Exponential Power Mixture Model for Regression: Estimation and Variable Selection

Lini Cao

School of Mathematics, Statistics and Actuarial Sciences
University of Kent, Canterbury

A Thesis Submitted for the Degree of Doctor in Philosophy

September, 2013

Acknowledgements

At the point of finishing this thesis, first and foremost, I would like to show my deepest gratitude to my supervisor, Prof. Jian Zhang, a respectable, responsible and resourceful scholar, who has provided me with valuable guidance in every stage of the writing of this thesis. Without his enlightening instruction, impressive kindness and patience, I could not have completed my thesis. His keen and vigorous academic observation enlightens me not only in this thesis but also in my future study.

Secondly, I'd like to thank those leaders, teachers and working staff in our department, especially Claire Carter, TC Dunning and Judith Broom. Without their help, it would be much harder for me to finish my study and this thesis.

I would like to express my heartfelt gratitude to my friend Chao Liu and Christiano Villa for their friendship and constructive suggestions, they constantly encouraged me when I felt frustrated with this thesis.

Special thanks would go to my beloved husband for his loving considerations and great confidence in me all through these years.

Last but not least, I am much indebted to my family, without whose affection and support this thesis could not have appeared in its final form.

Abstract

The mixture regression model is an important technique used in statistical modelling to investigate the relationship between variables. It has been applied in many fields such as genetics, finance and biology. In this research, we focus on its application to genetic data. As we know gene expression data normally contains unknown correlation structures even after normalization, hence it raises a great challenge for the existing clustering methods such as the Gaussian mixture(GM) model and k-mean. Here we use the exponential power distribution to robustly overcome the clustering of gene expression data by treating the data as a mixture of regression. The exponential power distribution (EPD) is a scale mixture of Gaussian distributions that has varying shape parameters. In this study we introduce and develop our method based on two different aspects of multiple regression with random errors distributed according to the exponential power distribution. The first aspect is estimation: we use both the Expectation-Maximisation algorithm (EM) and the Newton-Raphson method to estimate the parameters of the exponential power distribution mixture regression models. The second aspect is simultaneous variable selection and clustering: we develop a LASSO-type method to select only the related variables in a large dataset, especially for a high dimensional dataset. The novelty of this research regarding to the Expectation-Maximization algorithm is that we convert each penalised mixture regression estimation problem to a LASSO (Least absolute shrinkage and selection operator) problem. The performance of our method is assessed on both independent and dependent data. We also compared the EPD mixture regression with Gaussian mixture regressions by simulations and real data analyses. We also derive the model selection criteria such as AIC, BIC and EBIC for both EPD mixture and GM models.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The past few decades have witnessed a tremendous amount of studies on the problem of relating a response variable to a set of covariates through a regression-type model under a homogeneity assumption in which the regression coefficients are the same for different observations. Since these coefficients may change for different subgroups, the above assumption may be inadequate when the population under investigation is heterogeneous. Such heterogeneity can be modelled with a Finite Mixture Regression (FMR) model, for example, the Gaussian mixture regression (Staedler et al., 2010; Grun and Leisch, 2007; Sung, 2004; Fan and Li, 2001; Williams and Rasmussen, 1996). Gaussian mixture(GM) regression models have many applications in several research fields such as genetics, econometrics and social sciences. In Gaussian mixture models, given covariates, the response variable is assumed to follow a Gaussian mixture distribution and the observations are assumed to be independent of each other, which are often invalid in practice. For example, gene expression data may have a non-Gaussian mixture distribution with correlated structures (Zhang and Liang, 2010). Thus, a question is naturally raised: if the above Gaussian mixture assumptions do not hold, what kind of models should we use in order to avoid potential modelling biases? In this thesis, it's shown that mixtures of exponential power distributions(EPD) can be employed to address the above issue and to achieve a more flexible modelling.

The exponential power distribution introduced by Subbotin (1923) had not attracted any attention until Box and Tiao (1973) re-examined this model. Since then, it has been widely used in economics and finance as a generalized Gaussian distribution, as shown by Liu and Bozdogan (2008), Theodossiou (1998), Rachev and Mittnik (2000), and Toyli et al. (2002). Very recently, Zhang and Liang (2010) applied the exponential mixture model to provide a more robust analysis of some structured genetic data. Zeckhauser and Thompson (1970) was the first to study the simple linear regression model with exponential power error terms. The EPD is often used to fit the data with distribution tails heavier or lighter than Gaussian tails (Box and Tiao, 1973). The tail shape of an EPD is described by its shape parameter. The exponential power distribution (EPD) is a scale mixture of Gaussian distributions that has varying shape parameters. Therefore, the EPD is more flexible compared to a Gaussian distribution in practice. The use of the EPD mixture can reduce the modelling bias and also increase the robustness of data analysis to outliers.

In order to select a "better" model, we need to choose a suitable information criterion for model selection. The Akaike information criterion (AIC) and Bayesian information criterion (BIC) are the most widely used model selection criteria in the literature. The AIC proposed by Akaike (1974) uses the Kullback-Leibler distance to justify the goodness of a selected model. The AIC penalizes the number of parameters less strongly than the BIC does, which was developed by Schwarz (1978). The nature of BIC is different from AIC as it assumes that one of the models is the "true" model. Chen and Chen (2008) proposed an extended family of BIC (EBIC), which considers both the number of unknown parameters and the complexity of the model space. In this study, we use these information criteria to assess the performance of the selected model.

There are two goals in this thesis. The first one is to build a so-called Expectation-Maximisation (EM) algorithm to calculate the maximum likelihood estimators for exponential power mixture regression models. Another target for this thesis is to develop a method to perform cluster analysis and variable selection simultaneously for high-dimensional non-Gaussian regression data. Here, the novelty lies in that we convert a general penalised regression estimation problem to a special L_1 penalised regression (i.e., LASSO) problem, where LASSO is short for Least absolute shrinkage and selection operator (Tibshirani, 1996). The proposed method is assessed on both simulated and real data, and is also compared to the existing Gaussian mixture regression based approach. The simulations and the real data analysis suggest that exponential power mixture regression models can provide a more robust analysis than Gaussian mixture regression models.

The following chapters present the details with appropriate examples and illustrations.

Research Structure

The remainder of this thesis is organized as below: In Chapter 2, the backgrounds for gene expression data and information criteria used in model selection are reviewed. The algorithm for the maximum likelihood estimation is addressed in Chapter 3. In Chapters 4 and 5 the behaviour of the maximum likelihood estimation for Gaussian mixture and exponential power mixture regression are investigated. In chapter 6, a new method for variable selection is developed by using LASSO and forward selection arguments. In Chapter 7, the classification of exponential power mixture regressions is discussed. The final remarks and further discussion are given in Chapter 8.

Chapter 2

Background

2.1 Gene expression data analysis

Gene expression is the process by which the information encoded in a gene is used to direct the assembly of a protein molecule. It is like books in a library, where each gene contains information to make a protein.

Gene expression data can be normally described in a table which shows the performance of each gene under certain conditions as below:

	condition 1	condition 2	condition 3	condition 4	...
gene 1					
gene 2					
gene 3		gene expressions			
gene 4					
⋮					

What is a motif? In genetics, a motif is a pattern of nucleotides in a DNA sequence or in a protein sequence which is often used to predict the underlying gene functions. There are two kinds of sequence of motifs: the first one is the sequence of motifs that appears in the exon of a gene, which may also encode the structural motifs of proteins; the second is those sequences of motifs that are outside of the gene exons, they are normally at the up-strain or the down-strain of the gene expression and regularized, which can explain why regulatory sequence motifs exit.

There are two common ways to analyse genetic data: the column-centred analysis and row-centred analysis, as detailed in the following two subsections.

2.1.1 Column-centred analysis

In column centred analysis or condition-centred analysis, the conditions are divided into several groups under the assumption that each group is homogeneous. Table for this kind of analysis can be shown as below:

	group 1		group 2		
	condition 1	condition 2	condition 3	condition 4	...
gene 1					
gene 2					
gene 3			gene expressions		
gene 4					
⋮					

The analysis of the yeast stress dataset in Chapter 5 is condition-centred. This dataset contains 496 yeast genes under 173 experimental conditions. Their gene expressions are the log-intensities of the expression of yeast under the changes of the environment.

2.1.2 Row-centred analysis

Another common way for gene expression to be handled in data analysis is through row centred analysis or gene centred analysis. In this analysis, genes are grouped into different sub-populations under the assumption that the population is heterogeneous, as demonstrated in the table below:

		condition 1	condition 2	condition 3	condition 4	...
group 1	gene 1					
	gene 2					
group 2	gene 3		gene	expressions		
	gene 4					
	⋮					

The gene and motifs data which will be use in Chapter 6 is row-centred.

2.1.3 Link gene expression to transcript factors

The key mechanism to make a cell functional is transcriptional regulation which is often regulated by proteins. A transcription factor is a group of proteins that read and interpret the genetic information in the DNA. They bind to the DNA and help initiate a program of increased or decreased gene transcription. As such, they are vital to many important cellular processes. Understanding the structure and function of a transcription factor (in particular, finding transcription factor binding sites, i.e regulatory DNA motifs) is a crucial step to studying the regulatory mechanism of gene expression. Liu et al. (2002) assume that gene expressions link the scores of regulatory motifs through a linear function. So if the gene expression is described by the response variables and the candidate motifs are the covariates in a regression format, the problem of clustering gene expression can be equal to the problem of regression clustering.

The dataset in Conlon et al. (2003) and Liu et al. (2002) contains two parts, one shows relative expressions of $4,443$ genes and the other provides the so-called motif-matching scores of $2,155$ candidate motifs to each gene. The aim of the analysis is to identify a subset of candidate motifs in the regulatory region of a gene that can explain its relative expression level.

2.1.4 Issues

The traditional way to tackle the above problem is the linear regression modelling proposed by Conlon et al. (2003), which regress the genes against candidate motifs under the assumption that the population is homogeneous. However, the assumption may not be true as genes may work in various groups (or biological pathways). This motivates us to use the so-called mixture regression model, where the population can be allowed to be heterogeneous (Khalili, 2010). Khalili (2010) fitted the Gaussian mixture of regression to the above dataset.

In the Gaussian mixture regression modelling, the conditional distribution of the response variable given covariates is assumed to be Gaussian. This may be invalid as shown in the following example, where we used Gaussian mixture regression model to cluster the dataset of gene expression and motifs mentioned before. We obtained two groups and depicted the residuals for each group by using Q-Q plot as shown in Figure 2.1. It suggests that the gene expressions in both group are not Gaussian distributed.

Here, we propose an exponential power mixture model for the above data, removing the restrictions on the distribution shape parameter. The difference between the Gaussian distribution and exponential power distribution(EPD)is shown in below figure. We expect to obtain better fit to the data compared to Gaussian mixture regression.

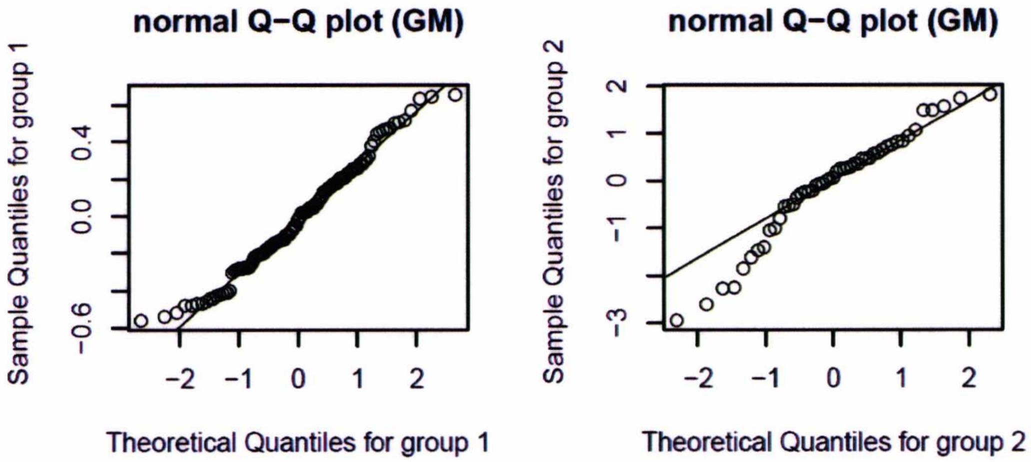

Figure 2.1: The Q-Q plot of the residuals for each GM group of the motifs dataset.

From the Figure 2.2, it can be seen that the shape of the exponential power distribution has a sharper head and heavier tails comparing its shape parameter $\alpha < 2$ compared to the Gaussian distribution's which with $\alpha = 2$, and the head of the exponential power distribution is flatter and the tail is lighter when $\alpha > 2$.

2.2 Finite mixture models

What is mixture modelling?

The mixture modelling of heterogeneity in a cluster analysis context is very useful, as it is very flexible from both the practical and theoretical perspectives. Mixture modelling is a method to model a mixture of subgroups within the population. It has been applied in many fields such as modelling gene expression microarray data, economics and engineering etc. There are two assumptions in finite mixture modelling:

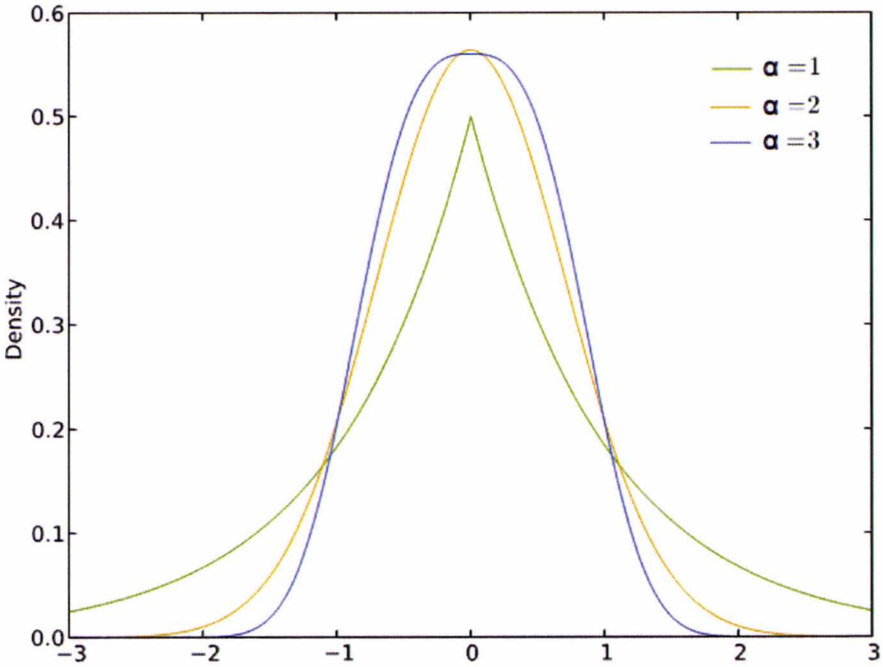

Figure 2.2: Exponential power distribution against Gaussian distributions.

- the population is made up of a finite number of homogeneous groups;
- the groups can be identified based on the similarity in the patterns of the response variables.

Parametric formulation of univariate mixture models We let $\mathbf{y} = (y_1, ..., y_n)^T$ be a random sample observed in a population, where the superscript T denotes vector transpose and y_i denotes an observed value for $i = 1, ..., n$. Let $f(y_i)$ be the probability density function of y_i, which can be expressed as

$$f(y_i) = \sum_{k=1}^{K} \pi_k f_k(y_i), \tag{2.1}$$

where the $f_k(y_i)$ are component density functions of the mixture, π_k is the mixing proportion or weight, such that $0 \leq \pi_k \leq 1$, and $\sum_{k=1}^{K} \pi_k = 1$, for $k = 1, ...K$.

