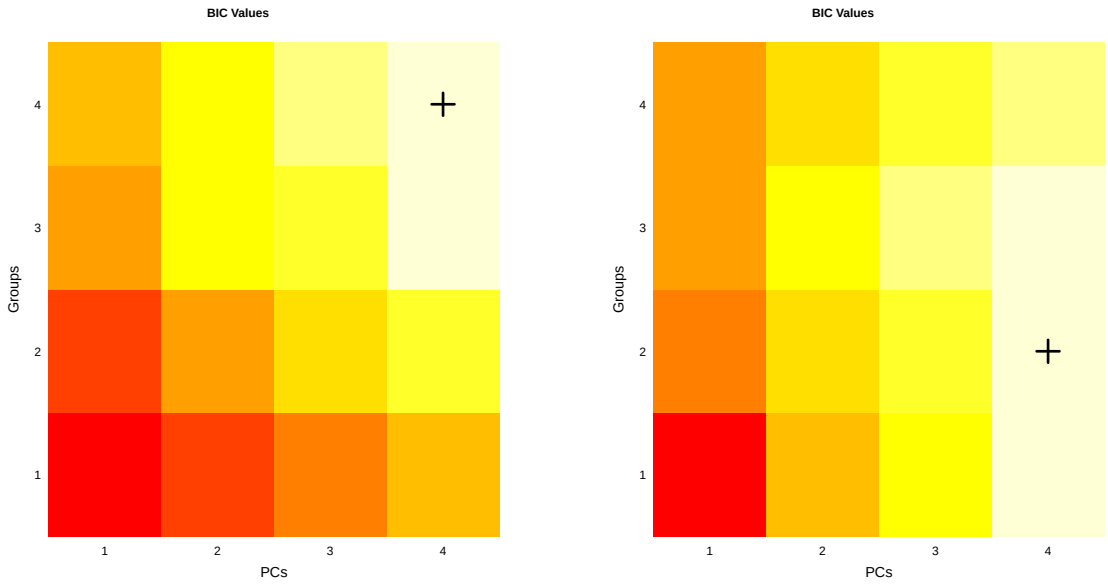




(c) Growth data: Cattell's test

(d) Growth data: BIC

Figure 5.6: Cattell's scree-test and BIC criterion based on High Dimensional Data Clustering (HDDC) for the discretized gun point and growth datasets suggest the intrinsic dimensions.



(a) Gun-point data

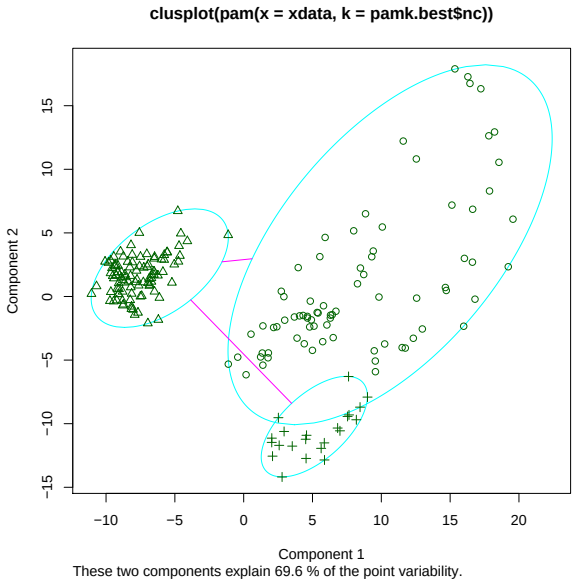
(b) Growth data

Figure 5.7: BIC plot based on the mixtures of probabilistic principle component analysers (MixtPPCA) for the discretized: (a) gun-point data suggests 4 PCs and 4 groups, and (b) growth data suggests 4 PCs and 2 groups.

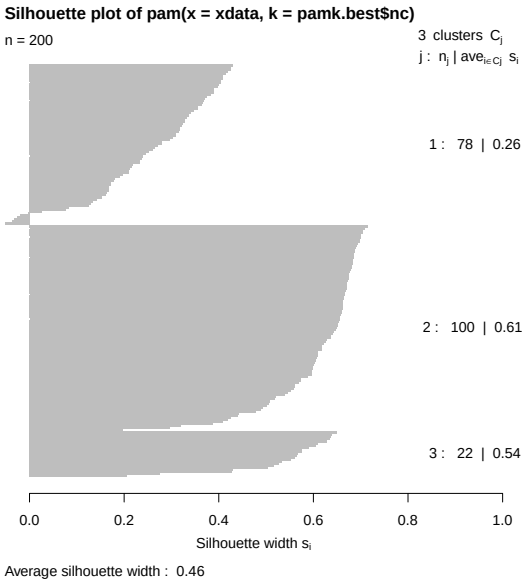
misclassification error will be 50% for the gun-point data and 31.18% for growth data.

We considered the partitioning around medoids (PAM) clustering (Reynolds et.al., 1992) as a raw-data functional clustering method. The optimum average silhouette width (Rousseeuw, 1987) has been considered on order to estimate the number of clusters. Figure 5.8 shows the bivariate cluster plot (clusplot) and silhouette plot based on the partitioning around medoids clustering for the discretized gun point and growth datasets. For the gun-point data, if K is unknown then the number of clusters estimated by optimum average silhouette width is 3, which is not the real number. On the other hand, if K is unknown in growth data, then the number of clusters estimated by optimum average silhouette width is the right number 2. According to PAM algorithm, when $K = 2$, then the rates of the misclassification error will be 50% for the gun-point data and 29.03% for growth data.

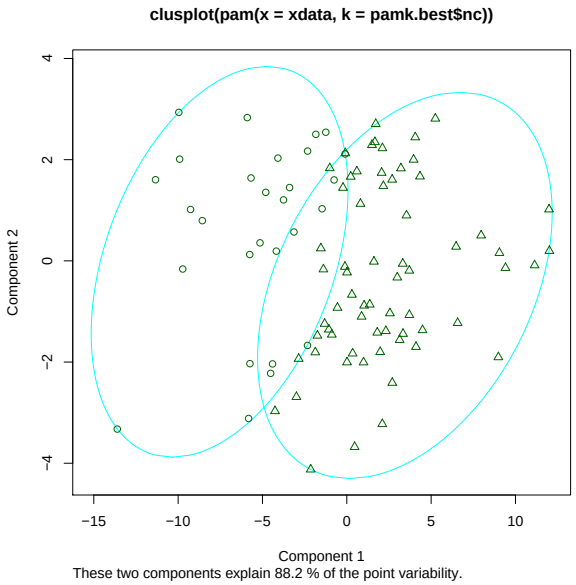
Clustering with Convex Clustering (cclust) based on the hard competitive learning



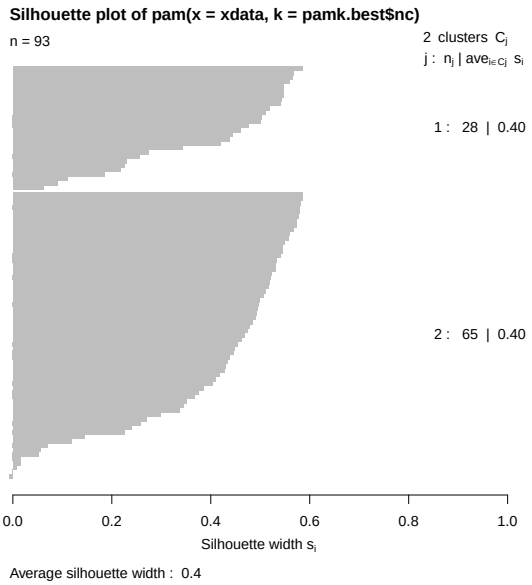
(a) Gun-point data: clusplot



(b) Gun-point data: silhouette plot

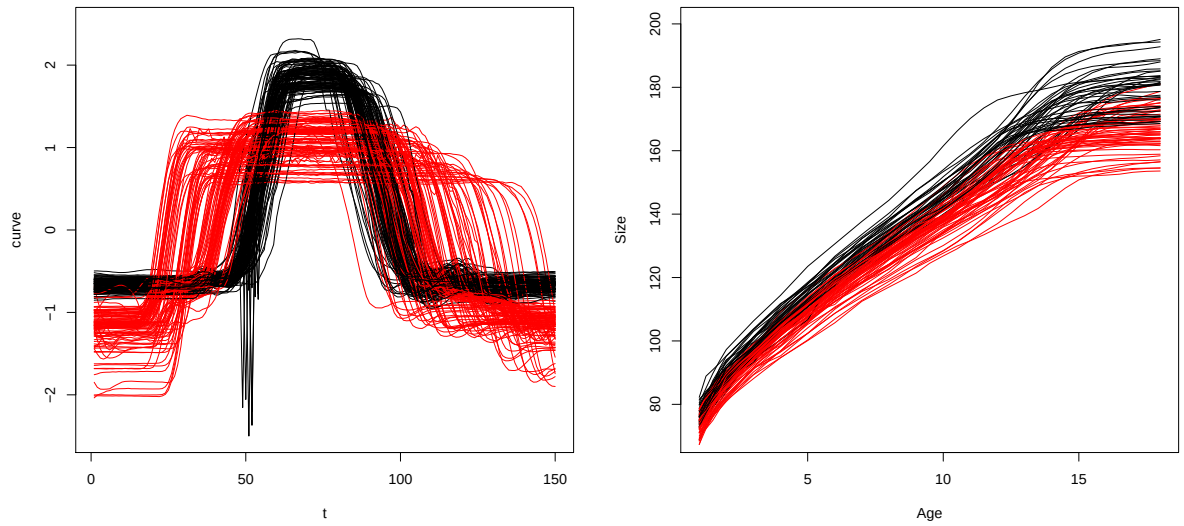


(c) Growth data: clusplot



(d) Growth data: silhouette plot

Figure 5.8: Bivariate cluster plot (clusplot) and silhouette plot based on the partitioning around medoids clustering (PAM) for the discretized: (a) gun-point data suggest 3 groups, and (b) growth data suggest 2 groups.



(a) Gun-point data

(b) Growth data

Figure 5.9: Convex clustering (cclust) based on the hard competitive learning method for the discretized: (a) gun point data and (b) growth data.

method has been introduced by Ripley (1996). The method works by randomly drawing an observation from x and moving the closest center towards that point. It is required to specify the number of clusters in this algorithm. Figure 5.9 shows the hard competitive learning clustering for the two functional datasets, with misclassification rate 50% and 50.39% for gun-point and growth data respectively.

DIANA (DIvisive ANAlysis Clustering) algorithm has been introduced by Kaufman and Rousseeuw (1990). It is a hierarchical clustering technique that constructs the hierarchy in the inverse order, which make a difference with the agglomerative method. Figure 5.10 displays the banner and dendrogram plots, where the banner shows the hierarchy of clusters and plots the diameter of each cluster being splitted. Like the other hierarchical methods, we can use the cutree command to cut the dendrogram into 2 groups. So, when $K = 2$, the rates of the misclassification error were 33.5% for the gun-point data and

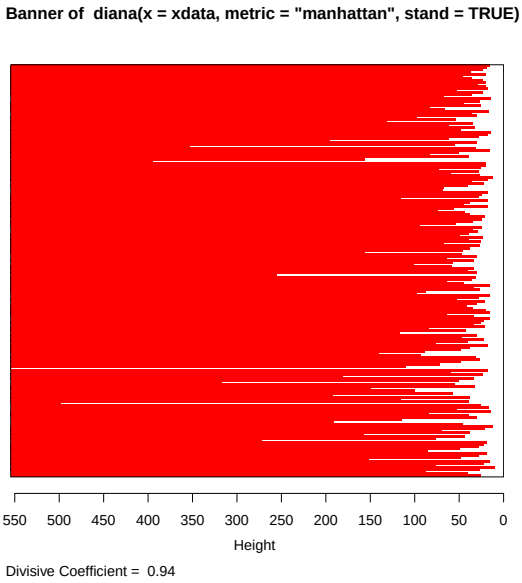
38.71% for growth data.

Kaufman and Rousseeuw (1990) suggested the clustering for large applications algorithm (CLARA), which performs a partitioning around medoids clustering with known number of clusters. The clara algorithm is an extension to the K-medoids approach for a large number of objects with the focus on clustering large numbers of elements in high dimensions. The main idea of this method is to start by clustering a sample from the dataset then assigning all the elements in the dataset to these clusters. It is required to specify the number of clusters in CLARA algorithm. Figure 5.11 shows the bivariate cluster plot (clusplot) and silhouette plot based on the clara algorithm for the discretized gun point and growth datasets. The number of clusters estimated by Calinski- Harabasz index, which suggests 2 clusters for both gun-point and growth data. The rates of the misclassification error based on CLARA algorithm are 50% for the gun-point data and 33.33% for growth data.

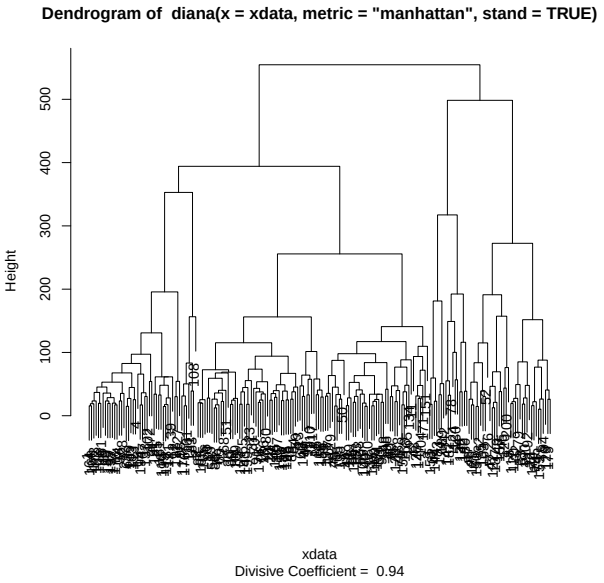
From the previous comparison, we can clearly notice that, the majority of these raw-data methods gave the same rate of misclassification error for gun-point data, which is 50%. This is due to the overlapping between the two classes that makes it hard to discriminate the two groups in this data. However, the least misclassification rate was for DIANA algorithm with a minimum percentage (33.5%). On the other hand, when we consider the growth data, we can easily see that the methods gave different rate of misclassification error and PAM algorithm gave the smallest rate (29.03%).

5.2.2 Filtering Methods

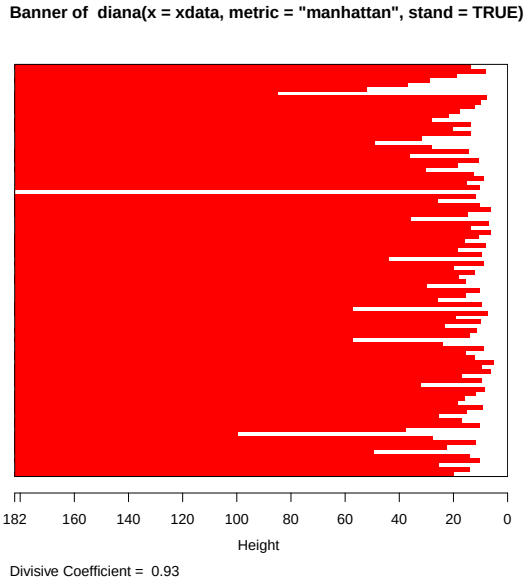
As we discussed earlier, in the filtering methods we start with approximating the curves into a known finite basis of functions and then we can perform the clustering approach using the basis expansion coefficients or the functional principle components scores. So, in the first step, which known as the filtering step (James and Sugar, 2003), we try to reduce



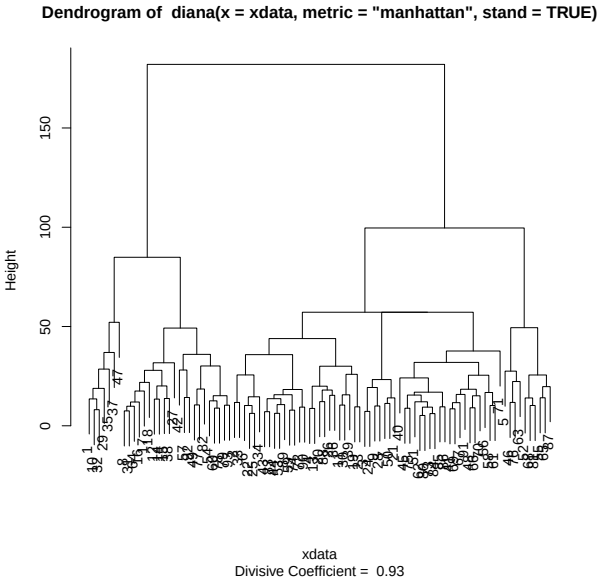
(a) Gun-point data: banner plot



(b) Gun-point data: dendrogram

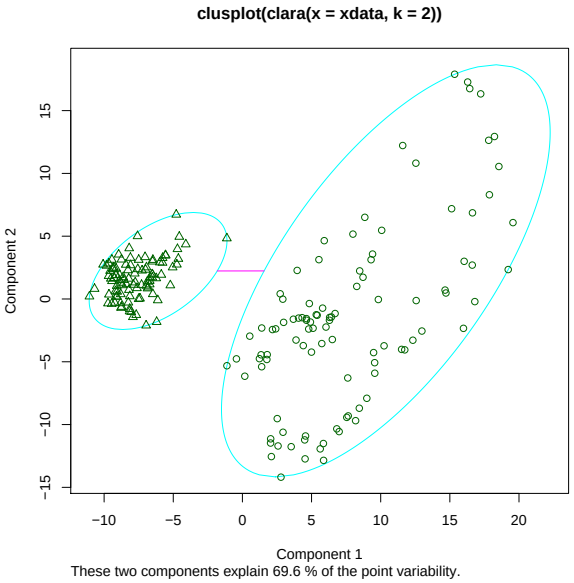


(c) Growth data: banner plot

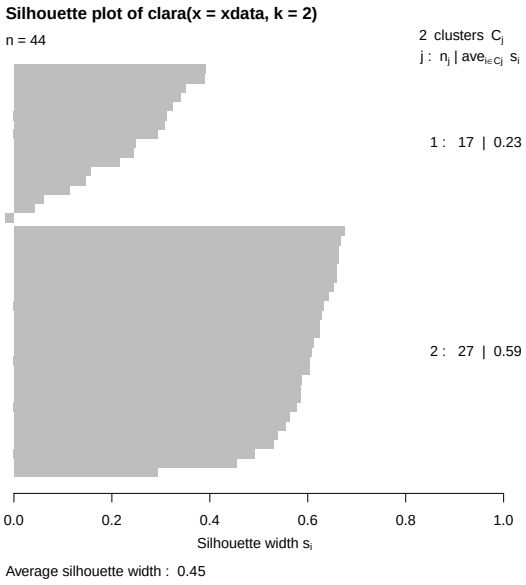


(d) Growth data: dendrogram

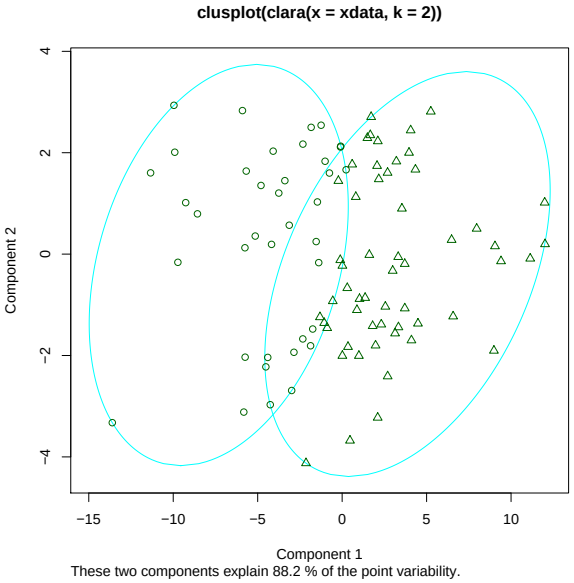
Figure 5.10: Banner and dendrogram plots of a divisive hierarchical clustering based on the DIvisive ANAlysis algorithm (DIANA) for the discretized gun-point and growth datasets.



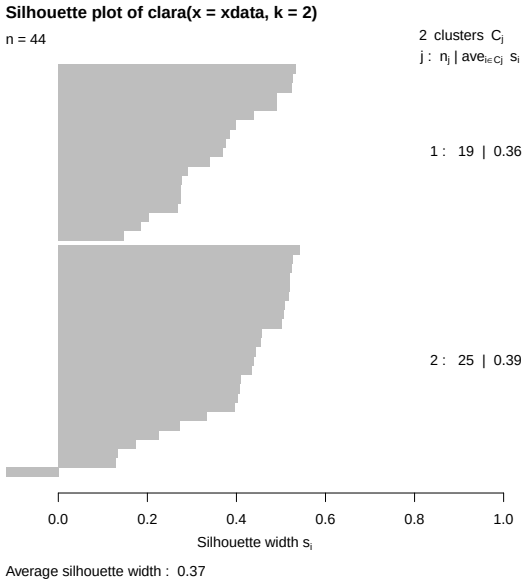
(a) Gun-point data: clusplot



(b) Gun-point data: silhouette plot



(c) Growth data: clusplot



(d) Growth data: silhouette plot

Figure 5.11: Bivariate cluster plot (clusplot) and silhouette plot based on the Clustering Large Applications (CLARA) algorithm for the discretized: (a) gun-point data and (b) growth data. Two groups are evident for both data.

the dimension of the data by approximating the curves into a finite basis of functions, while in the second step, which known as the clustering step, we use the clustering methods for the finite dimensional data that we obtained from the first step.

Two different tools are usually used in the filtering step. The first one is the functional principle components analysis (FPCA) that requires reconstructing the functional nature of the curves, by approximating them into a finite number of their principal component scores. The second tool is the spline basis which considers a common choice due to their optimal properties (Wahba, 1990). However the popularity of these two tools, some other tools like the ANOVA's coefficients for functional data using a mixture of Gaussian distributions that introduced by Serban and Jiang (2012), can be associated to the filtering methods category (Jacques and Preda, 2014). In the next two subsections, we discuss the filtering methods based on both functional principle components scores and the spline coefficients in details.

Filtering Methods on FPCA Scores

Recently, using of functional principal components analysis (FPCA) has been clearly increased, because it is one of the main tools that can be used in the exploratory analysis, modeling and classification of functional data. The main idea of the FPCA is that, it gives a small dimension space that captures the main patterns of variability of the data. The functional PCA is different from the multivariate PCA since the eigen functions that exhibit the main modes of variability of the data are also functions and can be naturally interpreted as modes of variability varying along time (for more details, see Ramsay and Silverman, 2002).

A recent review paper has been introduced by Hall (2011), who provided a detailed overview on the roles of FPCA in functional linear regression and density estimation for functional data. Along the line of Hall (2011), Shang (2014), in his recent survey paper, describes the roles of FPCA in functional data exploratory analysis, modeling

and forecasting functional data, and classification of functional data. There are two major viewpoints in the existing literature regarding to the use of FPCA; the linear operator viewpoint and the kernel viewpoint. For the first one, many literatures have been proposed, such as Besse (1992), Cardot et al. (1999), Bosq (2000), Ferraty and Vieu (2006), and Mas (2008). On the other hand, a particular attention is paid to the view of FPCA from the kernel aspect, for instance the work by Yao et al. (2005), Hall and Vial (2006), Hall and Hosseini (2006), Hall et al. (2006), Hall and Horowitz (2007), and Shen (2009) are related to the FPCA from the kernel aspect (Shang, 2014). For more comprehensive review of this subject, the reader is referred to Hall (2011) and Shang (2014).

The aim of FPCA is to represent the functional data in a small dimension space that captures the main modes of variability in the curves. Suppose that $\mathbb{H}$ is a Hilbert space of the real functions on $\mathcal{T}$, such that for each $t \in \mathcal{T}$, the evaluation functional L_t, which associates f with $f(t), L_t f \to f(t)$, is a bounded linear functional (Ingrassia and Costanzo, 2005), then the functional principal component's target is the orthogonal decomposition of the variance function:

$$\nu(t,u) = \frac{1}{n-1} \sum_{i=1}^{n} \left\{x_i(t) - \overline{x}(t)\right\} \left\{x_i(u) - \overline{x}(u)\right\}. \tag{5.2.1}$$

The decomposition of the variance function for the functional version can be defined as:

$$\nu(t,u) = \sum_{j} \lambda_j \xi_j(t) \xi_j(u), \tag{5.2.2}$$

where ξ_j are called the principle component weight functions, and $\lambda_j, \xi_j(t)$ satisfy the eigen-equation, $\left\langle \nu(s,), \xi_j \right\rangle_h = \lambda_j, \xi_j(u)$, and the eigenvalues $\lambda_j = \int_{\mathcal{T}} \xi_j(t) \nu(t,u) \xi_j(u)\, dt\, du$ are positive and non-decreasing while the eigen-functions must satisfy the constraints: $\int_{\mathcal{T}} \xi_j^2(t) dt = 1$ and $\int_{\mathcal{T}} \xi_j \xi_i(t) dt = 0$; $(i < j)$. Then, the principle component scores of the

units in the dataset are the values S_i given by:

$$S_i^{(j)} = \langle x_i, \xi_j \rangle = \int_{\mathcal{T}} \xi(t) x_i(t) dt. \tag{5.2.3}$$

The decomposition (5.2.2) defined by the eigen-equation permits a reduced rank least squares approximation to the covariance function ν. Thus, the leading eigen-functions ξ define the principal components of variation among the sample functions x_i. On the other hand, one can use the mean function $\mu(t)$ in order to get the principle components $\{C_j\}_{j \geqslant 1}$ such that:

$$C_j = \int_0^T \big(X(t) - \mu(t)\big)\xi_j(t) dt, \tag{5.2.4}$$

where $\mu = \big\{\mu(t) = \mathbb{E}\big[X(t)\big]\big\}_{t \in \mathcal{T}}$, the principal components $\{w_j\}_{j \geqslant 1}$ are zero-mean uncorrelated random variables with variance $\lambda_j, j \geqslant 1$ and X is a L_2-continuous stochastic process:

$$\forall t \in \mathcal{T}, \quad \lim_{h \to 0} \mathbb{E}\big[\, |X(t+h) - X(t)|^2 \,\big] = 0, \tag{5.2.5}$$

so, the KarhunenŰLoeve expansion (Karhunen, 1947; Loève, 1945) holds:

$$X(t) = \mu(t) + \sum_{j \geqslant 1} C_j \xi_j(t), \quad t \in \mathcal{T}, \tag{5.2.6}$$

and the best approximation in norm L_2 of $X(t)$ can be obtained by truncating (5.2.6) at the first q terms (Saporta 1981),

$$X^q(t) = \mu(t) + \sum_{j=1}^{q} C_j \xi_j(t), \quad t \in \mathcal{T}. \tag{5.2.7}$$

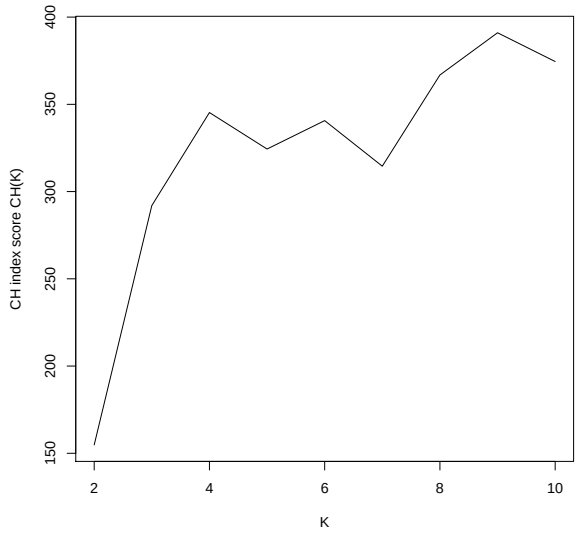
An example of the filtering method is the work that has been introduced by Peng and Müller (2008), where the K-means algorithm is used on principal component scores and the number of principal component scores is selected according to the percentage

of explained variance. Now, we apply the ten methods that we used before, based on the FPCA scores of the first two harmonics on the two case studies. Figure 5.12 shows the CH index scores and K-means clustering for the FPCA of gun-point and growth datasets. For the gun-point data, CH-index suggests 9 clusters and accordingly K-means assigned the different curves in 9 clusters. However, if we consider $K = 2$, then the rate of misclassification error is 50%. For the growth data, CH index suggests 5 clusters and when $K = 2$, the rate of misclassification error is 48.13%. It can be clearly seen that, the K-means algorithm based on the FPCA scores behaves so poorly in the two datasets.

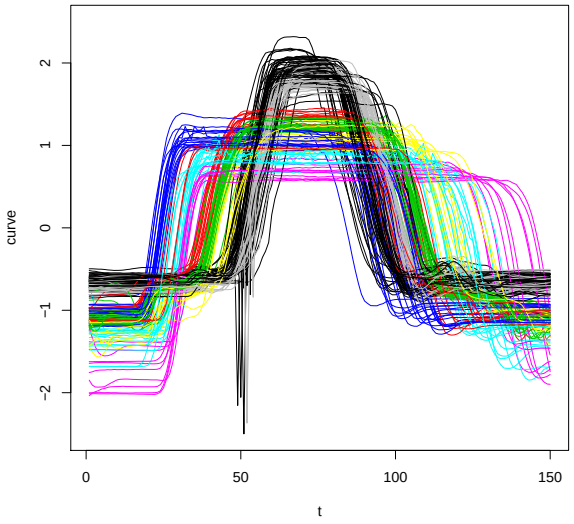
Figure 5.13 shows the K-centroids clustering based on the FPCA scores for the two functional datasets, with misclassification rate 50% and 51.38% for gun-point and growth data respectively. Compared to Figure 5.1, the K-centroids based on FPCA scores gives totally different clustering for growth data, as we can see that the curves have been assigned wrongly and the misclassification rate has been increased consequently.

Regarding to the linkage based methods (hclust) based on the FPCA scores, Figure 5.14 shows the dendrogram for the single, complete and average methods based on FPCA scores for gun point and growth datasets. For the gun-point data, the average linkage method outperforms both of the single and complete linkage methods, where it gave smaller misclassification rate, which is 39%, however it does not behave well with growth data, where the complete linkage gave smaller misclassification rate equals to 51.61%.

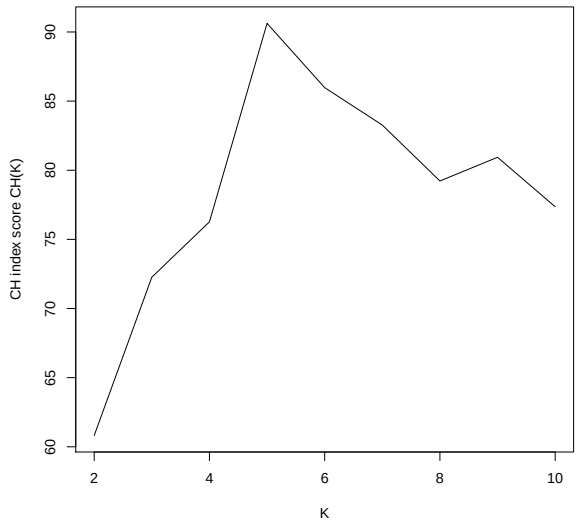
Figure 5.15 shows the optimal model according to BIC and the classification uncertainty plot for mclust algorithm based on the FPCA scores of gun point and growth datasets. For Gun point data, the best model according to BIC values is an equal-covariance model with 9 clusters, where the maximum BIC value (BIC=1360.639), was for the VEV model (elliposidal equal shape) which behaves poorly and giving wrong number of clusters. For Growth data, the best model according to the maximum BIC values (BIC=308.4225), was for the model EEV (ellipsoidal, equal volume and shape) model



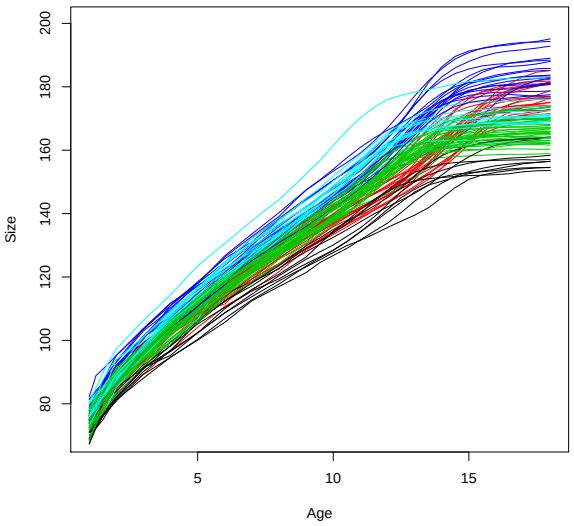
(a) Gun-point data: CH index



(b) Gun-point data: K-means

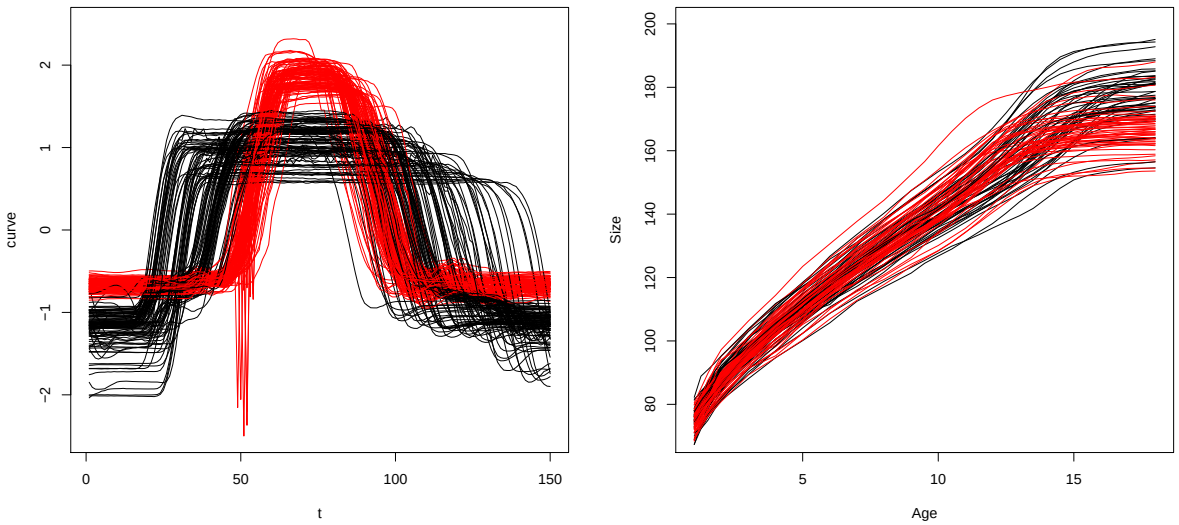


(c) Growth data: CH index



(d) Growth data: K-means

Figure 5.12: CH index scores and K-means clustering for the FPCA scores of gun-point and growth datasets. CH index suggests 9 and 5 clusters for gun point and growth data respectively.



