https://doi.org/10.26219/heal.aueb.5860

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

SCHOOL OF INFORMATION SCIENCES & TECHNOLOGY

DEPARTMENT OF STATISTICS

POSTGRADUATE PROGRAM

Variable Selection in Model-based clustering

By

Katerina M. Paragioudaki

A THESIS

Submitted to the Department of Statistics
of the Athens University of Economics and Business
in partial fulfilment of the requirements for
the degree of Master of Science in Statistics

Athens, Greece
September 2018

https://doi.org/10.26219/heal.aueb.5860

https://doi.org/10.26219/heal.aueb.5860

ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ & ΤΕΧΝΟΛΟΓΙΑΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ

Επιλογή μεταβλητών στην ομαδοποίηση με χρήση μοντέλων

Κατερίνα Μ. Παραγιουδάκη

ΔΙΑΤΡΙΒΗ

Που υποβλήθηκε στο Τμήμα Στατιστικής

του Οικονομικού Πανεπιστημίου Αθηνών

ως μέρος των απαιτήσεων για την απόκτηση

Μεταπτυχιακού Διπλώματος Ειδίκευσης στη Στατιστική

Αθήνα

Σεπτέμβριος 2018

https://doi.org/10.26219/heal.aueb.5860

https://doi.org/10.26219/heal.aueb.5860

DEDICATION

To my parents and to my teachers...

https://doi.org/10.26219/heal.aueb.5860

ACKNOWLEDGEMENTS

Firstly, I would like to thank my thesis advisor Prof. Papageorgiou Ioulia of Department of Statistics at Athens University of Economics and business because when I had question, always helped me on time.

Last but not the least, I would like to thank my family: my parents and my brother for supporting me spiritually throughout writing this thesis and my life in general.

Paragioudaki Katerina

https://doi.org/10.26219/heal.aueb.5860

https://doi.org/10.26219/heal.aueb.5860

VITA

I was born in Athens in 1992. From young age, I realize my interesting in the Mathematics. So, after completing my studies at Maroussi High School, I went on to the National and Kapodistrian University of Athens, where I studied mathematics and received my bachelor's degree in June 2016.During my studies in mathematics I realized how interesting is the science of statistics. This has resulted, in 2016, after my bachelor studies to continue my studies in department of Statistics at Athens University of Economics and Business.

https://doi.org/10.26219/heal.aueb.5860

https://doi.org/10.26219/heal.aueb.5860

ABSTRACT

Katerina Paragioudaki

Variable Selection in Model-Based Clustering

September 2018

As the technology is developed, the data that we are able to handle are more complicated. In many scientific fields, the datasets have much more variables than observations (high dimensional datasets). In model-based clustering, the problem with the high dimensional data sets is that the number of parameters that need to be estimated is much more than the observations. This number is further increased when the number of clusters increases. For reducing the parameters that need to be estimated, the assumptions for covariance components are more general.

For overcoming the high dimensional problem, many approaches have been proposed. Some approaches that have been proposed are the using of principal components analysis to reduce the dimensions and the variable selection for keeping in the clustering only the variables that contribute to the clustering. In this thesis we will focus in the approach of variable selection developing four methods that select automatic the variables that contribute to clustering in model-based clustering especially for “high dimension, low sample size” data, where the data dimension greatly exceeds the number of observations.

https://doi.org/10.26219/heal.aueb.5860

https://doi.org/10.26219/heal.aueb.5860

ΠΕΡΙΛΗΨΗ

Κατερίνα Παραγιουδάκη

Επιλογή μεταβλητών στην ομαδοποίηση με τη χρήση μοντέλων

Σεπτέμβριος 2018

Καθώς η τεχνολογία αναπτύσσεται, τα δεδομένα τα οποία μπορούμε να χειριστούμε γίνονται πιο πολύπλοκα. Σε πολλά επιστημονικά πεδία, υπάρχουν σετ δεδομένων που έχουν πολύ περισσότερες μεταβλητές από ότι δεδομένα (μεγάλων διαστάσεων σετ δεδομένα). Στην ομαδοποίηση με τη χρήση μοντέλων, το πρόβλημα στα μεγάλων διαστάσεων σετ δεδομένα είναι ότι ο αριθμός των παραμέτρων που χρειάζεται να εκτιμηθεί είναι πολύ μεγαλύτερος από τις παρατηρήσεις. Αυτός ο αριθμός αυξάνεται περεταίρω όταν ο αριθμός των ομάδων μεγαλώνει. Για να μειώσουμε τις παραμέτρους που χρειάζεται να εκτιμηθούν, οι υποθέσεις για τα στοιχεία του πίνακα συνδιακύμανσης είναι πιο γενικές.

Για να λυθεί το πρόβλημα στα μεγάλων διαστάσεων σετ δεδομένα, έχουν προταθεί πολλές επιστημονικές προσεγγίσεις. Κάποιες από αυτές είναι η χρήση της ανάλυσης σε κύριες συνιστώσες ώστε να μειωθούν οι διαστάσεις και η επιλογή των μεταβλητών που συνεισφέρουν στην ομαδοποίηση. Σε αυτήν την εργασία θα επικεντρωθούμε στη δεύτερη προσέγγιση, δηλαδή στην επιλογή των χρήσιμων μεταβλητών αναπτύσσοντας τέσσερις μεθόδους που πραγματοποιούν αυτόματη επιλογή μεταβλητών στην ομαδοποίηση με τη χρήση μοντέλου ειδικά σε μεγάλων διαστάσεων σετ δεδομένα όπου ο αριθμός των διαστάσεων υπερβαίνει κατά πολύ τον αριθμό των παρατηρήσεων.

.

https://doi.org/10.26219/heal.aueb.5860

https://doi.org/10.26219/heal.aueb.5860

TABLE OF CONTENTS

https://doi.org/10.26219/heal.aueb.5860

 https://doi.org/10.26219/heal.aueb.5860

LIST OF TABLES

Table **Page**

https://doi.org/10.26219/heal.aueb.5860

https://doi.org/10.26219/heal.aueb.5860

LIST OF FIGURES

Figure **Page**

https://doi.org/10.26219/heal.aueb.5860

https://doi.org/10.26219/heal.aueb.5860

CHAPTER 1

INTRODUCTION

As the technology is developed, the data that we are able to handle are more complicated. In many datasets like in gene expressions data sets, the characteristics (variables) that we have are much more than the observations. These data sets are called high dimensional datasets. In clustering the high dimensional datasets create some problems. Due to the existence of problems that are created from the high dimensional datasets, we should handle these datasets with different way than the classical data sets in which the number of observations is larger than the number of variables. Many approaches have been developed for handling them. Such approaches are the using of principal component analysis and the selection of variables that contribute to clustering.

Firstly, in Chapter 2, we introduce the definition of clustering. As we refer, clustering is the grouping of observations based on their characteristics. If two observations have same characteristics, they will be in the same group (cluster). Jian Guo et al (2010) claimed that the goal of the clustering is to organize data into a small number of homogeneous groups, thus aiding interpretation. For this purpose, different methods have been developed. Some of them use the notion of distance for measuring how same are the characteristics between the observations. The most important distance is the Euclidian distance and the most algorithms which are based on the distance, use mainly the Euclidian distance. These methods are the K-Means algorithm and the hierarchical algorithms which are developed in Chapter 2. Because these methods are not based on a model and so do not apply statistical inference, it was developed the method of model-based clustering that is based on a model of mixing distributions.

A problem that is occurred in high dimensional datasets is the 'curse of dimensionality' as Bellman (1957) claimed, which is described in Chapter 2. According to this, if the dimensions are increased, the observations are removed from each other. So, if the number of dimensions converges to the infinity, the furthest distance between the observations will converge in the closest distance

 https://doi.org/10.26219/heal.aueb.5860

between them. It has a negative effect in the algorithms that are based on the distance. In model-based clustering it is creates the problem of overparameterization, that is the number of parameters which have to be estimated are much more than the observations. This number is further increased when the number of clusters increases. As a result, the hypothesis made about the component covariance variances is more general. For example, according to Pan and Shen (2007), a hypothesis is that the covariance matrix is the same for all clusters and diagonal. As we'll see in the next Chapters, even in the case the method is working, it can fail to capture the existing clusters.

Generally, for reduction of the dimensions, many researchers proposed the implementation of PCA method (Principal Components Analysis) before analysis. The principal components resulting from this method is a linear combination of the original variables. As we will see in the Chapter 2, many researchers argued that if the principal components with the largest variances be kept, then the variability of the model will be reduced minimally. Other researcher such as Kung and Sharif (1980), disagreed with this and they turned out that the principal components with lower variances are also useful.

In clustering it is turned out by Yeung and Ruzzo (2001) that the implementation of PCA method before clustering using first few principal components destroy the clustering sculpture of the original data.

Also, because in clustering there is little or no knowledge about the analysis and the structure of the clustering that is to be done, it isn't possible to apply a variable selection before the cluster analysis. So, the variable selection should be into the algorithm. For this reason, in this thesis we focus on approach of selection of variables that contribute to the clustering, introducing the process of variable selection into the model-based clustering.

In Chapter 2 we also refer which variables are called informative and which noninformative for clustering. In a few words, according to Ji Zhu et al (2010), the variables that contribute to clustering are the informative variables. The other variables are the noninformative.

Finally, in Chapter 2, we refer some methods that has been proposed and doing variable selection. Some of the methods in Chapter 2 that are referred, developed by Brusco and Cradit (2001) and Friedman and Meulman (2004) use the process of variable selection in the K-Means algorithm and in the

https://doi.org/10.26219/heal.aueb.5860

hierarchical algorithms respectively. Because these methods don't have statistical criteria for the selection of clusters like in the model-based clustering, it is investigated the variable selection process in the model-based clustering. In model-based clustering the criterion that is used for the number of clusters is the BIC criterion as we will see in the Chapter 2.

In Chapter 3, we develop four methods in which the variable selection process is into the model of model-based clustering. The positive of variable selection in model-based clustering is that the BIC criterion is used for selecting the correct model and the number of the clusters. The methods which are developed in Chapter 3 are the AL1 method, the ALP-GMM method, the AHP-GMM method and the APFP method.

As we refer in 3rd Chapter, according to AL1 method, after centering the data, a variable is noninformative for separating two clusters if the estimated means of clusters are the same with the mean value of the variable which is 0. This method developed by Pan and Shen (2007). If all cluster means are equal with 0, then the variable is removed from the model.

The second method that we refer in the Chapter 3 is the ALP-GMM method. This method developed by Wang and Zhu (2008). According to this, after centering the data, if the maximum cluster mean in absolute value is shrunken to 0 (i.e, is estimated as 0) in a variable, then all the cluster means in this variable will be 0 and so the variable will be removed from the model. Because the ALP treats the cluster means with an 'all-in-all-out' manner, the AHP-GMM method is proposed as an improvement from the same researchers as the ALP-GMM method.

The AHP-GMM method selects to shrink to zero any cluster mean, not only the greatest in absolute value cluster mean. If the selected cluster mean is shrunken to zero, then all the cluster means in this variable are automatically 0. But if the selected cluster mean isn't shrunken to zero, then the other cluster means have the probability to be 0. So, the AHP-GMM method keeps the flexibility of AL1 method.

The last method is the APFP method and according to this, a variable will be noninformative for separating two clusters, if the means of these clusters are shrunken to the same value. This method developed by Jian Guo et al (2010). In APFP method the zero has no special meaning. The variable will be removed

https://doi.org/10.26219/heal.aueb.5860

from the model if the means of all clusters associated with this variable are shrunken to the same value.

In Chapter 4, we compare the methods APFP and AL1 with the classic model-based clustering (GMM) having all the noninformative variables into the model and with GMM without noise which is the GMM having subtract all the noninformative variables from this. The purpose for this is to turn out how important is the variable selection in clustering.

In the last Chapter 5 we write the conclusions that we have ended up according to the previous chapters.

So, the purpose of this thesis is

- To explain why the variable selection process is beneficial to be applied in model-based clustering in comparison with the other cluster algorithms
- To develop four methods which select automatic the variables that contribute to clustering.
- To compare theoretically these methods
- With some applications, compare them to each other.
- To show with simulated and real data sets how important is the variable selection in high dimensional data sets.

https://doi.org/10.26219/heal.aueb.5860

CHAPTER 2

INTRODUCTION TO CLUSTERING

2.1. Introduction

Clustering is a method designed to separate our data into groups (clusters) according to their characteristics. A successful analysis will have to create clusters, so that observations belonging to the same cluster to be as homogeneous as possible, while observations in different clusters to differ as much as possible. Many methods have been developed for clustering. Some of them use two important notions, the notion of similarity and the notion of distance (these are the hierarchical methods and K-Means which we analyze in the next paragraph), which are opposite and allow us to measure how similar are the observations between them. Between two identical observations the distance is small while the similarity is great, but on the contrary, the distance between two observations that differ enough is great and the similarity is small. So, we conclude that the distance between observations in the same cluster is small, whereas the distance between observations belonging to different clusters is great in correct clustering. Another method is the model-based clustering which is based on a model and uses statistical criteria for grouping the observations in clusters. Clustering has many applications in various scientific fields. Examples of where it applies is among other things in biology where biologists want to classify various animals in clusters according to their characteristics and in marketing where they want to group customers according to their characteristics. Such characteristics are, for example, if the customers are spenders or more discreet, what they tend to shop and generally their buying habits.

2.2. Hierarchical methods and K-Means

https://doi.org/10.26219/heal.aueb.5860

In this section we will refer the hierarchical methods and the K-Means algorithm which are the clustering methods that are based on the distance. We note that the methods of clustering are divided into two categories, on the hierarchical methods and on partition methods. Firstly, we will refer some things for the distance

2.2.1. Distance

At this point it is necessary to mention how is defined the distance between the data in the clusters analysis. The distance can be set in different ways. Some of them are the below.

- Euclidean distance: The Euclidean distance is defined as follow: As we have p variables and two observations $\boldsymbol{x_i}$= (x_{i1},…….,x_{ip}) and $\boldsymbol{x_j}$= (x_{j1},…….,x_{jp}). Then the Euclidean distance between the observations $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ is

$$d(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sqrt{\sum_{j=1}^{n} (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}$$

  A negative of the Euclidean distance is that it largely depends on the measurement scale. This means that if we change the scale, we'll get different distances. Also, according to Rosie Cornish (2007) the variables with large absolute values determine the result of the distance as well as prevail over the other variables. One example on this is the following: If we have two variables, the weight (in kg) and height (in mm), then the height having larger values of the weight, will determine the distance between the observations, so the observations will be grouped primarily based on their height. To solve this problem, we subtract from each variable the mean value of the variable and divide with the standard deviation (transformation Z). In this way the variance (distance) between the observations is reduced. The Euclidean distance is used most often.

 https://doi.org/10.26219/heal.aueb.5860

- Manhattan distance: If we have two observations $\boldsymbol{x_i}$= (x_{i1},…….,x_{ip}) and $\boldsymbol{x_j}$= (x_{j1},…….,x_{jp}), then the Manhattan distance between the observations $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ is defined as

$$d(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sum_{j=1}^{n} |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|$$

  The distance Manhattan looks like the Euclidean distance, so it gives about the same results as Euclidean.

- Chebyshev distance: If we have two observations $\boldsymbol{x_i}$= (x_{i1},…….,x_{ip}) and $\boldsymbol{x_j}$= (x_{j1},…….,x_{jp}), then the Chebushev distance between the observations $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ is defined as

$$d(\boldsymbol{x_i}, \boldsymbol{x_j}) = \max(x_{ik} - x_{jk}) \text{ για } k = 1, .., p$$

  The distance Chebyshev uses only the values of that variable for the two observations which the deviation is the largest.

- Minkowski distance: If we have two observations $\boldsymbol{x_i}$= (x_{i1},…….,x_{ip}) and $\boldsymbol{x_j}$= (x_{j1},………,x_{jp}) then the Minkowski distance between the observations $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$ is defined as

$$d(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sqrt[q]{\sum_{j=1}^{n} (x_{i1} - x_{j1})^q + (x_{i2} - x_{j2})^q + \cdots + (x_{ip} - x_{jp})^q}$$

  In the distance of Minkowski if we replace where q =2, then we get the Euclidean distance.

2.2.2. The Methods

The hierarchical methods are divided into the agglomerative and the divisive. The K-Means algorithm belongs to partition methods as well as the model-based clustering which we develop in the next paragraph.

https://doi.org/10.26219/heal.aueb.5860

Divisive: Initially, we have all the observations (total n) to a cluster. The observation which differs more from the other comes out of the cluster constituting a cluster itself. So, now in the large cluster the number of observations is $n-1$. We apply again the same for the observations belonging to the large cluster. So, we measure the distance between the observations (with one of the distances reported in paragraph 2.1). The observation that it differs more from the rest, is removed from the cluster. This observation will either form a cluster itself either form a cluster with the previous observation that left from the large cluster in the previous step. In the end, we will have formed clusters, where the distances between the observations within the clusters is smaller compared with the distance of the remaining data that are in different clusters.

Agglomerative: These algorithms are more widespread compared with the divisive. According to these algorithms, each observation is a cluster. So, if we have n observations, we have initially n clusters. By measuring the distance between the clusters (initially n in a number), we join the clusters that are closest, i.e. that the distance between them is less than the distance between other clusters. Namely, in a first step we are joining the clusters (observations) that the distance between them is the smallest of all the distances. So, these two observations would form themselves a cluster. In a second step we measure again the distances between the clusters (in this step, the number of groups that consist of one observation is $n-2$ and the number of groups that consist of two observations is 1) and either joining two observations (depending on whether it is the minimum distance), or joining with the cluster which has two observations, one observation, so now the cluster which had two observations has 3 observations. These steps will be repeated until all observations belong to one cluster.

https://doi.org/10.26219/heal.aueb.5860

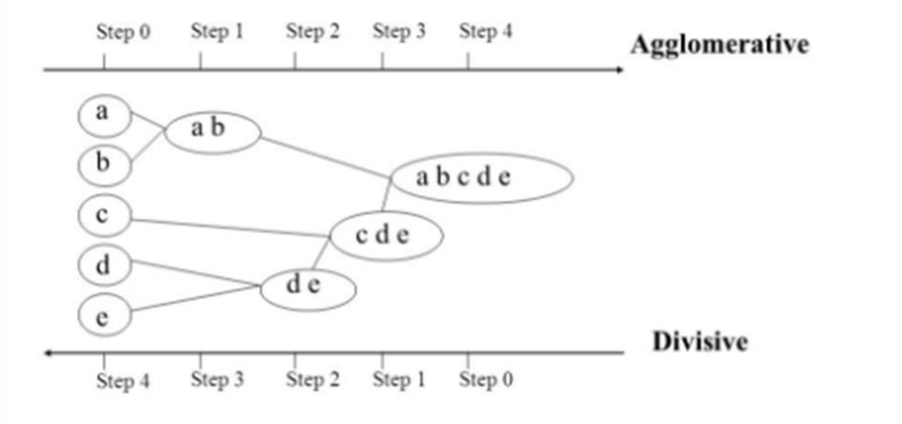

Figure 2.1: Agglomerative and divisive clustering in the data set {a, b, c, d, e}

We note that the hierarchical methods are not suitable for large data sets as well if we have too many observations, we will have a computational cost in finding, in each step, all the distances between the clusters.

But how do we choose the distance between the clusters? Some of the methods by which we can measure the distance between the clusters are the following:

i. Single linkage or the method of the nearest neighbor, where the distance between two cluster is determined by the distance of the two closest observations in the different clusters.

ii. Complete linkage or the method of the longest neighbor, i.e. the distance is determined by the largest distance between two observations that belong to different clusters.

iii. Centroid (Mean) linkage, where is measured the distance of centers between the two clusters.

iv. Average linkage, where it is calculated the average value of all the distances between all pairs of observations between the two clusters. As an example, we suppose that we have two clusters where the first cluster consists of observations 1, 2 and the second from the observations of 3,4,5. The average distance is the average value of the distances between in pairs (1, 3), (1,4), (1,5), (2,3), (2,4), (2,5).

In Figure 2 appears the distances between the clusters. The SL corresponds to the single linkage, CL in complete linkage, the AL in average linkage, and finally ML of the mean linkage

https://doi.org/10.26219/heal.aueb.5860

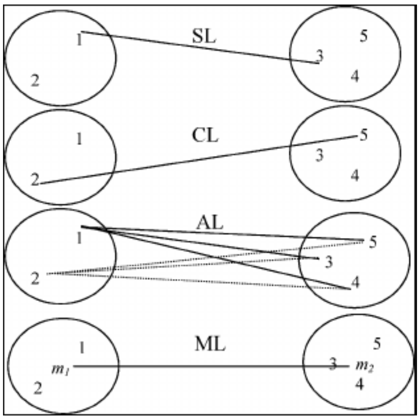

Figure 2.2: The distances between the clusters in hierarchical clustering

It is important to not confuse these distances with the distances reported in paragraph 2.1, as with the distances in paragraph 2.1 we measure the distance between the observations, while with the distances reported in this paragraph, we measure the distance between the clusters.

Next, is to develop the K-Means algorithm which belongs to partition methods

K-Means: We consider that the number of clusters is known. We start from some initial centers. We group the observations according to their distance from the centers of the clusters. So, an observation will be grouped to the cluster whose center is closer to it. After group the observations into clusters, we calculate the new centers of the clusters. If the new centers don't differ from the old initial centers, the algorithm stop, otherwise, group the observations into clusters depending on their distance from the new centers. This algorithm is repeated until the new and old centers do not differ. The algorithm K-means uses the Euclidean distance.

How is defined the center in a cluster? The center $\boldsymbol{\mu_k}$= $(\mu_{k1}, \mu_{k2},\ldots, \mu_{kp})$ of a cluster k is a vector of means, where each mean is the mean value of the observations of the k cluster in each of the p variables. For example, μ_{k1} is the mean of k cluster in the first variable.

In contrast to the hierarchical methods, the algorithm of K-means, is suitable for large data sets. But one of the negatives of the algorithm is that we need to

 https://doi.org/10.26219/heal.aueb.5860

know the number of clusters from the start. If we call K the number of clusters, we will need to run the algorithm with different values of K. Another negative of the algorithm is that it depends on the initial centers, which if we don't find properly, it can lead to a very different clustering. For this reason, we need to run the algorithm with several different initial centers. Also, George Karypis et al (1999) claimed that the K-Means algorithm as well as hierarchical methods fail on clusters of different sizes

2.3. Model-Based Clustering

The previous methods of clustering we analyzed are not supported in a model. So, we can't do statistical inference. A different approach which is able to provide the tools for statistical inference is model-based clustering. In this approach, as its name indicates we consider a model for the observations in the sample. We consider that the observations come from two or more populations, where each population is described by a distribution. Here we are going to focus on that each population follows the normal distribution. More specifically, we consider that we have n observations and p variables, i.e. each observation has p dimensions, so it is $\boldsymbol{x_i}= (x_{i1},\dots\dots,x_{ip})$.

