Universidad Carlos III de Madrid

Doctoral Thesis

CLUSTERING IN HIGH DIMENSION FOR MULTIVARIATE AND FUNCTIONAL DATA USING EXTREME KURTOSIS PROJECTIONS

A Thesis submitted by Janeth Carolina Rendón Aguirre
for the degree of Ph.D. in Mathematical Engineering

Supervised by:

Daniel Peña Sánchez de Rivera
Francisco Javier Prieto Fernández

Department of Statistics

Leganés, May 2017

Doctoral Thesis

CLUSTERING IN HIGH DIMENSION FOR MULTIVARIATE AND FUNCTIONAL DATA USING EXTREME KURTOSIS PROJECTIONS

Author: Janeth Carolina Rendón Aguirre

Advisors: Daniel Peña Sánchez de Rivera
Francisco Javier Prieto Fernández

Firma del Tribunal Calificador:

Presidente: _____

Vocal: _____

Secretario: _____

Calificación:

Leganés, de de 2017

Un día decidí partir para alcanzar lo que siempre demonimé como mi "gran sueño". Y este logro lo dedico a las personas que me apoyaron incondicionalmente y me dieron toda la motivación para seguir adelante.

Acknowledgments

I would like to express my sincerest thanks to those people who offered me their support during this stage of my life, without their help this thesis would not have concluded so successfully.

First of all, I would like to express my most heartfelt thanks to my supervisors Daniel Peña and Javier Prieto for the great confidence they have placed in me for the accomplishment of this work. I am very grateful for their demands and observations, especially for their patience and understanding in the most difficult moments. Thanks to them I have trained and matured both professionally and academically.

I would also like to thank Professor Roland Fried for accepting my visit to the statistical department of the Technical University of Dortmund, I am grateful for his hospitality and his attentions during my stay.

I would like to thank the statistical department staff for their kindness and availability at all times. In particular, I am very grateful to Professor Daniel Peña for allowing me to participate in the research project BES-2013-062929. I am also grateful to my co-workers and students, who have inspired me to value my profession more and more.

Thanks to my best friend Carlos Moreno, who during these seven years has been my great support, my joy and my accomplice. To Santiago Escudero for the wonderful moments that we lived together and for making me so happy. To Nancy and Nicolas for being like my parents in Spain. And those friends that despite the distance their remain unconditional and a great support in my life.

Finally, the most special thanks to my family, for being the fundamental pillar in my academic training and for believing in me. I am immensely grateful for all the love and unconditional support that they have shown me over the years.

Summary

Cluster analysis is a problem that consists of the analysis of the existence of clusters in a multivariate sample. This analysis is performed by algorithms that differ significantly in their notion of what constitutes a cluster and how to find them efficiently. In this thesis we are interested in large data problems and therefore we consider algorithms that use dimension reduction techniques for the identification of interesting structures in large data sets. Particularly in those algorithms that use the kurtosis coefficient to detect the clusters present in the data.

The thesis extends the work of Peña and Prieto (2001a) of identifying clusters in multivariate data using the univariate projections of the sample data on the directions that minimize and maximize the kurtosis coefficient of the projected data, and Peña et al. (2010) who used the eigenvalues of a kurtosis matrix to reduce the dimension.

This thesis has two main contributions:

First, we prove that the extreme kurtosis projections have some optimality properties for mixtures of normal distributions and we propose an algorithm to identify clusters when the data dimension and the number of clusters present in the sample are high. The good performance of the algorithm is shown through a simulations study where it is compared it with MCLUST, K-means and CLARA methods.

Second, we propose the extension of multivariate kurtosis for functional data,

and we analyze some of its properties for clustering. Additionally, we propose an algorithm based on kurtosis projections for functional data. Its good properties are compared with the results obtained by Functional Principal Components, Functional K-means and FunClust method.

The thesis is structured as follows: Chapter 1 is an introductory Chapter where we will review some theoretical concepts that will be used throughout the thesis.

In Chapter 2 we review in detail the concept of kurtosis. We study the properties of kurtosis. Give a detailed description of some algorithms proposed in the literature that use the kurtosis coefficient to detect the clusters present in the data.

In Chapter 3 we study the directions that may be interesting for the detection of several clusters in the sample and we analyze how the extreme kurtosis directions are related to these directions. In addition, we present a clustering algorithm for high-dimensional data using extreme kurtosis directions.

In Chapter 4 we introduce an extension of the multivariate kurtosis for the functional data and we analyze the properties of this measure regarding the identification of clusters. In addition, we present a clustering algorithm for functional data using extreme kurtosis directions.

We finish with some remarks and conclusions in the final Chapter.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Theoretical Foundation

Summary

The classification of observations is a basic problem that occurs in many disciplines. Classification methods can be grouped according to the statistical techniques used as parametric and non-parametric. Another useful grouping, based on the consideration of the available information used in the classification, is *supervised classification*, or *discriminant analysis*, and *unsupervised classification*, or *clustering*. In discriminant analysis we know a priori the groups whereas in clustering they are made from the data. In this work we focus on cluster analysis.

Clustering is a data analysis problem that has been extensively studied during the last decades. Its main aim is the partitioning of a data set into subsets. These groups, also called clusters, are constructed in such a way that an object in a given group should be similar, in some sense, to the rest of objects of the same group (*highly internally homogenous*), while objects in different groups should be significantly different (*highly externally heterogenous*).

In this introductory Chapter we will review some well known material that will be later used in the rest of the thesis. First, we will summarize some clustering methods divided into three categories: *partitional methods*, *model-based methods* and *hierarchical methods*. Second, we will present the most often used supervised classification method: *Fisher's linear discriminant analysis*. Finally, we will analyze some dimension reduction techniques for the identification of interesting structures in large data sets, these are: *Principal Components Analysis*, *Independent Component Analysis* and *Projection Pursuit*.

1.1 Cluster Methods

The unsupervised classification is an analytical procedure to find groups internally as homogeneous as possible. It consists in clustering a set of n objects, defined by p variables, or by a distance or dissimilarity matrix, in k groups, such that, 1) each element belongs to one and only one of the groups; 2) all the elements are classified; and 3) each group is internally homogeneous. The number of groups can be pre-set or not.

Many authors have proposed methods for cluster analysis. Traditionally they can be divided into three categories: *partitional methods*, *model-based methods* and *hierarchical methods*.

1.1.1 Partitional Methods

Given a set data $D = \{x_1, x_2, ..., x_n\}$, where $x_i \in \mathbb{R}^p, i = 1, ...n$. Partitional methods attempt to find K partitions of D, $C = C_1, C_2, ..., C_K$, $(K \leq n)$, such that the global distance between the data objects within each group is minimized.

The partitional clustering algorithms find all the clusters simultaneously as a partition of the data and do not impose a hierarchical structure, i.e, if K is the desired number of clusters, then partitional methods find all K clusters at once.

The most popular partitional methods are: *K-Means* and *K-Medoids*. Both methods represent a cluster by its center point. *K-means* uses the notion of a centroid defined by the mean or median coordinates of the elements in the group. In this case, the centroid has coordinates that may not coincide with to an actual data object. *K-medoid* uses the notion of a medoid, which is the most central data object of a group of objects. According to the definition of a medoid, it is required to be an actual data object.

K-Means Algorithm

The *K-means* method was proposed by MacQueen (1967), who obtained weak consistency results for the algorithm. Subsequently, Hartigan and Wong (1979) present a more efficient version. Pollard (1981) proved the strong consistency of the method, providing conditions that ensure the convergence of the cluster centers when sample size increases, generalizing one of Hartigan's results. Subsequently, Pollard (1982) proved its asymptotic distribution.

The main idea of the algorithm is to assign each point to the cluster whose center (also called centroid) is the nearest. The centroid is the average of all the points in the cluster, that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster. The centroid is updated iteratively until some convergence criteria are met.

The algorithm requires three user parameters. The first is the number of groups of the partition K. A common approach is to run the algorithm repeatedly with different K values and use some validation criteria to select the most appropriate value. Dubes (1987) provides guidance on this decision. The second parameter is the initial selection of cluster centers. One possible way to minimize the impact produced by this parameter is to run the algorithm several times with different initial partitions and choose the partition with the smallest squared error. The third parameter is the distance function, the most common is the Euclidean distance. Mao and Jain (1996) have used the Mahalanobis distance to obtain hyperellipsoidal clusters, but this requires a higher computational cost.

Once the initial parameters are assigned, the algorithm iteratively reassigns the observations to clusters according to a criterion of homogeneity. The most intuitive and frequently used homogeneity criterion in partitional clustering techniques is the Sum of Squares Within groups (SSW) for all variables, which is equivalent to the weighted sum of the variances of the variables in the groups.

For a sample of n items with p variables, define the sum of squares within groups by

$$SSW = \sum_{k=1}^{K} \sum_{j=1}^{p} \sum_{i=1}^{n_k} (x_{ijk} - \bar{x}_{jk})^2, \tag{1.1}$$

where x_{ijk} with $i = 1, ..., n$, is the value of the j^{th} variable in the i^{th} item belonging to the k^{th} cluster, from a total number of K clusters, with $K \leq n$, and $\bar{x}_{jk}$ is the mean of the k^{th} cluster. The criterion is written as

$$\min SSW = \min \sum_{k=1}^{K} \sum_{j=1}^{p} n_k s_{jk}^2, \tag{1.2}$$

where n_k is the number of items in group k and s_{jk}^2 is the variance of variable j in group k. The variances of the variables in the groups are a measure of heterogeneity in the classification, and by minimizing them, we obtain more homogenous groups.

The *K-means* algorithm looks for an optimal partition and consists of the following stages, see Jain and Dubes (1988):

1. Randomly select k items $\bar{x}_1, \bar{x}_2, ..., \bar{x}_K$ as the initial centroids for the K groups.

2. Calculate the Euclidean distance between each data point and the K centroids and use the criterion (1.1). Assign the data point i to the group k that minimizes the Euclidean distance.

3. For every centroid, recalculate the new centroid to the average of the points assigned to that centroid.

4. Recalculate the distance between each data point and new centroid obtained.

5. Repeat from step 3) until the centroid assignment no longer changes.

The main advantages of this algorithm are its simplicity and speed, which allows it to run on large datasets. Its disadvantage is that it does not yield the

same result with each run, since the resulting clusters depend on the initial random assignments. It maximizes inter-cluster (or minimizes intra-cluster) variance, but does not ensure that the result has a global minimum variance.

Several variants of the K-means algorithm have been reported in the literature. Two well-known variants of K-means in pattern recognition literature are Forgy, see Forgy (1965) and ISODATA, see Ball and Hall (1965). Forgy's algorithm is similar to the EM algorithm and consists of two-step major iterations that 1) reassign all the points to their nearest centroids, and 2) recompute centroids of newly assembled groups. Iterations continue until a stopping criterion is satisfied (for example, no reassignments happen), see Kogan et al. (2006). The ISODATA algorithm employs a technique of merging and splitting clusters. Typically, a cluster is split when its variance is above a pre-specified threshold, and two clusters are merged when the distance between their centroids is below another pre-specified threshold. Using this variant, it is possible to obtain the optimal partition starting from any arbitrary initial partition, provided proper threshold values are specified, see Ball and Hall (1967).

Another extension of K-means is Fuzzy c-means (FCM), proposed by Dunn (1973) and later improved by Bezdek (1981), where each point has a degree of belongingness to the clusters rather than belonging completely to just one cluster (soft clustering). Thus, points on the edge of a cluster, may be in the cluster to a lesser degree than points in the center of the cluster. For each point x we have a coefficient giving the degree of being in the k^{th} cluster $u_k(x)$. Usually, the sum of those coefficients is defined to be 1, so that $u_k(x)$ denotes a probability of belonging to a certain cluster. With the Fuzzy c-means algorithm the centroid $\bar{x}_k$ of the k^{th} cluster is the mean of all points in that cluster, weighted by their degree of belongingness to the cluster. The algorithm minimizes intra-cluster variance as well, but has the same problems as K-means: the minimum is a local minimum

and the results depend on the initial choice of weights. A good overview of fuzzy set based clustering is available in Backer (1978).

Cuesta-Albertos et al. (1997) proposed a modification of the *K-means* algorithm with emphasis on its robustness properties, called *Trimmed K-means*. This is a robust estimation technique based on removing part of the data, known as "impartial trimming". The methodology of "impartial trimming" is a way to obtain a trimmed set with the lowest possible variation. The *Trimmed K-means* consisting of the *K-mean* of the observations remaining after removing a fixed proportion of outliers observations. Cuesta-Albertos et al. (1997) proved their consistency for absolutely continuous multivariate distributions and García-Escudero et al. (1999) its asymptotic distribution.

K- Medoids Methods

The *K-medoid* method is a clustering algorithm related to the K-means algorithm. The objective of *K-medoid* clustering is to find a non-overlapping set of clusters such that each cluster has a most representative object, i.e., an object that is most centrally located with respect to some measure, such as distance. These representative objects are called *medoids* and a medoid can be defined as the object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal, i.e., it is a most centrally located point in the cluster, see Kaufman and Rousseeuw (1990).

Among many algorithms for *K-medoids* clustering, Partitioning Around Medoids (PAM) proposed by Kaufman and Rousseeuw (1990) is the most common. PAM is an iterative optimization method that combines relocation of points between clusters with renominating the points as potential medoids. It starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering.

However, finding a better medoid requires trying all points that are currently not medoids and this is computationally expensive. Thus, PAM has the drawback that it works inefficiently for a large data set due to its time complexity, see Han et al. (2001).

Kaufman and Rousseeuw (1990) also proposed an algorithm called CLARA (Clustering LARge Applications), is an adaptation of PAM for handling larger data sets. Instead of finding representative objects for the whole data set, CLARA, firstly, draws a sample of the data set by using random sampling method; and then, applies PAM on the sample to find the medoids of the sample. The point is that, if the sample drawn in a random way correctly represents the total data set, then the sample's medoids would approximate to the medoids of the whole data set. To come up with better approximations, CLARA draws multiple samples and gives the best clustering as the output. But the performance of CLARA drops rapidly with increasing number of clusters. Lucasius et al. (1993) proposed a new approach of K-medoid clustering using a genetic algorithm, whose performance is reported as better than CLARA but computational burden increases as the number of clusters increases. Wei et al. (2003) also compared performance of CLARA and some other variants for large data sets.

Ng and Han (1994) proposed an efficient algorithm based on a mixture of PAM and CLARA, CLARANS (Clustering LARge Applications based on RANdomized Search), this algorithm draws a sample with some randomness in each stage until it finds a better configuration. The difference between CLARANS and CLARA is that CLARANS works with all data objects, while CLARA only works with part of the complete data set.

Most of these algorithms are based on PAM, so the computational load remains a problem. Park and Jun (2009) proposed a new K-medoids clustering algorithm that calculates the distance matrix once and uses it for finding new medoids at

every iterative step. This method has better performance than K-means clustering and it requires shorter computation times than PAM. It can also be seen that the initial medoids selection employed in this method performs quite well when compared with other methods based on naively selecting initial medoids.

1.1.2 Model-Based Methods

In model-based clustering, it is assumed that the data are generated by a mixture of probability distributions in which each component represents a different cluster. Given observations $x = (x_1, ..., x_n)$, let $f_k(x_i|\theta_k)$ be the density of an observation x_i from the k^{th} component, where θ_k are the corresponding parameters, and let K be the number of components in the mixture. The model for clustering is usually formulated as *a mixture likelihood approach*, that is, we want to maximize the likelihood function

$$\mathcal{L}(\theta|x) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f_k(x_i|\theta_k), \tag{1.3}$$

where π_k is the probability that an observation belongs to the k^{th} component ($\pi_k \geq 0$; $\sum_{k=1}^{K} \pi_k = 1$). The support function of the sample is

$$\mathcal{L}(\theta|x) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k f_k(x_i|\theta_k) \right) \tag{1.4}$$

We are mainly interested in the case where $f_k(x_i|\theta_k)$ is multivariate normal (Gaussian). Then, the parameters θ_k consist of a mean vector μ_k and a covariance matrix Σ_k, and the density has the form

$$f_k(x_i|\mu_k, \Sigma_k) = \frac{\exp\{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1}(x_i - \mu_k)\}}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \tag{1.5}$$

where x represents the data, and k is an integer subscript specifying a particular cluster. Substituting these densities in (1.4), the likelihood will be

$$\mathcal{L}(\theta|x) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k \frac{\exp\{-\frac{1}{2}(x_i - \mu_k)^T \Sigma_k^{-1}(x_i - \mu_k)\}}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \right) \tag{1.6}$$

Introducing the restriction $\sum_{k=1}^{K} \pi_k = 1$ with a Lagrange multiplier in (1.4), the function to be maximized is

$$\mathcal{L}(\theta|x) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} \pi_k f_k(x_i|\theta_k) \right) - \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) \tag{1.7}$$

Equating to zero the derivative of this function with respect to the probabilities and multiplying by π_k, we obtain

$$\lambda \pi_k = \sum_{i=1}^{n} \pi_{ik} \tag{1.8}$$

where π_{ik} is defined as:

$$\pi_{ik} = \frac{\pi_k f_k(x_i|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k f_k(x_i|\mu_k, \Sigma_k)}, \tag{1.9}$$

which is the so-called probability *a posteriori*. This is the probability that, once observed, the data x_i have been generated by the normal $f_k(x_i|\mu_k, \Sigma_k)$.

Banfield and Raftery (1993) proposed a model-based framework for clustering in multivariate normal mixtures by parameterizing the covariance matrix in terms of its eigenvalue decomposition in the form

$$\Sigma_k = \lambda_k D_k A_k D_k^T, \tag{1.10}$$

where D_k is the orthogonal matrix of eigenvectors of Σ_k, A_k is a diagonal matrix whose elements are proportional to the eigenvalues of Σ_k, and λ_k is an associated constant of proportionality. The orientation of the principal components of Σ_k is determined by D_k, while A_k determines the shape of the density contours; λ_k specifies the volume of the corresponding ellipsoid, which is proportional to $\lambda_k^p |A_k|$, where p is the data dimension. Conventions for normalizing λ_k and A_k include requiring $|A_k| = 1$, so that $\lambda_k = |\Sigma_k|^{1/p}$, see Celeux and Govaert (1995). Or requiring $\max(A_k) = 1$, so that λ_k is the largest eigenvalue of Σ_k, see Banfield and Raftery (1993). This approach is particularly useful for two and three dimensional

data, where the geometric features can be identified visually. It may also be applicable for higher dimensional data when multivariate visualization analysis reveals some structure. Fraley (1999) developed efficient algorithms for hierarchical clustering with the various parametrizations (1.10) of Gaussian mixture models.

EM Algorithm

The estimation of (1.6) is performed using the Expectation-Maximization (EM) algorithm. It is a general approach to maximum likelihood in the presence of incomplete data. In EM for clustering, the "complete" data are considered to be $y_i = (x_i, z_i)$, where $z_i = (z_{i1}, ..., z_{iK})$ with

$$z_{ik} = \begin{cases} 1 & \text{if} \quad x_i \quad \text{belongs to group} \quad k \\ 0 & \text{otherwise} \end{cases} \tag{1.11}$$

constitutes the "missing" data. The model assumptions are that the density of an observation x_i given z_i is given by $\prod_{k=1}^{K} f_k(x_i|\theta_k)^{z_{ik}}$ and that each z_i is independent and identically distributed according to a multinomial distribution of one draw on K categories with probabilities $\pi_1, ..., \pi_K$. The resulting complete-data loglikelihood is

$$\mathcal{L}(\theta|x) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik}[\log \pi_k f_k(x_i|\theta_k)] \tag{1.12}$$

The quantity $\hat{z}_{ik} = \mathbb{E}[z_{ik}|x_i, \theta_1, ..., \theta_K]$ for model (1.12) is the conditional expectation of z_{ik} given the observation x_i and parameter values.

The EM algorithm iterates between an E-step in which values of $\hat{z}_{ik}$ are computed from the data with the current parameter estimates and an M-step in which the complete-data loglikelihood (1.12), with each z_{ik} replaced by its current conditional expectation $\hat{z}_{ik}$, is maximized with respect to the parameters. The EM algorithm is as follows:

M-step: compute maximum-likelihood parameter estimates given $\hat{z}_{ik}$

$$\begin{aligned} n_k &\longleftarrow \sum_{i=1}^{n} \hat{z}_{ik} \\ \hat{\pi}_k &\longleftarrow \frac{n_k}{n} \\ \hat{\mu}_k &\longleftarrow \frac{1}{n_k} \sum_{i=1}^{n} \hat{z}_{ik} x_i \\ \hat{\Sigma}_k &: \text{ depends on the model} \end{aligned}$$

E-step: compute $\hat{z}_{ik}$ given the parameter estimates from the M-step

$$\hat{z}_{ik} \longleftarrow \frac{\hat{\pi}_k f_k(x_i|\hat{\mu}_k, \hat{\Sigma}_k)}{\sum_{j=1}^{K} \hat{\pi}_j f_j(x_j|\hat{\mu}_j, \hat{\Sigma}_j)},$$

where f_k has the form (1.5).

Celeux and Govaert (1995) detail both the E- and M-steps for the case of multivariate normal mixture models parametrized via the eigenvalue decomposition in (1.10).

Application to Cluster Analysis

Different implementations of mixtures of normal distributions have been proposed for solving problems of clusters. Fraley and Raftery (1999) have designed a method based on mixtures of normal distributions and an algorithm MCLUST, which works well in practice. MCLUST is a contributed R package for model-based clustering, classification, and density estimation based on finite normal mixture modeling. It provides functions for parameter estimation via the EM algorithm for normal mixture models.

The algorithm assumes the sample has been generated from a mixture of K normal distributions and estimates the parameters of each population of the mixture together with the probability of membership for each observation of the sample, which is the probability *a posteriori* (1.9). The observation x_i will be

assigned, to the cluster k that maximizes π_{ik}. In order to compute (1.9) we need to estimate the parameters of the mixture, which is done via the logarithm of the correspondent likelihood function. The estimation is repeated for different assumptions on the number of components in the mixture and covariance matrices of the components, and the Bayesian Information Criterion (BIC) is used to choose the assumption more likely to be true. This allows comparison of models with differing parameterizations and/or differing numbers of clusters. In general the larger the value of the BIC, the stronger the evidence for the model and number of clusters. A standard convention for calibrating BIC differences is that differences of less than 2 correspond to weak evidence, differences between 2 and 6 to positive evidence, differences between 6 and 10 to strong evidence, and differences greater than 10 to very strong evidence, see Kass and Raftery (1995).

MCLUST provides the Mclust function, which aim to provide the optimal mixture model estimation according to BIC criteria. The input to function Mclust includes the data, the number of mixture components (clusters) for which the BIC is to be calculated and the covariance structures to consider. By default, Mclust compares BIC values for parameters optimized for up to nine components and all ten covariance structures currently available in the mclust software. The output includes the parameters of the maximum-BIC model (where the maximum is taken over all of the models and numbers of components considered), and the corresponding classification and uncertainty. The object produced by Mclust is a list with a components describing the estimated model. See Fraley and Raftery (2012).

Overall, MCLUST works well for low dimensional spaces. However, when the dimension of space is large, the computational time may become very expensive; MCLUST estimates several covariance matrices, and thus requires a large sample if the dimension of the data is large.

Detailed descriptions and numerous references regarding model-based clustering can be found in Dempster et al. (1977) and McLachlan and Krishnan (1997). Examples of this type include, SNOB (Wallace and Dowe, 1994), AUTOCLASS (Cheeseman and Stutz, 1996), COBWEB, CLASSIT (Chiu et al., 2001) and CLUSTER/2 (Michalski and Stepp, 1983).

1.1.3 Hierarchical Methods

Hierarchical methods are based on a matrix of distances or similarities between the elements of the sample and create a hierarchy based on a distance between the groups constructed from observations. If all variables are continuous, the most used distance is the Euclidean distance between standardized variables. Hierarchical algorithms build a series of nested partitions. These partitions can be obtained from agglomerative shape, in this case the clusters are joined together to form partitions with fewer clusters, or by division, when the groups are split so that the partitions produce more clusters. The main advantage of such algorithms is that they are able to capture the possible hierarchical data structure. A possible disadvantage is that they are appropriate only if the sample size is small. See Jain and Dubes (1988) and Kaufman and Rousseeuw (1990).

The graphical representation of a hierarchy of groups is commonly done using a tree called *dendrogram*. The leaf nodes represent the first partition on a agglomerative process (or the last on a divisive process), while internal nodes represent the union of several groups in an agglomerative process (or the division of a group on a divisive process). The height of each node generally corresponds to the distance between their child nodes. In this way, each partition of the hierarchy can be indicated by a horizontal line that cuts the different branches of the tree.

There are different approaches for computing distances among the groups. The most important are Single-link (S-link) (Sneath and Sokal,1973), Complete-link

(Com-link) (King, 1967) and Average-link (Ave-link) (Jain and Dubes, 1988). The Single-link distance between two subsets is the shortest distance between them, the Complete-link is the largest distance and Average link the average distance. It has been observed that the complete- link algorithm produces more useful hierarchies in many applications than the single-link algorithm, see Jain and Dubes (1988).

An algorithm suitable for large-scale clustering introduced by Guha et al. (1998) is CURE (Clustering Using REpresentatives). It takes random samples to cluster each sample separately and integrates the results in a final step. The algorithm ROCK, developed by the same researchers, Guha et al. (1999), is an improvement of CURE for dealing with enumeration data, which takes the effect on the similarity from the data around the cluster into consideration. Karypis et al. (1999) propose the CHAMELEON algorithm, which is composed of two phases: at first, it partitions the original data into sub-clusters with a smaller size based on the K-nearest neighbour graph, and then the clusters with small size are merged into a cluster with bigger size, based on an agglomerative algorithm, until the final clusters are obtained. The algorithm seems to find clusters of diverse shapes, densities, and sizes in two-dimensional space, see Song et al. (2011). Steinbach et al. (2000) proposed a hierarchical divisive version of K-means, called bisecting K-means, that recursively partitions the data into two clusters at each step.

1.2 Supervised Classification

The supervised classification, also called *discriminant analysis*, is a classification technique based on knowing the characteristics that differentiate (discriminate) between two or more groups. It is used to assign new observations to known groups.

Discriminant Analysis can be considered as a regression analysis where the

dependent variable is categorical and has as categories the group memberships labels, and the independent variables are continuous. There are several possible approaches to this problem, within the most important supervised classification techniques are: Logic based algorithms, Artificial Neural Networks (ANNs), Radial Basis Function (RBF), Naive Bayesian networks (NB), Bayesian Network (BN), Linear Discriminant Analysis (LDA), k-Nearest Neighbour (k-NN) and Support Vector Machines (SVMs).

In this paper we are particularly interested in the classical discriminant analysis developed by Fisher (1936) for its relation to kurtosis. We will make a brief description below

1.2.1 Fisher's Linear Discriminant

Linear discriminant analysis (LDA), also known as Fisher's linear discriminant analysis, is a widely used method aimed at finding linear combinations of observed features which best characterize or separate two or more classes of objects or events.

Fisher's linear function provide a rule for assigning a unit, whose group membership is unknown, to one out of the K known groups. As a particular case, the classical discrimination is for $K = 2$. Let's consider that the general matrix of data X, $n \times p$ (n individuals and p variables), is divided into K matrices corresponding to the subpopulations. We will call x_{ijk} to the elements of these submatrices where i represents the individual, j the variable, and k the group. We let n_k be the number of elements in group k and the total number of observations is: $n = \sum_{k=1}^{K} n_k$.

Fisher suggested to look for the linear combination $z = a^T x$, which best separates the groups. This amounts to look for the vector a such that, the projection of the data on this direction makes the groups as separated as possible.

The vector of means within each group, $\bar{x}_k = \frac{1}{n_k}\sum_{i=1}^{n_k} x_{ik}$, is a column vector of dimension p which contains the p means for the observations of the group k. The covariance matrix for the elements of group k is

$$\Sigma_k = \frac{1}{n_k - 1}\sum_{i=1}^{n_k}(x_{ik} - \bar{x}_k)(x_{ik} - \bar{x}_k)^T \tag{1.13}$$

The variable z will therefore have overall average $\bar{z} = a^T\bar{x}$, where $\bar{x} = \frac{1}{n}\sum_{k=1}^{K}\bar{x}_k n_k$.

For each group, the average value is $\bar{z}_k = a^T\bar{x}_k$ and $Var(z_k) = a^T\Sigma_k a$.

Therefore, $Var(z) = a^T W a$, where

$$W = \frac{1}{n - K}\sum_{k=1}^{K}(n_k - 1)\Sigma_k \tag{1.14}$$

is the *within group covariance matrix.*

The variability among the group means projected is given by $a^T B a$, where

$$B = \frac{1}{K - 1}\sum_{k=1}^{K} n_k(\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^T, \tag{1.15}$$

is the *between groups covariance matrix.* In the simple two group case the variance between groups has the simple expression $B = \sum_{k=1}^{2} n_k(\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^T$. After writing $\bar{x}$ it becomes

$$B = \frac{n_1 n_2}{n_1 + n_2}(\bar{x}_1 - \bar{x}_2)(\bar{x}_1 - \bar{x}_2)^T, \tag{1.16}$$

The criterion proposed by Fisher is to maximize:

$$\phi = \frac{a^T B a}{a^T W a} \tag{1.17}$$

In order to find the vector a for which ϕ is maximum we derive it with respect to a, we set the derivatives to 0, and we obtain

$$(W^{-1}B - \phi I)a = 0, \tag{1.18}$$

which implies that ϕ is an eigenvalue of $W^{-1}B$ and a is the corresponding eigenvector. As ϕ is the function we want to maximize we choose the largest eigenvalue and the corresponding eigenvector as the best discriminant direction. In the two-group case, the single discriminant direction a can be obtained as $a = W^{-1}(\bar{x}_1 - \bar{x}_2)$. The corresponding linear combination will therefore be $z = a^T x = (\bar{x}_1 - \bar{x}_2)^T W^{-1} x$. Let's denote by $\bar{z}_1$ the average projection of group 1 on a, $\bar{z}_1 = a^T \bar{x}_1 = (\bar{x}_1 - \bar{x}_2)^T W^{-1} \bar{x}_1$, and by $\bar{z}_2 = (\bar{x}_1 - \bar{x}_2)^T W^{-1} \bar{x}_2$, the average projection of group 2 on a. Let's also assume, without loss of generality, that $\bar{z}_1 > \bar{z}_2$. Let x_0 be the new unit we want to classify and $z_0 = a^T x_0 = (\bar{x}_1 - \bar{x}_2)^T W^{-1} x_0$ its projection on a. A natural allocation rule will consist in assigning x_0 to the group whose average it is closest to along a: i.e. assign x_0 to group 1 if $|z_0 - \bar{z}_1| < |z_0 - \bar{z}_2|$ and to group 2 if the opposite inequality holds. This amounts to assign x_0 to K_1 if $z_0 > \frac{\bar{z}_1 + \bar{z}_2}{2}$, that is, if:

$$(\bar{x}_1 - \bar{x}_2)^T W^{-1} x_0 > \frac{1}{2}(\bar{x}_1 - \bar{x}_2)^T W^{-1}(\bar{x}_1 + \bar{x}_2).$$

This is known as a linear classification rule as it is a linear function of the observed vector variable x.