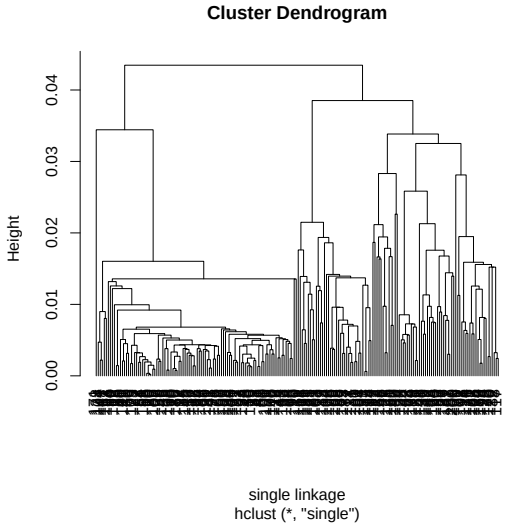
(a) Gun-point data

(b) Growth data

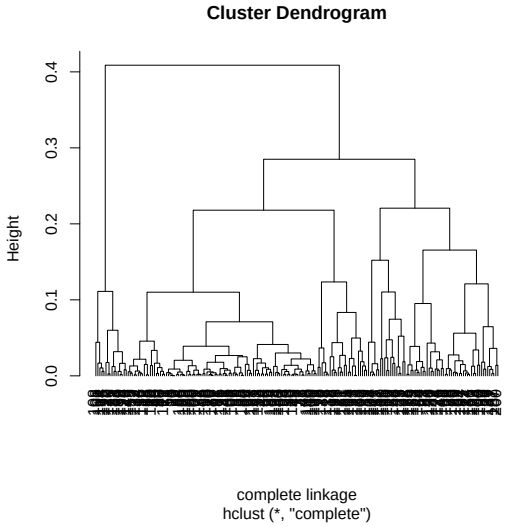
Figure 5.13: K-centroids clustering (kcca) for the FPCA scores of (a) gun-point data and (b) growth data.

with 2 clusters. Compared to the discretized mclust, when $K = 2$, the algorithm based on the FPCA scores gives better results for the two datasets, where the misclassification rates are 49.5% for the gun-point data and 3.23% for growth data.

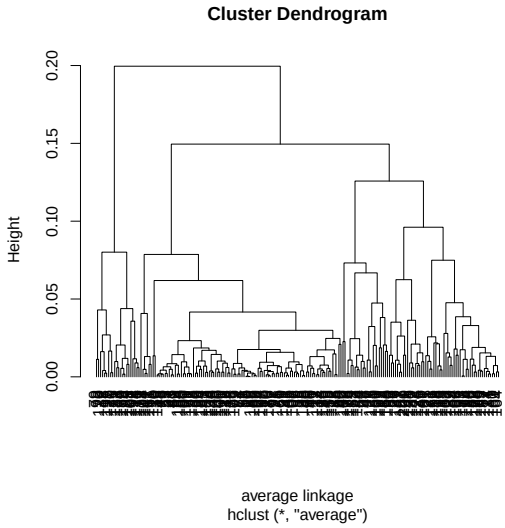
Now, we consider the high dimensional data clustering (HDDC) method based on FPCA scores. The HDDC algorithm based on FPCA scores cannot be applied for 2 components, so 3 components are considered for the two datasets. For gun-point data, if K is unknown, the selected model is AKJBKQKDK with 10 clusters and maximum BIC=2114.501. For growth data, if K is unknown, the selected model is also AKJBKQKDK with 2 clusters and maximum BIC=437.4071. Figure 5.16 shows the Cattell's scree-test and BIC criterion based on HDDC when $K = 2$. The rates of the misclassification error for HDDC based on the FPCA scores are 49.97% and 50.14% for the gun-point and growth data respectively.



(a) Gun-point data: Single linkage

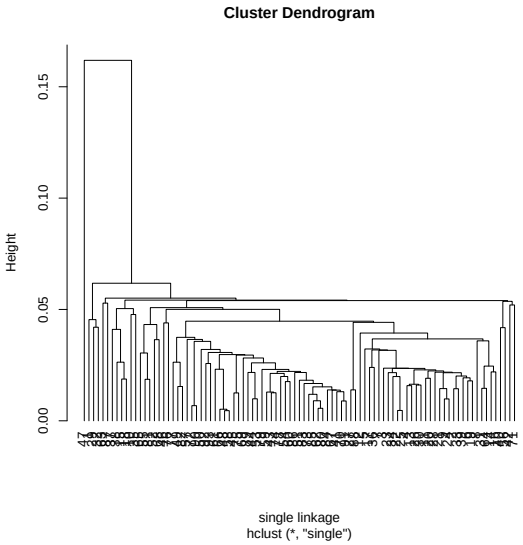


(b) Gun-point data: Complete linkage

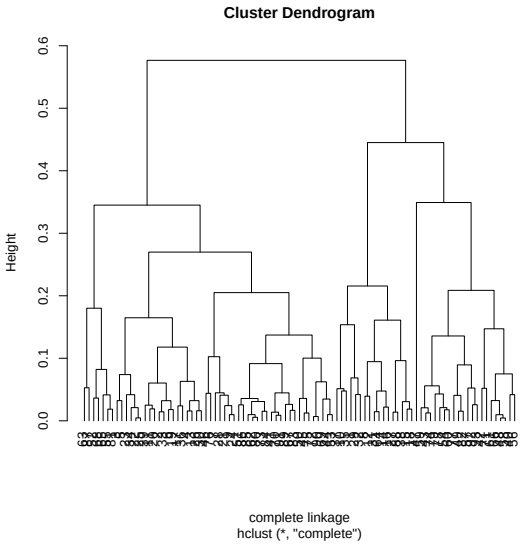


(c) Gun-point data: Average linkage

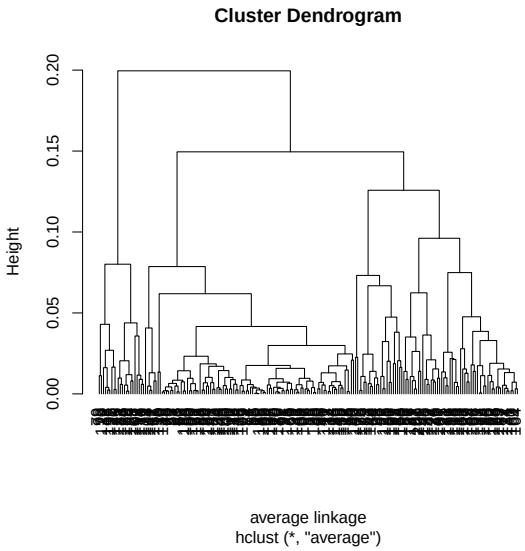
Figure 5.14: Linkage based methods for the FPCA scores of gun-point and growth datasets.



(d) Growth data: Single linkage

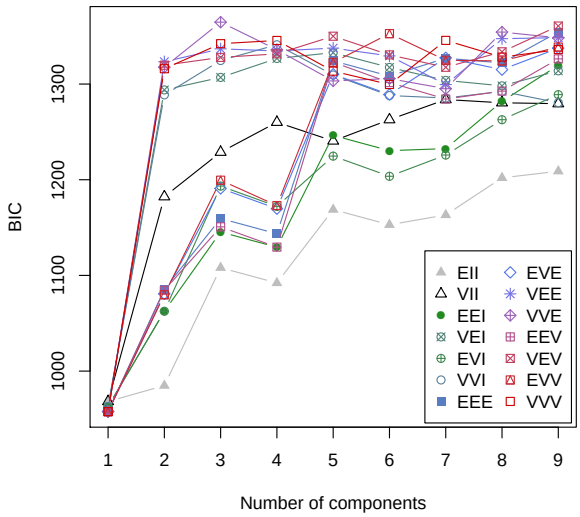


(e) Growth data: Complete linkage

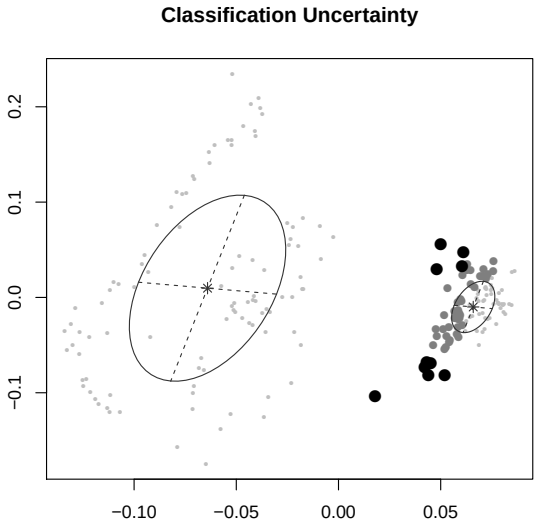


(f) Growth data: Average linkage

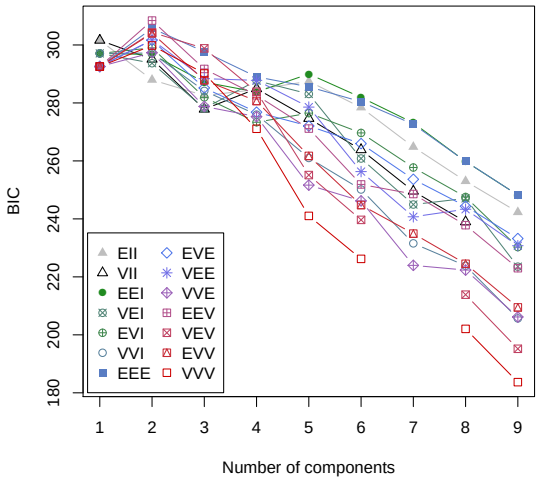
Figure 5.14: Continued.



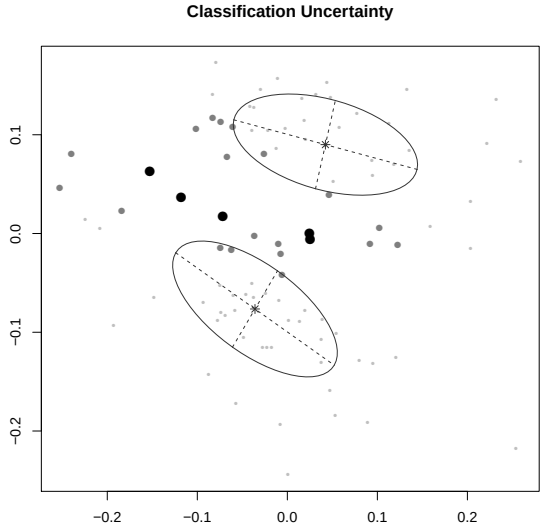
(a) Gun-point data: BIC



(b) Gun-point data: uncertainty plot

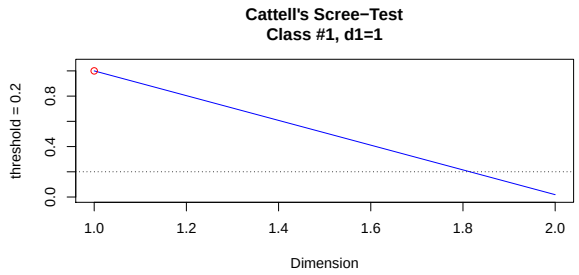


(c) Growth data: BIC

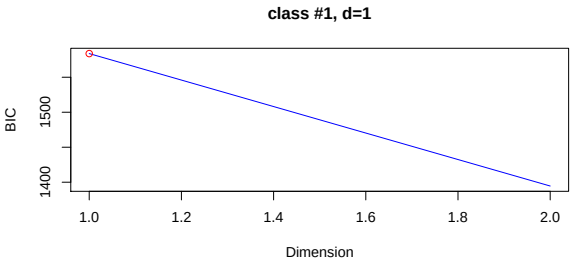


(d) Growth data: uncertainty plot

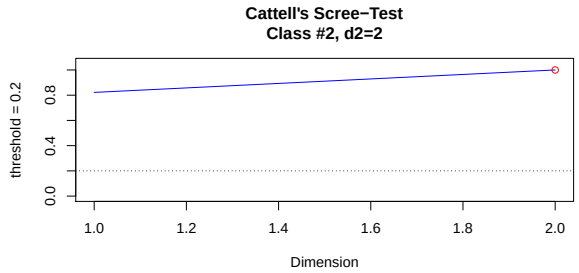
Figure 5.15: BIC and classification uncertainty plots based on Gaussian Mixture Models (GMM) for the FPCA scores of gun-point and growth datasets. Nine and two clusters are evident for gun-point and growth data respectively.

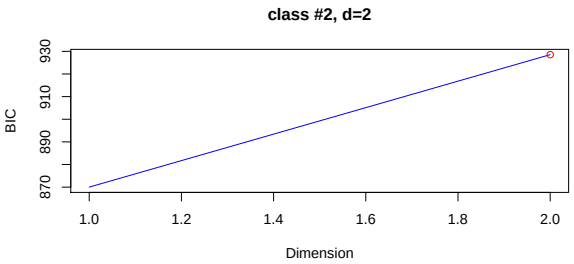


(a) Gun-point data: Cattell's test



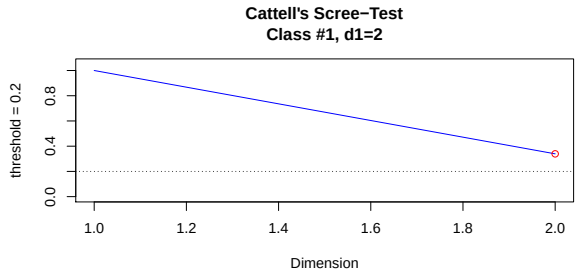
(b) Gun-point data: BIC

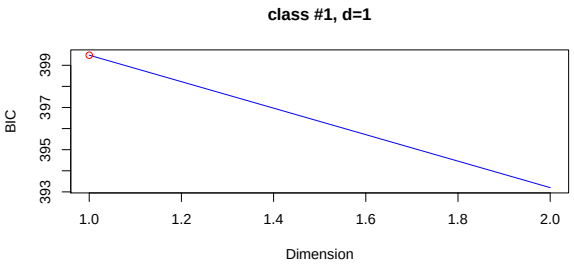


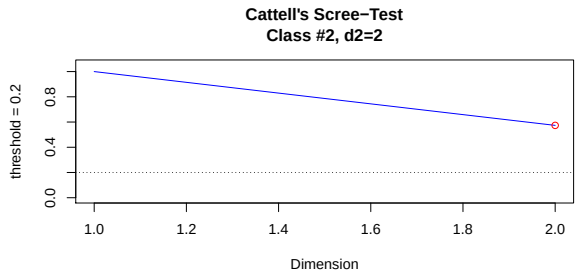


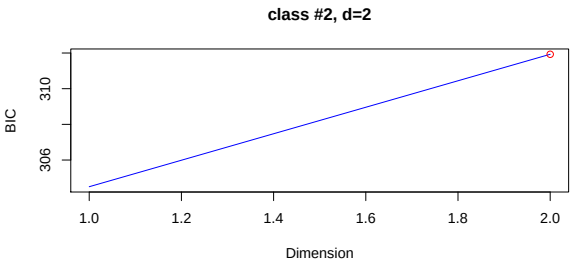
(a) Gun-point data: Cattell's test

(b) Gun-point data: BIC





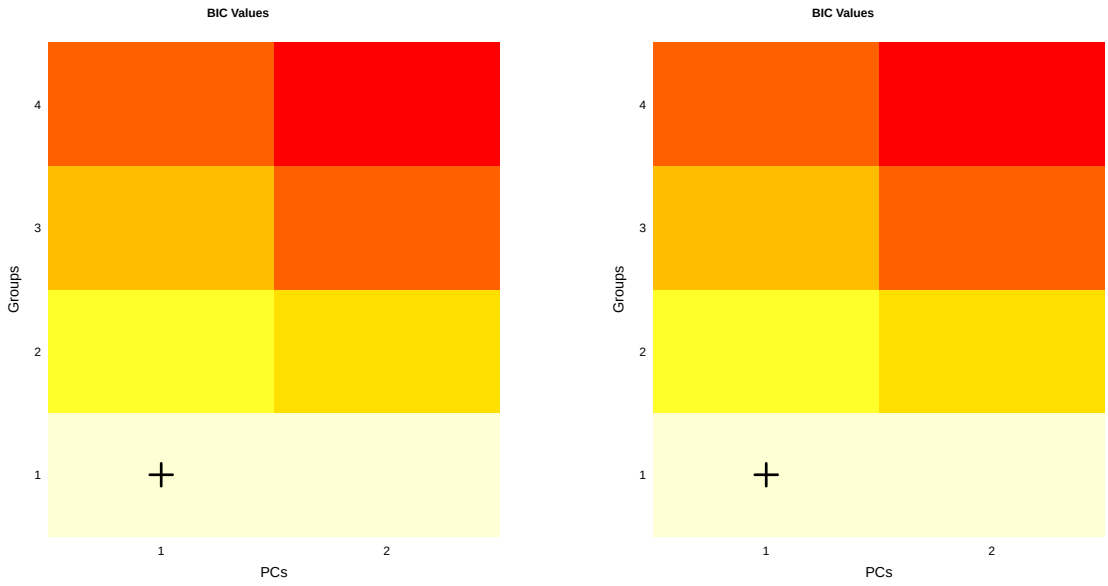




(c) Growth data: Cattell's test

(d) Growth data: BIC

Figure 5.16: Cattell's scree-test and BIC criterion based on High Dimensional Data Clustering (HDDC) for the FPCA scores of gun point and growth datasets suggest the intrinsic dimensions.



(a) Gun-point data

(b) Growth data

Figure 5.17: BIC plot based on the mixtures of probabilistic principle component analysers (MixtPPCA) for the FPCA scores of gun-point and growth datasets suggests 1 PC and 1 group for both data.

In Figure 5.17, the best models with the number of groups and PCs using BIC based on the mixtures of probabilistic principle component analysers for the FPCA scores of gun-point and growth data are shown. The MixtPPCA algorithm based on FPCA scores, cannot be applied for 2 components, so 3 components are considered for the two datasets. The plot of BIC suggests only 1 PC and 1 group and for both gun-point and growth data, which gives a wrong number of clusters. The rates of the misclassification error are 50% and 58.06% for the gun-point and growth datasets respectively.

Figure 5.18 shows the bivariate cluster plot (clusplot) and silhouette plot based on the partitioning around medoids clustering (PAM) for the first two harmonics of FPCA scores of gun point and growth datasets. For the gun-point data, if K is unknown then the number of clusters estimated by optimum average silhouette width is 4, which is not the right number. On the other hand, if K is unknown in growth data, then the number

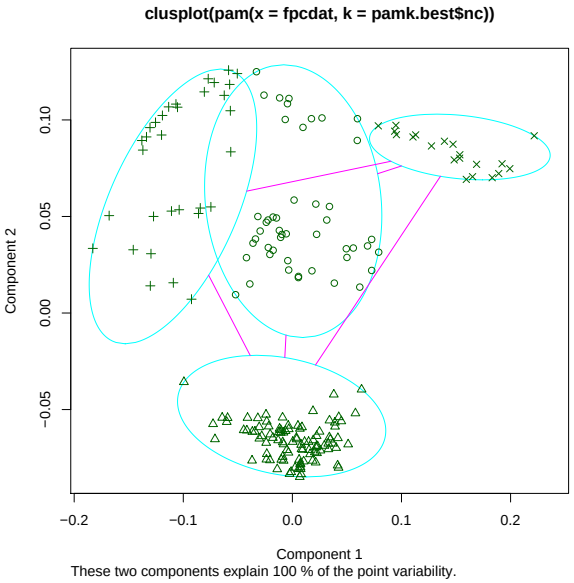
of clusters estimated by optimum average silhouette width is 5 clusters. However, when $K = 2$, then the rates of the misclassification error will be 50% for the gun-point data and 2.15% for growth data which is the smallest rate among all the other methods.

For the convex clustering (cclust) based on the hard competitive learning method, Figure 5.19 shows the hard competitive learning clustering for the FPCA scores of the two functional datasets, with misclassification rate 49.61% and 48.54% for gun-point and growth data respectively.

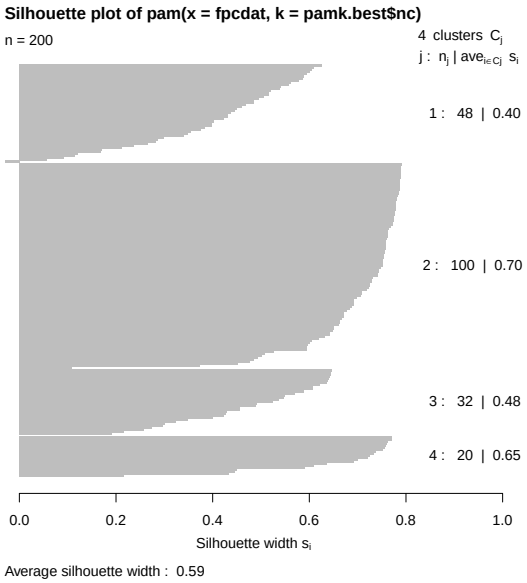
Figure 5.20 displays the banner and dendrogram plots based on DIANA algorithm for the FPCA scores of the two datasets. When $K = 2$, the rates of the misclassification error were 42.5% for the gun-point data and 10.75% for growth data, which is a better result compared to DIANA algorithm based on the discretized data.

Figure 5.21 shows the bivariate cluster plot (clusplot) and silhouette plot based on the clustering for large applications algorithm (CLARA), for the FPCA scores of gun point and growth datasets. The number of clusters estimated by Calinski- Harabasz index, which suggests 2 clusters for gun-point and 4 clusters for growth data. The rates of the misclassification error based on CLARA algorithm are 50% for the gun-point data and 2.15% for growth data, which is the same for PAM algorithm, where it is the smallest rate among all the other methods.

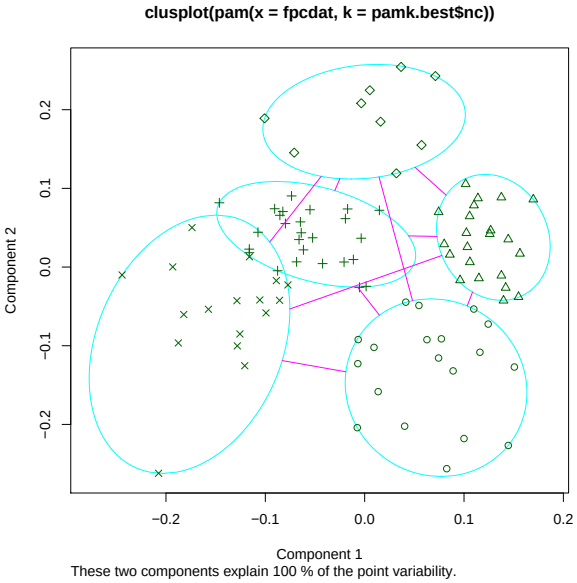
It was noted in the previous subsection that, the majority of these raw-data methods gave the same rate of misclassification error for gun-point data, which is 50%. Actually it is happened again when we considered the filtering methods based on FPCA, but the least misclassification rate, for gun-point data, this time was for the average linkage method with a minimum percentage 39%. Regarding to the growth data, we can easily see that the methods gave different rate of misclassification error and both PAM and CLARA algorithms gave the smallest rate of misclassification error (2.15%).



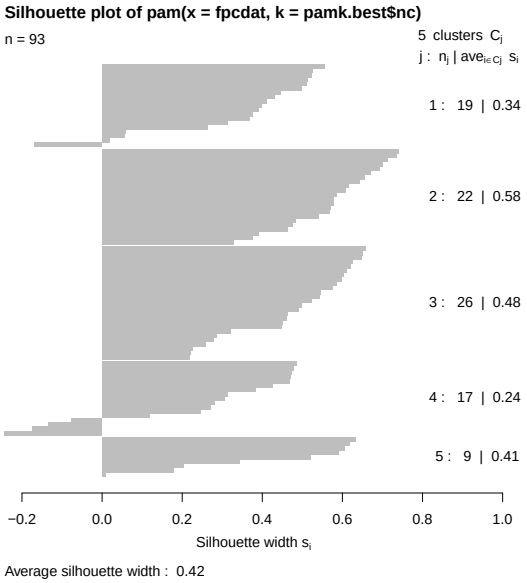
(a) Gun-point data: clusplot



(b) Gun-point data: silhouette plot

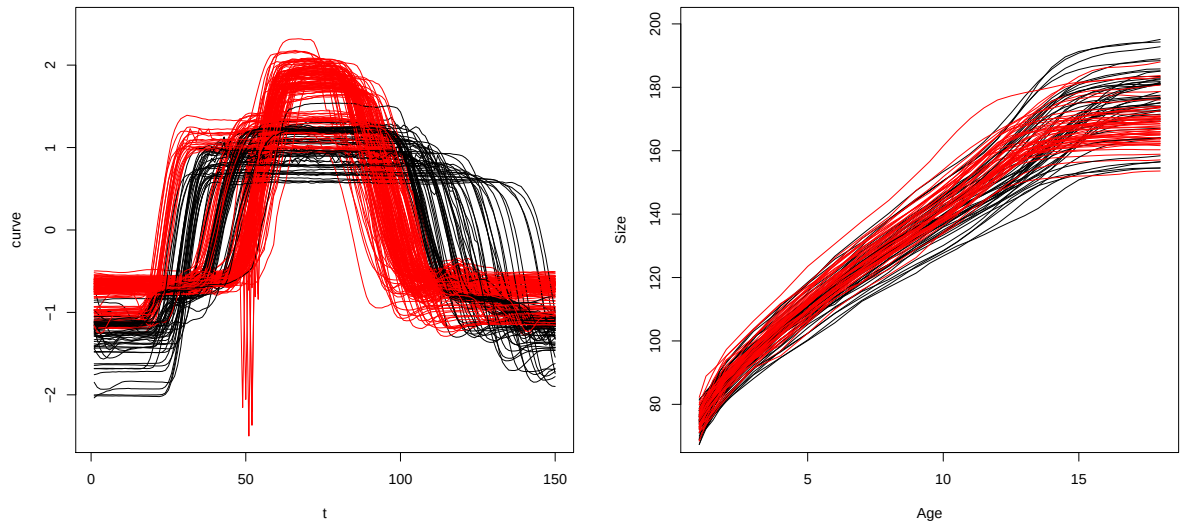


(c) Growth data: clusplot



(d) Growth data: silhouette plot

Figure 5.18: Bivariate cluster plot (clusplot) and silhouette plot based on the partitioning around medoids clustering (PAM) for the FPCA scores of: (a) gun-point data suggest 4 groups, and (b) growth data suggest 5 groups.



(a) Gun-point data

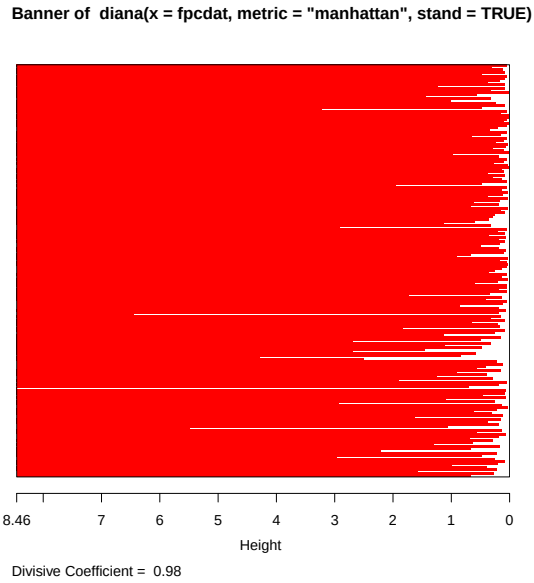
(b) Growth data

Figure 5.19: Convex clustering (cclust) based on the hard competitive learning method for the FPCA scores of: (a) gun point data and (b) growth data.

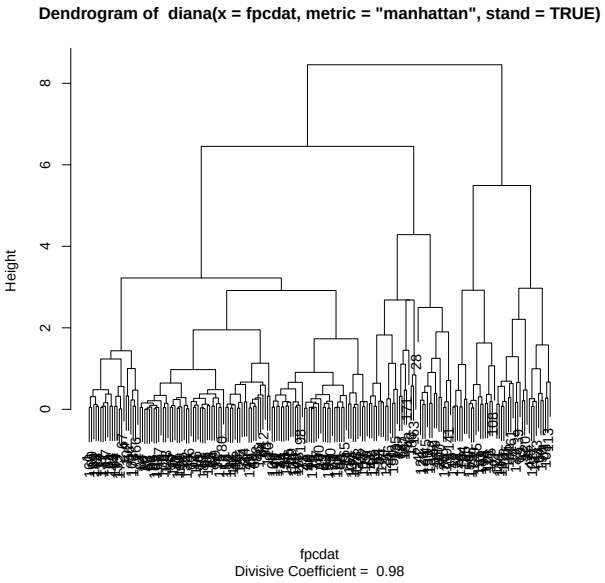
Filtering Methods on Spline Coefficients

A distinctive feature of the functional data analysis is the assumption that the observations are supposed to belong to an infinite dimensional space. Nevertheless, in reality we usually have discrete observations X_{ij} of each sample path $X_i(t)$ at a finite set of knots $\{t_{ij} : j = 1, \ldots, m_i\}$, which requires reconstructing the functional form of data from these discrete observations X_{ij}. Some literatures consider that the sample paths belong to a finite dimensional space spanned by some basis of functions like Ramsay and Silverman (2005), and others consider the non-parametric smoothing of functions like Ferraty and Vieu (2006).

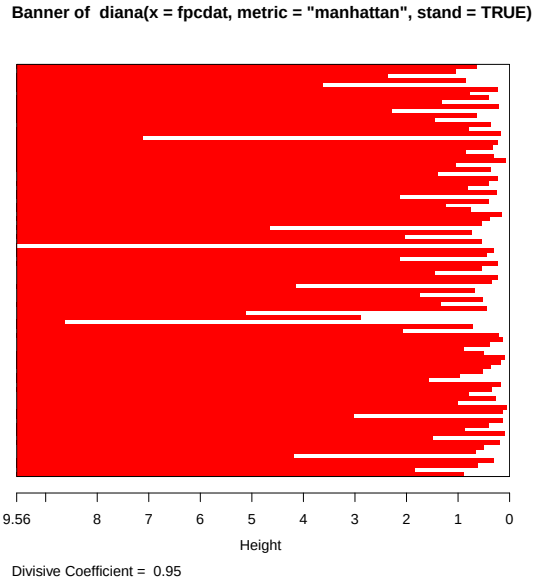
Now we review the mathematical presentation of the basis coefficients of each sample path $X_i(t)$ as shown in Jacques and Preda (2014). Suppose that the basis $\Phi = \{\phi_1, \ldots, \phi_L\}$ generating some space of functions in the Hilbert space $\mathbb{H}$ and assume that



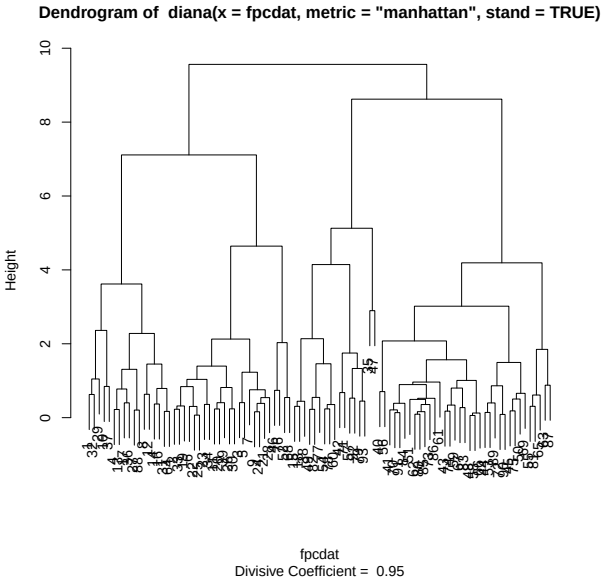
(a) Gun-point data: banner plot



(b) Gun-point data: dendrogram

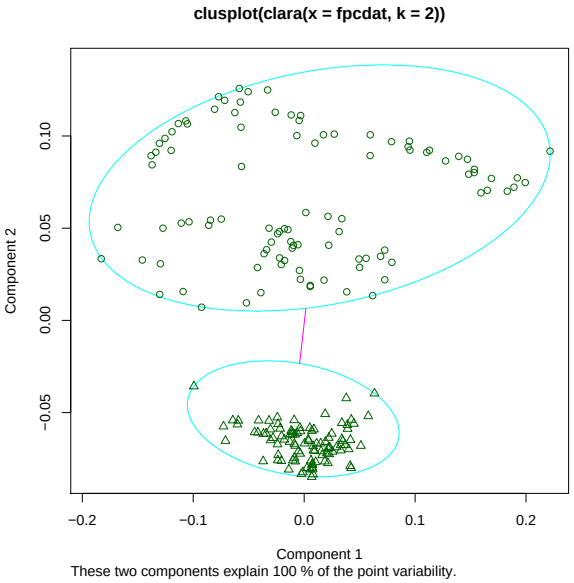


(c) Growth data: banner plot

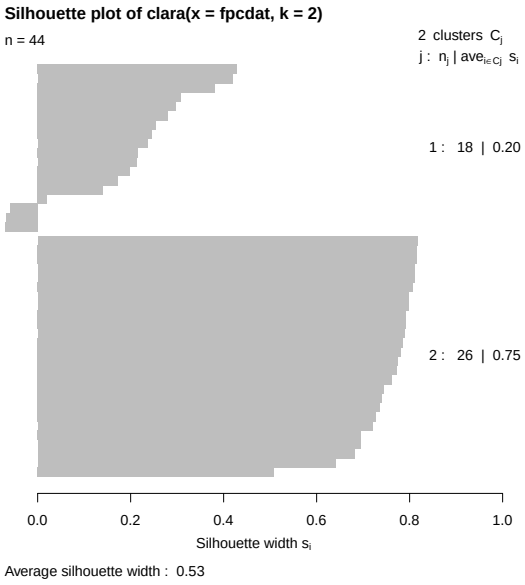


(d) Growth data: dendrogram

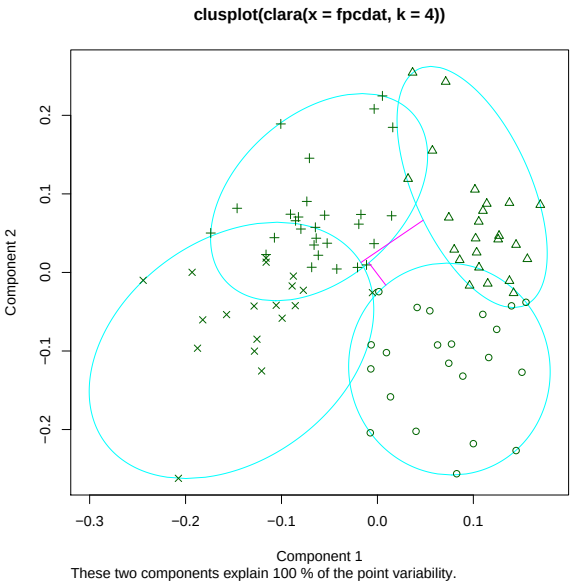
Figure 5.20: Banner and dendrogram plots of a divisive hierarchical clustering based on the DIvisive ANAlysis algorithm (DIANA) for the FPCA scores of gun-point and growth datasets.