Without loss of generality, we center the data, that is, for example, $\sum_{i=1}^{n} x_{ij} = 0$ for each j variable, $1 \le j \le p$, so each variable has a mean value 0. We also consider that the observations are independent of each other. Suppose that we have K populations, then the above assumptions are summarized in the assumption that the data follow the distribution

$$f(\boldsymbol{x_i}) = \sum_{k=1}^{K} w_k \varphi(\boldsymbol{x_i}, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$$

i.e., a mix of distributions, where w_k ($w_k \ge 0$ και $\sum_{k=1}^{K} w_k = 1$) is the probability that a randomly selected observation $\boldsymbol{x_i}$ comes from the k population (cluster), $1 \le k \le K$. $\varphi(\boldsymbol{x_i}, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$ is a Gaussian distribution with $\boldsymbol{\mu_k}= (\mu_{k,1},\dots,\mu_{k,p})$, i.e. $\boldsymbol{\mu_k}$ is the vector of means of the k cluster on p variables, and $\boldsymbol{\Sigma_k}$ is variance-covariance matrix of the k cluster.

https://doi.org/10.26219/heal.aueb.5860

The type of Gaussian distribution is the following:

$$\varphi(\boldsymbol{x_i}, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}) = \frac{1}{(2\pi)^{\frac{p}{2}} \det(\boldsymbol{\Sigma_k})^{\frac{1}{2}}} \times \exp\left\{-\frac{1}{2}(\boldsymbol{x_i} - \boldsymbol{\mu_k})\boldsymbol{\Sigma_k^{-1}}(\boldsymbol{x_i} - \boldsymbol{\mu_k})^T\right\}$$

To get information about the clusters, we need to estimate their means $\boldsymbol{\mu_k}$ for $1 \leq k \leq K$, as well as the $\boldsymbol{\Sigma_k}$, $1 \leq k \leq K$.

2.3.1. The variance-covariance matrix analysis in high dimensional data sets

We note that the variance-covariance matrix $\boldsymbol{\Sigma_k}$ in clusters, describe some geometrical characteristics. These features are their shape, their orientation and their volume. More specifically as Jeffrey D. Banfield and Adrian E. Raftery, (1993) claim, the matrix of variance-covariance $\boldsymbol{\Sigma_k}$ for the k cluster is written as follows:

$$\boldsymbol{\Sigma_k} = \boldsymbol{D_k}\boldsymbol{\Lambda_k}\boldsymbol{D_k^T}$$

$\boldsymbol{\Lambda_k}$ is a diagonal matrix which has the eigenvalues of the matrix $\boldsymbol{\Sigma_k}$ in the diagonal

$\boldsymbol{D_k}$ is an orthogonal[1] matrix of eigenvectors and determines the orientation of the cluster k for $1 \leq k \leq K$.

The matrix $\boldsymbol{\Lambda_k}$ can be written as

$$\boldsymbol{\Lambda_k} = \lambda_k \boldsymbol{A_k}$$

where λ_k corresponding to the maximum eigenvalue of the $\boldsymbol{\Sigma_k}$ and describes the volume of the k cluster, and $\boldsymbol{A_k} = \mathrm{diag}(a_{1,k}\ldots\ldots, a_{p,k})$ for 1=$a_{1,} \geq a_{2,k} \geq \ldots \geq a_{p,k} > 0$. $\boldsymbol{A_k}$ is a diagonal matrix which has elements proportional to the eigenvalues of the matrix $\boldsymbol{\Sigma_k}$ and describes the shape of the cluster k.

So, the variance covariance matrix can be rewritten as

[1] A square matrix B will be orthogonal if it satisfies the following condition:

$$B = B^{-1}$$

The orthogonal matrix B does satisfy the following relation

$$BB^T = B^T B = I$$

where I is the identity matrix of same order as B .

 https://doi.org/10.26219/heal.aueb.5860

$$\boldsymbol{\Sigma_k} = \lambda_k \boldsymbol{D_k A_k D_k^T}$$

These three features may be common between the K clusters. For example, if we assume that the K clusters have the equal shape, then it will be $\boldsymbol{A_k} = \boldsymbol{A}$. So, the variance-covariance matrix will be written as

$$\boldsymbol{\Sigma_k} = \lambda_k \boldsymbol{D_k A D_k^T}$$

If we assume that all the clusters have the equal shape and equal volume, but variable orientation, then the variance-covariance matrix will be written as

$$\boldsymbol{\Sigma_k} = \lambda \boldsymbol{D_k A D_k}^{\boldsymbol{T}}$$

In the Table 2.1, there are all the combinations of shape, orientation and volume between the clusters according to Adrian E. Raftery, Nema Dean, (2006)

Name	Model	Distribution	Volume	Shape	Orientation
EII	λI	Spherical	equal	equal	NA
VII	$\lambda_g I$	Spherical	variable	equal	NA
EEI	λA	Diagonal	equal	equal	coordinate axes
VEI	$\lambda_g A$	Diagonal	variable	equal	coordinate axes
EVI	λA_g	Diagonal	equal	variable	coordinate axes
VVI	$\lambda_g A_g$	Diagonal	variable	variable	coordinate axes
EEE	λDAD^T	Ellipsoidal	equal	equal	equal
VVV	$\lambda_g D_g A_g D_g^T$	Ellipsoidal	variable	variable	variable
EEV	$\lambda D_g A D_g^T$	Ellipsoidal	equal	equal	variable
VEV	$\lambda_g D_g A D_g^T$	Ellipsoidal	variable	equal	variable

Table 2.1: The covariance matrix of the clusters according to their shape, their volume and their orientation.

Because in this paper we focus on high dimensional data sets where the number of variables is high, and the number of data is low and with the purpose of selecting the right variables to make a correct clustering (the purpose of the work is analyzed in the next pages of this Chapter), we want to have as few parameters as possible to estimate. So, we will assume that all clusters have the same variance-covariance matrix $\boldsymbol{\Sigma_k} = \boldsymbol{\Sigma}$, which is diagonal, i.e. it is $\boldsymbol{\Sigma_k} = \boldsymbol{\Sigma}$ =diag $(\sigma_1^2, \sigma_2^2, \ldots, \sigma_p^2)$. That is, the shape and the volume in clusters are equal. This case is common in the high dimensional data sets, since we have to estimate fewer parameters. In paragraph 2.4 we refer an example with the number of parameters to be estimated in different types of covariance matrix. There are

https://doi.org/10.26219/heal.aueb.5860

also several tasks which show that in a gene expression data sets where the number of variables is too large, the classification algorithms lead to better results when the correlation between the variables is ignored. Such works are of Pedro Domingos (1997), David D. Lewis (1998). So, based on the above hypothesis, the researchers of the methods we will develop in the next chapter do not take into account the correlation between the variables in model-based clustering.

2.3.2. EM algorithm in Model-based clustering

A first thought for estimating the parameters, which are $\boldsymbol{\theta}=\{w_1,\dots w_k,\boldsymbol{\mu_1},\dots,\boldsymbol{\mu_k},\sigma_1^2,\dots\sigma_p^2\}$, is to find the log likelihood and estimate the MLEs of them. The log likelihood is

$$l(\boldsymbol{\theta})=\sum_{i=1}^{n}\log\left\{\sum_{k=1}^{K}w_k\varphi(\boldsymbol{x_i};\boldsymbol{\mu_k},\boldsymbol{\Sigma})\right\} \qquad (2.2.1)$$

However, finding of MLEs using (2.2.1) it is very difficult to do.
For this reason, we use the algorithm EM (Expectation-Maximization)

This algorithm is used when we have missing data. So, we deal with the problem like we have missing data. The missing data is that we do not know in which cluster belong our data. We consider the random variable of $Z_{i,k}$ where $Z_{i,k}$=1 when the i observation belongs to the k cluster and $Z_{i,k}$=0, otherwise. If the random variables $Z_{i,k}$ were observed, the loglikelihood is given by

$$\sum_{i=1}^{n}\sum_{k=1}^{K}Z_{i,k}\{\log w_k+\log\varphi(\boldsymbol{x_i};\boldsymbol{\mu_k},\boldsymbol{\Sigma})\} \qquad (2.2.2)$$

The EM algorithm consists of two steps: The E step where we estimate the missing data, i.e. we find the $\hat{Z}_{i,k}$ and the M step, where we put in (2.2.2) the $\hat{Z}_{i,k}$ to maximize the likelihood and to find the estimates of the parameters that we're looking for.

In more detail, according to Chris Fraley and Adrian E. Raftery, (1998)
We consider initial values $w_1,\dots w_K,\boldsymbol{\mu_1},\dots,\boldsymbol{\mu_K},\boldsymbol{\Sigma}$

 https://doi.org/10.26219/heal.aueb.5860

E-STEP

In this step we estimate the $Z_{i,k}$ as following

$$\hat{Z}_{i,k}^{(t+1)} = \frac{\widehat{w}_k^{(t)} \varphi(\boldsymbol{x_i}; \widehat{\boldsymbol{\mu}}_k^{(t)}, \widehat{\boldsymbol{\Sigma}}^{(t)})}{\sum_{k'=1}^{K} \widehat{w}_{k'}^{(t)} \varphi(\boldsymbol{x_i}; \widehat{\boldsymbol{\mu}}_{k'}^{(t)}, \widehat{\boldsymbol{\Sigma}}^{(t)})} =$$

$$= \frac{\widehat{w}_k^{(t)} (2\pi)^{-p/2} \, |\boldsymbol{\Sigma}|^{-1/2} \exp(-\frac{1}{2} \frac{\left(x_i - \hat{\mu}_k^{(t)}\right) \Sigma^{-1} \left(x_i - \hat{\mu}_k^{(t)}\right)'}{2})}{\sum_{k'=1}^{K} \widehat{w}_{k'}^{(t)} (2\pi)^{-p/2} \, |\boldsymbol{\Sigma}|^{-1/2} \exp(-\frac{1}{2} \frac{\left(x_i - \hat{\mu}_{k'}^{(t)}\right) \Sigma^{-1} \left(x_i - \hat{\mu}_{k'}^{(t)}\right)'}{2})}$$

$t = 0,1,2,3,\ldots.$ For $t = 0$, the $\widehat{w}_k^{(0)}$ is the initial value for estimating the w_k. Similar for $t = 0$, the $\widehat{\boldsymbol{\mu}}_k^{(0)}, \widehat{\boldsymbol{\Sigma}}^{(0)}$ are the initial values for estimating the $\boldsymbol{\mu_k}, \boldsymbol{\Sigma}$. So, putting the estimates of the missing data $\hat{Z}_{i,k}^{(t+1)}$ in (2.2.2), this will be

$$Q^{(t+1)} = \sum_{i=1}^{n} \sum_{k=1}^{K} \hat{Z}_{i,k}^{(t+1)} \{\log w_k + \log \varphi(\boldsymbol{x_i}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma})\} \tag{2.2.3}$$

M-STEP

In this step we maximize (2.2.3), in order to find the estimators of the parameters that we're looking for, finding the derivatives for the parameter we want to estimate and equating the derivatives equal to 0 as below:

$$\frac{dQ}{dw_k} = 0$$

Solving for w_k, we find that

$$\widehat{w}_k^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{Z}_{i,k}^{(t+1)}}{n}$$

We continue with the covariances for $1 \le k \le K$.

$$\frac{\partial Q}{\partial \sigma_j^2} = 0$$

Solving for σ_j^2, we find that

https://doi.org/10.26219/heal.aueb.5860

$$(\hat{\sigma}_j^{(t+1)})^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\hat{Z}_{i,k}{}^{(t+1)}(x_{i,j} - \hat{\mu}_{k,j}^{(t)})^2 \ \ ,1 \le j \le p$$

The last parameters which we want to estimate are the cluster means $\boldsymbol{\mu_k}, 1 \le k \le K$

$$\frac{\partial Q}{\partial \boldsymbol{\mu_k}} = 0$$

Solving for $\boldsymbol{\mu_k}$, we find that

$$\widehat{\boldsymbol{\mu}}_k^{(t+1)} = \frac{\sum_{i=1}^{n}\hat{Z}_{i,k}{}^{(t+1)}(\boldsymbol{x_i} - \boldsymbol{\mu_k^{(t)}})(\boldsymbol{x_i} - \boldsymbol{\mu_k^{(t)}})'}{\sum_{i=1}^{n}\hat{Z}_{i,k}{}^{(t+1)}}$$

If satisfied a convergence criterion (will be mentioned below), the iterations stop, otherwise continue by going to the E-step.

So, in summary, the algorithm repeats two steps until it converges. The steps that repeats are the following:

1st step: we consider the initial values $\widehat{\boldsymbol{\mu}}^{(\mathbf{0})}, \widehat{\boldsymbol{\Sigma}}^{(\mathbf{0})}, \widehat{w}^{(0)}$ and given them we find the estimates $\hat{Z}_{i,k}$. This step is the E step

2nd step: we take the estimates $\hat{Z}_{i,k}$ and given them we estimate the $\widehat{\boldsymbol{\mu}}^{(\mathbf{1})}$, $\widehat{\boldsymbol{\Sigma}}^{(1)}$,$\widehat{w}^{(1)}$. This step is the M step

After the 2nd step, the algorithm goes back to the 1st step, i.e. given the $\widehat{\boldsymbol{\mu}}^{(\mathbf{1})}$, $\widehat{\boldsymbol{\Sigma}}^{(1)}$,$\widehat{w}^{(1)}$, we estimate the $\hat{Z}_{i,k}{}^{(2)}$ and then given them we estimate the $\widehat{\boldsymbol{\mu}}^{(\mathbf{2})}$, $\widehat{\boldsymbol{\Sigma}}^{(\mathbf{2})}$,$\widehat{\boldsymbol{w}}^{(\mathbf{2})}$, etc. until converges. Once it converges we look at the final $\hat{Z}_{i,k}$ and observe for each observation, in which cluster has the highest probability and we group it there.

At this point there will be a reference of the convergence criteria of EM algorithm.

The first criterion is the below

$$\left|\frac{Q^{(t+1)} - Q^{(t)}}{Q^{(t+1)}}\right| \le \text{tol}$$

https://doi.org/10.26219/heal.aueb.5860

Where Q is the equation from (2.2.3). The tol is a small quantity near to 0, for example tol=10^{-10}.

A second criterion is the below

$$\max(\left|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\right| \leq \text{tol}$$

that is, if the value of the parameters is not changed between two consecutive steps, the algorithm stops.

A positive of the model-based clustering is that for each observation, we have probabilities for each cluster, so an observation can belong to more than one cluster.

As mentioned above, the hierarchical methods are not suitable for large data sets, because it requires large computational power because in each step is required to calculate the distance between all the clusters.

On the other hand, the method of K-means, is suitable for large data sets, but has the disadvantage that we need to know the number of clusters as well as to have chosen correct initial centers.

Finally, the model-based clustering method has the advantage of being based on the model in relation to the previous methods and so we can do statistical inference. The number of clusters can be estimated from the data. It is also a suitable method for large data sets.

Because in the last few years technology is developed rapidly, the number of data that can be analyzed becomes more and more big. There are many data sets in which the variables are much more than the number of data. Such sets are, for example, those related with data of gene expression. The aim of this project is therefore to develop methods by which we can handle high-dimensional data sets where the number of variables is much greater than the number of data and with which we can choose automatically the correct variables. Due to the statistical inference provided by the method of model-based clustering, suitability for large data sets and estimate of the number of clusters from the data, the methods we will develop in the next chapter will rely on this method. Because there are some problems with the high dimensional data sets where the

https://doi.org/10.26219/heal.aueb.5860

dimensions are more than the number of data, we will handle them different from that data sets with a normal number of variables where the data does not exceed the number of variables.

2.4. ‘Curse of Dimensionality’

In general, the reduction of dimensions in the cluster analysis is very important when we have too much high-dimensional data sets where the number of variables is much larger than the number of observations, which makes it difficult for the clustering. A problem that is created is the so-called ‘curse of dimensionality’ (Bellman (1957)). According to this problem, while the dimensions increase, the data become increasingly ‘sparse’. An example that helps in understanding of the problem is the following (Michael Steinbach, Levent Ertöz, and Vipin Kumar,2004): suppose we have 100 points which follow a uniform random distribution in the interval [0,1]. If the interval [0,1] is broken into 10 equal cells (i.e., the length of each cell is 0.1), then it is highly likely that all cells will contain some points. If we keep the number of points the same and the unit of discretization the same (that is 0.1 for each dimension) but distribute the points over the unit square (this means that each point is two dimensional) then we’ll have 10 ×10=100 two-dimensional cells. In this case it is very likely that some cells to be empty. If we get to 3 dimensions, having again 100 points, then we will have 10×10×10=1000 three-dimensional cells. In this case, it is certain that the more cells will be empty since there are far more cells than points. So, they'd need a lot more points to have all the cells from one point. In clustering, the ‘curse of dimensionality’ affects the distance or similarity between the data because as the dimensionality increases, the difference of the distances of the closest and farthest data points of an independently selected point goes to 0, that is

$$\lim_{d\to\infty} \frac{D\text{max} - D\text{min}}{D\text{min}} = 0$$

where d is the number of dimensions, Dmax is the farthest distance and Dmin is the closest distance. Because many clustering algorithms depends on distance between the observations like K-Means (this algorithm finds the Euclidean

https://doi.org/10.26219/heal.aueb.5860

distance between the observations), the ‘curse of dimensionality’ affect the effectiveness of them. According to Charles Bouveyron, and Camille Brunet (2014), in model-based clustering, the ‘curse of dimensionality’ creates the problem that we will have too many parameters in the model (over-parametrized model), so there will be the overparameterization problem.

In classical Gaussian mixture model (GMM), the total number of parameters to estimate is

$$(K-1)+Kd+\frac{Kd(d+1)}{2}$$

where d is the number of the variables and K is the number of the clusters. The $(K-1)$, Kd, $\frac{Kd(d+1)}{2}$ are the numbers of the parameters of proportions, means and covariance matrices respectively. The number of parameters to be estimated is a quadratic function of d. So, if we have 100 variables, the number of the parameters to be estimated is at least 10000. For example, if we have 200 variables and 30 observations that are grouped to 3 clusters, the number of parameters to be estimates are 60302. That is, with 30 observations with have to estimate 60302 parameters. Consequently, we need a large number of observations to correctly estimate those model parameters. A solution for decreasing the number of the parameters is to rewrite the covariance matrix as in Table 1 in 2.2.2 paragraph. As we referred, if we assume that the covariance matrix is the same for all clusters, we will have to estimate fewer parameters. In above example, if we assume same covariance matrix in 3 clusters, the number of parameters to be estimate will be 20702, i.e. we have about 40000 fewer parameters to estimate. If the covariance matrix is
$\boldsymbol{\Sigma_k}$= $\boldsymbol{\Sigma}$ =diag$(\sigma_1^2, \sigma_2^2, \ldots., \sigma_d^2)$ for all clusters, then the number of parameters to be estimated is 802 variables

In Table 2.2 appears the number of parameters in four cases. The first case is the Full-GMM when each cluster has different covariance matrix with all the parameters into the matrix. In the second case, the clusters have the same covariance matrix. The Diag-GMM case is when the clusters have different

covariance matrix, but these matrices are diagonal. The Diag Com-GMM is the case when the clusters have the same diagonal matrix, that is
$\boldsymbol{\Sigma_k} = \boldsymbol{\Sigma}$ =diag $(\sigma_1^2, \sigma_2^2, \ldots., \sigma_d^2)$. The last case in common in high dimensional data sets

Model	***Number of parameters***	***Number of parameters for K=3, d=200***
Full- GMM	$(K-1)+Kd+Kd(d+1)/2$	60902
Com-GMM	$(K-1)+Kd+d(d+1)/2$	20702
Diag-GMM	$(K-1)+Kd+Kd$	1202
Diag Com-GMM	$(K-1)+Kd+d$	802

Table 2.2: The number of parameters in GMM in four cases

A second solution for handling the high dimensional datasets is to decrease the dimensions (variables), so that the number of parameters to decrease. It is certain that there are some variables that are noninformative for clustering, so these variables will be useless for this.

2.5. Implementation of PCA before clustering: Good or Bad?

An idea for decreasing the dimensions before clustering is to apply the method of PCA. The variables (principal components) resulting from this method is a linear combination of the original variables. The principal components are uncorrelated to each other and are ranked so that the k principal component has the k greater variance among the other principal components. So, most of the times for the reduction of dimensions, the principal components that interpret the greatest variance are retained.

There have been several researchers in the past who support the application of the PCA method to reduce dimensions in many algorithms in machine learning. More specifically, Mansfield et al. (1977, p.38) argued that if only the principal components with the highest variance are maintained, the predictive capacity of the model in regression will be minimally reduced. Mosteller and Tukey (1977, pp. 397-398) also claimed that the principal components with the lowest variances did not contribute to regression at all.

 https://doi.org/10.26219/heal.aueb.5860

On the contrary, there are also researchers that don't support the PCA method to reduce dimensions. Some of them are Kung and Sharif (1980) demonstrated with an example that the most important principal components in the regression in this example are in turn the 8th, 2nd and 10th (the 2nd is the principal component having the second largest variable etc.). This shows that even the components with the lowest variance may be useful for the model.

Regarding the application of the PCA method before clustering, Yeung and Ruzzo (2001) showed that the PCA method degrades the quality of clustering. What they did was to apply different clustering algorithms to real data by initially applying the PCA method and later without it. What they concluded was that the results without applying the PCA method before clustering were much better. Also, by applying the PCA method before clustering, they showed that the results in clustering were better when they selected the principal components that haven't the highest variances than selecting the principal components with the highest variances. According to Sijian Wang and Ji Zhu (2008) because the resulting variables are usually a linear combination of the initial variables, there is no variable selection. Finally, Liu (2013) argued that the cost of applying this method is that we lose a proportion of the total variability. Also, applying this method before clustering is not a very good idea, because reducing dimensions before it may destroy the cluster structure in data.

So, the idea is to keep the structure of the data as much as possible while at the same time reduce the dimensions. The high dimensional data sets are increasing in many areas. As the number of dimensions increases, several methods of clustering degrade the quality of the results. For this reason, it is important to be able to make variable selection in clustering. Because in clustering, the grouping of observations is unknown, which means that we don't know from the start in which cluster has grouped each observation and because there is little or no knowledge about the analysis and the structure of the clustering that is to be done, it isn't possible to apply a technique for the rejection or acceptance of variables before the cluster analysis. For this reason, it is important to include the process of variable selection within the algorithm of the clustering, i.e. the selection of variables to be part of the algorithm of clustering. The selection of variables is difficult to do in cluster algorithms in

https://doi.org/10.26219/heal.aueb.5860

which there is a lack of statistical models. For this reason, it will be applied the selection procedure of variables in the algorithm of the model-based clustering.

2.6. Definition of the informative and noninformative variables

But when a variable is noninformative for the separation between two or more clusters?

Suppose that we have a two dimensions data set. In Figure 2 (Jian Guo et al., 2010) it seems the example that we develop. We group the observations of this data set into 3 clusters. The first variable separates the clusters 2.3, while the second variable separate the clusters 1,2. So, the first variable is informative for the separation of clusters 2,3, while it is noninformative for the separation of clusters 1,2. The 2nd variable is informative for the separation of clusters 1,2, and it is noninformative for the separation of clusters 2,3.