1.3 Clustering for Large Data Sets

A central problem in high dimensional data analysis is to find a small set of features that summarize the most significant aspects of their behavior.

High dimensionality presents two problems in clustering, see Berkhin (2006). First, the presence of irrelevant attributes, because they negatively affect proximity measures. Second, the dimensionality curse, that is a lack of data separation in high dimensional space. In order to solve this problem, two main approaches have been used. The first one is variable selection and the second one dimension reduction.

Variable selection for clustering consists of reducing the number of variables and focusing only on the relevant variables that carry discriminating information for clusters. This is an important feature since fewer variables could give a better partitioning of the data into clusters closer to the true clustering structure and could also greatly facilitate the interpretation of results. Variable selection can be made by some penalty function, as the Lasso method. For instance in model based clustering we can maximize the likelihood of the mixture of normals adding some penalty function in order to introduce variable selection, (see Pan and Shen (2007) and Wang and Zhu (2008)).

Also we can consider the problem of variable select for model-based clustering, as proposed by Raftery and Dean (2006). They proposed that, if we have a dataset Y, it can be partitioned into three sets of variables: variables included in the model ($Y^{(1)}$), variables currently under consideration ($Y^{(2)}$), and remaining variables ($Y^{(3)}$). The decision for inclusion or exclusion of ($Y^{(2)}$) from the set of clustering variables is taken after considering two models. In the formulation of the Model 1, it is assumed that the inclusion of the variables ($Y^{(2)}$) does not contribute to the model improvement as the clustering information is already contained in the set of already included variables ($Y^{(1)}$). Model 2 implies that ($Y^{(2)}$) does provide additional information about clustering membership, after ($Y^{(1)}$) has been observed. The models can be compared through an approximation to the Bayes factor that can be estimated by means of BIC. This variable selection technique is available through the R package "clustvarsel", see Dean et.al (2013). However the clustvarsel package can be very slow in high-dimensions.

Maugis et al. (2009) proposed a related approach more versatile which describes three possible roles for each variable: The relevant clustering variables, the irrelevant clustering variables dependent on a part of the relevant clustering variables and the irrelevant clustering variables totally independent of all the

relevant variables. They proposed a model selection criterion and a variable selection algorithm for this new modeling of variable roles. Finally, Maugis et al. (2102) extended this approach, denoted "selvarclust", by adding capabilities for handling missing values.

In addition to these procedures, other variable selection approaches are due to Steinley and Brusco (2008), who introduce measures of the capability of each variable to detect a fixed number of clusters; and to Fraiman et al. (2008), who propose a method to detect the noninformative variables in clustering. Witten and Tibshirani (2010) developped a cluster algorithm that can be applied to obtain sparse versions of K-means and hierarchical clustering. Some comparison of these methods and other related references can be found in Galimberti et al. (2017) and for a more comprehensive review of model-based clustering of high-dimensional data see Bouveyron and Brunet (2014).

The second approach is dimensionality reduction methods, where we try to identify some relevant subspace which include the relevant information for clustering. In a high-dimensional space, clustering algorithms that are based on the distance measure lose their efficiency and accuracy because the distance of a point to its nearest neighbour approaches the distance to its farthest neighbour as dimensionality increases, see Beyer et al. (1999). In order to solve this problem, dimensionality reduction methods have been proposed for cluster analysis, since in addition to reducing the computational cost, they provide a clearer image of the data. However, these methods inevitably cause some loss of information and can damage the interpretability of results, even distort the actual clusters.

In this paper we will be particularly interested in the reduction of the dimension for the identification of structures of interest in the data. Our approach is based in the reduction of the dimension of the sample by projecting the data onto a subspace of smaller dimension; in this subspace, if the group structure of the original sample

is kept, it should be easier the identify clusters.

Different techniques in multivariate analysis have been designed to reduce the dimensionality of the data and to help derive a simple description of a data set. Most of them proceed by defining a small number of new variables that summarize the information contained in the original ones. These techniques are divided in: attributes or variables transformations and domain decomposition.

Attribute transformations are simple functions of the existent variables or attributes. The most popular techniques for dimensionality reduction in this category, based on the covariance matrix of the variable, are Principal Components Analysis (PCA) and Singular Value Decomposition (SVD). Other methods, like projection pursuit and Independent Component Analysis (ICA) are more appropriate for non-Gaussian distributions since they do not rely on the second-order property of the data.

Domain decomposition consists in solving a global problem defined on a domain by iterative and independent resolution of subproblems defined in smaller subdomains. Based on this approach, McCallum et al. (2000) proposed a technique for clustering called "Canopy Clustering" for the following situation: high-dimension, big data and many clusters. The technique is performed in two stages: 1) canopy generation and 2) clustering. The first stage consists in divide the data into some number of overlapping subsets, called "canopies". A canopy is a subset of the data set that are within some distance threshold from a central point. An element may appear under more than one canopy and every element must appear in at least one canopy. The elements that not appearing in any common canopy are far enough apart that they could not possibly be in the same cluster. In the second stage, some clustering algorithm, such as K-means or Expectation-Maximization, is executed using the accurate distance measure only between the points that occur in a common canopy. A difference between this

method with other clustering methods is that this technique forms overlapping regions, thus it is tolerant to inaccuracies in the distance measure used to create the canopies because the canopies may overlap with each other.

We will describe some attributes transformations techniques that are of particular interest in our work.

1.3.1 Principal Component Analysis

Principal component analysis (PCA) was developed initially by Pearson (1901) and subsequently it was studied by Hotelling (1930). It is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance, that is, accounts for as much of the variability in the data as possible, and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to the preceding components, i.e., they are uncorrelated. Principal components are guaranteed to be independent only if the data set is jointly normally distributed.

Computing the Principal Components

Suppose that we have the values of p-variables in n elements of a population in a matrix X of dimensions $n \times p$, where the columns contain the variables and the rows the elements. Suppose that the variables of matrix X have zero mean and the covariance matrix is given by $S = \frac{1}{n}X^T X$. If we consider point $x_i = (x_{i1}, ..., x_{ip})$ we want to find a new set of variables z_i where $i = 1, ..., p$, uncorrelated each other, whose variances will decrease progressively. Each z_i is a linear combination of x_i,

that is:

$$z_i = a_{i1}x_{i1} + ... + a_{ip}x_{ip} = a_i^T x_i, \tag{1.19}$$

where $a_i^T = (a_{i1}, ..., a_{ip})^T$ is a constants vector of unit norm. The vector which represents the projection of point x_i on the direction a_i is $z_i a_i$.

The first principal component is calculated by choosing a_1 so that z_1 has maximum variance. The values of this first component of the n individuals will be represented by a vector z_1, given by

$$z_1 = Xa_1 \tag{1.20}$$

Since the original variables have a zero mean, z_1 will also have a zero mean. Its variance will be:

$$Var(z_1) = \frac{1}{n} z_1^T z_1 = \frac{1}{n} a_1^T X^T X a_1 = a_1^T S a_1 \tag{1.21}$$

We want to choose a_1 so that the maximization of (2.3) has a solution, subject to the constraint $a_1^T a_1 = 1$. Introducing this restriction using the Lagrange multiplier, the function to be maximized is

$$\mathcal{L}(a_1) = a_1^T S a_1 - \lambda(a_1^T a_1 - 1) \tag{1.22}$$

Taking the derivative with respect to the components of a_1 and setting the result to zero, the solution is:

$$Sa_1 = \lambda a_1 \tag{1.23}$$

which implies that a_1 is an eigenvector of the matrix S, and λ is its corresponding eigenvalue. From (2.3) we have

$$Var(z_1) = a_1^T S a_1 = a_1^T \lambda a_1 = \lambda a_1^T a_1 = \lambda \tag{1.24}$$

and we conclude that λ is the variance of z_1. As this is the quantity that we wish to maximize, λ will be the largest eigenvalue of the matrix S. Its associated vector, a_1, defines the coefficients of each variable in the first principal component.

The second principal component, $z_2 = Xa_2$, is obtained by a similar argument. In addition, we want z_2 is uncorrelated with the previous component z_1, that is, $Cov(z_2, z_1) = 0$. Then we have to

$$Cov(z_2, z_1) = a_2^T S a_1 = 0 \tag{1.25}$$

From (1.23) and (1.25), we have that $a_2^T S a_1 = \lambda a_2^T a_1 = 0$, this is equivalent to $a_2^T a_1 = 0$, that is, the vectors are orthogonal. Thus, we have to maximize the variance of z_2, $a_2^T S a_2$, subject to the following constraints: $a_2^T a_2 = 1$ and $a_2^T a_1 = 0$. Using the Lagrange multiplier we have the function

$$\mathcal{L}(a_2) = a_2^T S a_2 - \lambda(a_2^T a_2 - 1) - \delta a_2^T a_1 \tag{1.26}$$

Equating to zero the derivative with respect to the components of a_2, we obtain the solution $Sa_2 = \lambda a_2$. Using the same reasoning as above, we choose λ as the second largest eigenvalue of the matrix S, and a_2 is its corresponding eigenvector.

The previous reasoning can be generalized. The matrix Z whose columns are the values of the p components in the n individuals, can be expressed as the product of the matrix X that containing the original variables, multiplied by a matrix A formed by the eigenvectors

$$Z = XA, \tag{1.27}$$

where $A^T A = I$. Computing the principal components is equivalent to applying an orthogonal transformation A to the variables X (original axes) in order to obtain new variables Z which are uncorrelated with each other.

Liu et al. (2003) propose a dimension reduction method for the clustering data using principal component. However, Peña et al. (2010) illustrate by means of a

mixture of two normal populations, that the use of principal components to reduce the dimension is not always suitable, since if the data are projected onto one of the main components, the groups overlap.

PCA can be appropriate for Gaussian distributions since it relies on second-order relationships in the covariance matrix. Other linear transforms, like Independent Component Analysis (ICA) and projection pursuit, which use higher order statistical information, are more suited for non-Gaussian distributions.

1.3.2 Independent Component Analysis

Independent Component Analysis (ICA) is a statistical technique to find the independent latent factors in sets of multivariate random variables. The data variables are assumed to be linear combinations of some unknown latent variables. The latent variables are assumed to be non-Gaussian and mutually independent, and are referred to as independent components of the observed data, see Hyvärinen et al. (2001). In ICA, unlike Principal Components, the data are first standardized to be uncorrelated and then rotated so that independent factors can be found.

Assume that we observe n linear mixtures $x_1, ..., x_n$ of n independent components s_k, $k = 1, ..., n$. In the ICA model, the components of the observed random vector $x = (x_1, ..., x_n)^T$ are generated as a sum of the independent components s_k

$$x_i = a_{i1}s_1 + a_{i2}s_2 + ... + a_{in}s_n \tag{1.28}$$

Let us denote by s the random vector with elements $s_1, ..., s_n$ and by A the matrix with elements a_{ij}. Using this vector-matrix notation, the above mixing model is written as

$$x = As \tag{1.29}$$

The ICA model is a generative model, which means that it describes how the observed data are generated by a process of mixing the components s_i. The independent components are latent variables, meaning that they cannot be directly observed. Also the mixing matrix is assumed to be unknown. All we observe is the random vector x, and the task is to estimate both A and s using it.

The starting point for ICA is the assumption that the components s_i are statistically independent and must have non-Gaussian distributions, We are also assuming that the unknown mixing matrix A is square, see Hyvärinen et al. (2001). Then, after estimating the matrix A, we can compute its inverse, and obtain the independent component by

$$s = A^{-1}x \tag{1.30}$$

Huber (1985) emphasized that interesting projections are those that produce non-Gaussian distributions and therefore non-Gaussianity is one of the criteria used to find the factors. To use non-Gaussianity in ICA estimation, we must have a quantitative measure of non-Gaussianity of a random variable, say z. Let us assume that z is standardized to zero mean and unit variance. As the fourth moment equals 3, the kurtosis is zero for a Gaussian random variable. For most non-Gaussian random variables, kurtosis is nonzero.

Typically non-Gaussianity is measured by the absolute value of kurtosis, and this has been widely used in ICA. The square of kurtosis can also be used. Kurtosis has also some drawbacks in practice, when its value has to be estimated from a measured sample. The main problem is that kurtosis can be very sensitive to outliers (Huber, 1985). Its value may depend on only a few observations in the tails of the distribution, which may be erroneous or irrelevant observations. In other words, kurtosis is not a robust measure of non-Gaussianity, see Hyvärinen and Oja (2000).

There are many proposals to obtain independent components. An interesting

ICA algorithm was proposed by por Hyvärinen and Oja (1997), called FastICA. Is a fixed-point algorithm that finds a direction, i.e. a unit vector w such that the projection $w^T x$ maximizes the absolute value of the univariate kurtosis coefficient.

It is interesting to note how the approach to ICA proposed by Hyvärinen and Oja (2000), makes explicit the connection between ICA and projection pursuit. In the general formulation, ICA can be considered a variant of projection pursuit. In particular, the projection pursuit allows us to tackle the situation where there are less independent components s_i than original variables x_i.

Below we will briefly describe the projection pursuit technique and its applications.

1.3.3 Projection Pursuit

An alternative to the above procedures is to find directions of data projection where the different groups can be seen and then to look for groups in these univariate directions. The advantage to this approach is that it is not necessary to specify the number of groups a priori, nor to compare solutions with very different numbers of groups.

Friedman and Tukey (1974) presented an algorithm for the analysis of multivariate data called Projection Pursuit (PP). This technique is developed for finding "interesting" projections of multidimensional data. The algorithm consists in finding directions w such that the projection of the data, $w^T x$, has an interesting distribution, i.e., displays some structure. Such projections can then be used for optimal visualization of the clustering structure of the data, and for such purposes as density estimation and regression.

Most clustering techniques use the information from all variables in the dataset. With the Projection Pursuit, one may first reduce the dimensionality of the sample by projecting it on a lower dimensional subspace and then finding the clusters

there. The curse of dimensionality can thus be avoided, but care needs to be taken to make sure that the projected data preserve the cluster structure of the original sample.

Projection Pursuit algorithms usually proceed through the following steps: (1) Centralize the original data; (2) Choose an index; (3) Find an projection direction; (4) Project the data and evaluate the index; (5) If the index is not a maximum (or minimum) return to (3); (6) Analyze the projected data.

The central theoretical problem is the definition of the projection pursuit index. The index will define how interesting a direction can be and whether it is worth studying the proposed structure. Usually, the index is some measure of non-Gaussianity, see Huber (1985) and Jones and Sibson (1987). The projection index can be formulated to identify subspaces that reveal the presence of clusters or of outliers. Depending on the formulation of the index under maximization/minimization analytical methods exist.

1.4 Conclusion

The aim of this Chapter was to provide a comprehensive review of several clustering methods that are interesting for the development of this work.

In the Independent Component Analysis and Projection Pursuit techniques, kurtosis has some applications and properties that we will analyze later. In this thesis we have a particular interest in using the kurtosis coefficient as an interesting index to derive projections that can reveal the structure of the data. A more detailed study of the concept of kurtosis and its use in the literature for detecting outliers and for cluster analysis will be discussed in the next Chapter.

Chapter 2

Kurtosis for Cluster Analysis

Summary

The kurtosis coefficient has had different applications and interpretations in the literature. It has been used as a measure of the peakedness of the probability distribution and as a measure of bimodality. Peña and Prieto used kurtosis as projection index to derive projections that can reveal the structure of the data. They proved that maximization of the kurtosis coefficient of the projected data can be used to detect outliers in projections, see Peña and Prieto (2001b). On the other hand, for clustering, they proved that the directions that minimize the kurtosis can be more useful than the ones that maximize it, since for two groups of similar size the directions that minimize the kurtosis are optimal to show the cluster structure. They describe a procedure to identify clusters in multivariate data using information obtained from the univariate projections of the sample data on the directions that minimize and maximize the kurtosis coefficient of the projected data. Under certain conditions, these directions have optimal properties to visualize the different clusters that may be present in data, see Peña and Prieto (2001a).

One interesting property of kurtosis is that the univariate case can be easily generalized to multivariate kurtosis, which not only has the useful properties of univariate kurtosis, but also is independent of the choice of the basis for a subspace. Therefore, related directions can be obtained from a matrix representation of kurtosis. Peña et al. (2010) propose the eigenvectors associated with the extreme values of a kurtosis matrix as interesting directions to reveal the possible cluster structure of a dataset.

In this Chapter we will review in detail the concept of kurtosis. We will study the univariate kurtosis coefficient and the different interpretations that has been given to it in the literature, including the use of the kurtosis coefficient for the outlier detection and as a measure of heterogeneity. We will also study the different ways in which the kurtosis in a multivariate sample can be defined and we explore its properties for cluster analysis. In addition, we will give a detailed description of some algorithms proposed in the literature that use the kurtosis coefficient to detect the clusters present in the data.

2.1 Kurtosis of a Univariate Sample

In symmetrical univariate models, the kurtosis is a measure of the peakedness of the probability distribution of a real-valued random variable. Its value also reflects the presence of heavy tails or bimodality in the data. These properties allow the use of the kurtosis for the identification of the possible cluster structure and the existence of outliers in a data set.

2.1.1 Kurtosis as a Measure of Peakedness

There are different ways to quantify kurtosis: Karl Pearson (1905) introduced kurtosis as a measure of the flat shape of the top of a symmetrical distribution compared to a normal distribution with the same variance, and defined the kurtosis as a measure of deviation from normality, based on the fourth moment of the data.

If X is a random variable with mean μ and standard deviation σ, the univariate kurtosis coefficient defined by Pearson is given by:

$$\kappa = \frac{\mu_4}{\sigma^4},$$

where $\mu_4 = \mathbb{E}(X - \mu)^4$ is the central moment of the fourth order of X.

It is common to use an adjusted version of Pearson kurtosis, excess kurtosis or Fisher kurtosis, to provide a comparison of the form of a given distribution to the distribution of the normal distribution. Excess kurtosis is defined as:

$$\kappa' = \kappa - 3.$$

The term "minus 3" is explained as a correction to make the value of the kurtosis excess for a normal distribution equal to zero, since the normal distribution has kurtosis equal to 3.

The distributions with negative excess kurtosis $\kappa' < 0$ ($\kappa < 3$) are called platykurtic distributions. Compared to a normal distribution its central peak is

lower and wider, and shorter and thinner tail. The distributions with positive excess kurtosis $\kappa' > 0$ ($\kappa > 3$) are called leptokurtic distributions. Compared to a normal distribution its central peak is higher and sharper, and longer and heavier tail. The normal distribution $\kappa' = 0$ ($\kappa = 3$) is called mesokurtic.

Given a univariate random sample $x_1, x_2, \ldots, x_n$, drawn from the random variable X, the sample univariate kurtosis coefficient is

$$\kappa = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{s^4} = \frac{n\sum_{i=1}^{n}(x_i - \bar{x})^4}{[\sum_{i=1}^{n}(x_i - \bar{x})^2]^2}, \tag{2.1}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the sample mean and $s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$ is the sample variance.

It is easy to see that the kurtosis coefficient is equivariant, and its minimum value is one.

2.1.2 Kurtosis as a Measure of Bimodality

A relevant interpretation for our purposes is given by Darlington (1970), which describes the kurtosis not as a measure of peakedness of a distribution, as in most of the texts, but as a measure of unimodality against bimodality; the smaller the kurtosis, the greater the bimodality. κ and z calling

$$\kappa = \frac{1}{n}\sum_{i=1}^{n} {z_i}^4, \tag{2.2}$$

where, $z_i = s^{-1}(x_i - \bar{x})$, the mean of the squared scores is one and the variance of the squared scores is

$$s(z_i^2) = \frac{1}{n}\sum_{i=1}^{n}(z_i^2 - 1)^2 = \frac{1}{n}\sum_{i=1}^{n} z_i^4 - 1 = \kappa - 1 \tag{2.3}$$

Thus the kurtosis can be interpreted as a measure of the degree to which the values of z^2 cluster around their mean value 1. Consequently, if all observations

of the sample are approximately at the same distance to the mean, the variance of these distances is near zero, and the kurtosis will have a small value.

From (2.3), we can see that $s(z_i^2) = 0$ when κ is 1. If $z^2 = 1$, $z = 1$ or $z = -1$, i.e., all z's are concentrated at $+1$ and -1. Therefore, κ also can be interpreted as a measure of the degree to which a distribution's z-scores cluster around $+1$ and -1. This is a symmetric two-point distribution and this clustering can be interpreted as "bimodality". A unimodal distribution is completely concentrated at one point, while a bimodal distribution is a symmetric two-point distribution and this is the only distribution for which k is 1.

Considering the family of all two-point distributions with densities p and $q = 1 - p$ respectively, the kurtosis value is proven in Darlington (1970) to be

$$\kappa = \frac{1}{pq} - 3.$$

The minimum value of k is reached when $p = q = \frac{1}{2}$, which agrees with the results above regarding bimodality. On the other hand, k approaches infinity when p or q approaches zero, i.e. as the distribution concentrates on one point or the other. Note that the symmetric two point mass distribution is the only distribution that reaches the minimum kurtosis value of 1.

In the same direction, Hildebrand (1971) considers the family of symmetric beta distributions and confirms Darlington's (1970) statement. Otherwise, Moors (1986) claims that Darlington's result regarding bimodality should be reexamined and that the bimodal distributions can have large kurtosis; this occurs if the modes are not close to the points $z = \pm 1$. He formulate that kurtosis measures the dispersion around the two values $\mu - \sigma$ and $\mu + \sigma$, instead of the values -1 and $+1$. According to Moors, high kurtosis may arise in two situations that explain the confusion about the interpretation of kurtosis: (a) concentration of probability mass near μ, which corresponds to a peaked unimodal distribution, and (b) concentration of probability mass in the tails of the distribution.

2.1.3 Kurtosis Coefficient for Detecting Outliers and as a Measure of Heterogeneity

The use of the kurtosis coefficient to reveal the presence of outliers was proposed by Peña and Prieto (2001b). They analyze the effect of outliers on the kurtosis coefficient considering two cases: (1) The centered case. If we have outliers generated by an symmetric contaminated model, the kurtosis coefficient increases due to the presence of outliers. (2) The noncentered case. If we have a large proportion of outliers generated by an asymmetric contamination model, the kurtosis coefficient of the data is very small, but if the contamination is small the kurtosis coefficient will be large. Therefore, they propose a outlier detection procedure based on the analysis of the projections onto the directions that maximize and minimize the kurtosis coefficient of the projected data.

The kurtosis coefficient has also been considered as a measure of heterogeneity. Suppose that we define $d_i = (x_i - \bar{x})^2$ as the distances of observations to the mean. If the d_i's are very different, this suggests that some observations are very separated from the mean and therefore we have high heterogeneity. A possible measure of homogeneity is the variance of the d_i's, given by:

$$\frac{1}{n}\sum_{i=1}^{n}(d_i - s^2)^2 \tag{2.4}$$

where the variance of the sample $s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n}\sum_{i=1}^{n} d_i$ is also the mean of the d_i's.

We can define the *homogeneity coefficient*, as a dimensionless measure analogous to the coefficient of variation $s/\bar{x}$ as

$$H = \frac{\frac{1}{n}\sum_{i=1}^{n}(d_i - s^2)^2}{s^4} \tag{2.5}$$

Since $\sum_{i=1}^{n}(d_i - s^2)^2 = \sum_{i=1}^{n} d_i^2 + ns^4 - 2s^2\sum_{i=1}^{n} d_i = \sum_{i=1}^{n} d_i^2 - ns^4$, then we can

write (2.5) as

$$H = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{s^4} - 1 = \kappa - 1 \tag{2.6}$$

Therefore, the univariate kurtosis coefficient can be seen as a measure of heterogeneity.

In the following we comment two situations in which extreme values of the kurtosis coefficient are presented: (1) If we have a sample with several groups, the mean of the sample will be located near the largest group. If there are some outliers, the distances between the mean of the sample and the outliers will be large compared to the other observations, this will make the variances of the distances large, as well as the kurtosis coefficient. (2) If all the data in the sample are separated into two different data clusters of the same size that are approximately at the same distance to the mean, the variance of these distances is near zero and the kurtosis coefficient will have a small value. In this case the directions that minimize kurtosis could reveal the cluster structure.

Therefore, both the directions that minimize the kurtosis coefficient and the ones that maximize it are interesting in the sense that are able to identify structures with more than one cluster. Peña and Prieto (2001a) propose a one-dimensional projection pursuit algorithm based on directions obtained by both maximizing and minimizing the kurtosis coefficient of the projected data, assigning the observations to groups according to the clusters found in the directions. They showed that minimizing the kurtosis coefficient implies maximizing the bimodality of the projections, whereas maximizing the kurtosis coefficient implies detecting groups of outliers in the projections.

2.1.4 The Peña and Prieto Clustering Algorithm

The clustering algorithm proposed by Peña and Prieto (2001a) is based on the analysis of a set of $2p$ orthogonal directions for a p-dimensional random variable,

such that each direction minimizes or maximizes the kurtosis coefficient. The data are projected onto these directions to determine the clusters existence.

The procedure uses the sample spacings or first-order gaps between the ordered statistics of the projected points to detect patterns that may indicate the presence of clusters. If the univariate observations come from a unimodal distribution, there will be large gaps near the extremes of the distribution and small gaps near the center. However, this pattern will change if there are clusters in the data. For example, with two clusters of similar size we expect a large first-order gap separating the clusters, lying towards the center of the observations. The gaps or spacings of the sample are defined as the differences between two consecutive order statistics, more details on the properties of the gaps are found in Peña and Prieto (2001a). The cluster algorithm proceeds as described below.

Assume that we are given a sample of size n, $\{x_i\}$ $i = 1, \ldots, n$, from a p-dimensional random variable $X \in \mathbb{R}^p$.

1. The algorithm starts by computing a direction d that maximizes the kurtosis coefficient $\kappa(d)$ of the projected data $\{x_i^T d\}$. Let $k = 1$, denote this direction d as d_1 and let $k = 2$.

2. For $k = 2, \ldots, p-1$, the sample is projected onto the subspace orthogonal to $\{d_1, \ldots, d_{k-1}\}$ (in $\mathbb{R}^p$) and a new search is conducted to obtain a direction d_k that maximizes $\kappa(d)$ for the projected data among all directions in the subspace; we then increase k to $k+1$.

3. Letting d_p denote the unit direction corresponding to the subspace orthogonal to $\{d_1, \ldots, d_{p-1}\}$, at the end of this procedure we have a set of p orthogonal directions $\{d_1, \ldots, d_p\}$ in $\mathbb{R}^p$, maximizing the kurtosis coefficient on a sequence of nested subspaces.

4. Another set of p directions is then computed by repeating steps 1 to 3, but this time minimizing κ. Denote these directions as $\{d_{p+1}, \ldots, d_{2p}\}$.

   These directions will be used to reduce the problem of cluster identification to that of finding clusters in univariate samples, in the following manner:

5. For each one of the directions d_k, $k = 1, \ldots, 2p$, compute the univariate projections of the original observations $u_{ki} = x_i^T d_k$.

6. Standardize these observations, $z_{ki} = \frac{(u_{ki} - m_k)}{s_k}$, where $m_k = \frac{1}{n}\sum_i u_{ki}$ and $s_k = \frac{1}{n-1}\sum_i (u_{ki} - m_k)^2$.

7. The standardized observations are then transformed using the inverse of the standard normal distribution function, as $\bar{z}_{ki} = \Phi^{-1}(z_{ki})$. Note that if the original observations were normal, these transformed observations would follow a uniform distribution.

8. For each k the transformed observations $\bar{z}_{ki}$ are sorted in ascending order, to obtain their order statistics $\bar{z}_{k(i)}$.

   The gaps between consecutive (transformed) values are obtained as $w_{ki} = \bar{z}_{k(i+1)} - \bar{z}_{k(i)}$.

9. A search for the presence of significant gaps is conducted in $\{w_{ki}\}$.

   Each gap is compared with the value of a set threshold to decide if more than one cluster is present in the data. In particular, they introduce a threshold $\delta = \nu(c)$, where $\nu(c) = 1 - (1 - c)^{(1/n)}$ denotes the value of the c^{th} percentile of the distribution of the (uniform) spacings. For simulation experiments, they used $\log(1 - c) = \log(0.1) - \frac{10}{3}\log(p)$, and consequently $\delta = 1 - (0.1/p^{10/3})^{(1/n)}$.

If a gap is greater than δ, then it is considered to be significant. These significant gaps are used to assign different labels to observations assumed to belong to different groups.