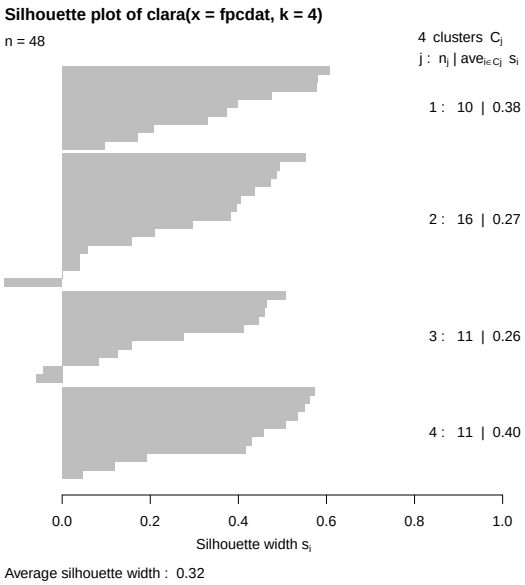
(a) Gun-point data: clusplot



(b) Gun-point data: silhouette plot



(c) Growth data: clusplot



(d) Growth data: silhouette plot

Figure 5.21: Bivariate cluster plot (clusplot) and silhouette plot based on the Clustering Large Applications (CLARA) algorithm for the FPCA scores of: (a) gun-point data suggest 2 groups and (b) growth data suggest 4 groups.

X admits the basis expansion, where $L \in N$ is the number of paths basis, and $\alpha_{il} \in \mathbb{R}$ is the coefficients of the sample paths basis, such that:

$$X_i(t) = \sum_{l=1}^{L} \alpha_{il}\phi_l(t). \tag{5.2.8}$$

In order to estimate coefficients of the sample paths basis α_{il} from the discrete observations X_{ij}, we need to use some numerical methods, such as the interpolation procedure that has been used in Escabias et al. (2005), where they proposed quasi-natural cubic spline interpolation to reconstruct annual temperatures curves from monthly values (Jacques and Preda, 2014). There are two major concerns in the existing literature regarding to the underlying model of curves; the first one is when the sample curves are observed without error, and in this case the functional predictor will be:

$$X_{ij} = X_i(t_{ij}); \quad j = 1, \ldots, m_i, \tag{5.2.9}$$

and the second one is when the sample curves are observed with error ε_{ij} such that:

$$X_{ij} = X_i(t_{ij}) + \varepsilon_{ij}; \quad j = 1, \ldots, m_i. \tag{5.2.10}$$

Ramsay and Silverman (2005) pointed out that after choosing a suitable basis, one can use the least squares smoothing. Some examples for the used basis are the trigonometric functions, B-splines or wavelets (More details regarding this topic can be found in Ramsay and Silverman, 2005). So, the basis coefficients of each sample path $X_i(t)$ are approximated by:

$$\widehat{\alpha}_i = \left(\acute{\Theta}_i\Theta_i\right)^{-1}\acute{\Theta}_i\tilde{X}_i, \tag{5.2.11}$$

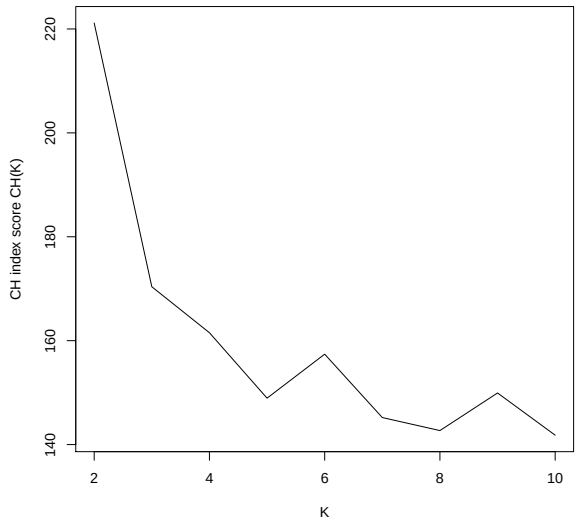
where $\widehat{\alpha}_i\big(\widehat{\alpha}_{i1}, \ldots, \widehat{\alpha}_{iL}\big)', \Theta_i = \big(\phi_l(t_{ij})\big), 1 \leqslant j \leqslant m_i, 1 \leqslant l \leqslant L$ and $\tilde{X}_i = \big(X_{i1}, \ldots, X_{im_i}\big)'$.

Some examples of the filtering methods based on the spline coefficients in literature

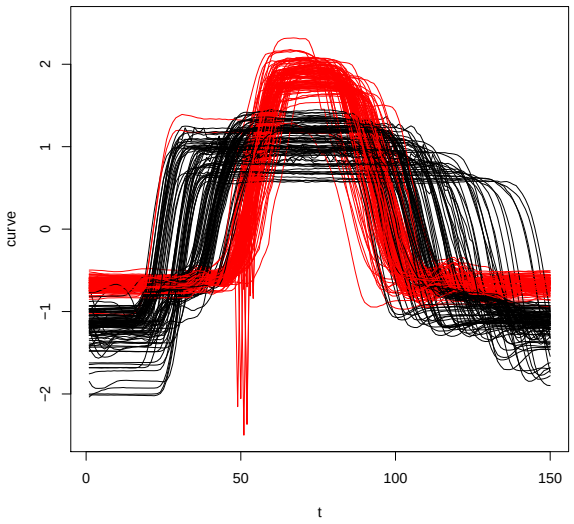
are the work that has been introduced by Abraham et al. (2003), Rossi et al. (2004) and Kayano et al. (2010), where both of them used B-splines basis. The K-means algorithm based on the B-splines coefficients is used in Abraham et al. (2003). Furthermore, Rossi et al. (2004) used an unsupervised neural network based on B-spline coefficient's basis, while Kayano et al. (2010) used Self-Organised Map based on B-spline and Gaussian coefficient's basis (Jacques and Preda, 2014). We apply now the ten methods that we used in the previous subsections, based on 10-spline coefficients on the two case studies. Figure 5.22 shows the CH index scores and k-means clustering for the spline coefficients of gun-point and growth datasets. In the two datasets, CH-index suggests 2 clusters which is the real number of groups and based on this number, the K-means assigned the different curves in 2 clusters as shown in panels (b) and (d) of the Figure. The rate of misclassification error is 50% for gun-point data, and 49.10% for the growth data. Furthermore, we can see that CH index suggested the right number of clusters in this time for growth data, where it suggested 7 and 5 clusters when the discretized points and the FPCA scores have been considered respectively. However, the K-means algorithm was not able to give a reasonable rate of the misclassification error.

The K-centroids clustering results based on the 10-spline coefficients of the two functional datasets are given in Figure 5.23. The misclassification rate in this case is 50.21% and 49.86% for gun-point and growth data respectively. Compared to Figures 5.3 and 5.13, the K-centroids based on the spline coefficients gives similar result as the discretized and FPCA scores cases for both of the two datasets.

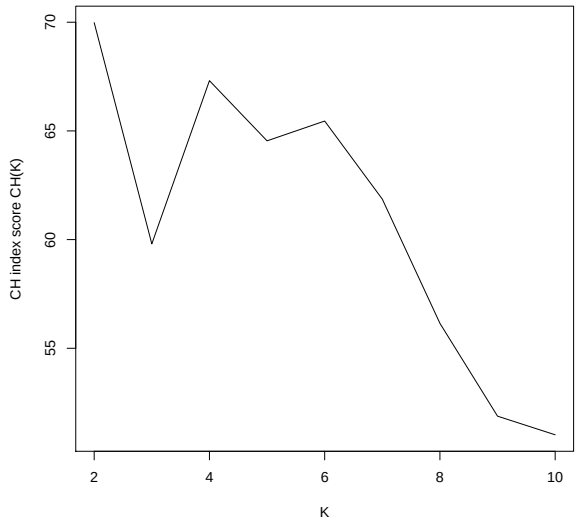
The dendrogram of the single, complete and average linkage methods (hclust) based on the 10-spline coefficients of gun point and growth datasets have been given in Figure 5.24. The complete linkage method, this time, outperforms both of the single and average linkage methods in the two datasets, where it gave smaller misclassification rates, that equal to 42.50% and 33.33% for gun-point and growth data respectively.



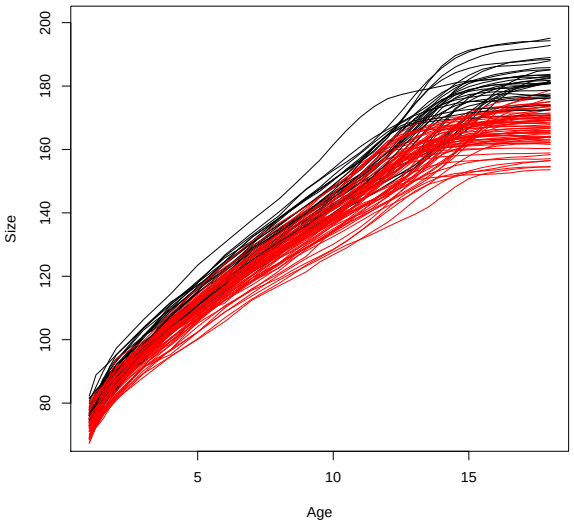
(a) Gun-point data: CH index



(b) Gun-point data: K-means

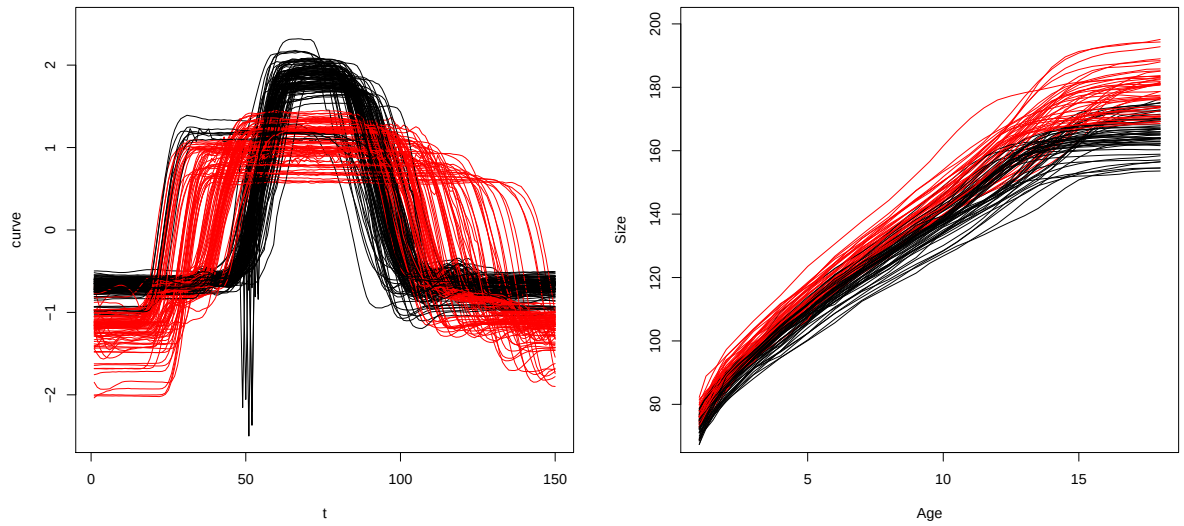


(c) Growth data: CH index



(d) Growth data: K-means

Figure 5.22: CH index scores and K-means clustering for the spline coefficients of gun-point and growth datasets. CH index suggests 2 clusters for both data.



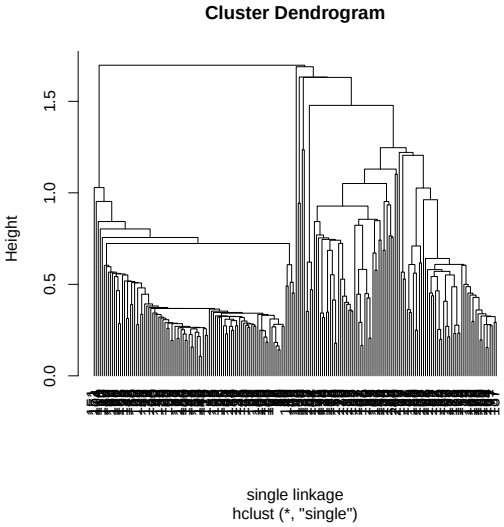
(a) Gun-point data

(b) Growth data

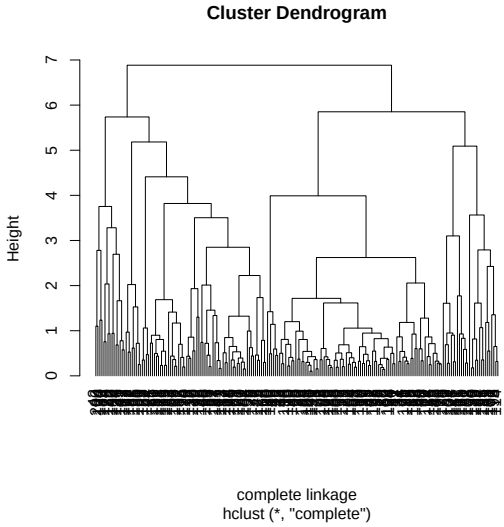
Figure 5.23: K-centroids clustering (kcca) for the spline coefficients of (a) gun-point data and (b) growth data.

Figure 5.25 shows the best model according to the BIC and ICL for mclust algorithm based on the spline coefficients of gun point and growth datasets. The optimal model for gun-point data, according to the BIC values, is an ellipsoidal, varying volume, shape, and orientation model with 7 clusters, where the maximum BIC value (BIC=4173.819), was for the VVV model, which behaves poorly and giving wrong number of clusters again. For growth data, the best model with BIC=-4729.44, was for the model EEE (elliposidal, equal volume, shape and orientation) model with 4 clusters. The misclassification rates are 50% and 34.41% for gun-point and growth data respectively. Compared to the discretized mclust, and mclust on FPCA scores, when $K = 2$, the algorithm based on the FPCA scores gives better results for the two datasets.

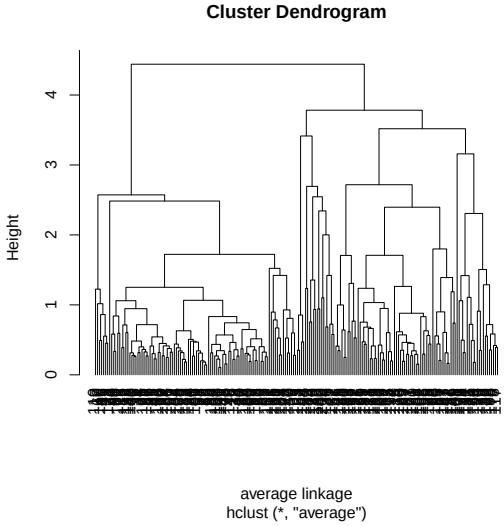
For the high dimensional data clustering (HDDC) method based on the 10-spline coefficients, if K is unknown, the selected model for gun-point data is AKJBKQKDK



(a) Gun-point data: Single linkage

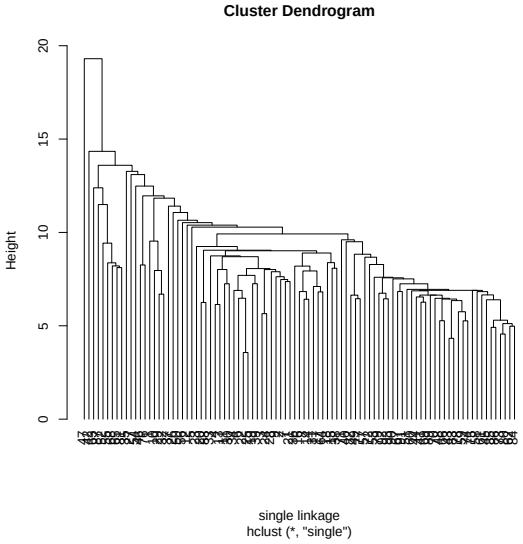


(b) Gun-point data: Complete linkage

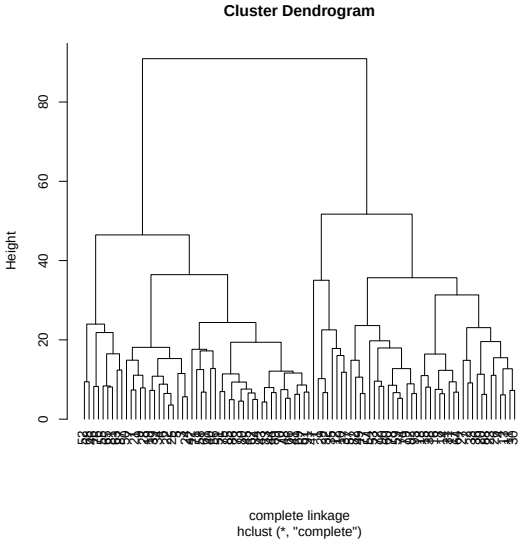


(c) Gun-point data: Average linkage

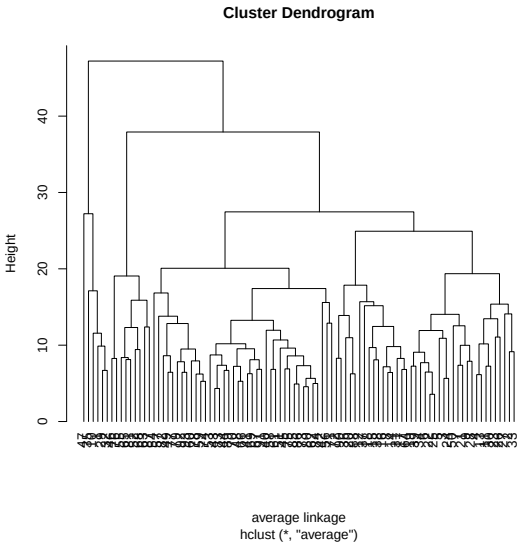
Figure 5.24: Linkage based methods for the spline coefficients of gun-point and growth datasets.



(d) Growth data: Single linkage

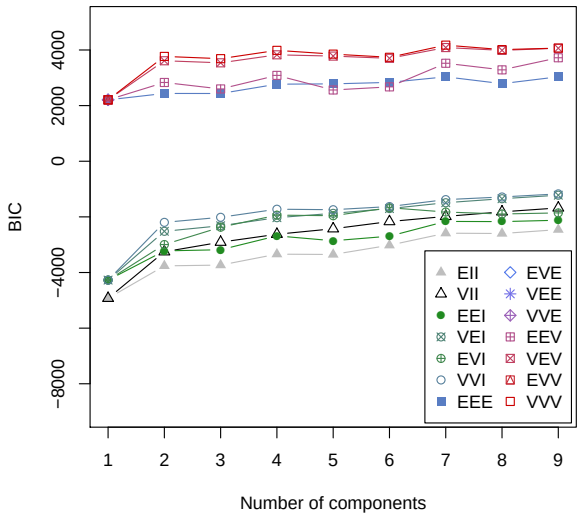


(e) Growth data: Complete linkage

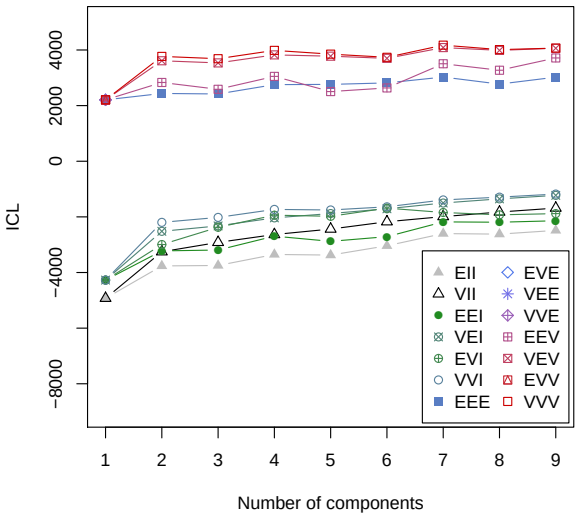


(f) Growth data: Average linkage

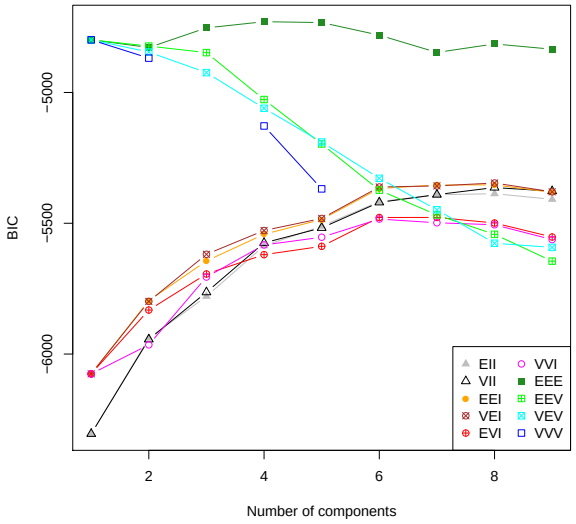
Figure 5.24: Continued.



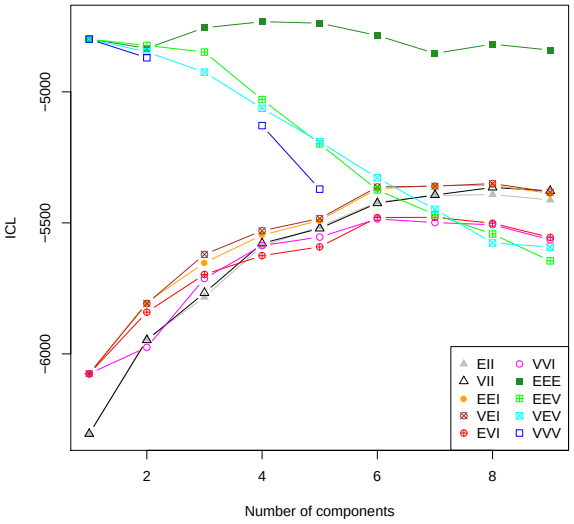
(a) Gun-point data: BIC



(b) Gun-point data: ICL



(c) Growth data: BIC



(d) Growth data: ICL

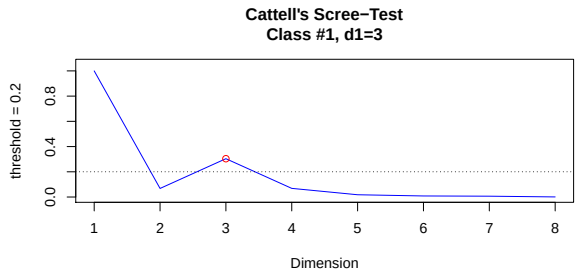
Figure 5.25: BIC and ICL plots based on Gaussian Mixture Models (GMM) for the spline coefficients of gun-point and growth datasets. Seven and three clusters are evident for gun-point and growth data respectively.

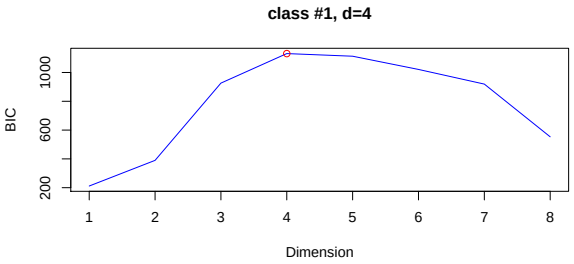
with 10 clusters and maximum BIC=-37.23889. It is the same result if we compare with HDDC based on discretized point or FPCA scores. For growth data, if K is unknown, the selected model is also AKJBKQKDK with 2 clusters and maximum BIC=-5111.048. Figure 5.26 shows the Cattell's scree-test and BIC criterion based on HDDC when $K = 2$. The rates of the misclassification error for HDDC based on the spline coefficients are 50.02% and 50.53% for the gun-point and growth data respectively.

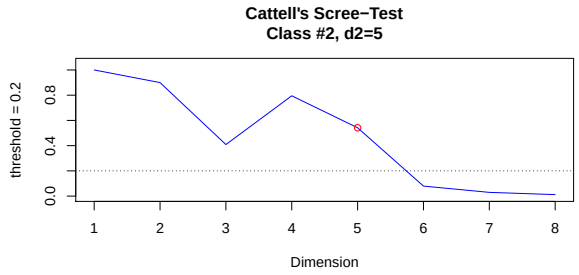
Figure 5.27 gives the best models with the number of groups and PCs using BIC based on the mixtures of probabilistic principle component analysers for the spline coefficients of gun-point and growth datasets. It can be clearly seen that, the plot of BIC suggests 4 PCs and 4 groups for gun-point data, which gives a wrong number of clusters. For the growth data, the suggested number of groups is 2, which is a right, and the suggested number of PCs is 4. The rates of the misclassification error are 53% and 80.65% for the gun-point and growth datasets respectively.

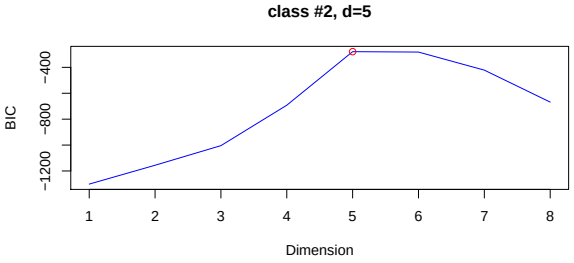
Regarding to the partitioning around medoids (PAM) method, Figure 5.28 shows the bivariate cluster plot (clusplot) and silhouette plot based on the PAM algorithm for the spline coefficients of gun point and growth datasets. For the gun-point data, if K is unknown then the number of clusters estimated by optimum average silhouette width is 3, which is a wrong number. On the other hand, if K is unknown in growth data, then the number of clusters estimated by optimum average silhouette width is the right number 2. However, when $K = 2$, then the rates of the misclassification error will be 47.50% for the gun-point data and 21.51% for growth data which is the smallest rate among all the other methods.

In addition, the convex clustering (cclust) based on the hard competitive learning method is given here. Figure 5.29 shows the hard competitive learning clustering for the spline coefficients of gun-point and growth datasets. The misclassification rates in this case are 49.99% and 49.59% for gun-point and growth data respectively. Compared to



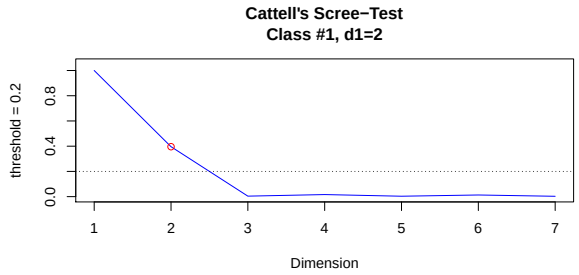


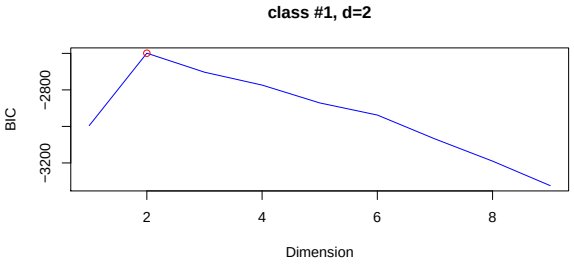


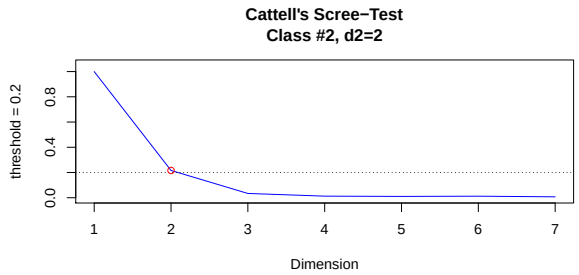


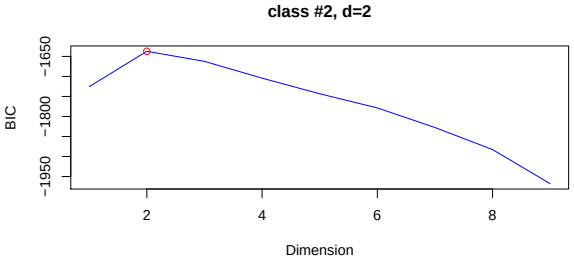
(a) Gun-point data: Cattell's test

(b) Gun-point data: BIC





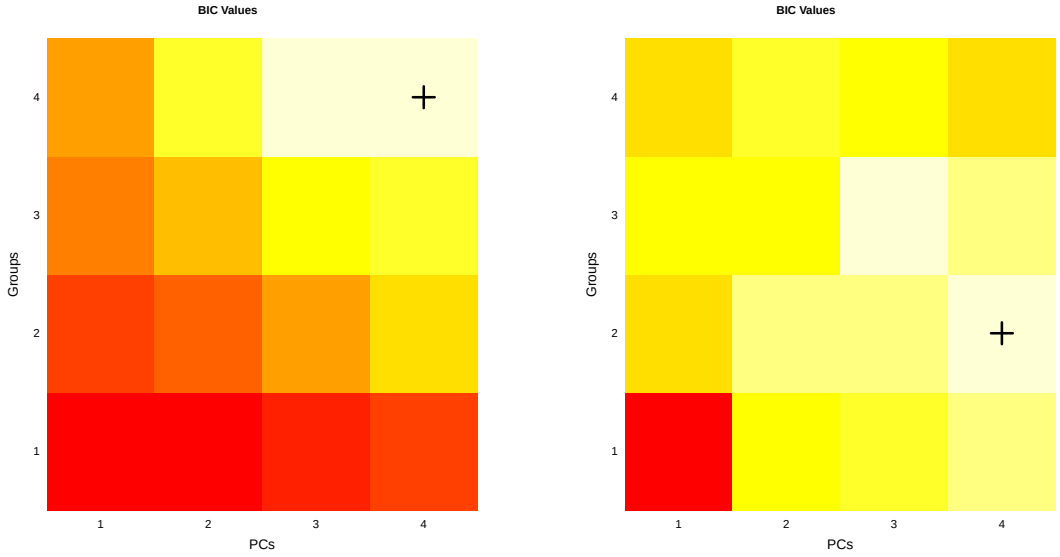




(c) Growth data: Cattell's test

(d) Growth data: BIC

Figure 5.26: Cattell's scree-test and BIC criterion based on High Dimensional Data Clustering (HDDC) for the spline coefficients of gun point and growth datasets suggest the intrinsic dimensions.



(a) Gun-point data

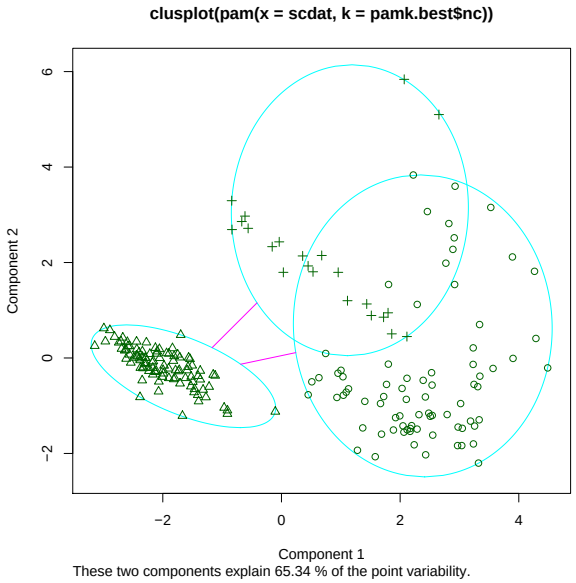
(b) Growth data

Figure 5.27: BIC plot based on the mixtures of probabilistic principle component analysers (MixtPPCA) for the spline coefficients of: (a) gun-point data suggests 4 PCs and 4 groups, and (b) growth data suggests 4 PCs and 2 groups.

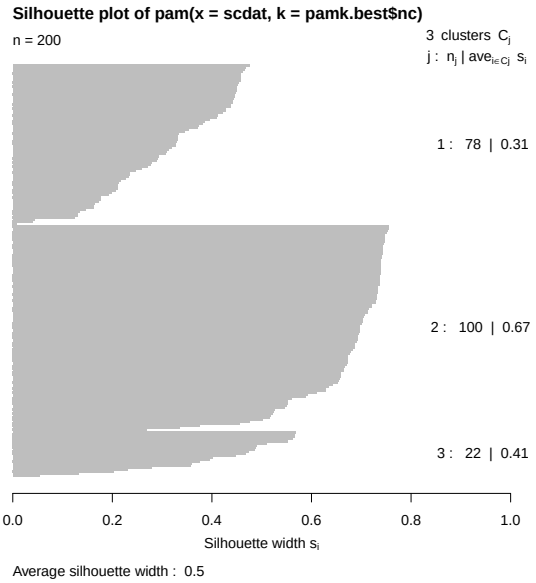
the cclust based on either discretized points or FPCA scores, when $K = 2$, the algorithm based on the FPCA scores gives better results for the two datasets.

Figure 5.30 displays the banner and dendrogram plots based on DIANA algorithm for the 10-spline coefficients of the two datasets. When $K = 2$, the rates of the misclassification error were 46% and 36.56% for gun-point and growth data respectively. Compared to the DIANA algorithm based on either discretized points or FPCA scores, the algorithm gives the best results based on the FPCA scores for growth data and based on discretized points for gun-point data.