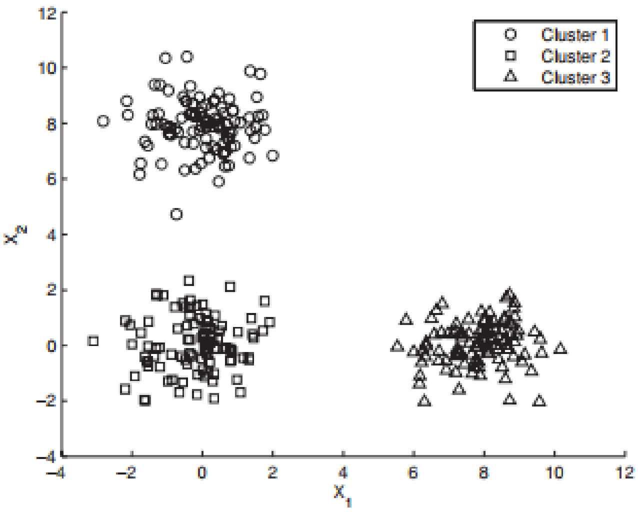

Figure 2.3: Variable 1 is informative for separating clusters 2 and 3, and variable 2 is informative for separating clusters 2 and 3

2.7. Different approaches in the issue of variable selection

The variables selection in clustering has started receiving increased attention. Several methods have been developed and some of them are listed below.

A first method is the best subset selection. According to this method, if we have p variables, we fit models with all combinations of the variables, i.e. we fit all $\binom{p}{k}$, $k = 1, \ldots, p$ and we choose the best model from the $\binom{p}{k}$, which we call M_k. M_0 is called the zero model. From the $M_0, M_1, \ldots., M_p$, we choose the best either with the AIC criterion either with the BIC criterion. However, the best subset selection is impossible to implement in data sets where there are a lot of variables. If, for example, we have p =1000 variables, there are more than 10300 possible models to be considered, which is disadvantageous in terms of time and computational cost as Wei Pan and Xiatong Shen (2007) claimed.

A second method is that of Friedman and Meulman (2004), where implement the variable selection in combination with the hierarchical clustering algorithm. This algorithm does not explicitly select variables, but assigns different weights to the variables, which can be used for extract of informative variables. Brusco and Cradit (2001) proposed a method of selecting variables with the K-Means algorithm. Because these algorithms work in combination with the algorithm of hierarchical clustering and the K-Means respectively, have serious limitations. One of them is the lack of a statistical criterion (i.e. BIC or AIC) for finding the number of clusters.

A third method is the comparison of nested models, i.e. models where have the same variables, just one of the two models has a variable above. We can decide if the above variable is significant or not with different statistical criteria (Raftery and Dean, 2006). This comparison is made using Bayesian factors. However, the method of Raftery and Dean is disadvantageous for large-dimensional set of data, as it has a computational cost.

Other methods are the methods that we will develop in the third chapter, which is appropriate for large sets of data. A positive of these methods is that the procedure of automatic selection of variables is assigned in the algorithm of clustering. More specifically, these methods realize automatic variable selection by subtracting from the type of (2.2.1) a separate quantity of each. The term that is subtracted is called penalty and is responsible for the selection of variables. These methods are based on a statistical model (the advantages of having statistical model mentioned above). Also, these methods don't have a large computational cost, that is very important for the problem that we study,

https://doi.org/10.26219/heal.aueb.5860

which is the selection of variables in high-dimensional data set where the number of variables is greater than the number of observations.

https://doi.org/10.26219/heal.aueb.5860

CHAPTER 3

DESCRIPTION OF METHODS WITH AUTOMATIC VARIABLE SELECTION

3.1. Introduction

Because as we mentioned in the previous section, the GMM (Gaussian mixture model without penalty) does not make variable selection, we will refer 4 methods of doing automatic variable selection, without applying a dimensional reduction algorithm before cluster analysis, such as PCA (which spoils the clustering structure in data) and without increasing computational costs. These methods are:

I. L_1 penalty which was developed from Wei Pan, Xiatong Shen (2007)

II. ALP-GMM (L_∞ penalty) and AHP-GMM (Hierarchical-GMM) (Sijian Wang and Ji Zhu, 2008)

III. Pairwise Fusion Penalty (PFP). We also refer the adaptive version of it, that is the Adaptive Pairwise Fusion Penalty (APFP) (Jian Guo, Elizaveta Levina, George Michailidis and Ji Zhu, 2010)

These methods differ from the GMM, because they subtract from the likelihood of GMM a penalty different in each of the 4 methods.

Also, mention that we will be dealing with high dimensions data sets, where the number of variables is larger than the number of observations. So, if n is the number of observations and p the number of variables, then $n << p$.

When we have too many variables, i.e. too many dimensions, it is important to choose correctly the variables in order to stay in the model only the informative variables and to remove noninformative variables. In this way, it will become easier the interpretation.

In the next paragraphs of the 3rd chapter detailed the methods mentioned above, starting from the method of L_1 penalty.

3.2. L_1 penalty

 https://doi.org/10.26219/heal.aueb.5860

This method was developed by Wei Pan, Xiao Shen (2007). In accordance with this method, a variable is noninformative for the separation of two clusters k, k', when the estimated means of the clusters are equal to the mean value of the variable.

Consequently, if two clusters have the same means that are different from the mean value of the variable, the variable will be informative for separating these clusters. In general, a variable is considered to be noninformative and excluded from the model, if the means of all the clusters associated with this variable are equal with the mean value of the variable. That is, if $\mu_{k,j} = \varphi_j + \delta_{k,j}$, where $\mu_{k,j}$ is the mean of the k cluster in a variable j, φ_j is the mean value of the variable j, and $\delta_{k,j}$ is a value that shows how far or close is the mean of the k cluster of the j variable from the mean value of the variable, then the variable j will be considered noninformative if $\delta_{k,j}$=0 for every

$1 \leq k \leq K$. And so, the variable won't be included in the model which we will develop below.

In more detail, we believe that we have a data set with large dimensions, i.e. n observations and p variables, where $n << p$, which we can organize it in the matrix

$$X_{i,j} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix}$$

We assume that the data are generated from a Gaussian mixture distribution, where each component of the Gaussian corresponding to a cluster. Specifically, the observations $\boldsymbol{x_i}$= $(x_{i1},\ldots\ldots,x_{ip})$ are assumed to be independent and generated from the density

$$f(\boldsymbol{x_i}) = \sum_{i=1}^{n} w_k \varphi(\boldsymbol{x_i}, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$$

A further simplifying assumption is that the variance-covariance matrix $\boldsymbol{\Sigma_k}$ is the same for all clusters and diagonal, i.e. $\boldsymbol{\Sigma_k} = \boldsymbol{\Sigma}$ =diag $(\sigma_1^2,\ldots,\sigma_p^2)$ for each

 https://doi.org/10.26219/heal.aueb.5860

cluster $1 \leq k \leq K$. This assumption is common in high-dimensional data sets because we have little computational costs, since we'll have to estimate only the parameters in the diagonal, that is $\sigma_1^2,\ldots,\sigma_p^2$ (we have mentioned the reasons in the previous chapter).

Before clustering we standardized the data. So, each variable has mean value 0 and variance 1.

The likelihood function in this method is

$$\log Lp(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log[\sum_{k=1}^{K} w_k \varphi(\boldsymbol{x_i}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma})] - h_\lambda(\boldsymbol{\theta}) \tag{3.2.1}$$

where the quantity $h_\lambda(\boldsymbol{\theta})$ is the penalty defined as $h_\lambda(\boldsymbol{\theta}) = \lambda \sum_k \sum_j |\mu_{kj}|$. The w_k is the probability an observation to belong to k cluster, $0 \leq w_k \leq 1$, $\sum w_k = 1$, $\boldsymbol{\Sigma}$ is the variance covariance matrix which is the same for all clusters, $\mu_{k,j}$ is the mean value of k cluster in j variable and λ is the shrinkage parameter for which we choose the values. According to Wang, Zhu (2008), as we can see, if we have an observation $\boldsymbol{x}^* = (x_1^*,..,x_p^*)$ and for the j variable all the means are shrunken to zero, that is $\mu_{kj} = 0,\ k = 1,..,K$ then

$$\exp\left(-\frac{(x_j^* - \mu_{kj})^2}{2\sigma_j^2}\right) = \exp\left(-\frac{(x_j^*)^2}{2\sigma_j^2}\right)$$

that does not depend on k. So, the j variable doesn't contribute on clustering. It means that the j variable is noninformative and so it can be removed for the model.

Because the maximization of (3.2.1) is difficult to be done, we will handle the problem like we have missing data, using EM algorithm.

Listed below are the steps of the algorithm as they appear in the work of Wei Pan work, Xiaotong Shen (2007).

We call $\tau_{i,k}$ the random variable which is equal to 1 if the observation i belongs to k cluster and 0 otherwise.

So, the likelihood will be,

https://doi.org/10.26219/heal.aueb.5860

$$\sum_{k=1}^{K}\sum_{i=1}^{n}\tau_{i,k}\{\log w_k+\log\varphi(\boldsymbol{x_i};\boldsymbol{\mu_k},\boldsymbol{\Sigma})\}-\lambda\sum_{k}\sum_{j}|\mu_{k,j}| \qquad (3.2.2)$$

The E-Step will be the same as the EM algorithm in GMM. The only difference will be in M step in which the estimated $\mu_{k,j}$ will be found with different way. We assume the initial values

$\boldsymbol{\theta}^{(\mathbf{0})}=\left\{w_1^{(0)},\dots,w_K^{(0)},\boldsymbol{\mu}_{\mathbf{1}}^{(\mathbf{0})},\dots,\boldsymbol{\mu}_{\boldsymbol{K}}^{(\mathbf{0})},\boldsymbol{\Sigma}^{(\mathbf{0})}\right\}$ which is for t=0.

E-STEP

The estimate of missing $\tau_{i,k}$ will be as follows: For the $(t+1)$ step,

$$\hat{\tau}_{i,k}{}^{(t+1)}=E(\tau_{i,k}|X,\widehat{\boldsymbol{\theta}}^t)=\mathrm{P}(\tau_{i,k}=1|X,\widehat{\boldsymbol{\theta}}^t)=\frac{\widehat{w}_k^{(t)}\varphi(\boldsymbol{x_i};\widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(t)},\widehat{\boldsymbol{\Sigma}}^{(t)})}{\sum_{k=1}^{K}\widehat{w}_k^{(t)}\varphi(\boldsymbol{x_i};\widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(t)},\widehat{\boldsymbol{\Sigma}}^{(t)})}=$$

$$=\frac{\widehat{w}_k^{(t)}(2\pi)^{-p/2}\,|\boldsymbol{\Sigma}|^{-1/2}\exp(-\frac{1}{2}\frac{\left(x_i-\widehat{\mu}_k^{(t)}\right)\boldsymbol{\Sigma}^{-1}\left(x_i-\widehat{\mu}_k^{(t)}\right)'}{2})}{\sum_{k=1}^{K}\widehat{w}_k^{(t)}(2\pi)^{-p/2}\,|\boldsymbol{\Sigma}|^{-1/2}\exp(-\frac{1}{2}\frac{\left(x_i-\widehat{\mu}_k^{(t)}\right)\boldsymbol{\Sigma}^{-1}\left(x_i-\widehat{\mu}_k^{(t)}\right)'}{2})},$$

where $t=0,1,2,\dots.$ (3.2.3)

Note that for $t=0$, the $\hat{\tau}_{i,k}{}^{(1)}$ are estimated from the initial values $w_1^{(0)},\dots,w_K^{(0)},\boldsymbol{\mu}_{\mathbf{1}}^{(\mathbf{0})},\dots,\boldsymbol{\mu}_{\boldsymbol{K}}^{(\mathbf{0})},\boldsymbol{\Sigma}^{(\mathbf{0})}$

So, if we replace the estimated $\hat{\tau}_{i,k}{}^{(t+1)}$ in (3.1.2) it will be

$$Q=\sum_{k=1}^{K}\sum_{i=1}^{n}\hat{\tau}_{i,k}^{(t+1)}\{\log w_k+\log\varphi(\boldsymbol{x_i};\boldsymbol{\mu_k},\boldsymbol{\Sigma})\}-\lambda\sum_{k=1}^{K}\sum_{j=1}^{p}|\mu_{k,j}| \qquad (3.2.4)$$

M-STEP

As in GMM, we will maximize (3.2.4) to find the estimators of the parameters we are looking for as following

https://doi.org/10.26219/heal.aueb.5860

$$\frac{dQ}{dw_k} = \sum_{i=1}^{n}\left(\frac{\tau_{i,k}^{(m)}}{w_k} - \frac{\tau_{i,K}^{(m)}}{w_K}\right) = 0 \text{ for } k = 1, \dots, K-1.$$

Solving to w_k we find that

$$\widehat{w}_k^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{\tau}_{i,k}{}^{(t+1)}}{n}$$

Continue with the diagonal parameters of variance-covariance matrix:

$$\frac{\partial Q}{\partial \sigma_j^2} = \sum_k \sum_i \hat{\tau}_{i,k}^{(m)} \left[-\frac{1}{2\sigma_j^2} + \frac{\left(x_{ij-}\mu_{kj}\right)^2}{2\sigma_j^4} \right].$$

$\frac{\partial Q}{\partial \sigma_j^2} = 0$. Solving to σ_j^2, it will be as following

$$(\hat{\sigma}_j^{(t+1)})^2 = \frac{1}{n} \sum_k \sum_i \hat{\tau}_{i,k}{}^{(t+1)} \left(x_{i,j} - \hat{\mu}_{k,j}^{(t)}\right)^2 \ , 1 \le j \le p, \qquad 1 \le k \le K$$

The cluster means are estimated with different way from GMM as following:

$$\frac{\partial Q}{\partial \boldsymbol{\mu_k}} = \sum_i \hat{\tau}_{i,k}{}^{(t+1)} \, V^{-1}(x_i - \boldsymbol{\mu_k}) - \lambda sign(\boldsymbol{\mu_k})$$

After many calculations, we end up in the following:

$$\widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(m+1)} = sign(\widetilde{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(m+1)})(\left|\widetilde{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(\boldsymbol{m+1})}\right| - \frac{\lambda}{\sum_i \tau_{i,k}^{(m+1)}} V^{(m+1)} 1)_+$$

where $\widetilde{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(\boldsymbol{m+1})} = \frac{\sum_i \tau_{i,k}^{(m+1)} x_i}{\sum_i \tau_{i,k}^{(m+1)}}$, is the mean of k cluster, for λ=0, i.e. with no penalty. For each $f, f+ = f$ if $f > 0$ and $f+ = 0$ if $f < 0$. 1 is a vector with all elements equal to 1. It turns out that if λ>$\left|\frac{\sum_i \tau_{i,k}^{(m+1)} x_{ij}}{\sigma_j^{2(m+1)}}\right|$, then $\widehat{\mu}_{kj}^{(m+1)} = 0$, otherwise the $\widehat{\mu}_{kj}^{(m+1)}$ are estimated, shrinking $\widetilde{\mu}_{kj}^{(m+1)}$ toward to 0 from the quantity $\frac{\lambda}{\sum_i \tau_{i,k}^{(m+1)}} \sigma_j^{2\,(m+1)}$. If this quantity is very large, i.e. near to $\widetilde{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(\boldsymbol{m+1})}$, then $\widetilde{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(\boldsymbol{m+1})}$ converges to 0.

The above algorithm is repeated until it converges, so after we take the MPLE $\widehat{\boldsymbol{\theta}}$. Then we use (3.2.3) to compute the posterior probabilities and we group the observations in the cluster with the highest probability. As it applies

 https://doi.org/10.26219/heal.aueb.5860

to all computational optimization problems, because there may be many local maxima, it is a good idea to run the EM algorithm with many different initial values to ensure that it does not converge to any local maxima. It can be shown that if $\hat{\mu}_{kj}$=0 for all k, then the variable j doesn't contribute to the clustering, so it will leave from the numerator and denominator of (3.2.3). On the contrary, in the clustering without penalty (λ=0), all variables contribute to the calculation of the posterior probabilities.

The parameters that we have to choose are K, λ. Initially we mention that in the standard model (i.e. GMM without penalty), we have to choose only the number of clusters, K. By selecting several values for K , we use the BIC criterion to arrive at the best model (that with the lowest BIC), and so in the number K that corresponds to the particular model. The type of BIC for the standard, is

$$\mathrm{BIC} = -2\log L(\widetilde{\boldsymbol{\theta}}) + \log(n)\, d$$

where $\widetilde{\boldsymbol{\theta}}$ is the MLE of $\boldsymbol{\theta}$, and d=dim ($\widetilde{\boldsymbol{\theta}}$) is the number of unknown parameters that we have to estimate. Because we have three sets of parameters, w_k, σ_j, $\mu_{k,j}$ with the restriction $\sum_{k=1}^{K} w_k = 1$, d will be $d = K + p + Kp - 1$.

In penalized clustering, in addition to K, we have to choose the shrinkage parameter λ. Following Zou et al. (2004), who refers that the number of non-zero parameters we need to estimate is unbiased estimator of the degrees of freedom in the method of LASSO (method of selection of variables), we will handle the d, as the number of parameters that have to be estimated and that are different from 0. So, for the penalized clustering, the type of BIC is

$$\mathrm{BIC} = -2\log L(\widetilde{\boldsymbol{\theta}}) + \log(n)\, d_e$$

where $\widetilde{\boldsymbol{\theta}}$ is the MPLE of $\boldsymbol{\theta}$, i.e. MLE for penalized clustering, and $de = K + p + Kp - 1 - q$, where q is the number of the parameters of the cluster means that are equal to 0.

 https://doi.org/10.26219/heal.aueb.5860

According to Wang and Zhu (2008), a negative of this method is that if we have two clusters k, k' in a variable j, the method doesn't use the information that the two cluster means $\mu_{k,j}, \mu_{k',j}$ are associated with the same variable j , so they have to be treated differently from the cluster mean $\mu_{k,j'}$ which is associated with the j' variable. Because it doesn't use the information that the cluster means associated with the same variable j are treated as a group, if a variable is noninformative, it doesn't shrink to 0 all the cluster means but some of them, and so the variable j is not removed from the model. From this reason the methods in section 3.3 are developed.

According to Jian Guo et al (2010), a second negative of this method is that it considers that a variable is noninformative for separating the two clusters, if the cluster means are the same as the mean value of the variable. But if they are the same between them without be equal to the mean value of the variable, the variable will not be considered noninformative for the separation between the two clusters. Therefore, if a subset of clusters associated with a variable have the same means which are different from the mean value of the variable (that is 0 after centering the data), the method cannot identify this variable as noninformative for separating these clusters.

3.3. L_∞ penalty (ALP-GMM)-Adaptive Hierarchically Penalized GMM (AHP-GMM)

These methods were developed by Wang and Zhu (2008). As mentioned in 3.2, a negative of L_1-penalty is that the cluster means belonging to the same variable are not be treated as a 'group'. In contrast, L_∞ and hierarchical methods have the positive that they treat as a group the cluster means belonging to the same variable. So, according to Wang and Zhu, the two methods remove noninformative variables more efficiently than L_1.

As before, we will deal with high dimensional datasets where the number of variables p is much larger than the number of observations n, i.e. $n \ll p$. The observation $\boldsymbol{x_i}$ is a vector with p dimensions, i.e. $\boldsymbol{x_i} = (x_{i1}, \dots, x_{ip})$. Initially, the data are centered, e.g. $\sum_{i=1}^{n} x_{ij} = 0$ for each variable j, $1 \le j \le p$. Therefore,

 https://doi.org/10.26219/heal.aueb.5860

the mean value of each variable is 0. The purpose is to separate the data into K clusters. We assume that every observation follows Gaussian distribution:

$$f(\boldsymbol{x_i}) = \sum_{k=1}^{K} w_k \varphi\left(\boldsymbol{x_i}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}\right)$$

where w_k is the probability that an observation $\boldsymbol{x_i}$ is derived from the k cluster, $\boldsymbol{\mu_k}$ is the Gaussian distribution mean vector of the k cluster, that is, $\boldsymbol{\mu_k}$=(μ_{k1},.., μ_{kp}) and $\boldsymbol{\Sigma_k}$ is the variance-covariance matrix of k cluster. We consider that the variance matrix is the same for all clusters and diagonal, i.e. $\boldsymbol{\Sigma_k} = \boldsymbol{\Sigma}$=diag $(\sigma_1^2, .., \sigma_p^2)$. We call $\boldsymbol{\theta}$ the set of parameters, that is

$\boldsymbol{\theta} = \{w_1, \dots, w_K, \boldsymbol{\mu_1}, \dots, \boldsymbol{\mu_K}, \boldsymbol{\Sigma}\}$

The loglikelihood of GMM is,

$$\log p(X|\boldsymbol{\theta}) = \sum_{i=1}^{n} \log\left\{\sum_{k=1}^{K} w_k \varphi(\boldsymbol{x_i}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma})\right\} \quad (3.3.1)$$

Initially we will deal with the L_∞ penalty

3.3.1. L_∞ penalty

The idea of the method is that if we have K clusters in a variable j, we will look for the cluster whose mean is larger in absolute value than the means of the other clusters. Let M_j be the cluster mean with the highest absolute value. The method shrinks the M_j toward to 0. If M_j is estimated as 0 then all means of the remaining $K - 1$ clusters in variable j will be estimated as 0. Thus, the variable j will be considered noninformative and will be removed from the model.

Let's develop the method in more detail.

Because the purpose is the variable selection, Wang and Zhu suggested the following loglikelihood

$$l = \log p(X|\boldsymbol{\theta}) - \lambda \sum_{j=1}^{p} \max(|\mu_{1j}|, \dots, |\mu_{Kj}|) \quad (3.3.2)$$

where $\max(|\mu_{1j}|, \dots, |\mu_{Kj}|) = \|(\mu_{1,j}, \dots\dots, \mu_{K,j})\|_\infty$.

https://doi.org/10.26219/heal.aueb.5860

In contrast to the L_1 penalty, notice that L_∞ estimates all means as 0 if after the shrinkage the maximum absolute cluster mean is estimated as 0, that is, as mentioned above, it treats the clusters that are in the same variable as a 'group'. We will use a more improved version of (3.3.2), the adaptive LP-GMM. The type of adaptive is

$$l = \ logp(X|\boldsymbol{\theta}) - \lambda \sum_{j=1}^{p} w_j \max(|\mu_{1j}|, \ldots, |\mu_{Kj}|) \tag{3.3.3}$$

where w_j are prespecified weights. The idea is that if the j variable is noninformative for clustering, then we would like the corresponding w_j to be large, and if the variable j is informative, we would like the corresponding w_j to be small.

Because the maximization of (3.3.3) is difficult, we will deal with the problem as if we have missing data using the EM algorithm.

Let $\tau_{i,k}$=1 if the i observation belongs to the k cluster and 0 otherwise. If $\tau_{i,k}$ were all observable, loglikelihood would be

$$\sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{i,k} \{\log w_k + \log\varphi(\boldsymbol{x_i}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma})\} - \lambda \sum_{j=1}^{p} w_j \, max(|\mu_{1j}|, \ldots, |\mu_{Kj}|) \tag{3.3.4}$$

The E Step will be the same as for GMM and L_1 . In M Step the only difference will be in how we will estimate the cluster means. There we will also use a theorem.