The general principle applied to the study of the projections and the gaps will be that whenever two (or more) groups are identified from these projections onto one of the directions, these groups will be treated as separate ones for the analysis of later projections, although they could be further subdivided if the gaps along these new projections are significant.

At the start, the same label l_1 is assigned to all observations.

10. All gaps greater than δ (the significant gaps) for a given direction (index) k are identified. The observations within each pair of consecutive significant gaps are assigned to new groups, in the following manner: the observations between each set of consecutive significant gaps that shared a common label before still share a common label after relabelling, but these new labels are different from those corresponding to observations between other sets of consecutive significant gaps.

11. Go to the next projection direction k and repeat steps 5 to 10 for $k = 1, \ldots, 2p$.

12. As the number of different labels assigned through this algorithm can be very large, a final step is conducted to reduce it by combining observations with other groups whenever their Mahalanobis distances to these groups are small enough.

This method works well when the data dimension is low and when the number of groups in the sample is small. But the method fails when the data dimension

increases. The results of an example of simulation where these failures can be observed are presented in Chapter 3.

In addition to that, Peña and Prieto (2001a) proved that under a mixture of two normal distributions with proportional scatter matrices, either the direction that maximizes or the one that minimizes the kurtosis coefficient is Fisher's linear discriminant function. Let α be the proportion of one of the normal distributions, if $\alpha \in (0, 0.2)$ then Fisher's linear discriminant function is the direction that maximizes the kurtosis coefficient, whereas for $\alpha \in (0.2, 0.5]$ the Fisher's function is the direction that minimizes it.

2.2 Kurtosis for Multivariate Data

Let $X \in \mathbb{R}^p$ be a multivariate random vector with mean μ, $\Sigma = \mathbb{E}[(X-\mu)(X-\mu)^T]$ its covariance matrix and $Z = \Sigma^{-1}(X-\mu)$ the corresponding standardized vector. The concept of kurtosis coefficient can be generalized to the multivariate case. One of the proposals most frequently used is given by Mardia (1970), who proposes to calculate a scalar value for the kurtosis coefficient of a multivariable sample as the second moment of the Mahalanobis distances,

$$\beta_{2,p} = \mathbb{E}[(X-\mu)^T\Sigma^{-1}(X-\mu)]^2 \tag{2.7}$$

For $\mu = 0$ and $S = I$, we have $\beta_{2,p}$ in terms of the standardized vector Z

$$\beta_{2,p} = \mathbb{E}[Z^T Z]^2 \tag{2.8}$$

which is invariant under orthogonal transformations. For a random sample $x_1, ..., x_n$, the measure of kurtosis corresponding to $\beta_{2,p}$ is

$$b_{2,p} = \frac{1}{n}\sum_{i=1}^{n}[(x_i-\bar{x})^T S^{-1}(x_i-\bar{x})]^2 \tag{2.9}$$

where $\bar{x}$ and S are the mean and covariance matrix of the sample. Mardia (1970) proposes to use $b_{2,p}$ to test for normality. Under a Gaussian distribution,

$$\beta_{2,p} = p(p+2), \tag{2.10}$$

therefore values of $b_{2,p}$ differing significantly from $p(p+2)$ indicate non-Gaussianity.

Koziol (1989) proposes the following measure of multivariate kurtosis

$$K_{2,p} = \sum_{j,k,l,m}^{p} \mathbb{E}(Z_j Z_k Z_l Z_m)^2 \tag{2.11}$$

For a random sample the measure of kurtosis corresponding to $\tilde{\beta}_{2,p}$ is

$$k_{2,p} = \sum_{j,k,l,m}^{p} (\frac{1}{n} \sum_{i=1}^{n} (z_{ji} z_{ki} z_{li} z_{mi})^2) \tag{2.12}$$

The difference between $\beta_{2,p}$ and $K_{2,p}$ is that $\beta_{2,p}$ is the sum of just the symmetric fourth-order moments, whereas $K_{2,p}$ is the sum of squares for all existing fourth-order moments of Z.

More interesting than a scalar criterion is the possibility to define a matrix representation of the kurtosis. In this case different alternatives are proposed in the literature. An interesting proposal, because of its simplicity, is given by Cardoso (1989) and Móri et al. (1993). They define the following kurtosis matrix

$$K = \mathbb{E}(Z^T Z Z Z^T) \tag{2.13}$$

The matrix K reduces to the univariate kurtosis coefficient in the univariate case,

$$K = \mathbb{E}(ZZZZ) = \mathbb{E}(Z^4) = \frac{\mu_4}{\sigma^4} \tag{2.14}$$

For a random sample, the kurtosis corresponding to K is

$$K_n = \frac{1}{n} \sum_{i=1}^{n} z_i^T z_i z_i z_i^T, \tag{2.15}$$

where $z_i = S^{-1/2}(x_i - \bar{x})$. The trace of K coincides with the Mardia's kurtosis coefficient $\beta_{2,p}$ raised in (2.7),

$$\operatorname{tr}(K) = \operatorname{tr}[\mathbb{E}(Z^T Z Z Z^T)] = \mathbb{E}[Z^T Z \operatorname{tr}(Z Z^T)] = \mathbb{E}[Z^T Z]^2 = \beta_{2,p} \tag{2.16}$$

Also, since K is a continuous function of the moments then K_n converges to K in probability and the matrix K_n is a consistent estimator of K.

There are other variants of these definitions, for example that given by Kollo (2008),

$$B = \mathbb{E}[(Z^T \mathbf{1})^2 Z Z^T)] \tag{2.17}$$

The matrix B is the sum of the p^2 blocks of size $p \times p$ of the $p^2 \times p^2$ matrix $M_4 = \mathbb{E}(ZZ^T \otimes ZZ^T)$ that collects all $p2 \times p2$ multivariate fourth-order central moments, where $\otimes$ denotes the Kronecker product. This matrix M_4 can be written as

$$M_4 = \mathbb{E}\begin{pmatrix} Z_1^2 \begin{bmatrix} Z_1 \\ \vdots \\ Z_p \end{bmatrix} [Z_1 \cdots Z_p] & \cdots & Z_1 Z_p \begin{bmatrix} Z_1 \\ \vdots \\ Z_p \end{bmatrix} [Z_1 \cdots Z_p] \\ \vdots & \ddots & \vdots \\ Z_p Z_1 \begin{bmatrix} Z_1 \\ \vdots \\ Z_p \end{bmatrix} [Z_1 \cdots Z_p] & \cdots & Z_p^2 \begin{bmatrix} Z_1 \\ \vdots \\ Z_p \end{bmatrix} [Z_1 \cdots Z_p] \end{pmatrix} \tag{2.18}$$

For a random sample, the kurtosis corresponding to B is

$$B_n = \frac{1}{n} \sum_{i=1}^{n} (z_i^T \mathbf{1})^2 z_i z_i^T \tag{2.19}$$

In the univariate case, B also reduces to the univariate kurtosis coefficient

$$B = \mathbb{E}(Z^2 Z Z) = \mathbb{E}(Z^4) = \frac{\mu_4}{\sigma^4} \tag{2.20}$$

Due to the convergence of moments, B_n converges to B in probability and is a consistent estimator of B.

The matrix K in (2.13) has an important invariant property which is not present in B in (3.8). If E is an orthogonal matrix whose columns are eigenvectors of K, the new coordinate system $E^T Z$ is invariant under affine transformations of X. However, the matrix B does not have this desirable property because its weights are not invariant under orthogonal transformations, see Peña et al. (2010).

As in the univariate case, from the definitions for kurtosis in the multivariate case, some proposals has been made in detecting multivariate outliers and the use of the multivariate kurtosis as a measure of heterogeneity.

2.2.1 Multivariate Kurtosis Matrix for Detecting Outliers and for Clustering

A classical approach to detecting multivariate outliers that is discussed in Jobson (2012) is to examine the squared Mahalanobis distance for each case; a large value indicating a multivariate outlier. Note that the Mahalanobis distances are also related to Mardia's measure of multivariate kurtosis, see (2.7). In fact, Mardia's measure of multivariate kurtosis has been shown to have good properties for detecting multivariate outliers in some situations. For example, a large value of Mardia's measure, relative to the expected value under multivariate normality, suggests the presence of one or more cases with large Mahalanobis distances, which are cases that are far from the centroid of all cases (potential outliers), see Schwager and Margolin (1982).

The multivariate kurtosis has also been used as a measure of heterogeneity. Peña et al. (2010) proposed the eigenvectors associated with the extreme values of a kurtosis matrix as interesting directions to reveal the possible cluster structure of a data set. In addition, they proved that under a mixture of two elliptical

distributions with the same scatter matrices, the eigenvectors of the fourth-order moment matrix corresponds to Fisher's linear discriminant subspace.

In Peña et al. (2010) they consider that for a p-dimensional random variable X corresponding to a mixture of two normal distributions with the same covariance matrix $X \sim \alpha_1 N(\mu_1, V) + \alpha_2 N(\mu_2, V)$, $X \in \mathbb{R}^p$, the matrix K can be expressed as

$$K = (p+2)I + \beta\varphi^T\varphi\varphi\varphi^T, \tag{2.21}$$

where $\beta = \alpha_1\alpha_2(1-6\alpha_1\alpha_2)$, $\Sigma = V + \alpha_1\alpha_2(\mu_2-\mu_1)(\mu_2-\mu_1)^T$ and $\varphi = \Sigma^{-1/2}(\mu_2 - \mu_1)$. The vector φ is an eigenvector of K with associated eigenvalue $\lambda = p + 2 + \beta(\varphi^T\varphi)^2$, the rest of the eigenvalues are equal $p+2$. Following expression (2.16), the trace of K is the Mardia's kurtosis coefficient. From (2.10), if the means of the two populations are the same then $\text{tr}(K) = p(p+2)$, but as we are in the mixture case then the expression for $\text{tr}(K)$ using (2.21) is

$$\text{tr}(K) = p(p+2) + \beta(\varphi^T\varphi)^2, \tag{2.22}$$

and $\det(K) = (p+2)^p + \beta(p+2)^{p-1}(\varphi^T\varphi)^2$. Note that φ is Fisher's best linear discriminant function in the Z-space. The eigenvalue λ is the largest if $\beta > 0$ and the smallest otherwise. $\beta > 0$ if $\alpha_1 \in (0, 0.2)$ and $\beta < 0$ if $\alpha_1 \in (0.2, 0.5]$. Therefore, if we have homogeneous clusters, the eigenvector associated with the smallest eigenvalue will be the one that better separates the clusters, while when the two clusters have very different sizes, the largest eigenvalue is the one that identifies the significant eigenvector. These values are the same values that arise in Peña and Prieto (2001a).

For this case both approaches give the same estimates. Therefore, the kurtosis matrix has similarities with the nonlinear cluster algorithm described in section 2.1.4, but it is necessary to analyze in which situations it is more satisfactory one than the other. Peña et al. (2010) showed for the case of a mixture of two normal distributions with equal scatter matrices that if the sample size is small, it

is better to use the clustering algorithm for univariate kurtosis directions described in section 2.1.4 instead of the eigenvectors of a kurtosis matrix, since the estimation of the elements of the matrix has a very low precision and the eigenvalues are not so useful. On the other hand,if the ratio n/p, where n is the sample size and p the dimension, is large, the estimation of the autovectors of the kurtosis matrix can be accurate and useful. To solve this optimization problem, they propose a computationally intensive nonlinear algorithm based on the eigenvectors of the kurtosis matrix, which is described below:

2.2.2 An Algorithm for Detecting Clusters Using Eigenvectors

The algorithm proposed by Peña et al. (2010) based on the eigenvectors of the kurtosis matrix, proceeds as described below:

1. The algorithm starts by standardizing the sample data, $Z = \Sigma^{-1}(X - \mu)$.

2. Compute a kurtosis matrix $K = \mathbb{E}(Z^T Z Z Z^T)$.

3. Compute the eigenvectors of this kurtosis matrix, K. Let $E = [E_1, E_2]$ be a matrix with two orthogonal columns corresponding to the eigenvectors of K associated to its maximum and minimum eigenvalues.

4. Obtain the projections of the standardized observations onto each of these two directions, $p = E^T Z$.

5. Analyze for each of the projections, $p_1 = E_1^T Z$ and $p_2 = E_2^T Z$, the existence of groups by studying their first-order gaps.

   The study of the first-order gaps is done in a similar way to that indicated in the Peña and Prieto Clustering Algorithm described in Section 2.1.4.

This algorithm is fast and computationally efficient to calculate the eigenvectors compared to the time needed to calculate the extreme kurtosis directions when the dimension of the data increases, since with this method a single eigenvalue computation is performed, while in the optimization of Kurtosis it is necessary to factorize the corresponding second derivative matrix in each iteration.

Although this method performs well in practice, in some cases, for example when the matrix K is diagonal, the eigenvectors will not identify the direction of the means.

2.3 Conclusion

In this Chapter we reviewed the concept of kurtosis and its different interpretations given in the literature.

For the univariate case, we summarized the interpretations of the kurtosis coefficient as a measure of the peakedness and as a measure of bimodality. We also reviewed how the kurtosis coefficient has been used as projection index to derive projections that can reveal the structure of the data. Then we analyzed the Peña and Prieto (2001b) proposal, they proved that maximization of the kurtosis coefficient of the projected data can be used to detect outliers. In Section 2.1.4 we did a detailed description of their clustering algorithm. The algorithm is based on analyzing a a full set o f $2p$ orthogonal directions, such that each direction minimizes or maximizes the kurtosis coefficient. The criteria used to identify the clusters present in the data are based on the analysis of the first-order gaps between the ordered statistics of the projections. The method works well when the data dimension is low and when the number of clusters present in the sample is small, but fails when the data dimension increases.

Additionally, we also studied the different interpretations of kurtosis in

multivariate samples and its matrix representation. We analyzed the Peña et al. (2010) proposal, where they indicated that the eigenvectors associated with the extreme values of a kurtosis matrix are interesting directions to reveal the possible cluster structure of a dataset. In the section 2.2.2 we described the algorithm for detecting clusters proposed by these authors based on the eigenvectors of the kurtosis matrix.

Since the methods described in this Chapter have some limitations. We will propose in the next Chapter a method that could be efficient in its application when the dimension of the data and the number of clusters in the sample are large.

Chapter 3

A Cluster Procedure for High-Dimensional Data

Summary

The methods for cluster analysis mentioned in the previous Chapter have some limitations in the presence of multiple clusters in the sample and also when the dimension of the data is large. In this Chapter we present a theoretical study of the kurtosis coefficient to identify clusters when we have more than two groups in the sample data. We also introduce a cluster procedure based on these results.

Peña and Prieto (2001) showed that the extreme kurtosis directions of projected data are optimal when the data has been generated by mixtures of two normal distributions. We generalize this result and we prove that the directions of extreme kurtosis generate the subspace of optimal directions for discrimination when we have a mixture of normal distribution with the same covariance matrix. We prove that this subspace contains directions which split the components of the mixture in two blocks, so that some components are projected jointly in one block and the others in another block. We call these directions two-block projection directions,

because they allow the identification of heterogeneity by splitting the data into two blocks.

We show that extreme kurtosis directions are asymptotically two-block projection directions. This result suggest a binary decision strategy in order to separate the groups where it is decided at each step if the data should be split into two groups or we should stop. The decision is based on fitting to the projected data both a normal and a mixture of two normal distributions and selecting the best model by using the BIC criterion. We develop an algorithm based on these ideas and we analyze its performance through a simulation study, in which we compare it with different proposals from the literature.

3.1 Extreme Projected Kurtosis as Optimal Directions for Discrimination

We are interested in studying the behavior of the kurtosis coefficient when we have a p-dimensional variable corresponding to a mixture of k normal distributions with the same covariance matrix. Let X be the p dimensional random variable such that $k \leq p+1$

$$X \sim \sum_{i=1}^{k} \alpha_i N(\mu_i^x, \Sigma), \quad X \in \mathbb{R}^p, \tag{3.1}$$

where $\mathbb{E}(X) = \sum_{i=1}^{k} \alpha_i \mu_i^x$ and $\sum_{i=1}^{k} \alpha_i = 1$. In what follows we will assume that the following condition holds, implying that the mixture is well-defined, as otherwise we could study an equivalent mixture having less than k components:

A1. It holds that $\alpha_i > 0$ for all i, and $\mu_i^x \neq \mu_j^x$ for all i, j, $i \neq j$. Also, the covariance matrix Σ has full rank.

Without loss of generality we consider the transformation $Y = \Sigma^{-1/2}(X - \mathbb{E}(X))$ which leads to

$$Y \sim \sum_{i=1}^{k} \alpha_i N(\mu_i, I), \quad \mu_i = \Sigma^{-1/2}(\mu_i^x - \mathbb{E}(X)), \quad \mathbb{E}(Y) = 0.$$

Consider now an arbitrary direction d, with $\|d\| = 1$, and the univariate projection $z = d^T Y$. This univariate projected variable has distribution

$$z \sim \sum_{i=1}^{k} \alpha_i N(m_i, 1), \quad z \in \mathbb{R} \tag{3.2}$$

where $m_i = d^T \mu_i$, $\quad \mathbb{E}(z) = d^T \sum_{i=1}^{k} \alpha_i \mu_i = 0$.

Our interest is to study those directions d that can reveal the heterogeneity in the data in the univariate projections z. We will show that the directions obtained

as extreme points for the kurtosis coefficient have this property. The coefficient of kurtosis for a univariate zero mean variable is given by

$$\kappa_z = \frac{m_z(4)}{m_z(2)^2} \tag{3.3}$$

where $m_z(k) = \mathbb{E}[z^k]$. Let

$$v_2 \equiv \sum_{i=1}^{k} \alpha_i m_i^2, \quad v_4 \equiv \sum_{i=1}^{k} \alpha_i m_i^4, \tag{3.4}$$

where v_2 is the variance and v_4 is the kurtosis of the projected means. Then, $m_z(2) = 1 + v_2$, $m_z(4) = 3 + 6v_2 + v_4$ and the kurtosis coefficient of the projected data can be written as

$$\kappa_z(d) = \frac{3 + 6v_2 + v_4}{(1 + v_2)^2} \tag{3.5}$$

Theorem 1 *The stationary points of the problem*

$$\begin{array}{lll} \min_d & \kappa_z(d) & \qquad \max_d \quad \kappa_z(d) \\ s.t. & d^T d = 1 & \qquad s.t. \quad d^T d = 1 \end{array} \tag{3.6}$$

satisfy $d \in span\{\mu_i - \mu_k\}$.

Proof The derivatives of the $\kappa_z(d)$ function are

$$\frac{\partial \kappa_z(v_2, v_4)}{\partial v_2} = \frac{-2(3v_2 + v_4)}{(1 + v_2)^3} \equiv A, \tag{3.7}$$

and

$$\frac{\partial \kappa_z(v_2, v_4)}{\partial v_4} = \frac{1}{(1 + v_2)^2} \equiv B. \tag{3.8}$$

Therefore,

$$\nabla_d \mathcal{L}(d, \lambda) = A \nabla_d v_2 + B \nabla_d v_4 - 2\lambda d, \tag{3.9}$$

where

$$\nabla_d v_2 = 2\sum_{i=1}^{k} \alpha_i m_i \mu_i \tag{3.10}$$

$$\nabla_d v_4 = 4\sum_{i=1}^{k} \alpha_i m_i^3 \mu_i \tag{3.11}$$

Thus, we obtain that $\lambda = -2(3v_2^2 - v_4)/(1+v_2)^3$. Note that this value will be different from zero if $3v_2^2 \neq v_4$, that is, if the kurtosis coefficient of the projected means is different from 3. In this case, the kurtosis coefficient (3.5) is constant, equal to 3, and any direction is a solution of the problem. Assuming that $3v_2^2 \neq v_4$, the stationary points of (3.6) satisfy

$$d = \sum_{i=1}^{k} c_i \mu_i, \tag{3.12}$$

for $c_i = \frac{1}{\lambda}\alpha_i m_i (A + 2Bm_i^2)$.

As a consequence, any stationary point d of (3.6) is a linear combination of the vectors $\{\mu_i\}$. Finally, as $\mu_i = \mu_i - \mu_k - \sum_{j=1}^{k-1} \alpha_j(\mu_j - \mu_k)$, it holds that $d = \sum_{i=1}^{k-1} \bar{c}_i(\mu_i - \mu_k)$, for $\bar{c}_i = c_i - \alpha_i \sum_{j=1}^{k} c_j$, the desired result. □

From Theorem 1, it holds that there exists an optimal direction d in the subspace generated by $\{\mu_1 - \mu_k, \ldots, \mu_{k-1} - \mu_k\}$. Note that for $k = 2$ the optimal direction is $\mu_1 - \mu_2 = \Sigma^{-1/2}(\mu_{x1} - \mu_{x2})$, the Fisher linear discriminant direction, as proved in Peña and Prieto (2001). For $k \geq 3$ the optimal direction depend on the relative positions of the μ_i and the mixing proportions α_i. For instance, if all the means are colinear and $\mu_i = b_i v$ where v is $p \times 1$ vector the optimal direction is v. In the next section we will consider the general case.

In this article we will concentrate in mixtures of normal distributions but the optimality properties of the extreme kurtosis projections can be extended to mixture of elliptical distributions, see Peña and Prieto (2001).

3.1.1 Two-block Projection Directions

We are interested in the study of directions that allow the detection of the different groups of the mixture from the study of the univariate projections of the observations. Suppose that we can find directions where the projected data appear in two separated blocks, each one corresponding to a subset of the components in the mixture. Then a iterative binary separation applied to Y would be effective to separate the groups, and the procedure could be applied for a large number of groups. These two-block projection directions d could be characterized using the following property:

$$\begin{aligned} d^T\mu_i &= D_1 > 0, \quad i \in I_1 \\ d^T\mu_i &= D_2, \qquad\quad i \in I_2, \end{aligned} \tag{3.13}$$

for some values D_1 and D_2, where $d^Td = 1$ and I_1, I_2 denote a partition of the labels $\{1, \ldots, k\}$, assuming both I_1 and I_2 are nonempty. Letting $\tilde{\alpha} = \sum_{i \in I_1} \alpha_i$, the values D_1 and D_2 are related by $D_1\tilde{\alpha} = -D_2(1 - \tilde{\alpha})$.

These directions would help to separate the groups associated with I_1 from the groups associated with I_2, as long as these groups are sufficiently removed from each other in the data, that is, whenever D_1 is large enough. The value D_1 is a function of the vectors μ_i, a property of the geometry of these centers. In order to justify the existence of these directions the means should not be colinear, and in what follows we will assume that the following condition holds:

A2. The vectors $\{\mu_i^x - \mu_k^x\}_{i=1}^{k-1}$ are linearly independent.

This assumption implies that the vectors $\{\mu_i - \mu_k\}_{i=1}^{k-1}$ are also linearly independent.

The following result proves that the directions introduced in (3.14) exist, and that there is a unique such direction in the subspace spanned by $\{\mu_i - \mu_k\}$.

Lemma 1 *Under condition* A2*, the directions d defined in (3.14) always exist and are unique on span$\{\mu_i - \mu_k\}_{i=1}^{k-1}$ for any non-empty partition (I_1, I_2).*

Proof We consider directions d defined as a linear combination of the vectors $\{\mu_i - \mu_k\}$,

$$d = \sum_{i=1}^{k-1} \gamma_i(\mu_i - \mu_k) = M\gamma,$$

for $M \in \mathbb{R}^{p\times(k-1)}$, a full-rank matrix with columns corresponding to the vectors $\mu_i - \mu_k$, and $\gamma \in \mathbb{R}^{k-1}$. Assume without loss of generality that $1 \in I_1$ and $k \in I_2$ (otherwise we exchange the numbers of the groups), so that $d^T\mu_1 = D_1$ and $d^T\mu_k = D_2$. We can substitute D_1 by $d^T\mu_1$ and D_2 by $d^T\mu_k$ in the conditions (3.14); ignoring the trivial conditions corresponding to $i = 1$ and $i = k$ we obtain $k-2$ conditions of the form $d^T(\mu_i - \mu_1) = 0$ for $i \in I_1\backslash\{1\}$ and $d^T(\mu_i - \mu_k) = 0$ for $i \in I_2\backslash\{k\}$. We have a system of equations of the form

$$\sum_{i=1}^{k-1} \gamma_i(\mu_i - \mu_k)^T(\mu_j - \mu_1) = 0 \text{ for } j \in I_1\backslash\{1\}$$
$$\sum_{i=1}^{k-1} \gamma_i(\mu_i - \mu_k)^T(\mu_j - \mu_k) = 0 \text{ for } j \in I_2\backslash\{k\},$$

that can be written as $N\gamma = 0$, where

$$N_{ij} = \begin{cases} (\mu_i - \mu_k)^T(\mu_j - \mu_1) & \text{if } j \in I_1\backslash\{1\},\ i = 1,\dots,k-1, \\ (\mu_i - \mu_k)^T(\mu_j - \mu_k) & \text{if } j \in I_2\backslash\{k\},\ i = 1,\dots,k-1, \end{cases}$$

Note that $N \in \mathbb{R}^{(k-2)\times(k-1)}$ and under assumption *A2*, it has full row rank, $k-2$. From the property that the span of N^T and the null space of N are orthogonal complements of $\mathbb{R}^{k-1}$, and as $\dim(\text{span}(N^T)) = k-2$, it holds that $k - 1 = \dim(\text{span}(N^T)) + \dim(\text{null}(N))$, implying $\dim(\text{null}(N)) = 1$. Thus, there exist a (unique) direction $d = M\gamma$ satisfying $d^Td = 1$ with $N\gamma = 0$, and such that $d^T\mu_i > 0$ for $i \in I_1$, completing the proof. □

The following result shows the relationship between these two-group projection directions and the extreme points of the kurtosis coefficient of the projected data obtained from (3.6). This result is asymptotic in nature, that is, we obtain this result as a limiting property of a collection of populations with component means that are progressively further apart from other means, in the sense that $D_1 \to \infty$. The result requires a regularity condition on the asymptotic behavior of the means μ_i along directions different from d. We introduce the following condition:

A3. There exists a constant L such that for all $i = 1, \ldots, k$

$$\|\mu_i\| \leq L D_1^2.$$

This condition ensures that the separation between means is not too large along other directions different from d, compared to the separation along d, which is of order D_1.

Theorem 2 *If conditions* A1, A2 *and* A3 *hold and d satisfies both (3.14) and $d^T d = 1$, then there exists some multiplier $\tilde{\lambda}$ such that the gradient of the lagrangian function of problems (3.6) satisfies*

$$\lim_{D_1 \to \infty} \left\| \nabla_d \mathcal{L}(d, \tilde{\lambda}) \right\| = 0.$$

Proof Assume d satisfies (3.14) and $d^T d = 1$. Using $\tilde{\alpha} \equiv \sum_{i \in I_1} \alpha_i$, it holds that

$$m_i = D_1, \quad i \in I_1, \qquad m_i = -\frac{\tilde{\alpha}}{1-\tilde{\alpha}} D_1, \quad i \in I_2. \tag{3.14}$$

The gradient of the objective function of (3.6) is given by $\nabla \kappa_z = A \nabla v_2 + B \nabla v_4$, and using (3.7), (3.8),(3.10) and (3.11) we have

$$v_2 = \frac{1}{1-\tilde{\alpha}} \tilde{\alpha} D_1^2 \qquad v_4 = \frac{(1-\tilde{\alpha})^3 + \tilde{\alpha}^3}{(1-\tilde{\alpha})^3} \tilde{\alpha} D_1^4$$

$$\nabla v_2 = \frac{2}{1-\tilde{\alpha}} D_1 \tilde{\mu} \qquad \nabla v_4 = \frac{4((1-\tilde{\alpha})^3 + \tilde{\alpha}^3)}{(1-\tilde{\alpha})^3} D_1^3 \tilde{\mu}$$

$$A = -2\frac{3(1-\tilde{\alpha})^2 + ((1-\tilde{\alpha})^3 + \tilde{\alpha}^3)D_1^2}{(1-\tilde{\alpha}+\tilde{\alpha}D_1^2)^3}\tilde{\alpha}D_1^2$$

$$B = \frac{(1-\tilde{\alpha})^2}{(1-\tilde{\alpha}+\tilde{\alpha}D_1^2)^2},$$

where $\tilde{\mu} \equiv \sum_{i\in I_1} \alpha_i \mu_i$. From these equalities we obtain

$$\nabla\kappa_z = -4\frac{3(1-\tilde{\alpha})^2 + ((1-\tilde{\alpha})^3 + \tilde{\alpha}^3)D_1^2}{(1-\tilde{\alpha}+\tilde{\alpha}D_1^2)^3}\frac{\tilde{\alpha}}{1-\tilde{\alpha}}D_1^3\tilde{\mu} + \frac{(1-\tilde{\alpha})^3+\tilde{\alpha}^3}{(1-\tilde{\alpha}+\tilde{\alpha}D_1^2)^2}\frac{4}{1-\tilde{\alpha}}D_1^3\tilde{\mu}$$

$$= 4\frac{1-6\tilde{\alpha}+6\tilde{\alpha}^2}{(1-\tilde{\alpha}+\tilde{\alpha}D_1^2)^3}D_1^3\tilde{\mu},$$

Define

$$\tilde{\lambda} \equiv d^T\nabla\kappa_z = 4\frac{1-6\tilde{\alpha}+6\tilde{\alpha}^2}{(1-\tilde{\alpha}+\tilde{\alpha}D_1^2)^3}\tilde{\alpha}D_1^4.$$

Using this value we have

$$\|\nabla_d\mathcal{L}(d,\tilde{\lambda})\| = \|\nabla\kappa_z - \tilde{\lambda}d\| = 4\left|\frac{(1-6\tilde{\alpha}+6\tilde{\alpha}^2)D_1^4}{(1-\tilde{\alpha}+\tilde{\alpha}D_1^2)^3}\right| \|\frac{1}{D_1}\tilde{\mu} - \tilde{\alpha}d\|.$$

From Assumption A3 it holds that $\|\tilde{\mu}\| \leq LD_1^2$, implying that

$$\|\nabla_d\mathcal{L}(d,\tilde{\lambda})\| \leq 4\left|\frac{(1-6\tilde{\alpha}+6\tilde{\alpha}^2)D_1^4}{(1-\tilde{\alpha}+\tilde{\alpha}D_1^2)^3}\right| (LD_1+\tilde{\alpha}) \to 0.$$

□

This theorem proves that if the groups are well separated the two-group projection directions will be found by the extreme directions of the kurtosis projections.