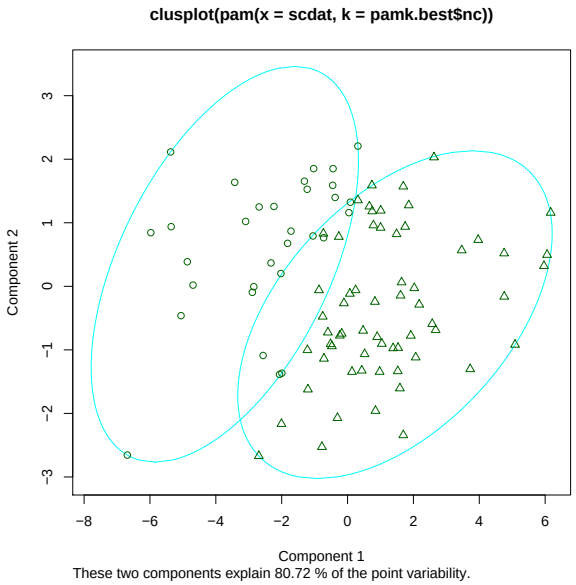
The bivariate cluster plot (clusplot) and silhouette plot based on the clustering for large applications algorithm (CLARA), for the spline coefficients of gun point and growth datasets are shown in Figure 5.31. The number of clusters estimated by Calinski- Harabasz index, which suggests 2 clusters for gun-point and 6 clusters for growth data. The rates



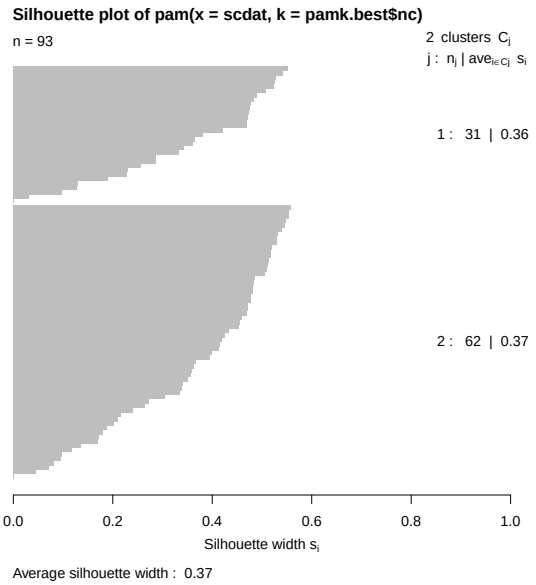
(a) Gun-point data: clusplot



(b) Gun-point data: silhouette plot

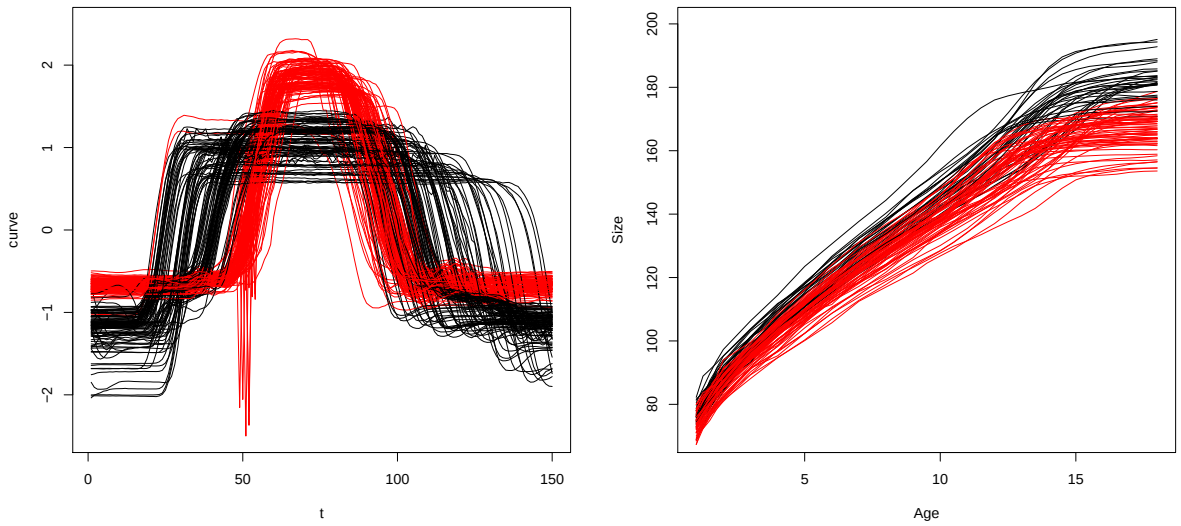


(c) Growth data: clusplot



(d) Growth data: silhouette plot

Figure 5.28: Bivariate cluster plot (clusplot) and silhouette plot based on the partitioning around medoids clustering (PAM) for the spline coefficients of: (a) gun-point data suggest 3 groups, and (b) growth data suggest 2 groups.



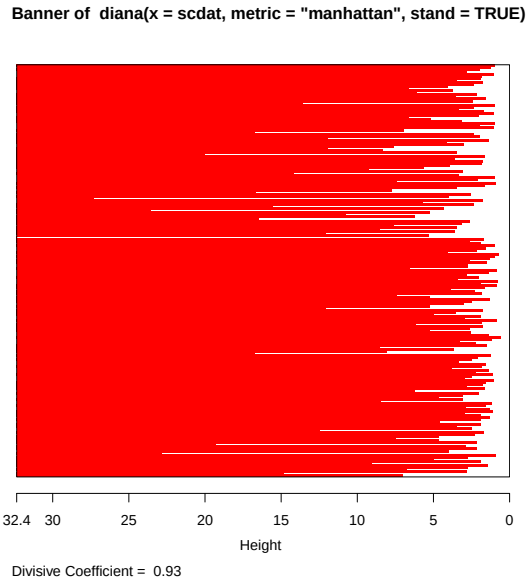
(a) Gun-point data

(b) Growth data

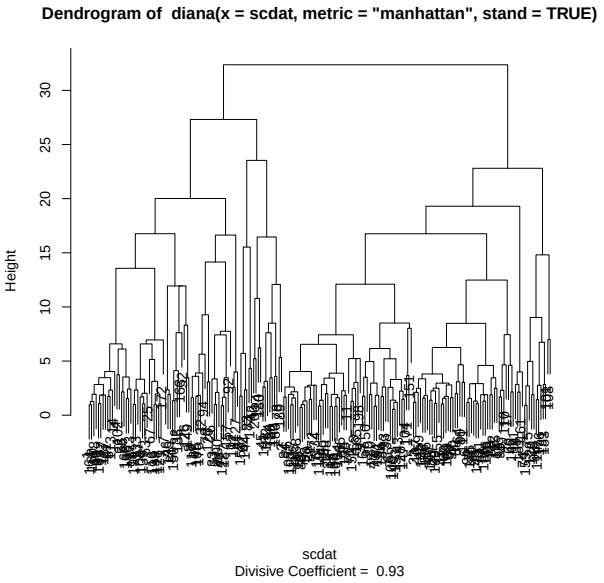
Figure 5.29: Convex clustering (cclust) based on the hard competitive learning method for the spline coefficients of: (a) gun point data and (b) growth data.

of the misclassification error based on CLARA algorithm are 47.5% for the gun-point data and 21.51% for growth data. Compared to the CLARA algorithm based on either discretized points or FPCA scores, the algorithm gives the best results based on the spline coefficients for gun-point data and based on FPCA scores for growth data.

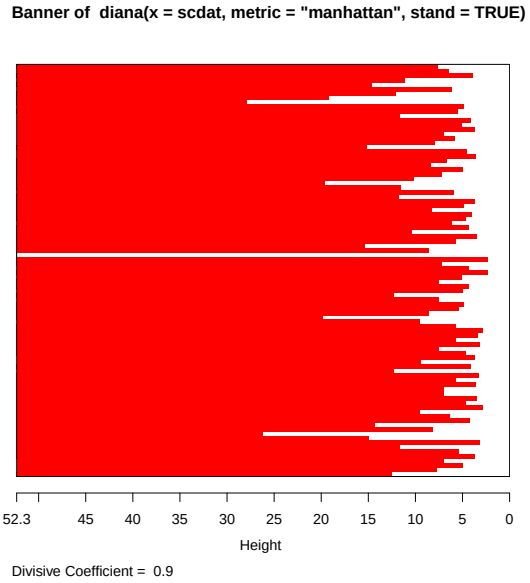
Like the raw-data methods and filtering methods based on FPCA scores, many of the filtering methods based on spline coefficients gave the same rate of misclassification error for gun-point data, which is 50%. In addition, the smallest misclassification rate for gun-point data among all the filtering methods based on spline coefficients was for the complete linkage method with a minimum percentage 42.5%. On the other hand, in growth data, we can see that the methods gave different rate of misclassification error and again both PAM and CLARA algorithms gave the smallest rate of misclassification error (21.51%).



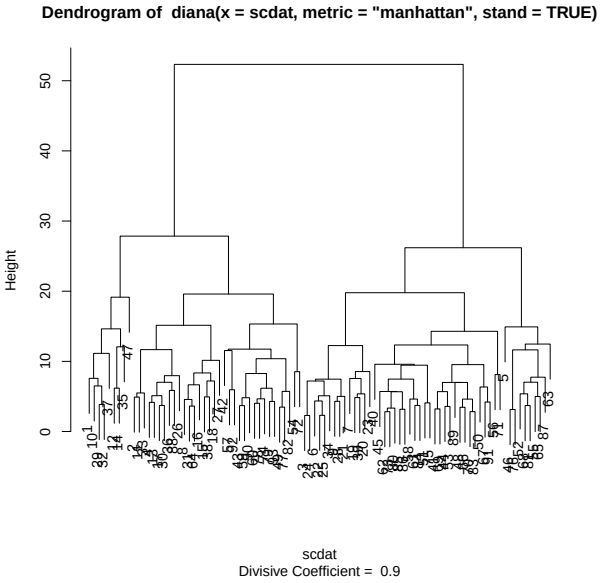
(a) Gun-point data: banner plot



(b) Gun-point data: dendrogram

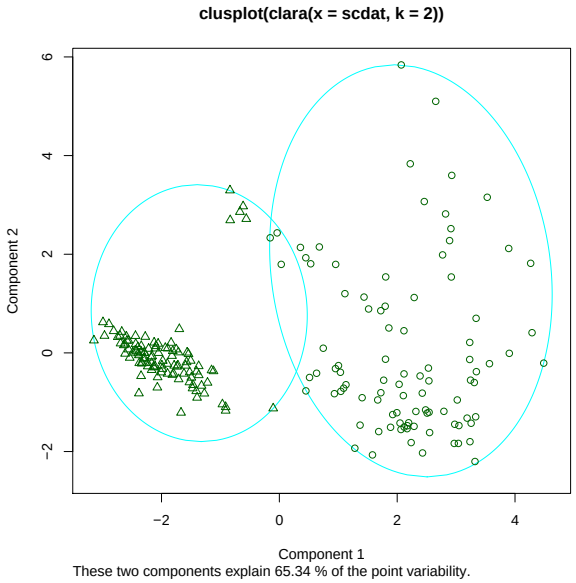


(c) Growth data: banner plot

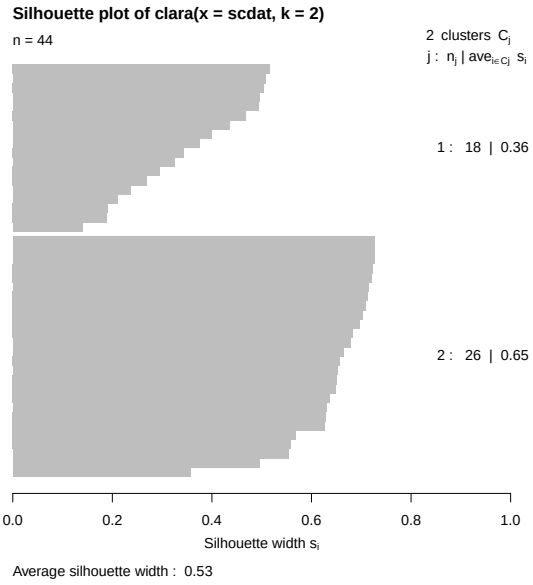


(d) Growth data: dendrogram

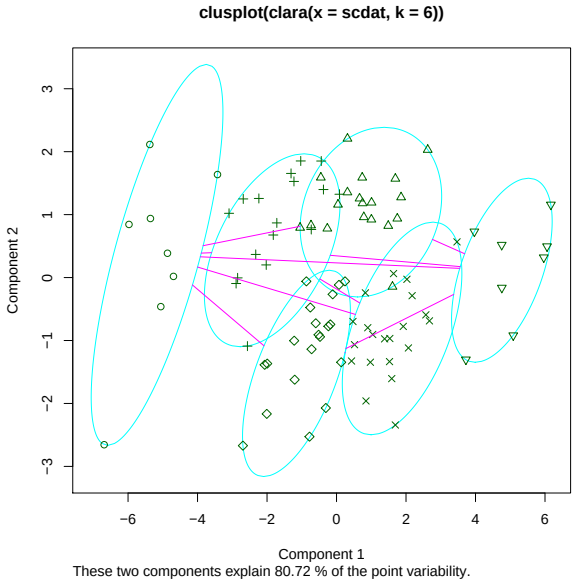
Figure 5.30: Banner and dendrogram plots of a divisive hierarchical clustering based on the DIvisive ANAlysis algorithm (DIANA) for the spline coefficients of gun-point and growth datasets.



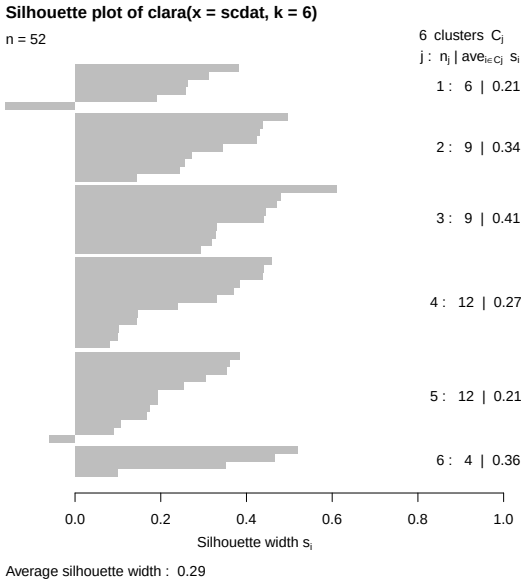
(a) Gun-point data: clusplot



(b) Gun-point data: silhouette plot



(c) Growth data: clusplot



(d) Growth data: silhouette plot

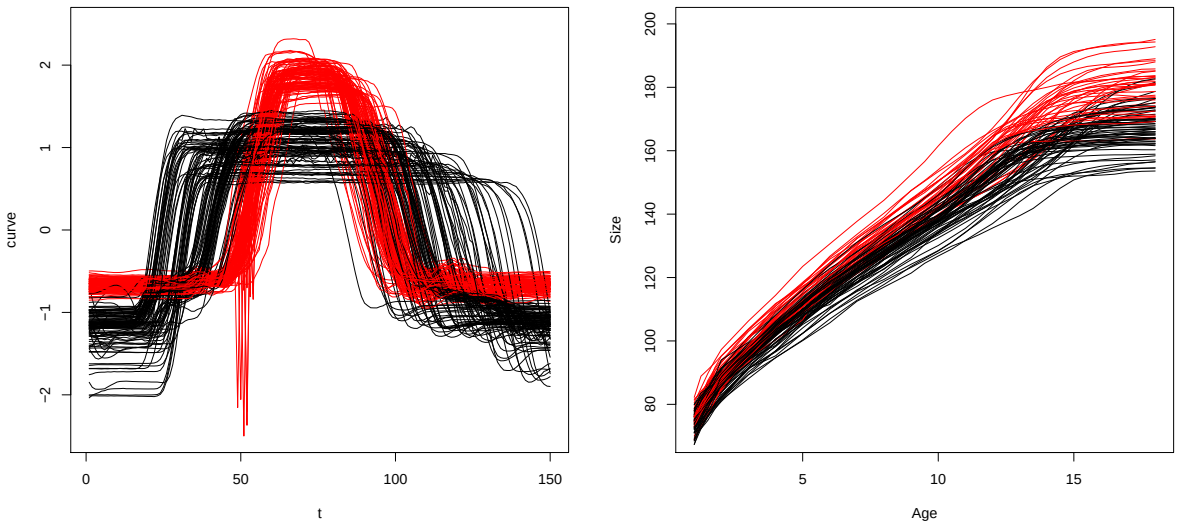
Figure 5.31: Bivariate cluster plot (clusplot) and silhouette plot based on the Clustering Large Applications (CLARA) algorithm for the spline coefficients of gun-point data suggest 2 groups and growth data suggest 6 groups.

Three filtering methods based on the spline coefficients gave the smallest rate of misclassification error for the gun-point data among the raw-data methods and filtering methods based on FPCA scores, these methods are Kmeans, PAM and CLARA. Furthermore, the complete linkage method based on the spline coefficients of growth data was the only filtering method based on spline coefficients that gave smallest rate of the misclassification error among the other raw-data methods and filtering methods based on FPCA scores. This in fact sheds the light on the filtering methods based on FPCA scores, where they gave the majority of the smallest misclassification rates amongst the raw-data methods and filtering methods based on spline coefficients.

5.2.3 Adaptive Methods

The adaptive methods group consists of methods giving a functional representation of the data depending on clusters, and performing a dimension reduction and clustering simultaneously. The main idea behind the adaptive methods is that, the basis expansion coefficients and the FPCA scores are considered to be random variables instead of considering them as parameters like the filtering methods. Furthermore, it is assumed in the adaptive methods that these random variables have a cluster-specific probability distribution (Jacques and Preda, 2014). In this group of methods, two major directions are usually considered. In the first one the methods find the probabilistic model of the basis expansion coefficients and then perform a dimension reduction and clustering simultaneously. Examples of the adaptive methods based on the probabilistic model of the basis expansion coefficients are the works by James and Sugar (2003), Heard et al. (2006), Ray and Mallick (2006), Samé et al. (2011) and Giacofci et al. (2012).

James and Sugar (2003) introduced a clustering approach for sparsely sampled functional data by giving a cluster-specific probability distribution to the basis expansion coefficients of the curves, which is a mixture Gaussian distribution with different means



(a) Gun-point data

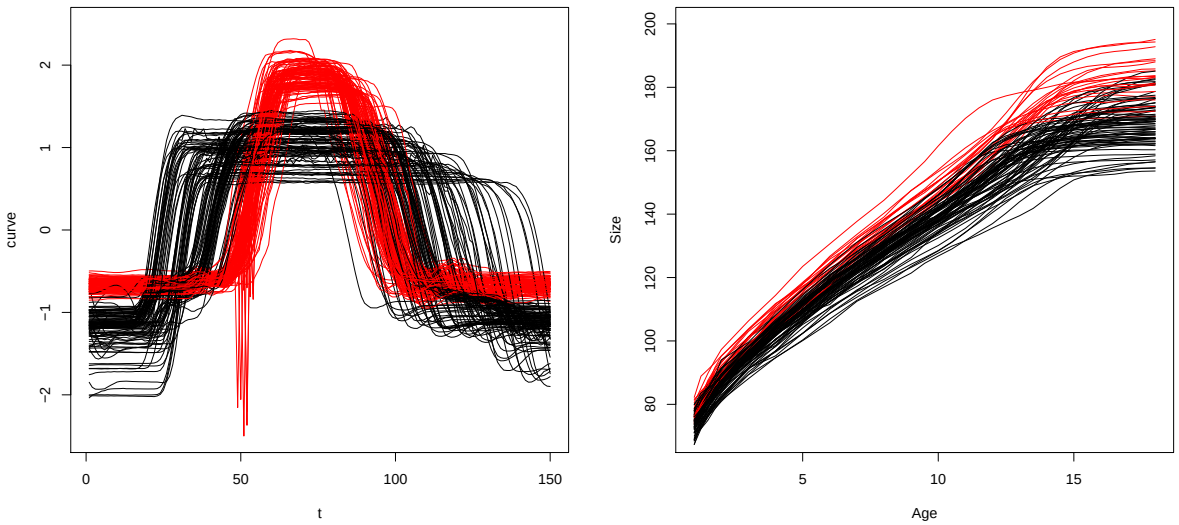
(b) Growth data

Figure 5.32: Clustering of gun-point and growth datasets using fclust method.

to each cluster and common variance, such that $\alpha_i \sim N(\mu_k, \Sigma)$. They supposed that the basis expansion coefficients are random variables, and defined some parsimonious clustering models based on the parameters in the assumed mixture Gaussian distribution. Their method is known in the literature as fclust method. Figure 5.32 shows the clustering of gun point and growth datasets using fclust method. It is required to specify the number of clusters in fclust algorithm. Supposing $K = 2$, then the misclassification rates based on the fclust method are 50% and 53.58% for gun-point and growth data respectively.

Heard et al. (2006) proposed Bayesian models to the basis expansion coefficients of the curves, and they supposed that the basis expansion coefficients are distributed normally with common mean μ and different variances $\sigma_k\Sigma$ to each cluster such that $\alpha_i \mid \sigma_k \sim N(\mu, \sigma_k\Sigma)$ and $\sigma_k \sim \mathcal{IG}(u, \nu)$ where $\mathcal{IG}$ is the Inverse-Gamma distribution with parameters u and ν.

In addition, in Ray and Mallick (2006), a hierarchical Bayesian model also been pro-



(a) Gun-point data

(b) Growth data

Figure 5.33: Clustering of gun-point and growth datasets using the wavelets-based method (curvclust algorithm)

posed for the basis expansion coefficients of the functional curves supposing that Σ is modelled by two sets of random variables controlling the sparsity of the wavelets decomposition and a scale effect (Jacques and Preda, 2014).

A Gaussian model on wavelet-based clustering for mixed-effects functional models in high dimension have been introduced by Giacofci et al. (2012). Their algorithm is known as `curvclust` in the **R** software. The function `getFCM` in the package `curvclust` performs the wavelet-based clustering. Figure 5.33 shows the clustering of gun point and growth datasets using curvclust. The number of clusters is assumed to be two in the curvclust algorithm, and the misclassification rates based on the curvclust are 50% and 66.18% for gun-point and growth data respectively.

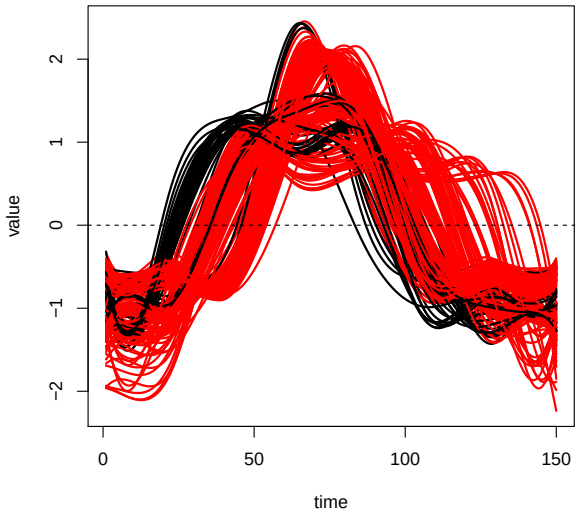
On the other hand, the adaptive methods, in the second direction, are depending on the probabilistic model of the FPCA scores then perform the dimension reduction and clus-

tering simultaneously. For instant, Chiou and Li (2007) proposed the K-centres algorithm which is K-means based on the L_2 distance between truncations of the Karhunen-Loeve expansions. Moreover, Delaige and Hall (2010) introduced a probabilistic model of the FPCA scores by using the density of principal components resulting from a FPCA of the curves, after getting an approximation of the probability density for functional random variables based on the Karhunen-Loeve expansion as well.

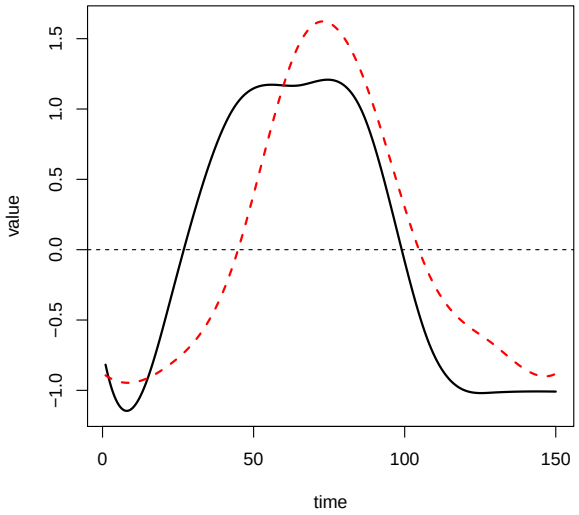
The functional high dimensional data clustering (FunHDDC) method has been introduced by Bouveyron and Jacques (2011). It is an extension of the high dimensional data clustering method (HDDC) that introduced earlier by Bouveyron, Girard, and Schmid (2007). Like the HDDC, the FunHDDC is model-based clustering method based on the Gaussian mixture model, with the difference that it deals with a functional version of data using some basis functions that used in FPCA approximation with considering a family of parsimonious sub-models based on the parsimony assumptions on the variance, which allows to cluster functional data by modeling each group within a specific functional subspace.

Figure 5.34 gives the functional curves and means based on FunHDDC coefficients for gun point and growth datasets. It is required to specify the number of clusters in this algorithm as well. The BIC value based on the FunHDDC suggests 3 clusters with maximum value of -47931.85 for gun-point data and -6827.67 for growth data. When $K = 2$, then the misclassification rates based on the FunHDDC method are 57.5% and 33.33% for gun-point and growth data respectively.

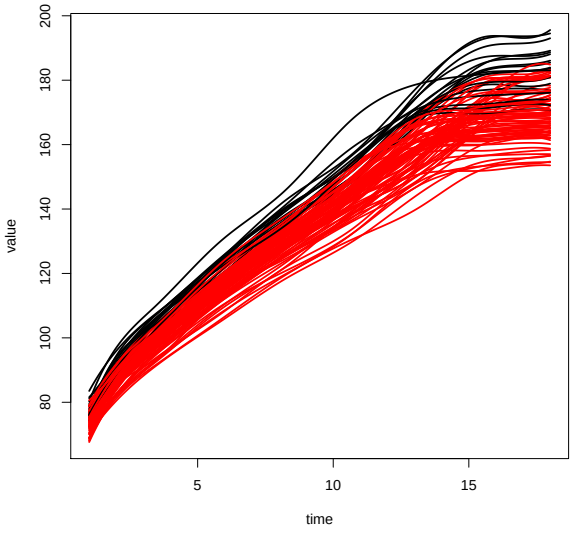
Along the line of Bouveyron and Jacques (2011), Jacques and Preda (2013) introduced the funclust algorithm which is using a similar algorithm based on the Gaussian distribution of the principal components and define a probabilistic model based clustering techniques. Like the other model-based clustering methods, the funclust algorithm uses the expectation-maximization algorithm to estimate the parameters of the model. Figure



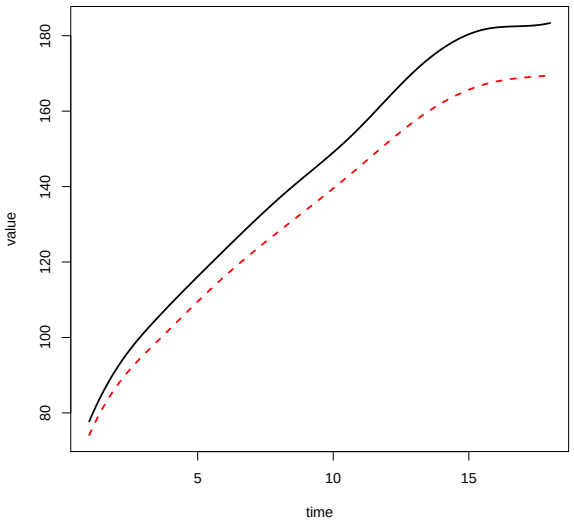
(a) Gun-point data: fd curves



(b) Gun-point data: fd means

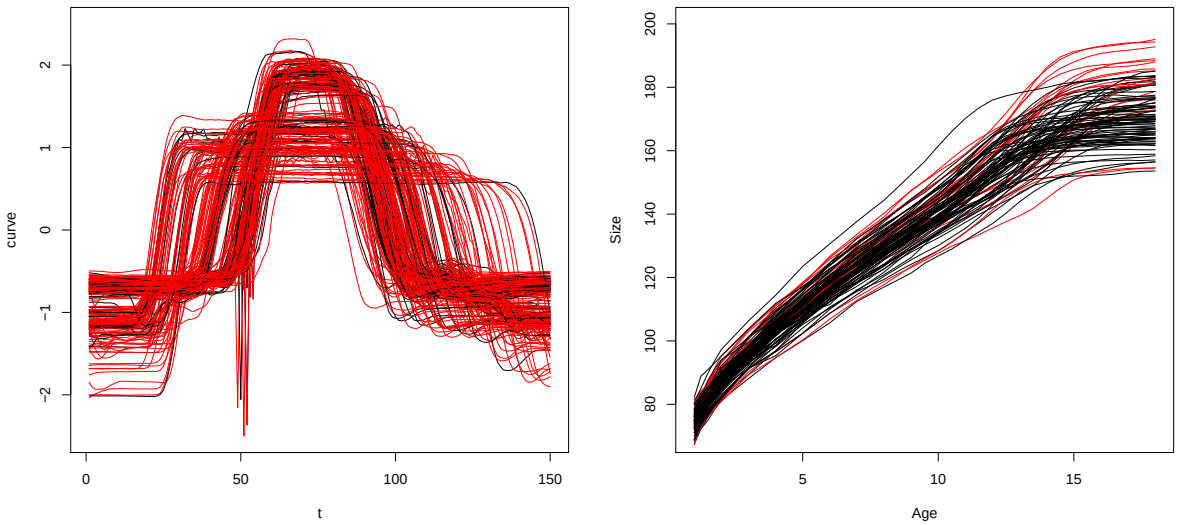


(c) Growth data: fd curves



(d) Growth data: fd means

Figure 5.34: FunHDDC clustering: Plots of functional data curves and functional data means based on FunHDDC coefficients for gun-point and growth datasets



(a) Gun-point data

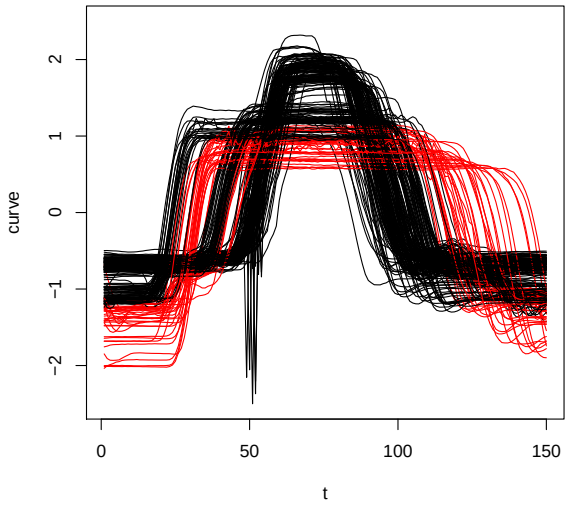
(b) Growth data

Figure 5.35: Clustering of gun point and growth datasets using funclust algorithm

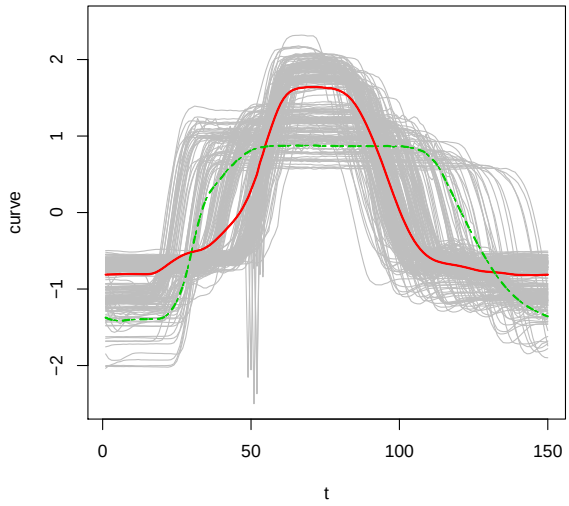
5.35 shows the clustering of gun point and growth datasets using funclust algorithm by Jacques and Preda (2013). As it is required to identify the number of clusters, we assumed $K = 2$. The misclassification rates based on the funclust method are 50.1% and 49.22% for gun-point and growth data respectively.

The `kmeans.fd` function in the `fda.usc` package, which performs the K-means clustering on functional data, has been considered as one of the adaptive methods. It depends on the algorithm that introduced earlier by Hartigan and Wong (1979). In Figure 5.36, the clustering and the updated centers based on `kmeans.fd` for gun-point and growth data are given. The green curves have been assigned to be in the first cluster and the red ones have been assigned to be in the second. The misclassification rates based on the kmeans.fd clustering are 49.63% and 49.89% for gun-point and growth data respectively.

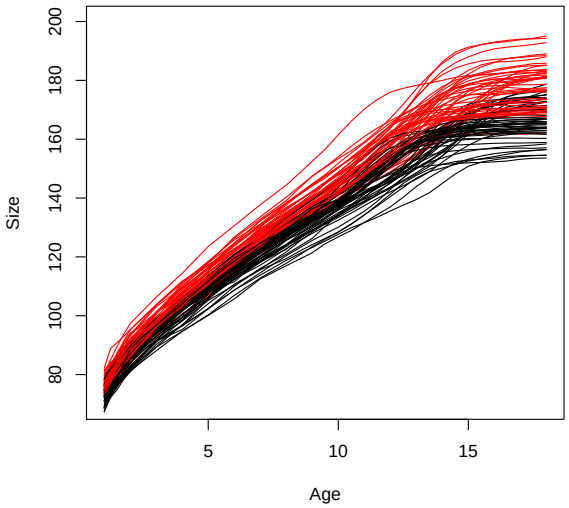
Acorrding to the results of the misclassification rates that have been given in the previous comparison, the kmeans.fd clustering algorithm gives the smallest rate of mis-



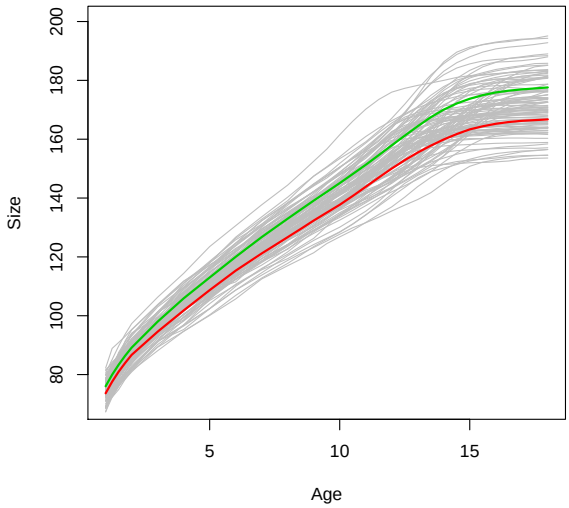
(a) Gun-point data: curves plot



(b) Gun-point data: updated centers



(c) Growth data: curves plot



(d) Growth data: updated centers

Figure 5.36: K-means clustering for functional data (kmeans.fd algorithm): plot of the curves and the updated centers based on kmeans.fd for the gun-point and growth datasets.

classification erorr (49.63%) for gun-point data amongst the other four adaptive methods. On the other hand, for the growth data, the FunHDDC method gives the smallest rate of misclassification erorr (33.33%) among the other four adaptive methods.