Let the initial values $\boldsymbol{\theta}^{(\mathbf{0})} = \left\{w_1^{(0)}, \ldots, w_K^{(0)}, \boldsymbol{\mu}_{\mathbf{1}}^{(\mathbf{0})}, \ldots, \boldsymbol{\mu}_{\boldsymbol{K}}^{(\mathbf{0})}, \boldsymbol{\Sigma}^{(\mathbf{0})}\right\}$, i.e. for t=0.

E-STEP

The $\tau_{i,k}$ are missing values, so we will estimate them as following:

$$\hat{\tau}_{kj}{}^{(t+1)} = E(\tau_{k,j}|X, \widehat{\boldsymbol{\theta}}^t) = \mathrm{P}(\tau_{k,j} = 1 | X, \widehat{\boldsymbol{\theta}}^t) = \frac{\widehat{w}_k^{(t)} \varphi(\boldsymbol{x_i}; \widehat{\boldsymbol{\mu}}_k^{(t)}, \widehat{\boldsymbol{\Sigma}}^{(t)})}{\sum_{k=1}^{K} \widehat{w}_k^{(t)} \varphi(\boldsymbol{x_i}; \widehat{\boldsymbol{\mu}}_k^{(t)}, \widehat{\boldsymbol{\Sigma}}^{(t)})} =$$

 https://doi.org/10.26219/heal.aueb.5860

$$=\frac{w_k^{(t)}(2\pi)^{-p/2}\,|\boldsymbol{\Sigma}|^{-1/2}\exp\left(-\frac{1}{2}\left(x_i-\hat{\mu}_k^{(t)}\right)\Sigma^{-1}\left(x_i-\hat{\mu}_k^{(t)}\right)'\right)}{\sum_{k=1}^{K}\hat{w}_k^{(t)}(2\pi)^{-p/2}\,|\boldsymbol{\Sigma}|^{-1/2}\exp\left(-\frac{1}{2}\left(x_i-\hat{\mu}_k^{(t)}\right)\Sigma^{-1}\left(x_i-\hat{\mu}_k^{(t)}\right)'\right)}, \qquad (3.3.5)$$

where t=0,1, 2,..

We replace the estimates of (3.3.5) in (3.3.4).

We will consider the set Ω where it will contain all the cluster means in each variable, that is $\Omega = \{\mu_{kj,} k = 1,..,K; j = 1,..p\}$.

If for practical reason we will define the penalty function as following:

$$J(\Omega) = \lambda\sum_{j=1}^{p} w_j \max(|\mu_{1j}|, \dots, |\mu_{Kj}|),$$

then the log likelihood of L_∞ -penalty method will be

$$Q = \sum_{k=1}^{K}\sum_{i=1}^{n}\hat{\tau}_{i,k}^{(t+1)}\{\log w_k + \log\varphi(\boldsymbol{x_i};\boldsymbol{\mu_k},\boldsymbol{\Sigma})\} - J(\Omega) \qquad (3.3.6)$$

M-STEP

We will maximize the (3.3.6) for finding the estimates of parameters.

$\frac{dQ}{dw_k} = 0$. Solving to w_k we find for the t+1 step:

$$\hat{w}_k^{(t+1)} = \frac{\sum_{i=1}^{n}\hat{\tau}_{i,k}{}^{(t+1)}}{n}$$

Continue with variances:

$\frac{\partial Q}{\partial\sigma_j^2} = 0$. Solving to σ_j^2, it will be:

$$(\hat{\sigma}_j^{(t+1)})^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\hat{\tau}_{i,k}{}^{(t+1)}(x_{i,j} - \hat{\mu}_{k,j}^{(t)})^2 \ ,1 \le j \le p$$

Next is the estimates of means:

In $(t+1)$ step the means will be

$$\hat{\Omega}^{t+1} = \operatorname{argmax}\left\{-\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{j=1}^{p}\hat{\tau}_{i,k}{}^{(t+1)}\frac{\left(x_{i,j}-\hat{\mu}_{k,j}^{(t)}\right)^2}{\hat{\sigma}_j^{2(t+1)}} - J(\Omega)\right\} \qquad (3.3.7)$$

Argmax is defined as the point where the maximization of Ω is achieved. This definition is the same as changing the signs within the brackets and finding the point where the minimization of Ω is achieved.

https://doi.org/10.26219/heal.aueb.5860

Estimation of μ_{kj} in ALP-GMM

If we change the signs in (3.3.7), it will be

$$\min\left\{\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{j=1}^{p}\hat{\tau}_{i,k}\frac{\left(x_{i,j}-\mu_{k,j}\right)^2}{\sigma_j^2}+J(\Omega)\right\} \tag{3.3.8}$$

The (3.3.8) can be separated in p minimizations problems, that is for each variable we will have a minimization problem, so the (3.3.8) will be written as

$$\min\left\{\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{i,k}\frac{\left(x_{i,j}-\mu_{k,j}\right)^2}{\sigma_j^2}+\lambda w_j \max\left(\left|\mu_{1j}\right|,\dots,\left|\mu_{Kj}\right|\right)\right\} \tag{3.3.9}$$

for each $j = 1,2,..,p$.

We define M_j as M_j=$\max\left(\left|\mu_{1j}\right|,\dots,\left|\mu_{Kj}\right|\right)$, so we will replace in $\max\left(\left|\mu_{1j}\right|,\dots,\left|\mu_{Kj}\right|\right)$, the M_j, therefore the (3.3.9) will become

$$\min\left\{\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{i,k}\frac{\left(x_{i,j}-\mu_{k,j}\right)^2}{\sigma_j^2}+\lambda w_j M_j\right\} \tag{3.3.10}$$

for each $j = 1,..,p$ where $-M_j \leq \mu_{k,j} \leq M_j, k = 1,..,K,\ M_j \geq 0$

We can show that the solution of (3.3.10) is achieved by shrinking a weighted average of different μ_{kj}^0 where

$$\mu_{kj}^0 = \frac{\sum_{i=1}^{n}\tau_{i,k}\, x_{i,j}}{\sum_{i=1}^{n}\tau_{i,k}}$$

are the k cluster means in j variable without penalty for $k = 1,..,K$, $j = 1,\dots,p$ (Wang and Zhu (2008))

For the solution of (3.3.9), Wang and Zhu propose the below theorem:

THEOREM 1: For each j minimization problem (3.3.9), if there are $k_1,..,k_r$ clusters, such as

$$\left|\hat{\mu}_{k1,j}\right| = \cdots \left|\hat{\mu}_{kr,j}\right| > \left|\hat{\mu}_{k,j}\right| , \tag{3.3.11}$$

and k doesn't belong to $\{k_1,..,k_r\}$, then

 https://doi.org/10.26219/heal.aueb.5860

$$\hat{\mu}_{kj} = \begin{cases} \mu_{kj}^0 \quad for\ k\ doesn't\ belong\ to\ \{k_1 .., k_r\} \\ sgn(\mu_{kj}^0)(\sum_{s=1}^r \frac{\tau_{.k_s}}{\sum_{s=1}^r \tau_{.k_s}} |\mu_{k_s}^0| - \frac{\lambda w_j \sigma_j^2}{\sum_{s=1}^r \tau_{.k_s}})_+ \end{cases}, k \in \{k_1 .., k_r\} \qquad (3.3.12)$$

where $\tau_{.k_s} = \sum_{i=1}^n \tau_{ik_s}$ and (.)+ is the positive part of the second branch of (3.3.12).

According to the above theorem, we conclude that if there are r cluster means which are greater than $\hat{\mu}_{kj}$, then only the corresponding μ_{kj}^0 (which are the estimated cluster means when there is no penalty, i.e. λ = 0) of the r clusters are shrunken, and these are shrunken to the same absolute value from the L_∞ penalty.

To apply the theorem, we must decide the number r of cluster means which are greater than $\hat{\mu}_{kj}$, and the set of clusters $\{k_1, .., k_r\}$, showing which μ_{kj}^0 should be shrunken toward to 0. When the number of clusters whose means are greater than the mean of k cluster is not too large, i.e. $K \leq 10$ for $1 \leq r \leq K$, then we try all the possible sets $\{k_1, .., k_r\}$. That is, in every possible set, we estimate the cluster means $\hat{\mu}_{kj}$ using (3.3.12), then check whether the assumption in (3.3.11) is satisfied, and if this assumption is satisfied, we replace the estimates in (3.3.9). For the various sets $\{k_1, .., k_r\}$, $K \leq 10$, $1 \leq r \leq K$, we will find different values for the mean $\hat{\mu}_{kj}$ in (3.3.9). We will select the set of clusters that will give the smallest mean $\hat{\mu}_{kj}$, i.e. the smallest value in (3.3.9). If the number of clusters is greater than 10, we will use (3.3.10). The above EM algorithm is repeated several times between steps E and M until it converges. We also run it with different initial values to make sure that the algorithm does not converges in a local maximum. Finally, having mentioned the positive aspects of the method at the beginning of paragraph 3.3, we will discuss some of the negatives that this method has.

This method, as we referred, for each variable is shrinking to 0 the cluster mean corresponding to the greatest absolute value of all cluster means associated with a variable j. If this mean is estimated to be 0, then all the cluster means associated with this variable will be estimated as 0 and therefore the variable as noninformative will be removed from the model. However, if the

https://doi.org/10.26219/heal.aueb.5860

greatest cluster mean isn't estimated as 0, then the variable will remain in the model. The negative is that this method cannot consider a noninformative variable to separate two clusters. So, the method will consider a variable either noninformative for all clusters or informative.

That is why Wang and Zhu have developed the method of the following paragraph

3.3.2. Hierarchical-GMM

This method was also developed by Wang and Zhu (2008). As we said in 3.3, L_∞ penalty treats the means of the clusters associated with the same variable as a group. The same is done by the hierarchical, just the hierarchical does not choose to shrink toward to 0 the greater in absolute value cluster mean, but it can shrink toward to 0 the mean of any cluster of the j variable. For L_∞ penalty, if M_j=max$\left(|\mu_{1j}|, \dots, |\mu_{Kj}|\right)$, the penalty will be λM_j and can be written as

$$\mu_{k,j} = M_j a_{k,j} \tag{3.3.13}$$

where -1≤ $a_{k,j} \leq 1$

L_∞ penalty either will make all cluster means equal to 0 (if the greater absolute mean becomes 0) or none. This led Wang and Zhu to choose to shrink toward to 0 the mean of any cluster and not the maximum M_j, so (3.3.13) according to their proposal is written

$$\mu_{k,j} = \gamma_j \theta_{k,j} \tag{3.3.14}$$

where $\gamma_j \geq 0$, k= $1, .., K, j = 1, .., p$

γ_j is the mean that is chosen to shrink toward to 0, i.e. it has a similar role with M_j in L_∞ penalty. $\theta_{k,j}$ is like $a_{k,j}$ but is not clogged by -1.1. This method treats every $\mu_{k,j}$ hierarchically. γ_j belongs to the first stage of the hierarchy by treating $\mu_{k,j}$ associated with the j variable as a group and $\theta_{k,j}$ in the second reflecting the differences between $\mu_{k,j}$. If γ_j becomes 0, then all the means $\mu_{k,j}$ associated with the j variable will become 0. If, however, γ_j doesn't shrink to 0, some of

https://doi.org/10.26219/heal.aueb.5860

the means $\mu_{k,j}$ still have the probability to be 0 if $\theta_{k,j}$ becomes 0, so the variable will be noninformative to separate some clusters, but not all clusters and so won't be removed from the model. So, the difference between the two methods is that in the second one if the selected cluster mean doesn't estimate as zero, the other cluster means have the probability of being zero. So, a variable can be noninformative for separating 2 or more clusters.

Let's develop the method in more detail.

From (3.3.1) we subtract another penalty, which consists of two terms, so the loglikelihood of the hierarchical is

$$lp(\Theta) = \log p(X|\Theta) - \lambda_\gamma \sum_{j=1}^{p} \gamma_j - \lambda_\theta \sum_{k=1}^{K} \sum_{j=1}^{p} |\theta_{kj}| \qquad (3.3.15)$$

As mentioned at the beginning of the paragraph, $\gamma_j \geq 0$ and is the mean of any cluster associated with the variable j (i.e. not the maximum mean) that will be shrunken toward to 0. If γ_j becomes 0 then all the cluster means of the variable j will become 0 and therefore the variable will be noninformative, so it will be removed from the model.

If γ_j is not 0, it may be 0 some of θ_{kj}, so from (3.3.14), some μ_{kj}, $k = 1,..,K$ have the probability to be estimated as 0. Therefore, the variable j will be noninformative to separate these clusters whose means will be 0. According to Wang and Zhu, in this case, the hierarchical penalty keeps the flexibility of L_1 penalty.

An improved version of (3.3.14) is its adaptive form. The loglikelihood of the adaptive form is as follows

$$lp(\Theta) = logp(X|\Theta) - \lambda_\gamma \sum_{j=1}^{p} w_j^\gamma \gamma_j - \lambda_\theta \sum_{k=1}^{K} \sum_{j=1}^{p} w_{kj}^\theta |\theta_{kj}| \qquad (3.3.16)$$

where $w_j^\gamma, w_{kj}^\theta$ are predetermined weights.

Because (3.3.16) cannot be solved, the EM algorithm will be used.

Let $\tau_{i,k} = 1$ if the i observation belongs to the k cluster and 0 else. If $\tau_{i,k}$ were all observable, loglikelihood would be

https://doi.org/10.26219/heal.aueb.5860

$$\sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{i,k}\{\log w_k+\log\varphi(\boldsymbol{x_i};\boldsymbol{\mu_k},\boldsymbol{\Sigma})\}-\lambda_\gamma\sum_{j=1}^{p}w_j^{\gamma}\gamma_j-\sum_{k=1}^{K}\sum_{j=1}^{p}w_{kj}^{\theta}|\theta_{kj}| \qquad (3.3.17)$$

And here the only difference from the other methods is in the estimating of the cluster means. There we will use a second theorem.

We consider initial values $\boldsymbol{\theta}^{(\mathbf{0})}=\left\{w_1^{(0)},\dots,w_K^{(0)},\boldsymbol{\mu}_{\mathbf{1}}^{(\mathbf{0})},\dots,\boldsymbol{\mu}_{\boldsymbol{K}}^{(\mathbf{0})},\boldsymbol{\Sigma}^{(\mathbf{0})}\right\}$

E-STEP

The $\tau_{i,k}$ are not observable, and we will estimate them:

$$\hat{\tau}_{i,k}{}^{(t+1)}=E\left(\tau_{i,k}\middle|X,\widehat{\boldsymbol{\Theta}}^{\boldsymbol{t}}\right)=\mathrm{P}\left(\tau_{k,j}=1\middle|X,\widehat{\boldsymbol{\Theta}}^{\boldsymbol{t}}\right)=\frac{\widehat{w}_k^{(t)}\varphi\left(\boldsymbol{x_i};\widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(\boldsymbol{t})},\widehat{\boldsymbol{\Sigma}}^{(\boldsymbol{t})}\right)}{\sum_{k=1}^{K}\widehat{w}_k^{(t)}\varphi\left(\boldsymbol{x_i};\widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(\boldsymbol{t})},\widehat{\boldsymbol{\Sigma}}^{(\boldsymbol{t})}\right)}=$$

$$=\frac{w_k^{(t)}(2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}|^{-\frac{p}{2}}\exp\left(-\frac{1}{2}\frac{\left(x_i-\widehat{\mu}_{\boldsymbol{k}}^{(t)}\right)\Sigma^{-1}\left(x_i-\widehat{\mu}_{\boldsymbol{k}}^{(t)}\right)'}{2}\right)}{\sum_{k=1}^{K}\widehat{w}_k^{(t)}(2\pi)^{-\frac{p}{2}}|\boldsymbol{\Sigma}|^{-\frac{p}{2}}\exp\left(-\frac{1}{2}\frac{\left(x_i-\widehat{\mu}_{\boldsymbol{k}}^{(t)}\right)\Sigma^{-1}\left(x_i-\widehat{\mu}_{\boldsymbol{k}}^{(t)}\right)'}{2}\right)}, \qquad (3.3.18)$$

where $t=0,1,2,..$

In this method also, we define $\Omega=\left\{\mu_{kj,}k=1,..,K;j=1,..p\right\}$.

We define the penalty function as following:

$$J(\Omega)=\lambda_\gamma\sum_{j=1}^{p}w_j^{\gamma}\gamma_j-\lambda_\theta\sum_{k=1}^{K}\sum_{j=1}^{p}w_{kj}^{\theta}|\theta_{kj}|$$

We replace the estimates of (3.3.18) in (3.3.17) and we end up in the type

$$Q=\sum_{i=1}^{n}\sum_{k=1}^{K}\hat{\tau}_{i,k}{}^{(t+1)}\{\log w_k+\log\varphi(\boldsymbol{x_i};\boldsymbol{\mu_k},\boldsymbol{\Sigma_k})\}-J(\Omega) \qquad (3.3.19)$$

M-STEP

We will maximize the (3.3.19) for finding the estimates of the parameters

https://doi.org/10.26219/heal.aueb.5860

$$\frac{dQ}{dw_k} = 0$$

Solving for w_k, we find that

$$\widehat{w}_k^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{\tau}_{i,k}{}^{(t+1)}}{n}$$

Continue with variances parameters:

$$\frac{\partial Q}{\partial \sigma_j^2} = 0$$

Solving for σ_j^2, it will be

$$(\hat{\sigma}_j^{(t+1)})^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} \hat{\tau}_{i,k}{}^{(t+1)}(x_{i,j} - \hat{\mu}_{k,j}^{(t)})^2 \quad ,1 \leq j \leq p$$

Next is to estimate the means parameters:

We will consider the set Ω where it will contain the means values of all the clusters in each variable, i.e. $\Omega = \{\mu_{kj,} k = 1,..,K; j = 1,..p\}$.

In $(t+1)$ step the means will be

$$\hat{\Omega}^{t+1} = \operatorname{argmax}\left\{-\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{j=1}^{p} \hat{\tau}_{i,k}{}^{(t+1)} \frac{\left(x_{i,j} - \hat{\mu}_{k,j}^{(t)}\right)^2}{\hat{\sigma}_j^{2(t+1)}} - J(\Omega)\right\} \qquad (3.3.20)$$

The (3.3.20) will become

$$\min\left\{\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{j=1}^{p} \hat{\tau}_{i,k}{}^{(t+1)} \frac{\left(x_{i,j} - \hat{\mu}_{k,j}^{(t)}\right)^2}{\hat{\sigma}_j^{2(t+1)}} + J(\Omega)\right\} \qquad (3.3.21)$$

That is, we changed the signs in (3.3.20), and so the max, is min.

We will estimate γ_j, θ_{kj} by applying an iterative method between the two parameters. That is, by fixing one parameter, e.g. γ_j, we estimate the other (θ_{kj}). Then we fix θ_{kj} and estimate γ_j. This iteration will take place until it converges. Because at each step the value of (3.3.21) decreases, the solution will surely converge.

THEOREM 2: Let $\mu_{kj}^0 = \frac{\sum_{i=1}^{n} \tau_{ik} x_{ij}}{\sum_{i=1}^{n} \tau_{ik}}$ and $\tau_{.k} = \sum_{i=1}^{n} \tau_{ik}$

- When θ_{kj} is fix, for $k = 1,..,K$, $j = 1,..,p$, then

https://doi.org/10.26219/heal.aueb.5860

$$\hat{\gamma}_j = I_{(\exists\, k, \theta_{kj} \neq 0)} \left(\sum_{k=1}^{K} \frac{\xi_k}{\sum_{k=1}^{K} \xi_k} \frac{\mu_{kj}^0}{\theta_{kj}} - \lambda_\gamma w_j^\gamma \frac{\sigma_j^2}{\sum_{k=1}^{K} \xi_k} \right)_+ \quad (3.3.22)$$

where $\xi_k = \tau_{.k} \theta_{kj}^2$

- When γ_j , $j = 1, .., p$ is fix

$$\hat{\theta}_{kj} = I_{(\gamma_j > 0)} sign(\mu_{kj}^0) \left(\frac{|\mu_{kj}^0|}{\gamma_j} - \lambda_\theta w_{kj}^\theta \frac{\sigma_j^2}{\gamma_j{}^2 \tau_{.k}} \right)_+ \quad (3.3.23)$$

From the above theorem, looking first the $\hat{\gamma}_j$, we conclude that if θ_{kj} is 0, then $\hat{\gamma}_j$ is 0. If not all θ_{kj} equal 0, i.e. if there are $\theta_{k_1 j}, \theta_{k_2 j}, \dots, \theta_{k_r j}$ different from 0, from (3.3.14) we will have $\gamma_j = \frac{\mu_{k_s j}}{\theta_{k_s j}}$ for each $1 \leq s \leq r$ (3.3.24). Replacing in (3.3.24) the $\mu_{k_s j}^0$ instead of $\mu_{k_s j}$, it will be $\tilde{\gamma}_j = \frac{\mu_{k_s j}^0}{\theta_{k_s j}}$ for each $1 \leq s \leq r$ (3.3.25).

So, we take r estimations for γ_j. An estimation of γ_j is a weighted average of the form $w_1 \frac{\mu_{k_1 j}^0}{\theta_{k_1 j}} + .. + \mathrm{w_r} \frac{\mu_{k_r j}^0}{\theta_{k_r j}}$. In (3.3.22) the $\sum_{k=1}^{K} \frac{\xi_k}{\sum_{k=1}^{K} \xi_k} \frac{\mu_{kj}^0}{\theta_{kj}}$ is a weighted average where the weights are proportional to ξ_k.

Looking at (3.3.23), we conclude that if γ_j=0, then θ_{kj} will be also 0. But if γ_j are different from 0, then we will have $\theta_{kj} = \frac{\mu_{kj}}{\gamma_j}$(3.3.26). Putting the μ_{kj}^0 in the position of μ_{kj}, (3.3.26) will be $\tilde{\theta}_{kj} = \frac{\mu_{kj}^0}{\gamma_j}$. The (3.3.23), decreases the $\tilde{\theta}_{kj}$ with an amount inversely proportional to γ_j^2.So, if γ_j is large, which means that the variable j is informative, then the amount which shrinks the $\tilde{\theta}_{kj}$ will be small(since it is inversely proportional to γ_j^2). Correspondingly, if γ_j is small, which means that the variable j is noninformative, then the quantity that shrinks $\tilde{\theta}_{kj}$ will be large.

3.4. Pairwise Fusion Penalty (PFP) and Adaptive Pairwise Fusion Penalty (APFP)

This method was developed by Jian Guo et al (2010) and considers that a variable is noninformative for separating two clusters, when the two clusters means are shrunken to the same value. That is, if $k,\ k'$ are two clusters, then a variable j will be noninformative for separating the two clusters if the estimated means are equal, i.e. $\hat{\mu}_{k,j=}\, \hat{\mu}_{k',j}$. Generally, a variable is removed for the model if all the cluster means associated with this variable are shrunken to the same value.

In more detail:

As mentioned above we consider that we have n observations and p variables, where $n << p$. That is, each observation is a vector of p dimensions, e.g. the $\boldsymbol{x_i}$ observation write $\boldsymbol{x_i}$= $(x_{i1},\ldots, x_{ip})$. Our data are generated from a Gaussian distribution, where as we have mentioned each Gaussian term corresponds to a cluster, that is

$$f(\boldsymbol{x_i}) = \sum_{i=1}^{K} w_k \varphi(\boldsymbol{x_i}, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k})$$

Also, the data is transformed so that each variable j for $1 \leq j \leq p$ has a mean value of 0.