For those $f_k(y_i)$ that belong to some parametric family, we write the component densities $f_k(y_i)$ as $f_k(y_i|\boldsymbol{\theta}_k)$, where $\boldsymbol{\theta}_k$ is the vector of unknown parameters for the k^{th} component density. Hence, we can rewrite the mixture density $f(y_i)$ in equation(2.1) in the form:

$$f(y_i|\Psi) = \sum_{k=1}^{K} \pi_k f_k(y_i|\boldsymbol{\theta}_k), \tag{2.2}$$

where $\Psi = (\pi_1, ..., \pi_{K-1}, \boldsymbol{\theta}_1^T, ..., \boldsymbol{\theta}_k^T)^T$ is the vector contains all the unknown parameters.

Finite mixture regression models

Regression models are used to predict one variable from one or more other variables, and finite mixture of regression models is a flexible tool for modelling data arising from many fields, such as biology, genetics, stocks etc. Mclachlan and Peel (2001) provide a review of finite mixture of regression models: When a random variable with a finite mixture distribution depends on certain covariates, we obtain a *finite mixture of regression*(FMR) model.

Considering a multiple linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{\beta}$ is a $(P+1)$ dimensional regression coefficients vector and $\mathbf{X}$ is a $n \times (P+1)$ covariates matrix. The error vector $\boldsymbol{\epsilon}$ satisfied that $E(\boldsymbol{\epsilon}) = 0$ and $Var(\boldsymbol{\epsilon}) = \sigma^2 I$.

Suppose we have a dataset $(y_i, \mathbf{x}_i)$, $i = 1, \ldots, n$, such that $E(y_i|\mathbf{x}_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_P x_{iP}$. where the i^{th} observation y_i on the response variable y depends on the i^{th} vector, $\mathbf{x}_i$, on the $n \times (P+1)$ covariates matrix, where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ and $\mathbf{x}_i = (1, x_{i1}, \ldots, x_{iP})$, $i = 1, \ldots, n$ with 1 being the design value for the intercept in the model.

In the finite mixture regression model, the parameter $\boldsymbol{\theta}_k$ in the k^{th} component density $f_k(y|\boldsymbol{\theta}_k)$ is a vector of unknown parameters. In particular, in a finite exponential power mixture regression model $\boldsymbol{\theta}_k = (\boldsymbol{\beta}_k^T, \sigma_k^2, \alpha_k)^T$, where $\boldsymbol{\beta}_k = (\beta_{k0}, \beta_{k1}, \ldots, \beta_{kP})^T$, for $k = 1, ..., K$, is the k^{th} regression coefficient vector, σ_k^2 and α_k are the dispersion and shape parameter of the k^{th} component respectively. The density in a finite exponential power mixture regression model can be written as follows:

$$f\left(y_i\left|\mathbf{x}_i, \boldsymbol{\Psi}\right.\right) = \sum_{k=1}^{K} \pi_k f_k\left(y_i\left|\mathbf{x}_i, \boldsymbol{\theta}_k\right.\right),$$

where the parameter vector $\Psi = (\pi_1, \ldots, \pi_{K-1}, \boldsymbol{\zeta}^T)^T$ is the vector containing all the unknown parameters in this mixture model, and $\boldsymbol{\zeta} = (\boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_K^T)^T$.

The density of the k^{th} component can be written as

$$f_k\left(y_i\left|\mathbf{x}_i, \boldsymbol{\theta}_k\right.\right) = \frac{\alpha_k}{2\sigma_k \Gamma\left(1/\alpha_k\right)} \exp\left(-\frac{\left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_k\right|^{\alpha_k}}{(\sigma_k^2)^{\alpha_k/2}}\right).$$

Note that, the Gaussian finite mixture regression model is a special case of the above model when setting the shape parameters $\alpha_k = 2$ for $k = 1, ..., K$.

In the following, we will provide a brief introduction to the methods for estimating finite mixture distributions. The advantages and disadvantages of each method will be discussed.

2.2.1 Estimation

Over the past hundred years, there have been various methods developed for estimating finite mixture models, including the method of moments, minimum-distance methods, maximum likelihood and Bayesian approaches.

The earliest work on estimating mixture models was based on the method of moments by Pearson (1894), who focused on two-component univariate Gaussian mixtures. Since then, the method of moments has been employed to estimating other mixture models.

The minimum distance method is another way to estimate the parameter set $\boldsymbol{\Psi}$ in a mixture model. Let the joint density function $F(\mathbf{y}|\mathbf{X}, \boldsymbol{\Psi}) = \prod_{i=1}^{n} f\left(y_i \left| \mathbf{x}_i, \boldsymbol{\Psi}\right.\right)$ under the assumption that $y_1, \ldots, y_n$ are identify and identity distributed. The minimum distance method minimizes the distance between the mixture density function $F(\mathbf{y}|\mathbf{X}, \boldsymbol{\Psi})$ and the estimated mixture density function $F(\mathbf{y}|\mathbf{X}, \hat{\boldsymbol{\Psi}})$ for $\mathbf{y} = (y_1, ..., y_n)^T$. The comprehensive properties for minimum distance methods were investigated by Titterington et al. (1985).

As a result of the invention of the high speed computer, many new methods have been developed recently. There are mainly two categories for these methods, namely, likelihood-based methods and Bayesian methods. Rao (1948) was the first to use the maximum likelihood to estimating mixture models. He fitted a mixture of two univariate distributions with equal variances to the data by using Fisher's method of scoring. However, his method has not been pursued further until Hasselblad (1966) addressed a few computational issues related to his method. Dempster et al. (1977) formulated the problem into an missing data framework and developed the expectation-maximization(EM) algorithm. After that, there have been extensive studies along this direction and the EM has become one of the most popular and efficient approaches to mixture model estimation.

The Bayesian approach to mixture modelling is based on posterior simulation via the Markov chain Monte Carlo(MCMC)approach. The first study that considered Bayesian estimation by using posterior simulation for mixture models was Gilks et al. (1989). The use of Bayesian methods for estimation was limited until Gelfand and Smith (1990) identified that most Bayesian computations can be done by using the Gibbs sampler. Recently, some case studies had also considered this method, such as (Dellaportas, 1998) and (Vounatsou et al., 1998), and there are also many references to Bayesian mixture analysis provided by Mclachlan and Peel (2001).

There are quite a lot of literatures on how to fit a finite mixture model. Here, we focus on the EM algorithm as the fitting of mixture models by ML is a classic example of the so-called incomplete data problem. A brief introduction will be given to the EM algorithm in Chapter 3, starting with the general setting of the arbitrary component distributions, which will be followed by how to apply the algorithm to particular cases. The EM algorithm will then be employed to estimate a Gaussian mixture distribution with two components based on twenty observations. The results will demonstrate that the EM algorithm can provide good estimative except for some scenarios in which the population means are quite close to each other or there are big differences in the variances of the component distributions.

2.2.2 Evaluating model fit and complexity

In some of the complicated regression models, such as the finite mixture model of regression, a large number of covariates are normally involved. Some of them may be correlated when making predictions of the response variable and some may not. Therefore, it is important to select a model which contains only the relative variables. However, as we do not know which covariates can be used to predict the response variable, we need methodologies to help us choose the "best" subset of covariates for the prediction.

What we need to be aware of is that there is no such an methodology that can provide a perfect model for the data. Due to the principle of parsimony, variable selection as one approach of model selection intends to make a trade-off between bias and variance. The more parameters selected in a model, the less bias and the larger variance that may exist, and vice versa. Therefore, the balance between under fitting and over fitting must be taken in into consideration in model selection.

There are many methods of model selection in the literature. Two issues are addressed as relevant to these methods. One is how do we define an appropriate selection criterion and the other is how do we implement the corresponding selection procedure? We will explore and discuss these issues in the next sections. The methodologies are described addressing the first issue in Section 2.3and the procedures for searching the "best" model are provided in Section 2.4.

2.3 Information criteria for model selection

The question of how many components should be included in a mixture model is difficult to be completely resolved. As far as we know, there are two approaches to modelling data with finite mixture distributions. The first is to modelling unknown distributional shapes by providing an appealing semi-parametric framework, such as the kernel density method; the second is to provide model based clustering by using the mixture model. In both cases, the number of components K in a mixture model needs to be known, which is called the problem of model order selection. For each component, we need to select the covariants for the regression equations, which is called the problem of feature selection. Therefore, we split the model selection problems into two sub-problems, namely, the problems of model order selection and feature selection.

Both empirical and theoretical approaches are considered. Empirical approaches include bootstrap and cross-validation, and the theoretical approaches include Kullback-Leibler information, Akaike information criterion(AIC), Bayesian information criterion(BIC) and others. AIC and BIC are the most commonly used approaches for model selection. The famous Japanese statistician Akaike (1974) proposed the Akaike information criteria(AIC) based on Kullback-Leibler information to justify the goodness of a selected model. The goal of AIC is to find the model that gives the best prediction without considering the correctness of the model. The AIC penalizes the number of parameters less strongly than the Bayesian information criterion (BIC), which was developed by Schwarz (1978), using Bayesian formalism, and the nature of BIC is different from AIC as it assumes one of the models is the true model.

2.3.1 AIC

Assume we have a data set $\mathbf{y} = (y_1, ..., y_n)^T$ which is drawn from some density f. Hence $f(\mathbf{y}|\mathbf{X})$ denotes the true density while $f_m(\mathbf{y}|\mathbf{X}, \hat{\mathbf{\Psi}}_m)$ denotes an estimator of f where m denotes the m^{th} model and $\hat{\mathbf{\Psi}}_m$ are the maximum likelihood estimators for model m. In order to measure the divergence of $f(\mathbf{y}|\mathbf{X})$ with respect to $f_m(\mathbf{y}|\mathbf{X}, \hat{\mathbf{\Psi}}_m)$, the Kullback-Leibler information is used:

$$\begin{aligned} K\{f(\mathbf{y}|\mathbf{X}), f_m(\mathbf{y}|\mathbf{X}, \hat{\mathbf{\Psi}}_m)\} &= \int f(\mathbf{y}|\mathbf{X}) \log \frac{f(\mathbf{y}|\mathbf{X})}{f_m(\mathbf{y}|\mathbf{X}, \hat{\mathbf{\Psi}}_m)} d\mathbf{y} \\ &= \int f(\mathbf{y}|\mathbf{X}) \log f(\mathbf{y}|\mathbf{X}) d\mathbf{y} \\ &\quad - \int f(\mathbf{y}|\mathbf{X}) \log f_m(\mathbf{y}|\mathbf{X}, \hat{\mathbf{\Psi}}_m) d\mathbf{y}. \qquad (2.3) \end{aligned}$$

The best fit model will give small Kullback-Leibler information, hence the aim is to minimise the result of equation 2.3. As the first term of the above equation does not depend on m, only the second term is related. Therefore, minimizing the Kullback-Leibler distance $K\{f(\mathbf{y}|\mathbf{X}), f_m(\mathbf{y}|\mathbf{X}, \hat{\mathbf{\Psi}}_m)\}$ is the same as maximizing

$$K_m = \int f(\mathbf{y}|\mathbf{X}) \log f_m(\mathbf{y}|\mathbf{X}, \hat{\mathbf{\Psi}}_m) d\mathbf{y}.$$

A simple way to estimate K_m is given by

$$\begin{aligned} \hat{K}_m &= \frac{1}{n} \sum_{i=1}^{n} \log f_m(\mathbf{y}_i|\mathbf{X}_i, \hat{\mathbf{\Psi}}_m) \\ &= \frac{1}{n} \log L(\hat{\mathbf{\Psi}}_m|\mathbf{y}, \mathbf{X}) \end{aligned}$$

where $L(\hat{\mathbf{\Psi}}_m|\mathbf{y}, \mathbf{X})$ is the likelihood function for model m. But this gives an overestimate of the expected log density, the bias is denoted as $b(K_m)$. Hence, an information criterion for model selection can be obtained by using bias-corrected log likelihood function:

$$\log L(\hat{\mathbf{\Psi}}_m|\mathbf{y}, \mathbf{X}) - \mathbf{b}(\mathbf{K_m}), \tag{2.4}$$

where the aim of the selected model is to maximize equation 2.5 and minimize the Kullback-Leibler information 2.3.

The general expression of equation(2.5) is in the form

$$-2\log L(\hat{\mathbf{\Psi}}_m|\mathbf{y}, \mathbf{X}) + \mathbf{2C}, \tag{2.5}$$

where $-2\log L(\hat{\mathbf{\Psi}}_m|\mathbf{y}, \mathbf{X})$ measures the goodness of fit and C is the penalty that shows the complexity of the model. The aim is to choose a model to minimize equation(2.5).

Akaike (1974) showed that $b(K_m)$ is equal to the total number of parameters in the selected model m, we denote it as d_m. Thus, Akaike's information criterion(AIC) of model m is defined as:

$$\mathrm{AIC}_m = -2\log L(\hat{\mathbf{\Psi}}_m|\mathbf{y},\mathbf{X}) + \mathbf{2d_m}. \tag{2.6}$$

In order to avoid over fitting, AIC penalizes -2 log likelihood, $-2\log L(\hat{\mathbf{\Psi}}|\mathbf{y},\mathbf{X})$, by adding twice the number of estimated parameters. For the same outcome variables, AIC selects the "best" model with the lowest value.

There are also some limitations to the AIC, such as those models which are related to time series analysis when it is not a consistent estimator of the order of an auto-regression. When the number of observations goes to infinity, the order chosen by AIC is not reliable as it's too large compared to the true order.

2.3.2 BIC

Both Akaike (1974) and Schwarz (1978) proved AIC by developed a Bayesian information criterion(BIC) for model selection. BIC has a heavier penalty for over-fitting compared with AIC when large sample sizes are applied.

Under the assumption that the observations are independent, Bayesian information criterion for selected model m is defined as:

$$\mathrm{BIC}_m = -2\log L(\hat{\mathbf{\Psi}}_m|\mathbf{y},\mathbf{X}) + \mathbf{d_m}\log(\mathbf{n}), \tag{2.7}$$

where n is the number of observations. BIC is much preferred when the number of observations is large, and even for the situation when n is not very large, BIC still should be preferred as it provides a heavier penalty than AIC does, as $\log(n) > 2$ when $n > 8$. It can be derived by using for Bayesian model evidence, the log-posterior problem of the model as follows:

Derivation The model selection through the Bayesian approach is to maximize the posterior probability of the selected model with the given dataset. The formula for calculating the posterior probability of model m is:

$$f(\text{model m}|\mathbf{y}, \mathbf{X}) = \frac{f(\mathbf{y}|\mathbf{X}, \text{model m})f(\text{model m})}{f(\mathbf{y})},$$

where $f(\mathbf{y}|\mathbf{X}, \text{model m})$ is the marginal likelihood of model m and $f(\text{model m})$ is the prior probability.

BIC assumes that the probability of each selected model is the same, therefore to maximize the posterior probability is the same as maximising the marginal likelihood:

$$\begin{aligned} f(\mathbf{y}|\mathbf{X}, \text{model m}) &= \int f(\mathbf{y}|\mathbf{X}, \text{model m}, \boldsymbol{\Psi}_m)\, d\boldsymbol{\Psi}_m \\ &= \int L(\Psi_m|\mathbf{y}, \mathbf{X})\, d\boldsymbol{\Psi}_m. \end{aligned}$$

Hence, to maximising the posterior probability is equivalent to maximising

$$\log \int L(\Psi_m|\mathbf{y}, \mathbf{X})\, d\boldsymbol{\Psi}_m$$

By Taylor series and the law of large number, we can show that:

$$\log \int L(\Psi_m|\mathbf{y}, \mathbf{X})\, d\boldsymbol{\Psi}_m \approx \log L(\hat{\boldsymbol{\Psi}}_m|\mathbf{y}, \mathbf{X}) - \frac{\mathbf{d_m}}{\mathbf{2}} \log(\mathbf{n}). \tag{2.8}$$

By times -2 on the left hand of equation(2.8), we have the BIC for a selected model m as:

$$\text{BIC}_m = -2\log L(\hat{\boldsymbol{\Psi}}_m|\mathbf{y}, \mathbf{X}) + \mathbf{d_m}\log(\mathbf{n}), \tag{2.9}$$

where d_m is the number of free parameters to be estimated. The smaller the

BIC the better the model fit. So we choose the model which minimizes the BIC.