5.2.4 Distance-Based Methods

The last group of the functional data clustering methods is the distance-based methods that use some clustering algorithms based on some dissimilarity measures and distance functions for the functional data. Different distance-based methods have been extended to work for the functional data clustering, like K-means, K-medoids and linkage-based methods. According to Jacques and Preda (2014), the general form of the distances measures between two curves $X_i(t)$ and $X_j(t)$ is:

$$d_l\big(X_i(t), X_j(t)\big) = \bigg(\int_{\mathcal{T}} \Big(X_i^{(l)}(t) - X_j^{(l)}(t) \Big)^2 dt \bigg)^{1/2}, \tag{5.2.12}$$

where $X_i^{(l)}(t)$ is the l-th derivative of $X_i(t)$. Hence, the distance d_0, which is the L_2-metric takes the formula:

$$d_0\big(X_i(t), X_j(t)\big) = \bigg(\int_{\mathcal{T}} \Big(X_i(t) - X_j(t) \Big)^2 dt \bigg)^{1/2}, \tag{5.2.13}$$

and the distance d_1 is:

$$d_1\big(X_i(t), X_j(t)\big) = \bigg(\int_{\mathcal{T}} \Big(X_i^{(1)}(t) - X_j^{(1)}(t) \Big)^2 dt \bigg)^{1/2}, \tag{5.2.14}$$

where $X_i^{(1)}(t)$ is the first derivative of $X_i(t)$.

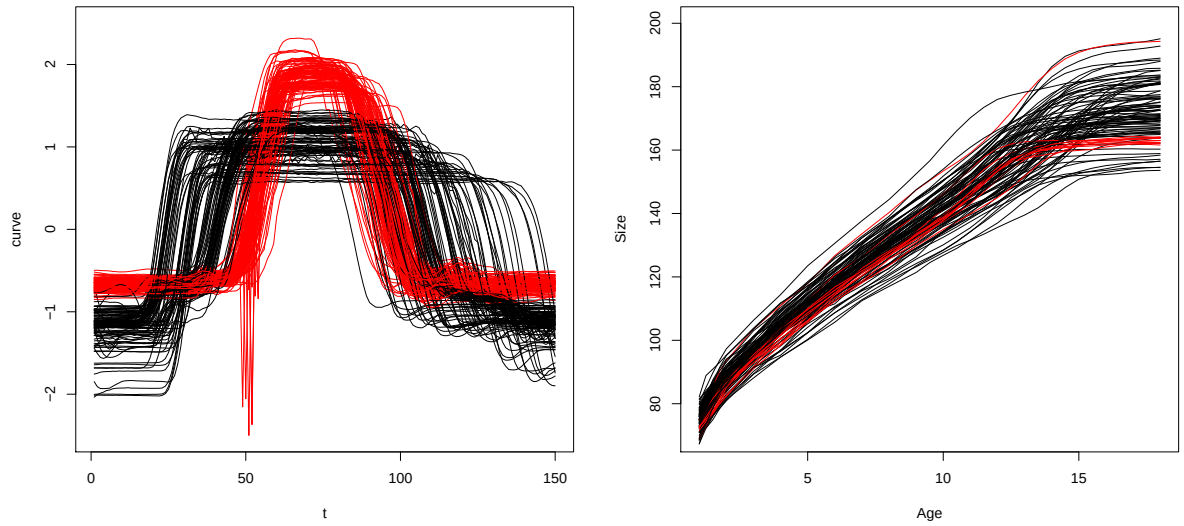
Examples of the distance-based methods using the distance d_0 for functional data are the work by Cuesta-Albertos and Fraiman (2000), where they used the K-means algorithm based on the distance d_0; the work by Tarpey and Kinateder (2003), who used the K-

means based on d_0 for some Gaussian processes; and the work by Tokushige et al. (2007) who used the distance d_0 with the K-means algorithm as a time-dependent clustering. Combinations between using the distance d_0 and d_2 have been considered in Ferraty and Vieu (2006) where they proposed a hierarchical clustering algorithm combined with the distances d_0 and d_2. On the other hand, Ieva et al. (2012) used the K-means algorithm with the distances d_0, d_1 and the combinations of d_0 and d_1 such as, $\left(d_0^2 + d_1^2\right)^{(1/2)}$.

We have used the function `eval.posfd` of the `fda` package under the **R** software in order to get the evaluations of the set of original functions first derivatives. In addition, the function `kma` in the `fdakma` package has been used to perform K-means clustering based on the distances d_0 and d_1, and alignment of a functional dataset. Figure 5.37 shows the clustering of gun point and growth datasets using the K-means algorithm based on the distance d_0. Assuming that $K = 2$, the misclassification rates based on the K-means based on the distance d_0 are 50% and 48.78% for gun-point and growth data respectively. On the other hand, Figure 5.38 shows the clustering of gun point and growth datasets using the K-means algorithm based on the distance d_1. The misclassification rates in this case are 49.1% and 46.74% for gun-point and growth data respectively. Clearly, we can see that the K-means algorithm based on the distance d_1 behaves better than the algorithm based on the distance d_0 for the two functional datasets.

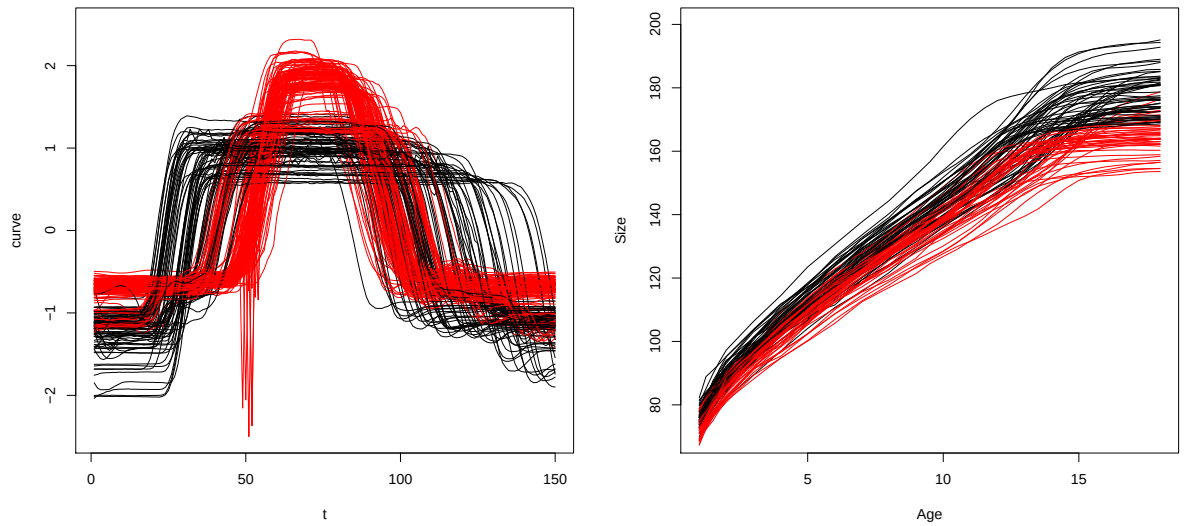
5.3 The Curse of Dimensionality in the Traditional Forward Search

The term "curse of dimensionality" was introduced by R. Bellman in his book (Bellman, 1957). It refers to all problems caused by the analysis of high-dimensional data, especially if some parametric models with parameters estimation are considered. As we discussed earlier, in order to run the traditional forward search algorithm based on Mahalanobis distances, we need to choose an initial subset $S(m)$, with $m = d + 1$. Practically, there



(a) Gun-point data (b) Growth data

Figure 5.37: Clustering of the gun-point and growth datasets using K-means based on the distance d_0 (Kmeans-d_0).



(a) Gun-point data (b) Growth data

Figure 5.38: Clustering of the gun-point and growth datasets using K-means based on the distance d_1 (Kmeans-d_1).

is no problem if the number of available observations is large compared to the number of variables. However, the traditional forward search algorithm will not be able to estimate the number of clusters efficiently when the dimension d is large.

On the contrary, if the number of observations is small compared to the number of variables, such a situation increases the problem difficulty, and the algorithm would be inapplicable. This is due to that, if the dimension d is bigger than the size of subset $S(m)$, we will not be able to estimate the variance-covariance matrix, and then we cannot proceed in the algorithm. That makes the using of the traditional forward search based on Mahalanobis distances not applicable for the functional data analysis, where the random variables taking values into an infinite dimensional space, whereas in practice we only have some sampled curves observed into a finite set of points.

This leads to the fact that the traditional forward search based on Mahalanobis distances can be only used for the multivariate samples, with finite dimensional setting less than the underlying sample size. In addition, if the number of variables is very big and there are some clusters with small size in the underlying data at the same time, then using the traditional forward search based on Mahalanobis distances will lead to loss information about the number of clusters to be determined. This in fact is due to that the search starts from subsets with size $d+1$ and if the size of the j-th cluster k_j is less than d, then the algorithm will not be able to detect the cluster. This in fact sheds the light on the curse of dimensionality in the traditional forward search, and makes us ask the question: what is the case if we use a forward search algorithm based on a nonparametric function which does not need to any parameters estimation?

To answer this question, we need to remind the reader that, our forward search algorithm based on the spatial ranks, can be started with subsets with any size, which makes it helpful in such situations that explained above. According to Baragilly and Chakraborty (2015), they pointed out that in the forward search based on spatial ranks, the initial sub-

set size can be anything more than 1 as the rank of any observation $\mathbf{x} \in \mathbb{R}^d$ with respect to a single data point is always 1. So, we can start with very small subset size which makes the algorithm robust in the case that the size of the j-th cluster k_j is less than the dimension d. On the other hand, if we consider the forward search algorithm based on the volume of central rank regions, we will not have the same amount of flexibility in starting the search with small subsets comparing to the spatial ranks algorithm. The reason behind this is that, computing volumes of central rank regions, which provides a measure of scale, is meaningful only when the number of observations is at least $d + 1$. Thus, purely for more stability in the algorithm, we choose an initial subset size of $d + 1$. That leads to, if there are large number of clusters and all are with sizes smaller than $d + 1$, then our algorithm will not be able to estimate the number of clusters efficiently, but that is a rarity for large sample size n (Baragilly and Chakraborty, 2016). For these reasons, we are not going to extend the forward search algorithm based on the volume of central rank regions to the functional data case, and we will limit the research to the forward search algorithm based on the functional spatial ranks.

5.4 Functional Data Clustering Based on Spatial Ranks

A functional forward search algorithm is considered in this section. It is based on non-parametric functional spatial ranks (FSR) and it is robust in terms of determining the number of clusters by the data itself without any need to parameters estimation or any filtering to be done a priori. That makes our method to be classified under the raw-data methods' group. The contribution of this work is to propose a forward search algorithm that can work with the functional data case while the traditional forward search cannot be extended to that. A key point for the FSR is to extend both $sign(\mathbf{x})$ and $Rank(\mathbf{x})$ naturally from $\mathbb{R}^d$ to any infinite-dimensional Hilbert space $\mathbb{H}$. We start with defining the functional sign and spatial rank functions with some related literature, the functional for-

ward search algorithm based on functional spatial ranks, and then we give some numerical examples in order to check the performance of the proposed method.

5.4.1 The Functional Spatial Rank (FSR)

It is well known that both of spatial depth and spatial ranks completely depend on each other for either the multivariate or functional data. The origins of the spatial approach date back to Brown (1983), when he introduced the idea of spatial median considering the problem of robust location estimation for two-dimensional spatial data. After that, the geometry notions of the data started to be used in different important nonparametric functions such that the multivariate spatial quantiles by Chaudhuri (1996) and the multivariate spatial depth function by Serfling (2002). Recently, the functional spatial depth (FSD) has been proposed by Chakraborty and Chaudhuri (2014), where they extended the notion of spatial depth from $\mathbb{R}^d$ into infinite dimensional spaces. They observed that the multivariate spatial depth function, $SD(\mathbf{x}) = 1 - \|E\{(\mathbf{x}-\mathbf{X})/\|\mathbf{x}-\mathbf{X}\|\}\|$, can be extended naturally to any Hilbert space, such that for any $\mathbf{x} \in \mathbb{H}$ and a random element $\mathbf{X} \in \mathbb{H}$, we can define the functional spatial depth $FSD(\mathbf{x})$ using the same expression as above, where $\|.\|$ is to be taken as the usual norm in $\mathbb{H}$ and the expectation is in the Bochner sense (Chakraborty and Chaudhuri, 2014; Araujo and Giné, 1980).

Serfling and Wijesuriya (2015) introduced a nonparametric description of the functional data using the spatial depth function. They used the functional version of spatial depth in some nonparametric descriptive features like the sample median curve, 50% most central region of sample curves, selected sample quantile curves, location outliers, and an additional possibility with functional data-shape outliers. Furthermore, Serfling and Wijesuriya (2015) pointed out that if $\mathbf{X}$ is a random function in the Hilbert space $\mathbb{H}$, then the population spatial quantile function $Q(\mathbf{u})$, indexed by elements $\mathbf{u}$ in the unit ball $\mathcal{B}$ in $\mathbb{H}$, is obtained by minimizing $\mathbb{E}\{\phi(\mathbf{u}, \mathbf{X}-\mathbf{x}) - \phi(\mathbf{u}, \mathbf{X})\}$ with respect to $\mathbf{x}$, where

$\phi(\mathbf{u}, \mathbf{z}) = \|\mathbf{u}\| + \langle u, z \rangle$, $\|\mathbf{u}\|$ is the L_2 norm, and $\langle u, z \rangle$ is the inner product defined on $\mathbb{H}$.

On the other hand, Cardot et al. (2013) proposed an averaged stochastic gradient algorithm in order to compute the functional spatial median in a Hilbert space in a fast way. Along the line, Chaouch and Goya (2012) used the functional spatial median that proposed by Cardot et al. (2013) as a robust measure of center for a data set of electricity loading curves.

The kernelized functional spatial depth (KFSD) has been introduced by Sguera et al. (2014). They used the KFSD based on the functional spatial depth, which introduced by Chakraborty and Chaudhuri (2014), for the classification of functional data. They performed the classification by using some robust methods that involve the use of a given functional depth functions, including FSD and KFSD. The functional K-nearest neighbor classifier has been used in their work as a benchmark procedure.

Definition 5.4.1 *: The functional sign function:*
Let $\{\mathbf{X}_1(t), \mathbf{X}_2(t), \ldots, \mathbf{X}_n(t)\}$ *be a functional dataset, generated by the functional random variables* $\{\mathcal{X}_1(t), \mathcal{X}_2(t), \ldots, \mathcal{X}_n(t)\}$*, taken values from Hilbert space* $\mathbb{H}$*, where* t *defined on some set* $\mathcal{T}$ *which represents an interval of time, of wavelengths or any other subset, then we define the functional spatial sign function for the curve* $\mathbf{x} \in \mathbb{H}$ *as:*

$$FSign(\mathbf{x}) = \begin{cases} \frac{\mathbf{x}}{\|\mathbf{x}\|} & \text{if } \mathbf{x} \neq 0 \\ 0 & \text{if } \mathbf{x} = 0, \end{cases} \tag{5.4.1}$$

where $\|\mathbf{x}\|$ *is the* L_2 *norm such that:*

$$\|\mathbf{x}\| = \left(\int_{\mathcal{T}} \Big(\mathbf{x}(t)\Big)^2 dt \right)^{1/2}, \tag{5.4.2}$$

for an infinite dimensional space.

According to Chakraborty and Chaudhuri (2014), a population functional version of

the spatial depth function is given by:

$$FSD(\mathbf{x}, \mathbf{P}) = 1 - \|\mathbb{E}\left\{FSign(\mathbf{x} - \mathbf{X})\right\}\|, \tag{5.4.3}$$

where $\mathbf{x} \in \mathbb{H}$ and $\mathbf{P}$ is a probability distribution on $\mathbb{H}$. On the other hand, when a sample of curves is observed, such that $\mathbf{x} \in \mathbf{X}_i; i = 1, \ldots, n$, then the population functional spatial depth function $FSD(\mathbf{x}, \mathbf{P})$ can be replaced with its corresponding sample version:

$$FSD_n(\mathbf{x}) = 1 - \frac{1}{n}\|\sum_{i=1}^{n} FSign(\mathbf{x} - \mathbf{X}_i)\|. \tag{5.4.4}$$

In Chakraborty and Chaudhuri (2014), some important properties of FSD have been considered and they can be applied for the functional spatial rank function as well, these properties are (Sguera et al., 2014):

1. $FSD(\mathbf{x}, \mathbf{P})$ is invariant under the class of linear transformations $T : \mathbb{H} \to \mathbb{H}$, where $T(\mathbf{x}) = cA\mathbf{x} + b$ for some $c > 0, b \in \mathbb{H}$ and an isometry A on $\mathbb{H}$.

2. If $\mathbf{P}$ is non-atomic, then $FSD(\mathbf{x}, \mathbf{P})$ is continuous in $\mathbf{x}$.

3. If $\mathbb{H}$ is strictly convex and $\mathbf{P}$ is non-atomic and not supported on a line in $\mathbb{H}$, then $FSD(\mathbf{x}, \mathbf{P})$ has a unique maximum at the spatial median m of Y and its maximum value is 1.

4. For any fixed non-zero $\mathbf{x} \in \mathbb{H}$ and sequence $\{m + n\mathbf{x}\}_{n \in N+}$, the following holds: $FSD(m + n\mathbf{x}, \mathbf{P}) \to 0$ as $n \to \infty$.

5. $FSD(\mathbf{x}, \mathbf{P})$ does not suffer from degeneracy for many infinite dimensional probabilities distributions.

Definition 5.4.2 *: The functional spatial rank function:*
Suppose that $\mathbf{X}(t) \in \mathbb{H}$ *has an infinite dimensional distribution* F*, where* t *defined on*

some set $\mathcal{T}$ which represents an interval of time, of wavelengths or any other subset, then a population functional spatial rank function of the curve $\mathbf{x} \in \mathbb{H}$ can be defined as:

$$FSR_F(\mathbf{x}) = \mathbb{E}\left(\frac{\mathbf{x}-\mathbf{X}}{\|\mathbf{x}-\mathbf{X}\|}\right). \tag{5.4.5}$$

In reality we have sampled curves observed into a finite set of observations, such that we have discrete observations X_{ij} of each sample path $\mathbf{X}_i(t)$ at a finite set of knots $\{t_{ij} : j = 1, \ldots, m_i\}$. Now suppose that $\{\mathbf{X}_1(t), \mathbf{X}_2(t), \ldots, \mathbf{X}_n(t)\}$ is sampled curves observed into a finite set of observations, then the sample version of the functional spatial rank of $\mathbf{x}(t)$ with respect to $\mathbf{X}_1(t), \mathbf{X}_2(t), \ldots, \mathbf{X}_n(t)$ is given by:

$$FSR_{F_n}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} FSign(\mathbf{x}-\mathbf{X}_i) = \frac{1}{n}\sum_{i=1}^{n}\frac{\mathbf{x}-\mathbf{X}_i}{\|\mathbf{x}-\mathbf{X}_i\|}. \tag{5.4.6}$$

Two different kinds of the sampled curves are usually considered in the functional data literature. The first one is the regularly sampled curves, and the second one is the irregularly sampled curves. In the regularly sampled curves, the evaluation points $t \subset \mathcal{T}$ are supposed to be fixed for each curve with the same length and knots, such that we have discrete observations X_{ij} of each sample path $\mathbf{X}_i(t)$ at a finite set of knots $\{t_{ij} : j = 1, \ldots, m\}; i = 1, \ldots, n$. Note that the curves' length (m) is fixed among the different functions. As example for the regularly sampled curves is the curves of gun-point and growth data that shown in Figure 5.1, where each curve consists of discrete observations based on the same set of evaluation points (time and age knots in gun-point and growth data respectively).

On the other hand, the irregularly sampled curves assume that the evaluation points are different and each curve has its own length based on the number of knots that represent the discrete observations in the sampled curve. In other words, we have discrete observations X_{ij} of each sample path $\mathbf{X}_i(t)$ at a finite set of knots $\{t_{ij} : j = 1, \ldots, m_i\}; i =$

$1, \ldots, n$, where the curves' length (m_i) changes for each function. An example for the irregularly functional sampled curves is given in Figure 1 of James and Hastie (2001), where they used the functional linear discriminate analysis for irregularly functional sampled data. The Figure gives measurements of spinal bone mineral density for 280 males and females with different periods of time for each individual (curve).

In principle, the proposed functional spatial ranks can be applied for both of regularly and irregularly sampled curves, since the functional spatial ranks are supposed to be calculated in general concept using the integrations instead of the summations quantities, then with a formal procedures and methods we can estimate the integral functions and get the estimated values of the functional spatial ranks. However, in this study we only consider the regularly sampled curves in order to keep the simplicity of the functional forward search algorithm and to save on computational time, but we are looking into the irregularly sampled curves case as a future work.

Preprocessing functional data is usually used in the functional data analysis in two steps (Jacques and Preda, 2014). The first one is by reconstructing the functional form of data because the curves are usually observed at discrete observation points, which requires being in functions frame. The second step is to center and scale the curves in order to eliminate both phase and amplitude variations into the curve's dataset. The last step is known as the data registration (Ramsay and Silverman, 2005).

Preprocessing functional data in clustering analysis has been considered a questionable point. Jacques and Preda (2014) gave good examples of the side effects that are usually happened under considering the data registration in the clustering framework. The loss of cluster information is one of these effects and can be misleading in different cases. Furthermore, Jacques and Preda (2014) pointed out that amplitude variation of the data is considered a source of differentiation between clusters, and doing the preprocessing for the data may remove this variation and consequently cause a loss of the clustering information.

For this reason, the majority of the existing works does not perform data registration, assuming that either this effect is only limited or it contains cluster information.

Example for works that consider the registration step is the work by Liu and Yang (2009), and example for works that identify the clusters due to phase and amplitude variations is Slaets et al. (2012). On the other hand, Liu and Yang (2009), Sangalli et al. (2010 a,b) proposed different procedures that simultaneously align and cluster the curves without performing data registration before clustering. It is worth mentioning in this context that, in our proposed method we do not perform data registration or any preprocessing step before the clustering work.

5.4.2 The Functional Forward Search Algorithm Based on FSR

In the forward search algorithm, let $S(m)$ be a subset of size m at a particular stage. Then define the functional spatial ranks of individual curve corresponding to the subset $S(m)$ as:

$$r_i(m) = \frac{1}{m} \sum_{j \in S(m)} \frac{\mathbf{X}_i - \mathbf{X}_j}{\|\mathbf{X}_i - \mathbf{X}_j\|}, \tag{5.4.7}$$

for $i = 1, \ldots, n$. In this context, we point out that the forward search procedure based on the functional spatial ranks is similar to the one based on the multivariate spatial ranks that introduced in (Baragilly and Chakraborty, 2016), with only two differences. The first difference is that we suppose that $S(m)$ is the initial subset with size $m = 3$ instead of $d+1$. The second difference is that we deal here with sampled curves $\mathbf{X}_1(t), \mathbf{X}_2(t), \ldots, \mathbf{X}_n(t)$ taken values from Hilbert space $\mathbb{H}$, not a random sample$\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n \in \mathbb{R}^d$. The functional forward search algorithm with functional spatial ranks is:

1. Start the search with an initial subset $S(m)$ with size $m = 3$, then one search can be run from this starting point.

2. Calculate the functional spatial ranks $r_i(m)$ depending on the curves in the subset

$S(m)$.

3. Compute $r_{min}(m)$, where $r_{min}(m) = \min \|r_i(m)\|$; $i \notin S(m)$.

4. Grow the subset $S(m)$ to $S(m+1)$ by taking $m+1$ observations $\mathbf{X}_i(t)$'s, which correspond to smallest $m+1$ $\|r_i(m)\|$'s. Set $m = m+1$.

5. Iterate $2-4$ until $m = n-1$.

6. The forward plot of the spatial ranks can be obtained by plotting the $r_{min}(m)$ against the corresponding subset sizes m.

Like the forward search algorithm with spatial ranks for the multivariate case, this algorithm is computationally easy and straightforward as well. When the curves in $S(m)$ belong to the same cluster, $\|r_i(m)\|$ for a curve $\mathbf{X}_i(t)$ belonging to the same cluster is expected to be smaller than that of curve from a different cluster, and whenever $S(m)$ grows bigger than the cluster it originally belonged to, we expected to see a jump in the magnitude of the rank function as the nearest point to $S(m)$ is then from a different cluster.

5.4.3 Numerical Examples

In this section, we apply the proposed functional forward search algorithm on some simulated and real data. We start with the simulated data, where we considered three models, with different mean functions and number of groups. Then, we implement the algorithm on three real datasets, two of them have been discussed already in Section 5.2 (gun-point and growth data) and the third one is ECG data. More discussion about ECG data is given in this Section.

Simulated Data

In order to check the performance of our proposed method, we give some numerical example based on simulated data. Three models have been considered here, with different

mean functions and numbers of clusters. In the first model we assume that we have two groups, the first one consists of the curves of the process:

$$X(t) = m_0(t) + e(t), \tag{5.4.8}$$

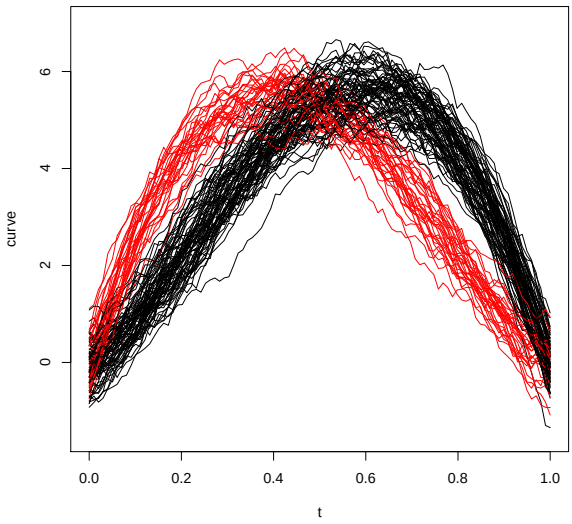
where the mean function $m_0(t) = 30(1-t)t^{1.5}$ and $e(t)$ is a Gaussian process with mean 0 and $Cov(X(s), X(t)) = 0.2exp(-|s-t|/0.3)$. On the other hand, the second group consists of the curves of the process:

$$Y(t) = m_1(t) + e(t), \tag{5.4.9}$$

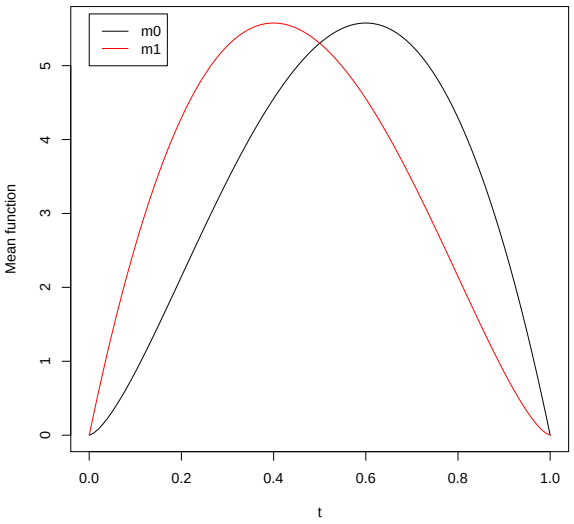
with mean function $m_1(t) = 30t(1-t)^{1.5}$ and same $e(t)$ and $Cov(X(s), X(t))$ in the process $X(t)$.

Figure 5.39 (a) and (b) show the curves of model 1 and its mean function respectively. Again, the black color refers to the curves located in the first cluster while the red color refers to the curves that simulated to be in the second cluster. Initially, we can clearly see that we have two different groups since we have simulated the curves to be divided into two clusters with mixing proportion p which is taken to be 0.3 , and a sample size $n = 100$. So we expect to see two clear peaks around subsets with sizes 30 and 70. The panels (c) and (d) of Figure 5.39 give the forward plot and the entry plot of the first simulated functional data (model 1) based on the functional spatial ranks. It can be clearly seen that, there are two obvious maxima around subsets with sizes 30 and 70, which suggest dividing the data into two groups with sizes 30 and 70. Obviously, it is a perfect result, where the method was able to detect the number of clusters and determine their sizes.

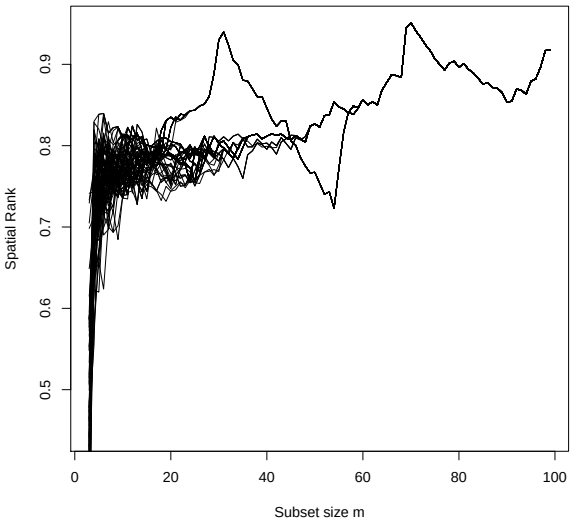
Now, we move to the second model which is a little bit complicated. In this model we simulated the curves to be divided into two groups as well with the same mixing proportion ($p = 0.3$) and sample size $n = 100$. The difference in this model is that,



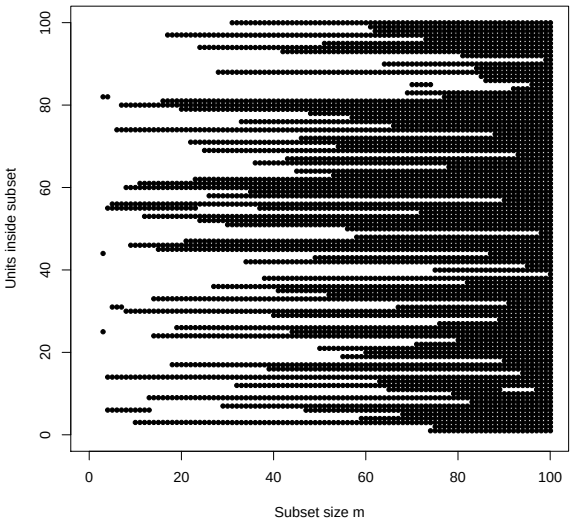
(a) Model 1 curves



(b) Mean function



(c) Forward plot based on FSR



(d) Entry plot based on FSR

Figure 5.39: Simulated data, Model 1: (a) the observed curves with two groups, (b) the mean function, (c) the forward plot based on FSR and (d) the entry plot based on FSR.

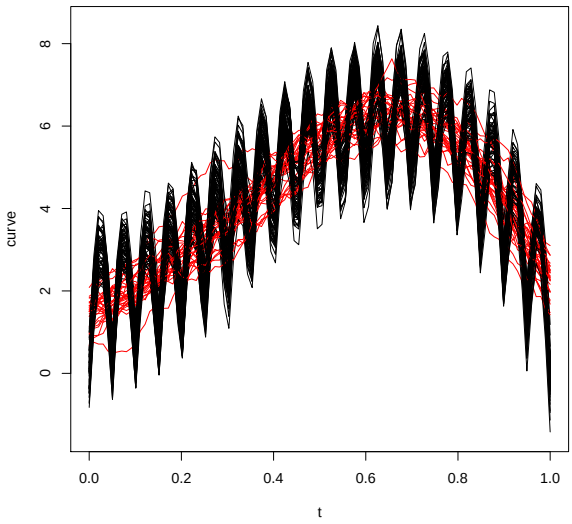
the second group is a smoothing of the curves of the first group which makes it more difficult to discriminate the overlapping between the two groups. Here, we assumed that the first cluster consists of the curves of the process $X(t)$ that defined in (5.4.8), but with mean function: $m_0(t) = 30(1-t)t^2 + 3\,|sin(20\pi t)|$ and $e(t)$ is a Gaussian process with mean 0 and $Cov(X(s), X(t)) = 0.2exp(-|s-t|/0.3)$. The second group is made of spline approximations, with 8 knots, of trajectories from the previous process.

Figure 5.40 shows the curves of model 2, its mean function, the forward plot based on the functional spatial rank, and the entry plot. Like model 1, our target is to get two clear maxima around the subset with sizes 30 and 70. From the forward plot, we can clearly see the two peaks at $m = 30$ and 70, which means that our method successfully determined the number of clusters and their sizes as well.

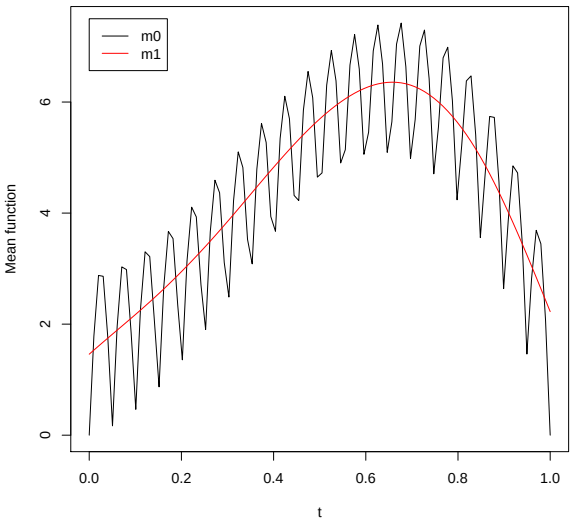
In order to check the stability of our method with bigger numbers of clusters, we assumed that we have three groups in the third model, such that model 3 is a combination between model 1 and model 2, since we assumed that we have the two groups in model 2 plus a third group which consists of the curves of the process (5.4.8) with the same mean and covariance functions. This model is more complicated than the other two models. The target now is to get three peaks around subsets with sizes 20, 30 and 50 since we simulated the data to include three clusters with sizes 20, 30 and 50. Figure 5.41 gives the curves of model 3, its mean function, the forward plot based on the functional spatial ranks and the entry plot. The forward plot shows three clear peaks around sizes 20, 30 and 50 which successfully clusters the simulated data into three groups.