3.2 The Proposed Cluster Algorithm

The previous result suggests an iterative procedure to find the possible clusters, as follows: (1) The data are projected onto the directions of maximum and minimum kurtosis; (2) A criterion is applied to decide if the projected points

can be divided into two groups along these directions; (3) Assuming that the data are divided into two groups, consider each of the groups as new samples and apply to each of them steps (1) and (2); (4) The procedure is repeated until no more groups are identified.

These ideas are used to define to the following algorithm.

1. The algorithm starts by standardizing the sample data, $Z = \Sigma^{-1}(X - \mu)$.

2. Compute the directions d_{max} and d_{min} that maximizes and minimizes the kurtosis coefficient $\kappa(d)$ of the projected data $\{d^T Z\}$ and obtain the univariate projections of the standardized observations, $p_{max} = d_{max}^T Z$ and $p_{min} = d_{min}^T Z$.

3. Analyze in each of projections, p_{max} and p_{min} if we have a single distribution or a mixture of two distributions. This decision is made by fitting both a normal distribution and a mixture of two normals and comparing these fits using their BIC criteria.

   If the BIC value for the mixture of two distributions is greater than one for a single distribution, then assume heterogeneity.

   We may find heterogeneity in the two directions, only in one of them or in none of them. In the first case we choose the direction with larger BIC value and its associated two groups and go to the next step. In the second one, we select this direction and the associated groups and go to the next step. In the third case, we stop the procedure.

4. Consider the two groups found in step 3 as new data to be explored for heterogeneity and repeat steps 1 to 4 for the data in these two group until more groups are identified in the sample.

There are several important differences between this algorithm and the one proposed by Peña and Prieto (2001). First, the algorithm works in a binary way checking for heterogeneity in the projections, whereas in the PP algorithm several groups can be identified in one projection. Second, we only look at projections which are extremes of the kurtosis of the data we are studying for heterogeneity at each stage, whereas the PP algorithm searches on a set of $2p$ orthogonal directions obtained from the original data. Third, we check for heterogeneity by fitting a mixture of two normal distributions and comparing it with a single distribution instead of using the maximum gap found in the projections. We have verified through Monte Carlo experiments that this second approach is less effective than the proposed approach.

3.2.1 An Illustrative Example

To illustrate the procedure we present an example based on a sample obtained from a mixture of five populations with normal distribution and with the same covariance matrix, in dimension 10. The populations are generated as follows: populations 1 and 2 are generated on the first coordinate axis. The populations are separated by a distance $dst1$ as follows: the average of the population 1 is at a distance $dst1/2$ from the origin and the average of the population 2 is located at the same distance $dst1/2$ from the origin but in an opposite direction to population 1. Population 3 is generated on the second coordinate axis, at a distance $dst2$ from the origin. Population 4 is at a distance $dst3$ from the origin along a direction with an angle of 60°. Population 5 is at the same distance $dst3$ from the origin, but along a direction with an angle of 120°.

The parameters in the simulations are given in Table 3.1. The first population has 400 data, the second population 500, the third population 300, the fourth population 300 and the fifth population 500 data.

Parameter	
$n = 2000$	Number of total observations
$p = 10$	Dimension of the data
$\alpha_1 = 0.20$	Percentage of data in population 1
$\alpha_2 = 0.25$	Percentage of data in population 2
$\alpha_3 = 0.15$	Percentage of data in population 3
$\alpha_4 = 0.15$	Percentage of data in population 4
$\alpha_5 = 0.25$	Percentage of data in population 5
$dst1 = 6\sqrt{p}/\sqrt{2}$	Distance between the means of populations 1 and 2
$dst2 = 8\sqrt{p}/\sqrt{2}$	Distance from the origin to the mean of the population 3
$dst3 = 10\sqrt{p}/\sqrt{2}$	Distance from the origin to the means of the populations 4 and 5

Table 3.1: Simulation Parameters for a Mixture of Five Normal Populations with the Same Covariance Matrix

In figure 3.1 we plot the first two principal components and we can see that the populations are mixed.

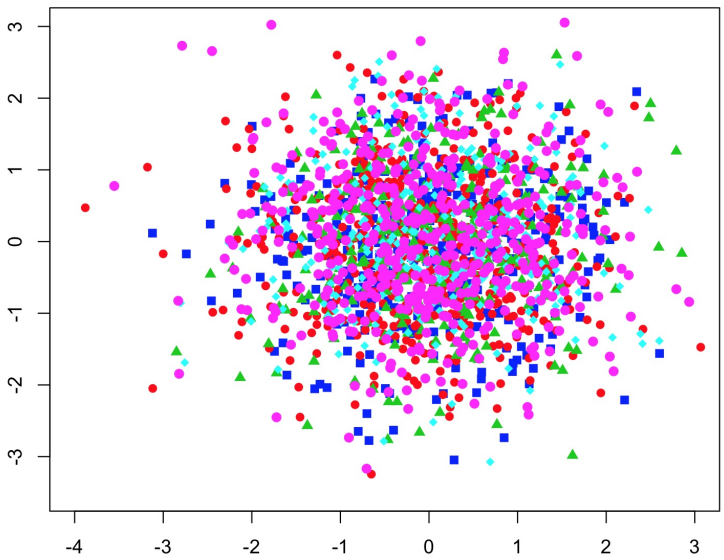

Figure 3.1: First Two Principal Components for Five Normal Populations Case

Applying the proposed Clustering Algorithm, we obtain the following results: with the maximum kurtosis direction ($\kappa = 3.07$), the BIC value for the fitting of a mixture of two normals is -5705.46 and for the fitting of one normal distribution is -5689.96. Since the BIC value for one distribution is greater than for two normal distributions, then heterogeneity is not identified with this direction. With the minimum kurtosis direction ($\kappa = 1.08$), the BIC value for the one normal distribution fitting is -5689.96 and for the fitting of a mixture of two normals is 122.68. Thus, the sample is separated into two blocks of groups: Group 1 contains 900 data and consists of 100% of populations 1 and 2, Group 2 contains 1100 data and consists of 100% of populations 3, 4 and 5. In Figure 3.2 we plot the projection of the data onto the direction of minimum kurtosis and we can see the separation of the sample into two blocks of groups.

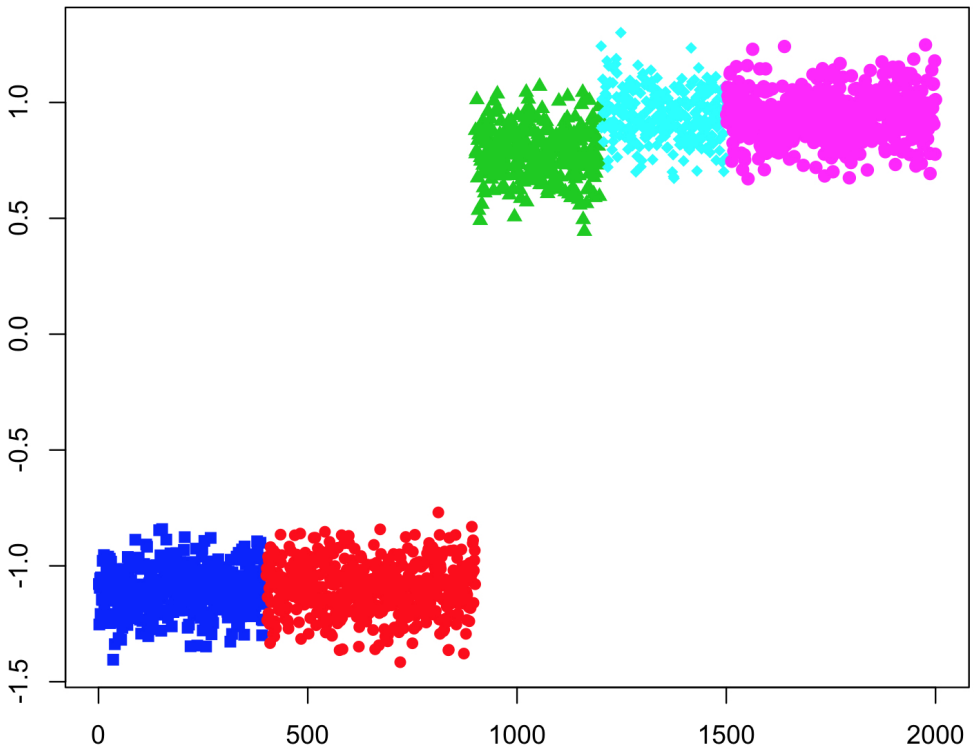

Figure 3.2: First Clustering Stage for Five Normal Populations

We then apply the procedure to Group 1. With the maximum kurtosis direction ($\kappa = 3.91$), the BIC value for the fitting of one normal distribution (-2566.70) is greater than for the fitting of two normals (-2577.30). Thus, heterogeneity is not assumed to exist in the projected data.

With the minimum kurtosis direction ($\kappa = 1.13$), the BIC value for the fitting of a mixture of two normal (-309.95) is greater than for the fitting for a single normal distribution (-2566.70). Therefore, Group 1 is separated into two subgroups. Subgroup 1 contains 400 data and is composed of 100% of population 1, Subgroup 2 contains 500 data and is composed of 100% of the population 2. In Figure 3.3 we plot the projection on the minimum kurtosis direction and we can see the separation of Group 1 into two subgroups.

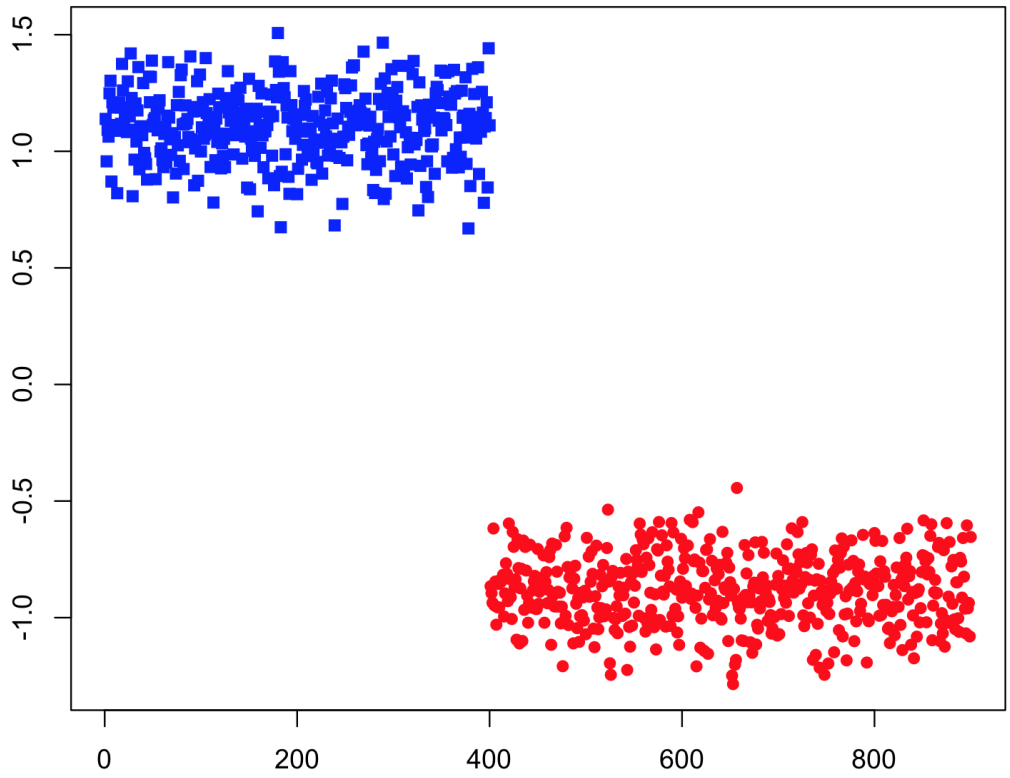

Figure 3.3: Second Clustering Stage for Five Normal Populations

Implementing the procedure for each of the subgroups obtained from Group 1, we have: for Subgroup 1 (400 data), with the maximum kurtosis direction ($\kappa = 3.38$), the BIC value for the fitting of one normal distribution is greater than for the fitting of a mixture of two normal distributions. A similar result is obtained for the minimum kurtosis direction ($\kappa = 2.21$), see Table (3.2). Then, the first subgroup is considered homogeneous.

Subgroup 2 (500 data) is also considered homogeneous, since for both the maximum kurtosis direction ($\kappa = 3.78$) and the minimum kurtosis direction ($\kappa = 2.24$), the BIC value for the one normal distribution fitting is greater than for the fitting of a mixture of two normals, see Table (3.2).

We now apply the method to Group 2: in the maximum kurtosis direction ($\kappa = 4.13$), the BIC value for the fitting of a mixture of two normals (-3128.801) is greater than for the fitting of a single normal distribution (-3134.67). With the minimum kurtosis direction ($\kappa = 1.46$), the BIC value for the fitting of a mixture of two normals (-2191.02) is greater than for the fitting for one normal distribution (-3134.67). Thus, we have found heterogeneity in the two directions. Then, we choose the direction with larger BIC value for a mixture of two normals. In this case, it corresponds to the minimum kurtosis direction. Using this direction, Group 2 is separated into two blocks of subgroups. Subgroup 3 contains 500 data and is composed of 100% of population 5. Subgroup 4 contains 600 data and consists of 100% of populations 3 and 4, see figure 3.4.

In Subgroup 3 (500 data), we obtain with the maximum kurtosis direction ($\kappa = 24.93$) that the BIC value for the fitting of one normal distribution is greater than for the fitting of a mixture of two normal distributions. A similar result is obtained for the minimum kurtosis direction ($\kappa = 1.87$), see Table (3.2). Therefore, Subgroup 3 is considered homogeneous.

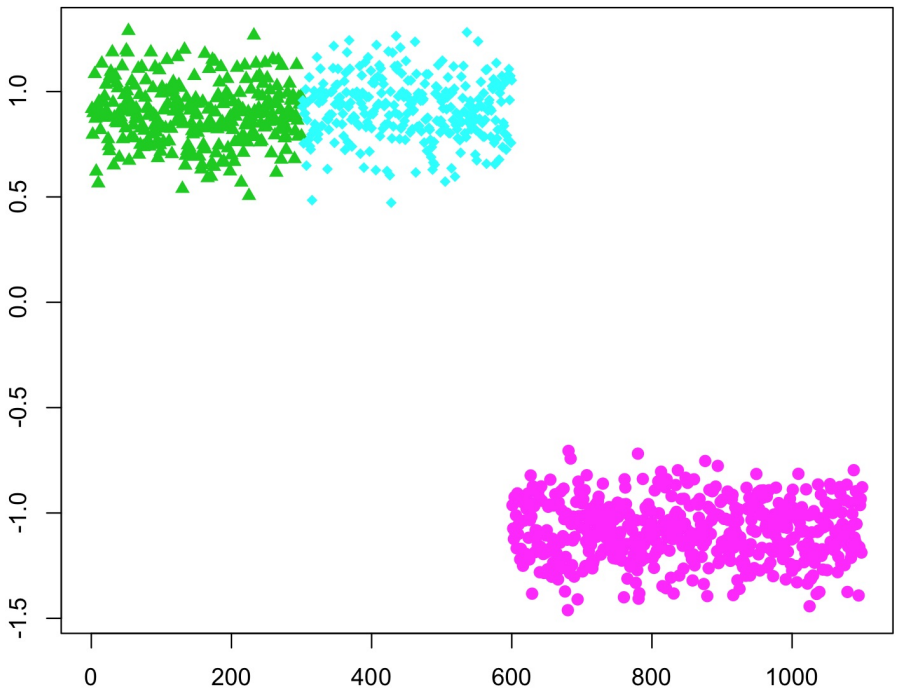

Figure 3.4: Third Clustering Stage for Five Normal Populations

For Subgroup 4 (600 data), we can see the BIC values for each direction in the Table (3.2). Since for both directions ($\kappa_{max} = 5.17$, $\kappa_{min} = 1.12$) the BIC value for the fitting of a mixture of two normal distributions is greater than for the fit of one normal distribution, then we have found heterogeneity. We choose the direction with the largest BIC value for a mixture of two normals. Using the minimum kurtosis direction, Subgroup 4 is separated into two (further) subgroups. Subgroup 5 contains 300 data and is composed of 100% of population 3 and Subgroup 6 contains 300 and is composed of 100% of population 4, see figure 3.5.

Finally, Subgroup 5 (300 data) and Subgroup 6 (300 data) are considered homogeneous, since for both subgroups the BIC value for the fitting of one normal distribution is greater than for the fitting of a mixture of two normals, both in the case of the maximum kurtosis direction and for the minimum kurtosis direction, see Table (3.2).

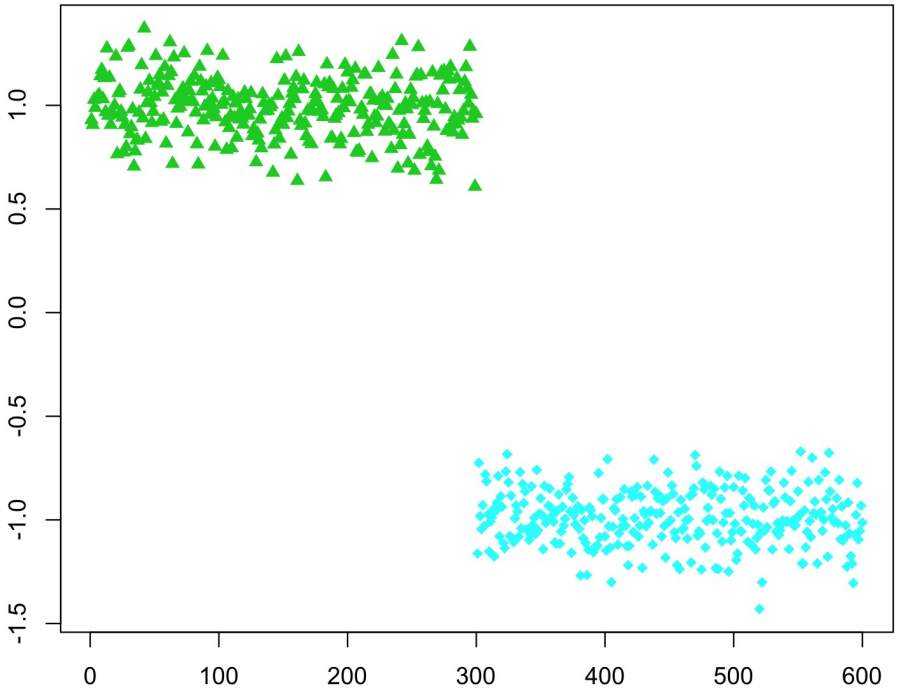

Figure 3.5: Fourth Clustering Stage for Five Normal Populations

	Maximum Kurtosis		**Minimum Kurtosis**	
	BIC_1	BIC_2	BIC_1	BIC_2
Subgroup 1	−1146.13	−1158.20	−1146.13	−1170.97
Subgroup 2	−1430.37	−1439.51	−1430.37	−1441.93
Subgroup 3	−1346.73	−1468.99	−1346.73	−1352.98
Subgroup 4	−1714.52	−1708.83	−1714.52	−486.14
Subgroup 5	−847.16	−851.98	−847.16	−861.77
Subgroup 6	−855.01	−866.23	−855.01	−861.77

Table 3.2: BIC Value for the One Normal Distribution Fitting (BIC_1) and BIC Value for the Fitting of a Mixture of Two Normal Distributions (BIC_2) for the Maximum and Minimum Kurtosis Directions in Each Subgroup

From the previous reasoning and from Figures 3.3, 3.4 and 3.5 we can conclude that the procedure has efficiently identified the existence of the five groups (Subgroup 1, Subgroup 2, Subgroup 3, Subgroup 5 and Subgroup 6).

3.3 Monte Carlo Experiment

In this section we present some computational results of a Monte Carlo experiment to compare the proposed algorithm with the Peñaa and Prieto Clustering Algorithm (2001a), the Mclust Algorithm of Fraley and Raftery (1999), CLARA and K-means.

We will consider three types of simulations. In the first one we compare the proposed algorithm with the algorithm of Peña and Prieto by studying a sample from by a mixture of three populations. This number of clusters has been chosen to show the details of the improvements of the proposed algorithm with respect to previous procedures. In the second one we compare the clustering results of the proposed algorithm with other cluster procedures such as MCLUST, CLARA and K-means. To determine the number of clusters, in K-means we used the method proposed by Hartigan (1975) and in CLARA we used "Silhouette", see Rousseeuw (1987). For the MCLUST algorithm we used general parameters for the distributions with the "VVV" option. In the third one we make this comparison for mixtures of different numbers of distributions generated from Normal, Uniform and Student-t distributions, and normal distributions contaminated by outliers.

We present several tables with the percentage of observations correctly grouped, obtained from 1000 replications for each model.

3.3.1 Comparing to the PP Kurtosis Algorithm

We will generate populations as follows: populations 1 and 2 are generated on the first coordinate axis. The populations are separated by a distance $dst1$ and the mean of population 1 is at a distance $dst1/2$ from the origin and the mean of population 2 is located at the same distance $dst1/2$ from the origin but in opposite direction to population 1. The population 3 is at a distance $dst2$ from the origin with an inclination angle. The angles that are used in the simulations are 30°, 60° and 90°. Figure 3.6 shows an example of data generated with this set-up.

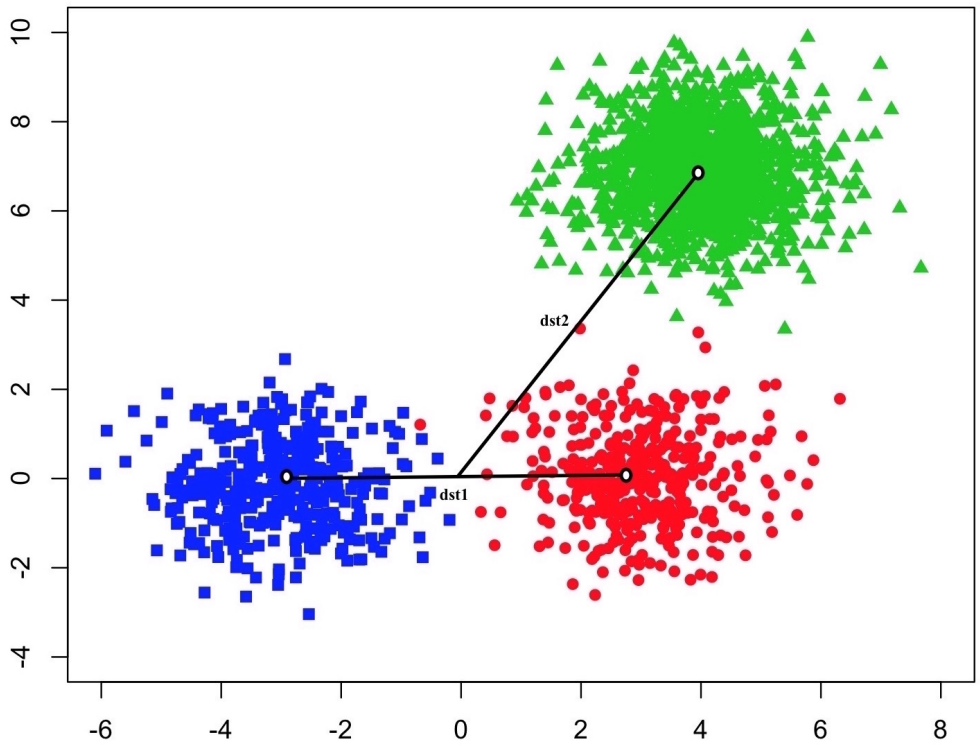

Figure 3.6: Original Data for Three Normal Populations

The parameters in the simulations are given in Table 3.3.

Our interest is to study the existence of clusters in the data using the kurtosis coefficient when the parameters α_1, α_2 and α_3 change.

Parameter	
n	Number of total observations
p	Dimension of the data
r	Cosine of the angle in which the population 3 is located
α_1	Percentage of data in population 1
α_2	Percentage of data in population 2
$\alpha_3 = 1 - (\alpha_1 + \alpha_2)$	Percentage of data in population 3
$dst1$: $6\sqrt{p}/\sqrt{2}$	Distance between the means of populations 1 and 2
$dst2$: $8\sqrt{p}/\sqrt{2}$	Distance from the origin to the mean of the population 3

Table 3.3: Simulation Parameters for a Mixture of Three Normal Populations with the Same Covariance Matrix

The cases that we will consider in the simulations are in Table 3.4.

Case	α_1	α_2	α_3
050590	0.05	0.05	0.90
101080	0.10	0.10	0.80
151570	0.15	0.15	0.70
201070	0.20	0.10	0.70
202060	0.20	0.20	0.60
301060	0.30	0.10	0.60
302050	0.30	0.20	0.50
401050	0.40	0.10	0.50
402040	0.40	0.20	0.40
303040	0.30	0.30	0.40

Table 3.4: Cases to Study for Three Normal Populations

Success Criteria

In order to compare the algorithms, we need assess the success for the clustering procedures. In the case of three groups the cluster detection is done in two stages. The first stage consists in the separation of the first two groups and in the second stage the missing group is detected. Therefore, the following criteria of success in the clustering during the two stages have been applied:

- **First stage**

  If two groups are obtained, we compare each group with the three original populations and analyze the coincidences. If one of the two groups obtained includes at least 80% of one of the initial populations and the other group at least 80% of another population, we consider that the clustering is successful during this stage. Otherwise, it is labeled as a failure.

- **Second stage**

  This stage only happens if the first stage is not a failure. The algorithm is applied to the two groups obtained in the first stage. We consider the clustering successful if one of the groups is divided into two subgroups and each subgroup includes at least 80% of one of the initial populations, and the other group is not divided into subgroups; note that from the first stage success criterion, it will include at least 80% of one of the initial populations. Otherwise the second stage, and the whole procedure, is considered to have failed.

Results Table

The results are presented in a table with the percentage representing the number of cases in which the clustering has been considered to be successful. The table is divided as follows: in the rows are the proportions $n/p = 10, 20, 50, 100$ for each p. The columns are divided into four: the first and second columns indicate p and the corresponding proportions. In the third and fourth columns the results of success obtained using the Peña and Prieto Clustering Algorithm and our proposed algorithm respectively are presented. These columns are divided into three columns corresponding to the angle at which the third population is located, which may be 30°, 60° y 90°. The results were obtained from 100 repetitions of the model.

		Average Success Rate					
		P&P			**MMx**		
p	Angle / n/p	30°	60°	90°	30°	60°	90°
10	**20**	0.15	0.45	0.36	0.84	0.87	0.85
	50	0.33	0.56	0.61	1	1	1
	100	0.40	0.69	0.70	1	1	1
	250	0.40	0.78	0.79	1	1	1
	Mean	**0.32**	**0.62**	**0.61**	**0.96**	**0.97**	**0.96**
20	**20**	0.16	0.25	0.26	0.75	0.85	0.85
	50	0.35	0.41	0.41	0.97	0.99	1
	100	0.45	0.49	0.50	1	1	1
	250	0.48	0.58	0.58	1	1	1
	Mean	**0.36**	**0.43**	**0.44**	**0.93**	**0.96**	**0.96**
50	**50**	0.10	0.12	0.11	0.39	0.41	0.43
	50	0.14	0.17	0.15	0.60	0.62	0.63
	100	0.20	0.25	0.23	0.94	0.93	0.92
	250	0.35	0.37	0.37	0.97	0.96	0.96
	Mean	**0.20**	**0.22**	**0.22**	**0.73**	**0.73**	**0.74**

Table 3.5: P&P Cluster Algorithm vs. Proposed Algorithm

In Table 3.5 we have the results comparing the proposed algorithm to the Peña and Prieto Clustering Algorithm. This table shows that the performance of both algorithms does not depend on the angle and that they improve with n/p, as expected. On the other hand, the new algorithm is more powerful than the previous one.

3.3.2 Comparing to Other Cluster Procedures

We perform the same simulation experiments with three normal populations for other methods commonly used in the literature such as MCLUST, CLARA and K-means. In Table 3.6 we present the average success rate in the coincidence with the original data using the different methods.

		Average Success Rate											
		Kurtosis			**MCLUST**			**CLARA**			**Kmeans**		
p	Angle / n/p	30°	60°	90°	30°	60°	90°	30°	60°	90°	30°	60°	90°
10	**20**	0.84	0.87	0.85	0.17	0.18	0.19	0.44	0.80	1	0.58	0.38	0.32
	50	1	1	1	0.78	0.78	0.79	0.44	0.80	1	0.58	0.42	0.37
	100	1	1	1	0.90	0.90	0.90	0.41	0.82	1	0.63	0.42	0.37
	250	1	1	1	1	1	1	0.40	0.83	1	0.67	0.42	0.39
	Mean	**0.96**	**0.97**	**0.96**	**0.71**	**0.71**	**0.72**	**0.42**	**0.81**	**1**	**0.61**	**0.41**	**0.36**
20	**20**	0.75	0.85	0.85	0	0	0	0.40	0.77	0.80	0.52	0.41	0.43
	50	0.97	0.99	1	0.40	0.40	0.40	0.41	0.78	0.70	0.63	0.38	0.41
	100	1	1	1	0.70	0.80	0.80	0.40	0.81	0.80	0.63	0.36	0.39
	250	1	1	1	0.80	0.90	0.90	0.40	0.83	0.90	0.59	0.34	0.38
	Mean	**0.96**	**0.96**	**0.93**	**0.48**	**0.52**	**0.52**	**0.40**	**0.80**	**0.80**	**0.59**	**0.37**	**0.40**

Table 3.6: Comparing to Other Cluster Procedures

From Table 3.6 we can conclude that our proposed algorithm is more efficient than other methods for detecting the three groups present in the sample. These other methods tend to fail for large p. In addition, our method has a computational advantage for large data dimensions. For example, for p greater than 20 we failed to obtain results from MCLUST.