The BIC (Schwarz, 1978) provides a good balance between the log-likelihood and the number of free parameters. For the GM model, the number of parameters of π is $K-1$, σ is K and $\boldsymbol{\beta}$ is $(p+1)\times K$, by letting shape parameters equal to 2, it gives

$$\text{BIC}_m = -2\log L(\hat{\boldsymbol{\Psi}}_m|\mathbf{y},\mathbf{X}) + (2K-1+(p+1)\times K)\log(n).$$

For the EPD mixture model, the number of parameters of π is $K-1$, σ is K, $\boldsymbol{\beta}$ is $(P+1)\times K$ and α is K, leading to

$$\text{BIC}_m = -2\log L(\hat{\boldsymbol{\Psi}}_m|\mathbf{y},\mathbf{X}) + (3K-1+(P+1)\times K)\log(n).$$

2.3.3 AIC vs. BIC

Both the AIC and BIC are penalized-likelihood criteria used for model selection, they are used to choose the "best" predictor subsets and to compare non-nested models. AIC is the relative distance between the unknown true likelihood function of the data and the fitted likelihood function of the model plus an estimated bias. The lower the AIC value, the closer the fitted model is to the truth. BIC is to estimate the function of the posterior probability to evaluate a model is true or not, thus, the smaller the minimum BIC's value is, the more likely the fitted model is the true model. Hence we can see that AIC tries to find the model with the best prediction while BIC tries to find the model that is most likely to be true under the assumption that one of the models is true.

The theory of AIC and BIC are completely different, but in practise, the only difference is the penalty. The penalty of BIC is larger than the penalty of AIC when the number of observation is large. Hence, BIC is much preferred in complex models. For a model with a large sample size, the AIC may face the risk of choosing a model with too many parameters because its penalty is not related to the sample size, while BIC may face the risk of choosing too few parameters when the sample size is very large.

For some large data, the model selection by using AIC or BIC directly can be extremely computationally intensive, hence how to efficiently select a model for large dataset is a problem. In the next part the methods of feature selection are considered and focused on the Lasso method.

2.3.4 EBIC

There are two goals of model selection: one is to select the "best" model to undertake a prediction and then to focus on the accuracy of the prediction. The other is to identify the selected features and focus on the consistency of the selection. The methods of model selection such as AIC, cross-validation (CV), generalized cross-validation (GCV) are based on the predicted accuracy of the selected model, while the BIC assumes the prior is uniform over all models. If the number of features is not large and all the features are fixed, we do not have a feature selection problem. In such situation, the above criterion work well for the prediction accuracy which is our first goal, and it does not conflict with feature selection. But when the dimension $(P+1)$ is more huge compared to a moderate sample size n, the criterion such as AIC, CV, GCV and BIC, etc. are too liberal as they tend to choose too many features.

Chen and Chen (2008) proposed an extended family of BIC, which considered both the number of unknown parameters and the complexity of the model space. Chen and Chen (2008) also showed that for a large dataset, the extended Bayesian information criterion(EBIC) had very small loss in the positive selection rate while it was tightly central to the false discovery rate.

Assume we have a finite mixture regression model with large dimension $(P+1)$ and relatively small sample of size n, such that:

$$f\left(y_i\left|\mathbf{x}_i, \boldsymbol{\Psi}\right.\right) = \sum_{k=1}^{K} \pi_k f_k\left(y_i\left|\mathbf{x}_i, \boldsymbol{\theta}_k\right.\right),$$

where the response variable y_i for the i^{th} entity depends on $\mathbf{x}_i$ where $\mathbf{x}_i$ is a vector of covariates $\mathbf{X}$, for $i = 1, ..., n$. The observations on covariates can be written as an n by $(P+1)$ matrix such that $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$. The regression coefficient $\boldsymbol{\beta}$ is a $(P+1)$ by K matrix, here $\boldsymbol{\beta}_k$ is sparse which means there are only few elements that are non-zero. Each $\boldsymbol{\beta}_k$ is a $(P+1)$ dimensional column vector, for $k = 1, ..., K$.

Now we let s be a subset of $(1, \ldots, (P+1))$. Denote $\hat{\boldsymbol{\Psi}}(s)$ with those parameters outside s equals to zero. Let S_j be the set of all combinations of j indices in $(1, \ldots, (P+1))$. The prior/probability on S_j is inversely proportional to the size of S_j, $\kappa(S_j)$, where $\kappa(S_j) = \begin{pmatrix} P+1 \\ j \end{pmatrix}$. For each s in the same subspace S_j, assign an eaual probability, i.e. $pr(s|S_j) = \frac{1}{\kappa(S_j)}$ for any $s \in S_j$ as all the models in S_j are equallly plausible. We assign the probability $pr(S_j)$ propotional to $\kappa^{\xi}(S_j)$ in EBIC, $0 \leq \xi \leq 1$, instead assign the probability $pr(S_j)$ propotional to $\log(n)$ in BIC. Therefor the prior $p(s)$ is propotional to $\kappa^{-\gamma}(S_j)$ for $\gamma = 1 - \xi$. Hence, the family of extended BIC of a model m is defined as follows:

$$\mathrm{EBIC}_m(s) = -2\log L(\hat{\boldsymbol{\Psi}}_m(s)|\mathbf{y}, \mathbf{X}) + \mathbf{d_m(s)}\log(\mathbf{n}) + \mathbf{2fl}\log\,\breve{}\,(\mathbf{S_j}), \qquad (2.10)$$

for $0 \leq \gamma \leq 1$. Where $\hat{\boldsymbol{\Psi}}_m(s)$ is the maximum likelihood estimator of $\boldsymbol{\Psi}_m(s)$, $d_m(s)$ is the number of parameters in s for selected model m. In this study, we consider $j = 2$, thus $\kappa(S_j) = \begin{pmatrix} P+1 \\ 2 \end{pmatrix} = \frac{(P+1)P}{2}$.

2.3.5 BIC vs. EBIC

Recall that

$$\mathrm{BIC}_m = -2\log L(\hat{\boldsymbol{\Psi}}_m|\mathbf{y},\mathbf{X}) + \mathbf{d_m}\log(\mathbf{n}),$$

$$\mathrm{EBIC}_m(s) = -2\log L(\hat{\boldsymbol{\Psi}}_m(s)|\mathbf{y},\mathbf{X}) + \mathbf{d_m}(\mathbf{s})\log(\mathbf{n}) + \mathbf{2fl}\log\check{}\,(\mathbf{S_j}).$$

For a high dimension dataset, there are two issues related to BIC: the first one is that it penalizes too much with $d_m \log(n)$; the other one is it penalizes too little with the prior.

Unlike the BIC which selects an equal prior probability for each model, the EBIC provides a different prior probability for the model in different submodel classes. So instead of assigning the prior probability of S_j which is proportional to $\kappa(S_j)$ in ordinary BIC, the EBIC assigns the prior probability of S_j which is proportional to $\kappa(S_j)^\xi$, for $0 < \xi < 1$.

2.4 Regression shrinkage and selection

As a result of the rapid improvement of scientific technology in recent decades, large data from various fields are now widely collected by scientists. As a result, model selection which was developed to solve the problem of how to estimate those large data(in N,P or both) become an extremely important part of statistical modelling. Some limitations can easily be identified in the traditional methods in which stepwise regression with AIC and BIC criteria for the choice of the optimal model were commonly used. Tibshirani (1996) proposed a new model selection method called Lasso which overcame the limitations. Efron et al. (2004) proposed that an effective algorithm, LARS, to solve Lasso. In this section a brief review of the basic idea and history of Lasso and LARS will be given based on the original paper on regression shrinkage and selection via Lasso (Tibshirani, 1996) and lease angle regression (Efron et al., 2004).

2.4.1 Background

At the very beginning of building a model, as many independent variables as possible are chosen to avoid bias that may exist in the models which caused by lacking of important variables. But actually what we really need is to identify the most related variables of the observed variable, i.e. variable selection(or model selection). Therefore, variable selection is a very important step in the process of building a model.

Bradley Efron, a professor at Stanford University who proposed bootstrapping, said that the most important problem in the modern statistics field is variable selection. However the problem remained in research on the AIC as the information criterion becomes incapable of action as a result of too much computation when the number of model variables is too large, and the method is incapable of action for higher dimension model selection problems despite there are many guidelines to improve the rules, such as BIC etc. Stepwise regression combined with AIC and BIC criterion for optimal model selection has generally been used to solve classification and regression problems. It has been proved that the practicability of this method was acceptable. But there were still many problems with this traditional method: the research of Breiman (1995) pointed out that using this method to chose a model was very unstable. Fan and Li (2001) pointed out that random errors existed in the calculation process of this method, and it was also difficult to study its theoretical properties, and for classification or regression problems of larger datasets, a large amount of calculations were always required.
In general the model selection should meet the following requirements:
(1) High accurate prediction;
(2) Scientific significance of the selected variables;

(3) High stability of the model;
(4) should avoid the partial in hypothesis tests;
(5) Low computational complexity.

But only some of the above requirements can be achieved by using approaches such as stepwise regression, the traditional optimal subset selection, ridge regression, principal component regression and partial least squares. Therefore, how to effectively tackle these problems to achieve the goal of statistical modelling has become one of the hot topics in statistical research. The proposal of the Lasso method with its effective algorithm undoubtedly provides a feasible solution to these problems. A brief introduction of Lasso is given as below.

2.4.2 Lasso and LARS

Breiman (1995) proposed a new method of model selection based on the idea of penalized least squares, called "Non-negative Garrotte". Later on Tibshirani (1996) inspired a new variable selection method—Least absolute shrinkage and selection operator, so called Lasso, based both on the "Bridge Regression" which was proposed by Frank and Friedman (1993) and "Non-negative Garrotte" which was proposed by Breiman (1995). The Lasso method used the function of the absolute value of coefficients of the model as a penalty to shrink the model coefficients; the small absolute value of the coefficient of the model automatic shrunk to zero. By doing so, a little bias was sacrificed in order to reduce the variance of the predicted values and may improve the accuracy of the overall prediction. Compared with the traditional methods of model selection, the Lasso method did better to overcome the short comings of the traditional methods which meant that it received great attention in the field of statistics. In order to solve the drawback of lacking an effective algorithm in this method, lots of research were undertaken: first, Fu (1998) putted forward the "Shooting" algorithm, then Osborne et al. (2000) proposed the corresponding homotopy algorithm after he found that the solution path of Lasso regression was piecewise linear. Although Lasso regression problem were better solved by using these algorithms compared to originally used off-the-shelf quadratic program solvers, its effectiveness was still unable to meet the requirements until Efron et al. (2004) proposed the Least Angle Regression(LARS) algorithm to solve the calculation problems of Lasso which made the Lasso method more popular and more widely used.

2.4.3 Lasso and its other related methods

We denote the coefficients of a model by $\boldsymbol{\beta}$, which corresponds to the loss function $l(\boldsymbol{\beta})$, here we use the log-likelihood function. Let $\boldsymbol{\beta}$ be a $(P+1)$ dimensional vector, then the penalized likelihood function of parameters:

$$l(\boldsymbol{\beta}) + \sum_{j=1}^{P} P_{\lambda_j}(|\boldsymbol{\beta}_j|).$$

when $l(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^2$, $P_{\lambda_j}(|\beta_j|) = \lambda|\beta_j|^q$ this becomes the "Bridge Regression" by Frank and Friedman (1993). When $q = 1$, this is the Lasso regression which is also called a L_1 regularization. In fact, when $q = 2$, this is the ridge regression which is also called a L_2 regularization.

Considering a multiple linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where the response variables $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$, the predictor matrix of $\mathbf{y}$ is $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)^T$. For $i = 1, 2, \ldots, n$, we have $\mathbf{x}_i = (1, x_{i1}, x_{i2}, ..., x_{iP})^T$, $\boldsymbol{\beta}$ is a $(P+1)$ by K matrix where $\boldsymbol{\beta}_k$ is a $(P+1)$ dimensional column vector, for $k = 1, ..., K$. The error vector $\boldsymbol{\epsilon}$ satisfied that $E(\boldsymbol{\epsilon}) = 0$ and $Var(\boldsymbol{\epsilon}) = \sigma^2 I$. We also assume: $E(y_i|\mathbf{x}_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_P x_{iP}$. Be aware that this is a sparse model i.e there are some coefficients equal to 0 in $\beta_0, \beta_1, \beta_2, \dots, \beta_P$, the purpose of model selection (or variable selection, feature selection) is to identify those coefficients equal to 0, and estimate the other non-zero parameters according to the acquired data, namely finding the sparse model.

For the linear model, the model selection can be expressed as the following optimization problems:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda |\boldsymbol{\beta}| \right\} \tag{2.11}$$

where $|\boldsymbol{\beta}|$ is the submission of full absolute values of least square estimates, i.e $|\boldsymbol{\beta}| = \sum_{j=1}^{P} |\boldsymbol{\beta}_j|$.

There are two processes of the above function: find the coefficients for significant variables and estimate those corresponding coefficients. These two processes are carried out separately when treated with traditional methods. However in actual processing they often have difficulties because there are not any other restrictions on the parameter space and the two processes of Lasso and associated method are carried out simultaneously. Lasso is actually equivalent to considering the following issues:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left[\frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \right] \text{ subject to } \sum_{j=1}^{P} |\boldsymbol{\beta}_j| \leq t.$$

The above inequality equation effectively restricts the parameter space.

Let

$$f(\boldsymbol{\beta}) = \frac{1}{2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda |\boldsymbol{\beta}|_0, \tag{2.12}$$

in order to minimize $f(\boldsymbol{\beta})$ for the t^{th} component, assume the predict variables are all relative to $\mathbf{y}$, i.e. $\beta_{tj} \neq 0$ for $j = (1, \dots, P)$ and $1 \leq t \leq K$, we differentiate the function with respect to $\boldsymbol{\beta}_t$ to obtain:

$$\frac{\partial f(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_t} = \sum_{i=1}^{n} -x_{it}(y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t) + \lambda \text{sign}(\boldsymbol{\beta}_t).$$

Let the above derivative equal to zero, we have

$$\sum_{i=1}^{n} x_{ti}(y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t) = \lambda \text{sign}(\boldsymbol{\beta}_t)$$

where sign$(\boldsymbol{\beta}_t)$ denotes the signal of $\boldsymbol{\beta}_t$.

If the variable are sparse, i.e there are some variables not related to the observations, this implies we have some $\beta_{tj} = 0$ such that $\beta_{tj} = 0$,for $j = 1, ..., P$ and $t = 1, ..., K$, where the function is not differentiable at the point, then by using the Karush-Kuhn-Tucker(KKT) theory, we have

$$\sum_{i=1}^{n} x_{tj}(y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t) \in [-\lambda, \lambda].$$

The KKT conditions for optimising $f(\boldsymbol{\beta})$ for the t^{th} component can be written as:

$$\sum_{i=1}^{n} x_{it}(y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t) = \lambda \gamma_t, \tag{2.13}$$

such that, $\gamma_t = \text{sign}(\boldsymbol{\beta}_{tj})$ if $\boldsymbol{\beta}_{tj} \neq 0$, $\gamma_t = [-1, 1]$ if $\boldsymbol{\beta}_{tj} = 0$, for $t = 1, \ldots, K$ and $j = 1, \ldots, P$. Therefore, $\boldsymbol{\beta}_t$ is a solution of the t^{th} component of $\boldsymbol{\beta}$ in the equation(2.11) if and only if it satisfied equation(2.13).

However this does not scale well and is not transparent. Then the student of Tibshirani, Fu (1998) proposed a more efficient algorithm according to the "bridge regression". The current popular approach is the least angle regression algorithm proposed by Efron et al. (2004). It is also efficient to solve the Lasso problem and connects the Lasso to forward stagewise regression.