Real Data

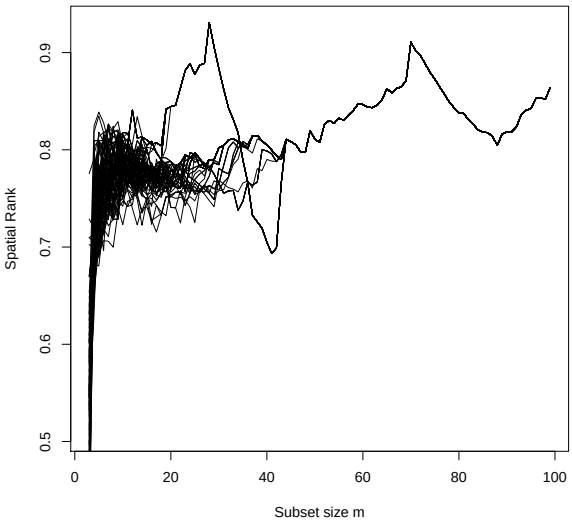
More numerical examples are given in this Section where we considered three different real datasets. The first and second datasets are the gun-point and growth data that have been analyzed in Section 5.2. The third one is known as ECG data. It is taken from the UCR Time Series Classification and Clustering Archive. This dataset consists



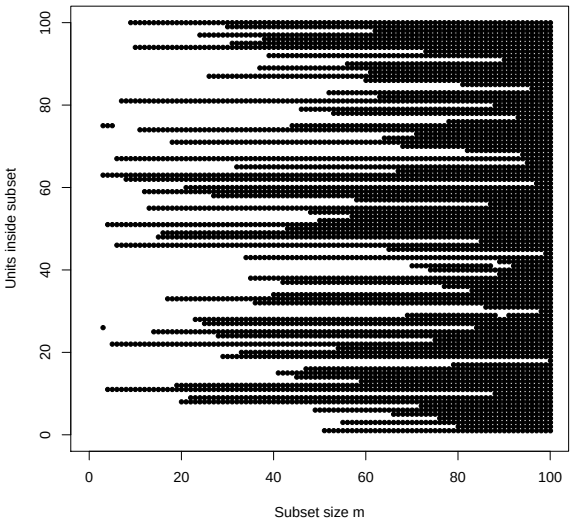
(a) Model 2 curves



(b) Mean function

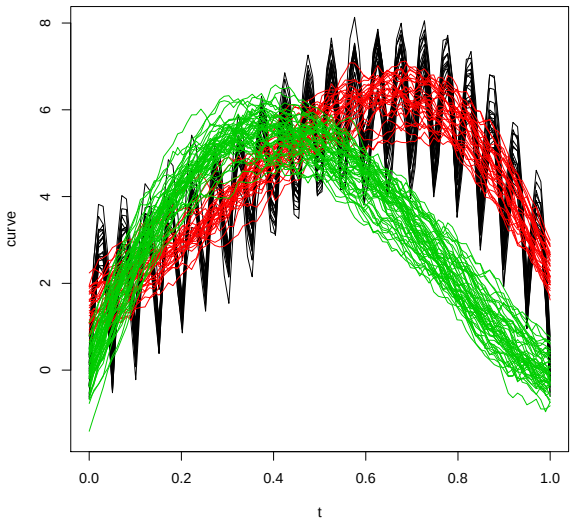


(c) Forward plot based on FSR

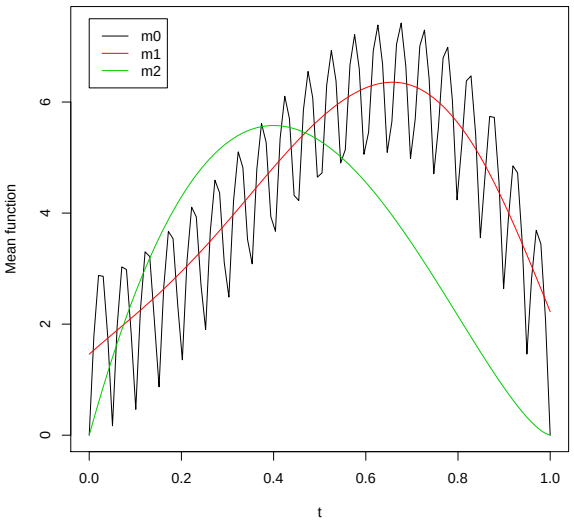


(d) Entry plot based on FSR

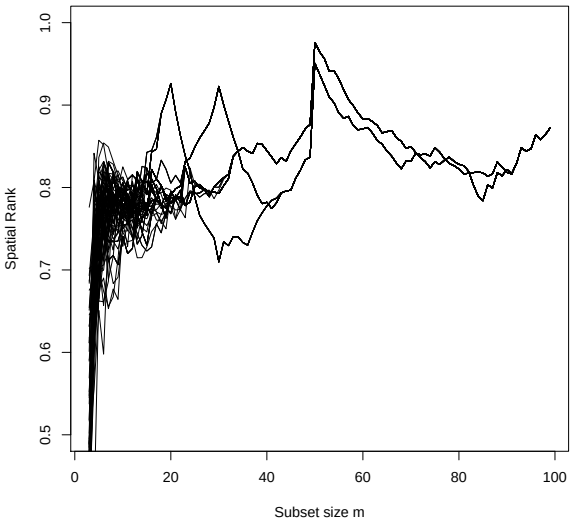
Figure 5.40: Simulated data, Model 2: (a) the observed curves with two groups, (b) the mean function, (c) the forward plot based on FSR and (d) the entry plot based on FSR.



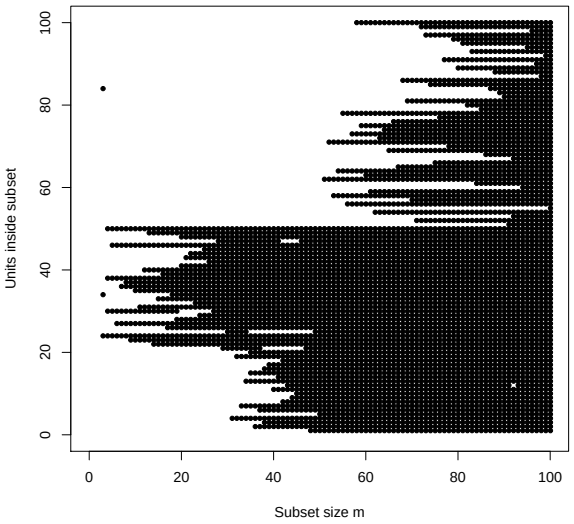
(a) Model 3 curves



(b) Mean function



(c) Forward plot based on FSR



(d) Entry plot based on FSR

Figure 5.41: Simulated data, Model 3: (a) the observed curves with three groups, (b) the mean function, (c) the forward plot based on FSR and (d) the entry plot based on FSR.

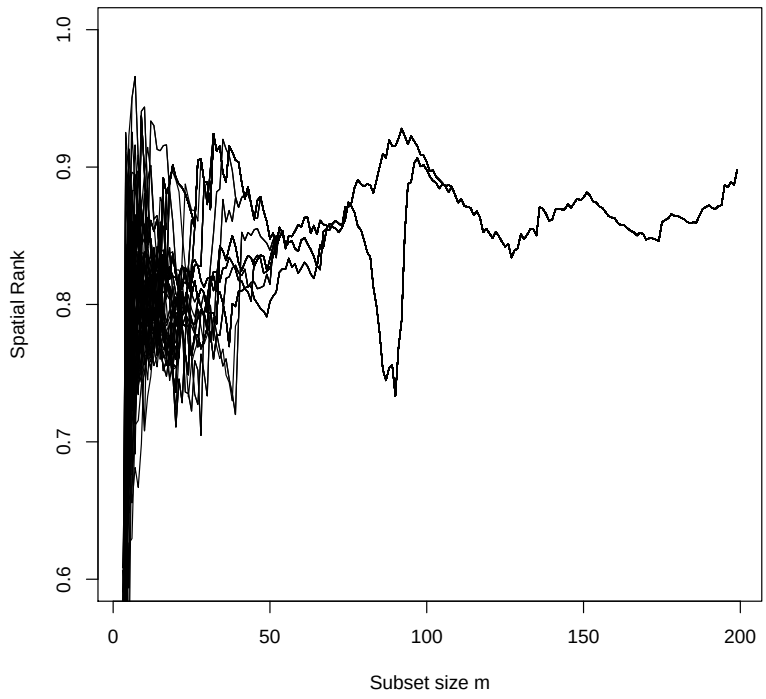


Figure 5.42: Gun-point data: Forward plot based on the functional spatial ranks. Two clusters are evident around subsets with sizes 100.

of 200 electrocardiograms from 2 groups of patients sampled at 96 time instants, of which 133 are normal and 67 are abnormal. The ECG data uses two electrodes to collect data during one heartbeat which is classified of normal or abnormal. Abnormal heartbeats are representative of a cardiac pathology known as supraventricular premature beat. The data has already been studied in Olszewski (2001).

Figure 5.42 gives the forward plot based on the functional spatial rank for the gun-point data. It has mentioned earlier that this data consists of two clusters each one includes 100 curves. From the forward plot we can see that there are two clear maxima, both of them are around subsets with sizes 100. Obviously each peak is located in different trajectory and suggests the right number of clusters and their sizes.

Regarding to growth data, it consists of two clusters, the first one includes 39 males

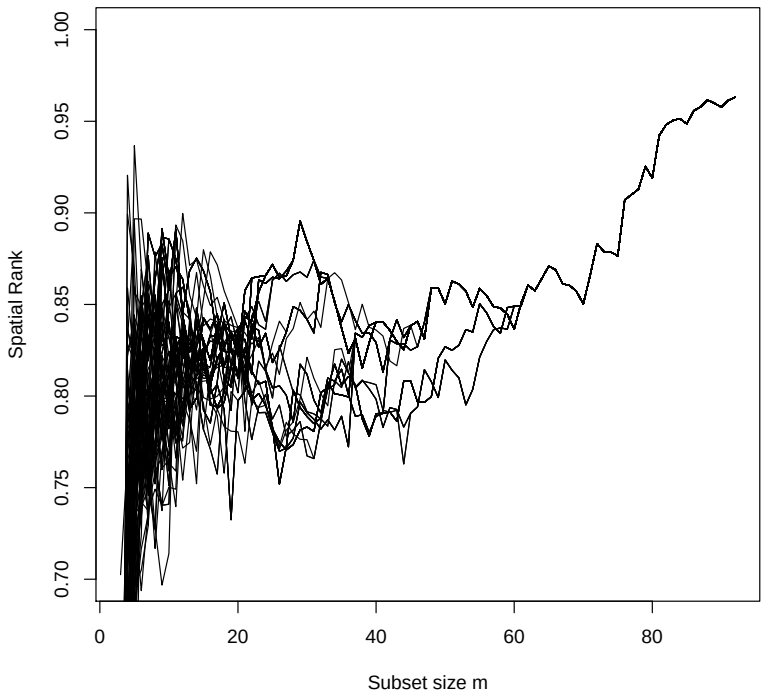
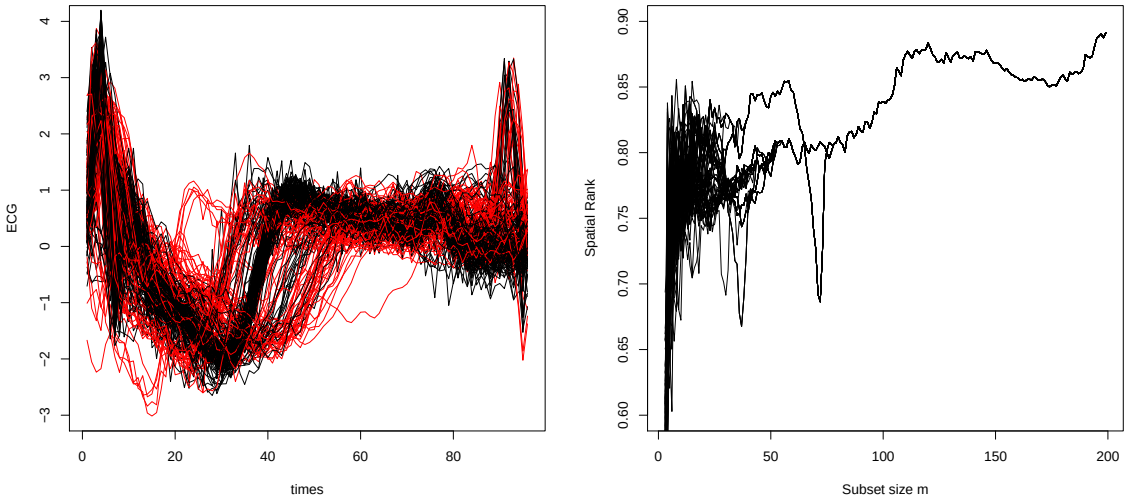


Figure 5.43: Growth data: Forward plot based on the functional spatial ranks. Two clusters are evident around subsets with sizes 39 and 54.

and the second one includes 54 females. Figure 5.43 shows the forward plot based on the functional spatial rank for growth data. It can be clearly seen that there are two peaks in the plot, one is around 39 and the other is around 54 which successfully suggests the right number of clusters and their sizes as well.

Figure 5.44 shows the curves located in the first and second clusters of ECG data, and the forward plot based on functional spatial ranks of ECG data. As we can see from panel (b) of Figure 5.44, there are two peaks in the forward plot, the first one is around 67 and the second one is around 133 which suggests two clusters with sizes 67 and 133 and this is the right size of the normal and abnormal heartbeats' groups. At the end of this section we can conclude that the forward search plots based on functional spatial ranks show that the algorithm performs well under different functional data models.



(a) ECG curves

(b) Forward plot based on FSR

Figure 5.44: ECG data: panel (a) is the observed curves with two groups and panel (b) is the forward plot based on the functional spatial ranks. Two clusters are evident around subsets with sizes 67 and 133.

The greatest advantage of the proposed forward search is that it is not necessary to assume that the underlying data are coming from parametric family or any known distribution as we are not estimating any parameters in our method. However, for large number of clusters the proposed forward search plots may produce many peaks and makes it difficult visually to determine the exact number of clusters and the cluster size. This is because that the points start joining in from different cultures. Nevertheless, we are looking into the problem of estimating the number of clusters from the forward search plot by using formal procedure as a future research.

5.5 Functional Data Clustering Based on Weighted Spatial Ranks

The weighted functional spatial ranks (WFSR) method is proposed in this section. It is an extension work to the WSR that has been proposed in chapter 4, where we considered the data to be in curves or functions version. A dimensional reduction using FPCA is required as a first step (filtering step) in order to get the weighted ranks contours for a low-dimensional input space, and consequently determine the number of clusters. As a second step (clustering step), we use the weighted functional spatial rank classifier (WFSRC) to assign each curve to the suitable cluster. The proposed WFSR method can be classified under the group of filtering methods based on FPCA scores. As discussed earlier, the selection of a proper weight function leads to better identification of clusters. Here, we use the Gaussian kernel weights that have been used in the multivariate case.

5.5.1 The Weighted Functional Spatial Rank (WFSR)

Definition 5.5.1 *: The weighted functional spatial ranks function:*
Let $\{\mathbf{X}_1(t), \mathbf{X}_2(t), \ldots, \mathbf{X}_n(t)\}$ *be a sampled curves observed into a finite set of observations and generated by the functional random variables* $\{\mathcal{X}_1(t), \mathcal{X}_2(t), \ldots, \mathcal{X}_n(t)\}$*, taken values from Hilbert space* $\mathbb{H}$*, where* t *defined on some set* $\mathcal{T}$ *which represents an interval of time, of wavelengths or any other subset, then the sample version of the weighted functional spatial rank of* $\mathbf{x}(t)$ *with respect to* $\mathbf{X}_1(t), \mathbf{X}_2(t), \ldots, \mathbf{X}_n(t)$ *is given by:*

$$WFSR_{F_n}(\mathbf{x}) = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i FSign(\mathbf{x} - \mathbf{X}_i) = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i \frac{\mathbf{x} - \mathbf{X}_i}{\|\mathbf{x} - \mathbf{X}_i\|}, \qquad (5.5.1)$$

and the second weighted functional spatial ranks function that corresponds to (4.3.2) is:

$$WFSR_{F_n}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} w_i FSign(\mathbf{x}-\mathbf{X}_i) = \frac{1}{n}\sum_{i=1}^{n} w_i \frac{\mathbf{x}-\mathbf{X}_i}{\|\mathbf{x}-\mathbf{X}_i\|}. \tag{5.5.2}$$

and their L_2 norm is:

$$WFSRN_{F_n}(\mathbf{x}) = \|WFSR_{F_n}(\mathbf{x})\|. \tag{5.5.3}$$

where the weights w_i is the Gaussian kernel weights that were giving the best identification of clusters as we presented in Chapter 4, can be written as:

$$w_i = e^{-\frac{\|\mathbf{x}-\mathbf{X}_i\|^2}{2}}. \tag{5.5.4}$$

Now, we discuss how to use the WFSR as a clustering tool for the functional data. Actually, the WFSR is more accurate in the purpose of intuitive visualization since we can easily determine the number of clusters from the weighted functional ranks contours for a low-dimensional input space, using dimension reduction. In order to plot the weighted functional ranks contour we use the FPCA components $\{C_j\}_{j\geqslant 1}$ that are defined in equation (5.2.4), where we need to get the first 2 components C_1 and C_2 such that:

$$C_1 = \int_0^T \big(X(t)-\mu(t)\big)\xi_1(t)dt, \tag{5.5.5}$$

and,

$$C_2 = \int_0^T \big(X(t)-\mu(t)\big)\xi_2(t)dt, \tag{5.5.6}$$

and let S_{C_1} and S_{C_2} are two vectors of the generated regular sequences from the minimum to maximum value of C_1 and C_2 respectively with length l, and let $\mathbf{S}$ be a matrix consists of the two victors S_{C_1} and S_{C_2} and $\mathbf{C}$ be a matrix consists of the first two components C_1 and C_2, then the outer product of the arrays S_{C_1} and S_{C_2} is the array Z based on the

WFSRN of $\mathbf{s}$ with respect to $\mathbf{C}_i$, where $\mathbf{s} \in \mathbf{S}$:

$$WFSRN(\mathbf{s}) = \|WFSR(\mathbf{s})\| = \|\frac{1}{n}\sum_{i=1}^{n} w_i \frac{\mathbf{s}-\mathbf{C}_i}{\|\mathbf{s}-\mathbf{C}_i\|}\|, \tag{5.5.7}$$

where $i = 1, \ldots, n$ and $w_i = exp(-\|\mathbf{s}-\mathbf{C}_i\|^2/2)$, then the weighted functional ranks contour lines can be existed based on S_{C_1}, S_{C_2} and Z. So, we can get the weighted functional ranks contour that can directly capture the clusters structure and suggest the number of clusters consequently.

The main idea behind WFSR is to define a dissimilarity measure locally based on a localized version of functional ranks. As a result, the proposed method can be used to determine the assumed number of clusters, and to assign each observation to its cluster. Selection of a proper weight function will lead to better identification of clusters when the data do not follow any standard parametric distribution.

5.5.2 Confirmatory Analysis Based on Weighted Functional Spatial Ranks Classifier

In this section, we propose a confirmatory nonparametric classifier that can be used to check and confirm if the curves' assignment based on the weighted functional spatial rank contours are right or not. The proposed confirmatory classifier is based on the weighted functional spatial ranks, and it is simple and easy to compute without any need to parameter estimates of the underlying distributions.

Two clusters case:

Suppose that we have two groups, with distributions F and G respectively, then based on the weighted functional ranks classifier (WFSRC) rule, we can assign the curve $\mathbf{x}(t)$ to the first group if the L_2 norm of the weighted functional spatial ranks of the curve $\mathbf{x}(t)$ based on F is less than the L_2 norm of the weighted spatial ranks of the curve $\mathbf{x}(t)$ based

on G such that assign $\mathbf{x}(t)$ to the group with distribution F if:

$$WFSRN_F(\mathbf{x}) < WFSRN_G(\mathbf{x}), \tag{5.5.8}$$

and assign it to the second group with distribution G otherwise, where $WFSRN_F(\mathbf{x}) = \|WFSR_F(\mathbf{x})\|$ and $WFSRN_G(\mathbf{x}) = \|WFSR_G(\mathbf{x})\|$ as defined in equation (5.5.3).

More than two clusters:

Suppose that we have K groups, with distributions $F_1, F_2, \ldots, F_k$, then we can assign the curve $\mathbf{x}(t)$ to the i-th group if:

$$WFSRN_{F_i}(\mathbf{x}) = \min_{1 \leqslant j \leqslant k} WFSRN_{F_j}(\mathbf{x}), \tag{5.5.9}$$

where $i \neq j, 1 \leqslant i \leqslant k$.

Thus, like the multivariate case, we can use the above weighted functional spatial ranks classifier as a confirmatory analysis to assign each observation to the most suitable clusters, after determining the number of clusters using the weighted functional spatial ranks contours. Furthermore, we can use the confirmatory plot based on the weighted functional spatial ranks, which can be used easily to now the assignment of each curve as we present in the numerical examples section.

5.5.3 The Weighted Functional Spatial Ranks Based Clustering Algorithm

The steps of the weighted functional spatial ranks based clustering algorithm are:

1. Use the functional principle component analysis FPCA for the purpose of the dimension reduction, and get the first 2 components C_1 and C_2 as shown in (5.5.5) and (5.5.6) and construct the matrix $\mathbf{C}$ that consists of C_1 and C_2.

2. Get S_{C_1} and S_{C_2} as vectors of the generated regular sequences from the minimum to maximum value of C_1 and C_2 respectively with length l.

3. For each $\mathbf{s} \in \mathbf{S}$, calculate $WFSRN(\mathbf{s})$ with respect to $\mathbf{C}$ as shown in (5.5.7).

4. Get Z as the outer product of the arrays S_{C_1} and S_{C_2} based on the $WFSRN(\mathbf{s})$ in step 3.

5. Plot the weighted functional spatial ranks contour based on S_{C_1}, S_{C_2} and Z, and determine the number of clusters K from the contour lines.

6. Based on the contour lines, specify the observations that are allocated in each cluster. We can use a lower contour level for better visualization.

7. Use the confirmatory weighted functional spatial rank classifier's rule that is shown in (5.5.8)/(5.5.9) to confirm the assignment of each curve, assign the unassigned curves to the proper cluster, and get the confirmatory plot.

5.5.4 Numerical Examples

Here, we apply the proposed weighted functional spatial rank method on some simulated and real data. Again, we start with the simulated data, where we considered the three models that have been proposed in the last Section. Then we implement the method on the gun-point and growth datasets.

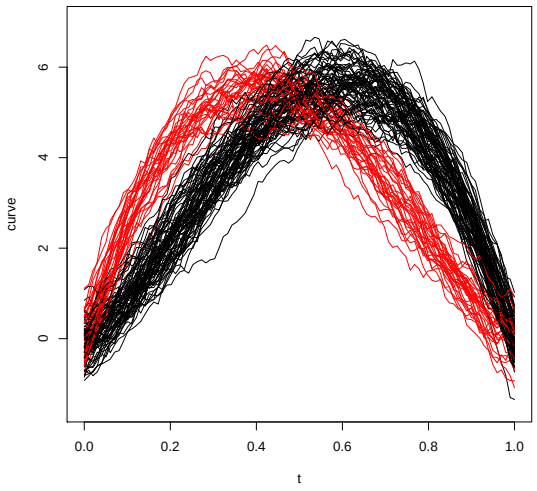
Simulated Data

Now, we consider model 1 which has been proposed in the previous Section. Panel (b) of Figure 5.45 gives plot of the functional PCA with the percentage of the total variance explained based on the scores of the first two harmonics of model 1. They explain 90.5% of the total variances. According to the contour plot of the first two harmonics based on WFSR, we can clearly see that the WFSR contour captures the clusters structure

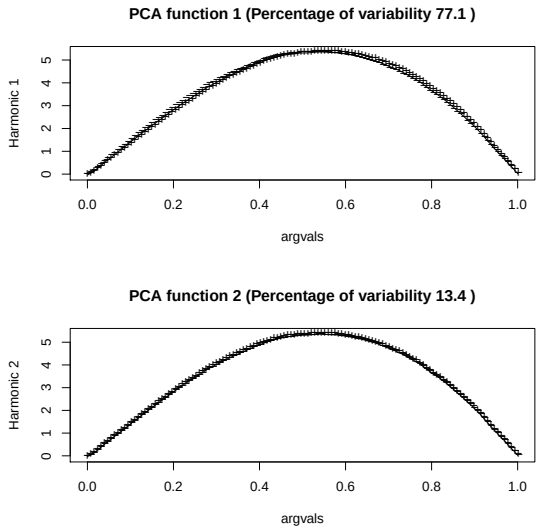
and shows two clear groups in the reduced dimension. In order to accurately assign each curve to the proper cluster, we use a low- level of the WFSR contour as shown in panel (d) of Figure 5.45. It can be clearly seen that all the score-points have been allocated properly in one of the two clusters. However, only two points (68 and 85) have not been assigned to any of them. In this case, the WFSRC plays an important role as a nonparametric classifier that assigns these points to the suitable group. Panels (e) and (f) show the confirmatory plots based on WFSRC for the first 2 components and the original data respectively. Comparing the curves in confirmatory plot with the observed curves, we can see that our method successfully clustered the simulated data without any misclassification error. This is a good result however; the real datasets will not be easy to get such a perfect result, as we will see in the next subsection.

Now we move to the second model. Figure 5.46 gives the observed curves of model 2 with two groups, plot of the functional PCA showing the total variance explained by each harmonic, the WFSR contours, the contour at level 0.005 and confirmatory plots based on WFSRC for the first 2 components and the original data respectively. As we can see from Figure 5.46 (b), the first two harmonics of model 2 explain 68.2% of the total variances. The contour plot of these two harmonics based on WFSR, shows that the WFSR contour captures the cluster structure and shows two clear groups. As shown in panel (d), we use a low-level of the WFSR contour at level 0.005 to clearly assign each curve to the proper cluster. As we can see, all the score-points have been allocated properly in one of the two clusters except the point number 18. By using the WFSRC, we confirmed that the curve number 18 belongs to the second group. Again we can see that our method successfully clustered this model without any misclassification error.

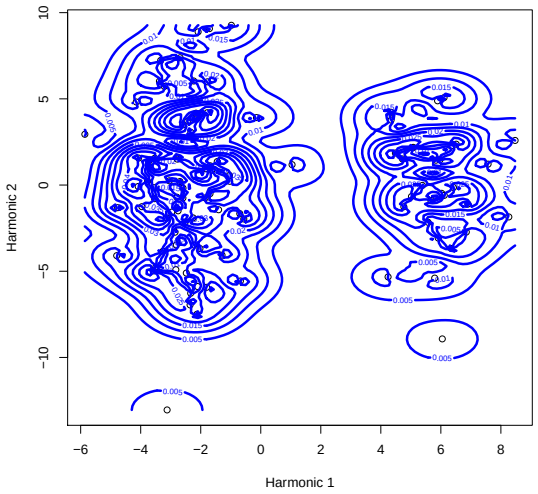
For the third model, which includes 3 clusters, Figure 5.47 gives the observed curves of model 3 with three groups, plot of the functional PCA with the total variances explained by each harmonic, the WFSR contours, the contour at level 0.01 and confirmatory plots



(a) Model 1 curves

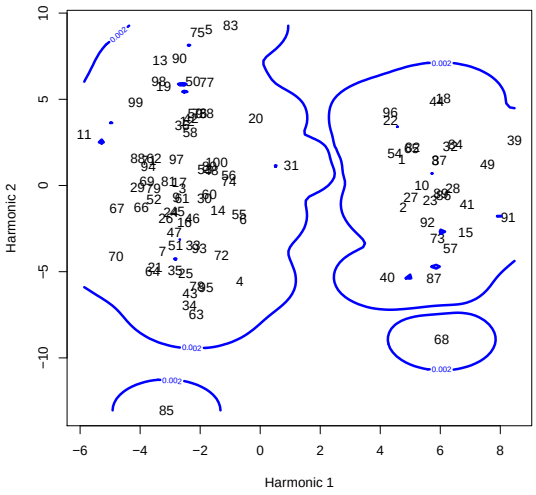


(b) Functional PCA

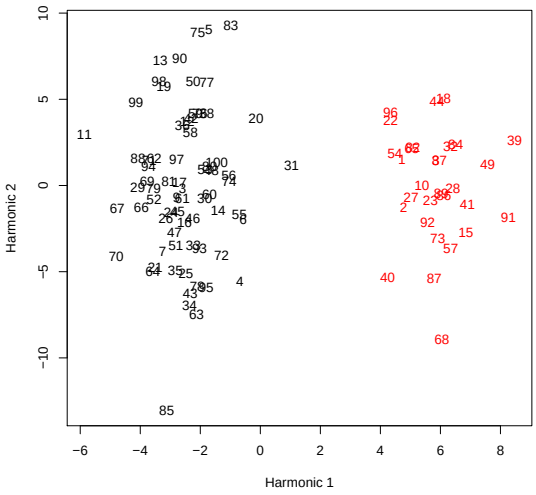


(c) WFSR contours

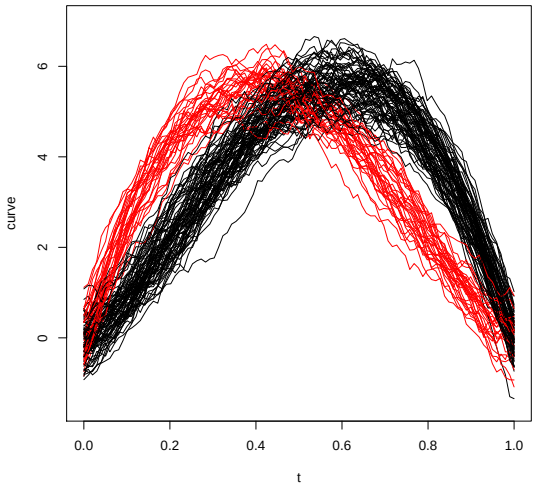
Figure 5.45: Simulated data, Model 1: (a) the observed curves with two groups, (b) plot of the functional PCA with the total variance explained by each harmonic, (c) the WFSR contours, (d) the contour at level 0.002, and confirmatory plots based on WFSRC for (e) the first 2 components and (f) the observed curves.



(d) Contour at level 0.002

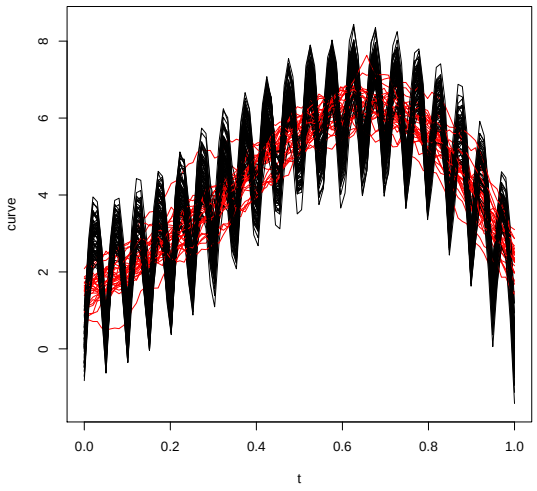


(e) Confirmatory plot: first 2 components

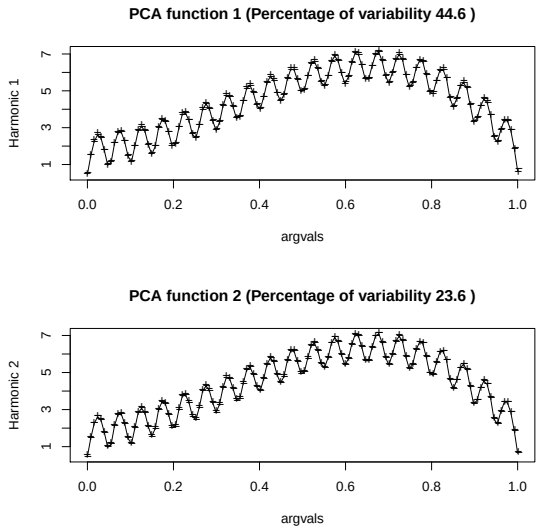


(f) Confirmatory plot: observed curves

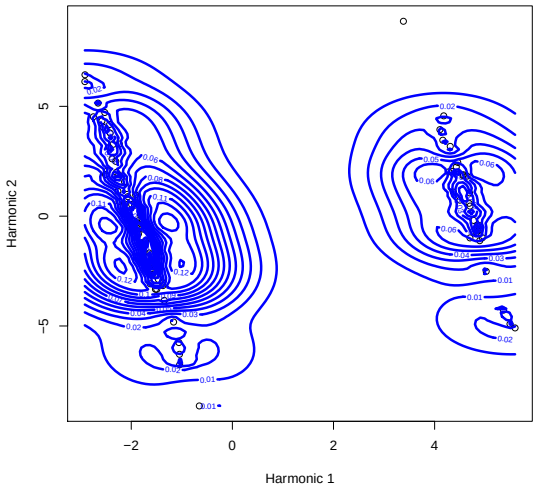
Figure 5.45: Continued.