We also assume that the variance-covariance matrix $\boldsymbol{\Sigma_k}$ is the same for all clusters and diagonal, i.e. the variance-covariance matrix has the form $\boldsymbol{\Sigma_k} = \boldsymbol{\Sigma}$=diag$(\sigma_1^2,\ldots.,\sigma_p^2)$ for each cluster $1 \leq k \leq K$. The assumption of $\boldsymbol{\Sigma}$ to be diagonal is common for such problems we are negotiating.

Let $\boldsymbol{\theta}$= $\{w_1, \ldots w_K, \boldsymbol{\mu_1}, \ldots, \boldsymbol{\mu_K}, \sigma_1^2, \ldots \sigma_p^2\}$. The log likelihood of the PFP method is

$$\log p(X|\boldsymbol{\theta}) = \sum_{i=1}^{n} \log\left\{\sum_{k=1}^{K} w_k \varphi(\boldsymbol{x_i}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma})\right\} - \lambda \sum_{j=1}^{p} \sum_{1 \leq k < k' \leq K} \left|\mu_{k,j} - \mu_{k',j}\right| \quad (3.4.1)$$

The term $\sum_{j=1}^{p} \sum_{1 \leq k < k' \leq K} |\mu_{k,j} - \mu_{k',j}|$ is referred as PFP (Pairwise Fusion Penalty by Jian Guo et al (2010)) and λ is the shrinkage parameter in which we select for it different values

 https://doi.org/10.26219/heal.aueb.5860

As we have said, the purpose of the PFP penalty is to reduce to 0 the difference between the cluster means k, k'. From the (3.4.1) we observe that if the means of the clusters k, k' have the same value, the PFP penalty becomes 0 and therefore the variable j is noninformative for the separation of the clusters k, k'. However, a variable that is noninformative to separate the clusters k, k', may be informative for separating other clusters.

An improved version of the PFP method is the Adaptive PFP Method (APFP) (Jian Guo et al, 2010)

The type of APFP

$$\log p(X|\boldsymbol{\theta}) = \sum_{i=1}^{n} \log\left\{\sum_{k=1}^{K} w_k \varphi(\boldsymbol{x_i}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma})\right\} - \lambda \sum_{j}^{p} \sum_{1 \le k < k' \le K} \tau_{k,k'}^{(j)} \left|\mu_{k,j} - \mu_{k',j}\right| \quad (3.4.2)$$

where $\tau_{k,k'}^{(j)}$ are predetermined weights. The idea of the above method is that if the j variable is informative to separate the k, k' clusters then the corresponding $\tau_{k,k'}^{(j)}$ is small, whereas if the j variable is noninformative for the separation of clusters k, k', then the corresponding $\tau_{k,k'}^{(j)}$ is large. That is, in noninformative variables we give a large penalty for the difference between $\mu_{k,j}, \mu_{k',j}$, while in the informative the penalty is small for the difference between the means of the clusters k, k'. The type for predefined weights is the following:

$$\tau_{k,k'}^{(j)} = |\tilde{\mu}_{k,j} - \tilde{\mu}_{k',j}|^{-1}$$

where $\tilde{\mu}_{k,j}$ is the estimate of the mean $\mu_{k,j}$ when there is no penalty, i.e. for $\lambda = 0$.

The parameters to be selected by the researcher are two, the number of clusters K and the parameter λ. The model selection is as follows: we choose numbers for the parameters K, λ and run models with all the combinations between the parameters K, λ. From all models we choose the one with the lowest BIC, so the corresponding K, λ. The type of the BIC is the following

$$BIC(K, \lambda) = -2 \sum_{i=1}^{n} \log\left\{\sum_{k=1}^{K} \widehat{w}_k \varphi\left(\boldsymbol{x_i}; \boldsymbol{\widehat{\mu}_k}, \boldsymbol{\widehat{\Sigma}}\right)\right\} + d \log n \quad (3.4.3)$$

https://doi.org/10.26219/heal.aueb.5860

where $\{\widehat{w}_k, \widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}, \widehat{\boldsymbol{\Sigma}}\}_{k=1}^{K}$ are the parameters to be estimated by K, λ. Parameter d is degrees of freedom and is defined as the nonzero estimates. The type is $d = K + p - 1 + e(\hat{\mu})$, where e $(\hat{\mu})$ is the number of $\{\hat{\mu}_{k,j}\}$ that are different from zero. Because the maximization of (3.4.2) is difficult to be done and therefore to be solved, we will use the EM algorithm. We consider an index, let us call it $Z_{i,k}$ that is 1 if the x_i observation belongs to the k cluster, and 0 otherwise. Loglikelihood will be written as follows:

$$\log p(X|\boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{k=1}^{K} Z_{i,k}\{\log w_k + \log\varphi(\boldsymbol{x_i}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma})\}$$

$$- \lambda \sum_{j}^{p} \sum_{1\le k<k'\le K} \tau_{k,k'}^{(j)} \left|\mu_{k,j} - \mu_{k',j}\right| \qquad (3.4.4)$$

For the above log likelihood, we will use the EM algorithm as we did in previous methods. The E Step will be the same as the previous method The only difference will be in the M Step in which we will estimate with different way the cluster means $\mu_{k,j}$. We consider the initial values

$\boldsymbol{\theta}^{(\mathbf{0})} = \left\{w_1^{(0)}, \dots, w_K^{(0)}, \boldsymbol{\mu}_{\mathbf{1}}^{(\mathbf{0})}, \dots, \boldsymbol{\mu}_{\boldsymbol{K}}^{(\mathbf{0})}, \boldsymbol{\Sigma}^{(\mathbf{0})}\right\}$, i.e. for t = 0

E-STEP

As with the GMM, the estimate of missing $Z_{i,k}$ is as follows below: For the (t + 1) step,

$$\hat{Z}_{i,k}^{(t+1)} = \frac{\widehat{w}_k^{(t)} \varphi(\boldsymbol{x_i}; \widehat{\boldsymbol{\mu}}_{\boldsymbol{k}}^{(\boldsymbol{t})}, \widehat{\boldsymbol{\Sigma}}^{(\boldsymbol{t})})}{\sum_{k'=1}^{K} \widehat{w}_{k'}^{(t)} \varphi(\boldsymbol{x_i}; \widehat{\boldsymbol{\mu}}_{\boldsymbol{k'}}^{(\boldsymbol{t})}, \widehat{\boldsymbol{\Sigma}}^{(\boldsymbol{t})})} =$$

$$= \frac{\widehat{w}_k^{(t)} (2\pi)^{-p/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}\left(x_i - \widehat{\mu}_{\boldsymbol{k}}^{(\boldsymbol{t})}\right)\boldsymbol{\Sigma}^{-\mathbf{1}}\left(x_i - \widehat{\mu}_{\boldsymbol{k}}^{(\boldsymbol{t})}\right)'\right)}{\sum_{k'=1}^{K} \widehat{w}_{k'}^{(t)} (2\pi)^{-p/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}\left(x_i - \widehat{\mu}_{\boldsymbol{k'}}^{(\boldsymbol{t})}\right)\boldsymbol{\Sigma}^{-\mathbf{1}}\left(x_i - \widehat{\mu}_{\boldsymbol{k'}}^{(\boldsymbol{t})}\right)'\right)}, \quad t=0,1,2,\dots..$$

We note that for t = 0, $\hat{Z}_{i,k}^{(1)}$ are estimated from the initial values

$w_1^{(0)}, \dots, w_K^{(0)}, \boldsymbol{\mu}_1^{(\mathbf{0})}, \dots, \boldsymbol{\mu}_K^{(\mathbf{0})}, \boldsymbol{\Sigma}^{(\mathbf{0})}$. So, by replacing the estimates of missing in (3.4.4), in the $(t+1)$ step this will be done as following

$$\log p(X|\boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{k=1}^{K} \hat{Z}_{i,k}^{(t+1)}\{\log w_k + \log\varphi(\boldsymbol{x_i}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma})\} - \lambda \sum_{j}^{p} \sum_{1\le k<k'\le K} \tau_{k,k'}^{(j)} \left|\mu_{k,j} - \mu_{k',j}\right| \tag{3.4.5}$$

M-STEP

As in GMM, we will maximize (3.4.5) to find the estimators of the parameters we are looking for by derivate the parameter we want to estimate and equalizing the derivatives with 0, i.e.

$$\frac{dQ}{dw_k} = 0$$

Solving for w_k we find that

$$\widehat{w}_k^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{Z}_{i,k}{}^{(t+1)}}{n}$$

We continue with the covariances.

$$\frac{\partial Q}{\partial \sigma_j^2} = 0$$

Solving for σ_j^2 we find that

$$(\hat{\sigma}_j^{(t+1)})^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} \hat{Z}_{i,k}{}^{(t+1)}(x_{i,j} - \hat{\mu}_{k,j}^{(t)})^2 \ \ , 1 \le j \le p$$

We continue with the estimate of the means.

Initially, the mean value in $(t+1)$ repeat is written as follows

https://doi.org/10.26219/heal.aueb.5860

$$\hat{\mu}^{(t+1)} = \text{argmin} \left\{ \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{K} \hat{Z}_{i,k}{}^{(t+1)} \sum_{j=1}^{p} \frac{(x_{i,j} - \mu_{k,j})^2}{(\hat{\sigma}_j^{(t)})^2} \right.$$
$$\left. + \lambda \sum_{j=1}^{p} \sum_{1 \le k \le k' \le K} \tau_{k,k'}^{(j)} \left| \mu_{k,j} - \mu_{k',j} \right| \right\} \quad (3.4.6)$$

As we can see, the above formula is an equivalent expression of Q so that the problem is a problem of minimization. The difference between the means of clusters k, k' will be approximated using a local quadratic approximation (its converge properties have been studied by Fan and Li (2001) and Hunter and Li (2005)) with the following formula:

$$\left| \mu_{k,j}^{(s+1)} - \mu_{k',j}^{(s+1)} \right| \approx \frac{\left(\mu_{k,j}^{(s+1)} - \mu_{k,j}^{(s+1)} \right)^2}{2 \left| \hat{\mu}_{k,j}^{(s)} - \hat{\mu}_{k',j}^{(s)} \right|} + \frac{1}{2} \left| \hat{\mu}_{k,j}^{(s)} - \hat{\mu}_{k',j}^{(s)} \right| \quad (3.4.7)$$

Note that we shouldn't confuse the t of the algorithm EM with the s of the local square approximation, as one denotes the repetition in the EM algorithm and the other the iteration in the square local approximation which is step in the M step of the EM algorithm. The $\hat{\mu}_{k,j}^{(s)}$) is the estimate of the mean of k cluster in the j variable in the previous iteration.

For each variable j, we can replace in the penalty in (3.4.6) (which is the difference of means between the k, k' clusters) the estimates of the means of the cluster k, k' in the j variable as in (3.4.7), so the (3.4.6) will be done for each j,

$$\min_{\mu_{(j)}^{(s+1)}} \left\{ \frac{1}{2 \left(\hat{\sigma}_j^{(t)} \right)^2} \sum_{i=1}^{n} \sum_{k=1}^{K} \hat{Z}_{i,k}^{(t+1)} \left(x_{i,j} - \mu_{k,j}^{(s+!)} \right)^2 \right.$$
$$\left. + \lambda \sum_{1 \le k < k\prime \le K} \tau_{k,k\prime}^{(j)} \frac{\left(\mu_{k,j}^{(s+1)} - \mu_{k,j}^{(s+1)} \right)^2}{2 \left| \hat{\mu}_{k,j}^{(s)} - \hat{\mu}_{k\prime,j}^{(s)} \right|} \right\}$$

One criterion for stopping the local quadratic approximation is that the difference between the means of the two clusters is less than or equal to 10^{-10}, i.e.,

https://doi.org/10.26219/heal.aueb.5860

$$\left|\mu_{k,j}^{(s+1)} - \mu_{k',j}^{(s+1)}\right| \leq 10^{-10}$$

The differences in which their absolute value is 10^{-10}, we set them equal to 0. The above steps (E and M) are repeated several times until the algorithm converges. Finally, looking at the final $\hat{Z}_{i,k}$we will group each observation in the cluster with the highest probability according to $\hat{Z}_{i,k}$.

3.5. Summary and Comparison between Methods

Guo, Levina, Michailidis, Zhu (2010) compare the adaptive form of PFP (APFP) with the following adaptive forms of L_1, L_∞

- For L_1

$$\sum_{i=1}^{n} \log\left\{\sum_{k=1}^{K} w_k \varphi(\boldsymbol{x_i}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma})\right\} - \lambda \sum_{j=1}^{p}\sum_{k=1}^{K} \tau_{k,j}^{l_1} \left|\mu_{k,j}\right|$$

where $\tau_{k,j}^{l_1}$ are the prespecified weights and defined as $\tau_{k,j}^{l_1} = \frac{1}{|\tilde{\mu}_{k,j}|}$ for each $1 \leq k \leq K$ and for each $1 \leq j \leq p$. The $\tilde{\mu}_{k,j}$ are the estimated cluster means of GMM, that is the estimates of cluster means when in model-based clustering there isn't penalty. An important step to be applied to the adaptive form of L_1 (Adaptive L_1 ή AL1) is the data to be centered so that the mean values of the variables are 0. For the APFP method, this step is not needed. In Chapter 4, in which compares some methods, we use AL1.

- For L_∞

$$\sum_{i=1}^{n} \log\left\{\sum_{k=1}^{K} w_k \varphi(\boldsymbol{x_i}; \boldsymbol{\mu_k}, \boldsymbol{\Sigma})\right\} - \lambda \sum_{j=1}^{p} \tau_j^{l_\infty} \max\left(\left|\mu_{1,j}\right|, \dots, \left|\mu_{k,j}\right|, \dots., \left|\mu_{K,j}\right|\right)$$

where $\tau_{k,j}^{l_\infty}$ are the predetermined weights and defined as $\tau_{k,j}^{l_\infty} = \frac{1}{\max(|\tilde{\mu}_{1,j}|,\dots,|\tilde{\mu}_{k,j}|,\dots,|\tilde{\mu}_{K,j}|)}$ for each $1 \leq k \leq K$ and for each $1 \leq j \leq p$. The $\tilde{\mu}_{k,j}$ are the estimates of cluster means in GMM , i.e. the estimates of cluster means in model-based clustering when there isn't penalty.

https://doi.org/10.26219/heal.aueb.5860

The methods mentioned above present as expected disadvantages and advantages. Some methods are better under certain conditions of data and variables.

We summarize the description of the methods mentioned above and compare them. We assume that the data has been transformed so that all variables have a mean value of 0 in all methods.

As mentioned in 3.2, the L_1 method for a variable shrinks toward to the mean value of the variable (i.e., 0) the means of the clusters. The variable is noninformative for separating two clusters k, k' if the estimated means of these two clusters are equal to 0. If all the estimated cluster means corresponding to the variable are equal to 0, then the variable is removed from the model.

In 3.3 we mentioned the L_∞ and the hierarchical method. L_∞ for a variable j shrinks to 0 the cluster means having the greatest absolute value. If these are shrunken to 0, then the means of the remaining clusters will be 0, so the variable as noninformative will leave the model. The hierarchical does not shrink towards to 0 the mean of the cluster with the greatest absolute value but the mean of any cluster. If this mean is estimated as 0, then the means of the remaining clusters will be 0, so the variable as noninformative will be removed from the model. If, however, it does not become 0, then some means continue to have the probability of being 0, so the variable may be noninformative for separating two clusters.

Finally, in 3.4 we developed the APFP method, according to which if we have two clusters k, k', the method will shrink the mean of the k cluster toward to the mean of the k' cluster. If the two means are shrunken to the same value, then the variable will be considered as noninformative to separate the two clusters. If the means of all clusters associated with the variable are shrunken to the same value, the variable as noninformative will leave the model.

We continue with the comparisons between the methods:

We assume that the variables have a mean value of 0. Initially, the L_1 method does not recognize as noninformative for separating two clusters k, k', variables in which the means are the same but different from 0. In contrast, the

 https://doi.org/10.26219/heal.aueb.5860

APFP method recognizes as noninformative to separate two clusters variables where the means of the clusters are the same, whether it is equal to 0 or not. Thus, the APFP method is more efficient than L_1 in some cases. These cases are when a subset of clusters corresponding to a variable have the same means that are different from the mean value of the variable (that is 0). Then, APFP will consider the variable noninformative for separating these clusters, while L_1 informative because means differ from 0. A negative of L_1 according to Wang and Zhu (2008) is also that doesn't treat the cluster means associated with a variable as a 'group'. For this reason, the L_∞ method is developed which either will do all cluster means equal to 0 or will not make any equal to 0. But the L_∞ method treats the variables with an 'all-in-all-out' manner. It is not possible with this method to be identified variables that will be noninformative to separate two clusters. Conversely, the hierarchical allows the means of different clusters to be 0, either the cluster mean we chose is shrunken to 0 or not. Thus, it allows the identification of noninformative variables to separate two clusters. According to this, by applying the hierarchical, we can get more clustering information than L_∞ for each variable. Also, in relation to the previous two methods, L_∞ and hierarchical treat clusters belonging to the same variable as a group. According to Wang and Zhu (2008), this has the effect of removing noninformative variables more easily.

In the next Chapter we will compare the methods APFP, AL1, GMM with all the informative variables into the model and GMM without the noninformative variables into the model with numerical results.

https://doi.org/10.26219/heal.aueb.5860

 https://doi.org/10.26219/heal.aueb.5860

CHAPTER 4
APPLICATIONS FOR COMPARISON THE METHODS GMM, AL1, APFP

4.1. Introduction

In this section, we will explore the methodologies and their performance from the practical point of view using five simulations and a real dataset (the Golub dataset). A comparison study is conducted, and the methods included are: GMM (Gaussian mixture model without penalty), L_1 (AL1), APFP (adaptive pairwise fusion penalty) and GMM without noise. The last method is the known GMM without penalty, but we will have removed the noninformative variables, so the clustering will be done with only the informative variables. This will not be done in the Golub dataset. The using of GMM without noise is because the GMM method does not make automatic selection of variables, so if we do not remove them, we will not be able to look at how effective the method is and how correct is the clustering having only the informative variables. In this way we will examine how helpful to clustering is the removing of noninformative variables in high dimensional data sets. The 5 simulation specifications are included in the paper of Jian Guo et al (2010). We mention that for AL1, APFP and GMM were used the software which is in the package PARSE. The command for APFP is the apfp, this which corresponding to the AL1 method is apL1 and finally, the command for GMM is nopenalty as in GMM the shrinkage parameter λ equal to 0, so there is no penalty.

In simulations 1,2,4 and 5 the number of observations is the same in all clusters, while in simulation 3, the number of observations is different among the clusters. That is, cluster 1 and 2 have the same number of observations which is different from clusters 3 and 4. The number of observations in clusters 3 and 4 is much larger than the number of observations in clusters 1 and 2.
We handle the simulations with two ways. Initially through the BIC criterion, we choose the number of K-clusters and the λ parameter for each simulation. With the second approach, we consider that the number of clusters K is known, so the only parameter that is chosen from BIC criterion is the λ parameter. This

https://doi.org/10.26219/heal.aueb.5860

is done to examine how effective the methods are when we do not know the number of clusters and this is compared to the known K methods. We generate 50 random data sets for each simulation scheme, where each data set consists of data according to the means of the clusters in each variable. Note that the GMM method does not have a λ parameter, so in this method we initially run the command where K is unknown, and then we run the command by knowing the number of K clusters. This is done in both GMM and GMM without noise. In addition, because each iteration has different initial values, it is necessary to run each method for each data set several times to make sure that the EM algorithm does not converge to a local maximum. For this reason, for L_1 (AL1) and APFP we run 20 times each of 50 data set and keep the result of repetition where the smallest BIC is recorded. This is because it has been observed that methods are more effective in iterations where the BIC criterion is small relative to the other iterations.

In Appendices there is the algorithm from the first simulation and the explanation of each step.

Finally, we mention that in applications, in the commands we use for the L_1 (AL1), APFP and GMM methods, we give the selection of the number of clusters as well as λ (in GMM we only choose K). For example, the options given in the simulations 1,2 and 3 are the following:

- For the number of clusters, the options are K = 2,3,4
- For the shrinkage parameter λ the options are 2,5,10,15,30

4.2. Simulation 1

We have 4 clusters and 220 variables, where the first 20 are informative i.e. they are independent and follow the distribution $\mathrm{N}(\mu_{k,j}, 1)$ for the k cluster, where $\mu_{k,j}$ is the mean of k cluster corresponding to the j variable. More specifically for the variables 1-10 the means for the clusters 1, 2, 3 and 4 are 2.5, 0, 0, -2.5 and for the variables 11-20 the means are 1.5, 1.5, -1.5, -1.5 respectively. The remaining 200 variables are noninformative, i.e. they are independent and follow the $\mathrm{N}(0, 1)$ distribution for all clusters. Each cluster consist of 20 observations. So, the clustering is as follows:

- Observations 1-20 belong to the same cluster
- Observations 21-40 correspond to a second cluster
- Observations 41-60 are grouped into a third cluster
- Finally, observations 61-80 belong to a fourth cluster

Table 4.2.1 lists the cluster means of the first simulation for the informative variables:

variables	***Cluster 1***	***Cluster 2***	***Cluster 3***	***Cluster 4***
1-10	2.5	0	0	-2.5
11-20	1.5	1.5	-1.5	-1.5

Table 4.2.1: Cluster means in simulation 1

The graph in Figure 4.2.1 presents the clusters in simulation 1. Specifically, it is shown how the clusters are separated by two informative variables where one informative variable is chosen between 1-10 (in the graph is the 1st variable) while the other informative is chosen from 11-20 (in the graph is 11). What seems to be is that variables 1-10 are noninformative to separate clusters 2,3, while variables 11-20 are noninformative to separate clusters 1 and 2 as well as 3,4.

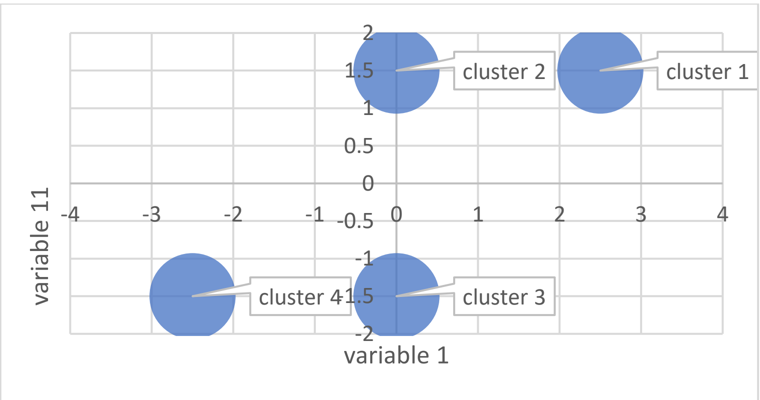

Figure 4.2.1: Clusters for simulation 1

 https://doi.org/10.26219/heal.aueb.5860

4.2.1. Implementation of AL1 method

We start with the AL1 method. As mentioned, we generate 50 random datasets for this simulation scheme. Each data set is repeated 20 times for converges reasons of EM algorithm and use the iteration where the lowest value was found in the BIC criterion. The first thing we do for the AL1 method is to transform the data so that the mean value of each variable is 0.