2.5 Rand index

Mixture models can be used to partition data points into meaningful groups. The methods to measure the accuracy of clustering and to compare differences in clustering are very important issues in clustering. The Rand index(short for statistician W.M.Rand (Rand, 1971)) is a famous criterion for clustering comparisons, which gives the degree of agreement for two partitions.

Suppose we have a set of n elements $S = \{o_1, ..., o_n\}$, let $U = \{u_1, ..., u_R\}$ and $V = \{v_1, ..., v_C\}$ are two partitions with R subsets and C subsets respectively. Let a be the number of pairs of elements that are in the same sets in U and in the same sets in V, b be the number of pairs of elements that are in the different sets in U and in the different sets in V, c be the number of pairs of elements that are in the same sets in U but in the different sets in V, and d be the number of pairs of elements that are in the different sets in U but in the same sets in V. For example, let $C = (1,2,3), (4,5,6)$ and $C' = (1,2), (3,4,5), (6)$ are two sets. In this case, there are 2 paires of elements that are in both sets, i.e. pair $(1,2)$ and $(4,5)$; there are 7 pairs of elements that are seperate in both sets, i.e. pair $(1,4), (1,5), (1,6), (2,4), (2,5), (2,6)$ and $(3,6)$; there are 2 pairs of elements that are in set C but not in set C', i.e. pair $(1,3)$ and $(2,3)$; there are 4 pairs of elements that are in set C' but not in set C, i.e. $(3,4), (3,5), (4,6)$ and $(5,6)$. Hence we have $a = 2$, $b = 6$, $c = 2$, and $d = 4$ in this example.

We can see that a and b represent the agreement degree and c and d represent disagreement degree. Hence, the Rand index, R, can be written as:

$$R = \frac{a+b}{a+b+c+d}. \tag{2.14}$$

The range of the Rand index is between 0 and 1. If the Rand index equals to 1 this means the two data clusters U and V perfectly agree. On the other hand, if the Rand index equals to 0 this means the two data clusters do not agree on any pair of elements, i.e. $a = b = 0$. The value of the Rand index being equal to 0 is an extreme case, but it is desirable for the similarity index to take a constant value or to take a value close to 0. The issue of the Rand index is that its expected value between two random partitions is not constant value.

The adjusted Rand index was proposed by (Hubert and Arabie, 1985) by taking the generalized hyper-geometric distribution as the model of randomness, i.e the two partitions are picked randomly such that both the number of cluster and the elements in each cluster are fixed under the assumption that the number of two clusters in the two clusterings must be same.

Let n_{ij} be the number of elements that are in both set u_i and v_j, where $n_{i.}$ represents the total number of elements in set u_i and $n_{.j}$ represents the total number of elements in set v_j. These notations are shown in Table 2.1.

Table 2.1: Table of notations

U / V	v_1	v_2	$\dots$	v_C	sum of u
u_1	n_{11}	n_{12}	$\dots$	n_{1C}	$n_{1.}$
u_2	n_{21}	n_{22}	$\dots$	n_{2C}	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
u_R	n_{R1}	n_{R2}	$\dots$	n_{RC}	$n_{R.}$
sum of v	$n_{.1}$	$n_{.2}$	$\dots$	$n_{.C}$	n

The adjusted Rand index is a corrected-for-change version of the Rand index. The general form of the adjusted Rand index, ARI, is defined as:

$$\text{ARI} = \frac{\text{Index-Expected Index}}{\text{MaxIndex-Expected Index}}, \tag{2.15}$$

where the expected index under the generalized hyper-geometric model was shown by (Hubert and Arabie, 1985) to be:

$$E\left[\sum_{ij}\binom{n_{ij}}{2}\right]=\left[\sum_{i}\binom{n_{i.}}{2}\sum_{j}\binom{n_{.j}}{2}\right]\Big/\binom{n}{2}$$

Thus, the adjusted Rand index of equation 2.15 can be rewritten in the following form:

$$\text{ARI}=\frac{\sum_{ij}\binom{n_{ij}}{2}-\left[\sum_{i}\binom{n_{i.}}{2}\sum_{j}\binom{n_{.j}}{2}\right]\Big/\binom{n}{2}}{\frac{1}{2}\left[\sum_{i}\binom{n_{i.}}{2}+\sum_{j}\binom{n_{.j}}{2}\right]-\left[\sum_{i}\binom{n_{i.}}{2}\sum_{j}\binom{n_{.j}}{2}\right]\Big/\binom{n}{2}}. \quad (2.16)$$

As mentioned above, the range of the Rand index is between 0 and 1, hence the expectation of the Rand index is greater or equal to 0. The upper bound of the adjusted Rand index is 1 and its expectation is 0, and the wider range gives a more sensitive index.

In our next simulations, by given the minimum BIC of the mixture model of regression $\sum_{k=1}^{\widehat{K}} \widehat{\pi}_i\phi(y_i|\mathbf{x}_i,\widehat{\boldsymbol{\theta}}_k)$, we estimated τ_{ij} by let $\tau_{ij}=\frac{\widehat{\pi_k} f_k(y_i|\mathbf{x}_i,\widehat{\boldsymbol{\theta}_k})}{\sum_{t=1}^{\widehat{K}}\widehat{\pi_k} g(y_i|\mathbf{x}_i,\widehat{\boldsymbol{\theta}}_t)}$. Therefore if $\tau_{ij}=\max_{1\le t\le \widehat{K}}\tau_{tj}$, we assign the i^{th} observation to the k^{th} cluster C_k. This leads to the partition $\widehat{C}$ of y_i. Here we use the adjusted RAND index ρ of (Hubert and Arabie, 1985) to assess the level of agreement between $\widehat{C}$ and another partition C. The larger the value of ρ, the higher the level of agreement the two partitions have. The maximum value of ρ equals to 1 when the two partitions are identical. In our simulation section, we use the RAND index directly to assess the quality of a clustering based on the true partition C is known.

Chapter 3

Estimation in finite mixture regression models

In this Chapter, we first define the concept of maximum likelihood (ML) estimation for a general mixture regression model $f(y|\mathbf{x}, \mathbf{\Psi})$, where $\mathbf{\Psi}$ is a parameter. Then, we introduce two approaches for calculating the ML estimator: a Newton-Raphson iteration-based direct approach and the Expectation-Maximisation algorithm-based indirect approach.Then we implement these algorithms for estimating Gaussian mixture regression and exponential mixture regression respectively followed by illustrate these algorithms by simulations and real data analyses.

3.1 Maximum Likelihood Estimation

Suppose that we have an independent sample $(y_i, \mathbf{x}_i)$, $1 \leq i \leq n$, where conditional on $\mathbf{x}_i$, y_i drawn from the regression density $f(y|\mathbf{x}, \mathbf{\Psi})$ with parameter $\mathbf{\Psi}$. The likelihood function of $\mathbf{\Psi}$ can be written as follows:

$$L(\mathbf{\Psi}|\mathbf{y}, \mathbf{X}) = \prod_{i=1}^{n} f(y_i|\mathbf{x}_i, \mathbf{\Psi}),$$

where $\mathbf{y} = (y_1, \ldots, y_n)^T$ and $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$. The log-likelihood can be written in the following form:

$$\log L(\mathbf{\Psi}|\mathbf{y}, \mathbf{X}) = \sum_{i=1}^{n} \log f(y_i|\mathbf{x_i}, \mathbf{\Psi}). \tag{3.1}$$

When the parameter $\mathbf{\Psi}$ is unconstrained, its ML estimator $\hat{\mathbf{\Psi}}$ can be obtained by solving the score function:

$$S(\mathbf{y}|\mathbf{X}, \mathbf{\Psi}) = \frac{\partial \log \mathbf{L}(\mathbf{\Psi}|\mathbf{y}, \mathbf{X})}{\partial \mathbf{\Psi}} = \mathbf{0}. \tag{3.2}$$

When the parameter $\mathbf{\Psi}$ is constrained, we need to replace the above log-likelihood function by an appropriate Lagrange multiplier.

In the following subsections, how to calculate the maximum likelihood estimator of $\mathbf{\Psi}$ by the direct and indirect approaches is discussed.

3.2 Direct Approach

In this subsection, we derive a set of equations for estimating $\mathbf{\Psi}$ in the model

$$f(y_i|\mathbf{x}_i, \mathbf{\Psi}) = \sum_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k),$$

where $\mathbf{y} = (y_1, \ldots, y_n)^T$ are observations on the response variable y, $\mathbf{X} = (\mathbf{x_1}, \ldots, \mathbf{x_n})^{\mathbf{T}}$ are observations on the covariate $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_P})^{\mathbf{T}}$, and the parameter $\boldsymbol{\zeta} = (\boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_K^T)^T$, $\mathbf{\Psi} = (\pi_1, \ldots, \pi_{K-1}, \boldsymbol{\zeta}^T)^T$ is a vector containing all the unknown parameters in the above model, with a constraint $\sum\limits_{k=1}^{K} \pi_k = 1$.

We have the log likelihood function

$$\log L(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X}) \;=\; \sum_{i=1}^{n} \log f(y_i|\mathbf{x}_i,\boldsymbol{\Psi}) = \sum_{i=1}^{n} \log\left\{\sum_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i,\boldsymbol{\theta}_k)\right\} \quad (3.3)$$

Note that π_k, $k = 1,\ldots,n$ are constrained. We need to construct the following Lagrange multiplier $\log L(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X}) - \lambda(\sum_{k=1}^{K}\pi_k - 1)$. In order to estimate the parameters of the t^{th} component, we differentiating the equation (3.3) with respect to $\boldsymbol{\Psi}_t$ and setting it to zero, we have

$$\frac{\partial \log L(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X})}{\partial \boldsymbol{\theta}_t} = 0, \qquad t = 1,\ldots,K. \quad (3.4)$$

and

$$\frac{\partial[\log L(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X}) - \lambda(\sum_{k=1}^{K}\pi_k - 1)]}{\partial \pi_t} = 0, \qquad t = 1,\ldots,K. \quad (3.5)$$

In the following, we further solve the equations (3.4) and (3.5) respectively: Regarding equation (3.4), we note that:

$$\begin{aligned}
\log L(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X}) &= \sum_{i=1}^{n} \log f(y_i|\mathbf{x}_i,\boldsymbol{\Psi}) \\
&= \sum_{i=1}^{n} \log\left\{\sum_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i,\boldsymbol{\theta}_k)\right\} \\
&= \sum_{i=1}^{n} \log\left\{\pi_1 f_1(y_i|\mathbf{x}_i,\boldsymbol{\theta}_1) + \cdots + \pi_K f_K(y_i|\mathbf{x}_i,\boldsymbol{\theta}_K)\right\}.
\end{aligned}$$

So we can rewrite equation (3.4) in the following form:

$$
\begin{aligned}
\frac{\partial \log L(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X})}{\partial \boldsymbol{\theta}_t} &= \frac{\partial(\sum\limits_{i=1}^{n} \log\{\sum\limits_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i,\boldsymbol{\theta}_k)\})}{\partial \boldsymbol{\theta}_t} \\
&= \sum_{i=1}^{n} \frac{\pi_t f_t'(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)}{\sum\limits_{t=1}^{K} \pi_t f_t(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)} \\
&= \sum_{i=1}^{n} \frac{\pi_t f_t'((y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)}{\sum\limits_{t=1}^{K} \pi_t f_t(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)} \frac{f_t(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)}{f_t(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)} \\
&= \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i,\boldsymbol{\Psi}) \frac{\partial \log f_t(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}_t}, \qquad (3.6)
\end{aligned}
$$

where for $t = 1, \ldots, K$,

$$
\tau_t(y_i|\mathbf{x}_i,\boldsymbol{\Psi}) = \frac{\pi_t f_t(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)}{\sum\limits_{t=1}^{K} \pi_t f_t(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)}
$$

is the posterior probability that y_i comes from the t^{th} component of the mixture model.

By defining the score function

$$
S(\mathbf{y}|\mathbf{X},\boldsymbol{\zeta}) = \sum_{i=1}^{n} \tau_t \frac{\partial \log f_t(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}_t},
$$

for $t = 1, \ldots, K$, the equation (3.6) becomes $S(\mathbf{y}|\mathbf{X},\boldsymbol{\zeta}) = 0$. Then we adopt the Newton-Raphson iteration

$$
\boldsymbol{\zeta}^{(s+1)} = \boldsymbol{\zeta}^{(s)} + [I(\boldsymbol{\zeta}^{(s)}|\mathbf{y},\mathbf{X})]^{-1} S(\mathbf{y}|\mathbf{X},\boldsymbol{\zeta}^{(s)}),
$$

where the $I(\boldsymbol{\zeta}^{(s)}|\mathbf{y},\mathbf{X})$ is the information matrix

$$\begin{aligned} I(\boldsymbol{\zeta}^{(s)}|\mathbf{y},\mathbf{X}) &= -\frac{\partial S(\mathbf{y}|\mathbf{X},\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}^{(s)T}} \\ &= -\sum_{i=1}^{n} \tau_t \frac{\partial^2 \log f_t(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}_t \partial \boldsymbol{\theta}_t^T} \end{aligned}$$

and $\boldsymbol{\zeta}^{(s+1)}$ is the $(s+1)^{th}$ iteration of $\boldsymbol{\zeta}$.

By calculate he information matrix with respect to β, α and σ, we have:

$$-\sum_{i=1}^{n} \tau_t \frac{\partial^2 \log f_t(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)}{\partial \boldsymbol{\beta}_t \partial \boldsymbol{\beta}_t^T} = \alpha_t \frac{\tau_t(y_i|\mathbf{x}_i,\boldsymbol{\Psi}^{(s)})}{(\sigma_t^2)^{\alpha_t/2}} \sum_{i=1}^{n} \left(\left(\left| y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t \right|^2 \right)^{(\alpha_t/2)-1} \left(\mathbf{x}_i \mathbf{x}_i^T \right) \left(\mathbf{x}_t - 1 \right) \right),$$

$$\begin{aligned} -\sum_{i=1}^{n} \tau_t \frac{\partial^2 \log f_t(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)}{\partial \alpha_t \partial \alpha_t^T} &= \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i,\boldsymbol{\Psi}^{(s)}) \left(\frac{1}{\alpha_t^2} + \frac{2}{\alpha_t^3} \frac{\Gamma'(1/\alpha_t)}{\Gamma(1/\alpha_t)} \right) \\ &+ \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i,\boldsymbol{\Psi}^{(s)}) \left(\frac{1}{\alpha_t^4} \left(\frac{\Gamma''(1/\alpha_t)}{\Gamma(1/\alpha_t)} - \left(\frac{\Gamma'(1/\alpha_t)}{\Gamma(1/\alpha_t)} \right)^2 \right) \right) \\ &+ \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i,\boldsymbol{\Psi}^{(s)}) \left(\frac{\left| y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t \right|}{\sigma_t} \right)^{\alpha_t} \left(\log \left(\frac{\left| y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t \right|}{\sigma_t} \right) \right)^2, \end{aligned}$$

$$\begin{aligned} -\sum_{i=1}^{n} \tau_t \frac{\partial^2 \log f_t(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)}{\partial \sigma^2} &= \sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i,\boldsymbol{\Psi}^{(s)})}{2\sigma_t^2} \\ &- \frac{\alpha_t}{2} \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i,\boldsymbol{\Psi}^{(s)}) \left| y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t \right|^{\alpha_t} \left(\sigma_t^2 \right)^{(-\alpha_t/2)-1}. \end{aligned}$$

To solve the equation (3.5), we let

$$\begin{aligned} A &= \log L(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X}) - \lambda(\sum_{k=1}^{K} \pi_k - 1) \\ &= \sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i,\boldsymbol{\theta}_k) \right\} - \lambda(\sum_{k=1}^{K} \pi_k - 1). \end{aligned}$$

Then equation(3.5) becomes

$$\begin{aligned}\frac{\partial A}{\partial \pi_t} &= \frac{\partial \sum\limits_{i=1}^{n} \log\left\{\sum\limits_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)\right\}}{\partial \pi_t} - \lambda \\ &= \sum_{i=1}^{n} \frac{f_t(y_i|\mathbf{x}_i, \boldsymbol{\theta}_t)}{\sum\limits_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)} - \lambda = 0, \qquad t = 1, \dots K. \qquad (3.7)\end{aligned}$$

Multiplying both sides of the above equation by $\sum\limits_{k=1}^{K} \pi_k$, we have

$$\begin{aligned}\sum_{k=1}^{K} \pi_k \frac{\partial A}{\partial \pi_t} &= \sum_{k=1}^{K} \sum_{i=1}^{n} \frac{\pi_t f_t(y_i|\mathbf{x}_i, \boldsymbol{\theta}_t)}{\sum\limits_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)} - \lambda \sum_{k=1}^{K} \pi_k \\ &= \sum_{i=1}^{n} \frac{\sum\limits_{t=1}^{K} \pi_t f_t(y_i|\mathbf{x}_i, \boldsymbol{\theta}_t)}{\sum\limits_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)} - \lambda \\ &= n - \lambda = 0,\end{aligned}$$

which implies $\lambda = n$.