3.3.3 Comparing to Other Algorithms for Normal, Uniform, Student-t Data and Normal with Outliers

In this section we present a more general comparison of the cluster methods. We generated sets of $20p$, $50p$ and $100p$ random observations in dimensions $p = 4, 8, 15, 30, 50$ from a mixture of k multivariate distributions (Normal, Uniform and Student-t with p degrees of freedom), with $k = 2, 3, 4, 8$. The number of observations in each population is determined randomly, but making sure that each cluster contains at least $2p$ observations.

The means for each distribution are chosen as values from a multivariate normal distribution $N(0, fI)$, see Table 3.7 for the values of f.

p	**4**				**8**				**15**				**30**				**50**			
k	2	3	4	8	2	3	4	8	2	3	4	8	2	3	4	8	2	3	4	8
f	14	18	22	38	20	25	31	54	27	35	43	74	38	49	60	104	49	64	78	134

Table 3.7: Factors f to Generate the Observations for the Simulations

The covariance matrices are generated as $S = UDU^T$, using a random orthogonal matrix U and a diagonal matrix D with entries from a uniform distribution on $[10^{-3}, 5\sqrt{p}]$, see Peña and Prieto (2001a).

In Tables 3.8, 3.9 and 3.10, we show for the proportion n/p the average percentage of the observations that have been labeled correctly. A more detailed description of these tables is found in the Appendix, where we present for each p and for the different values of k, the percentage of observations that coincide with the original groups, obtained from 1000 replications for each model. To provide better understanding of the behavior of the procedure, we compare the proposed method with the MCLUST, CLARA and K-means methods.

From the results in Tables, we can conclude that the proposed procedure works better than the other commonly used methods to identify several cluster in the sample. We can see that with our method the success in the clustering remains stable with the increase of the dimension, while the other methods are considerably affected.

The results show that K-means has the similar average success regardless of the distribution, MCLUST is method the most affected with the increase of the dimension and although CLARA works well, the results are better using our method.

In the Tables presented in the appendix we can see how the increase in the data dimension and in the number of groups affects each method. We can conclude that our proposal is more efficient when we have several clusters in the sample and the data dimension is high.

Finally, in Table 3.11 we have the results of a simulation study to determine the behavior of the methods in the presence of outliers. For this study, the data have been generated as indicated above for the normal case, but 10% of the data are now outliers. For each cluster in the sample, 10% of its observations have been generated as a group of outliers at a distance $4\chi^2_{p,0.99}$ in a group along a random direction, and a single outlier along another random direction.

The results show that our proposed procedure works better than the MCLUST, CLARA and K-means methods to detect clusters, even if we have outliers in the sample.

Multivariate Normal Distributions					
		Average Success Rate			
p	n/p	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
4	**20**	0.75	0.58	0.77	0.54
	50	0.97	0.79	0.69	0.52
	100	0.99	0.87	0.68	0.52
	Mean	**0.90**	**0.75**	**0.72**	**0.53**
8	**20**	0.78	0.47	0.78	0.55
	50	0.99	0.76	0.75	0.53
	100	1	0.87	0.72	0.54
	Mean	**0.92**	**0.70**	**0.75**	**0.54**
15	**20**	0.79	0.41	0.70	0.60
	50	1	0.66	0.82	0.53
	100	1	0.83	0.87	0.55
	Mean	**0.93**	**0.63**	**0.80**	**0.56**
30	**20**	0.75	0.37	0.70	0.54
	50	0.98	0.64	0.89	0.54
	100	1	0.82	0.84	0.50
	Mean	**0.91**	**0.61**	**0.81**	**0.53**
50	**20**	0.66	0.27	0.63	0.59
	50	0.96	0.60	0.75	0.56
	100	0.98	0.77	0.80	0.54
	Mean	**0.86**	**0.55**	**0.73**	**0.56**

Table 3.8: Average Success in Clustering for the proportion n/p using the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms. Normal Observations

Multivariate Uniform Distributions					
		Average Success Rate			
p	n/p	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
4	**20**	0.89	0.89	0.88	0.53
	50	0.99	0.98	0.81	0.56
	100	1	0.97	0.80	0.53
	Mean	**0.96**	**0.95**	**0.83**	**0.54**
8	**20**	0.92	0.90	0.90	0.54
	50	1	0.97	0.94	0.50
	100	1	0.98	0.94	0.53
	Mean	**0.97**	**0.95**	**0.93**	**0.52**
15	**20**	0.90	0.73	0.91	0.57
	50	1	0.89	0.89	0.54
	100	1	0.95	0.89	0.49
	Mean	**0.97**	**0.86**	**0.89**	**0.53**
30	**20**	0.81	0.55	0.81	0.56
	50	0.99	0.77	0.89	0.56
	100	1	0.78	0.90	0.51
	Mean	**0.93**	**0.70**	**0.87**	**0.54**
50	**20**	0.76	0.40	0.75	0.58
	50	0.98	0.65	0.78	0.55
	100	0.99	0.73	0.92	0.49
	Mean	**0.91**	**0.59**	**0.81**	**0.54**

Table 3.9: Average Success in Clustering for the proportion n/p using the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms. Uniform Observations

Multivariate Student-t Distributions					
		Average Success Rate			
p	n/p	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
4	**20**	0.66	0.50	0.48	0.46
	50	0.88	0.76	0.53	0.43
	100	0.93	0.77	0.53	0.38
	Mean	**0.82**	**0.68**	**0.51**	**0.43**
8	**20**	0.73	0.43	0.63	0.56
	50	0.96	0.71	0.66	0.51
	100	1	0.79	0.62	0.52
	Mean	**0.89**	**0.64**	**0.63**	**0.53**
15	**20**	0.73	0.39	0.74	0.57
	50	1	0.67	0.78	0.55
	100	1	0.74	0.80	0.50
	Mean	**0.91**	**0.60**	**0.77**	**0.54**
30	**20**	0.74	0.34	0.77	0.55
	50	0.96	0.65	0.74	0.53
	100	0.99	0.74	0.82	0.51
	Mean	**0.90**	**0.58**	**0.78**	**0.53**
50	**20**	0.72	0.28	0.65	0.59
	50	0.95	0.64	0.79	0.55
	100	0.99	0.75	0.84	0.51
	Mean	**0.88**	**0.56**	**0.76**	**0.55**

Table 3.10: Average Success in Clustering for the proportion n/p using the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms.

Student-t Observations

Normal Observations with Outliers					
		Average Success Rate			
p	n/p	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
4	**20**	0.74	0.34	0.70	0.50
	50	0.92	0.27	0.63	0.53
	100	0.95	0.17	0.53	0.49
	Mean	**0.87**	**0.26**	**0.62**	**0.50**
8	**20**	0.78	0.30	0.77	0.50
	50	0.98	0.21	0.73	0.55
	100	0.98	0.26	0.68	0.51
	Mean	**0.91**	**0.26**	**0.73**	**0.52**
15	**20**	0.75	0.33	0.80	0.52
	50	0.96	0.41	0.76	0.52
	100	0.98	0.44	0.71	0.54
	Mean	**0.90**	**0.39**	**0.75**	**0.52**
30	**20**	0.72	0.17	0.78	0.56
	50	0.93	0.56	0.79	0.55
	100	0.97	0.84	0.70	0.46
	Mean	**0.88**	**0.52**	**0.76**	**0.52**
50	**20**	0.71	0.13	0.72	0.52
	50	0.88	0.63	0.70	0.62
	100	0.98	0.85	0.66	0.49
	Mean	**0.85**	**0.53**	**0.69**	**0.54**

Table 3.11: Average Success in Clustering for the proportion n/p using the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms. Normal Observations with Outliers

3.4 Conclusion

In this Chapter we have presented an iterative binary clustering algorithm based on directions that project the observations onto two blocks. We have shown that these directions can be approximated from the extreme directions of kurtosis. These kurtosis directions have been shown to be asymptotically two-block projection directions.

Based on this property, we have defined our algorithm to we check for a mixture of two distributions using the BIC criterion, for each one of the projections of the data onto the directions of maximum and minimum kurtosis.

Finally, from some simulation examples, we shown that the algorithm with a mixture of normals is more efficient than the algorithm proposed by Peña and Prieto (2001a), MCLUST, CLARA and K-means models when the data dimension and the number of clusters present in the sample are large.

Chapter 4

Kurtosis for Functional Data Analysis

Summary

A large number of generalizations of multivariate techniques to the functional data case have been proposed in the literature in recent years. This chapter introduces an extension of clustering techniques based on multivariate kurtosis directions, to the analysis of functional data. This proposal is closely related to the one presented in the preceding chapter.

We analyze if our proposal preserves some of the good properties of kurtosis-based procedures applied to the multivariate case, regarding the identification of cluster structures.

We have also conducted an experimental analysis comparing the performance of the proposed algorithm with the results obtained by Functional Principal Components, Functional K-means and the Funclust method on simulated data.

4.1 A Review of Functional Data Analysis

An area of recent interest has been the development of new statistical procedures for Functional Data Analysis (FDA). FDA comprises all the statistical techniques developed for the analysis of curves or surfaces that vary in time. Initially, the research in this area was intended to be an almost direct extension of the techniques of classical multivariate analysis. However, the special structure associated to the functional data implies the need for adapted techniques, and motivates the development of new methodologies and procedures.

The purpose of any statistical analysis procedure in this setting is to make use of any time (or other independent variable) dependency structure associated to the functions generating the data, to obtain better estimates for those magnitudes of interest in the data. Functional data is very relevant in many fields of application of statistics such as health sciences, economics, environmental studies, among others.

Well-known references in the field of FDA are the books written by Ramsay and Silverman (1997) and Ferraty and Vieu (2006). In 2005, Ramsay and Silverman wrote a second book of a more applied character in which solutions to the problems associated to concrete datasets were studied. The same authors presented a considerable number of applications in another book, see Ramsay and Silverman (2002). Ramsay et al. (2009) includes many Functional Data Analysis applications and algorithmic implementations in R and MATLAB.

Functional data are inherently high-dimensional. Their numerical treatment requires reducing this dimension to a manageable size. This is usually done by approximating the functions through a representation in some appropriate (usually orthonormal) functional basis. A finite number K of elements in the basis are chosen to represent the data, transforming the problem into a multivariate setting. The choice of both the parameter K and the most appropriate basis for the observed data is a basic one in functional data analysis, but there seems to be

no universal rule providing an optimal selection.

The value K acts as a smoothing parameter for the functional data. If K is small we have a very tractable model but possibly relevant information is lost. While if K is big, the data are represented with high precision but computational complexity issues become relevant.

The most usual bases in functional data analysis are Fourier bases, B-Splines bases, Wavelets bases, exponential functions, or polynomial bases, among others, see Ramsay and Silverman (1997). The choice of basis may have an impact on the results obtained; this will be an issue we will consider in the experimental analysis of our proposal.

Regarding specific techniques, Ramsay and Silverman (1997) developed an adaptation of Principal Component Analysis to the functional case, the Functional Principal Component Analysis (FPCA) technique. This dimension reduction technique summarizes the information available in the data by identifying a finite set of scalar variables obtained as generalized linear combinations of the curves with maximum variance and has been used as the basic tool to analysing and clustering functional data. However, the technique has well-known shortcomings, such as a high sensitivity to the presence of outliers. Also, the summarizing combinations can be difficult to interpret and do not always provide a completely understandable presentation of the structure of the variability in the observed data.

Classification for functional data has been recently considered by several authors. One of the early references on the subject was that of Hastie et al. (1995). They adapt the general ideas from functional discriminant analysis, based on a penalized method for regularization. This setting allows them to cast the classification problem as a regression problem via optimal scoring.

In the framework of supervised classification some extensions have been made to the functional case. It is worth mentioning the study of Ferraty and Vieu

(2003), where they proposed a nonparametric supervised classification model by introducing a consistent kernel estimator applied to a sample of curves. López-Pintado and Romo (2006) considered the use of continuity information from the data and proposed robust procedures for the supervised classification of curves based on this information.

In the context of unsupervised classification, the one most relevant for this chapter, K-means was one of the first methods to be adapted to the functional case. Various implementations and variations have emerged, among them those by Abraham et al. (2003), where they propose a clustering method based on fitting the functional data using B-splines and partitioning the estimated model coefficients using a K-means algorithm. Biau et al. (2005) applied K-means in infinite dimensional Hilbert spaces by using a nonparametric method.

James and Sugar (2003) developed a flexible model-based procedure for clustering functional data. The technique can be applied to any type of data generated from curves, but is particularly useful when individuals are observed at a sparse set of time points. Also they extend the model to handle multiple functional and finite-dimensional covariates.

Serban and Wasserman (2005), proposed a technique for nonparametrically estimating and clustering a large number of curves called *CATS: Clustering After Transformation and Smoothing*. In this method they estimated the error due to the fact that we are clustering the estimated curves rather than the true ones. CATS is quite general, but they described and analyzed the method mostly in the context of microarray experiments.

Jacques and Preda (2014) presented a review of the different methods for functional data clustering, classifying them into three categories: (1) two-stage methods, which first reduce data dimension and then perform clustering. (2) Non-parametric methods, which use distances or dissimilarities between curves

combined with K-means or hierarchical clustering. (3) Model-based methods, which assume an approximation for the probability distribution of the functional data. But note the associated difficulty that for functional data the notion of probability density generally does not exist, see Delaigle and Hall (2010). Related to this last category, the authors proposed a model-based functional clustering method using the Karhunen-Loève expansion of a stochastic process to define an approximation to the probability density of the functional coefficients. The method is called *Funclust* and is available in **R**, see Jacques and Preda (2013).

The proposal described in the following sections introduces a method corresponding to the first class mentioned above: it first reduces the dimension of the data by projecting it onto certain functions related to a kurtosis operator, and then it studies the presence of clusters in the projections. It is closely related to the algorithm proposed in Chapter 3 for the multivariate analysis case, as it also uses the kurtosis information in the data to reduce its dimension. Furthermore, it applies the same model-based analysis to the resulting projections.

4.2 Description of the Kurtosis operator

Our proposal for a kurtosis-based clustering algorithm is based on an extension of multivariate kurtosis matrices to a functional data setting. In particular, it will be defined as an adaptation of the proposal presented in Peñaa et al. (2010) for the multivariate case. This proposal generated directions of interest from some eigenvectors of a kurtosis matrix, and then studied the univariate projections of the data to analyze the possible presence of clusters in the multivariate data.

Our multivariate reference, following Móri et al. (1993), will be the kurtosis matrix for a multivariate centered random variable X defined as

$$K = E[(X^T \Sigma^{-1} X) X X^T], \tag{4.1}$$

where $\Sigma = \mathrm{Var}(X)$. For clustering applications, this matrix has the property that one of its eigenvalues corresponds to the Fisher direction for a mixture of two multivariate normal populations, $\Sigma^{-1}(\mu_1 - \mu_2)$, see Theorem 1 in Peña et al. (2010), for example. This type of result is the one we aim to replicate in a functional data setting.

The following sections introduce a kurtosis operator for functional data related to this kurtosis matrix, and having similar properties with respect to clustering applications.

4.2.1 Defining a Kurtosis Operator for Functional Data

Our data are real functions defined on an interval $T \equiv [a, b] \subset \mathbb{R}$. We assume that these functions are realizations of a stochastic process with a finite mean function. We assume that the realizations of the associated centered process (after subtracting its mean function) belong to a Hilbert space $L^2(T)$. If u, v are in this Hilbert space, their inner product is defined as $\langle u, v\rangle = \int_a^b u(t)v(t)dt$. We will use the norm $\|u\|^2 = \langle u, u\rangle$.

Our proposal for a kurtosis operator starts from a stochastic process $\tilde{\xi}$ with finite mean. To simplify the notation, let $\xi \equiv \tilde{\xi} - m_\xi$, where $m_\xi(t) \equiv E[\xi(t)]$. We assume that ξ has a continuous and finite covariance operator. The Karhunen-Loève expansion of ξ is given by

$$\xi(t) = \sum_{i=0}^{\infty} B_i \phi_i(t), \tag{4.2}$$

where the B_i are independent random variables satisfying $E[B_i] = 0$, $E[B_i^2] = \lambda_i$ and ϕ_i are orthogonal deterministic functions on T, having unit norm.

The finiteness of the covariance operator is equivalente to assuming that the following condition holds:

A1. $\sum_i E[B_i^2] = \sum_i \lambda_i < \infty$.

Consider our reference multivariate kurtosis matrix (4.1). A difficulty to introduce a functional equivalent to this matrix is associated to the fact that, in general, the Mahalanobis distance may take infinite values in a functional setting. We propose to use some functional approximation to this Mahalanobis distance, a function φ_ξ, and a kurtosis operator based on it, of the form

$$k(s,t) \equiv E[\varphi_\xi(\xi)\xi(s)\xi(t)], \tag{4.3}$$

for some function $\varphi_\xi : L^2(T) \to \mathbb{R}_+$ satisfying

C1. $0 \leq \varphi_\xi(x) \leq C \max(C, \langle x, x\rangle)$

and some constant $C > 0$. If the function φ plays the role of the Mahalanobis distance $X^T\Sigma^{-1}X$, then the function k shows a direct correspondence to the definition of the matrix K in (4.1).

To ensure that this kernel k is well-defined, we introduce the following additional condition on the process ξ:

A2. $\sum_i E[B_i^4] < \infty$.

Condition A2 is equivalent to assuming that $E[\xi(s)^4] < \infty$ for all $s \in T$. Note that A2 holds for a gaussian process with finite covariance function.

Condition C1 implies

$$\begin{aligned} k(s,t) &\leq CE[\langle\xi,\xi\rangle\xi(s)\xi(t)] = CE\left[\sum_i B_i^2 \sum_j B_j\phi_j(s) \sum_k B_k\phi_k(t)\right] \\ &= C\sum_i E[B_i^4]\phi_i(s)\phi_i(t) + C\sum_{j\neq i} E[B_i^2]E[B_j^2]\phi_i(s)\phi_j(t) \\ &= C\sum_i \left(E[B_i^4] - E[B_i^2]^2\right)\phi_i(s)\phi_i(t) + C\left(\sum_i E[B_i^2]\phi_i(s)\right)\left(\sum_j E[B_j^2]\phi_j(t)\right) \leq C', \end{aligned}$$

for some constant C', where we have used that assumption A1 implies $\sum_i E[B_i^2]^2 < \infty$. Thus, the proposed function $k(s,t)$ is well-defined at all $s,t \in T$ for any $\xi \in L^2(T)$, as long as A1, A2 and C1 hold.

For any (deterministic) function $u(t) \in L^2(T)$, we denote the induced kurtosis operator as K,

$$K(u,t) \equiv E[\langle \xi, u \rangle \varphi_\xi(\xi)\xi(t)]. \tag{4.4}$$

As particular examples, for $\varphi_\xi(x) \equiv \langle x, x \rangle$ we have a definition of the kurtosis operator related to (4.1) if we replace Σ^{-1} with the identity, while if $\varphi_\xi(x) \equiv 1$ we have the covariance operator.

4.2.2 Optimal Classification Rules for a Mixture of Gaussian Processes

Our choice of a particular form for the function φ_ξ in (4.3) will be guided by our interest in ensuring good separation properties for the kurtosis operator K in (4.4). With this aim, in this section we extend some basic results related to optimal discriminant functions to the functional case.

Our definition of φ_ξ will be a consequence of the study of the optimal classification rules for functional observations in a particular case of a mixture of two normal multivariate distributions with the same covariance matrix, the reference case giving rise to the Fisher discriminant function. In the functional setting we consider that the (functional) data have been obtained from a mixture of two gaussian processes with the same covariance operator.

Consider a process $\bar{\xi}(t)$ defined as a mixture, with probability α, of two gaussian processes $\bar{\xi}_1$ and $\bar{\xi}_2$ with mean functions $\bar{m}_1$ and $\bar{m}_2$ and the same covariance function $r(s,t)$. Let ϕ_i denote the set of orthogonal eigenfunctions for $r(s,t)$ and $\lambda_i \geq 0$ their corresponding eigenvalues. Then

$$r(s,t) = \sum_{i=0}^{\infty} \lambda_i \phi_i(s)\phi_i(t), \qquad R(u)(t) = \int_T r(s,t)u(s)ds = \sum_{i=0}^{\infty} \lambda_i u_i \phi_i(t), \tag{4.5}$$

for $u = \sum_i u_i \phi_i$. Also, let

$$m(t) \equiv \bar{m}_1(t) - \bar{m}_2(t) = \sum_{i=0}^{\infty} m_i \phi_i(t).$$

We will use the notation $\bar{\xi}_1(t) = \bar{m}_1(t) + \sum_i B_{1i}\phi_i(t)$, $\bar{\xi}_2(t) = \bar{m}_2(t) + \sum_i B_{2i}\phi_i(t)$ and

$$\bar{\xi}(t) = \begin{cases} \bar{m}_1(t) + \sum_i B_{1i}\phi_i(t) & \text{w.p. } \alpha \\ \bar{m}_2(t) + \sum_i B_{2i}\phi_i(t) & \text{w.p. } 1-\alpha, \end{cases} \tag{4.6}$$

where B_{1i} and B_{2i} are independent standard normal random variables.

Our goal is to define a kurtosis operator (4.3), that is, to select a function φ_ξ, with the property that the eigenfunctions of k provide information useful to separate the two components in the mixture defined in (4.6). This is equivalent to using the eigenvectors of a kurtosis matrix in the multivariate case to construct the Fisher discriminant function, as described in Peña et al. (2010).

A first step will be to identify a separation criterion appropriate for the functional case. By analogy with the multivariate case, we will study the ratio of the variability between groups and the variability within the groups, for the functional observations projected onto a given function ψ, defined for our data model as

$$\Delta(\psi) \equiv \frac{BTG(\psi)}{WTG(\psi)} = \frac{\alpha(1-\alpha)\langle \psi, m_1 - m_2 \rangle^2}{\iint \psi(s)\psi(t)r(s,t)dsdt}.$$

A first result establishes an equivalence with the Fisher discriminant function in the multivariate case.

Lemma 2 *Assume that $\lambda_i > 0$ for all i. The function $\psi(t)$ that maximizes the value of Δ is given by*

$$\psi(t) = \sum_{i=0}^{\infty} \omega_i^* \phi_i(t), \qquad \omega_i^* = C\frac{m_i}{\lambda_i}, \tag{4.7}$$

for some constant C.

Proof For $\psi(t) = \sum_{i=0}^{\infty} \omega_i \phi_i(t)$, if we rewrite Δ in terms of the eigenfunctions and eigenvalues of k we have

$$\Delta(\psi) = \frac{\alpha(1-\alpha)(\sum_i m_i\omega_i)^2}{\sum_i \lambda_i \omega_i^2}.$$

If $\lambda_i > 0$, the first-order optimality conditions are

$$\frac{2\alpha(1-\alpha)(\sum_i m_i\omega_i)(\sum_i \lambda_i\omega_i^2)m_j - 2\alpha(1-\alpha)(\sum_k m_i\omega_i)^2\lambda_j\omega_j}{(\sum_i \lambda_i\omega_i^2)^2} = 0, \tag{4.8}$$

and

$$\frac{2\alpha(1-\alpha)(\sum_i m_i\omega_i)m_j}{\sum_i \lambda_i\omega_i^2} = 0.$$

otherwise.

The solutions for these equations are either $\sum_i m_i\omega_i = 0$ (the minimizer of the problem) or the one indicated in (4.7). □

The form of ψ in (4.7) is the direct equivalent of the vector $\Sigma^{-1}(\mu_1 - \mu_2)$ in the multivariate case. But this representation of ψ has the undesirable property of not having a bounded norm. As in practice we will work with a finite basis representation of the data, it seems interesting to ensure that the norm of the functions we use is bounded, to guarantee reasonable truncation properties. A modification of the preceding lemma is given below.

Lemma 3 *The function that solves the problem*

$$\max_{\psi} \Delta(\psi) \qquad s.t.\ \|\psi\| \leq V,$$

is given by

$$\psi(t) = \sum_{i=0}^{\infty} \omega_i^* \phi_i(t), \qquad \omega_i^* = C\frac{m_i}{\lambda_i + \delta}, \tag{4.9}$$

for some constants C and $\delta > 0$.

Proof If we rewrite $\Delta(\psi)$ in terms of the eigenfunctions and eigenvalues of k, as in (4.8), we have that the problem of interest in this case is

$$\max_{\omega} \frac{\alpha(1-\alpha)(\sum_i m_i\omega_i)^2}{\sum_i \lambda_i\omega_i^2} \qquad \text{s.t. } \sum_i \omega_i^2 \leq V^2.$$

Its first-order optimality conditions are

$$\frac{2\alpha(1-\alpha)(\sum_i m_i\omega_i)(\sum_i \lambda_i\omega_i^2)m_j - 2\alpha(1-\alpha)(\sum_i m_i\omega_i)^2\lambda_j\omega_j}{(\sum_i \lambda_i\omega_i^2)^2} + 2\mu\omega_j = 0.$$

The optimal solution in (4.9) follows from this equality. □

If there exists a smallest index i' such that $\lambda_i = 0$ for all $i > i'$, the problem has no (bounded) solution unless $m_i = 0$ for all $i > i'$, in which case we again obtain the preceding solution, with $\omega_i = 0$ for $i > i'$.

From this result, an interesting replacement for any computation carried out on the inverse covariance operator r^{-1} (such as computing a Mahalanobis distance) can be implemented by replacing it with a modified inverse covariance operator r_δ^{-1}, defined as

$$\begin{aligned} r_\delta^{-1}(s,t) &= \sum_{i=0}^{\infty} \frac{1}{\lambda_i + \delta}\phi_i(s)\phi_i(t), \\ S_\delta^{-1}(u)(t) &= \int r_\delta^{-1}(s,t)u(s)ds = \sum_{i=0}^{\infty} \frac{1}{\lambda_i + \delta}\langle\phi_i, u\rangle\phi_i(t), \end{aligned} \tag{4.10}$$

for some positive value δ. This modified operator can be considered as the inverse of a regularized version of the covariance operator S.

Using this operator, the optimal function ψ in (4.9) can be represented as

$$\psi(t) = C\int r_\delta^{-1}(s,t)m(s)ds = CS_\delta^{-1}(m). \tag{4.11}$$

4.2.3 The Proposed Kurtosis Operator

We specify in this section our proposal of a kurtosis operator having the form introduced in (4.4). This proposal will be inspired by the result obtained in Lemma 3, and it will be based on the regularized inverse covariance operator (4.10).

For a general stochastic process ξ with mean zero and covariance operator $S(u)$, we define our proposed function $\varphi_{\xi,\delta}$ as

$$\varphi_{\xi,\delta}(\xi') \equiv \langle \xi', S_\delta^{-1}(\xi') \rangle, \tag{4.12}$$

where the operator $S_\delta^{-1}(u)$ is defined according to (4.10). Our kurtosis operator, obtained by replacing (4.12) in (4.4), will be

$$K(u,t) \equiv E[\langle \xi, S_\delta^{-1}(\xi) \rangle \langle \xi, u \rangle \xi(t)]. \tag{4.13}$$

4.2.4 Discriminating Properties of Some Eigenfunctions of the Kurtosis Operator

We now relate the properties of our proposed kurtosis operator (4.13) to the optimal discriminating properties discussed in Section 4.2.2. Our main goal is the identification of relationships between the eigenfunctions of this kurtosis operator and the previously introduced optimal classification function (4.11).

As it was done in Section 4.2.2, due to the difficulties associated to conducting a theoretical study for a general case, our goal in this section focuses on verifying that the proposal introduced in (4.13) presents acceptable classification properties for the case of a mixture of gaussian processes.

Consider the process ξ defined in (4.6), where $m = m_1 - m_2$ denotes the difference of the mean functions, and $R(u) = \sum_i \lambda_i \langle \phi_i, u \rangle \phi_i$ denotes the common covariance operator for both populations. The covariance operator of ξ is given by $S(u) = R(u) + \alpha(1-\alpha)\langle m, u \rangle m$.

If we define S_δ^{-1} according to (4.10), it holds that

$$R_\delta^{-1}(u) \;=\; (R + \delta I)^{-1}(u) \tag{4.14}$$

$$\begin{aligned} S_\delta^{-1}(u) &= (S+\delta I)^{-1}(u) = (R+\delta I+\alpha(1-\alpha)\langle m,\cdot\rangle m)^{-1}(u) \qquad (4.15)\\ &= R_\delta^{-1}(u) - \gamma\langle R_\delta^{-1}(m), u\rangle R_\delta^{-1}(m) \\ \gamma &= \frac{\alpha(1-\alpha)}{1+\alpha(1-\alpha)\langle m, R_\delta^{-1}(m)\rangle}, \end{aligned}$$

where $R_\delta^{-1} = (R+\delta I)^{-1}$.

For ξ defined in this manner, φ in (4.12) satisfies

$$\varphi_{\xi,\delta}(\xi') = \langle \xi', R_\delta^{-1}(\xi')\rangle - \gamma\langle \xi', R_\delta^{-1}(m)\rangle^2. \qquad (4.16)$$

To simplify the presentation of the proof, we first derive a result providing a representation for the expected value of a fourth-order moment of ξ.