First, we will report how effective the method is in identifying informative variables and in removal of noninformative variables. Regarding the identification of informative, the method in all the 50 data sets recognizes correctly the 20 variables as informative, i.e. these variables always keep them in the model. Of the noninformative, it incorrectly recognizes as informative an average of 1.4 in 200 variables with a standard deviation of 2 variables. Thus, the methodology has performed very well in this first task and on average, it recognizes 9.7% of the variables (instead of $\frac{20}{220}\% = 9.09\%$ which is the percentage of informative variables) with a standard deviation of 0.6%.

We continue with clustering. The method in all the 50 data sets selects the correct number of clusters that are four. Also, the only error that has been made is in the 44th dataset where the observation 3 is grouped with the cluster of observations 21-40 (Table 4.1.3) This means that the mean misclustering rate in all data sets is almost 0. More specifically, the mean error rate is 0.025% with a standard deviation of 0.17%. We note that not to all repetitions of the implementation the correct solution with respect to number of clusters was obtained. The correct number of clusters has been recorded only in iterations where a small BIC criterion was noted. So, for each data set, holding only the results of the iteration where the lowest value was recorded in the BIC criterion, we conclude that the clustering applied by this method is correct.

The mean λ selected from the 50 data sets (where in each dataset we kept the result of iteration with the minimum BIC) is 5 with a standard deviation of 0 since in all 50 datasets the value of 5 for λ is chosen.

The average BIC of the 50 data sets is 50951.68 with a standard deviation of 208,136.

As we have mentioned, the method does not recognize as noninformative to separate between 2 clusters, variables where the clusters haven't means equal

 https://doi.org/10.26219/heal.aueb.5860

to the mean value of the variable that is 0. So, we expect variables 1-10 to be noninformative to separate clusters 2 and 3, as these clusters have means equal to 0, as 0 is the mean value of variables 1-10. Having taken into account all 50 data sets, variables 1-10 are indeed noninformative according to the method AL1 to separate clusters 2 and 3 with an average of 99.8% (instead of 100%) and a standard deviation of 1.4%. On the contrary, we expect that variables 11-20 will be informative to separate clusters 1 and 2 as well as clusters 3 and 4. We expect this because while the means of clusters 1 and 2 as well as 3 and 4 are equal (they are 1.5, 1.5, -1.5, -1.5 respectively), they are not equal to the mean value of variables 11-20 (which is 0). Indeed, the method is never able to shrink (or to estimate) the means of the clusters with equal means so that they are equal to 0. If the method was able to estimate the means of the clusters so as to be 0, the informative variables would be removed from the model, which would not help in proper clustering.

In Table 4.1.6 a small sample of the cluster means is recorded for some of the variables 1-10 and 11-20.

4.2.2. Implementation of APFP method

The next method is the APFP method. As in AL1, we generate 50 random data sets where each data set consists of data according to the means of the clusters in each variable and having a variance of 1. As before, for each data set the implementation of the method is repeated 20 times and we only keep the result of the repetition where the lowest BIC criterion was noted.

Regarding the identification of informative variables, the APFP method also recognizes in each data set the 20 informative variables. Of the noninformative it chose an average of 2.72 variables with a standard deviation of 5.35 variables. According to the above, the average percentage of the informative variables selected is 100% (i.e., the method chooses all the informative variables in all 50 data sets), while the average percentage of noninformative variables selected for the model is 1.3% with a standard deviation of 2.6%. Overall, this method selects an average of 10.3% of the variables (instead of 9.09% variables which is the real percentage of the informative variables). The mean λ is 4.49 with a standard deviation of 2.15. BIC is 51846.97 with a standard deviation of 208.9631. Regarding whether the method performs a proper clustering or not, in all 50 data sets it selects the correct number of clusters that are four, while no observation is grouped in a wrong cluster. Therefore, the average error rate is 0. We also note here as in the previous method for each data set, the correct clustering was achieved only in iterations where the BIC criterion is small compared to other iterations.

As has been noted, the APFP method considers noninformative the variables for separating between two clusters if the means of the clusters are equal, irrespective of whether they are equal to the mean value of the variable or not. So, we expect that variables 1-10 to be noninformative for separating the clusters 2 and 3 as these two clusters have equal means and variables 11-20 are noninformative for separating clusters 1 and 2 as well as for separating clusters 3 and 4. Indeed, variables 1-10 taking into account all 50 data sets are noninformative on average 88.2% with a standard deviation of 14.2% for separating the cluster 2 and 3. That is on average 8.2/10 variables out of the variables 1-10 are noninformative to separate the clusters 2 and 3. Variables 11-20 are noninformative to separate clusters 1 and 2 by 94% and a standard deviation of 8.3%, while for clusters 3 and 4 they are noninformative with a percentage of 94.2% and a standard deviation of 9%. In Table 4.1.7 a small sample of the cluster means is recorded for some of the variables 1-10 and 11-20.

The conclusions that we developed in 4.1.1 and 4.1.2 subparagraphs for the present experiment are performed in Table 4.2.2. We note that in Table 4.2.2, the values in parenthesis are the standard deviations in each mean value.

https://doi.org/10.26219/heal.aueb.5860

Method	***Variables***	***Pair of clusters***	***Percentage of variables considered noninformative to separate clusters (%)***
AL1	Variables 1-10	2/3	99.8 (1.4)
	Variables 11-20	1/2	0 (0)
		3/4	0 (0)
APFP	Variables 1-10	2/3	88.2 (14.2)
	Variables 11-20	1/2	94 (8.3)
		3/4	94.2 (9)

Table 4.2.2: Pairwise variable selection in simulation 1

4.2.3. Implementation of GMM method

The following is the GMM method, with the 220 variables. As mentioned above, the GMM method does not automatically select variables, so therefore all variables will be included in the model.

For GMM we will use the nopenalty option available from the PARSE package. For each dataset it is not necessary to repeat many repetitions as applied to the previous methods as the nopenalty command automatically finds the best BIC by applying 100 reps. Also, the only derivation from this method is the number of clusters and not λ as the GMM has no penalty

($\lambda = 0$). The number of clusters is selected using the BIC criterion. The number of clusters selected by GMM is 2 in all data sets rather than 4. The first cluster consists of observations 1-40 and the second consists of observations 41-80. Accordingly, the misclustering error rate is 50%. The reason why GMM chose a wrong number of clusters is due to the fact that all the noninformative variables are included in the model, and according these variables all 80 observations for one homogeneous cluster. So, the performance of GMM with

 https://doi.org/10.26219/heal.aueb.5860

all noninformative variables in the model is degraded in comparison with the performance of the previous two methods.

Finally, we examine the effectiveness of GMM having removed all noninformative variables. In this case, GMM selects 4 clusters in 50 data sets. In 2 datasets, specifically in datasets 47 and 44, the method groups incorrectly two observations. More specifically, in dataset 47, observation 62 belonging to the cluster of observations 61-80, was grouped with observations 41-60. In dataset 44, observation 3 instead of being grouped with the observation group 1-20, was grouped in the cluster of observations 21-40 (the same wrong allocation was done by applying the AL1 method). These misclustering errors perform to Tables 4.12 and 4.1.3. Thus, the average misclustering error rate in this case taking into account all 50 data sets is an amount of 5×10^{-2}%, i.e. almost 0 with a standard deviation of 0.2%.

What we have just examined is how important is a correct selection of variables in high dimensional data sets. The clustering of observations facilitated and improved when the meaningful variables are included in the model. Choosing multiple variables that do not add any information in the clustering, it leads to an incorrect estimation of the number of clusters as well as to the wrong clustering in the data. GMM without noninformative variables can be compared with the penalized methods which select automatic the variables. But in the high dimensions real data sets, we do not know which variables do not contribute to the correct clustering of the data, concluding that GMM without noise cannot be applied as a standalone technique.

We select to perform some results for 2 of 50 data sets, regarding to performance of the method in localization of the clusters in the observations. In the Tables 4.2.3 and 4.2.4, perform the results of clustering of 47 and 44 dataset respectively for GMM method without noise. As we have mentioned, the GMM without noise incorrectly group the observation 62 in cluster 3 with the observations 41-60 in dataset 47. Also, in dataset 44 incorrectly grouped the observation 3 in cluster 2 with the observations 21-40 and the observation 62 with the observations 41-60. The same errors in dataset 44 are also from the AL1 method.

https://doi.org/10.26219/heal.aueb.5860

Data set 47	*1st cluster*	*2nd cluster*	*3rd cluster*	*4th cluster*
1st cluster according to method	1-20			
2nd cluster according to method		21-40		
3rd cluster according to method			41-60	62
4th cluster according to method				61,63-80

Table 4.2.3: The clustering performed in data set 47 in the GMM without noise method. The first line is the simulated clusters. In the first column are the clusters found according to the method

Data set 44	*1st cluster*	*2nd cluster*	*3rd cluster*	*4th cluster*
1st cluster according to the method	1,2,4-20			
2nd cluster according to the method	3	21-40		
3rd cluster according to the method			41-60	62
4th cluster according to the method				61,63-80

Table 4.2.4: Clustering performed in repeat 44 in the GMM without noise method as in AL1. The first line is the regular clusters. In the first column are the clusters found according to the method

Table 4.2.5 has the summary of the results mentioned in the above subparagraphs. In column K, we note for each method the average number of clusters that the methods select which was calculated from the 50 datasets. The ERR (%) column shows the average misclustering error rate of the observations. The ERR (%) K known mention the average misclustering error rate when the number of cluster is known. The INFORMATIVE column indicates the average percentage of informative variables inserted in the model

https://doi.org/10.26219/heal.aueb.5860

in each method, calculated from the 50 data sets. The NONINFORMATIVE column calculates the average percentage of noninformative variables incorrectly inserted in the model. Column λ mention the average λ. The latter two methods have no value at λ as GMM has no penalty. The last column shows the average BIC. In parenthesis there are the standard deviation of each mean value

METHOD	***K***	***ERR (%)***	***ERR (%) K known***	***INFORMATIVE (%)***	***NONINFORMATIVE (%)***	***λ***	***BIC***
AL1	4(0)	0.025 (0.17)	0(0)	100	0.7(1)	5(0)	50951.68 (208.36)
APFP	4 (0)	0(0)	0(0)	100	1.3(2.6)	4.48 (2.15)	51846.97 (208.96)
GMM	2 (0)	50(0)	$5*10^{-2}$ (0.2)	100	100	-	53056 (207.4)
GMM without noise	4 (0)	$5*10^{-2}$ (0.2)	$5*10^{-2}$ (0.2)	100	0	-	5126.422 (56.4)

Table 4.2.5: Summary of the results mentioned in the simulation 1

In the Table 4.2.6 and in the Table 4.2.7, there is a selected sample of estimated means for the AL1 method and APFP method respectively for the variables 1-5 and 11-15. Note that for the variables 6-10 and 16-20, the results are similar with these of variables 1-5 and 11-15 respectively. As we see, in AL1 method the estimated cluster means are different in clusters with the same mean where the cluster means are different from 0. For example, for variables 11-15, the cluster means are 1.5, 1.5, -1.5, -1.5. The estimated means are different between the clusters. So, the variables are informative for the separation of clusters 1,2 and 3,4. For variables 1-5, the estimated means for the clusters 2 and 3 (that is 0) are the same and so the variables 1-15 are noninformative for separating the clusters 2,3. It happens because the mean value of the variables 1-5 is 0 and the AL1 considers as noninformative only the variables that the clusters belong to them have means equal to 0 (that is the mean value of the variables). On the other side, in the APFP method, the

 https://doi.org/10.26219/heal.aueb.5860

estimated cluster means are the same for the clusters with the same means. Also, the 0 has no special meaning in the APFP method as in the AL1 method. For example, for the variables 11-15, the cluster means are 1.5, 1.5, -1.5, -1.5. The estimated cluster means are the same in clusters 1,2 and 3,4 as it is performed in the Table 4.2.7. So, the variables 11-15 are noninformative for separating the clusters 1,2 and 3,4. This is the main different between the 2 methods.

Variable / Cluster	*1*	*2*	*3*	*4*	*5*	*11*	*12*	*13*	*14*	*15*
Cluster 1	2.28689	2.39478	2.76163	2.78576	3.11789	1.413453	0.84896	1.23675	1.26264	1.39758
Cluster 2	0	0	0	0	0	1.41894	1.32967	1.51986	1.21475	1.04856
Cluster 3	0	0	0	0	0	-1.06374	-1.83234	-1.68349	-1.24768	-1.59679
Cluster 4	-2.56673	-2.26459	-2.41387	-2.44783	-2.12467	-1.68784	-1.46456	-1.57982	-1.18916	-1.56016

Table 4.2.6: Estimated means for AL1 for variables 1-5 and 11-15

Variable / Cluster	*1*	*2*	*3*	*4*	*5*	*11*	*12*	*13*	*14*	*15*
Cluster 1	2.16986	2.27367	2.25284	2.62647	3.65748	1.43657	1.09781	1.38234	1.23567	1.206812
Cluster 2	-0.12648	-0.08522	-0.02731	-0.05003	0.39501	1.43657	1.09781	1.38234	1.23567	1.206812
Cluster 3	-0.12648	-0.08522	-0.02731	-0.05003	0.39501	-1.35234	-1.64345	-1.62456	-1.21890	-1.54526
Cluster 4	-2.42126	-2.12958	-1.96648	-2.25345	-2.02789	-1.35234	-1.64345	-1.62456	-1.21890	-1.54526

Table 4.2.7 Estimated means for APFP for variables 1-5 and 11-15

https://doi.org/10.26219/heal.aueb.5860

The conclusions from the above are that APFP, APL1 methods effectively remove a large percentage of noninformative variables. Clustering on these methods is correct in all 50 datasets. On the other hand, the GMM method when all variables are kept in the model was never able to find the correct number of clusters but once all the noninformative variables are removed the identification of the clusters was revealed. Also, having removed all the noninformative variables, GMM is able to apply a proper clustering. So, at a first stage it shows how useful it is to apply methods of automatic selection of variables, as having many noninformative variables within the model (and especially much more than informative) leads to unsuccessful clustering.

As we have previously pointed out, we compared the results we found above with results where the number of clusters is known. We showed that the GMM method groups almost correctly the observations when the K is known.

The second issue we will examine, is how effective are the methods when the variance isn't 1 which is the below. The next paragraph and the simulation experiment 2 will allow us to study this issue.

4.3. Simulation 2

In this simulation we will examine the efficiency of the methods when the variance is different among the informative variables. The existing clusters will be the same as those in simulation 1, i.e. each cluster will have 20 observations. Also, informative variables will continue to be the first 20. More specifically, the informative variables are i.i.d., following the N $(\mu_{k,j,}4)$ where $\mu_{k,j,}$ is the mean of the k cluster in the j variable, $1 \leq j \leq 20$. The 200 noninformative variables follow N(0,1). The following table summarizes the means of the clusters for the informative variables.

Variables	***Cluster 1***	***Cluster 2***	***Cluster 3***	***Cluster 4***
1-10	2.5	0	0	-2.5
11-20	1.5	1.5	-1.5	-1.5

Table 4.3.1: The cluster means in simulation 2.

https://doi.org/10.26219/heal.aueb.5860

Analytically, in this simulation the clustering is as follows:

• The observations 1-20 form cluster

• Observations 21-40 form a second cluster

• Observations 41-60 form a third cluster

• Observations 61-80 form a fourth cluster

4.3.1. Implementation of AL1 method

We start with the AL1 method. Firstly, we centered the data so as to each variable has mean value equal to 0. The average number of variables chosen is 21.48 variables with a standard deviation of 2.4179. The average rate of variables that AL1 chooses is 9.76% (instead of 9.09% that is the percentage of 20/220 variables) with a standard deviation of 1.1%. From the informative variables, they do not all select them in all iterations. Specifically, in 8 data sets it selects 19 informative variables out of 20, while in 3 data sets it selects 18 informative variables instead of 20. The corresponding average rate for selecting of informative variables is 98.6% with a standard deviation of 2.8% (that is 19.72/20 with a standard deviation 0.56 variables). Of the noninformative, it selects an average of 1.1% with a standard deviation of 1.5% (2.2/200 noninformative variables with standard deviation 3 variables). So, we conclude that the method has been slightly degraded to this simulation where the variance is increased. This seems also from removing incorrectly informative variables in some iterations. The average BIC is 53253.5 with a standard deviation of 190. The mean λ is 4.93 with a standard deviation of 0.5.

How does the method behave in clustering? An interesting question which arises is if it is as effective it was in the first simulation where the variance is 1. The answer to this question is that the method has been degraded to the part of clustering as the average number of clusters it has selected is 3.4 with a standard deviation of 0.78. Of the 50 data sets, it selects in 29 of them 4 clusters. The average misclustering error rate is 32.6% with a standard deviation of 18%.

But how do variables 1-20 behave in cluster discrimination? The answer is that when the method selects 4 clusters, on average 93.7% of the 1-10 variables are considered noninformative for the separation of clusters 2 and 3, with a standard deviation of 18.5% (9.37/10 variables instead of 10/10 variables). For variables 11-20, on average 2.4% of the variables consider the clusters 1 and 2 as one with a standard deviation of 6.8% (0.24/10 instead of 10/10). That is, a few times the method is able to shrink the means of the two clusters to 0. Finally, for variables 11-20 on average, 3% of the variables with a standard deviation of 8.1% are regarded as noninformative for the separation of clusters 3 and 4.

4.3.2. Implementation of APFP method

The next method is APFP. On average, it selects 12.7% (instead of 9.09%) of variables with a standard deviation of 2.466% variables (27.94/220 with standard deviation 5.28). From the informative variables, they are all chosen. Of the noninformative it chose an average of 3.6% variables, with a standard deviation of 2.4% (7.2/200 with standard deviation 4.8/200 variables). Regarding clustering, in all data sets it selects 4 clusters. The average error rate is 7.55% with a standard deviation of 5.9%. The average BIC is 54051.09 with a standard deviation of 208.02. In all data sets it selects through the BIC criterion, $\lambda = 2$.

Of the variables 1-10, on average are considered as noninformative to separate clusters 2 and 3, 78.2% variables, with a standard deviation of 14.3% (that is 7.82 variables instead of 10). For the variables 11-20, 84.4% of them with a standard deviation of 14.1% is considered noninformative for the separation of clusters 1 and 2 (8.44/20 with standard deviation 1.41/10 instead of 10/10 variables), while 84.8% is considered noninformative for the separation of clusters 3 and 4. Comparing the results of this simulation with those of the simulation 1 for the APFP method, we notice that the APFP method has also been negatively affected. However, it is more efficient in clustering than the AL1 method since it is able to find the correct number of clusters in all iterations and the average error rate is much smaller than AL1. In the previous simulation, both methods were effective. In this, APFP is more effective than AL1.

Table 4.3.2 contains all above comparative results between AL1 and APFP for simulation 2.

https://doi.org/10.26219/heal.aueb.5860

Method	***Variables***	***Pair of clusters***	***Percentage of variables considered noninformative to separate clusters (%)***
AL1	Variable 1-10	2/3	93.7 (18.5)
	Variables 11-20	1/2	2.4 (6.8)
		3/4	3 (8)
APFP	Variables 1-10	2/3	78.2 (14.3)
	Variables 11-20	1/2	84.4% (14.1%)
		3/4	84.8% (14%)

Table 4.3.2: Pairwise variable selection in simulation 2

4.3.3. Implementation of GMM method

We continue with the GMM method having all the variables in the model. The average BIC is 54768.46 with a standard deviation of 210.1. In all 50 sets of data, two clusters are selected. So, this means that the average misclustering rate is at least 50%. Also, in all iterations there has been a wrong clustering in the 2 clusters, i.e. many observations that belong to the first cluster have been incorrectly grouped in the second and vice versa. This means that the performance of GMM has worsened in relation to previous simulations and that the increase in variance has negatively affected the efficiency of clustering.

Next, we examine the effectiveness of GMM if we remove all noninformative variables. GMM without noise chooses an average of 3.48 groups with a standard deviation of 0.76, so, concluding that having removed the noninformative variables, the effectiveness of GMM over previous simulations has fallen. Apparently, the larger variances resulted in overlapping groups and the clustering problem became more difficult. More specifically,

 https://doi.org/10.26219/heal.aueb.5860

the means of groups 1, 2, 3 and 4, as shown in Table 4.2.1 are such as in case of variance 4 for the informative variables groups 1,2 and 3,4 respectively are very close to each other and overlapping. Of the 50 datasets, there are 8, in which only 2 clusters are selected, and in 32 datasets 4 clusters are selected. On average, the misclustering error is 12.46% observations with a standard deviation of 13% observations. The BIC of the method is 7329.468 with a standard deviation of 55.4

In the table 4.3.3 is the summary of the results mentioned above. In column K , we note for each method mean number of clusters. The ERR (%) column shows the average misclustering error rate of the observations. The ERR (%) K known mentions the average misclustering error rate when the number of cluster is known. The INFORMATIVE column indicates the percentage of informative variables inserted in the model in each method, calculated from the 50 data sets. The NONINFORMATIVE column calculates the average percentage of noninformative variables incorrectly inserted in the model. Column λ mention the average λ. The latter two methods have no value at λ as GMM has no penalty. The last column shows the BIC mean. In parenthesis there are the standard deviation of each mean value

METHOD	***K***	***ERR (%)***	***ERR (%) K known***	***INFORMATIVE (%)***	***NONINFORMATIVE (%)***	***λ***	***BIC***
AL1	3.4 (0.78)	32.6 (18)	16.3(15.8)	98.6(2.8)	1.1(1.5)	4.93 (0.5)	53253.5 (190)
APFP	4 (0)	7.55 (5.9)	5.7 (2.6)	100(0)	3.6(2.4)	2(0)	54051.09(208.02)
GMM	2(0)	>50	5.3 (2.4)	100	100	-	54768.46 (210.08)
GMM without noise	3.48 (0.76)	12.46 (13)	4.7 (2.26)	100	0	-	7329.468 (55.4)

Table 4.3.3: The summary of the results mentioned in the simulation 2.

https://doi.org/10.26219/heal.aueb.5860

That we observe from above simulation is that when the variance is higher than 1, the performance of the methods is degraded. Also, the GMM without noise although it hasn't any noninformative variable in the model, has more misclustering errors that the APFP method which has more noninformative variables in the model. In conclusion, experiment 2 has verified that the number of noninformative variable selected from the method is not the only parameter.

4.4. Simulation 3

In this simulation, we will look at how the methods behave when the clusters have different sizes. As is well known, many machine learning algorithms are degraded when there are clusters with a much larger number of observations than other clusters. In classical clustering these situations lead to either domination of the larger groups over to the remaining or merging of the smaller distinct groups to an artificial single group. For example, in theory we have mentioned that the algorithm K-Means isn't effective in clustering when we have clusters with different sizes. In this simulation clusters 1 and 2 have 20 observations each, while clusters 3 and 4 have 200 observations each. As in the previous simulation, each informative variable follows normal distribution with mean value $\mu_{k,j,}$ and variance 1, i.e. follows $\mathrm{N}(\mu_{k,j,}, 1)$, and the noninformative variables follow the $\mathrm{N}(0,1)$ distribution for all clusters. Cluster means are the same as simulation 1. What we want to examine with this simulation is whether the clusters with different sizes effect the clustering. That is, we want to examine if the methods are effective

- in correctly grouping the data when the clusters have different sizes.
- to remove the noninformative variables and
- to recognize the informative variables

The clusters that are created in this simulation are:

- Observations 1-20 are a group
- Observations 21-40 form a second group
- Observations 41-240 are a third group
- Observations 241-440 form a fourth group

Informative variables are 1-20, while the remaining 200 are noninformative. Because in this simulation the main issue is to examine the effectiveness of methods when we have different cluster sizes, we will not take into account that the variables will be less than the observations.