Now we substitute $\lambda = n$ into (3.7) to obtain:

$$\sum_{i=1}^{n} \frac{f_t(y_i|\mathbf{x}_i, \boldsymbol{\theta}_t)}{\sum\limits_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)} = n.$$

This implies

$$\frac{1}{n} \sum_{i=1}^{n} \frac{1}{\pi_t} \frac{\pi_t f_t(y_i|\mathbf{x}_i, \boldsymbol{\theta}_t)}{\sum\limits_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)} = \frac{1}{n} \sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}_t)}{\pi_t} = 1.$$

Consequently, the proportion parameter π_t can be estimated by

$$\hat{\pi_t^{s+1}} = \frac{\sum_{i=1}^{n} \tau_t^s(y_i|\mathbf{x}_i, \hat{\mathbf{\Psi}}_t)}{n}, \qquad t = 1, \ldots K.$$

3.3 Indirect approach

In statistics, an expectation-maximization(EM) algorithm is an indirect approach to finding the maximum likelihood(ML) or the maximum a posterior (MAP) estimates of parameters in a statistical model, which depends on unobserved latent variables. EM is an iterative method which alternates between performing an expectation(E) step and maximization(M) step. In the E-step, the missing data are estimated by using the observed data and current estimate of the model parameters. In the M-step, the expected complete likelihood function is maximized under the assumption that the missing data are known. The estimates of the missing data from the E-step are used in lieu of the actual missing data.

Following Dempster et al. (1977), to apply the EM algorithm, we first need to formulate the above problem into an incomplete-data problem in the next subsection.

3.3.1 Formulation as an Incomplete-Data Problem

Suppose $(y_i, \mathbf{x}_i)$, $i = 1, \ldots, n$ are independent observations drawn from the mixture regression density

$$f(y|\mathbf{x}, \mathbf{\Psi}) = \sum_{k=1}^{K} \pi_k f_k(y|\mathbf{x}, \mathbf{\Psi}).$$

In the EM framework, the observed-data $\mathbf{y} = (y_1, \ldots, y_n)^T$ and $\mathbf{X} = (\mathbf{x_1}, \ldots, \mathbf{x_n})^{\mathbf{T}}$

are viewed as being incomplete, as the associated component-label vectors $\mathbf{z}_1, \ldots, \mathbf{z}_n$ are not available, where $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)^T$ with z_i being a K dimensional vector, defined by

$$z_{ik} = (z_i)_k = \begin{cases} 1 & y_i \in \text{the } k\text{th component} \\ 0 & y_i \notin \text{the } k\text{th component} \end{cases}$$

Note that z_i is distributed according to a multinomial distribution consisting of one drawn on K categories with probabilities $\pi_1, \ldots, \pi_K$, that is

$$\Pr\{Z_i = \mathbf{z}_i\} = \pi_1^{z_{1i}} \pi_2^{z_{2i}} \cdots \pi_K^{z_{Ki}},$$

where $\mathbf{z}_i = (z_{1i}, z_{2i}, \ldots, z_{Ki})^T$. We can estimate $\mathbf{z}_i = (z_{1i}, z_{2i}, \ldots, z_{Ki})^T$ by let $z_{ki} = 1$, and $z_{ti} = 0$ for $t = 1, \ldots, K$ if $\tau_k > \tau_t$ where $t \neq k$. The complete-data is therefore defined as

$$\mathbf{y}_c = (\mathbf{y}^T, \mathbf{z}^T)^T = \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix}.$$

Incomplete and complete data

Here, $f(\mathbf{y}|\mathbf{X}, \mathbf{\Psi})$ and $f(\mathbf{y}, \mathbf{z}|\mathbf{X}, \mathbf{\Psi})$ are viewed as incomplete and complete density functions respectively.

The incomplete density function is related to the conditional density $f(\mathbf{y}|\mathbf{z}, \mathbf{X}, \mathbf{\Psi})$ by

$$f(\mathbf{y}|\mathbf{X}, \mathbf{\Psi}) = \int_{\mathbf{z}} f(\mathbf{y}|\mathbf{z}, \mathbf{X}, \mathbf{\Psi}) f(\mathbf{z}|\mathbf{X}, \mathbf{\Psi}) dz, \tag{3.8}$$

which can be proved as follows.

$$\int_{\mathbf{z}} f(\mathbf{y}|\mathbf{z}, \mathbf{X}, \mathbf{\Psi}) f(\mathbf{z}|\mathbf{X}, \mathbf{\Psi}) d\mathbf{z} = \int_{\mathbf{z}} \frac{f(\mathbf{y}, \mathbf{z}, \mathbf{X}, \mathbf{\Psi})}{f(\mathbf{z}, \mathbf{X}, \mathbf{\Psi})} \frac{f(\mathbf{z}, \mathbf{X}, \mathbf{\Psi})}{f(\mathbf{X}, \mathbf{\Psi})} d\mathbf{z}$$

$$
\begin{aligned}
&= \int_{\mathbf{z}} \frac{f(\mathbf{y}, \mathbf{z}, \mathbf{X}, \boldsymbol{\Psi})}{f(\mathbf{X}, \boldsymbol{\Psi})} d\mathbf{z} \\
&= \int_{\mathbf{z}} f(\mathbf{y}, \mathbf{z}|\mathbf{X}, \boldsymbol{\Psi}) d\mathbf{z} \\
&= f(\mathbf{y}|\mathbf{X}, \boldsymbol{\Psi})
\end{aligned}
$$

i.e. $f(\mathbf{y}|\mathbf{X}, \boldsymbol{\Psi})$ is a marginal function of $f(\mathbf{y}, \mathbf{z}|\mathbf{X}, \boldsymbol{\Psi})$.

From equation (3.8) we obtain the complete-data likelihood for $\boldsymbol{\Psi}$, which is

$$L_c(\boldsymbol{\Psi}|\mathbf{y}, \mathbf{X}) = f(\mathbf{y}, \mathbf{z}|\mathbf{X}, \boldsymbol{\Psi}) = f(\mathbf{y}|\mathbf{z}, \mathbf{X}, \boldsymbol{\Psi}) f(\mathbf{z}|\mathbf{X}, \boldsymbol{\Psi}).$$

Note that

$$f(\mathbf{y}|\mathbf{z}, \mathbf{X}, \boldsymbol{\Psi}) = \prod_{i=1}^{n} f(y_i|z_i, \mathbf{x}_i, \boldsymbol{\Psi}) = \prod_{i=1}^{n} \prod_{k=1}^{K} f_k^{z_{ik}}(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)$$

and also that

$$f(\mathbf{z}|\boldsymbol{\Psi}) = \prod_{i=1}^{n} \prod_{k=1}^{K} \pi_k^{z_{ik}}.$$

We have the complete-data likelihood of $\boldsymbol{\Psi}$

$$
\begin{aligned}
L_c(\boldsymbol{\Psi}|\mathbf{y}, \mathbf{X}) &= \prod_{i=1}^{n} \prod_{k=1}^{K} f_k^{z_{ik}}(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k) \pi_k^{z_{ik}} \\
&= \prod_{i=1}^{n} \prod_{k=1}^{K} \{\pi_k f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)\}^{z_{ik}},
\end{aligned}
$$

and the complete-data log-likelihood of $\boldsymbol{\Psi}$ as follows:

$$\log L_c(\boldsymbol{\Psi}|\mathbf{y}, \mathbf{X}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \{\log \pi_k + \log f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)\}. \quad (3.9)$$

3.3.2 EM algorithm

After obtaining the complete log-likelihood, we use the EM algorithm to find the ML estimator of $\boldsymbol{\Psi}$. The algorithm includes two steps: the E-step and the M-step. In the E-step, we calculate the expectation of the complete log-likelihood of $\boldsymbol{\Psi}$; and in the M-step, we maximise the above expectation.

E-Step

Let $\boldsymbol{\Psi}^{(0)}$ be the initial value for $\boldsymbol{\Psi}$. Given the data $(\mathbf{y}, \mathbf{X})$ and the current value of $\boldsymbol{\Psi}^{(s)}$, we calculate the of the conditional expectation of $\log L_c(\boldsymbol{\Psi}|\mathbf{y}, \mathbf{X})$ with respect to $\mathbf{z}$. We have

$$\begin{aligned} Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(s)}) &= E_{\boldsymbol{\Psi}^{(s)}}\{\log L_c(\boldsymbol{\Psi}|\mathbf{y}, \mathbf{X})|\mathbf{y}, \mathbf{X}\} \qquad (3.10) \\ &= E_{\boldsymbol{\Psi}^{(s)}}\left\{\left[\sum_{k=1}^{K}\sum_{i=1}^{n} z_{ik}\left\{ \log \pi_k + \log f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)\right\}\right] |\mathbf{y}, \mathbf{X}\right\} \end{aligned}$$

The above equation shows that this expectation is being effected by using $\boldsymbol{\Psi}^{(0)}$ as the first iteration for $\boldsymbol{\Psi}$. On the $(s+1)^{th}$ iteration, we use $Q(\boldsymbol{\Psi}; \boldsymbol{\Psi}^{(s)})$, where $\boldsymbol{\Psi}^{(s)}$ is the value of $\boldsymbol{\Psi}$ after the s^{th} EM iteration.

From equation(3.10), we can see that $\log L_c(\boldsymbol{\Psi}|\mathbf{y}, \mathbf{X})$ is linear in the unobservable data z_{ik}, therefore, the E-step at the $(s+1)$th iteration simply requires the calculation of the current conditional expectation of z_{ik} given the observed data $\mathbf{y}$, i.e.

$$\begin{aligned} Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(s)}) &= E_{\boldsymbol{\Psi}^{(s)}}\{\log L_c(\boldsymbol{\Psi}|\mathbf{y}, \mathbf{X})|\mathbf{y}, \mathbf{X}\} \qquad (3.11) \\ &= E_{\boldsymbol{\Psi}^{(s)}}\left\{\left[\sum_{k=1}^{K}\sum_{i=1}^{n} z_{ik}\{\log \pi_k + \log f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)\}\right] |\mathbf{y}, \mathbf{X}\right\} \\ &= E_{\boldsymbol{\Psi}^{(s)}}\left\{\left[\sum_{k=1}^{K}\sum_{i=1}^{n}(z_{ik}|\mathbf{y}, \mathbf{X})[\log \pi_k + \log f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)\right]\right\} \end{aligned}$$

$$= \sum_{k=1}^{K}\sum_{i=1}^{n} E_{\mathbf{\Psi}^{(s)}}(z_{ik}|\mathbf{y},\mathbf{X})\left\{\log\pi_k + \log f_k(y_i|\mathbf{x}_i,\boldsymbol{\theta}_k)\right\}.$$

Note that

$$\begin{aligned}
E_{\mathbf{\Psi}^{(s)}}(z_{ik}|\mathbf{y},\mathbf{X}) &= 0\cdot p(z_{ik}=0|y_i,\mathbf{x}_i,\mathbf{\Psi}^{(s)}) + 1\cdot p(z_{ik}=1|y_i,\mathbf{x}_i,\mathbf{\Psi}^{(s)})\\
&= p(z_{ik}=1|y_i,\mathbf{x}_i,\mathbf{\Psi}^{(s)})\\
&= p(z_{ik}=1, z_{it}=0, t\neq i|y_i,\mathbf{x}_i,\mathbf{\Psi}^{(s)})\\
&= \frac{p(z_{ik}=1, z_{it}=0, t\neq i, y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})}{f(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})}\\
&= \frac{f_k(y_i,\mathbf{x}_i,\mathbf{\Psi}^{(s)})p(z_{ik}=1, z_{it}=0, t\neq i)}{f(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})}\\
&= \frac{f_k(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})\pi_k^{(s)}}{f(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})}\\
&= \tau_k(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)}).
\end{aligned}$$

Substituting the above formula into equation (3.11), we have

$$\begin{aligned}
Q(\mathbf{\Psi}|\mathbf{\Psi}^{(s)}) &= \sum_{k=1}^{K}\sum_{i=1}^{n}\tau_k(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})\left\{\log\pi_k + \log f_k(y_i|\boldsymbol{\theta}_k)\right\}\\
&= \tau_1(y_1|\mathbf{x}_1,\mathbf{\Psi}^{(s)})\left\{\log\pi_1 + \log f_1(y_1|\mathbf{x}_1,\boldsymbol{\theta}_1)\right\} + \cdots\\
&\quad + \tau_K(y_n|\mathbf{x}_n,\mathbf{\Psi}^{(s)})\left\{\log\pi_K + \log f_K(y_n|\mathbf{x}_n,\boldsymbol{\theta}_K)\right\}.
\end{aligned}$$

M-Step

Similar to the E-step, in order to find the updated estimate $\mathbf{\Psi}^{(s+1)}$, we maximise $Q(\mathbf{\Psi}|\mathbf{\Psi}^{(s)})$ w.r.t $\mathbf{\Psi}$ over the parameter space Ω.

We know from Section 3.2 that, for the finite mixture model, the calculation of $\pi_k^{(s+1)}$ is independent of $\boldsymbol{\zeta}^{(s+1)}$ as shown below.