Lemma 4 *For ξ defined in (4.6), $i = 1, 2$, it holds that*

$$E[\langle \xi_i - m_i, S_\delta^{-1}(\xi_i - m_i)\rangle\langle \xi_i - m_i, u\rangle(\xi_i - m_i)] = \rho_1 R(u) - \rho_2(u)m - 2\delta Q_\delta(u), \quad (4.17)$$

for some ρ_1, $\rho_2(u)$ and $Q_\delta(u)$.

Proof To simplify the notation, we will use $\bar{\xi}_i \equiv \xi_i - m_i$, and also

$$T_i(u) \equiv E[\langle \xi_i - m_i, S_\delta^{-1}(\xi_i - m_i)\rangle\langle \xi_i - m_i, u\rangle(\xi_i - m_i)].$$

We have from (4.16)

$$\begin{aligned} T_i(u) &= E\left[\langle \bar{\xi}_i, S_\delta^{-1}(\bar{\xi}_i)\rangle\langle \bar{\xi}_i, u\rangle\bar{\xi}_i\right] \\ &= E\left[\langle \bar{\xi}_i, R_\delta^{-1}(\bar{\xi}_i)\rangle\langle \bar{\xi}_i, u\rangle\bar{\xi}_i\right] - \gamma E\left[\left(\langle \bar{\xi}_i, R_\delta^{-1}(m)\rangle\right)^2\langle \bar{\xi}_i, u\rangle\bar{\xi}_i\right]. \end{aligned}$$

Using the Karhunen-Loève expansion of $\bar{\xi}_i$, $\bar{\xi}_i = \sum_k B_{ik}\phi_k$, the representation $u = \sum_k u_k\phi_k$ and the normal distribution of B_{ik}, implying $E[B_{ik}^2] = \lambda_k$,

$E[B_{ik}^4] = 3\lambda_k^2$, and $R_\delta^{-1}(u) = \sum_j \frac{1}{\lambda_j+\delta}\langle\phi_j, u\rangle\phi_j$ we have

$$\begin{aligned} T_{i1}(u) &\equiv E\left[\langle\bar{\xi}_i, R_\delta^{-1}(\bar{\xi}_i)\rangle\langle\bar{\xi}_i, u\rangle\bar{\xi}_i\right] = \sum_{jkl}\frac{E[B_{ij}^2 B_{ik}B_{il}]}{\lambda_j+\delta}u_k\phi_l \\ &= \sum_j \frac{3\lambda_j^2}{\lambda_j+\delta}u_j\phi_j + \sum_{l\neq j}\frac{\lambda_j\lambda_l}{\lambda_j+\delta}u_l\phi_l \\ &= 2\sum_j\frac{\lambda_j^2}{\lambda_j+\delta}u_j\phi_j + \left(\sum_j\frac{\lambda_j}{\lambda_j+\delta}\right)R(u) \\ &= \left(2+\sum_j\frac{\lambda_j}{\lambda_j+\delta}\right)R(u) - 2\delta R_\delta^{-1}(R(u)). \end{aligned}$$

Also,

$$\begin{aligned} T_{i2}(u) &\equiv E\left[\langle\bar{\xi}_i, R_\delta^{-1}(m)\rangle^2\langle\bar{\xi}_i, u\rangle\bar{\xi}_i\right] = \sum_{kl}E\left[\left(\sum_j\frac{B_{ij}m_j}{\lambda_j+\delta}\right)^2 B_{ik}B_{il}\right]u_k\phi_l \\ &= \sum_j\frac{3\lambda_j^2m_j^2}{(\lambda_j+\delta)^2}u_j\phi_j + \sum_{l\neq j}\frac{m_j^2\lambda_j\lambda_l}{(\lambda_j+\delta)^2}u_l\phi_l + 2\sum_{l\neq j}\frac{m_j\lambda_j m_l\lambda_l}{(\lambda_j+\delta)(\lambda_l+\delta)}u_j\phi_l \\ &= \sum_j\frac{m_j^2\lambda_j}{(\lambda_j+\delta)^2}R(u) + 2\left(\sum_j\frac{m_j\lambda_j}{\lambda_j+\delta}u_j\right)\left(\sum_l\frac{m_l\lambda_l}{\lambda_l+\delta}\phi_l\right) \\ &= \langle R_\delta^{-1}(m), R_\delta^{-1}(R(m))\rangle R(u) + 2\langle R_\delta^{-1}(R(u)), m\rangle R_\delta^{-1}(R(m)) \\ &= \langle R_\delta^{-1}(m), R_\delta^{-1}(R(m))\rangle R(u) + 2\langle R_\delta^{-1}(R(u)), m\rangle m \\ &\quad - 2\delta\langle R_\delta^{-1}(R(u)), m\rangle R_\delta^{-1}(m), \end{aligned}$$

where we have used $R_\delta^{-1}(R(m)) = m - \delta R_\delta^{-1}(m)$.

Collecting these results we obtain

$$\begin{aligned} T_i(u) &= \left(2+\sum_j\frac{\lambda_j}{\lambda_j+\delta} - \gamma\langle R_\delta^{-1}(m), R_\delta^{-1}(R(m))\rangle\right)R(u) \\ &\quad - 2\gamma\langle R_\delta^{-1}(R(u)), m\rangle m \\ &\quad - 2\delta\left(R_\delta^{-1}(R(u)) - \gamma\langle R_\delta^{-1}(R(u)), m\rangle R_\delta^{-1}(m)\right). \end{aligned}$$

This equality implies that, letting

$$\begin{aligned} \rho_1 &\equiv 2 + \sum_j \frac{\lambda_j}{\lambda_j + \delta} - \gamma \langle R_\delta^{-1}(m), R_\delta^{-1}(R(m)) \rangle \\ \rho_2(u) &\equiv 2\gamma \langle R_\delta^{-1}(R(u)), m \rangle \\ Q_\delta(u) &\equiv R_\delta^{-1}(R(u)) - \gamma \langle R_\delta^{-1}(R(u)), m \rangle R_\delta^{-1}(m), \end{aligned} \tag{4.18}$$

we have the desired result. □

We now present our main result, relating the optimal separation functions studied in Lemma 3, with some functions related to eigenfunctions of the kurtosis operator (4.13).

The multivariate case has been the reference for our functional kurtosis operator. In that case the Fisher discriminant direction is associated to an eigenvector of the kurtosis matrix. Our next result explores this relationship for the functional case and our proposed kurtosis operator.

It holds that in the functional case, due to the approximation we have introduced in the definition of the Mahalanobis distance, and the need to approximate the Fisher discriminant direction in a functional setting, the preceding equivalence only holds asymptotically. The Fisher discriminant function ψ defined in (4.9) satisfies an approximate generalized eigenvalue equation based on the kurtosis operator, with an error term that is arbitrarily small as the regularization parameter δ goes to zero.

This result provides theoretical support to use eigenfunctions obtained from the eigenvalue equation as approximations to the Fisher discriminant function. These functions are thus useful to reveal the presence of clusters by analysing the univariate projections of the data onto them.

Theorem 3 *For the mixture of two gaussian processes with zero mean ξ, defined in (4.6), let K be the operator defined in (4.13). Also, let R_δ^{-1}, S_δ^{-1} be defined as*

in (4.14)–(4.15). Consider a function $\psi(t)$ of the form

$$\psi = S_\delta^{-1}(m).$$

For $\tilde{\psi} = \psi/\|\psi\|$ it holds that

$$K(\tilde{\psi}) = \beta_\delta S(\tilde{\psi}) - \delta v(\delta), \tag{4.19}$$

for some functions of δ, β_δ and $v(\delta)$.

Furthermore, if for any $\delta > 0$,

$$\langle m, R_\delta^{-1}(m)/\|R_\delta^{-1}(m)\|\rangle \geq L > 0, \tag{4.20}$$

holds for some constant L, then

$$\|v(\delta)\| \leq L' \max(\beta_\delta, 1) \tag{4.21}$$

for some constant L' and any $\delta > 0$.

Proof We start by obtaining a representation of $K(u)$ for our mixture of two gaussian processes. From (4.13) we have

$$\begin{aligned} K(u) &= E[\langle \xi, S_\delta^{-1}(\xi)\rangle\langle \xi, u\rangle \xi] \\ &= \alpha E[\langle \xi_1, S_\delta^{-1}(\xi_1)\rangle\langle \xi_1, u\rangle \xi_1] + (1-\alpha)E[\langle \xi_2, S_\delta^{-1}(\xi_2)\rangle\langle \xi_2, u\rangle \xi_2] \\ &= \alpha K_1(u) + (1-\alpha)K_2(u), \end{aligned}$$

where we have introduced $K_i \equiv E[\langle \xi_i, S_\delta^{-1}(\xi_i)\rangle\langle \xi_i, u\rangle \xi_i]$. Defining $\bar{\xi}_i \equiv \xi_i - m_i$, for each of these operators we have

$$\begin{aligned} K_i &= E[\langle \bar{\xi}_i, S_\delta^{-1}(\bar{\xi}_i)\rangle\langle \bar{\xi}_i, u\rangle \bar{\xi}_i] + \langle m_i, S_\delta^{-1}(m_i)\rangle E[\langle \bar{\xi}_i, u\rangle \bar{\xi}_i] + 2\langle m_i, u\rangle E[\langle \bar{\xi}_i, S_\delta^{-1}(m_i)\rangle \bar{\xi}_i] \\ &\quad + 2E[\langle \bar{\xi}_i, S_\delta^{-1}(m_i)\rangle\langle \bar{\xi}_i, u\rangle]m_i + \langle m_i, u\rangle E[\langle \bar{\xi}_i, S_\delta^{-1}(\bar{\xi}_i)\rangle]m_i + \langle m_i, S_\delta^{-1}(m_i)\rangle\langle m_i, u\rangle m_i, \end{aligned}$$

where we have used the symmetry of the distribution of $\bar{\xi}_i$ to cancel the first- and third-order terms in this expansion. This expression is equivalent to

$$\begin{aligned} K_i &= E[\langle \bar{\xi}_i, S_\delta^{-1}(\bar{\xi}_i)\rangle\langle \bar{\xi}_i, u\rangle \bar{\xi}_i] + \langle m_i, S_\delta^{-1}(m_i)\rangle R(u) + 2\langle m_i, u\rangle R(S_\delta^{-1}(m_i)) \\ &\quad + \left(2\langle R(S_\delta^{-1}(m_i)), u\rangle + \langle m_i, u\rangle E[\langle \bar{\xi}_i, S_\delta^{-1}(\bar{\xi}_i)\rangle] + \langle m_i, S_\delta^{-1}(m_i)\rangle\langle m_i, u\rangle\right) m_i. \end{aligned}$$

Letting $m_i = \omega_i m$, $\omega_1 = 1-\alpha$, $\omega_2 = -\alpha$, from (4.15),

$$\begin{aligned} S_\delta^{-1}(m_i) &= \omega_i S_\delta^{-1}(m) = \omega_i(1-\gamma\langle m, R_\delta^{-1}(m)\rangle)R_\delta^{-1}(m) \\ R(S_\delta^{-1}(m_i)) &= \omega_i(1-\gamma\langle m, R_\delta^{-1}(m)\rangle)R(R_\delta^{-1}(m)) \\ &= \omega_i(1-\gamma\langle m, R_\delta^{-1}(m)\rangle)(m-\delta R_\delta^{-1}(m)). \end{aligned} \tag{4.22}$$

Replacing the result from Lemma 4 we obtain

$$\begin{aligned} K_i &= (\rho_1 + \langle m_i, S_\delta^{-1}(m_i)\rangle)R(u) \\ &\quad + (\omega_i^2(\langle m,u\rangle(1-\gamma\langle m, R_\delta^{-1}(m)\rangle) + 2\langle R(S_\delta^{-1}(m)),u\rangle + \langle m,u\rangle E[\langle\bar{\xi}_i, S_\delta^{-1}(\bar{\xi}_i)\rangle] \\ &\quad +\omega_i^2\langle m, S_\delta^{-1}(m)\rangle\langle m,u\rangle) - 2\rho_2(u))m \\ &\quad -\delta\left(2Q_\delta(u) + \omega_i^2(1-\gamma\langle m, R_\delta^{-1}(m)\rangle)\langle m,u\rangle R_\delta^{-1}(m)\right). \end{aligned}$$

As a consequence, using $\alpha\omega_1^2 + (1-\alpha)\omega_2^2 = \alpha(1-\alpha)$,

$$\begin{aligned} K(u) &= \bar{\rho}_1 R(u) + \bar{\rho}_2(u)m - \delta\left(2Q_\delta(u) + \alpha(1-\alpha)(1-\gamma\langle m, R_\delta^{-1}(m)\rangle)\langle m,u\rangle R_\delta^{-1}(m)\right) \\ &= \bar{\rho}_1 R(u) + \bar{\rho}_2(u)m - \delta\left(2Q_\delta(u) + \gamma\langle m,u\rangle R_\delta^{-1}(m)\right), \end{aligned} \tag{4.23}$$

where

$$\begin{aligned} \bar{\rho}_1 &= \rho_1 + \alpha(1-\alpha)\langle m, S_\delta^{-1}(m)\rangle \\ \bar{\rho}_2(u) &= \alpha(1-\alpha)\left((1-\gamma\langle m, R_\delta^{-1}(m)\rangle)\langle m,u\rangle + 2\langle R(S_\delta^{-1}(m)),u\rangle + \langle m,u\rangle E[\langle\bar{\xi}_i, S_\delta^{-1}(\bar{\xi}_i)\rangle]\right) \\ &\quad + (\alpha(1-\alpha)^4 + (1-\alpha)\alpha^4)\langle m, S_\delta^{-1}(m)\rangle\langle m,u\rangle - 2\rho_2(u). \end{aligned}$$

Consider now $\psi = S_\delta^{-1}(m)$. From (4.23), it holds that

$$K(\psi) = \bar{\rho}_1 R(S_\delta^{-1}(m)) + \bar{\rho}_2(\psi)m - \delta\left(2Q_\delta(\psi) + \gamma\langle m,\psi\rangle R_\delta^{-1}(m)\right),$$

and using (4.22) we obtain

$$\begin{aligned} K(\psi) &= \left(\bar{\rho}_1(1-\gamma\langle m, R_\delta^{-1}(m)\rangle) + \bar{\rho}_2(\psi)\right)m - \delta\left(2Q_\delta(\psi) + \gamma\langle m,\psi\rangle R_\delta^{-1}(m)\right) \\ &= \beta_\delta m - \delta\left(2Q_\delta(\psi) + \gamma\langle m,\psi\rangle R_\delta^{-1}(m)\right), \end{aligned} \tag{4.24}$$

for

$$\beta_\delta \equiv \bar{\rho}_1(1 - \gamma\langle m, R_\delta^{-1}(m)\rangle) + \bar{\rho}_2(\psi).$$

As

$$S(\psi) = S(S_\delta^{-1}(m)) = m - \delta S_\delta^{-1}(m) \Rightarrow m = S(\psi) + \delta S_\delta^{-1}(m) = S(\psi) + \delta\psi,$$

we can write (4.24) in the form

$$\begin{aligned} K(\psi) &= \beta_\delta S(\psi) + \delta\left(\beta_\delta\psi - 2Q_\delta(\psi) - \gamma\langle m, \psi\rangle R_\delta^{-1}(m)\right) \\ \Rightarrow K(\tilde{\psi}) &= \beta_\delta S(\tilde{\psi}) + \delta\left(\beta_\delta\tilde{\psi} - 2R_\delta^{-1}(R(\tilde{\psi})) + \gamma\langle(2R_\delta^{-1}(R(\tilde{\psi})) - \tilde{\psi}), m\rangle R_\delta^{-1}(m)\right) \end{aligned} \quad (4.25)$$

where we have introduced $\tilde{\psi} = \psi/\|\psi\|$, and replaced Q_δ using (4.18). The desired result follows for

$$v(\delta) \equiv \beta_\delta\tilde{\psi} - 2R_\delta^{-1}(R(\tilde{\psi})) + \gamma\langle(2R_\delta^{-1}(R(\tilde{\psi})) - \tilde{\psi}), m\rangle R_\delta^{-1}(m). \quad (4.26)$$

Studying the sizes of the terms in (4.26) and using (4.20), it holds that

$$\begin{aligned} \gamma\|R_\delta^{-1}(m)\| &= \frac{\alpha(1-\alpha)\|R_\delta^{-1}(m)\|}{1 + \alpha(1-\alpha)\langle m, R_\delta^{-1}(m)\rangle} \leq \frac{1}{\langle m, R_\delta^{-1}(m)/\|R_\delta^{-1}(m)\|\rangle} \leq \frac{1}{L} \\ \|R_\delta^{-1}(R(\tilde{\psi}))\|^2 &= \sum_i \frac{\lambda_i^2\tilde{\psi}_i^2}{(\lambda_i + \delta)^2} \leq \sum_i \tilde{\psi}_i^2 = 1, \end{aligned}$$

implying

$$\|v(\delta)\| \leq \beta_\delta + 2 + \frac{3\|m\|}{L},$$

and establishing the desired result for the size of $\|v(\delta)\|$ given in (4.21). □

This result provides theoretical support for a clustering procedure defined through the following basic steps: i) obtain a function u from the solution of the generalized eigenvalue equation $K(u) = \lambda S(u)$; ii) project the functional observations onto u and study the clustering properties of the resulting univariate projections. In the following section we present the details of a proposed implementation for this procedure.

4.3 Implementation of the Proposed Clustering Algorithm for Functional Data

In this section we describe the procedure that implements the proposed clustering algorithm for functional data. This implementation consists of two main steps:

1. Obtain projection functions from eigenfunctions of the kurtosis operator defined in Section 4.2.3.

2. Analyze the projected (univariate) observations to detect the possible presence of clusters in the data.

4.3.1 Implementation of the Proposed Kurtosis Operator

Our first step in the algorithm requires computing a projection function ψ, obtained from the equation $K(\psi) = \lambda S(\psi)$. This function will be related to the optimal separation function, as shown in Lemma 3 and Theorem 3.

We assume we are given a sample of multivariate observations, generated from a functional data model. These data have the form

$$\hat{x}_i(t_j), \quad i = 1, \ldots, n, \quad t_j \in [a, b], \quad j = 0, \ldots, p \tag{4.27}$$

The projection functions obtained from the kurtosis operator defined in (4.13) are computed by performing a series of steps on the data, which are described below:

1. *Representation.* We wish to use the structure of the data as functional objects, to improve on any results obtained from any direct treatment of the data as multivariate objects. In particular for our case, to perform a cluster

analysis. Our first step will be to find a reasonable functional representation for our data.

To obtain this representation, we start by selecting a functional basis. Let $\phi_k(t)$ for $t \in [a, b]$ and $k = 1, ..., m$ denote a truncated basis. We select a value for m providing a reasonable balance between precision and computational complexity.

We obtain values for a set of coefficients $\tilde{b}_{ik}$ (using regularized least-squares, or some other related method) such that

$$\hat{x}_i(t) \approx \tilde{x}_i(t) = \sum_k \tilde{b}_{ik}\phi_k(t). \tag{4.28}$$

Extending the usual matrix notation to this (mixed) case, we will write $\tilde{x} \equiv \tilde{B}\phi$ to represent the preceding equality, for $\tilde{x}$ and ϕ vectors of functions in $[a, b]$ and $\tilde{B} \in \mathbb{R}^{n\times m}$. We assume the vector ϕ having as components the m functions $\phi_k(t)$ corresponding to our chosen basis, and the vector $\hat{x}$ having as components the n (smoothed) sample functions $\tilde{x}_i(t)$.

2. *Centering the functional data.* We subtract the mean from the data,

$$\begin{aligned} \bar{x}(t) &= \frac{1}{n}\sum_i \tilde{x}_i(t) = \frac{1}{n}\sum_{ik} \tilde{b}_{ik}\phi_k(t) = \sum_k \left(\frac{1}{n}\sum_i \tilde{b}_{ik}\right)\phi_k(t) \\ x_i(t) &= \tilde{x}_i(t) - \bar{x}(t) = \sum_k \left(\tilde{b}_{ik} - \frac{1}{n}\sum_l \tilde{b}_{lk}\right)\phi_k(t) = \sum_k b_{ik}\phi_k(t), \end{aligned}$$

for $b_{ik} \equiv \tilde{b}_{ik} - \frac{1}{n}\sum_l \tilde{b}_{lk}$.

This operation can be written using the preceding compact matrix form as $x(t) = (I - \frac{1}{n}ee^T)\tilde{B}\phi(t) = B\phi(t)$, for $B \equiv (I - \frac{1}{n}ee^T)\tilde{B}$.

3. *Kurtosis operator.* Let W denote the (invertible) matrix with elements $w_{ij} = \langle\phi_i, \phi_j\rangle$. For an arbitrary function $u = \gamma^T\phi$ we have that

$$S(u) = \frac{1}{n}\sum_i \langle x_i, u\rangle x_i = \frac{1}{n}\sum_i \langle b_i^T\phi, \gamma^T\phi\rangle b_i^T\phi = \gamma^T W\left(\frac{1}{n}B^TB\right)\phi = \gamma^T C\phi,$$

where b_i denotes the i-th column of B^T and we have defined $C \equiv W\left(\frac{1}{n}B^TB\right)$. From this result, it holds that

$$\begin{aligned} S_\delta^{-1}(u) &= \gamma^T (C+\delta I)^{-1}\phi \\ S_\delta^{-1}(x_i) &= b_i^T (C+\delta I)^{-1}\phi \\ \langle x_i, S_\delta^{-1}(x_i)\rangle &= b_i^T W (C+\delta I)^{-1} b_i. \end{aligned}$$

For the kurtosis operator defined in (4.13),

$$\begin{aligned} K(u) &= \frac{1}{n}\sum_i \langle x_i, S_\delta^{-1}(x_i)\rangle\langle x_i, u\rangle x_i(t) \\ &= \frac{1}{n}\sum_i b_i^T W (C+\delta I)^{-1} b_i \gamma^T W b_i b_i^T \phi. \end{aligned} \tag{4.29}$$

We can write this equality in more compact form as

$$K(u) = (K_f\gamma)^T\phi, \quad K_f \equiv \frac{1}{n}B^TD_fBW \tag{4.30}$$

where K_f is a matrix that represents the proposed functional kurtosis operator, when we use a finite basis representation of our functional data. This matrix is defined in terms of a diagonal matrix D_f with entries $(D_f)_{ii} = b_i^T W (C+\delta I)^{-1} b_i$, that is,

$$D_f \equiv \mathrm{diag}(BW (C+\delta I)^{-1} B^T).$$

4. *Eigenfunctions.* From Theorem 3, the eigenfunction and eigenvalue having optimal separation properties can be approximated from a solution of

$$K(\psi) = \lambda S(\psi).$$

Or in our equivalent matrix form, if $\psi \equiv \bar{\gamma}^T\phi$, this function can be obtained by solving the generalized eigenvalue problem

$$B^TD_fBW\bar{\gamma} = \lambda B^TBW\bar{\gamma}.$$

The same solution can be obtained from the eigenvalue problem

$$\left(B^TB\right)^{-1}B^TD_fBW\bar{\gamma}=\lambda W\bar{\gamma}.$$

It may be more efficient numerically to work with the symmetric matrix

$$K_s=\left(B^TB\right)^{-1/2}B^TD_fB\left(B^TB\right)^{-1/2},$$

and the eigenfunctions of interest will be obtained from the eigenvectors of this matrix, $\hat{\gamma}$, as

$$K_s\hat{\gamma}=\lambda\hat{\gamma},\qquad \bar{\gamma}=W^{-1}\left(B^TB\right)^{-1/2}\hat{\gamma}.$$

This representation of the eigenvectors and eigenvalues of the modified kurtosis operator allows for an interesting comparison with the direct application of the kurtosis matrix proposed by Móri et al. (1993) to the coefficients in the representation of the functional data, b_{ik}.

From (4.1), the multivariate kurtosis matrix corresponding to the centered coefficients B is given by

$$K_m=\frac{1}{n}\sum_{i=1}^{n}\left(b_i^T\left(\frac{1}{n}B^TB\right)^{-1}b_i\right)b_ib_i^T=\frac{1}{n}B^TD_mB, \tag{4.31}$$

where $D_m=\mathrm{diag}(B(\frac{1}{n}B^TB)^{-1}B^T)$.

If we compare (4.31) and (4.30), we have that

$$\begin{aligned}K_f &= \frac{1}{n}B^T\mathrm{diag}\left(BW\left(W\frac{1}{n}B^TB+\delta I\right)^{-1}B^T\right)BW\\ K_m &= \frac{1}{n}B^T\mathrm{diag}\left(B\left(\frac{1}{n}B^TB\right)^{-1}B^T\right)B.\end{aligned}$$

From these representations, we conclude that the application of the functional kurtosis procedure based on the matrix K_f is identical to using the Móri

multivariate kurtosis matrix K_m, computed from the values of the (centered) coefficients in the basis expansion of the functional data B, that is, letting $B = X$ in (4.1), whenever the following conditions hold: i) $W = I$, we represent our data using an orthonormal basis, and ii) $\delta = 0$, we introduce no regularization in the definition of our operator.

4.3.2 Cluster Analysis of the Univariate Observations

Once the eigenfunctions of the kurtosis operator have been obtained from the eigenvectors of the matrix K_s, as described in Section 4.3.2, we project the centered functional data onto the functions associated to its maximum and minimum eigenvalues.

We perform a cluster analysis on the projected data by studying the existence of a mixture of two normal distributions on each of the projections, in a manner similar to that implemented in our algorithm for the multivariate case, following the procedure described in Section 3.2. We assume heterogeneity if the BIC value for the fitting of a mixture of two normal distributions is greater than for the fitting of one normal distribution. If two groups are obtained, they are considered as new data to be explored for heterogeneity and we repeat the procedure until no more groups are identified in the sample, see Section 3.2 for additional details.

4.4 Computational Results

In this Section we present several results obtained from the application of the proposed kurtosis clustering algorithm to functional data.

Our algorithm has been implemented on the R package *fda*, which includes some utilities for Functional Data Analysis, using both B-splines and Fourier functional bases. We have conducted simulation experiments, and we have also

used publicly available datasets such as *Growth*, *ECG* and the *CanadianWeather* data set. For more details about these data sets, see Ramsay and Silverman (2005).

The implementation of the procedure to simulated data has been conducted as described in Section 4.3. We have compared our method with Functional Principal Components, Functional K-means and the Funclust method. The R packages used to run each method are: *fda*, *fda.usc* and *Funclustering*, respectively.

4.4.1 Real Data Study

Canadian Daily Weather

The *CanadianWeather* data set consists of daily measurements at 35 Canadian weather stations, divided into four climate zones. For this example, we have compared our classification results to these four distinct classes specified in the database: Atlantic, Pacific, Continental and Arctic.

The observation locations and the climate regions are plotted on the map of Canada shown in Figure 4.1, where the black diamonds correspond to the Arctic zone stations, the red ones to Atlantic stations, the green ones to Continental stations and the blue ones to Pacific stations.

We have used B-Spline and Fourier bases to represent the data. After applying our procedure to estimate the kurtosis operator eigenfunctions, we have projected the data onto the two directions of maximum and minimum kurtosis, as well as those associated to the two largest functional principal components. The results are shown in Figures 4.2 and 4.3.

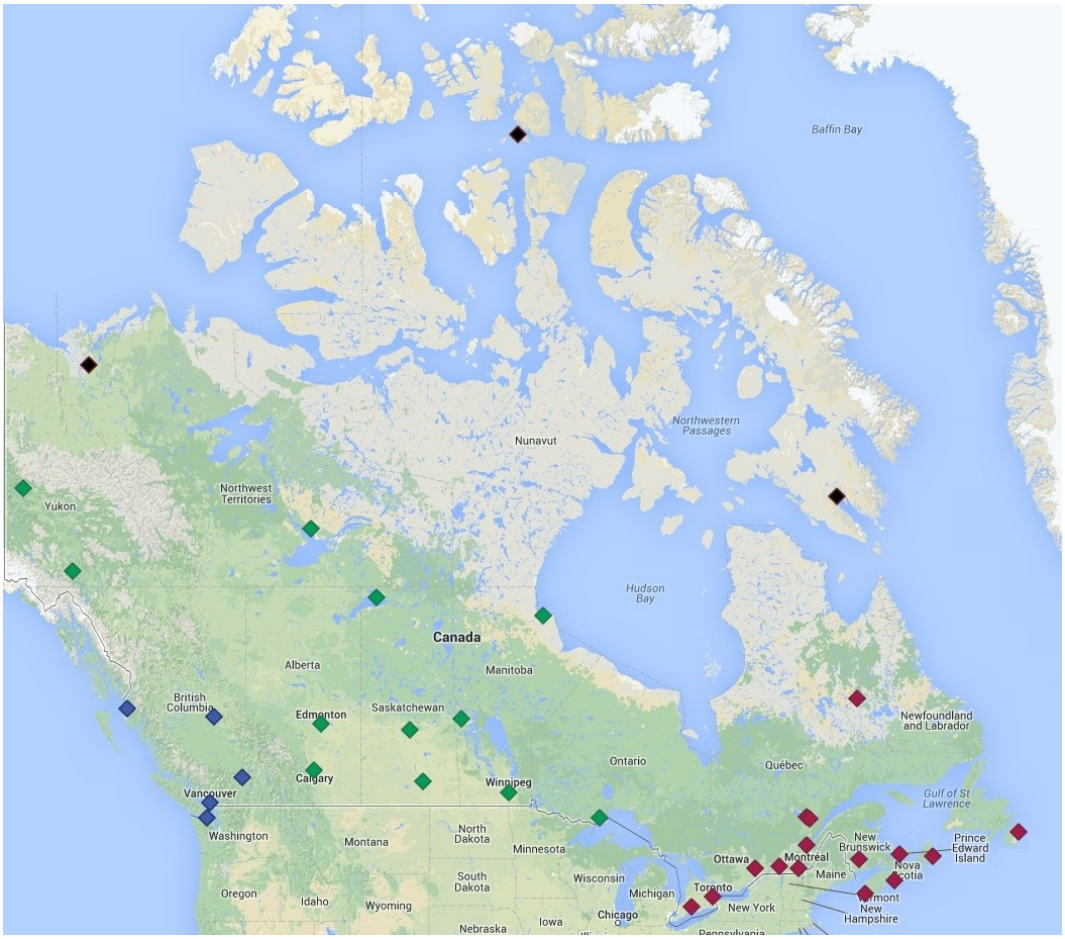

Figure 4.1: Canadian Weather Regions

Our results provide a much better, although not perfect, separation between the observations corresponding to different regions, when compared to the groupings that could be obtained from the principal components directions. In particular, the Atlantic, Continental and Pacific regions are clearly separated by the minimum kurtosis directions, while the Arctic region can be separated using the maximum kurtosis directions.