The following table summarizes the means of the four clusters in the informative variables as well as the number of observations from which each cluster is composed:

Variables	***Cluster 1*** ***20 observations***	***Cluster 2*** ***20 observations***	***Cluster 3*** ***200 observations***	***Cluster 4*** ***200 observations***
1-10	2.5	0	0	-2.5
11-20	1.5	1.5	-1.5	-1.5

Table 4.4.1 Cluster means in simulation 3

The graph below shows the clusters for variable 11 (representative of variables 11-20) as well as the clusters for variables 1 (representative of variables 1-10). What appears to be the case is that variable 11 distinguishes two clusters, while variable 1 distinguishes 3 clusters.

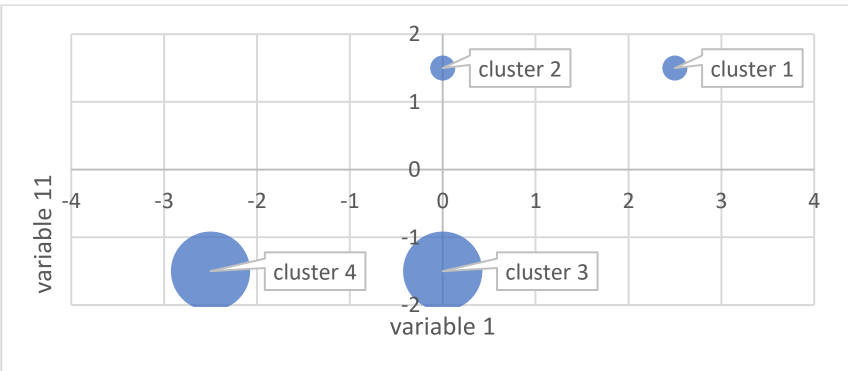

Figure 4.4.1 Clusters in simulation 3

4.4.1. Implementation of AL1 method

Because in AL1 method we have centered the data (i.e. each variable has mean value equal to 0), position of clusters plotted on variables 1 and 11 is as shown in Figure 4.4.2:

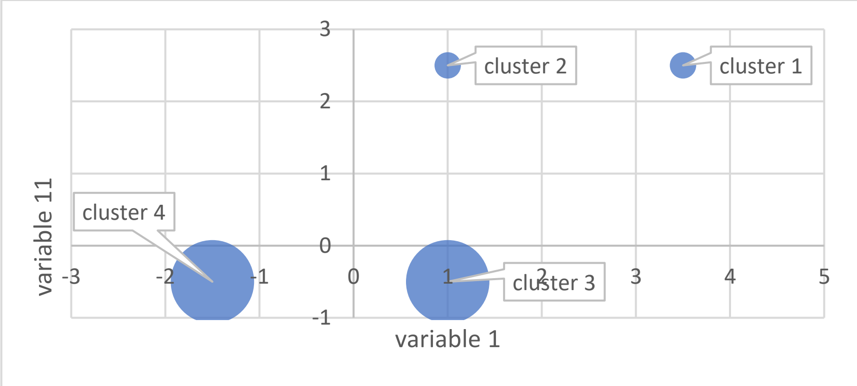

Figure 4.4.2 Clusters in simulation 3 after centering the data. As we observe, none of the cluster means in each variable are equal to 0.

As we see, when the data are centered, the means of cluster 1, 2, 3, 4 are 3.5, 1, 1, -1,5 respectively for the variables 1-10. For the variables 11-20, the means are 2.5, 2.5, -0.5, -0.5 for the clusters 1, 2, 3, 4 respectively. So, it means that for the AL1 method the variables 1-10 will be informative for the clusters 2 and 3 because their means are 1 and no 0.

After the implementation of the AL1 method in a similar way with the previous experiments, the mean BIC is 277090 with standard deviation 303.84. The mean value of λ is 6.3 with standard deviation 3. The variables selected to be kept in the model are on average 10.6% with standard deviation 2.3% (instead of 9.09% which is the real percentage of the informative variables). The method selects all the informative variables to be in the model. For noninformative variables, it selects on average 1.6% with standard deviation 2.5% (3.2/200 with standard deviation 5/200 variables).

Also, the variables 1-10 are informative to separate the clusters 2 and 3. The method never shrinks to 0 the cluster means of these two clusters. In addition, the variables 11-20 are informative to separate the clusters 1 and 2 as well as the clusters 3 and 4.

The next step is to examine how effective is the AL1 in clustering. After 50 repetitions, the result is that the clustering is correct. The clustering error rate is 0 with standard deviation 0. So, the method is very effective in this simulation despite the fact of disanalogous in size clusters. As we'll see in Table 4.3.2, which summarizes the results, the results for known K is also correct.

 https://doi.org/10.26219/heal.aueb.5860

4.4.2. Implementation of APFP method

Continuing with the APFP method, the mean of BIC is 278229.8 with standard deviation 389.29. The mean of λ is 5.6 with standard deviation 1.64. For all variables, it selects for the model, 9.4% variables with standard deviation 0.5% (instead of 9.09% variables). The method selects all the informative variables for the model. From the noninformative variables, it selects 0.3% with standard deviation 0.5% (0.6/200 with standard deviation 1/200). The variables 1-10, are noninformative for separating the clusters 2 and 3 with rate 95% and standard deviation 5.7% (9.5 variables instead of 10 variables). The variables 11-20 are noninformative for separating the clusters 1 and 2 with rate 94% and standard deviation 6.7% (9.4 variables instead of 10) and are noninformative for separating the clusters 3 and 4 with rate 94.3% and standard deviation 7.2% (9.43 variables instead of 10).

The clustering is also correct as well as in the APFP method when the K is known.

In the Table 4.3.2 contains all above comparative results between AL1 and APFP for simulation

Method	***Variables***	***Pair of clusters***	***Percentage of variables considered noninformative to separate clusters (%)***
AL1	Variables 1-10	2/3	0 (0)
	Variables 11-20	1/2	0 (0)
		3/4	0 (0)
APFP	Variables 1-10	2/3	95 (5.7)
	Variables 11-20	1/2	94 (6.7)
		3/4	94.3 (7.2)

Table 4.4.2: Pairwise variable selection in simulation 3

https://doi.org/10.26219/heal.aueb.5860

4.4.3. Implementation of GMM

GMM with all the noninformative variables in the model, always chooses 3 clusters. The wrong clustering rate is 4.5% as the cluster error is that it will always group a cluster of 20 observations with a cluster of 200 observations. The average BIC of the method is 28066.07 with a standard deviation of 408.6. In this simulation also, the GMM method fail to discover the real number of clusters.

How effective in clustering is the GMM if all noninformative variables are removed? GMM having removed all noninformative variables is very effective as it always chooses 4 clusters without grouping any observation wrong. Thus, the wrong clustering rate is 0 with a standard deviation of 0. The BIC is 26376 with a standard deviation of 150.8. Commenting on the result, we report in this experiment also the performance of GMM without noise is better than the performance of GMM with all the noninformative variables in the model. So, once again, it turns out to be important the variable selection in model-based clustering

The GMM method by keeping in the model all the noninformative variables for a known number of clusters is very effective as the incorrect grouping is 0 with a standard deviation of 0. This means that for GMM to work effectively in high dimensions we need to know always the number of clusters beforehand. This degrades the GMM compared with AL1, APFP methods where have been shown to be effective either knowing the number of clusters before or not (in simulations 1 and 3). In simulation 2 in which the variance is increased, the differences in clustering errors between knowing the number of clusters and not are smaller in AL1 and APFP than this of GMM.

The average BIC in GMM without noise is 79465 with standard deviation 68.3.

Finally, GMM having removed all noninformative variables is as effective as GMM having in the model the noninformative variables for known K as the mean misclustering error rate is 0 with a standard deviation of 0.

In Table 4.4.3 is the summary of the results mentioned in the simulation 3 for all the methods. In column K, we note for each method mean number of clusters. The ERR (%) column shows the average misclustering error rate of

https://doi.org/10.26219/heal.aueb.5860

the observations. The ERR (%) K known mention the average misclustering error rate when the number of cluster is known. The INFORMATIVE column indicates the percentage of informative variables inserted in the model in each method, calculated from the 50 data sets. The NONINFORMATIVE column calculates the average percentage of noninformative variables incorrectly inserted in the model. Column λ mention the average λ. The latter two methods have no value at λ as GMM has no penalty. The last column shows the BIC mean. In parenthesis there are the standard deviation of each mean value

METHOD	***K***	***ERR (%)***	***ERR (%) K known***	***INFORMATIVE (%)***	***NONINFORMATIVE (%)***	***λ***	***BIC***
AL1	4(0)	0(0)	0 (0)	100	1.6(2.5)	6.3 (3)	277090 (303.84)
APFP	4(0)	0(0)	0 (0)	100	0.3(0.5)	5.6 (1.64)	2782298 (389.29)
GMM	3(0)	4.5(0)	0 (0)	100	100	-	28066.07 (408.6)
GMM without noise	4(0)	0 (0)	0 (0)	100	0	-	26376 (150)

Table 4.4.3: The summary of the results mentioned in the simulation 3

4.5. Simulation 4

In this simulation a 5-cluster scenario is assumed, and we have 30 informative variables. This simulation is defined in larger dimensions and we discover the performance and the computational demand of the methodologies. Each cluster has 20 observations. The informative variables follow normal distribution with mean value $\mu_{k,j,}$ and variance 1, i.e. follows N $(\mu_{k,j,}, 1)$. The variables are 230. The remaining 200 are noninformative so, they are iid and follows $\mathrm{N}(0, 1)$ distribution.

The clusters that there are in this simulation are the below

• Observations 1-20 are a cluster

https://doi.org/10.26219/heal.aueb.5860

• Observations 21-40 constitute a second cluster

• Observations 41-60 are a third cluster

• Observations 61-80 are grouped into a fourth cluster

• Observations 81-100 are grouped into a fifth cluster

The option for K and λ for the methods is the below

- The options for K are 3, 4, 5
- The options for λ are 2, 5, 10, 15, 30

Variables	***Cluster 1***	***Cluster 2***	***Cluster 3***	***Cluster 4***	***Cluster 5***
1-10	2.5	2.5	0	0	-2.5
11-20	-2.5	0	0	0	2.5
21-30	2.5	0	0	-2.5	-2.5

Table 4.5.1 The cluster means in simulation 4

<u>4.5.1. Implementation of AL1 method</u>

Start with AL1 method. Before clustering we center the data. So, each variable will have mean value 0. That means that for variable 1-10, the cluster means will be 2, 2, -0.5, -0.5, -3 for each cluster respectively. For variables 21-30 the cluster means will be 3, 0.5, 0.5, -2, -2 respectively for each cluster.

The AL1 method inserts in the model 40 of 230 variables on average with standard deviation 13.7. That is 17.3% of variables are inserted in the model with standard deviation 5.96 (instead of $\frac{30}{230}\% = 13\%$ which is the percentage of the informative variables). It inserts in the model all the informative variables. From the noninformative variables are inserted in the model on average 4.94% variables with standard deviation 6.85 (9.88/200 with standard deviation 13.7 variables).

The variables 1-10 are always informative for separating the clusters 1 and 2. It means that the variables never manage to shrink their means to zero. They are noninformative for separating the clusters 3 and 4 with rate only 38.4% (instead of 100%) and standard deviation 32%. That happens because after

 https://doi.org/10.26219/heal.aueb.5860

centering the data, the means of cluster 3 and 4 are 0,5 and no 0. So, few times the method is able to shrink the cluster means to 0. After centering the data, the cluster means of variables 11-20 are the same. So, the variables are able to shrink the means of clusters 2,3 and 4 to 0. The variables 11-20 are noninformative for separating the clusters 2,3 and 4 with percentage of 98.4% with standard deviation 4.2%. Also, the variables always are noninformative for separating two of the three clusters. Finally, because after centering the data, the means of cluster 2 and 3 for variables 21-30 are 0.5 both, the variables are noninformative for separating them with rate only 35.6% and standard deviation 31.1%.

Regarding the clustering, the mean number of cluster is 4.98 with standard deviation 0.14. That is because in one dataset out of 50 the method chooses 4 clusters and no five. The misclustering error rate is 0.3% with standard deviation 2.79%.

4.5.2. Implementation of APFP method

Continue with the APFP method. The variables that chooses to insert in the model is 15.4% with standard deviation 3.07% (instead of 13% which is the percentage of informative variables). From the informative variables, the method always selects all of them. From the noninformative variables, the method selects to insert in the model 2.7% of them with standard deviation 3.5% (5.4/200 variables with standard deviation 7 variables). As we have mentioned, the APFP shrinks the clusters means toward to each other and no to 0. So, the zero hasn't any special meaning in this method. That means that the centering of the data doesn't affect on the performance of the method towards identifying the variables. So, the variables 1-10 are noninformative for separating clusters 1 and 2 with rate 85.8% with standard deviation 17.7%. The variables 11-20 are noninformative for separating the three clusters with rate 85.6% and standard deviation 17.2%. The variables 21-30 are noninformative for separating the clusters 2 and 3 with rate 84% and standard deviation 19.4% and are noninformative to separate the clusters 4 and 5 with rate 86.4% with standard deviation 13.6%. Regarding the clustering, the average number of clusters that is selects, is 4.86 with standard deviation 0.35. The misclustering error rate is 2.4% with standard deviation 6.56%.

 https://doi.org/10.26219/heal.aueb.5860

The Table 4.5.2 contains the above comparative results between the methods.

Method	***Variables***	***Pair of clusters***	***Percentage of variables considered noninformative to separate clusters (%)***
AL1	Variables 1-10	1/2	0 (0)
		3/4	38.4 (32)
	Variables 11-20	2/3/4	98.4 (4.2)
	Variables 21-30	2/3	35.6 (31.1)
		4/5	0 (0)
APFP	Variables 1-10	1/2	85.8 (17.7)
		3/4	81.6 (21.9)
	Variables 11-20	2/3/4	85.6 (17.2)
	Variables 21-30	2/3	84 (19.4)
		4/5	86.4 (13.6)

Table 4.5.2: Pairwise variable selection in simulation 4

4.5.3. Implementation of GMM

The GMM with all informative variables in the model selects always 3 clusters. So, the average misclustering error rate is 40 with standard deviation 0. When the K is known always selects 5 clusters. In this case the misclustering error rate is 0.02% with standard deviation 0.14%.

https://doi.org/10.26219/heal.aueb.5860

If remove all the noninformative variables from GMM, the number of clusters that the method selects is always 5.

Summary of the results mentioned above are in the Table 4.5.3. In column K, we note for each method mean number of clusters. The ERR (%) column shows the average misclustering error rate of the observations. The ERR (%) K known mention the average misclustering error rate when the number of cluster is known. The INFORMATIVE column indicates the percentage of informative variables inserted in the model in each method, calculated from the 50 data sets. The NONINFORMATIVE column calculates the average percentage of noninformative variables incorrectly inserted in the model. Column λ mention the average λ. The latter two methods have no value at λ as GMM has no penalty. The last column shows the BIC mean. In parenthesis there are the standard deviation of each mean value

METHOD	***K***	***ERR (%)***	***ERR (%) K known***	***INFORMATIVE (%)***	***NONINFORMATIVE (%)***	***λ***	***BIC***
AL1	4.98 (0.14)	0.3 (2.79)	0 (0)	100	4.94 (6.85)	6.02 (1.7)	66828.4(256.7)
APFP	4.86 (0.35)	2.4 (6.5)	0 .4(2.8)	100	2.7 (3.5)	5.5 (1.2)	67623 (218.7)
GMM	3(0)	40(0)	0 .02(0.14)	100	100	-	70037.6 (241)
GMM without noise	5(0)	0 (0)	0 (0)	100	0	-	9511.47 (85.8)

Table 4.5.3: Summary of the results from mentioned in simulation 3

4.6. Simulation 5

In this simulation the clusters are five as in simulation 4. Also, the variables are 230 where the 30 are informative as in simulation 4. The difference will be that the variance is 4. That is, the informative variables will follow

 https://doi.org/10.26219/heal.aueb.5860

$N(\mu_{k,j,}, 4)$ distribution. The remaining 200 are noninformative so, they are iid and follows $N(0,1)$ distribution. Table 4.6.1 gives the specific details with respect to the means of the assumed clusters per set of variables.

The clusters that there are in this simulation are the below

- Observations 1-20 are a cluster
- Observations 21-40 constitute a second cluster
- Observations 41-60 are a third cluster
- Observations 61-80 are grouped into a fourth cluster
- Observations 81-100 are grouped into a fifth cluster

Variables	***Cluster 1***	***Cluster 2***	***Cluster 3***	***Cluster 4***	***Cluster 5***
1-10	2.5	2.5	0	0	-2.5
11-20	-2.5	0	0	0	2.5
21-30	2.5	0	0	-2.5	-2.5

Table 4.6.1 The cluster means in simulation 5

4.6.1. Implementation of AL1 method

Start with AL1 method. Before clustering we center the data. So, each variable will have mean value 0. That means that for variable 1-10, the cluster means will be 2,2, -0.5, -0.5, -3 for each cluster respectively as in simulation 4. For variables 21-30 the cluster means will be 3, 0.5, 0.5, -2, -2 respectively for each cluster.

The method selects on average 15.06% variables (instead of 13% which is the percentage of the informative variables) with standard deviation 1.2%. From the noninformative variables, it selects 2.3% variables with standard deviation 1.4%. The method always inserts in the model all the informative variables.

The variables 1-10, are noninformative for separating the clusters 1 and 2 with percentage 0.2% and standard deviation of 1.4% (instead of 100%). For separating clusters 3 and 4 the variables are noninformative for separating them with an average rate of 90% and standard deviation 11.2%. The variables 11-

20 are noninformative to separate clusters 2, 3 and 4 with average rate of 96.2% (that is on average 9.6/10 instead of 10/10 variables are noninformative to separate the clusters 2,3 and 4) and standard deviation 5.6%. Finally, the variables 21-30 are never noninformative to separate clusters 4 and 5, but they are noninformative to separate clusters 2 and 3 with average rate of 84.2% and standard deviation 16.6%. Although the means of clusters 3, 4 corresponding to variables 1-10 are 0.5 both after centering, the rate of variables 1-10 that are noninformative for separating the clusters 3, 4 is large. Maybe it happens because when the variance is equal to 4, the data that are generated from the normal distribution is so that the means of clusters to be as closest to 0 as possible, for example cluster mean equal to 0.2 instead of 0.5.

Regarding clustering, the misclustering error rate is 15.3% with standard deviation 12.8%. The mean number of clusters that the method selects is 4.42 clusters with standard deviation 0.78 clusters. So, the method has degraded in clustering.

4.6.2. Implementation of APFP method

The APFP method selects on average 42.02 variables to insert in the model with standard deviation 7.25 variables (instead of 30 which are the number of informative variables). For them, the rate of the noninformative variables is 6% variables with standard deviation 3.6% variables. From the informative variables, all of them are selected for the model.

The variables 1-10 are noninformative for clusters 1 and 2 on average 79.8% with standard deviation 13.6%. For clusters 3 and 4, the variables 1-10 are noninformative for separating them with average rate 78.2% and standard deviation 12%. The variables 11-20, are noninformative for separating the clusters 3, 4 and 5 with average rate 79% and standard deviation 17.7%. The variables 21-30 are noninformative for separating the clusters 2 and 3 with average rate 74% and standard deviation 14% and they are noninformative for separating the clusters 4 and 5 with average rate 78.8% and standard deviation 14.7%

The clustering has also been degraded when the variance is 4. More specifically, the method selects on average 4.5 clusters with standard deviation

 https://doi.org/10.26219/heal.aueb.5860

0.64. The average misclustering error rate is 13.1% with standard deviation 10.3%.

Method	***Variables***	***Pair of clusters***	***Percentage of variables considered noninformative to separate clusters (%)***
AL1	Variables 1-10	1/2	0.2 (1.4)
		3/4	90 (11.2)
	Variables 11-20	2/3/4	96.2 (5.6)
	Variables 21-30	2/3	84.2 (16.6)
		4/5	0 (0)
APFP	Variables 1-10	1/2	79.8 (13.6)
		3/4	78.2 (12)
	Variables 11-20	2/3/4	79 (17.7)
	Variables 21-30	2/3	74 (14)
		4/5	78.8 (14.7)

Table 4.6.2: Pairwise variable selection in simulation 5

4.6.3. Implementation of GMM method

Finally, the GMM with all the informative variables in the model selects always 3 clusters because always it groups 2 clusters with other clusters. For example, it groups the cluster 1 with cluster 2 and cluster 3 with cluster 4. The average misclustering error rate is 40.14% with standard deviation 0.4%.

If the noninformative variables are removed, the method selects on average 4.02 clusters with standard deviation 0.14 clusters. The misclustering error rate is 19.4% with standard deviation 3.85%

https://doi.org/10.26219/heal.aueb.5860

In the Table 4.6.3 is the summary of the results mentioned above. In column K , we note for each method mean number of clusters. The ERR (%) column shows the average misclustering error rate of the observations. The ERR (%) K known mention the average misclustering error rate when the number of cluster is known. The INFORMATIVE column indicates the percentage of informative variables inserted in the model in each method, calculated from the 50 data sets. The NONINFORMATIVE column calculates the average percentage of noninformative variables incorrectly inserted in the model. Column λ mention the average λ. The latter two methods have no value at λ as GMM has no penalty. The last column shows the BIC mean. In parenthesis there are the standard deviation of each mean value

METHOD	***K***	***ERR (%)***	***ERR (%) K known***	***INFORMATIVE (%)***	***NONINFORMATIVE (%)***	***λ***	***BIC***
AL1	4.42 (0.78)	15.2 (12.8)	10.2 (10.87)	100(0)	2.3(1.4)	5 (0)	70890 (255)
APFP	4.5 (0.64)	13.1 (10.3)	10.7 (8.9)	100(0)	6(3.6)	2(0)	71856.8 (244)
GMM	3(0)	40.14 (0.4)	5.3 (2.4)	100	100	-	73334 (240)
GMM without noise	4.02 (0.14)	19.4 (3.85)	4 (2.1)	100	0	-	13630 (82.5)

Table 4.6.3: Summary of the results mentioned in simulation 5

4.7. GOLUB DATASET

Golub et al. (1999) studied discovering two subtypes of human acute leukemia using gene expression data. These subtypes are the acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). The separation of

https://doi.org/10.26219/heal.aueb.5860

the two subtypes is important because the medication is different between the people who belong to the different subtypes. The dataset there is in the 'plsgenomics' library in the R. It consists of 3051 genes that are the variables. The observation are 38 patients. 27 of them belongs to AML subtype and the 11 of them belong to ALL subtype. Specifically, the observations 1-27 are the AML subtype and the observations 28-38 are in the ALL subtype.