(a) Model 2 curves

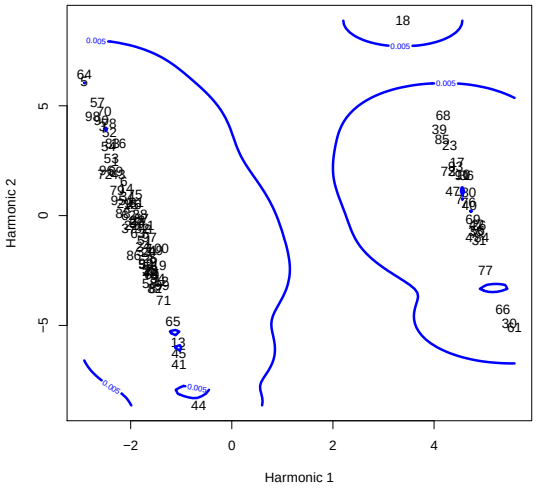


(b) Functional PCA

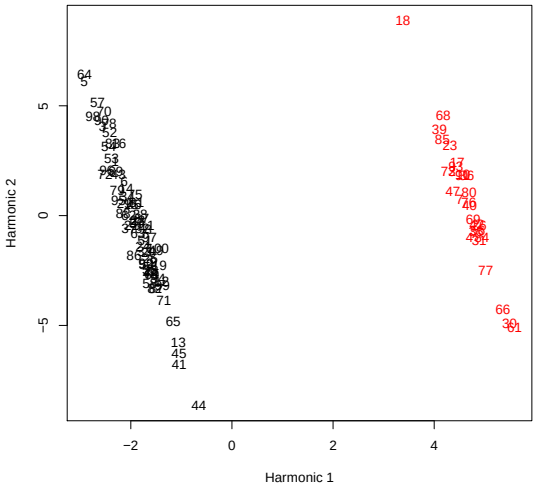


(c) WFSR contours

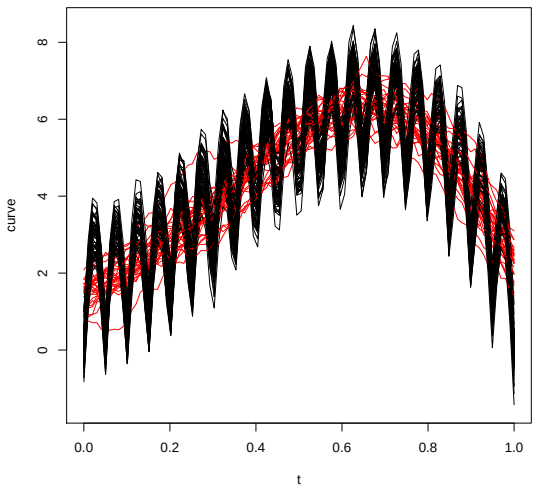
Figure 5.46: Simulated data, Model 2: (a) the observed curves with two groups, (b) plot of the functional PCA with the total variance explained by each harmonic, (c) the WFSR contours, (d) the contour at level 0.005, and confirmatory plots based on WFSRC for (e) the first 2 components and (f) the observed curves.



(d) Contour at level 0.005



(e) Confirmatory plot: first 2 components



(f) Confirmatory plot: observed curves

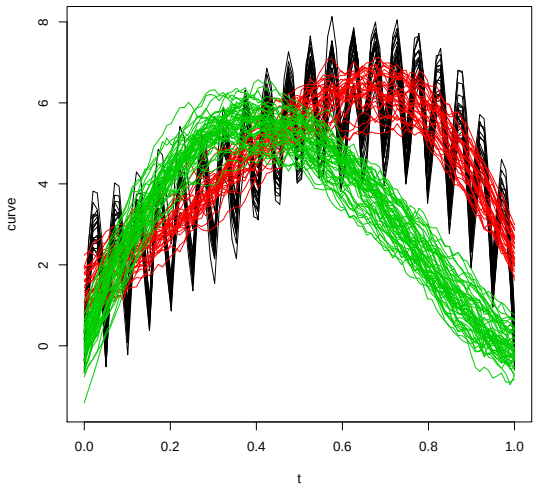
Figure 5.46: Continued.

based on WFSRC for the first 2 harmonics and the original curves respectively. The first two harmonics of model 3 explain 91.4% of the total variances. Furthermore, the WFSR contour captures the cluster structure the three clusters. From the low-level of the WFSR contour at level 0.01, we can clearly see that all the score-points have been assigned properly in one of the three clusters without any misclassification error.

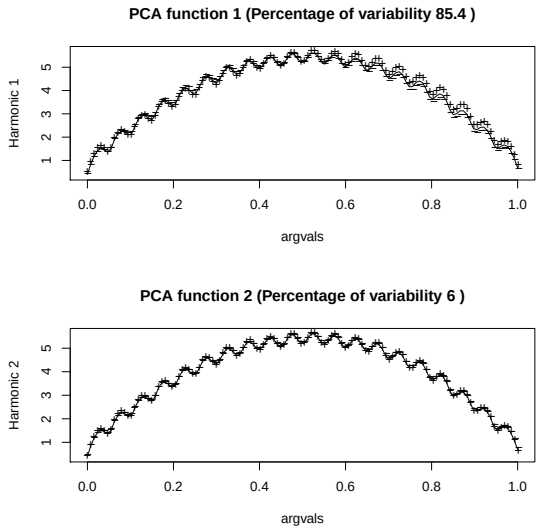
Real Data

Here, we check the performance of the weighted functional spatial ranks based clustering algorithm when we consider some real functional data. The real functional datasets are gun-point and growth datasets that have been extensively analyzed in Section 5.2. Figure 5.48 and 5.49 give the plot of fd curves after smoothing, plot of the functional PCA showing the total variance explained by each harmonic, the WFSR contours, the contour at level 0.02 and 0.011 and confirmatory plots based on WFSRC for the first 2 components and the original curves for gun-point and growth data respectively. From the FPCA plot, we see that the first two harmonics of gun-point data and growth data explain 79% and 94.8% respectively of the total variances. Moreover, the WFSR contour detects two clusters in each of the two functional datasets. As we did before, we used a low-level of the WFSR contour to assign each curve to the proper cluster. Clearly, most of the points have been allocated properly in one of the two clusters except some points. By using the WFSRC, we were able to assign the unassigned points to the suitable group.

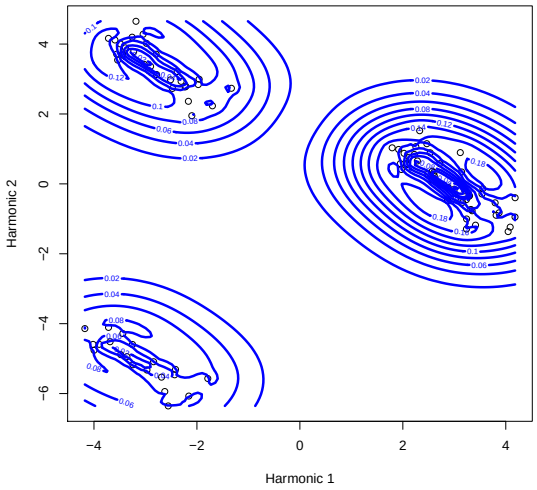
In Table 5.1 and 5.2 we give the probabilities of misclassification error based on the different functional data clustering approaches that are considered in this Chapter for the gun-point and growth datasets respectively. Some of the methods that have been discussed in Section 5.2 give different rate of misclassification error in every time we use them. This is due to the difference in the initial partitioning of the data points in every time. To address this problem and to calculate a fair version of the probability of misclassification error, we have repeated the algorithms 1000 times, and then we took the average of their



(a) Model 3 curves

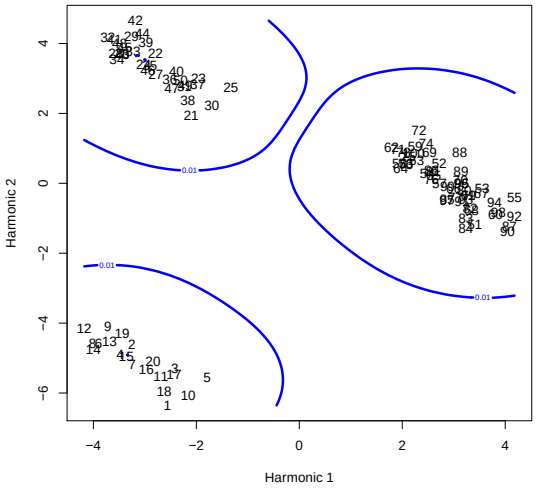


(b) Functional PCA

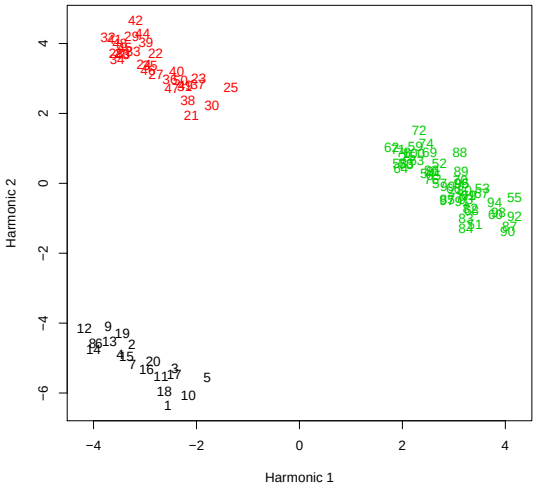


(c) WFSR contours

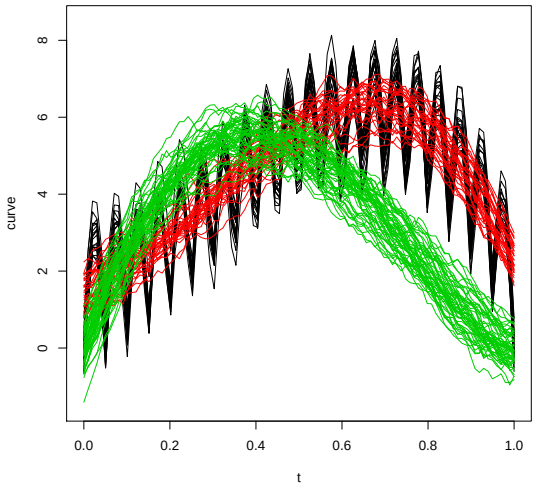
Figure 5.47: Simulated data, Model 3: (a) the observed curves with three groups, (b) plot of the functional PCA with the total variance explained by each harmonic, (c) the WFSR contours, (d) the contour at level 0.01, and confirmatory plots based on WFSRC for (e) the first 2 components and (f) the observed curves.



(d) Contour at level 0.01

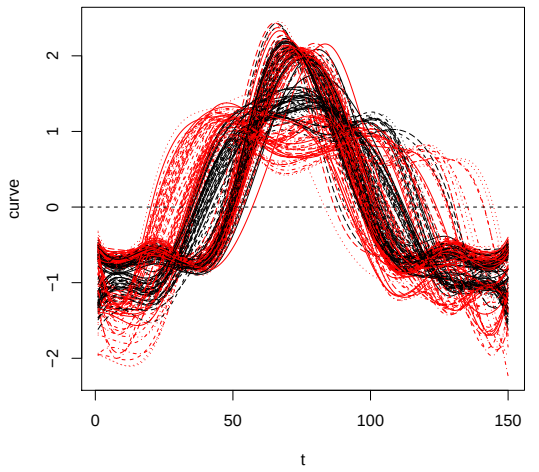


(e) Confirmatory plot: first 2 components

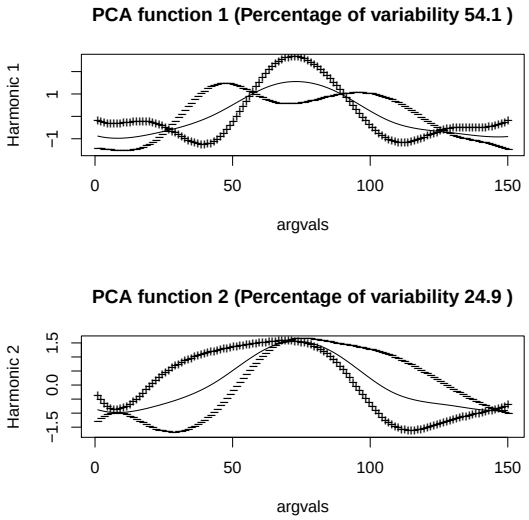


(f) Confirmatory plot: observed curves

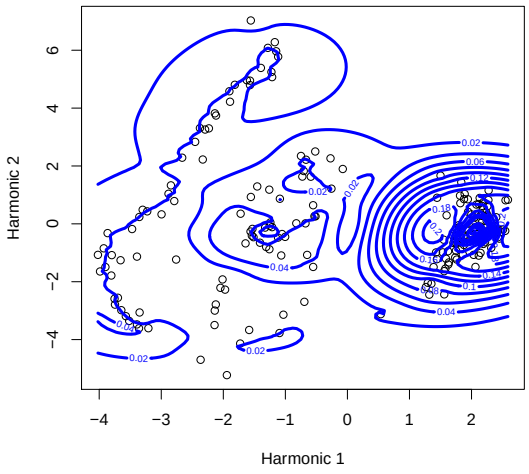
Figure 5.47: Continued.



(a) fd curves after smoothing

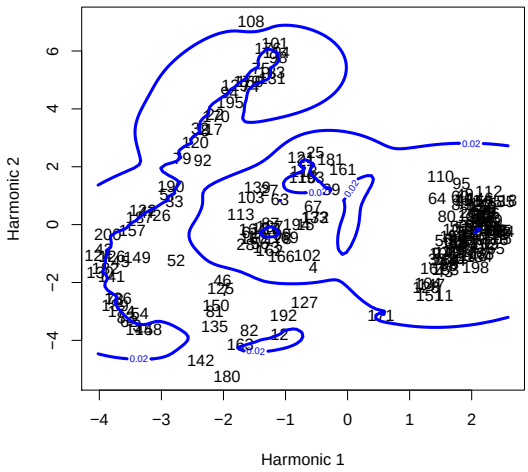


(b) Functional PCA

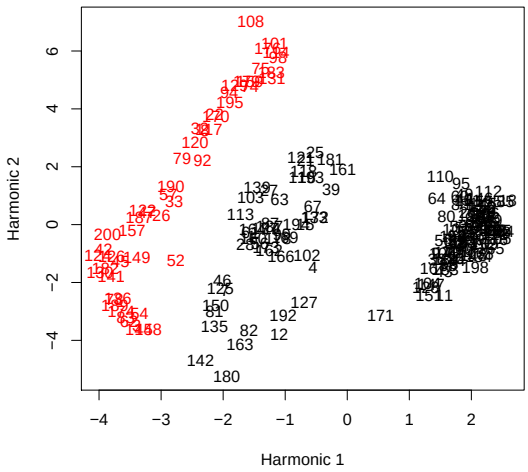


(c) WFSR contours

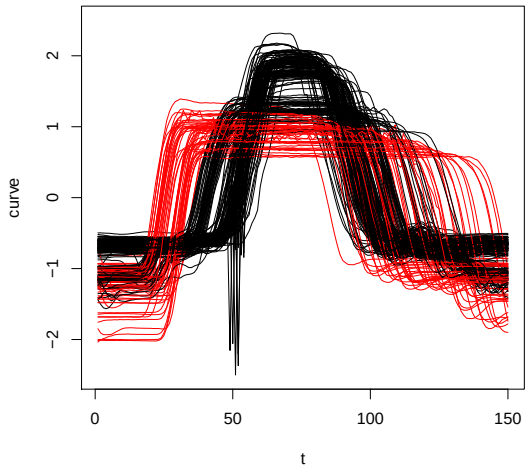
Figure 5.48: Gun-point data: (a) plot of fd curves after smoothing, (b) plot of the functional PCA with the total variance explained by each harmonic, (c) the WFSR contours, (d) the contour at level 0.02, and confirmatory plots based on WFSRC for (e) the first 2 components and (f) the observed curves.



(d) Contour at level 0.02

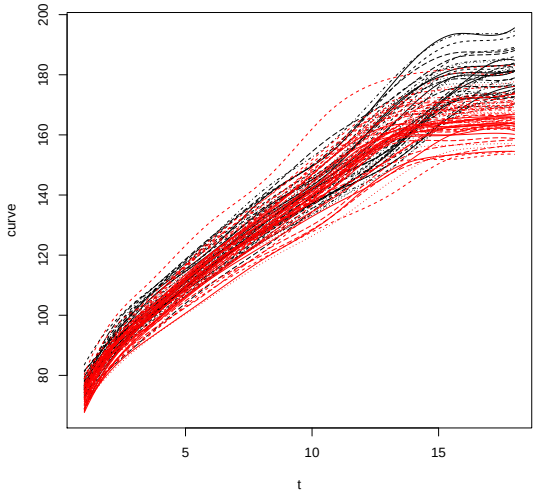


(e) Confirmatory plot: first 2 components

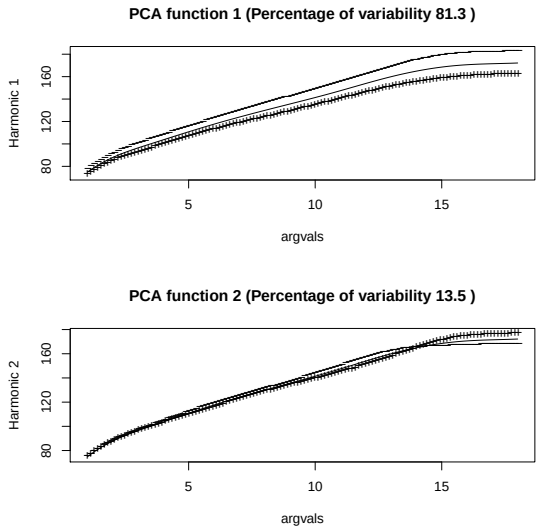


(f) Confirmatory plot: observed curves

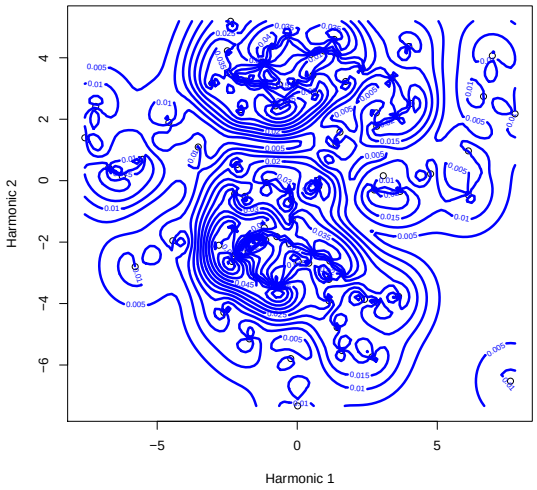
Figure 5.48: Continued.



(a) fd curves after smoothing

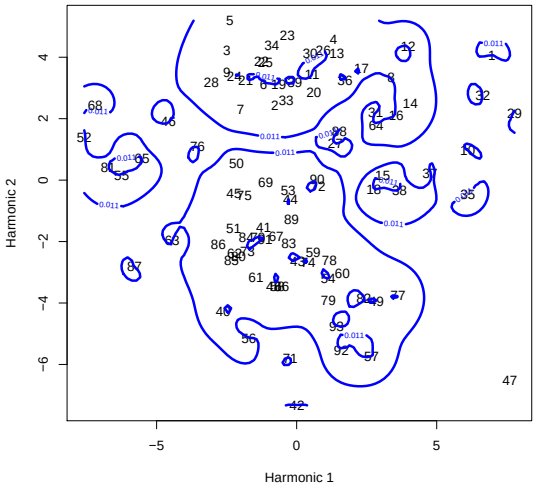


(b) Functional PCA

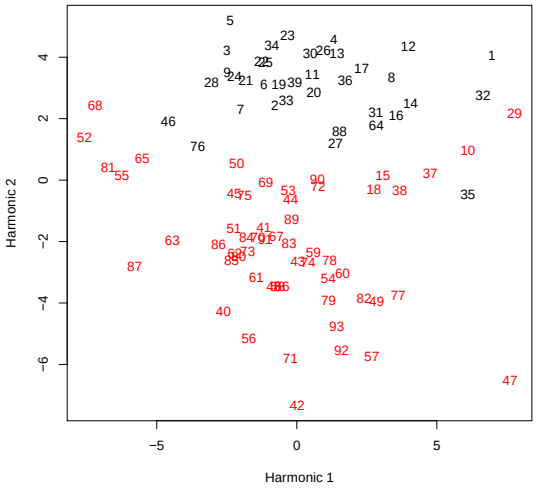


(c) WFSR contours

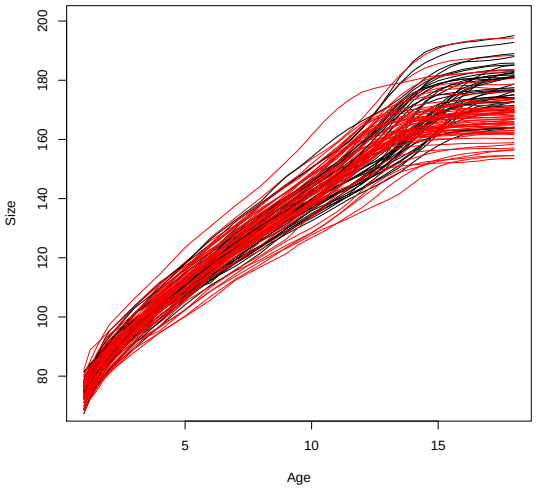
Figure 5.49: Growth data: (a) plot of fd curves after smoothing, (b) plot of the functional PCA with the total variance explained by each harmonic, (c) the WFSR contours, (d) the contour at level 0.011, and confirmatory plots based on WFSRC for (e) the first 2 components and (f) the observed curves.



(d) Contour at level 0.011



(e) Confirmatory plot: first 2 components



(f) Confirmatory plot: observed curves

Figure 5.49: Continued.

correspondence probabilities.

From Table 5.1, which gives the probabilities of misclassification error based on the different functional data clustering approaches for gun-point dataset, we can see that our proposed method, the weighted functional spatial ranks (WFSR) gives the smallest rate of misclassification error (25%) for gun-point data amongst all the other methods.

On the other hand, for the growth dataset, the weighted functional spatial ranks method gives the third smallest rate of misclassification error which is 10.75%, after PAM, CLARA and mclust algorithms. Finally, we can conclude that WFSR algorithm gives optimal and competitive results with different simulated and real functional datasets.

Table 5.1: The probabilities of misclassification error based on the different functional data clustering approaches for gun-point dataset.

Adaptive methods		Raw-data/ 2-stage methods	Raw-data methods discretized (200 instants)	Filtering methods with spline coeff. (10 splines)	Filtering methods with FPCA scores (2 components)
Funclust	0.500865	WFSR	-	-	**0.250000**
		GMM	0.500000	0.500000	0.495000
kmeans.fd	0.496275	Kmeans	0.500000	0.499980	0.500000
		hclust (single)	0.500000	0.500000	0.500000
fclust	0.500000	hclust(complete)	0.460000	0.425000	0.420000
		hclust(average)	0.440000	0.500000	0.390000
curvclust	0.500000	HDDC	0.500000	0.500280	0.499730*
		MixtPPCA	0.500000	0.530000	0.500000*,**
FunHDDC	0.575000	PAM	0.500000	0.475000	0.500000
Distance-based methods:		K-Centroids	0.500000	0.502080	0.500000
Kmeans-d_0	0.500000	Convex cclust	0.500000	0.499880	0.496095
		DIANA	0.335000	0.460000	0.425000
Kmeans-d_1	0.491000	Clara	0.500000	0.475000	0.500000

*In FPCA 3 components are considered, cannot apply for 2 components.

**Only one cluster is considered based on BIC.

Table 5.2: The probabilities of misclassification error based on the different functional data clustering approaches for growth dataset.

Adaptive methods		Raw-data/ 2-stage methods	Raw-data methods discretized (93 instants)	Filtering methods with spline coeff. (10 splines)	Filtering methods with FPCA scores (2 components)
Funclust	0.492161	WFSR	-	-	0.107527
		GMM	0.344086	0.344086	0.032258
kmeans.fd	0.498936	Kmeans	0.495323	0.4910108	0.481290
		hclust (single)	0.569893	0.569893	0.569893
fclust	0.535807	hclust(complete)	0.483871	0.333333	0.516129
		hclust(average)	0.376344	0.376344	0.569893
curvclust	0.661828	HDDC	0.516140	0.505323	0.501387*
		MixtPPCA	0.311828	0.806452	0.580645*,**
FunHDDC	0.333333	PAM	0.290323	0.215054	**0.021505**
Distance-based methods:		K-Centroids	0.496237	0.498570	0.513828
Kmeans-d_0	0.487781	Convex cclust	0.503903	0.495903	0.485430
		DIANA	0.387097	0.365591	0.107527
Kmeans-d_1	0.467419	Clara	0.333333	0.215054	**0.021505**

*In FPCA 3 components are considered, cannot apply for 2 components.

**Only one cluster is considered based on BIC.

Chapter 6

Concluding Remarks and Further Research

6.1 Concluding Remarks

Determining number of clusters in the multivariate and functional data has become one of the most important issues in very diversified areas of scientific disciplines. In this study, four different clustering methods are proposed for multivariate and functional data. We considered the data that represented by a mixture model in which each component corresponds to a different cluster without any prior knowledge of the number of clusters. Firstly, for the multivariate case, a forward search algorithm is considered in this thesis, which is based on nonparametric multivariate spatial rank functions and it is robust in terms of determining the number of clusters by the data itself. All the previous literature assumed Mahalanobis distance as the distance measure to be used in the forward search procedure. It is well known that Mahalanobis distance is invariant under all nonsingular transformations and it also performs well with the Gaussian mixture models (GMM), however, it cannot be correctly applied to asymmetric distributions and more generally to distributions, which depart from the elliptical symmetry assumptions.

According to the numerical examples that introduced in Chapter 2, it was noticed that using the forward search based on Mahalanobis distances does not give the efficient performance. In other words, the forward plots based on Mahalanobis distances did not give us reasonable results, where they did not detect the clusters in the data coming from either bivariate or trivariate Laplace and t mixture distributions with either correlated or uncorrelated variables. That means it is not suitable for the heavy tailed distributions like multivariate Laplace, Student's t, Cauchy and Log-normal distributions, and the performance is getting worse when higher dimensional data with elliptic symmetry problems are considered. Moreover, in the traditional forward search the mixture densities $f_1, \ldots, f_k$ should be from the same family of distributions.

In order to address this limitation, in this study, we proposed a new forward search methodology based on spatial ranks and volume of central rank regions (Chaudhuri, 1996; Serfling, 2002) to tackle the problem of heavy tailed mixture distributions with higher dimensional data. Using nonparametric approaches helps to get techniques which are less sensitive to the statistical model assumptions, and solve problems such as the heavy tailed distributed data with high level of correlation among the variables. For last two decades, spatial ranks are being used in analyzing multivariate data nonparametrically. They are easy to compute, but do not depend on parameter estimates of the underlying distributions, which make them robust against distributional assumptions. Koltchinskii (1997) also proved that the spatial ranks characterize a multivariate distribution. More robust results can be obtained by using the ranks instead of the original values.

The forward search plots based on spatial ranks show that the algorithm performs well under heavy tailed mixture distributions with spherical and elliptic symmetry and it outperforms the forward search based on Mahalanobis distances for non-normal mixture distributions. The modified forward search plots based on volume of central rank regions outperforms the forward search based on Mahalanobis distances and spatial ranks as illus-

trated in the numerical examples. More visually clear results have been obtained by using the volume of central rank regions, since it gives forward plots with a clearer structure of clusters. The proposed algorithms are computationally easy and straightforward. In all of numerical examples, the mixture densities $f_1, \ldots, f_k$ are from the same family of distributions. We should mention that it is not necessary to assume them to be coming from the same parametric family as we are not estimating any parameters in our proposed visual tool. This is one of the greatest advantages of the proposed method.

It is well known that spatial ranks are invariant under orthogonal transformations, but they are not invariant under general affine transformations of the data and hence the proposed algorithms are not affine invariant. To make the algorithms affine invariant, one may look for affine invariant versions of spatial ranks (see for example, (Chakraborty, 2001)) and follow the same algorithm to construct the forward search plot. To keep the simplicity of the algorithms and to save on computational time, we refrained from using affine invariant versions of spatial ranks in this work. Using affine invariant ranks may improve the results if the scales of different clusters are not similar.

In addition, we proposed another multivariate clustering method in this study. It is a new nonparametric clustering method based on different weighted spatial rank (WSR) functions. They are completely data-driven and easy to compute without any need to parameter estimates of the underlying distributions, which make them robust against distributional assumptions. The WSR is more accurate in the purpose of intuitive visualization since we can easily determine the number of clusters from the weighted ranks contours for a low-dimensional input space, using dimension reduction. The main idea behind WSR is to define a dissimilarity measure locally based on a localized version of multivariate ranks. As a result, the proposed method can be used to determine the assumed number of clusters, and to assign each observation to its cluster. Selection of a proper weight function will lead to better identification of clusters when the data do not

follow any standard parametric distribution.

We have considered parametric and nonparametric weights for comparison. We have also introduced some WSR functions based on different robust weights like Mallow weight that has been introduced by Simpson et al. (1992) and Naranjo and Hettmansperger (1994). Moreover, many other different kernel weight functions have been considered. From the numerical examples, it was noticed that more visually clear results have been obtained by using the WSR functions based on the Gaussian kernel weights. The weighted ranks contours based on Gaussian weights are more accurate and can fit better to the shape of the clusters structure. In many of the numerical examples, the weighted ranks contours based on Gaussian weights captured each observation carefully and assigned it in the true group with very small probability of misclassification error.

As an extension work of using the ordinary and weighted spatial ranks, we proposed two different clustering methods for functional data. The first method is an extension to the forward search based on spatial ranks that we proposed for the multivariate case, and it is used to identify the number of clusters in the underlying functional data. So, we can consider this method as a new raw-data method since we do not use any preprocessing functional data steps, and we do not need to perform a data registration or a dimension reduction before clustering. In the second method, we extended the WSR method that has been introduced for the multivariate case to the functional data analysis. The proposed weighted functional spatial ranks (WFSR) method is considered as one of the 2-stage methods, or the filtering methods, where it first approximate the curves into some basis functions and reduce the dimension using the functional principle components analysis (FPCA) and then perform the clustering using the basis expansion coefficients and the functional principle components scores. The WFSR method is used to determine the assumed number of clusters, and assign each curve to its cluster. Both methods are completely data-driven and easy to compute without any need to estimate parameters of

the underlying distributions, which make them robust against distributional assumptions. Different numerical examples from simulated and real data have been given in order to check the reliability of the proposed methods. Comparison between the existing methods, using the probability of misclassification error, has been considered as well. The results showed that the two proposed methods give a competitive and quite reasonable clustering analysis.

6.2 Further Research

As we mentioned before, in the forward search algorithm, when we consider data with a large number of clusters, then the forward search plots may produce too many peaks and makes it very difficult visually to determine the number of clusters and the cluster sizes. The future of our work includes study some formal procedures to estimate the number of clusters from this plot. With a formal procedure, we should able to validate the estimate against the model assumptions. One of the suggested procedures to determine the number of clusters is to find the number of peaks computationally in the forward plot based on the volume of central rank regions. This can be done by using some distance measure to calculate the distance between each two successive values of the volume of central rank regions, then we need to find the global maximum of these distances considering the sudden increase in the values of volume at the end of the search.

Another future research direction will be on proposing some nonparametric multivariate methods based on ranks to improve the traditional forward search algorithm in order to solve problems of principal components analysis, and discriminant analysis; with study the mathematical properties of the proposed methods as well as their implementation details with performing some simulations to illustrate their performances for various kinds of data.

On the other hand, we do not have any formal procedure to estimate the number of

clusters from the WSR contour plot at the moment, and we are looking into that problem as a future research project. Using some computational procedure, we may able to validate the estimate against the model assumptions and get an automated clustering technique. In addition, the future of our work includes the possible extension of the weighted spatial ranks, to consider L_p ranks in general and using L_p distance instead of the Euclidean norm. This may give better results in different type of data, which requires performing some simulations to check the performances for various kinds of data. Moreover, we may use some other criterion and methods like cross validation for the purpose of choosing the optimal value of p.

Furthermore, we consider using the spatial median, L_p median, spatial ranks and L_p ranks as cluster-analysis stopping rules in order to determine the optimal number of clusters in the multivariate data set. Determining the optimal value of p can be considered an important point in order to get the most efficient stopping-rule.

In addition, as discussed earlier, the proposed functional spatial ranks can be applied for both of regularly and irregularly sampled curves, since the functional spatial ranks are supposed to be calculated in general concept using the integrations instead of the summations quantities, then with a formal procedures and methods we can estimate the integral functions and get the estimated values of the functional spatial ranks. However, in this study we only consider the regularly sampled curves in order to keep the simplicity of the functional forward search algorithm and to save on computational time, but we are looking into the irregularly sampled curves case as a future work.

References

[1] Abraham, C., Cornillon, P. A., Matzner-Løber, E. and Molinari, N. (2003). Unsupervised curve clustering using B-splines. *Scand J Stat Theory Appl*. Volume 30, Number 3, pp. 581 - 595.

[2] Aldenderfer, M. S. and Blashfield, R. K. (1984). Cluster Analysis. Sage Publications, Inc.

[3] Araujo, A. and Giné, E. (1980). The central limit theorem for real and Banach valued random variables. New York: Wiley.

[4] Atkinson, A. C. (1993). Stalactite plots and robust estimation for the detection of multivariate outliers. In: New Directions in Statistical Data Analysis and Robustness, eds. Morgenthaler, S., Ronchetti, E. and Stahel, W. A. Basel: Birkhause.

[5] Atkinson, A. C. (1994). Fast very robust methods for the detection of multiple outliers. *Journal of the American Statistical Association*, Volume 89, pp. 1329 - 1339.

[6] Atkinson, A. C. and Mulira, H. (1993). The Stalactite Plot for the Detection of Multivariate Outliers. *Statistics and Computing*, Volume 3, pp. 27 - 35.

[7] Atkinson, A. C. and Riani, M. (2006). Distribution theory and simulations for tests of outliers in regression. *Journal of Computational and Graphical Statistics*. Volume 15, Number 2, pp. 460 - 476.

[8] Atkinson, A. C. and Riani, M. (2007). Exploratory Tools for Clustering Multivariate Data. *Computational Statistics and Data Analysis*. Volume 52, pp. 272 - 285.

[9] Atkinson, A. C. and Riani, M. (2012). Discussion on the paper by Spiegelhalter, Sherlawjohnson, Bardsley, Blunt, Wood and Grigg. *Journal of the Royal Statistical Society*. Volume 175.

[10] Atkinson, A. C., Riani, M. and Cerioli, A. (2004). Exploring Multivariate Data with the Forward Search. Springer, NewYork.

[11] Atkinson, A. C., Riani, M. and Cerioli, A. (2006). Random start forward searches with envelopes for detecting clusters in multivariate data. In: Zani, S., Cerioli, A., Riani, M., Vichi, M. (Eds.), Data Analysis, Classification and the Forward Search. Springer, Berlin, pp. 163 - 171.