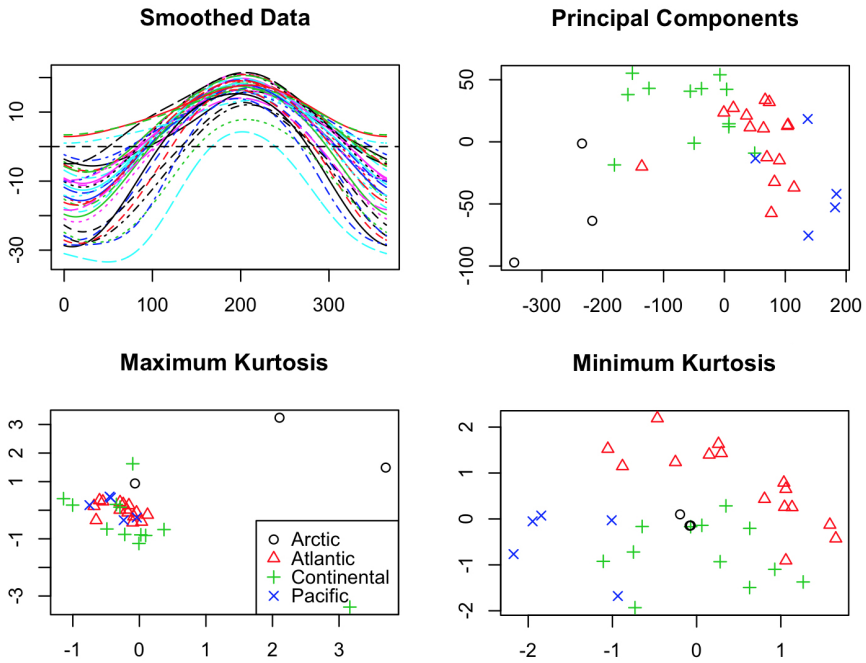

Figure 4.2: Fourier Basis to Represent Canadian Weather Regions

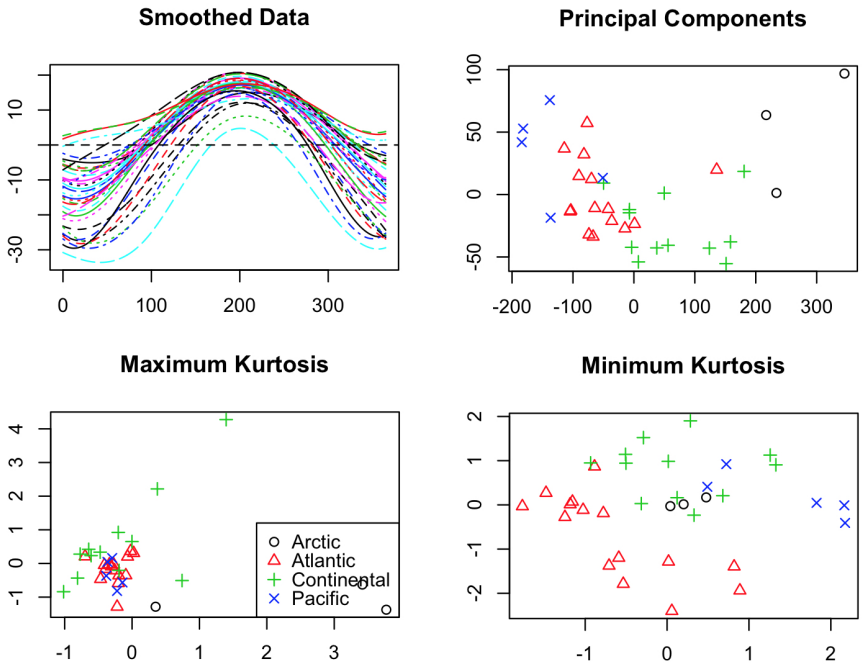

Figure 4.3: B-Spline Bases to Represent Canadian Weather Regions

Growth Data

The *Growth* data set consists of 93 curves: the heights of 39 boys and 54 girls as a function of their age. The heights were measured in 31 stages, between 1 and 18 years. Figure 4.4 shows the smoothed curves using 5 Fourier bases. we wish to identify the clusters present in the sample and to determine if the resulting clusters correspond to the different genders.

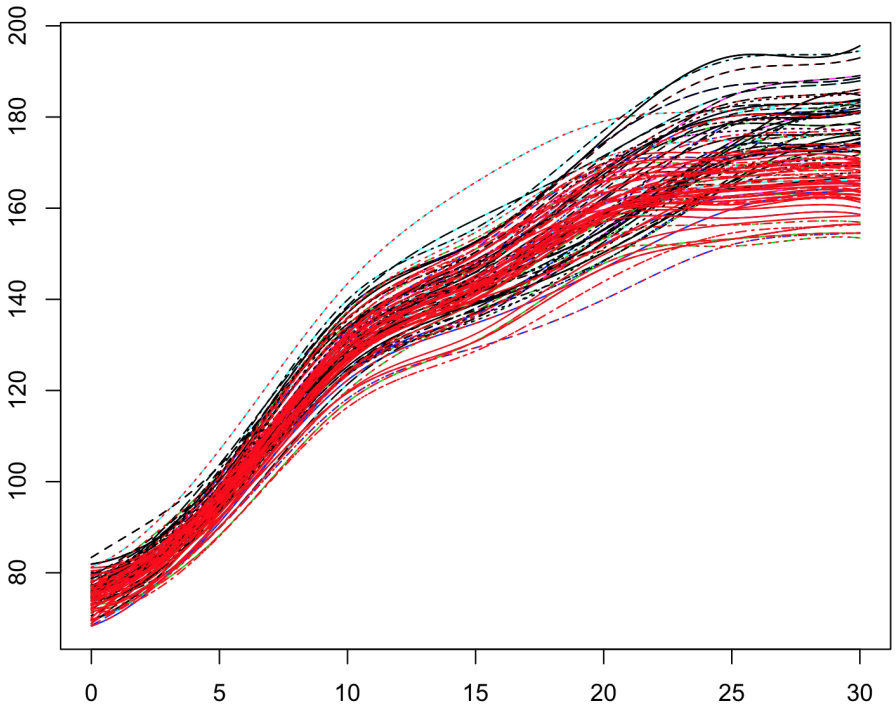

Figure 4.4: Growth Data with 5 Fourier Basis

In Tables 4.1 and 4.2 we present the percentage of successful identifications for each group (gender differences) using extreme kurtosis directions, Functional Principal Components, Funclust and Functional K-means. We have performed several simulations with different base sizes, for Fourier and B-Spline bases.

Base	Method	Boys	Girls	Procedure
B-Spline	**Kurtosis**	0.98	0.88	**0.93**
	Pr. Comp.	0.59	0.70	**0.64**
	FunClust	0.44	0.90	**0.67**
	Kmeans	0.59	0.69	**0.64**
Fourier	**Kurtosis**	0.98	0.85	**0.92**
	Pr. Comp.	0.59	0.69	**0.64**
	FunClust	0.39	0.94	**0.66**
	Kmeans	0.71	0.48	**0.59**

Table 4.1: Success in Clustering with Growth Data for the Proposed Procedure, Functional Principal Components, Funclust and Functional K-means Methods Using 5 Functional Basis (B-Spline or Fourier)

Base	Method	Boys	Girls	Procedure
B-Spline	**Kurtosis**	0.95	0.85	**0.90**
	Pr. Comp.	0.59	0.70	**0.64**
	FunClust	0.35	0.90	**0.62**
	Kmeans	0.58	0.70	**0.64**
Fourier	**Kurtosis**	0.89	0.87	**0.88**
	Pr. Comp.	0.59	0.69	**0.64**
	FunClust	0	0	**0**
	Kmeans	0.59	0.72	**0.65**

Table 4.2: Success in Clustering with Growth Data for the Proposed Procedure, Functional Principal Components, Funclust and Functional K-means Methods Using 9 Functional Basis (B-Spline or Fourier)

ECG Data

The *ECG* dataset consists of 200 electrocardiograms from 2 patient groups (133 normal and 67 abnormal) at 96 time instants. For this example we have used 21 B-spline basis functions to represent the data. In Figure 4.5 the smoothed curves are represented and Table 4.3 shows the percentage of successful identifications for the two patient groups.

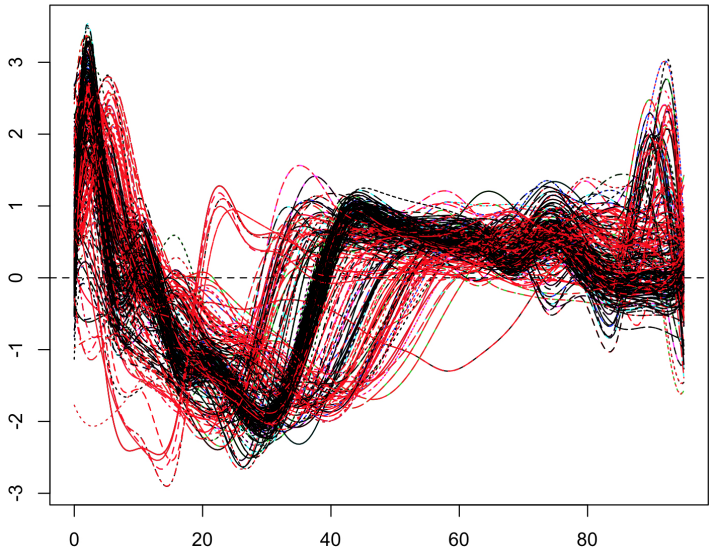

Figure 4.5: ECG Data with 21 B-Spline Basis

Base	Method	Normal	Abnormal	Procedure
B-Spline	**Kurtosis**	0.70	1	**0.85**
	Pr. Comp.	0.85	0.52	**0.69**
	FunClust	0.42	1	**0.71**
	Kmeans	0.73	0.60	**0.66**

Table 4.3: Success in Clustering with ECG Data for the Proposed Procedure, Functional Principal Components, Funclust and Functional K-means Methods Using 21 B-Spline Basis

From the results shown in Tables 4.1, 4.2 and 4.3, we conclude that our proposed method is more efficient than other methods commonly used for functional data clustering, to identify the gender difference in the *Growth* data and the two groups of patients in the *ECG* data.

4.4.2 Simulation Study (Gaussian Processes)

We have performed two sets of simulation studies with the aim of comparing the performance of our proposed kurtosis-based method with Functional Principal Components, Functional K-means and the Funclust method for unsupervised functional data clustering. The comparisons have been carried out by applying the methods to samples generated from different mixtures of gaussian processes. The models have been selected as they are simple ones and allow us to verify the existence of a good fit with the theoretical results presented in Section 4.2.4. The analysis of the results should provide us with interesting insights on the behavior of the proposed method in a controlled environment.

In both cases the two populations of gaussian processes have been defined to share the same quadratic covariance operator, $(\exp(-(x-y)^2/2l^2))$, with parameter $l = 15$. The same numbers of observations have been generated from each group ($n_1 = n_2 = n/2$). The observations have been obtained for $t \in [1, T]$ with $T = 20$. 20 equidistant observations of each process in $[1, 20]$ have been selected, with observation noise $\epsilon_{it} \sim N(0, 0.1)$. The values obtained are multivariate vectors in $\mathbb{R}^{20}$.

Both simulation examples differ in the choice of mean functions for each group, and in the preprocessing of the information before applying our proposed procedure.

We have used both Fourier and B-spline bases to represent (and smooth) the data. From the smoothed data we have obtained the directions corresponding to

the two largest eigenfunctions obtained using Functional Principal Components. We have also obtained the two directions corresponding to the smallest eigenfunctions of the kurtosis operator. We have projected our data onto these two pairs of directions.

To analyze the results, we have measured inter- vs. intra-group variability in the projections for each of the two groups, by comparing the traces of the corresponding covariance matrices. We have also checked the classification results. Finally, for one example we have prepared a graphical representation of the clusters obtained by using principal component and kurtosis directions, to illustrate how the kurtosis directions may be more efficient for cluster identification.

The basis used to represent data (Fourier or B-spline), the number of basis functions used and the number of observations for each group are modified between experiments. Each simulation experiment has been replicated 1000 times.

Simulation 1

In the first set of simulations we have used as mean functions for the two groups $m_i(t) = \sin(2\pi\mu_i t/T)$, $i = 1, 2$. The values μ_i are selected as -2.2 and 2, respectively.

In this case we wish to test if our method behaves reasonably well when the variability information has been removed from the data. To do that, and before fitting the data to our chosen bases, we have introduced a linear transformation on the multivariate data so that the mean of the transformed sample is equal to zero and its covariance matrix is the identity. We might expect principal components to have some difficulty separating the two modified groups; but note that functional principal components will work on the functional representation of the data (which has not been modified), and should still capture some of that variability information. Our main interest is to check that kurtosis is able to

identify the groups by using information beyond that of the variability in the data available through the covariance matrix.

Simulation 1. Fourier Basis Using a Fourier basis and the sample values described above, we obtain the results shown in Table 4.4 for inter- vs. intra-group variability.

Bases	n	**Variability Kurtosis**	**Variability PC**
7	**70**	0.68	0.06
	140	0.79	0.06
	280	0.90	0.06
15	**150**	0.41	0.01
	300	0.60	0.01
	600	0.75	0.01

Table 4.4: Inter- vs. Intra-group Variability in Kurtosis and Principal Components Projections Using Fourier Basis in Simulation 1 (The Variability Information has been Removed From the Data)

From the results in Table 4.4, it is interesting to note that the intra-group variability information is captured by the relevant directions of the kurtosis operator. The principal components operator searches for global variability, and mostly misses the intra-group variability information that would be most interesting for clustering applications. Thus, the theoretical properties of the kurtosis operator have a direct translation in practice to the capture of information relevant for cluster analysis.

Table 4.5 presents the success percentage in clustering when the projections have been obtained using Kurtosis and Functional Principal Components

directions. In this second case, and in a manner similar to the proposed kurtosis algorithm, we have analyzed if we have the mixture of two distributions according to the BIC criteria for each projection direction. In the table we also include the results obtained when using Functional K-means and the Funclust method.

To compare the methods, we have used the following success criteria for clustering: if two groups are identified, we compare each group with the two original populations and analyze the coincidences. If one of the groups includes at least 50% of the observations from one of the initial populations and the other group includes at least 50% of the observations in the second population, we consider that the clustering algorithm has been successful.

Bases	n	**Kurtosis**	**Pr. Comp.**	**Kmeans**	**FunClust**
7	**70**	0.82	0.28	0.16	0.32
	140	1	0.51	0.18	0.34
	280	1	0.58	0.24	0.41
15	**150**	0.22	0.25	0.02	0.46
	300	0.94	0.33	0.06	0.64
	600	1	0.42	0.10	0.71

Table 4.5: Success in Clustering for the Proposed Procedure, Functional Principal Components, Funclust and Functional K-means Using 7 and 15 Fourier Basis in Simulation 1 (The Variability Information has been Removed From the Data)

The results obtained from the kurtosis algorithm are better than those obtained for the principal components directions. Also, the results are better than those obtained with Functional K-means and Funclust method, implying a clear advantage of the use of kurtosis directions for functional data clustering.

Figure 4.6 shows the plots corresponding to $n = 280$. The projections have been

obtained for the directions that minimize the kurtosis and principal components, using 7 functions in the basis representation.

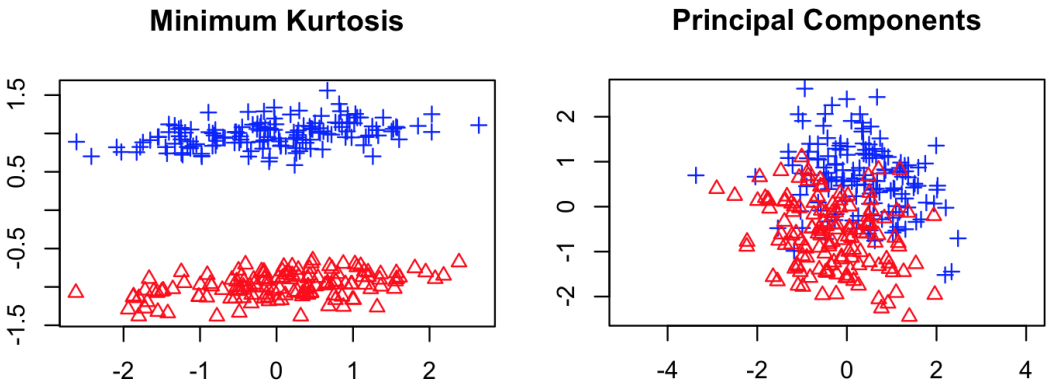

Figure 4.6: Simulation 1 with 7 Fourier Basis and $n = 280$

Simulation 1. B-Spline Bases For the next set of results we use a B-Splines basis and the same samples as in the preceding experiment. We analyze the success from the application of the algorithms as in the previous example. Table 4.6 presents the proportion of successful identifications.

Bases	n	**Kurtosis**	**Pr. Comp.**	**Kmeans**	**FunClust**
7	**70**	0.40	0.32	0.15	0.59
	140	0.93	0.67	0.18	0.57
	280	1	0.89	0.25	0.78
15	**150**	0.24	0.30	0.09	0.61
	300	0.93	0.44	0.11	0.78
	600	1	0.54	0.15	0.87

Table 4.6: Success in Clustering for the Proposed Procedure, Functional Principal Components, Funclust and Functional K-means Using 7 and 15 B-Spline Bases in Simulation 1 (The Variability Information has been Removed From the Data)

The results obtained for the proposed kurtosis method using a B-spline basis are interestingly worse than those using a Fourier basis. We believe this may be due to the basis providing a worse representation for the objects of interest (mean functions, covariance operator). This implies that the performance of the kurtosis operator may be sensitive to the choice of basis, at least in some cases, although this dependence would require a more detailed analysis.

Simulation 2

We conduct a second experiment, similar to the preceding one, where we use mean functions equal to zero for the first group, and $0.2\cos(2\pi t/(T/r))$, for $r = 1.5$. Again, we have used both a Fourier and a B-Splines basis; in both simulations the number of functions chosen for the basis is equal to 7. We have not included other basis sizes, as the preceding experiment seemed to indicate that this was a reasonable choice.

In this case we have not carried out any additional transformation of the multivariate data. Our goal is to test how well our proposed method performs when compared with functional principal components, if variability information is available in the covariance matrix to help classify the data. In this case we still expect our method to perform reasonably well, as we are using a model under which we have shown the proposed method has good classification properties. We wish to compare how much difference there may be between the use of the functional principal component directions and the kurtosis directions to reveal heterogeneity in the data.

Using a Fourier base and the values mentioned above, we present in Table 4.7 the percentage of success in the clustering results using our proposed method and Functional Principal Components, including also the results for the Functional K-means and Funclust methods.

Bases	n	**Kurtosis**	**Pr. Comp.**	**Kmeans**	**FunClust**
7	**70**	0.31	0.16	0.22	0.20
	140	0.81	0.16	0.13	0.44
	280	1	0.11	0.06	0.50
	1400	1	0.09	0.01	0.53

Table 4.7: Success in Clustering for the Proposed Procedure, Functional Principal Components, Funclust and Functional K-means Using 7 Fourier Bases in Simulation 2 (The Variability Information has not been Removed From the Data)

We can see that we again obtain significantly improved results with respect to Functional Principal Components, Functional K-means and Funclust method. It seems interesting to note that the performance of the proposed method improves markedly with the sample size.

In Table 4.8 we present the percentage of success in the clustering using a B-Splines basis and the values mentioned above.

Bases	n	**Kurtosis**	**Pr. Comp.**	**Kmeans**	**FunClust**
7	**70**	0.40	0.18	0.24	0.42
	140	0.92	0.17	0.13	0.43
	280	1	0.10	0.08	0.51
	1400	1	0.11	0.01	0.57

Table 4.8: Success in Clustering for the Proposed Procedure, Functional Principal Components, Funclust and Functional K-means Using 7 B-Spline Bases in Simulation 2 (The Variability Information has not been Removed From the Data)

In this case, the dependence of the results on the choice of basis is very limited, as the values we obtain are nearly identical for both basis choices.

In summary, from the results in the preceding tables it follows that, under the models considered in the experiments, using the kurtosis algorithm provides an efficient way to reduce the dimension without affecting the heterogeneity in the data, and to perform clustering analysis. It would also seem to provide a powerful tool for the exploratory analysis of these data.

These results illustrate a marked improvement on the corresponding success rates obtained using Functional Principal Components, Functional K-means or the Funclust method. Thus, we believe that in many cases our proposed method may provide clear advantages for the study of heterogeneous data, and the application of clustering techniques to these data.

4.5 Conclusion

In this Chapter we have introduced a kurtosis operator for functional data, inspired by the multivariate kurtosis matrix proposed by Móri et al. (1993), and adapted to ensure good clustering properties in a functional setting.

The modifications in the proposed procedure are motivated by the form in the functional setting of an optimal discriminant function with bounded norm, which has been derived in the thesis. This function has been used to define a bounded Mahalanobis distance, and from it the proposed kurtosis operator.

Regarding its theoretical properties, and for the case of gaussian processes with the same covariance operator, we have shown that there exists an eigenfunction of the operator that is asymptotically optimal, in the sense that it is arbitrarily close to the optimal separation function we have derived for this case. These results approximate the corresponding properties of the multivariate proposal studied in Peña et al. (2010).

We have also shown, through some examples based on real and simulated data, that the proposed method is more efficient than other clustering methods described in the literature for functional data, such as Functional Principal Components, Functional K-means and FunClust. In summary, the proposed method is an interesting contribution to identify structures removed from normality and in particular to identify clusters in functional data sets.

Conclusions and Further Research

Conclusions

In this thesis we have considered two main lines of research, both of them related to extending and improving existing algorithms for clustering applications based on kurtosis directions. These methods are motivated by the method proposed by Peña and Prieto (2001a), which has been shown to work well when the dimension of the data is low and when the number of clusters present in the sample is small.

Our first interest has been to propose a cluster identification method suitable for high-dimensional data when a large number of clusters is present in the sample. In Chapter 3 we have proposed a procedure based on the iterative binary separation of the existing clusters that works very efficiently in practice.

The proposed algorithm detects the presence of clusters by applying a two-step procedure: i) it projects the data onto directions that minimize or maximize the kurtosis coefficient; and ii) for each one of the projections we analyze if we have a mixture of two distributions using the BIC model selection criterion. The procedure is applied recursively in the case in which the BIC value for the mixture of two distributions is greater than the BIC value for one distribution, otherwise the procedure is finalized.

The main improvements associated to this method are the search for binary separations in the projections, and the use of a model-based approach to the identification of the univariate clusters. The proposed method presents some advantages over the method in Peña and Prieto (2001a): (1) compared to the use of first-order gaps to identify clusters in the univariate data, it is less susceptible to the presence of outliers and therefore provides better results when the clusters are close; (2) its theoretical properties indicate that the projection directions tend to identify pairs of groups, implying that a binary division procedure should work better than trying to identify many clusters at the same time; (3) this procedure tends to avoid an excessive subdivision of the observations.

We have conducted several simulation studies. We have started by considering the case of samples formed by a mixture of three normal populations. Comparing the results obtained by applying our proposed clustering algorithm with the methods MCLUST, CLARA, K-means and the one proposed by Peña and Prieto (2001a), we conclude that our method is more efficient for the identification of the three groups.

In a second simulation study we have considered random observations generated from a mixture of Normal, Uniform, and Student-t multivariate distributions. From the results we again conclude that our proposed method is more efficient to identify clusters present in the sample than other methods commonly used in the literature. Additionally, we have also shown the efficiency of our method when we have a mixture of several normal distributions with outliers.

We have also conducted a theoretical study of the properties of the extreme kurtosis directions regarding the identification of the different clusters in the sample. In the presence of many groups, some interesting projection directions are those that project the observations into two blocks of groups. We have characterized these projection directions, and we have proved that they can

be approximated by the extreme kurtosis directions, through an asymptotic relationship that ensures the equivalence of the directions when there exists an arbitrarily large separation between the groups in the data.

Our second interest in the thesis has been to extend the kurtosis-based clustering procedure to the case in which we wish to analyze functional data. This extension has been presented in Chapter 4.

We have defined a kurtosis operator for functional data based on an extension of a multivariate kurtosis matrix, as an adaptation of the proposal presented in Peña et al. (2010). The definition of this operator has been based on the characterization of optimal classification functions in the case of a mixture of gaussian processes. Based on the form of an optimal classification function, we have defined a Mahalanobis distance with finite values, based on a regularized inverse covariance operator. This definition has been used as a reference to introduce our proposed kurtosis operator. We have shown that this kurtosis operator has the property that the eigenfunctions corresponding to a generalized eigenvalue problem are asymptotically equivalent to the optimal classification function, as the regularization parameter goes to zero, in the case of a mixture of two gaussian processes.

From this definition of a kurtosis operator, and based on an implementation of the calculation of the eigenfunctions for the operator, we complete our clustering algorithm for functional data by analyzing the univariate projections obtained for the relevant eigenfunctions using the same model-based approach presented in Chapter 3.

We have presented several results for the application of the proposed kurtosis operator to the clustering problem in functional data. For the *CanadianWeather*, *Growth* and *ECG* data sets we obtain better results compared to the groupings obtained from several clustering algorithms, such as Functional Principal

Components, Functional K-means and FunClust. We have also conducted simulation experiments and compared the results to these other functional clustering methods. From the results obtained in these simulation experiments we conclude that the proposed algorithm obtains results that outperform those of the alternative algorithms used in the comparison.

Further Research

The theoretical and computational results obtained for the algorithms proposed in this thesis pose several open problems that would be of interest to analyze in further work. Some of those we consider most relevant are:

- The proposed multivariate algorithm tends to present worse results when the ratio n/p is small. It would be interesting to find ways to complement the method with other procedures that may work better in those cases.

- In the case of very large dimensional data (thousands of dimensions), the method becomes computationally very expensive. For those cases, it would be interesting to explore the application of an initial dimension-reduction stage, with reduced computational cost, before continuing with the application of the proposed procedure.

- In the functional case, an open problem is the choice of basis for the representation of the data. One possible approach would be to use a data-defined basis. It would be of interest to study the efficiency of representation of the data in terms of eigenfunctions of the covariance or the kurtosis operators, for example.

- Another relevant aspect in this functional case has to do with the introduction of a method for the computation of projection functions based

on the optimization of a kurtosis coefficient, instead of working with eigenfunctions of a kurtosis operator. This approach provides better results for the multivariate case, for example, but implies significant complications in the extension of the corresponding procedure in the functional case. One possible approach for this case could be based on the use of local linear approximations, for example.

Appendix

In this additional chapter, we show in more detail the results presented in Tables 3.8, 3.9, 3.10 and 3.11. We will present the results obtained for each distribution with the different values of p and $k = 2, 3, 4, 8$.

Tables show the percentage of observations that coincide with the original clusters.