Let

$$\begin{aligned}
B &= Q(\mathbf{\Psi}|\mathbf{\Psi}^{(s)}) - \lambda(\sum_{k=1}^{K}\pi_k - 1) \\
&= E_{\mathbf{\Psi}^{(s)}}\{\log L_c(\mathbf{\Psi}|\mathbf{y},\mathbf{X})|\mathbf{y},\mathbf{X}\} - \lambda(\sum_{k=1}^{K}\pi_k - 1) \\
&= \tau_1(y_1|\mathbf{x}_1,\mathbf{\Psi}^{(s)})\{\log\pi_1 + \log f_1(y_1|\mathbf{x}_1,\boldsymbol{\theta}_1)\} + \cdots \\
&\quad +\tau_K(y_n|\mathbf{x}_n,\mathbf{\Psi}^{(s)})\{\log\pi_K + \log f_K(y_n|\mathbf{x}_n,\boldsymbol{\theta}_K)\} - \lambda(\sum_{k=1}^{K}\pi_K - 1) \\
&= \sum_{k\neq t}^{K}\sum_{i=1}^{n}\tau_k(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})\{\log\pi_k + \log f_k(y_i|\mathbf{x}_i,\boldsymbol{\theta}_k)\} \\
&\quad +\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})\{\log\pi_t + \log f_t(y_i|\mathbf{x}_i,\boldsymbol{\theta}_t)\} - \lambda(\sum_{k=1}^{K}\pi_k - 1).
\end{aligned}$$

In order to find the global maximum estimate of π_t, we need to solve the derivative equation of B with respect to π_t and let it be equal to zero:

$$\frac{\partial B}{\partial \pi_t} = \sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})\frac{1}{\pi_t} - \lambda = 0.$$

By rearranging the equation we obtain

$$\pi_t = \frac{\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})}{\lambda} \tag{3.12}$$

As we know the summation of the proportion estimators π_k, $k = 1,\ldots,K$ equals to 1, we can solve the above equation by summing up the terms from 1 to K on both sides of the equation:

$$\sum_{t=1}^{K}\pi_t = \sum_{t=1}^{K}\frac{\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})}{\lambda} = 1.$$

Rearranging the equation, we get:

$$
\begin{aligned}
\lambda &= \sum_{t=1}^{K}\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \\
&= \sum_{t=1}^{K}\sum_{i=1}^{n} \frac{\pi_t^{(s)} f_t(y_i|\mathbf{x}_i, \boldsymbol{\theta}_t^{(s)})}{\sum\limits_{k=1}^{K} \pi_k^{(K)} f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k^{(s)})} \\
&= \sum_{i=1}^{n} \frac{\sum\limits_{t=1}^{K} \pi_t^{(s)} f_t(y_i|\mathbf{x}_i, \boldsymbol{\theta}_t^{(s)})}{\sum\limits_{k=1}^{K} \pi_k^{(s)} f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k^{(s)})} \\
&= \sum_{i=1}^{n} 1 = n
\end{aligned}
$$

From equation (3.12), we have the updated estimate of π_t at the $(s+1)^{th}$ iteration, which is given by

$$
\pi_t^{(s+1)} = \frac{\sum\limits_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)})}{n}, \qquad t = 1, \ldots, K. \tag{3.13}
$$

Concerning the updating of the remaining parameters $\boldsymbol{\zeta}$ on the M-step of the $(s+1)^{th}$ iteration, we differentiate $Q(\mathbf{\Psi}|\mathbf{\Psi}^{(s)})$ with respect to $\boldsymbol{\zeta}$ and set it to zero. We have

$$
\begin{aligned}
\frac{\partial Q(\mathbf{\Psi}|\mathbf{\Psi}^{(s)})}{\partial \zeta} &= \frac{\partial E\left\{\log L_c(\mathbf{\Psi}|\mathbf{y}, \mathbf{X})|\mathbf{y}, \mathbf{X}\right\}}{\partial \zeta} \\
&= \sum_{k=1}^{K}\sum_{i=1}^{n} \tau_k(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \frac{\partial \log f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)}{\partial \zeta} \\
&= 0.
\end{aligned}
$$

The detail of how to estimate the parameters via the second derivative of $Q(\mathbf{\Psi}|\mathbf{\Psi}^{(s)})$ can be find in Chapter 5.1.

The E-step and M-step are alternated repeatedly until the difference $L(\mathbf{\Psi}^{(s+1)}|\mathbf{y},\mathbf{X})-L(\mathbf{\Psi}^{(s)}|\mathbf{y},\mathbf{X})$ is less than a pre-specified value.

Following Dempster et al. (1977), we show that the incomplete-data likelihood function $L(\mathbf{\Psi}|\mathbf{y},\mathbf{X})$ is not decreased after an EM iteration; that is $L(\mathbf{\Psi}^{(s+1)}|\mathbf{y},\mathbf{X}) \geqslant L(\mathbf{\Psi}^{(s)}|\mathbf{y},\mathbf{X})$, for $s=0,1,2\ldots$ as follows.

Proof. We define:

$$l(\mathbf{y}|\mathbf{X},\mathbf{\Psi}) \;=\; \log f(\mathbf{y}|\mathbf{X},\mathbf{\Psi}) = \log \int f(\mathbf{y},\mathbf{z}|\mathbf{X},\mathbf{\Psi})d\mathbf{z}$$

We know from the E-step that:

$$\begin{aligned} Q(\mathbf{\Psi}|\mathbf{\Psi}^{(s)}) &= E\left\{\log L(\mathbf{\Psi}|\mathbf{y},\mathbf{X})|\mathbf{y},\mathbf{X},\mathbf{\Psi}^{(s)}\right\} \\ &= E\left\{\log f(\mathbf{y},\mathbf{z}|\mathbf{X},\mathbf{\Psi})|\mathbf{y},\mathbf{\Psi}^{(s)}\right\}, \end{aligned}$$

we want to prove that if

$$Q(\mathbf{\Psi}^{(s+1)}|\mathbf{\Psi}^{(s)}) \geqslant Q(\mathbf{\Psi}^{(s)}|\mathbf{\Psi}^{(s)}).$$

then

$$l(\mathbf{y}|\mathbf{X},\mathbf{\Psi}^{(s+1)}) \geqslant l(\mathbf{y}|\mathbf{X},\mathbf{\Psi}^{(s)})$$

we can rewrite the $l(\mathbf{y}|\mathbf{X},\mathbf{\Psi})$ in the form:

$$\begin{aligned} l(\mathbf{y}|\mathbf{X},\mathbf{\Psi}) &= E\left\{\log \frac{f(\mathbf{y},\mathbf{z}|\mathbf{X},\mathbf{\Psi})}{f(\mathbf{z}|\mathbf{y},\mathbf{X},\mathbf{\Psi})}|\mathbf{y},\mathbf{X},\mathbf{\Psi}^{(s)}\right\} \\ &= E\left\{\log f(\mathbf{y},\mathbf{z}|\mathbf{X},\mathbf{\Psi})|\mathbf{y},\mathbf{X},\mathbf{\Psi}^{(s)}\right\} - E\left\{\log f(\mathbf{z}|\mathbf{y},\mathbf{X},\mathbf{\Psi})|\mathbf{y},\mathbf{X},\mathbf{\Psi}^{(s)}\right\} \\ &= Q(\mathbf{\Psi}|\mathbf{\Psi}^{(s)}) - H(\mathbf{\Psi}|\mathbf{\Psi}^{(s)}), \end{aligned}$$

where we define that

$$H(\mathbf{\Psi}|\mathbf{\Psi}^{(s)}) = E\left\{\log f(\mathbf{z}|\mathbf{y}, \mathbf{X}, \mathbf{\Psi})|\mathbf{y}, \mathbf{X}, \mathbf{\Psi}^{(s)}\right\}.$$

Now we want to show that if

$$l(\mathbf{y}|\mathbf{X}, \mathbf{\Psi}^{(s+1)}) = Q(\mathbf{\Psi}^{(s+1)}|\mathbf{\Psi}^{(s)}) - H(\mathbf{\Psi}^{(s+1)}|\mathbf{\Psi}^{(s)})$$

and

$$l(\mathbf{y}|\mathbf{X}, \mathbf{\Psi}^{(s)}) = Q(\mathbf{\Psi}^{(s)}|\mathbf{\Psi}^{(s)}) - H(\mathbf{\Psi}^{(s)}|\mathbf{\Psi}^{(s)})$$

then,

$$Q(\mathbf{\Psi}^{(s+1)}|\mathbf{\Psi}^{(s)}) \geqslant Q(\mathbf{\Psi}^{(s)}|\mathbf{\Psi}^{(s)}).$$

Here we prove by contradiction, i.e. we aim to prove the opposite of our goal is not true, hence we prove

$$H(\mathbf{\Psi}^{(k+1)}|\mathbf{\Psi}^{(k)}) \leqslant H(\mathbf{\Psi}^{(k)}|\mathbf{\Psi}^{(k)})$$

is not true.
By rearranging the above equation, we have

$$H(\mathbf{\Psi}^{(s+1)}|\mathbf{\Psi}^{(s)}) - H(\mathbf{\Psi}^{(s)}|\mathbf{\Psi}^{(s)}) \leqslant 0$$
$$E[\log f(\mathbf{z}|\mathbf{y}, \mathbf{X}, \mathbf{\Psi}^{(s+1)})|\mathbf{y}, \mathbf{X}, \mathbf{\Psi}^{(s)}] - E[\log f(\mathbf{z}|\mathbf{y}, \mathbf{X}, \mathbf{\Psi}^{(s)})|\mathbf{y}, \mathbf{X}, \mathbf{\Psi}^{(s)}] \leqslant 0,$$

which implies,

$$E\left\{\log \frac{f(\mathbf{z}|\mathbf{y}, \mathbf{X}, \mathbf{\Psi}^{(s+1)})}{f(\mathbf{z}|\mathbf{y}, \mathbf{X}, \mathbf{\Psi}^{(s)})}|\mathbf{y}, \mathbf{X}, \mathbf{\Psi}^{(s)}\right\} \leqslant 0.$$

Now we can calculate it as

$$\begin{aligned}
E\left\{\log\frac{f(\mathbf{z}|\mathbf{y},\mathbf{X},\mathbf{\Psi}^{(s+1)})}{f(\mathbf{z}|\mathbf{y},\mathbf{X},\mathbf{\Psi}^{(s)})}|\mathbf{y},\mathbf{X},\mathbf{\Psi}^{(s)}\right\} &\leqslant \log E\left\{\frac{f(\mathbf{z}|\mathbf{y},\mathbf{X},\mathbf{\Psi}^{(s+1)})}{f(\mathbf{z}|\mathbf{y},\mathbf{X},\mathbf{\Psi}^{(s)})}|\mathbf{y},\mathbf{X},\mathbf{\Psi}^{(s)}\right\} \\
&= \log\int\frac{f(\mathbf{z}|\mathbf{y},\mathbf{X},\mathbf{\Psi}^{(s+1)})}{f(\mathbf{z}|\mathbf{y},\mathbf{X},\mathbf{\Psi}^{(s)})}f(\mathbf{z}|\mathbf{y},\mathbf{X},\mathbf{\Psi}^{(s)})d\mathbf{z} \\
&= \log 1 = 0
\end{aligned}$$

Hence convergence must be obtained with a sequence of likelihood values that are bounded as described above. □

Chapter 4

Maximum Likelihood Estimation of Gaussian Mixture Regressions

A Gaussian mixture regression model is a parametric density function which can be represented as the sum of weighted Gaussian density components. It is often applied in biometric studies. The most common method of estimating the parameters is based on the Maximum Likelihood. There are many applications of Gaussian mixture regression models, such as imitation learning for multiple tasks (Cederborg et al., 2010), regression classification (Sung, 2004), trajectory clustering (Gaffney and Smyth, 1999), and among others.

In this chapter we will firstly illustrate that how the EM algorithm is used to estimate Gaussian mixture regression models, followed by build a Gaussian mixture model to reflect grouping structures in the data. At last, we will describe two simulations, one of them shows the performance of the EM algorithm in estimating a two-component Gaussian mixture model, and the other assesses the effect of the sample size on the accuracy of Gaussian mixture regression-based clustering.

4.1 Methodology

Suppose we have an independent sample $(y_i, \mathbf{x}_i)$, $1 \leq i \leq n$, where conditional on $\mathbf{x}$, y_i drawn from the model

$$f(y|\mathbf{x}, \boldsymbol{\beta}, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(y - \mathbf{x}^T\boldsymbol{\beta})^2}{2\sigma^2}),$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_P)^T$, and $\mathbf{x}_i = (1, x_{i1}, \ldots, x_{iP})^T$.

The likelihood function can be written as

$$\begin{aligned} L(\boldsymbol{\Psi}|\mathbf{y}, \mathbf{X}) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2}{2\sigma^2}) \\ &= (2\pi\sigma^2)^{-\frac{n}{2}} \exp(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2), \end{aligned}$$

and the log-likelihood function is

$$\log L(\boldsymbol{\Psi}|\mathbf{y}, \mathbf{X}) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2.$$

Then we differentiate the log-likelihood function with respect to $\boldsymbol{\zeta}$ and let it equal to 0. By solving the equation, we have the following estimators:

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}, \\ \hat{\sigma^2} &= \frac{\mathbf{y}(I_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X})\mathbf{y}}{n}, \end{aligned}$$

where I_n is an $n \times n$ unit matrix.

Now suppose that $(y_i, \mathbf{x}_i)$, $i = 1, \ldots, n$ are drawn from the following finite Gaussian mixture regression model with K components,

$$f(y|\mathbf{x}, \boldsymbol{\Psi}) = \sum_{k=1}^{K} \pi_k f_k(y|\mathbf{x}, \boldsymbol{\theta}_k).$$

In the EM framework, the observed-data $\mathbf{D} = (y_1, \mathbf{x}_1; \ldots; y_n, \mathbf{x}_n)^T$ are viewed as an incomplete data. We combine it with the indicator z_{ik} to form the complete-data

$$\mathbf{D}_c = (\mathbf{D}, \mathbf{z}).$$

It follows from equation (3.9) that the complete-data log-likelihood for Ψ can be written as

$$\begin{aligned}\log L_c(\boldsymbol{\Psi}|\mathbf{y}, \mathbf{X}) &= \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\{\log \pi_k + \log f_k(y_i|\mathbf{x}_i, \theta_k)\} \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\{\log \pi_k - \frac{1}{2}\log(2\pi\sigma_k^2) - \frac{(y_i - \mathbf{x}_i^T\boldsymbol{\beta}_k)^2}{2\sigma_k^2}\}.\end{aligned}$$

4.1.1 E-Step

We calculate the expectation of the complete-data log-likelihood based on the current values of parameters $\Psi^{(s)}$ below:

$$\begin{aligned}Q(\Psi|\Psi^{(s)}) &= E_{\Psi^{(s)}}\{\log L_c(\boldsymbol{\Psi}|\mathbf{y}, \mathbf{X})\,|\mathbf{y}, \mathbf{X}\} \\ &= \tau_1(y_1|\mathbf{x}_1, \Psi^{(s)})\{\log \pi_1 + \log f_1(y_1|\mathbf{x}_1, \theta_1)\} + \cdots \\ &\quad + \tau_K(y_n|\mathbf{x}_n, \Psi^{(s)})\{\log \pi_K + \log f_K(y_n|\mathbf{x}_n, \theta_K)\} \\ &= \tau_1(y_1|\mathbf{x}_1, \Psi^{(s)})\{\log \pi_1 - \frac{1}{2}\log(2\pi\sigma_1^2) - \frac{(y_1 - \mathbf{x}_1^T\boldsymbol{\beta}_1)^2}{2\sigma_1^2}\} + \cdots \\ &\quad + \tau_K(y_n|\mathbf{x}_n, \Psi^{(s)})\{\log \pi_K - \frac{1}{2}\log(2\pi\sigma_K^2) - \frac{(y_n - \mathbf{x}_n^T\boldsymbol{\beta}_K)^2}{2\sigma_K^2}\}.\end{aligned}$$

4.1.2 M-Step

In this step we estimate the parameters in the $(s+1)^{th}$ iteration, based on the parameters obtained from the s^{th} iteration. For this purpose, we maximise the expectation of the complete-data log-likelihood from the last step.

To update the value of π_t, we define

$$\begin{aligned}
B &= Q(\Psi|\Psi^{(s)}) - \lambda(\sum_{i=1}^{K} \pi_k - 1) \\
&= \tau_1(y_1|\mathbf{x}_1, \Psi^{(s)})\{\log \pi_1 - \frac{1}{2}\log(2\pi\sigma_1^2) - \frac{(y_1 - \mathbf{x}_1^T\boldsymbol{\beta}_1)^2}{2\sigma_1^2}\} + \cdots \\
&+ \tau_K(y_n|\mathbf{x}_n, \Psi^{(s)})\{\log \pi_K - \frac{1}{2}\log(2\pi\sigma_K^2) - \frac{(y_n - \mathbf{x}_n^T\boldsymbol{\beta}_K)^2}{2\sigma_K^2}\} - \lambda(\sum_{k=1}^{K} \pi_k - 1).
\end{aligned}$$

Then, we differentiate B with respect to π_t and set it to be zero. This gives $\lambda = n$, and

$$\hat{\pi}_t^{(s+1)} = \frac{\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \Psi^{(s)})}{n}.$$

Following on, we update the remaining parameters

$$\boldsymbol{\zeta} = \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\sigma}^2 \end{pmatrix},$$

where $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_K^T)^T$ and $\boldsymbol{\sigma}^2 = (\sigma_1^2, ..., \sigma_k^2)^T$.