[12] Atkinson, A. C., Riani, M. and Cerioli, A. (2010). The forward search: Theory and data analysis. *Journal of the Korean Statistical Society*. Volume 39, pp. 117 - 134.

[13] Azzalini, A. and Bowman, A. (1990). A look at some data on the old faithful geyser. *Journal of the Royal Statistical Society*. Volume 39, Number 3, pp. 357 - 365.

[14] Baker, F. B. and Hubert, L. J. (1975) Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association*. Volume 70, pp. 31 - 38.

[15] Ball, G. and Hall, D. (1965). A novel method of data analysis and pattern classification. Technical report NTIS AD 699616. Stanford Research Institute, Stanford, CA.

[16] Banfield, J. and Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*. Volume 49, pp. 803 - 821.

[17] Baragilly, M. and Chakraborty, B. (2016). Determining the number of clusters using multivariate ranks. In: Agostinelli, C., Basu, A., Filzmoser, P., Mukherjee, D. (Eds.), *Recent Advances in Robust Statistics: Theory and Applications*. Springer, India, Chapter 2, pp. 19 - 36.

[18] Barber, C. B., Dobkin, D. P. and Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*. Volume 22, Number 4, pp. 469 - 483.

[19] Beale, E. M. L. (1969). Euclidean cluster analysis, in Bulletin of the International Statistical Institute: *Proceedings of the 37th Session (London)*, Book 2, pp. 92 - 94. ISI, Voorburg, Netherlands.

[20] Bellman, R. (1957). Dynamic Programming. Princeton University Press.

[21] Ben-Hur, A. and Guyon, I. (2003). Detecting stable clusters using principal component analysis. *Series Methods in Molecular Biology*, Volume 224, pp. 159 - 182.

[22] Berge, L., Bouveyron, C. and Girard, S. (2012). HDclassif: An R package for model-based clustering and discriminant analysis of high-dimensional data. *Journal of Statistical Software*. Volume 46, Number 6, pp. 1 - 29.

[23] Besse, P. (1992). PCA stability and choice of dimensionality. *Stat. Probab. Lett.* Volume 13, Number 5, pp. 405 - 410.

[24] Bosq, D. (2000). Linear Processes in Function Spaces: Theory and Applications. Springer, New York.

[25] Boullé, M. (2012). Functional data clustering via piecewise constant nonparametric density estimation. *Pattern Recognition*. Volume 45, Number 12, pp. 4389 - 4401.

[26] Bouveyron, C. and Brunet, C. (2013). Model-based clustering of high-dimensional data : a review. *Computational Statistics and Data Analysis*. Volume 71, pp. 52 - 78.

[27] Bouveyron, C. and Brunet-Saumard, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics and Data Analysis*. Volume 71, pp. 52 - 78.

[28] Bouveyron, C. and Jacques, J. (2011). Model-based clustering of time series in group-specific functional subspaces. *Adv Data Anal Classif*. Volume 5, Number 4, pp. 281 - 300.

[29] Bouveyron, C., Girard, S. and Schmid, C. (2007). High dimensional data clustering. *Comput Stat Data Anal*. Volume 52, pp. 502 - 519.

[30] Bracewell, R. (2000). Heaviside's Unit Step Function: The Fourier Transform and Its Applications, 3rd ed. New York: McGraw-Hill, pp. 61 - 65.

[31] Brown, B. M. (1983). Statistical uses of the spatial median. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*. Volume 45, pp. 25 - 30.

[32] Brusco, M. J. and Kohn, H. F. (2009). Exemplar-based clustering via simulated annealing. *Psychometrika*. Volume 74, pp. 457 - 475.

[33] Calinski, R. B. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*. Volume 3, pp. 1 - 27.

[34] Cardot, H., Cenac, P. and Zitt, P. A. (2013). Efficient and fast estimation of the geometric median in Hilbert spaces with an averaged stochastic gradient algorithm. *Bernoulli*. Volume 19, pp. 18 - 43.

[35] Cardot, H., Ferraty, F. and Sarda, P. (1999). Functional linear model. *Stat. Probab. Lett*. Volume 45, Number 1, 11 - 22.

[36] Carmichael, J. W. and Sneath, P. H. A. (1969). Taxometric maps. *Systematic Zoology*. Volume 18, pp. 402 - 415.

[37] Cerioli, A. and Riani, M. (1999). The ordering of spatial data and the detection of multiple outliers. *Journal of Computational and Graphical Statistics*. Volume 8, Number 2, pp. 239 - 258.

[38] Chakraborty, A. and Chaudhuri, P. (2014). On data depth in infinite dimensional spaces. *Ann Inst Stat Math*. Volume 66, pp. 303 - 324.

[39] Chakraborty, B. (2001). On affine equivariant multivariate quantiles. *Annals of the Institute of Statistical Mathematics*. Volume 53, pp. 380 - 403.

[40] Chang, W. H., McKean, J. W., Naranjo, J. D. and Sheather, S. J. (1999). High-breakdown rank regression. *Journal of the American Statistical Association*. Volume 94, Issue 445, pp. 205 - 219.

[41] Chaouch, M. and Goga, C. (2012). Using complex surveys to estimate the L_1-median of a functional variable: application to electricity load curves. *International Statistical Review*. Volume 80, pp. 40 - 59.

[42] Chaudhuri, P. (1996). On a geometric notion of multivariate data. *Journal of the American Statistical Association*. Volume 90, pp. 862 - 872.

[43] Chen, H., Gnanadesikan, R. and Kettenring, J. R. (1974). Statistical methods for grouping corporations. *Sankhya Ser. B*. Volume 36, pp. 1 - 28.

[44] Chen, Y., Dang, X., Peng, H. and Bart. J. (2009). Outlier Detection with the Kernelized Spatial Depth Function. *IEEE Trans Pattern Anal Mach Intell*. Volume 31, Number 2, pp. 288 - 305.

[45] Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A. and Batista, G. (2015). The UCR Time Series Classification Archive. www.cs.ucr.edu/~eamonn/time_series_data/

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

[47] Chmielewski, M. A. (1981). Elliptically symmetric distributions: A review and bibliography. *International Statistical Review*. Volume 49, pp. 67 - 74.

[48] Claeskens, G., Hubert, M., Slaets, L. and Vakili, K. (2014). Multivariate functional halfspace depth. *Journal of the American Statistical Association*. Volume 109, Issue 505, pp. 411 - 423.

[49] Cook, R. D. and Hawkins, D. M. (1990). Comment on "Unmasking multivariate outliers and leverage points", by Rousseeuw, P. J. and van Zomeren, B. C. *Journal of the American Statistical Association*. Volume 85, pp. 640 - 644.

[50] Cristianini, N. and Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines and other Kernel-based Learning Methods. Cambridge University Press, Cambridge.

[51] Cuesta-Albertos, J. and Fraiman, R. (2000). Impartial trimmed k-means for functional data. *Comput Stat Data Anal*. Volume 51, pp. 4864 - 4877.

[52] Davies, D. and Bouldin, D. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Volume 1, Number 2, pp. 224 - 227.

[53] Defays, D. (1977). An efficient algorithm for a complete link method. *The Computer Journal (British Computer Society)*. Volume 20, Number 4, pp. 364 - 366.

[54] Delaigle, A. and Hall, P. (2010). Defining probability density for a distribution of random functions. *Ann Stat*. Volume 38, pp. 1171 - 1193.

[55] Different handwritten numbers image. Retrieved from: `http://scientificiv.com/projects/project_6/HDR_1.png`

[56] Ding, C. and He, X. (2004). K-means clustering via principal component analysis. *Proceedings of the twenty-first international conference on Machine learning*, pp. 29 - 36.

[57] Duda, R. O. and Hart, P. E. (1973). Pattern Classification and Scene Analysis. John Wiley and Sons, Inc., New York.

[58] Dyen, I., Kruskal, J. and Black, P. (1992). An Indoeuropean classification, a lexicostatistical experiment. *Transactions of the American Philosophical Society*. Volume 82, pp. 1 - 132.

[59] Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*. Volume 1, pp. 183 - 187.

[60] Eisen, M., Spellman, P., Brown, P. and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*. Volume 95,Number 25, pp. 14863 - 14868.

[61] Engelmann, L. and Hartigan, J. A. (1969). Percentage points on a test for clusters. *Journal of the American Statistical Association*. Volume 64, pp. 1647-1648.

[62] Epanechnikov, V. A. (1969). Non-parametric estimation of a multivariate probability density. *Theory Probab. Appl*. Volume 14, Number 1, pp. 153 - 158.

[63] Escabias, M., Aguilera, A. and Valderrama, M. (2005). Modeling environmental data by functional principal component logistic regression. *Environmetrics*. Volume 16, pp. 95 - 107.

[64] Everitt, B. S. (1977). The analysis of Contingency Tables. Chapman and Hall, London.

[65] Everitt, B. S. (1980). Cluster Analysis. 2nd ed. New York: Halsted Press.

[66] Everitt, B. S. (1993). Cluster analysis. John Wiley and Sons.

[67] Everitt, B. S., Landau, S., Leese, M. and Stahl, D. (2011). Cluster Analysis. 5th Edition, John Wiley and Sons, Inc., Chichester.

[68] Evolutionary tree image. Retrieved from: `http://www.bio.miami.edu/dana/160/160S13_5.html`

[69] Fang, K. T., Kotz, S. and Ng. K. W. (1990). Symmetric Multivariate and Related Distributions. Chapman and Hall, London.

[70] Ferguson, T. (1967). Mathematical Statistics: A Decision Theoretic Approach. Academic Press, New York.

[71] Ferraty, F. and Vieu, P. (2006). Nonparametric functional data analysis. Springer Series in Statistics. Springer, New York.

[72] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*. Volume 7, Part II, pp. 179 - 188.

[73] Florek, K., Lukaszewicz, J., Perkal, H., Steinhaus, H. and Zubrzychi, S. (1951). Sur la liaison et la division des points d'un ensemble fini. *Colloquium Mathematicum*. Volume 2, pp. 282 - 285.

[74] Fraley, C. (1998). Algorithms for model-based Gaussian hierarchical clustering. *SIAM Journal on Scientific Computing*. Volume 20, pp. 270 - 281.

[75] Fraley, C. and Raftery, A. (2003). Enhanced model-based clustering, density estimation and discriminant analysis: Mclust. *Journal of Classification*. Volume 20, pp. 263 - 286.

[76] Fraley, C. and Raftery, A. E. (1999). MCLUST: software for model-based cluster analysis. *Journal of Classification*. Volume 16, pp. 297 - 306.

[77] Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*. Volume 97 (458).

[78] Gan, G., Ma, C. and Wu, J. (2007). Data Clustering Theory, Algorithms, and Applications. ASA-SIAM Series on Statistics and Applied Probability, Philadelphia, PA: SIAM.

[79] General tree of life image. Retrieved from: `http://1mkturin.files.wordpress.com/2008/10/classes.jpg`

[80] Genton, M. G. (2001). Classes of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*. Volume 2, pp. 299 - 312.

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

[82] Gift, N., Gormley, I. C., Brennan, L. and the R Core team (2010). MetabolAnalyze: probabilistic principal components analysis for metabolomic data. R package version 1.3.

[83] Gitman, I. and Levine, M. D. (1970). An algorithm for detecting unimodal fuzzy sets and its application as a clustering technique. *IEEE Transactions on Computers*. Volume 19, Issue 7, pp. 583 - 593.

[84] Gordon, A. D. (1998). Cluster validation, in Data Science, Classification and Related Methods (eds. Hayashi, C., Ohsumi, N. and Yajima, K.),pp. 22 - 39. Springer-Verlag, Tokyo.

[85] Gower, J. C. (1966). Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*. Volume 53, pp. 325 - 338.

[86] Gower, J. C. (1967). A comparison of some methods of cluster analysis. *Biometrics*. Volume 23, pp. 623 - 628.

[87] Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics*. Volume 27, pp. 857 - 872.

[88] Gower, J. C. and Legendre, P. (1986). Metric and Euclidean properties of dissimilarity coefficients. *Journal of Classification*. Volume 5, pp. 5 - 48.

[89] Guha, P. (2012). On scale-scale curves for multivariate data based on rank regions. Ph.D. thesis, University of Birmingham.

[90] Hadi, A. S. (1992). Identifying multiple outliers in multivariate data. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*. Volume 54, pp. 761 - 771.

[91] Hadi, A. S. and Simonoff J. S. (1993). Procedures for the identification of multiple outliers in linear models. *Journal of the American Statistical Association*. Volume 88, Number 424, pp. 1264 - 1272.

[92] Hall, P. (2011). Principal component analysis for functional data: methodology, theory and discussion. In: The Oxford Handbook of Functional Data Analysis. Oxford University Press, Oxford, pp. 210 - 234.

[93] Hall, P. and Horowitz, J. L. (2007). Methodology and convergence rates for functional linear regression. *Ann. Stat.* Volume 35, Number 1, pp. 70 - 91.

[94] Hall, P. and Hosseini-Nasab, M. (2006). On properties of functional principal components analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology).* Volume 68, Number 1, pp. 109 - 126.

[95] Hall, P. and Vial, C. (2006). Assessing the finite dimensionality of functional data. *Journal of the Royal Statistical Society. Series B (Statistical Methodology).* Volume 68, Number 4, pp. 689 - 705.

[96] Hall, P., Müller, H. G. and Wang, J. L. (2006). Properties of principal component methods for functional and longitudinal data analysis. *Ann. Stat.*. Volume 34, Number 3, pp. 1493 - 1517.

[97] Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J. and Stahel W.A. (1986). Robust Statistics. The Approach Based on Influence Functions. John Wiley and Sons.

[98] Harrison, I. (1968). Cluster Analysis. *Metra.* Volume 7, pp. 513 - 528.

[99] Hartigan, J. A. (1975). Clustering Algorithms. John Wiley and Sons, Inc., New York.

[100] Hartigan, J. A. and Wong, M. A. (1979). Algorithm AS 136. A k-means clustering algorithm. *Applied Statistics.* Volume 28, pp. 100 - 108.

[101] Hartman, P. and Wintner, A. (1940). On the spherical approach to the normal distribution law. *American Journal of Mathematics.* Volume 62, pp. 759 - 779.

[102] Hawkins, D. (1980). Identification of Outliers. Chpaman and Hall. London.

[103] Hawkins, D. M. and Simonoff, J. S. (1993). AS 282 high breakdown regression and multivariate estimation. *Applied Statistics*. Volume 42, pp. 423 - 432.

[104] Heard, N., Holmes, C. and Stephens, D. (2006). A quantitative study of gene regulation involved in the immune response of anopheline mosquitoes: an application of Bayesian hierarchical clustering of curves. *Journal of the American Statistical Association*. Volume 101, Issue 473, pp. 18 - 29.

[105] Hierarchical clustering of handwritten numbers image. Retrieved from: `http://patentimages.storage.googleapis.com/WO1994009447A1/imgf000033_0001.png`

[106] Hofmann, T., Schölkopf, B. and Smola A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*. Volume 36, Number 3, pp. 1171 - 1220.

[107] Hoppner, F., Klawonn, F., Kruse, R. and Runkler, T. (1999). Fuzzy Cluster Analysis. John Wiley and Sons, Inc., Chichester.

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

[109] Ingrassia, S. and Costanzo, G. D. (2005). Functional principal component analysis of financial time series. In: *New developments in classification and data analysis. Proceedings of the meeting of the classification and data analysis group (CLADAG) of the Italian statistical society*, University of Bologna, pp. 351 - 358.

[110] Jaccard, P. (1908). Nouvelles recherches sur la distribution florale. *Bulletin de la Societe Vaudoise de Sciences Naturelles*. Volume 44, 223 - 370.

[111] Jacques, J. and Preda, C. (2013a). Funclust: a curves clustering method using functional random variable density approximation. *Neurocomputing, Elsevier*. Volume 112, pp. 164 - 171.

[112] Jacques, J. and Preda, C. (2013b). Model-based clustering for multivariate functional data. *Computational Statistics and Data Analysis*. Volume 71, pp. 92 - 106.

[113] Jacques, J. and Preda, C. (2014). Functional data clustering: a survey. *Advances in Data Analysis and Classification*. Volume 8, Issue 3, pp. 231 - 255.

[114] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*. Volume 31, pp. 651 - 666.

[115] Jain, A. K. and Dubes, R. C. (1988). Algorithms for Clustering Data. Prentice Hall.

[116] James, G. and Hastie, T. J. (2001). Functional linear discriminant analysis for irregularly sampled curves. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*. Volume 63, Number 3, pp. 533 - 550.

[117] James, G. and Sugar, C. (2003). Clustering for sparsely sampled functional data. *Journal of the American Statistical Association*. Volume 98, Issue 462, pp. 397 - 408.

[118] Johnson, R. A. and Wichern, D. W. (2007). Applied Multivariate Statistical Analysis, 6th edition. Pearson Education, Inc.

[119] Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*. Volume 32, pp. 241 - 54.

[120] Jolliffe, I. T. (2002). Principal Component Analysis. 2nd edition. Springer.

[121] Jörnsten, R. (2004). Clustering and classification based on the L_1 data depth. *Journal of Multivariate Analysis*. Volume 90, pp. 67 - 89.

[122] Jukes, T. H. and Cantor, C. (1969) Evolution of Protein Molecules. Academic Press, New York.

[123] Karhunen, K. (1947). Über lineare Methoden in der Wahrscheinlichkeitsrechnung. *Annales Academiae scientiarum Fennicae. Series A. 1, Mathematica-physica*. Issue 37, 79 (German).

[124] Kaufman, L. and Rousseeuw, P. J. (1990). Finding Groups in Data. An Introduction to Cluster Analysis. John Wiley and Sons, Inc., New York.

[125] Kaufman, L. and Rousseeuw, P. J. (2005). Finding Groups in Data. An Introduction to Cluster Analysis. Second edition. John Wiley and Sons, Inc., New York.

[126] Kaufman, L. and Rousseeuw, P.J. (1987). Clustering by means of medoids, in Statistical Data Analysis Based on the L_1ŰNorm and Related Methods, edited by Dodge, Y. North-Holland, pp. 405 - 416

[127] Kayano, M., Dozono, K. and Konishi, S. (2010). Functional cluster analysis via orthonormalized gaussian basis expansions and its application. *Journal of Classification*. Volume 27, pp. 211 - 230.

[128] King, B. (1967). Step-wise clustering procedures. *J. Am. Stat. Ass.*. Volume 62, pp. 86 - 101.

[129] Knorr, E. and Ng. R. (1998). Algorithms for mining distance-based outliers in large datasets. *In Proc. 24th Int. Conf. Very Large Data Bases, VLDB*. pp. 392 - 403.

[130] Kohn, H. F., Steinley, D. and Brusco, M. J. (2010). The p-median model as a tool for clustering psychological data. *Psychological Methods*. Volume 15, pp. 87 - 95.

[131] Koltchinskii, V. (1997). M-estimation, convexity and quantiles. *Annals of Statistics*. Volume 25, pp. 435 - 477.

[132] Lance, G. N. and Williams,W. T. (1967). A general theory of classificatory sorting strategies: 1. Hierarchical systems. *Computer Journal*. Volume 9, pp. 373 - 380.

[133] Leisch, F. (2006). A toolbox for k-Centroids cluster analysis. *Computational Statistics and Data Analysis*. Volume 51, Number 2, pp. 526 - 544.

[134] Lennington, R. K. and Flake, R. H. (1975). Statistical evaluation of a family of clustering methods. *Proceedings of the Eight International Conference of Numerical Taxonomy, W.H. Freeman and Co, San Francisco*. pp. 1 - 37.

[135] Liu, X. and Yang, M. (2009). Simultaneous curve registration and clustering for functional data. *Comput Stat Data Anal*. Volume 53, pp. 1361 - 1376.

[136] Lleti, R., Ortiz, M. C. , Sarabia, L. A. and Sánchez, M. S. (2004). Selecting variables for k-Means cluster analysis by using a genetic algorithm that optimises the silhouettes. *Analytica Chimica Acta*. Volume 515, pp. 87 - 100.

[137] Lloyd, S. P. (1957). Least square quantization in PCM. *Bell Telephone Laboratories Paper*. Published in journal much later: Lloyd., S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, Volume 28 (2), pp. 129 - 137.

[138] Lloyd, S. P. (1957). Least squares quantization in PCM. Technical Note, Bell Laboratories. Published in 1982 in *IEEE Transactions on Information Theory*. Volume 28, pp. 128 - 137.

[139] Loève, M. (1945). Fonctions aléatoires de second ordre. *C R Acad Sci Paris*, Issue 220, 469.

[140] Lord, R. D. (1974). The use of the Hankel transform in statistics. I. General theory and examples. *Biometrika*. Volume 41, pp. 44 - 55.

[141] MacNaughton-Smith, P., Williams, W. T., Dale, M. B. and Mockett, L. G. (1964). Dissimilarity analysis. *Nature*. Volume 202, pp. 1034 - 1035.

[142] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (eds. LeCam, L. and Neymen, J.)*. Volume 1, pp. 281 - 297. University of California Press, Berkeley.

[143] Mahalanobis, P. C. (1936). On the generalized distance in statistics. *Proceedings of the National Institute of Science, India*. Volume 12, pp. 49 - 55.

[144] Marriott, F. H. C. (1971). Practical problems in a method of cluster analysis. *Biometrics*. Volume 27, pp. 501 - 514.

[145] Mas, A. (2008). Local functional principal component analysis. *Complex Anal. Oper. Theory*. Volume 2, Number 1, pp. 135 - 167.

[146] Maxwell, J. C. (1860). Illustration of the dynamical theory of gases Ũ part I. On the motions and collisions of perfectly elastic bodies. *Taylor's Philosophical Magazine*. Volume 19, pp. 19 - 32.

[147] McLachlan, G. J. and Basford, K. E. (1988). Mixture Models: Inference and Applications to Clustering. Marcel Dekker, New York.

[148] McLachlan, G. J. and Peel, D. (2000). Finite Mixture Models. Wiley Interscience, New York.

[149] McQuitty, L. L. (1957). Elementary linkage analysis for isolating orthogonal and oblique types and typal relevancies. *Educational and Psychological Measurement*. Volume 17, pp. 207 - 229.

[150] Melnykov, V. and Maitra, R. (2010). Finite mixture models and model-based clustering. *Statistics Surveys* Volume 4 , pp. 80 - 116.

[151] Milligan, G. W. (1980). An examination of the effect of six types of error perturbation on fifteen clustering algorithms. *Psychometrika*. Volume 45, pp. 325 - 342.

[152] Milligan, G. W. (1981). A review of Monte Carlo tests of cluster analysis. *Multivariate Behavioral Research*. Volume 16, pp. 379 - 407.

[153] Milligan, G. W. and Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*. Volume 50, pp. 159 - 179.

[154] Mojena, R. (1977). Hierarchical grouping methods and stopping rules: an evaluation. *Computer Journal*. Volume 20, pp. 359 - 363.

[155] Morrison, D. (1967). Measurement problems in cluster analysis. *Management Science (Series B, Managerial)*. Volume 13, Number 12, pp.775 - 780.

[156] Möttönen, J. and Oja, H. (1995). Multivariate spatial sign and rank methods. *Journal of Nonparametric Statistics*. Volume 5, Issue 2, 201 - 213.

[157] Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In: *Proc. Internat. Conf. on Computer Vision Theory and Applications (VISAPP 09)*.

[158] Naranjo, J. D. and Hettmansperger, T. P. (1994). Bounded influence rank regression. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*. Volume 56, Number 1, pp. 209 - 220.

[159] Needham, R. M. (1967). Automatic classification in linguistics. *The Statistician*. Volume 17, pp. 45 - 54.

[160] Ng, A., Jordan, M. and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Proc. Advances in Neural Information Processing Systems*, pp. 849 - 856.

[161] Oja, H. (1999). Affine invariant multivariate sign and rank tests and corresponding estimates: A review. *Scandinavian Journal of Statistics*. Volume 26, Number 3, pp. 319 - 343.

[162] Oja, H. (2010). Multivariate Nonparametric Methods with R: An Approach Based on Spatial Signs and Ranks. Springer, New York.

[163] Olszewski, R. T. (2001). Generalized feature extraction for structural pattern recognition in time-series data. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA.

[164] Overall, J. E. and Magee, K. N. (1992). Replication as a rule for determining the number of clusters in hierarchical cluster analysis. *Applied Psychological Measurement*. Volume 16, pp. 119 - 128.

[165] Parker-Rhodes, A. F. and Jackson, D. M. (1969). Automatic classification in the ecology of the higher fungi, in Numerical Taxonomy (eds. Cole, A. J.). Academic Press, New York.

[166] Pearson K. (1920). Notes on the history of correlation. *Biometrika*. Volume 13, pp. 25 - 45.

[167] Pearson, K. (1894). Contribution to the mathematical theory of evolution. *Philosophical Transactions A*. Volume 185, pp. 71 - 110.

[168] Peng, J. and Müller, H. G. (2008). Distance-based clustering of sparsely observed stochastic processes, with applications to online auctions. *Ann Appl Stat*. Volume 2, Number 3, pp. 1056 - 1077.

[169] Phylogenetic tree of life image. Retrieved from: `http://en.wikipedia.org/wiki/Phylogenetic_tree`

[170] Plant tissue classification image. Retrieved from: `http://www.tutorvista.com/content/science/science-i/tissues/classification-plant-tissues.php`

[171] Ramsay, J. O. and Silverman, B. W. (2005). Functional data analysis. Second edition. Springer Series in Statistics. Springer, New York.

[172] Ratanamahatana, C. A. and Keogh, E. (2004). Making time-series classification more accurate using learned constraints. In *proceedings of SDM International conference*, pp. 11 - 22.

[173] Ray, S. and Mallick, B. (2006). Functional clustering by Bayesian wavelet methods. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*. Volume 68, Number 2, pp. 305 - 332.

[174] Reynolds, A., Richards, G., Iglesia, B. and Rayward-Smith, V. (1992). Clustering rules: A comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms*. Volume 5, pp. 475 - 504.

[175] Riani, M. and Cerioli, A. (1999). Graphical tools for the detection of multiple outliers in spatial statistics models. In (eds. Gaul, W. and LocarekJunge, H.), Classification in the Information Age, pp. 233 - 240. Berlin: Springer-Verlag.

[176] Riani, M., Atkinson, A. C. and Cerioli, A. (2009). Finding an unknown number of multivariate outliers. *Journal of the Royal Statistical Society: series B (statistical methodology)*. Volume 71, Number 2, pp. 447 - 466.

[177] Ripley, B. D. (1996). Pattern Recognition and Neural Networks. Cambridge.

[178] Rogers, D. J. and Tanimoto, T. T. (1960). A computer program for classifying plants. *Science*. Volume 132, pp. 1115 - 1118.

[179] Rossi, F., Conan-Guez, B. and GolliA, E. L. (2004). Clustering functional data with the som algorithm. In: *Proceedings of ESANN 2004*. Bruges, Belgium, pp. 305 - 312.

[180] Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics*. Volume 20, pp. 53 - 65.

[181] Rousseeuw, P. J. and Leroy, A. M. (1987). Robust Regression and Outlier Detection. New York: John Wiley.

[182] Rousseeuw, P. J. and van Zomeren, B. C. (1990). Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association*. Volume 85, pp. 633 - 639.

[183] Rubin, J. (1967). Optimal classification into groups: an approach for solving the taxonomy problem. *Journal of Theoretical Biology*. Volume 15, pp. 103 - 144.

[184] Samé, A., Chamroukhi, F., Govaert, G. and Aknin, P. (2011). Model-based clustering and segmentation of times series with changes in regime. *Adv Data Anal Classif*. Volume 5, Number 4, pp. 301 - 322.

[185] Sangalli, L., Secchi, P., Vantini, S. and Vitelli, V. (2010a). Functional clustering and alignment methods with applications. *Commun App Ind Math*. Volume 1 Issue 1, pp. 205 - 224.

[186] Sangalli, L., Secchi, P., Vantini, S. and Vitelli, V. (2010b). K-mean alignment for curve clustering. *Comput Stat Data Anal*. Volume 54 Issue 5, pp. 1219 - 1233.

[187] Saporta, G. (1981). Méthodes exploratoires d'analyse de données temporelles. *Cahiers du BURO*. pp. 37 - 38.

[188] Scott, A. J. and Symons, M. J. (1971). Clustering methods based on likelihood ratio criteria. *Biometrics*. Volume 27, pp. 387 - 397.

[189] Serban, N. and Jiang, H. (2012). Multilevel functional clustering analysis. *Biometrics*. Volume 68, Number 3, pp. 805 - 814.

[190] Serfling, R. (2002). A depth function and a scale curve based on spatial quantiles. *Statistical Data Analysis Based On the L_1-Norm and Related Methods. (eds. Dodge, Y.), Birkhaeuser*, pp. 25 - 38.

[191] Serfling, R. J. (2006b). Depth functions in nonparametric multivariate inference. *In Data Depth: Robust Multivariate Analysis, Computational Geometry and Applications (eds. Liu, R. Y., Sering, R. and Souvaine, D. L.), American Mathematical Society. DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. Volume 72, pp. 1 - 16.

[192] Serfling, R. J.(2006a). Multivariate symmetry and asymmetry. *Encyclopedia of Statistical Sciences*. Volume 8, pp. 5338 - 5345.

[193] Serflinga, R. and Wijesuriya, U. (2015). Nonparametric Description of Functional Data Using Spatial Depth. Preprint submitted to Elsevier. http://www.utdallas.edu/~serfling/papers/Serfling_and_Wijesuriya_September_13_2015.pdf

[194] Sguera C., Galeano, P. and Lillo, R. (2014). Spatial depth based classification for functional data. *Test*. Volume 23, Issue 4, pp. 725 - 750.

[195] Shang, H. L. (2014). A survey of functional principal component analysis. *Advances in Statistical Analysis*. Volume 98, Issue 2, pp. 121 - 142.

[196] Shen, H. (2009). On modeling and forecasting time series of smooth curves. *Technometrics*. Volume 51, Number 3, pp. 227 - 238.

[197] Sibson, R. (1973). SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal (British Computer Society)*. Volume 16, Number 1, pp. 30 - 34.

[198] Simpson, D. G., Ruppert, D. and Carroll, R. J. (1992). On one-step GM-estimates and stability of inferences in linear regression. *Journal of the American Statistical Association*. Volume 87, pp. 439 - 450.

[199] Slaets, L., Claeskens, G. and Hubert, M. (2012). Phase and amplitude-based clustering for functional data. *Comput Stat Data Anal*. Volume 56, Issue 7, pp. 2360 - 2374.

[200] Sneath, P. H. A. (1957). The application of computers to taxonomy. *J. Gen. Microbiol*. Volume 17, pp. 201 - 26.

[201] Sneath, P. H. A. and Sokal, R. R. (1973). Numerical Taxonomy.W. H. Freeman, San Francisco.

[202] Sokal, R. R. (1966). Numerical taxonomy. *Scientific American*, pp. 106 - 116.

[203] Sokal, R. R. and Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *Univ. Kansus Sci. Bull.* Volume 38, pp. 1409 - 1438.

[204] Sokal, R. R. and Sneath, P. H. A. (1963). Principles of Numerical Taxonomy. Freeman, London.

[205] Souza, C. R. (2010). Kernel functions for machine learning applications. `http://crsouza.blogspot.com/2010/03/kernel-functions-for-machine-learning.html`

[206] Steinhaus, H. (1957). Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci. (in French).* Volume 4, Number 12, pp. 801 - 804.

[207] Sugar, C. A. and James, G. M. (2003). Finding the number of clusters in a data set: An information theoretic approach. *Journal of the American Statistical Association.* Volume 98, pp. 750 - 763.

[208] Tajima, F. (1993). Unbiased estimation of evolutionary distance between nucleotide sequences. *Molecular Biology and Evolution.* Volume 10, pp. 677 - 688.

[209] Tarpey, T. and Kinateder, K. (2003). Clustering functional data. *Journal of Classification.* Volume 20, Number 1, pp. 93 - 114.

[210] Thorndike, R. L. (1953). Who belongs in a family? *Psychometrika.* Volume 18, pp. 267 - 276.

[211] Tibshirani, R., Walther, G. and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology).* Volume 63, Issue 2, pp. 411 - 423.

[212] Tipping, M. E. and Bishop, C. (1999). Mixtures of principal component analyzers. *Neural Comput.* Volume 11, Number 2, pp. 443 - 482.

[213] Tokushige, S., Yadohisa, H. and Inada, K. (2007). Crisp and fuzzy k-means clustering algorithms for multivariate functional data. *Comput Stat*. Volume 22, pp. 1 - 16.

[214] Tuddenham, R. and Snyder, M. (1954) Physical growth of California boys and girls from birth to eighteen years. University of California. Public Child Dev.1, pp. 188 - 364.

[215] Tukey, J. W. (1977). Exploratory Data Analysis. Addison-Wesley.

[216] Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer, New York.

[217] Venables, W. and Ripley, B. (2002). Modern Applied Statistics with S, fourth edition. Springer, New York.

[218] Wahba, G. (1990). Spline Models for Observational Data. Society for Industrial and Applied Mathematics. Philadelphia.

[219] Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*. Volume 58 , pp. 236 - 244.

[220] Wilks, S. S. (1938). The large sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*. Volume 9, Number 1, pp. 60 - 62.

[221] Wishart, D. (1969). Mode analysis, in Numerical Taxonomy (eds. Cole, A. J.) Academic Press, New York.

[222] Wolfe, J. H. (1963). Object cluster analysis of social areas. Master's thesis, University of California, Berkeley.

[223] Wolfe, J. H. (1971). A Monte Carlo Study of the Sampling Distribution of the Likelihood Ratio for Mixtures of Multinormal Distributions. Technical Bulletin, STB 72-2, Naval Personnel and Training Research Laboratory, San Diego, CA.

[224] Woodruff, D. and Rocke, D. M. (1993). Heuristic Search Algorithms for the Minimum Volume Ellipsoid. *Journal of Computational and Graphical Statistics*. Volume 2, pp. 69 - 95.

[225] Yao, F., Müller, H. G. and Wang, J. L. (2005b). Functional linear regression analysis for longitudinal data. *Ann. Stat.* Volume 33, Number 6, pp. 2873 - 2903.

[226] Zha, H., Ding, C., Gu, M., He, X. and Simon, H. (2002). Spectral relaxation for K-means clustering. *Proc. Advances in Neural Information Processing Systems*, pp. 1057 - 1064.