Multivariate Normal Distributions

Multivariate Normal Distributions					
$p = 4$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.98	0.92	0.92	0.99
	3	0.90	0.76	0.74	0.73
	4	0.79	0.52	0.73	0.42
	8	0.31	0.12	0.69	0.02
	Mean	**0.75**	**0.58**	**0.77**	**0.54**
50	**2**	0.99	0.98	0.90	0.97
	3	0.99	0.84	0.69	0.66
	4	0.96	0.78	0.65	0.39
	8	0.93	0.57	0.53	0.07
	Mean	**0.97**	**0.79**	**0.69**	**0.52**
100	**2**	1	0.99	0.89	0.98
	3	1	0.90	0.69	0.71
	4	0.98	0.85	0.63	0.35
	8	0.96	0.75	0.53	0.03
	Mean	**0.99**	**0.87**	**0.68**	**0.52**

Table 4.9: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Normal Observations and $p = 4$

Multivariate Normal Distributions					
$p = 8$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	1	0.93	0.93	1
	3	0.91	0.59	0.89	0.70
	4	0.87	0.33	0.75	0.47
	8	0.35	0.02	0.55	0.04
	Mean	**0.78**	**0.47**	**0.78**	**0.55**
50	**2**	1	0.99	0.93	1
	3	1	0.88	0.90	0.70
	4	1	0.72	0.74	0.43
	8	0.98	0.44	0.43	0
	Mean	**0.99**	**0.76**	**0.75**	**0.53**
100	**2**	1	1	0.93	1
	3	1	0.94	0.90	0.78
	4	1	0.82	0.79	0.36
	8	0.98	0.71	0.26	0
	Mean	**1**	**0.87**	**0.72**	**0.54**

Table 4.10: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Normal Observations and $p = 8$

Multivariate Normal Distributions					
$p = 15$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	1	0.91	1	0.98
	3	0.93	0.58	0.93	0.77
	4	0.89	0.15	0.70	0.52
	8	0.34	0	0.17	0.11
	Mean	**0.79**	**0.41**	**0.70**	**0.60**
50	**2**	1	0.99	0.97	0.97
	3	1	0.86	0.91	0.72
	4	1	0.66	0.86	0.36
	8	0.99	0.11	0.52	0.08
	Mean	**1**	**0.66**	**0.82**	**0.53**
100	**2**	1	1	0.96	0.96
	3	1	0.93	0.89	0.68
	4	1	0.90	0.90	0.49
	8	0.98	0.47	0.73	0.05
	Mean	**1**	**0.83**	**0.87**	**0.55**

Table 4.11: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Normal Observations and $p = 15$

Multivariate Normal Distributions					
$p = 30$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.98	0.97	0.97	0.96
	3	0.92	0.43	0.90	0.73
	4	0.79	0.08	0.69	0.45
	8	0.30	0.02	0.23	0.03
	Mean	**0.75**	**0.37**	**0.70**	**0.54**
50	**2**	1	0.96	0.95	0.94
	3	0.95	0.84	0.91	0.77
	4	0.98	0.67	0.93	0.41
	8	0.98	0.07	0.75	0.03
	Mean	**0.98**	**0.64**	**0.89**	**0.54**
100	**2**	1	1	0.94	0.94
	3	1	1	0.90	0.63
	4	0.99	0.75	0.93	0.42
	8	0.99	0.53	0.61	0.01
	Mean	**1**	**0.82**	**0.84**	**0.50**

Table 4.12: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Normal Observations and $p = 30$

Multivariate Normal Distributions					
$p = 50$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.98	0.87	0.97	1
	3	0.86	0.09	0.87	0.71
	4	0.55	0.11	0.55	0.56
	8	0.23	0	0.12	0.07
	Mean	**0.66**	**0.27**	**0.63**	**0.59**
50	**2**	1	0.98	0.95	1
	3	0.93	0.84	0.93	0.70
	4	0.95	0.55	0.77	0.52
	8	0.96	0.04	0.35	0
	Mean	**0.96**	**0.60**	**0.75**	**0.56**
100	**2**	1	0.99	0.88	1
	3	0.98	0.92	0.94	0.71
	4	0.98	0.85	0.87	0.45
	8	0.96	0.32	0.53	0
	Mean	**0.98**	**0.77**	**0.80**	**0.54**

Table 4.13: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Normal Observations and $p = 50$

Multivariate Uniform Distributions

Multivariate Uniform Distributions					
$p = 4$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.99	0.98	0.98	0.99
	3	0.95	0.93	0.88	0.64
	4	0.95	0.92	0.84	0.47
	8	0.65	0.71	0.80	0.03
	Mean	**0.89**	**0.89**	**0.88**	**0.53**
50	**2**	1	0.99	0.97	0.98
	3	0.98	1	0.81	0.75
	4	0.99	0.98	0.79	0.44
	8	0.99	0.96	0.67	0.05
	Mean	**0.99**	**0.98**	**0.81**	**0.56**
100	**2**	1	0.99	0.94	0.99
	3	1	0.98	0.85	0.71
	4	1	0.96	0.79	0.37
	8	0.99	0.94	0.61	0.04
	Mean	**1**	**0.97**	**0.80**	**0.53**

Table 4.14: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Uniform Observations and $p = 4$

Multivariate Uniform Distributions					
$p = 8$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	1	1	1	1
	3	1	0.96	0.98	0.63
	4	0.94	0.92	0.93	0.43
	8	0.72	0.70	0.70	0.09
	Mean	**0.92**	**0.90**	**0.90**	**0.54**
50	**2**	1	1	1	0.99
	3	1	0.99	0.99	0.58
	4	0.99	0.99	0.93	0.36
	8	0.99	0.89	0.84	0.07
	Mean	**1**	**0.97**	**0.94**	**0.50**
100	**2**	1	1	1	0.98
	3	1	1	0.91	0.64
	4	1	0.99	0.91	0.45
	8	0.99	0.93	0.93	0.05
	Mean	**1**	**0.98**	**0.94**	**0.53**

Table 4.15: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Uniform Observations and $p = 8$

Multivariate Uniform Distributions					
$p = 15$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	1	1	1	1
	3	1	0.97	0.99	0.72
	4	0.90	0.84	0.90	0.52
	8	0.70	0.11	0.73	0.02
	Mean	**0.90**	**0.73**	**0.91**	**0.57**
50	**2**	1	1	1	1
	3	1	1	1	0.70
	4	1	0.94	0.86	0.40
	8	1	0.60	0.69	0.06
	Mean	**1**	**0.89**	**0.89**	**0.54**
100	**2**	1	1	0.99	1
	3	1	1	0.99	0.62
	4	1	0.98	0.88	0.30
	8	1	0.83	0.70	0.04
	Mean	**1**	**0.95**	**0.89**	**0.49**

Table 4.16: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Uniform Observations and $p = 15$

Multivariate Uniform Distributions					
$p = 30$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	1	1	1	1
	3	0.98	0.79	1	0.67
	4	0.83	0.32	0.92	0.52
	8	0.43	0.10	0.32	0.05
	Mean	**0.81**	**0.55**	**0.81**	**0.56**
50	**2**	0.99	1	1	0.99
	3	0.99	0.99	0.99	0.71
	4	0.98	0.87	0.87	0.42
	8	0.98	0.22	0.71	0.11
	Mean	**0.99**	**0.77**	**0.89**	**0.56**
100	**2**	1	1	1	1
	3	1	0.91	0.97	0.70
	4	1	0.79	0.87	0.33
	8	1	0.43	0.75	0
	Mean	**1**	**0.78**	**0.90**	**0.51**

Table 4.17: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Uniform Observations and $p = 30$

Multivariate Uniform Distributions					
$p = 50$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.98	0.99	1	1
	3	0.92	0.34	0.95	0.71
	4	0.74	0.23	0.89	0.51
	8	0.39	0.05	0.14	0.10
	Mean	**0.76**	**0.40**	**0.75**	**0.58**
50	**2**	0.97	0.98	0.98	1
	3	0.98	0.94	0.93	0.72
	4	0.96	0.55	0.80	0.45
	8	1	0.13	0.41	0.01
	Mean	**0.98**	**0.65**	**0.78**	**0.55**
100	**2**	1	0.99	1	0.98
	3	0.99	0.95	0.96	0.68
	4	0.99	0.56	0.88	0.28
	8	0.98	0.40	0.82	0.02
	Mean	**0.99**	**0.73**	**0.92**	**0.49**

Table 4.18: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Uniform Observations and $p = 50$

Multivariate Student-t Distributions

Multivariate Student-t Distributions					
$p = 4$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.92	0.86	0.67	0.83
	3	0.81	0.66	0.50	0.60
	4	0.68	0.42	0.46	0.37
	8	0.21	0.04	0.29	0.05
	Mean	**0.66**	**0.50**	**0.48**	**0.46**
50	**2**	0.98	0.96	0.78	0.87
	3	0.94	0.80	0.44	0.52
	4	0.86	0.75	0.45	0.30
	8	0.75	0.53	0.46	0.04
	Mean	**0.88**	**0.76**	**0.53**	**0.43**
100	**2**	0.98	0.97	0.79	0.82
	3	0.94	0.93	0.44	0.48
	4	0.93	0.85	0.43	0.22
	8	0.86	0.33	0.44	0
	Mean	**0.93**	**0.77**	**0.53**	**0.38**

Table 4.19: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Student-t Observations and $p = 4$

Multivariate Student-t Distributions					
$p = 8$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.97	0.93	0.90	0.97
	3	0.96	0.52	0.73	0.80
	4	0.74	0.24	0.66	0.41
	8	0.25	0.01	0.21	0.07
	Mean	**0.73**	**0.43**	**0.63**	**0.56**
50	**2**	1	0.98	0.89	0.96
	3	1	0.82	0.72	0.68
	4	0.96	0.75	0.64	0.34
	8	0.86	0.28	0.39	0.04
	Mean	**0.96**	**0.71**	**0.66**	**0.51**
100	**2**	1	1	0.83	0.91
	3	1	0.93	0.71	0.76
	4	1	0.78	0.61	0.37
	8	0.99	0.44	0.31	0.04
	Mean	**1**	**0.79**	**0.62**	**0.52**

Table 4.20: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Student-t Observations and $p = 8$

Multivariate Student-t Distributions					
$p = 15$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.99	0.93	0.94	0.99
	3	0.98	0.51	0.92	0.71
	4	0.73	0.11	0.76	0.48
	8	0.21	0	0.35	0.11
	Mean	**0.73**	**0.39**	**0.74**	**0.57**
50	**2**	1	1	0.93	1
	3	1	0.89	0.93	0.70
	4	1	0.75	0.83	0.42
	8	1	0.04	0.41	0.07
	Mean	**1**	**0.67**	**0.78**	**0.55**
100	**2**	1	1	0.95	1
	3	0.99	0.86	0.91	0.61
	4	0.99	0.80	0.84	0.34
	8	1	0.29	0.50	0.05
	Mean	**1**	**0.74**	**0.80**	**0.50**

Table 4.21: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Student-t Observations and $p = 15$

Multivariate Student-t Distributions					
$p = 30$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.98	0.91	0.97	1
	3	0.90	0.35	0.98	0.68
	4	0.76	0.08	0.79	0.44
	8	0.32	0.02	0.32	0.06
	Mean	**0.74**	**0.34**	**0.77**	**0.55**
50	**2**	0.99	1	0.91	1
	3	0.95	0.94	0.92	0.66
	4	0.95	0.64	0.78	0.37
	8	0.93	0	0.35	0.09
	Mean	**0.96**	**0.65**	**0.74**	**0.53**
100	**2**	1	0.97	0.90	1
	3	1	0.92	0.92	0.70
	4	0.99	0.73	0.89	0.32
	8	0.98	0.34	0.58	0.02
	Mean	**0.99**	**0.74**	**0.82**	**0.51**

Table 4.22: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Student-t Observations and $p = 30$

Multivariate Student-t Distributions					
$p = 50$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.99	0.85	0.95	1
	3	0.87	0.13	0.90	0.75
	4	0.75	0.10	0.57	0.52
	8	0.25	0.05	0.18	0.07
	Mean	**0.72**	**0.28**	**0.65**	**0.59**
50	**2**	0.98	0.98	0.93	0.98
	3	0.93	0.84	0.92	0.74
	4	0.92	0.61	0.76	0.46
	8	0.98	0.12	0.55	0
	Mean	**0.95**	**0.64**	**0.79**	**0.55**
100	**2**	1	1	1	0.95
	3	1	0.93	0.96	0.68
	4	0.98	0.83	0.90	0.41
	8	0.96	0.25	0.51	0
	Mean	**0.99**	**0.75**	**0.84**	**0.51**

Table 4.23: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Student-t Observations and $p = 50$

Normal Observations with Outliers

Normal Observations with Outliers					
$p = 4$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.94	0.52	0.86	0.91
	3	0.91	0.49	0.58	0.76
	4	0.72	0.29	0.61	0.25
	8	0.40	0.06	0.73	0.08
	Mean	**0.74**	**0.34**	**0.70**	**0.50**
50	**2**	0.99	0.21	0.83	0.94
	3	0.91	0.19	0.56	0.75
	4	0.90	0.26	0.53	0.39
	8	0.88	0.42	0.61	0.02
	Mean	**0.92**	**0.27**	**0.63**	**0.53**
100	**2**	1	0.25	0.81	0.89
	3	0.93	0.09	0.43	0.67
	4	0.95	0.05	0.36	0.36
	8	0.90	0.29	0.50	0.03
	Mean	**0.95**	**0.17**	**0.53**	**0.49**

Table 4.24: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Normal Observations with Outliers and $p = 4$

Normal Observations with Outliers					
$p = 8$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.98	0.67	0.90	0.90
	3	0.93	0.34	0.63	0.71
	4	0.76	0.17	0.69	0.32
	8	0.43	0.01	0.86	0.08
	Mean	**0.78**	**0.30**	**0.77**	**0.50**
50	**2**	0.99	0.11	0.85	0.94
	3	0.98	0.19	0.60	0.74
	4	0.98	0.27	0.64	0.43
	8	0.96	0.28	0.82	0.08
	Mean	**0.98**	**0.21**	**0.73**	**0.55**
100	**2**	1	0.40	0.82	0.96
	3	0.98	0.16	0.53	0.69
	4	0.97	0.11	0.60	0.34
	8	0.98	0.35	0.77	0.06
	Mean	**0.98**	**0.26**	**0.68**	**0.51**

Table 4.25: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Normal Observations with Outliers and $p = 8$

Normal Observations with Outliers					
$p = 15$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.96	0.55	0.93	0.87
	3	0.92	0.43	0.65	0.68
	4	0.73	0.33	0.75	0.43
	8	0.40	0	0.85	0.08
	Mean	**0.75**	**0.33**	**0.80**	**0.52**
50	**2**	0.97	0.66	0.90	0.95
	3	0.97	0.38	0.60	0.66
	4	0.96	0.30	0.72	0.38
	8	0.95	0.29	0.80	0.08
	Mean	**0.96**	**0.41**	**0.76**	**0.52**
100	**2**	0.99	0.66	0.87	0.95
	3	0.98	0.34	0.55	0.80
	4	0.98	0.25	0.68	0.39
	8	0.97	0.50	0.73	0.02
	Mean	**0.98**	**0.44**	**0.71**	**0.54**

Table 4.26: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Normal Observations with Outliers and $p = 15$

Normal Observations with Outliers					
$p = 30$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.95	0.44	0.95	0.84
	3	0.90	0.21	0.63	0.70
	4	0.75	0.02	0.77	0.57
	8	0.29	0	0.75	0.11
	Mean	**0.72**	**0.17**	**0.78**	**0.56**
50	**2**	0.97	0.99	0.95	0.96
	3	0.94	0.70	0.69	0.76
	4	0.92	0.45	0.75	0.37
	8	0.90	0.10	0.77	0.09
	Mean	**0.93**	**0.56**	**0.79**	**0.55**
100	**2**	0.99	0.93	0.93	0.86
	3	0.98	0.91	0.60	0.61
	4	0.98	0.90	0.69	0.37
	8	0.93	0.63	0.59	0
	Mean	**0.97**	**0.84**	**0.70**	**0.46**

Table 4.27: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Normal Observations with Outliers and $p = 30$

Normal Observations with Outliers					
$p = 50$		*Average Success Rate*			
n/p	k	**Kurtosis**	**MCLUST**	**CLARA**	**Kmeans**
20	**2**	0.93	0.40	0.90	0.82
	3	0.87	0.08	0.61	0.63
	4	0.75	0.02	0.75	0.53
	8	0.27	0	0.60	0.10
	Mean	**0.71**	**0.13**	**0.72**	**0.52**
50	**2**	0.96	0.97	0.89	1
	3	0.92	0.90	0.63	0.84
	4	0.82	0.55	0.73	0.55
	8	0.80	0.08	0.54	0.08
	Mean	**0.88**	**0.63**	**0.70**	**0.62**
100	**2**	1	0.95	0.86	0.97
	3	0.99	0.89	0.60	0.50
	4	0.98	0.90	0.69	0.46
	8	0.95	0.65	0.50	0.02
	Mean	**0.98**	**0.85**	**0.66**	**0.49**

Table 4.28: Average Success in Clustering for the Proposed Method and the MCLUST, CLARA and Kmeans Algorithms with Normal Observations with Outliers and $p = 50$

Bibliography

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

[2] Backer, E. (1978). Cluster Analysis by Optimal Decomposition of Induced Fuzzy Sets. *Delft University Press*.

[3] Ball, G. and Hall, D. (1967). A clustering technique for summarizing multivariate data. *Behavioral Science*, 12 (2), 153-155.

[4] Ball, G. and Hall, D. (1965). ISODATA, a novel method of data analysis and pattern classification. *Stanford Research Institute California*.

[5] Banfield, J.D. and Raftery, A. E. (1993). Model-based Gaussian and Non-Gaussian Clustering. *Biometrics*, 49, 803-821.

[6] Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data*, 25-71. Springer Berlin Heidelberg.

[7] Beyer, K., Goldstein, J., Ramakrishnan, R. and Shaft, U. (1999). When is "nearest neighbor" meaningful?. In *International conference on database theory*. Springer Berlin Heidelberg, 217-235.

[8] Bezdek, J. C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. *Plenum Press*.

[9] Biau, G., Bunea, F. and Wegkamp, M. (2005). Functional classification in Hilbert spaces. *IEEE Transactions on Information Theory*, 51, 2162-2172.

[10] Bouveyron, C., and Brunet-Saumard, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics Data Analysis*, 71, 52-78.

[11] Cardoso, J. F. (1989). Source separation using higher order moments. *Acoustics, Speech, and Signal Processing. ICASSP-89*, 2109-2112.

[12] Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28, 781-793.

[13] Cheeseman, P. and Stutz, J. (1996). Bayesian Classification (AutoClass): Theory and Results. In *Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy , R. (Eds.) Advances in Knowledge Discovery and Data Mining, AAAI Press/MIT Press*.

[14] Chiu, T., Fang, D., Chen, J., and Wang, Y. (2001). A Robust and scalable clustering algorithm for mixed type attributes in large database environments. In *Proceedings of the 7th ACM SIGKDD*, 263-268, San Francisco, CA.

[15] Cuesta-Albertos, J. A., Gordaliza, A. and Matrán, C. (1997). Trimmed K-means: An attempt to robustify quantizers. *The Annals of Statistics*, 25(2), 553-576.

[16] Darlington, R. B. (1970). Is kurtosis really "Peakedness"?. *The American Statistician*, 24 (2), 19-22.

[17] Dean, N., Raftery, A. and Scrucca, L. (2013). Package clustvarsel: variable selection for model-based clustering.

[18] Delaigle, A. and Hall, P. (2010). Defining probability density for a distribution of random functions. *The Annals of Statistics*, 1171-1193.

[19] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B, 39 (1), 1-38.

[20] Dubes, R. C. (1987). How many clusters are best? an experiment. *Pattern Recognition*, 20 (6), 645-663.

[21] Dunn, J.C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybernet*, 3, 32-57.

[22] Ferraty, F., and Vieu, P. (2003). Curves Discrimination: A Nonparametric Functional Approach. Computational *Statistics and Data Analysis*, 44, 161-173.

[23] Ferraty, F., and Vieu, P. (2006). Nonparametric Functional Data Analysis: Theory and Practice. *Springer*.

[24] Fisher, R. A. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*. 7 (2), 179-188.

[25] Forgy, E.W. (1965). Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics* 21, 768-769.

[26] Fraiman, R., Justel, A., and Svarc, M. (2008). Selection of variables for cluster analysis and classification rules. *Journal of the American Statistical Association*,103(483), 1294 - 1303.

[27] Fraley, C. (1999). Algorithms for model-based Gaussian hierarchical clustering. *SIAM J. Sci. Comput.*, 20, 270-281.

[28] Fraley, C. and Raftery, A. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41 (8), 578-588.

[29] Fraley, C. and Raftery, A. (1999). MCLUST: Software for model-based cluster and discriminant analysis. *Technical Report* 342, Dept. Statistics, Univ. of Washington.

[30] Fraley, C. and Raftery, A. (2012). MCLUST Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation. *Technical Report* 597, University of Washington, Department of Statistics.

[31] Friedman, J. H. (1987). Exploratory projection pursuit. *Journal of the American statistical association*, 82(397), 249-266.

[32] Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. *Journal of the American statistical Association*, 76(376), 817-823.

[33] Friedman, J. H. and Tukey, J. W. (1974).A Projection Pursuit Algorithm for Exploratory Data Analysis. *IEEE Transactions on Computers*, 23(9), 881-890.

[34] Galimberti, G., Manisi, A., and Soffritti, G. (2017). Modelling the role of variables in model-based cluster analysis. *Statistics and Computing*, 1-25.

[35] Guha, S., Rastogi R. and Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. In *Proceedings of the ACM SIGMOD Conference*, 73-84, Seattle, WA.

[36] Guha, S., Rastogi R. and Shim, K. (1999). ROCK: A robust clustering algorithm for categorical attributes. In *Proceedings of the 15th ICDE*, 512-521, Sydney, Australia.

[37] Han, J., Kamber, M. and Tung, A.K.H. (2001). Spatial clustering methods in data mining: A survey. *Geographic data mining and knowledge discovery, Taylor and Francis.*

[38] Hart, P. (1968). The condensed nearest neighbor rule. *IEEE Trans. on Inform. Th.*, 14, 515-516.

[39] Hartigan, J. A. and Wong, M. A. (1979). A K-means clustering algorithm. *Journal of the Royal Statistical Society*, Series C (Applied Statistics), 28, 100-108.

[40] Hastie, T., Buja, A. and Tibshirani, R. (1995). Penalized discriminant analysis. *The Annals of Statistics*, 23, 73-102.

[41] Hildebrand, D. K. (1971). Kurtosis measures bimodality?. *The American Statistician*, 25(1), 42-43.

[42] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24, p. 417-441, and 498-520.

[43] Huber, P. J. (1985). Projection Pursuit. *The Annals of Statistics*, 13, 435-525.

[44] Hyvärinen, A. (1999). Survey on independent component analysis. *Neural Comput. Surv.*, 2, p. 94-128.

[45] Hyvärinen, A. and Oja, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9, 1483-1492.

[46] Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks*, 13(4), 411-430.

[47] Hyvärinen, A., Karhunen, J. and Oja E. M. (2001). Independent Component Analysis. *New York: John Wiley.*

[48] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31 (8), 651-666.

[49] Jain, A. K., Dubes, R. C. (1988). Algorithms for Clustering Data. *Prentice Hall.*

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

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

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

[53] Jobson, J. (2012). Applied multivariate data analysis: volume II: Categorical and Multivariate Methods. *Springer Science and Business Media.*

[54] Jones, M. C. and Sibson, R. (1987). What is projection pursuit?. *Journal of the Royal Statistical Society.* Series A (General), 1-37.

[55] Karhunen, J., Oja, E., Wang, L., Vigario, R. and Joutsensalo, J. (1997). A class of neural networks for independent component analysis. IEEE Transactions on Neural Networks, 8(3), 486-504.

[56] Karypis, G., Han, E.-H., and Kumar, V. (1999). CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. *COMPUTER*, 32, 68-75.

[57] Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90, 773-795.

[58] Kaufman, L. and Rousseeuw, P.J. (1990). Finding groups in data: An introduction to cluster analysis. *Wiley, New York*.

[59] King, B. (1967). Step-wise clustering procedures. *J. Am. Stat. Assoc.* 69, 86-101.

[60] Kogan, J., Nicholas, C. and Teboulle, M. (2006). Grouping Multidimensional Data: Recent Advances in Clustering. *Springer-Verlag Berlin Heidelberg*.

[61] Kollo, T. (2008). Multivariate skewness and kurtosis measures with an application in ICA. *Journal of Multivariate Analysis*, 99(10), 2328-2338.

[62] Koziol, J. A. (1989). A note on measures of multivariate kurtosis. *Biometrical journal*, 31(5), 619-624.

[63] Lee, T. W. (1998). Independent component analysis. In *Independent Component Analysis*, 27-66. Springer US.

[64] Lee, E. K., Cook, D., Klinke, S. and Lumley, T. (2012). Projection pursuit for exploratory supervised classification. *Journal of Computational and Graphical Statistics*.

[65] Li, Y., and Wu, H. (2012). A clustering method based on K-means algorithm. *Physics Procedia*, 25, 1104-1109.

[66] Liu, J., Zhang, J., Palumbo, M. and Lawrence C. (2003): Bayesian clustering with variable and transformation selections. *Bayesian Statistics*, 7, eds, J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, Oxford: University Press, 249 - 75.

[67] López-Pintado, S., and Romo, J. (2006). Depth-based classification for functional data. Data Depth: Robust Multivariate Analysis, Computational Geometry and Applications. *American Mathematical Society*. DIMACS Series, 72, 103-121.

[68] Lucasius, C.B., Dane, A.D. and Kateman, G. (1993). On k-medoid clustering of large data sets with the aid of a genetic algorithm: Background, feasibility and comparison. *Analytica Chimica Acta*, 282, 647-669.

[69] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press*, 281-297.

[70] Mao, J., Jain, A. K. (1996). A self-organizing network for hyper-ellipsoidal clustering (HEC). *IEEE Trans. Neural Networks* 7 (January), 16-29.

[71] Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika* 57, 519-530.

[72] Maugis, C., Celeux, G. and Martin-Magniette, M.L. (2009). Variable selection in model-based clustering: a general variable role modeling. *Computational Statistics and Data Analysis*, 53(11), 3872-3882.

[73] Maugis, C., Marie, M. M., and Sandra, P. (2012). SelvarClustMV: Variable selection approach in model-based clustering allowing for missing values. *Journal de la SFdS*, 15(2), 21-36.

[74] McCallum, A., Nigam, K. and Ungar, L. H. (2000). Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 169-178.

[75] McLachlan, G. and Basford (1988). Mixture Models: Inference and Applications to Clustering. *Marcel Dekker, New York, NY. K.*

[76] McLachlan, G. and Krishnan, T. (1997). The EM Algorithm and Extensions. *John Wiley and Sons, New York, NY.*

[77] Michalski, R.S. and Stepp, R. (1983). Learning from observations: conceptual clustering. In *Machine Learning: An Artificial Intelligence Approach.* San Mateo, CA, Morgan Kaufmann.

[78] Moors, J. J. A. (1986). The meaning of kurtosis: Darlington reexamined. *The American Statistician*, 40(4), 283-284.

[79] Móri, T. F., Rohatgi, V. K. and Székely, G. J. (1993). On multivariate skewness and kurtosis. *Theory of Probability and its Applications*, 38, 547-551.

[80] Ng, R. and Han, J. (1994). Efficient and effective clustering methods for spatial data mining. In *Proceedings of the 20th international conference on very large databases*, 144-155. Santiago, Chile.

[81] Oja, E. (1997). The nonlinear PCA learning rule in independent component analysis. *Neurocomputing*, 17(1), 25-45.

[82] Pan, W., andShen, X. (2007). Penalized model-based clustering with application to variable selection.*Journal of Machine Learning Research*, 8(May), 1145-1164.

[83] Park, H.S. and Jun, C.H. (2009): A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications*, 36 (2), 333-3341.

[84] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2 (11), 559-572.

[85] Pearson, K. (1905). Das Fehlergesetz und Seine Verallgemeinerungen Durch Fechner und Pearson. A Rejoinder. *Biometrika*, 4, 169-212.

[86] Peña, D. and F. J. Prieto (2001a). Cluster identication using projections. J*ournal of the American Statistical Association*, 96, 1433 - 1445.

[87] Peña, D. and F. J. Prieto (2001b). Robust covariance matrix estimation and Multivariate outlier detection. *Technometrics*, 43, 286-310.

[88] Peña, D., Prieto, F. J. and Viladomat, J. (2010). Eigenvectors of a kurtosis matrix as interesting directions to reveal cluster structure. *Journal of Multivariate Analysis*, 101, 1995-2007.

[89] Pollard, D. (1981). Strong consistency of K-means clustering. *The Annals of Statistics*, 9(1), 135-140.

[90] Pollard, D. (1982). A central limit theorem for K-means clustering. *The Annals of Probability*, 10(4), 919-926.

[91] Rafsanjani, M. K., Varzaneh, Z. A. and Chukanlo, N. E. (2012). A survey of hierarchical clustering algorithms. *The Journal of Mathematics and Computer Science*, 5(3), 229-240.

[92] Raftery, A. E., and Dean, N. (2006). Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473), 168-178.

[93] Ramsay, J. O., Hooker, G., and Graves, S. (2009). Functional Data Analysis with R and MATLAB (Use R). *Springer*.

[94] Ramsay, J. O, and Silverman, B.W. (2002). Applied Functional Data Analysis: Methods and case studies. *Springer*.

[95] Ramsay, J. O, and Silverman, B.W. (1997). Functional data analysis. *Springer*.

[96] Ramsay, J. O, and Silverman, B.W. (2005). Functional Data Analysis, Second Edition. Springer.

[97] Rousseeuw, P.J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20 (1), p. 53-65.

[98] Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6 (2), 461-464.

[99] Schwager, S. J. and Margolin, B. H. (1982). Detection of multivariate normal outliers. *The Annals of Statistics*, 10(3), 943-954.

[100] Serban, N. and Wasserman, L. (2004). CATS: clustering after transformation and smoothing. *Journal of the American Statistical Association*, 100, 990-999.

[101] Sneath, P. H. A. and Sokal, R. R. (1973). Numerical Taxonomy. *Freeman*, London, UK.

[102] Song, Y., Jin, S. and Shen, J. (2011). A unique property of single-link distance and its application in data clustering. *Data and Knowledge Engineering*, 70, 984-1003.

[103] Steinbach, M., Karypis, G. and Kumar, V. (2000). A comparison of document clustering techniques. In: *KDD Workshop on Text Mining*.

[104] Steinley, D., and Brusco, M. J. (2008). A new variable weighting and selection procedure for K-means cluster analysis. *Multivariate Behavioral Research*, 43(1), 77-108.

[105] Tyler, D. E., F. Critchley, L. Dumbgen, and H. Oja (2009).Invariate co-ordinate selection (with discussion). Journal of the Royal Statistical Society: Serie B (Statistical Methodology), 71 (3), p. 1-27.

[106] Wallace, C. and Dowe, D. (1994).Intrinsic classification by MML ? the Snob program. In *the Proceedings of the 7th Australian Joint Conference on Artificial Intelligence*, 37- 44, UNE, World Scientific Publishing Co., Armidale, Australia.

[107] Wang, S., and Zhu, J. (2008), "Variable Selection for Model-Based High-Dimensional Clustering and Its Application to Microarray Data. *Biometrics*, 64, 440-448.

[108] Ward Jr., J. H. (1963).Hierarchical grouping to optimize an objective function, *Journal of the American Statistical Association*, 58, 236-244.

[109] Wei, C.-P. Lee, Y.-H. and Hsu C.-M. (2003).Empirical comparison of fast partitioning-based clustering algorithms for large data sets. *Expert Systems with Applications*, 24 (4), 351-363

[110] Witten, D. M., and Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490), 713-726.

[111] Zhang, B. (2000).Generalized K-Harmonic Means-Dynamic. Weighting of Data in Unsupervised Learning. In *SDM*, 1-13.