In the M-Step of the $(s+1)^{th}$ iteration, function $Q(\Psi|\Psi^{(s)})$ is differentiated with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\sigma}^2$ respectively:

$$\frac{\partial Q(\Psi|\Psi^{(s)})}{\partial \zeta} = \begin{pmatrix} \frac{\partial Q(\Psi|\Psi^{(s)})}{\partial \boldsymbol{\beta}} \\ \frac{\partial Q(\Psi|\Psi^{(s)})}{\partial \boldsymbol{\sigma}^2} \end{pmatrix} = 0.$$

By solving the above equations, we get the estimated $\boldsymbol{\beta}$ and σ^2 of the t^{th}

component for the $(s+1)^{th}$ iteration:

$$
\begin{aligned}
\hat{\boldsymbol{\beta}}_t^{(s+1)} &= \frac{\sum\limits_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \Psi^{(s)})\mathbf{x}_i y_i}{\sum\limits_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \Psi^{(s)})\mathbf{x}_i\mathbf{x}_i^T}, \\
\hat{\sigma}_t^{(s+1)} &= \frac{\sum\limits_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \Psi^{(s)})(y_i - \mathbf{x}_i^T\hat{\boldsymbol{\beta}}_t^{s})^2}{\sum\limits_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \Psi^{(s)})}.
\end{aligned}
$$

We define $\mathbf{W}_t^{(s)} = \frac{n}{\sum\limits_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i,\Psi^{(s)})} \begin{pmatrix} \tau_t(y_1|\mathbf{x}_1, \Psi^{(s)}) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \tau_t(y_n|\mathbf{x}_n, \Psi^{(s)}) \end{pmatrix}$,

let $\mathbf{X}^* = \mathbf{W}^{1/2}\mathbf{X}$, and $\mathbf{y}^* = \mathbf{W}^{1/2}\mathbf{y}$, hence, we converted the generalised regression estimation problem to a least square problem, such that

$$
\begin{aligned}
\hat{\boldsymbol{\beta}}_t^{(s+1)} &= (\mathbf{X}^{*T}\mathbf{X}^*)^{-1}\mathbf{X}^{*T}\mathbf{y}^*, \\
\hat{\sigma}^{2\,(s+1)}_{t} &= \frac{\mathbf{y}^*(I_n - \mathbf{X}^*(\mathbf{X}^{*T}\mathbf{X}^*)^{-1}\mathbf{X}^*)\mathbf{y}^*}{n}.
\end{aligned}
$$

The similar approach has been used by Benagliz et al. (2009).

4.2 Simulation

After discussing the EM algorithm and the process of its iterations, we assess two simulations: the first one is to discuss the accuracy of estimation by using EM algorithm in a two-component Gaussian mixture model; the second one is to assesses the effect of the sample size on the accuracy of Gaussian mixture regression-based clustering, under different scenarios.

4.2.1 Simulation 1 (Univariate Gaussian mixture model)

In this simulation, we assess the accuracy of the EM-based estimation for a two-component univariate Gaussian mixture model.

Suppose we have a simple univariate Gaussian mixture model $f = \pi_1 f_1(y) + \pi_2 f_2(y)$, where $f_1(y)$ and $f_2(y)$ are Gaussian probability density functions, π_1 and π_2 are proportions of each component with the constraint $\sum_{k=1}^{2} \pi_k = 1$.

We simulated 100 observations $y = (y_1, \ldots, y_{100})^T$ from two sub-populations: $N(5, 2)$ and $N(1, 1)$ with proportions 0.6 and 0.4 respectively. The real log-likelihood of the parameters is -219.516. The initial values of the parameters are randomly chose with constraints that $\pi_1 + \pi_2 = 1$, $\sigma_1^2 \geq 0$ and $\sigma_2^2 \geq 0$.

After run the simulation 100 times, we obtained 100 estimates for each parameter and 100 estimated log-likelihood. We averaged these estimates and calculated the corresponding empirical bias for each parameter and log-likelihood. The mean square errors for the estimator were also obtained. The results are shown in Table 4.1.

Table 4.1: Simulation of a 2 components univariate Gaussian mixture model.

Two components Gaussian mixture model							
	π		μ		σ		$L(\Psi\|\mathbf{y}, \mathbf{X})$
	π_1	π_2	μ_1	μ_2	σ_1	σ_2	
Real	0.600	0.400	5.000	1.000	2.000	1.000	−219.516
E-Bias	−0.030	0.027	0.280	0.023	−0.055	−0.027	−0.001
Mean squared error	0.008	0.005	0.086	0.060	0.038	0.010	0.003

From the above Table 4.1, we realise that the estimated log-likelihood is very close to the real log-likelihood. The mean square errors are very small for all parameters in general, but the mean square errors for the first group are larger than those for the second group. It suggests that the mean square error, which indicates the accuracy, of each estimate is proportional to the underlying sub-population variances.

4.2.2 Simulation 2 (Gaussian mixture regression models)

In this simulation we assesses the effect of the sample size on the accuracy of Gaussian mixture regression-based clustering, under the following scenarios:

1. different number of components;
2. different proportion weight of each component;
3. different correlation structure of covariates;
4. equal and unequal component variances;
5. various regression coefficients.

Suppose we have a parametric mixture model of regression:

$$f(y_i|\mathbf{x}_i, \Psi) = \sum_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i, \theta_k),$$

where each component is Gaussian distributed.
Let $\mathbf{X} = (\mathbf{x_1}, \ldots, \mathbf{x_n})^{\mathbf{T}}$ be the covariate matrix, and $\mathbf{x_i} = (1, x_{i1}, \ldots, x_{i6})^T = (1, \mathbf{x}_i^*)^T$, where $\mathbf{x}_i^* = (x_{i1}, \ldots, x_{i6})^T$ denote the i^{th} observations on 6 covariates. We generated $\mathbf{x}_i^*$ from $N(0, \Sigma)$, where $\Sigma = (\rho^{|l-m|})_{1 \leq l,m \leq 6}$ and $0 \leq \rho \leq 1$ is a constant. For $i = 1, \ldots, n$, conditional on $\mathbf{x}_i$, we generated y_i from K-component Gaussian regression model. We standardised both $\mathbf{y}$ and the covariates in $\mathbf{X}$ in order to compare groups and penalties. The results are summarised in terms of the averaged RAND index over 100 replicates in the following tables.

Simulation 2.1 (Two components)

In this simulation we assess the effect of various regression coefficients on the accuracy of Gaussian mixture regression-based clustering. We consider two different scenarios with large or small distances between the regression coefficients respectively. Both scenarios contain the following cases: (1) component variances are equal or unequal; (2) component proportions are equal or unequal; (3) the correlation structure of covariates is different.

Data with relatively large distance between the regression coefficients We let the regression coefficients $\boldsymbol{\beta}_1 = (0, 3, 3, 3, 3, 3, 3)^T$ and $\boldsymbol{\beta}_2 = (0, -1, -1, -1, -1, -1, -1)^T$, and consider the different combinations with the rest parameters: $(\sigma_1, \sigma_2) : (1, 1), (2, 2), (1.5, 2.5), (1.5, 3.5)$, $\rho : 0.5, 0.75$, $(\pi_1, \pi_2) : (0.5, 0.5), (0.3, 0.7)$. The results of averaged RAND indexes under the different scenarios are summarized in Table 4.2. Figure 4.1 shows the box plots of RAND indexes when the two component variances are equal while Figure 4.2 shows the box plots of RAND indexes when the two component variances are unequal. In the Figures, from the left to the right the following cases are considered: ($\pi_1 = \pi_2 = 0.5$, $\rho = 0.5$), ($\pi_1 = \pi_2 = 0.5$,$\rho = 0.75$), ($\pi_1 = 0.3$, $\pi_2 = 0.7$,$\rho = 0.5$), ($\pi_1 = 0.3$, $\pi_2 = 0.7$, $\rho = 0.75$).

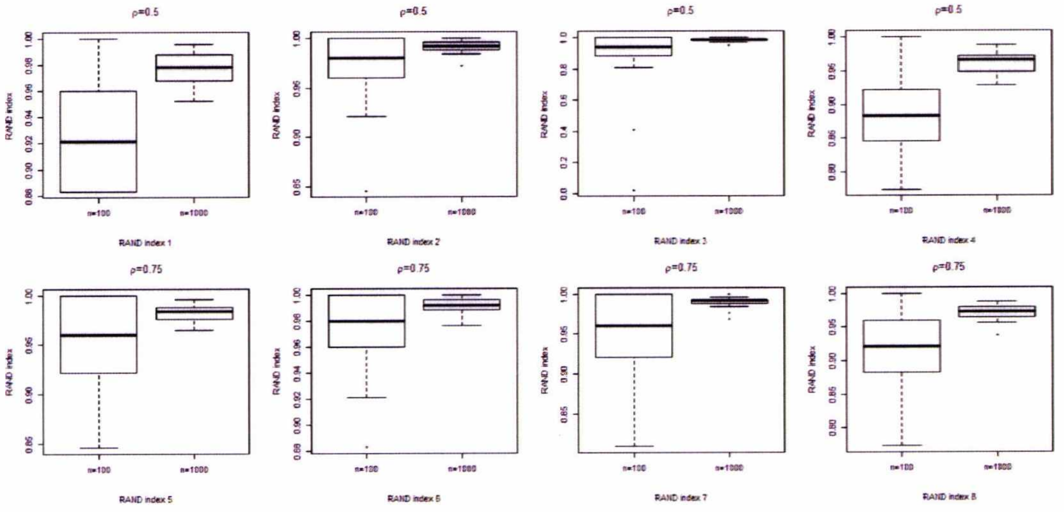

Figure 4.1: Rand indexes of GM regression model with 2 equal variance components when the distance of the regression coefficients between the 2 groups are relatively large.

From Table 4.2, It can be seen that when the distinction between two groups are clear, the RAND indexes of GM model with a large sample size are about

Table 4.2: Averaged RAND indexes for GM regression model with 2 components when the distance of the regression coefficients between the 2 groups are relatively large. (The numbers in the parenthesis are the standard errors)

Equal variance cases: $k = 2$ and $p = 6$					
Cases		$\pi = 0.5$		$\pi = 0.3$	
ρ	σ	$n = 100$	$n = 1000$	$n = 100$	$n = 1000$
0.50	1	$0.931_{(0.008)}$	$0.975_{(0.002)}$	$0.972_{(0.006)}$	$0.993_{(0.001)}$
	2	$0.881_{(0.010)}$	$0.976_{(0.003)}$	$0.882_{(0.036)}$	$0.988_{(0.002)}$
0.75	1	$0.958_{(0.008)}$	$0.988_{(0.001)}$	$0.964_{(0.006)}$	$0.992_{(0.001)}$
	2	$0.915_{(0.012)}$	$0.975_{(0.002)}$	$0.951_{(0,008)}$	$0.989_{(0.001)}$

Unequal variance cases: $k = 2$ and $p = 6$						
Cases			$\pi = 0.5$		$\pi = 0.3$	
ρ	σ_1	σ_2	$n = 100$	$n = 1000$	$n = 100$	$n = 1000$
0.5	1.5	2.5	$0.926_{(0.013)}$	$0.983_{(0.002)}$	$0.925_{(0.010)}$	$0.987_{(0.002)}$
	1.5	3.5	$0.885_{(0.032)}$	$0.978_{(0.002)}$	$0.914_{(0.013)}$	$0.977_{(0.002)}$
0.75	1.5	2.5	$0.950_{(0.008)}$	$0.988_{(0.002)}$	$0.956_{(0.008)}$	$0.985_{(0.001)}$
	1.5	3.5	$0.927_{(0.010)}$	$0.981_{(0.002)}$	$0.941_{(0.009)}$	$0.983_{(0.002)}$

$3\% - 8\%$ larger than the one with a small sample size under different scenarios. The standard errors of RAND indexes are also smaller for GM model with a large sample. When the variance of each component are equal, the difference of RAND indexes between large data and small data when the variances are equals to 2 is bigger than the difference of RAND indexes when the variance are equals to 1. Similarly, in the scenario where the component variances are unequal, the RAND indexes were improved more when variance are 1.5 and 3.5 compared to variance are 1.5 and 2.5 respectively. It shows that the advantage of clustering with large sample size is more obviously when the variance are large as it takes more information.

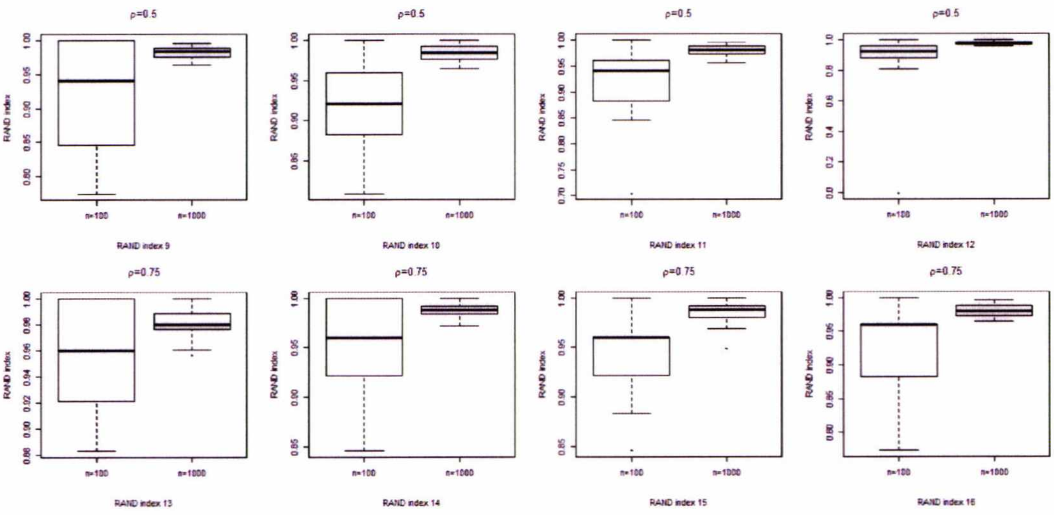

Figure 4.2: RAND indexes of GM regression model with 2 unequal variance components when the distance of the regression coefficients between the 2 groups are relatively large.

From the Figure 4.1 and 4.2, we can see that the dispersion degrees of RAND indexes with small sample size is much bigger than the one with large sample size under all scenarios, which means the model with large sample size has higher stability. Therefore, the more reliable clustering results can be expected when the sample size become larger.

To sum up, when regression coefficients of the two groups are significantly different from each other, the GM model under large sample size has better performance on clustering compared to relatively small sample size in all situations.

Data with relatively small distance between the regression coefficients In this simulation, we illustrate the performance of RAND index of GM regression model with a small dataset and a relatively large dataset when the distance between the regression coefficients for each component are relatively small, i.e the distinction between the two groups are not clear.

Firstly, we let the regression coefficients $\boldsymbol{\beta}_1 = (0,1,1,1,1,1,1)^T$ and $\boldsymbol{\beta}_2 = (0,1.5,1.5,1.5,1.5,1.5,1.5)^T$ respectively. Similar as in the previous simulation, we consider the different combinations of the regression coefficients with the rest parameters: $(\sigma_1, \sigma_2) : (1,1), (2,2), (1.5,2.5), (1.5,3.5)$, $\rho : 0.5, 0.75$, $(\pi_1, \pi_2) : (0.5,0.5), (0.3,0.7)$. The results of averaged RAND indexes under the different scenarios are summarized in Table 4.3. Figure 4.3 shows the box plots of RAND indexes when the two component variances are equal while Figure 4.4 shows the box plots of RAND indexes when the two component variances are unequal. In the Figures, from the left to the right the following cases are considered: ($\pi_1 = \pi_2 = 0.5$, $\rho = 0.5$), ($\pi_1 = \pi_2 = 0.5$,$\rho = 0.75$), ($\pi_1 = 0.3$, $\pi_2 = 0.7$,$\rho = 0.5$), ($\pi_1 = 0.3$, $\pi_2 = 0.7$, $\rho = 0.75$).