The selection of number of clusters is 1,2,3,4 and the selection of the λ parameter is 5,10,15,30.

We repeat 50 times the dataset for each method and then we select the time with the minimum BIC criterion.

4.7.1. Implementation of AL1 method

The AL1 method selects to insert into the model $\frac{35}{3051}$ variables. The BIC is 179244.8 and the λ is 15

Regarding the clustering, AL1 groups the observations in 3 clusters.
The first cluster consist of the observation 17
The second cluster consist of the observations 2, 3, 6, 9, 10, 11, 14, 23
The third cluster consists of the observations 1, 4, 5, 7,8, 12, 13,15,16,18,19,20,21, 22, 24,25,26,27, 28, 29,30,31,32,33,34,35,36,37,38

Group / ***Cluster***	***AML***	***ALL***
Cluster 1	1	0
Cluster 2	8	0
Cluster 3	18	11

Table 4.7.1: The clustering results of AL1 method in Golub data

As we see from the Table 4.7.1, the misclustering error is

$$\frac{18+1}{38}=\frac{19}{38}$$

When the only selection is K=2, the first cluster consists of the observations1,3-11,13-24,26,27,35 and the second cluster consist of the observations 2,12,25,28-34,36,38. In the Table 4.7.2 appears the clustering when the only option is K=2

 https://doi.org/10.26219/heal.aueb.5860

Group / Cluster	*AML*	*ALL*
Cluster 1	24	1
Cluster 2	3	10

Table 4.7.2: The clustering results of AL1 method in Golub data when K is known

When the K=2, the misclustering error is

$$\frac{3+1}{38} = \frac{4}{38}$$

So, the clustering is better when the number of clusters is known.

4.7.2. Implementation of APFP method

The performance of APFP is better than AL1. It selects 3 clusters. More specifically, the first cluster consist of the observations
1,4,5,7,8,13,15,16,17,18,19,20,21,22,24,25,26,27
The second cluster consists of the observations 2,3,6,9,10,11,14,23.
The third cluster consists of the observations 12,28-38
It selects 86/3051 variables

Group / Cluster	*AML*	*ALL*
Cluster 1	18	0
Cluster 2	8	0
Cluster 3	1	11

Table 4.7.3: The clustering results of APFP method in Golub data

As we see from the above Table, the observations of the AML subtype have been grouped in clusters 1 and 2 (except of 1 observations which has grouped in cluster 3). All the observations of ALL subtype have been grouped correctly in cluster 3. So, the misclustering error is

$$\frac{8+1}{38} = \frac{9}{38}$$

https://doi.org/10.26219/heal.aueb.5860

When the only option for the number of clusters is 2 (which is the number of subtypes), then the misclustering error is 2. Specifically, the first cluster consist of the observations 1-11, 13-24,26,27 and the second cluster consists of the observations 12,25,28-38. In the Table 4.7.4 appears the clustering when the only option is K=2

Group / ***Cluster***	***AML***	***ALL***
Cluster 1	25	0
Cluster 2	2	11

Table 4.7.4: The clustering results of APFP method in Golub data when the K is known

4.7.3. Implementation of GMM

The GMM always selects 1 cluster. That happens because all the noninformative variables are into the model.

If the only selection for the number of clusters is 2, the clustering of nopenalty is good. Specifically, the first cluster consists of the observations 1-11,13-24,26,27

The second cluster consists of the observations 12,25,28-38

Group / ***Cluster***	***AML***	***ALL***
Cluster 1	25	0
Cluster 2	2	11

Table 4.7.5: The clustering results of GMM method in Golub data when K is known

Therefore, the only selection is K=2, the misclustering error of GMM is

$$\frac{2+0}{38} = \frac{2}{38}$$

So, we observe that the performance of GMM isn't good because all the variables that select one cluster (noninformative variables) are into the model. Between the AL1 and APFP, AL1 has the highest error in this dataset.

So, in this dataset, the APFP is performed better than AL1

 https://doi.org/10.26219/heal.aueb.5860

4.8. Summary of the results of the applications

Summarizing, we will mainly comment on the differences between them. The AL1 method doesn't consider a variable noninformative for separating two clusters if the cluster means aren't equal to mean value of variable that is 0. On the other hand, according to the APFP method a variable is noninformative if the cluster means are equal. In APFP the 0 has no special meaning. This is the main difference between them. This difference affect the performance of AL1 in some of the above applications results in comparison with APFP method. In simulation 1, for variable 11-20, the cluster means are 1.5, 1.5, -1.5, -.1.5. Because the AL1 consider as noninformative only the variables in which the cluster have the same means as the mean value of variable, the variables 11-20 are informative for separating the clusters 1,2 and 3,4. Also, the means of cluster 2 and 3 in simulation 1 and 2 are shrunken with higher rate in 0 with AL1 method because the 0 is the value that the AL1 shrinks the cluster means. In the first simulation the two penalized methods are very effective. Both have a very good performance in clustering. About the removing of noninformative variables, the AL1 is better. It has inserted in the model 0.7% noninformative variables, while the APFP has inserted in the model 1.3% noninformative variables. In the second simulation ($\sigma^2 = 4, K = 4$), the performance of the methods has been degraded. Between the two penalized methods, the APFP method has the lower misclustering error rate, so it is better in this simulation's clustering (the misclustering error rate of APFP is 7.55, while the rate of AL1 is 32.6). But the AL1 method has correctly removed more noninformative variables (the rate of noninformative variables inserting in model is 1.1 for AL1 and 3.6 for APFP). Continuing with simulation 3 where we have unbalanced data, both of penalized method have zero misclustering error rate. So, both are effective in clustering of different size clusters. In this simulation the APFP is more effective in removing of noninformative variables. In simulation 4 where we have 5 clusters, the AL1 has slightly lower misclustering error rate than APFP (AL1→0,3%, APFP→2.4%). But, the APFP is better in removing noninformative variables (AL1 keeps in the model 4.94% noninformative variables. The APFP keeps on average 2.7% noninformative variables). In simulation 5, the APFP method has lower misclustering error rate than APFP (AL1→15.2%, APFP→13.1%) but AL1 is better in removing noninformative variables (AL1→2.3%, APFP→6%).

In Table 4.8.1 perform the percentage of clustering errors of all applications for each method

Method / ***App***	***APFP***	***AL1***	***GMM with noise***	***GMM without noise***
Simulation 1	0 (0)	0.025 (0.17)	12.46 (18)	5×10^{-2} (0.2)
Simulation 2	7.55 (5.9)	32.6 (18)	>50	12.46 (13)
Simulation 3	0 (0)	0 (0)	4.5 (0)	0 (0)
Simulation 4	2.4 (6.5)	0.3 (2.79)	40 (0)	0 (0)
Simulation 5	13.1 (10.3)	15.2 (12.8)	40.14 (0.14)	19.4 (3.85)
Golub data	9/38	19/38	One cluster	-

Table 4.8.1: Summary of misclustering error rate of all methods in all applications

The performance of GMM method is always degraded in clustering in comparison with the two penalized methods. If we remove the noninformative variables from the model, the performance of GMM is comparable with the performance of penalized methods. Consequently, it is very useful to have remove of any noninformative variables in order to have better results.

Another conclusion is that if a method is better in removing variables, that doesn't mean that this method will be better in clustering. For example, in simulation 2, the AL1 is better in removing variables that APFP (AL1 has keep into the model 1.1% noninformative variables and the APFP has keep 3.6% noninformative variables), but the misclustering error rate of APFP is much lower than AL1 (AL1→32.6, APFP→7.55). Also, in simulation 5, the GMM method without noninformative variables has higher misclustering error rate than the penalized methods.

 https://doi.org/10.26219/heal.aueb.5860

This confirms that the two tasks of

(i) removing noninformative variable and

(ii) perform a cluster analysis can be considered as independent tasks and combine possibly different techniques at each stage.

In general, as in cluster analysis itself, a standard procedure is to implement several techniques before concluding to a classification. Especially when in real situations the true classification of the data is not known as in simulated data and trying various methods may help to reveal special structure in the data or help to exploit special characteristics (advantages) one method may have compared to others.

Another result from the comparative study we have conducted is that the performance of methods is degraded if the variance is getting larger (keeping the means constant) which is expected as in this case the distinction between groups is less clear.

Finally, the APFP method has more computational cost than AL1 method. That happens because the AL1 shrinks the cluster means to a specific value, the 0. The APFP method using the local quadratic approximation, try to shrink the cluster means toward to each other. For this reason, the method has more computational cost.

https://doi.org/10.26219/heal.aueb.5860

CHAPTER 5
CONCLUSION

In this thesis we have developed four penalized methods proposed in the literature to deal with the problem of make automatic variable selection in clustering. As pointed out in previous chapter, in high dimensional data sets where the number of variables is much larger than the number of observations the problem of 'curse of dimensionality' is introduced (Michael Steinbach et al, 2002). According to it, as the dimensions increase, the observations are become more 'sparse'. It means that as the number of dimensions became larger, the distance between the observations become also larger. It is explained in Chapter 2 with an example. Therefore, If the number of dimensions converges to infinity, then the furthest distance between the observations converges to closest distance. It has negative impact in cluster algorithms which depend on distance between the observations such as the K-Means algorithm and hierarchical algorithms.

Regarding to GMM, the problem leads to overparamerized model. Overparamerization causes poor performance of machine learning algorithms like the GMM algorithm. According to an example that we developed, if we have 30 observations which are grouped in 3 clusters and the number of variables is 200, then the number of parameters to be estimate is about 60000, that is very large in comparison with the number of observations. For a correctly estimate, we need a large number of observations. A solution for decreasing the parameters is to assume that the clusters have the same covariance matrix. Then the number of parameters will be about 20000. But, the number of parameters to estimate continue to be very large in comparison with the observations. If we assume that $\boldsymbol{\Sigma_k}= \boldsymbol{\Sigma}$ =diag $(\sigma_1^2, \sigma_2^2, \ldots, \sigma_p^2)$, then the number of parameters is about 800. For this reason, we assume the above variance-covariance matrix. But we still have many parameters compared to the observations. So, it is useful to be removed some dimensions.

Many researchers have suggested the PCA method for reducing dimensions. The variables (principal components) resulting from this method is a linear combination of the original variables. Such researchers as we have mentioned

 https://doi.org/10.26219/heal.aueb.5860

in Chapter 2 are Mansfield et al (1977, p.38) who argued that if only the principal components with the largest variance are kept, then the variability of the model in regression will be minimally reduced. Mosteller and Tukey examined the PCA method in variable selection in regression. According to them the principal components with the lowest variances don't contribute to regression. But, there are researcher (such as Kung and Sharif (1980)), that disagree with the above researchers and have proved that the principal components with the lower variance can be more important for the model than the ones with greatest variances. Yeung and Ruzzo (2001) have claimed that the PCA method before clustering degrades the quality of clustering. They have proved it because they compared the results without applying PCA method before clustering with the results applying PCA before clustering. Conclude that the PCA method isn't such a good idea, since the results are better without applying PCA before clustering.

Also, as we have referred, because in clustering we don't know from the start in which cluster has grouped each observation and because there is little or no knowledge about the analysis and the structure of the clustering that is to be done, it isn't possible to apply a technique for the rejection or acceptance of variables before the cluster analysis. So, we conclude that the process of variable selection must be inserted into the model and no prior of clustering because if it is applied before clustering, the clustering sculpture is destroyed. Because with the hierarchical and K-Means algorithms we can't do statistical inference, the process of variable selection is inserted to model-based clustering. In model- based clustering, the choice of the model become with the BIC criterion. So, we have a criterion for the choice of the number of clusters.

For this reason, the penalized methods which are analyzed in Chapter 3 are developed. These methods apply variable selection into the model. According to AL1 method, after centering the data, a variable is noninformative for separating two clusters if the means of clusters are shrunken in the same value with the mean value of the variable which is 0 (Pan and Shen, 2007). If all cluster means are equal with the mean value of the variable, then the variable is removed from the model. The second method that we refer in the Chapter 3 is the ALP-GMM method (Wang and Zhu, 2008). According to this, if the maximum cluster mean in absolute value is shrunken to 0 in a variable, then all

https://doi.org/10.26219/heal.aueb.5860

the cluster means in this variable will be 0 and so the variable will be removed from the model. Because the ALP treats the cluster means with an 'all-in-all-out' manner, the AHP-GMM method is proposed as an improvement from the same researchers as the ALP-GMM method. According to AHP-GMM, any cluster mean can be shrunken to zero, not only the greatest in absolute value cluster mean. If the selected cluster mean is shrunken to zero (i.e. is estimated 0), then all the cluster mean in this variable will be shrunken to 0 (i.e. will be estimated 0). But if the selected cluster means isn't shrunken to zero, then the other cluster means have the probability to be 0. So, the AHP-GMM method keeps the flexibility of AL1 method. The last method is the APFP method and according to this, a variable will be noninformative for separating two clusters, if the means of these clusters are shrunken to the same value (developed by Jian Guo, 2010). In APFP method the zero has no special meaning. The variable will be removed from the model if the means of all clusters associated with variable are shrunken to the same value.

In Chapter 4, we have compared the AL1 method and the APFP method with GMM method and GMM without noise method. With these applications, we examined the performance of the methods when we have equal cluster sizes with variance 1, when the variance is 4 and the case in which we have inequal cluster sizes where the size of 2 clusters is 10 times larger than the size of the other 2 clusters. According to simulations, the performance of the penalized method is degraded when the variance is 4. But, the inequal cluster sizes don't affect the performance of them. Also, the GMM with all noninformative variables in the model never find the correct number of clusters, but when all the noninformative variables are removed, the performance of GMM is comparable with this of penalized methods. This shows how important is to remove the noninformative variables from the model because it is leads to a more correct clustering. So, from the above, such it has been turned out, the two tasks of

(i) removing noninformative variables and
(ii) perform a cluster analysis

can be considered as independent tasks, because if a method removes more noninformative variables than another method, it doesn't mean that the first method will group more correct the observations. It has been turned out in simulations.

Something else that is concluded from the simulations is that when the cluster means are not equal to 0 (0 is the means value of the variables after centering the data), then the AL1 method assume that the variables are informative for separating the clusters. It is appeared in the Table 4.3.2 in which after centering the data, the cluster means are different from zero. This is the main difference between the AL1 and APFP method which affect in the results of AL1. Another difference between them is that the APFP method has more computational cost than AL1.

Finally, after the simulations we compare the methods in Golub dataset. In this dataset, the APFP performs better than AL1 method.

So, a last conclusion is that the APFP performs better than AL1.But, this does not mean that the use of APFP alone is sufficient. For more certain results, it is good to use both methods.

https://doi.org/10.26219/heal.aueb.5860

APPENDICES

The code which was used for the simulation 1 is the below.

```
library(PARSE)
AA=matrix(rep(0,50*80),nrow=50,ncol=80)# in this matrix will be the observations of 1st
##cluster. In each row are the observations in first the cluster in each of 50 datasets
BB=matrix(rep(0,50*80),nrow=50,ncol=80)

CC=matrix(rep(0,50*80),nrow=50,ncol=80)
DD=matrix(rep(0,50*80),nrow=50,ncol=80)

KBEST=NULL###in this will be the number of cluster which are chosen in each of 50
####datasets
LBEST=NULL###in this will be the parameter λ which is chosen in each of 50 datasets

BICBEST=NULL###in this will be the bic which is chosen from the 20 repetitions for each
##of 50 datasets
MEANSS=matrix(rep(0,80*50),nrow=80,ncol=50) ####the cluster means in each variable in
###each of the 50 datasets

INFOO=matrix(rep(0,50*220),nrow=50,ncol=220) )# in this matrix will be the variables
####which are selected for the model in each of 50 datasets

for (k in 1:50){
A=matrix(rep(0,20*80),nrow=20,ncol=80) )# in this matrix will be the observations of 1st
##cluster. In each row are the observations in first the cluster in each of 20 repetitions

B=matrix(rep(0,20*80),nrow=20,ncol=80)

C=matrix(rep(0,20*80),nrow=20,ncol=80)

D=matrix(rep(0,20*80),nrow=20,ncol=80)
INFO=matrix(rep(0,20*220),nrow=20,ncol=220) # in this matrix will be the variables
####which are selected for the model in each of 20 repetitions. It is chosen the repetition with
####the lowest BIC value
```

https://doi.org/10.26219/heal.aueb.5860

MEANS=matrix(rep(0,80*20),nrow=80,ncol=20) ####the cluster means in each variable in ###each of the 20 repetitions. It is chosen the repetition with the lowest BIC value

set.seed(k)

y1=rbind(matrix(rnorm(200,2.5,1),ncol=10),matrix(rnorm(200,0,1),ncol=10),matrix(rnorm(200,0,1),ncol=10),matrix(rnorm(200,-2.5,1),ncol=10)) ###the cluster means for the variables ####1-10

y2=rbind(matrix(rnorm(200,1.5,1),ncol=10),matrix(rnorm(200,1.5,1),ncol=10),matrix(rnorm(200,-1.5,1),ncol=10),matrix(rnorm(200,-1.5,1),ncol=10))##the cluster means for the variables ##11-20

y3=rbind(matrix(rnorm(4000,0,1),ncol=200),matrix(rnorm(4000,0,1),ncol=200),matrix(rnorm(4000,0,1),ncol=200),matrix(rnorm(4000,0,1),ncol=200))##the noninformative variables

y=cbind(y1,y2,y3)

y=scale(y[,1:220], center = TRUE, scale = FALSE)###we center the data before the analysis

kbest=NULL ###in this will be the number of cluster in each of the 20 repetitions. It is ####chosen the repetition with the lower BIC value

lbest=NULL####in this will be the λ parameter in each of the 20 repetitions. It is chosen the #####repetition with the lower BIC value

bicbest=NULL####in this will be the BIC value in each of the 20 repetitions. It is chosen the #####repetition with the lower BIC value

for (i in 1:20){

om1=NULL ##vector which keep the the observations in 1st cluster in each repetition

om2=NULL

om3=NULL

om4=NULL

means=NULL###vector with the cluster means in each variable

info=NULL###vector which keep the variables that are selected in each repetition

output1<-apL1(K=c(2,3,4),lambda=c(2,5,10,15,30),y=y)###the R command for AL1 method

info=c(summary(output1,y,eps.diff=1e-5)$info.name, rep(0,220-length(summary(output1,y,eps.diff=1e-5)$info.name))) ###command which appear the ####selecting variables in each repetition

https://doi.org/10.26219/heal.aueb.5860

```
INFO[i,]=info##the selecting variables from the above line are inserted in a row in INFO
##matrix
kbest=c(kbest,output1$K.best)
lbest=c(lbest,output1$lambda.best)
bicbest=c(bicbest,output1$bic.best)
means=c(output1$mu.hat.best[,1:20],means)

for (j in 1:80){
if (output1$s.hat.best[j]==1){
om1=c(om1,j)
}
if (output1$s.hat.best[j]==2){
om2=c(om2,j)
}

if (output1$s.hat.best[j]==3){
om3=c(om3,j)
}
if (output1$s.hat.best[j]==4){
om4=c(om4,j)
}
om11=c(om1,rep(0,80-length(om1)))##an 80-length vector which has the observations of the
###1st cluster. This vector will be a row in A matrix
om22=c(om2,rep(0,80-length(om2)))
om33=c(om3,rep(0,80-length(om3)))
om44=c(om4,rep(0,80-length(om4)))
meanss=c(means,rep(0,80-length(means)))##an 80-length vector which has the cluster means
###for the 20 informative variables. This will be a row in MEANS matrix
A[i,]=om11
B[i,]=om22
C[i,]=om33
D[i,]=om44
MEANS[,i]=meanss
}
}
minn<-which.min(bicbest)##we select the repetition with the lowest BIC
AA[k,]=A[minn,]
```

```
BB[k,]=B[minn,]
CC[k,]=C[minn,]
DD[k,]=D[minn,]
INFOO[k,]=INFO[minn,]
KBEST=c(KBEST,kbest[minn])
LBEST=c(LBEST,lbest[minn])
BICBEST=c(BICBEST,bicbest[minn])
MEANSS[,k]=MEANS[,minn]
}
```

REFERENCES

Adrian E. Raftery, Nema Dean (2006). Variable Selection for Model-Based Clustering, *Journal of the American Statistical Association 100, 602-617*

Bouveyron C. & Brunet - Saumard, C., (2014) Model-Based Clustering of HighDimensional Data, A review, *Computational Statistics & Data Analysis,* 71, 52-78.

Brusco, M. J., and Cradit, J. D. (2001). A Variable Selection Heuristic for k-Means Clustering, *Psychometrika*, 66, 249–270.

Charles Bouveyron, Camille Brunet (2013), Model-Based Clustering of High-Dimensional Data: A review, *Computational Statistics and Data Analysis, Elsevier,* 71, pp. 52-78.

Chris Fraley and Adrian E. Raftery (1998). How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis

David D. Lewis (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In C. Ne´dellec and C. Rouveirol (eds), *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pp. 4–15. *Heidelberg: Springer-Verlag.*

Domingos, P. and Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zeroone loss. *Machine Learning*, 29, 103–130.

Friedman, J. and Meulman, J (2004). Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society, Series B* 66**,** 815-849

George Karypis Eui-Hong (Sam) Han Vipin Kumar (1999). CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling, *IEEE Computer Society*, p. 68-75

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

Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu, (2010). Pairwise Variable Selection for High Dimensional Model Based Clustering, *Biometrics* 66, 793-804

https://doi.org/10.26219/heal.aueb.5860

Kung, E. C. and Sharif, T. A. (1980). Multi-regression forecasting of the Indian summer monsoon with antecedent pattern of the large-scale circulation. *In WMO Symposium on Probabilistic and Statistical Methods in Weather Forecasting*, pp. 295-302

K. Y. Yeung and W. L. Ruzzo (2001) Principal component analysis for clustering gene expression data, pages 763-774

Liu, Zhang,Palumbo,Lawrence (2003). Bayesian Clustering with Variable and Transformation Selection (with discussion). *Bayesian Statistics* **7**,249-275

Mansfield, E. R., Webster, J. T. and Gunst, R. F. (1977). An analytic variable selection technique for principal component regression. *Appl. Statist.,* 26, 34-40.

Michael Steinbach, Levent Ertöz, and Vipin Kumar (2004). The Challenges of Clustering High Dimensional Data*

Mosteller, F. and Tukey, J. W. (1977). Data Analysis and Regression: A Second Course in Statistics. *Reading, Mass.: Addison-Wesley*

Raftery and Dean, (2006). Variable Selection for model-based clustering. *Journal of the American Statistical Association* 101, 168-178

Rosie Cornish (2007). 3.1 *Cluster Analysis*. .Mathematics Learning Support Centre

Sijian Wang and Ji Zhu (2008). Variable Selection for Model Based Clustering and Its Application to Microarray Data, *Biometrics* 64, 440-448

Wei Pan, Xiatong Shen (2007). Penalized model-based clustering with application to variable selection, *Journal of Machine Learning Researcher* 8 1145-1164

ZHEXUE HUANG (1998). Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values, *Data Mining and Knowledge Discovery* 2, 283-304

https://doi.org/10.26219/heal.aueb.5860