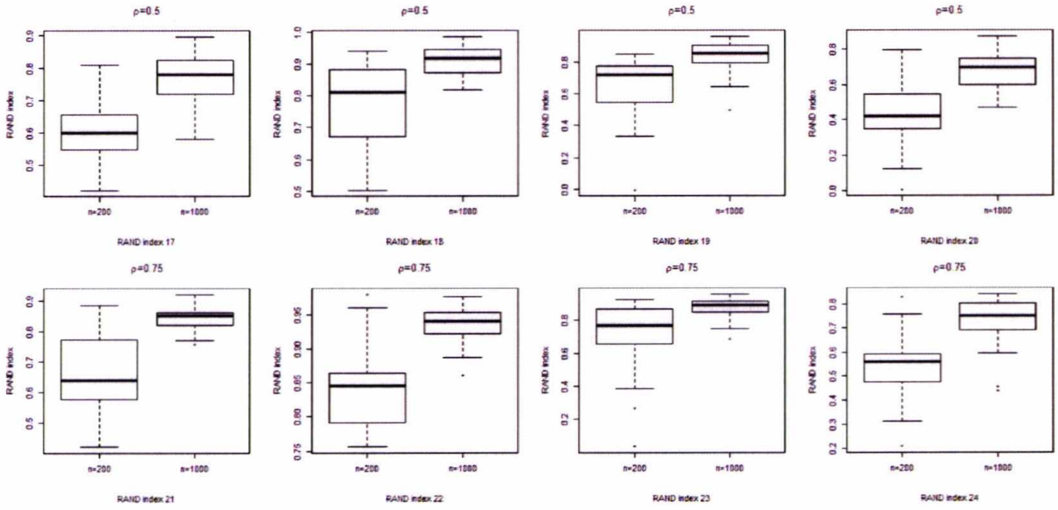

Figure 4.3: Rand indexed of GM regression model with 2 equal variance components when the distance of correlation coefficients between the 2 groups is relatively small.

From 4.2 we can see, when the distance of regression coefficients is small, the accuracy of clustering of both samples is about 20% worse compared to the situation when the distance is large under all scenarios. However, the difference between the RAND index of two samples is larger, compare with the previous simulation. The RAND indexes with the large sample improved about

Table 4.3: Averaged RAND indexes of GM regression model with 2 components when the distance of the regression coefficients between the 2 groups is relatively small. (The numbers in the parenthesis are the standard errors)

Equal variance cases: $k = 2$ and $p = 6$

Cases		$\pi = 0.5$		$\pi = 0.3$	
ρ	σ	$n = 200$	$n = 1000$	$n = 200$	$n = 1000$
0.50	1	$0.611_{(0.018)}$	$0.775_{(0.013)}$	$0.766_{(0.023)}$	$0.915_{(0.008)}$
	2	$0.432_{(0.032)}$	$0.676_{(0.019)}$	$0.651_{(0.034)}$	$0.824_{(0.021)}$
0.75	1	$0.661_{(0.025)}$	$0.842_{(0.008)}$	$0.845_{(0.011)}$	$0.938_{(0.005)}$
	2	$0.551_{(0.025)}$	$0.732_{(0.018)}$	$0.714_{(0,037)}$	$0.878_{(0.012)}$

Unequal variance cases: $k = 2$ and $p = 6$

Cases			$\pi = 0.5$		$\pi = 0.3$	
ρ	σ_1	σ_2	$n = 200$	$n = 1000$	$n = 200$	$n = 1000$
0.5	1.5	2.5	$0.523_{(0.028)}$	$0.775_{(0.012)}$	$0.681_{(0.026)}$	$0.8468_{(0.014)}$
	1.5	3.5	$0.616_{(0.030)}$	$0.760_{(0.0017)}$	$0.583_{(0.029)}$	$0.757_{(0.022)}$
0.75	1.5	2.5	$0.614_{(0.028)}$	$0.818_{(0.011)}$	$0.667_{(0.020)}$	$0.890_{(0.009)}$
	1.5	3.5	$0.636_{(0.024)}$	$0.827_{(0.013)}$	$0.660_{(0.032)}$	$0.826_{(0.018)}$

$15\% - 20\%$ compared to the RAND indexes with the small sample. The standard errors of survival rates are also smaller under large sample.

From the Figure 4.3 and 4.4, we can see that the dispersion degrees of RAND indexes with the small sample is much bigger than the one with the large sample under all scenarios, which means the model with large sample size has higher stability. Therefore, the more reliable clustering results can be expected when the sample size are larger.

Overall, when the distinction between two groups are not clear, the GM model

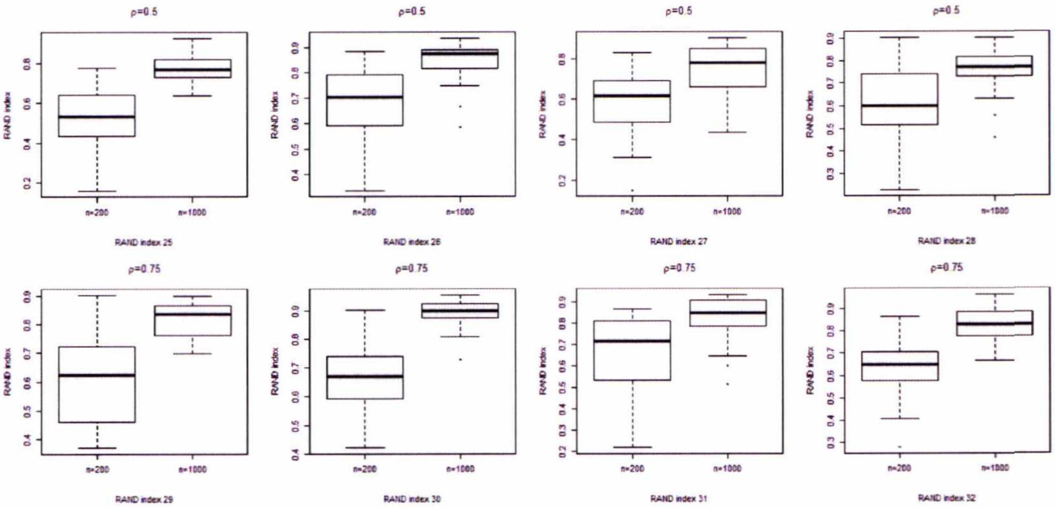

Figure 4.4: Rand indexed of GM regression model with 2 unequal variance components when the distance of correlation coefficients between the 2 groups is relatively small.

with large sample size obviously has better performance than GM model with relatively small sample size in all situations, and the advantage are more outstanding in this simulation compared with the last one.

Simulation 2.2 (Three and four components)

The above two simulations show the effects of RAND indexes under several scenarios when there are only 2 components of the Gaussian mixture model. In this simulation we consider more complicated situations that when the Gaussian mixture model with 3 and 4 components receptively, under the following scenarios: 1. Component proportions are different; 2. The correlation structure of covariates are different.

In the case that the mixture model has components, we let the correlation coefficients are $\boldsymbol{\beta}_1 = (0,3,3,3,3,3,3)^T$, $\boldsymbol{\beta}_2 = (0,1.5,1.5,1.5,1.5,1.5,1.5)^T$ and $\boldsymbol{\beta}_3 = (0,-1,-1,-1,-1,-1,-1)^T$ respectively, and consider the different combinations with the rest parameters: $(\sigma_1, \sigma_2, \sigma_3) : (1,1,1), (2,2,2)$, $\rho : 0.5, 0.75$,

$(\pi_1, \pi_2, \pi_3) : (0.2, 0.3, 0.5), (0.4, 0.2, 0.2)$. When the number of components are 4, we let $\boldsymbol{\beta}_1 = (0, 3, 3, 3, 3, 3, 3)^T$, $\boldsymbol{\beta}_2 = (0, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5)^T$, $\boldsymbol{\beta}_3 = (0, 1, 1, 1, 1, 1, 1)^T$ and $\boldsymbol{\beta}_4 = (0, -1, -1, -1, -1, -1, -1)^T$, and consider the different combinations with the rest parameters: $(\sigma_1, \sigma_2, \sigma_3, \sigma_4) : (1, 1, 1, 1), (2, 2, 2, 2)$, $\rho : 0.5, 0.75$, $(\pi_1, \pi_2, \pi_3, \pi_4) : (0.1, 0.2, 0.3, 0.4), (0.25, 0.25, 0.25, 0.25)$. The results of averaged RAND indexes under the different scenarios are summarized in Table 4.4. Figure 4.5 shows the box plots when the number of components is 3, in this Figure, from the left to the right the following cases are considered: $((\pi_1, \pi_2, \pi_3) = (0.2, 0.3, 0.5)$, $\rho = 0.5)$, $((\pi_1, \pi_2, \pi_3) = (0.2, 0.3, 0.5)$,$\rho = 0.75)$, $((\pi_1, \pi_2, \pi_3) = (0.4, 0.2, 0.2)$,$\rho = 0.5)$, $((\pi_1, \pi_2, \pi_3) = (0.4, 0.2, 0.2)$, $\rho = 0.75)$. While Figure 4.6 shows the box plots when the number of components is 4, in this Figure, from the left to the right the following cases are considered: $((\pi_1, \pi_2, \pi_3, \pi_4) = (0.1, 0.2, 0.3, 0.4)$, $\rho = 0.5)$, $((\pi_1, \pi_2, \pi_3, \pi_4) = (0.1, 0.2, 0.3, 0.4)$,$\rho = 0.75)$, $((\pi_1, \pi_2, \pi_3, \pi_4) = (0.25, 0.25, 0.25, 0.25)$,$\rho = 0.5)$, $((\pi_1, \pi_2, \pi_3, \pi_4) = (0.25, 0.25, 0.25, 0.25)$, $\rho = 0.75)$.

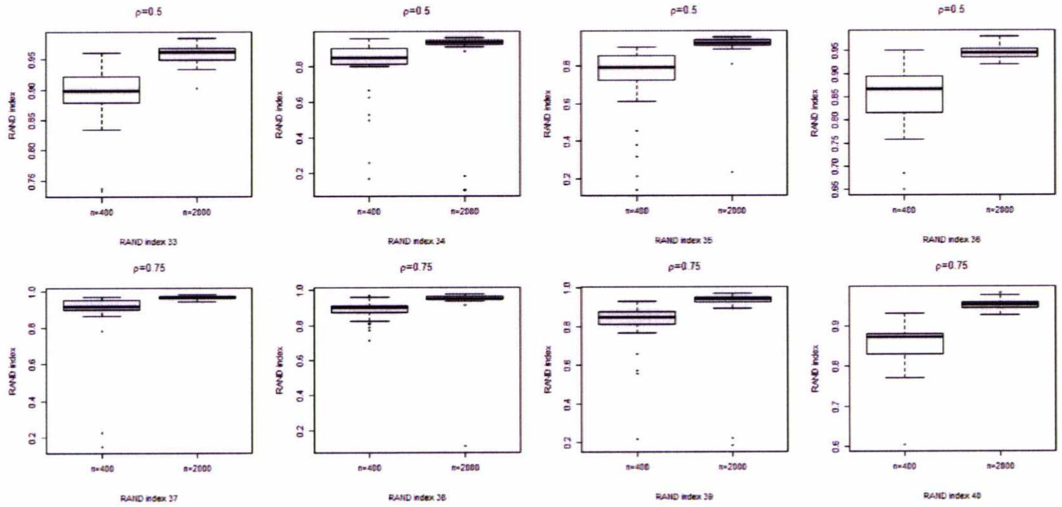

Figure 4.5: Rand indexed of GM regression model with 3 components.

From Table 4.4 we can see, on one hand, the accuracy of clustering by using RAND indexes of GM model with a large sample is much better than the one with a small sample, and the standard errors of survival rates are also smaller

Table 4.4: Averaged RAND indexes for GM regression model with 3 and 4 components. (The numbers in the parenthesis are the standard errors of the survival rates)

Equal variance cases: $k = 3$ and $p = 6$

Cases		$\pi = (0.2, 0.3, 0.5)$		$\pi = (0.4, 0.4, 0.2)$	
ρ	σ	$n = 400$	$n = 2000$	$n = 400$	$n = 2000$
0.50	1	$0.891_{(0.010)}$	$0.958_{(0.003)}$	$0.794_{(0.035)}$	$0.859_{(0.045)}$
	2	$0.849_{(0.013)}$	$0.946_{(0.002)}$	$0.718_{(0.037)}$	$0.898_{(0.023)}$
0.75	1	$0.869_{(0.035)}$	$0.966_{(0.002)}$	$0.884_{(0.011)}$	$0.926_{(0.028)}$
	2	$0.857_{(0.011)}$	$0.953_{(0.003)}$	$0.808_{(0,0026)}$	$0.890_{(0.034)}$

Equal variance cases: $k = 4$ and $p = 6$

Cases		$\pi = (0.1, 0.2, 0.3, 0.4)$		$\pi = (0.25, 0.25, 0.25, 0.25)$	
ρ	σ	$n = 400$	$n = 2000$	$n = 400$	$n = 2000$
0.50	1	$0.712_{(0.028)}$	$0.841_{(0.041)}$	$0.620_{(0.033)}$	$0.828_{(0.025)}$
	2	$0.618_{(0.026)}$	$0.835_{(0.026)}$	$0.538_{(0.029)}$	$0.776_{(0.012)}$
0.75	1	$0.748_{(0.036)}$	$0.922_{(0.004)}$	$0.720_{(0.026)}$	$0.852_{(0.025)}$
	2	$0.676_{(0.019)}$	$0.859_{(0.024)}$	$0.590_{(0,023)}$	$0.835_{(0.006)}$

for GM model with large sample for all cases. The table also shows that the less the number of components are, the better the performance it has on clustering. On the other hand, the improvement of clustering between large data and small data are bigger when it has more components.

Difference between the results of box plots is shown in the last simulations, the Figure 4.5 and 4.6 illustrate that although the degrees of dispersion of RAND indexes for the small sample are higher than the ones for large sample, the difference of the stability of the GM model between two samples is less obvious.

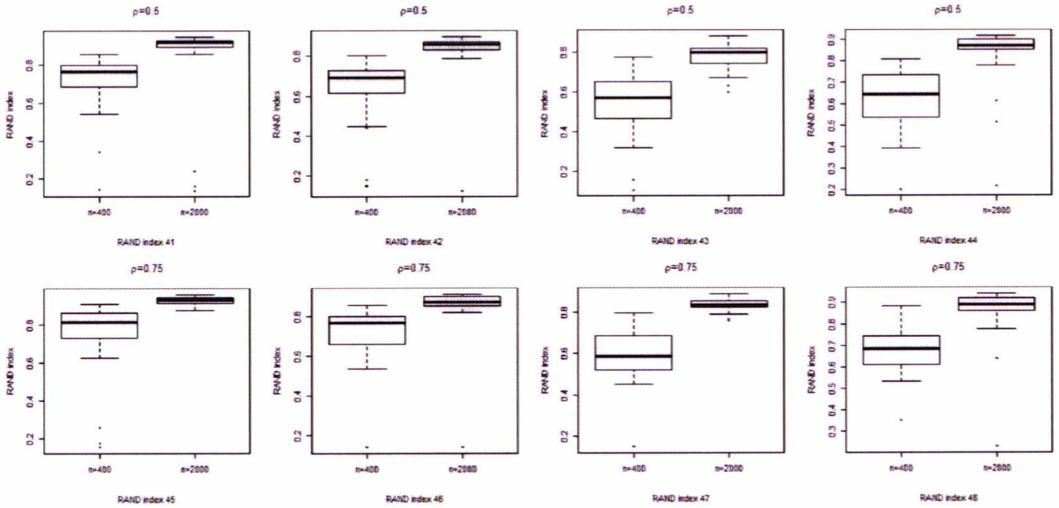

Figure 4.6: Rand index of GM regression model with 4 components.

Generally speaking, all our simulation results show that, the GM model with a large sample gives a better clustering prediction compared to the model with a small sample in all scenarios. However the accuracy of clustering changes under different situations.

Chapter 5

Maximum Likelihood Estimation of Exponential Power Mixture Regressions

In the Gaussian mixture regression modelling, the conditional distribution of the response variable given covariates is assumed to be Gaussian. As we mentioned before, this may be invalid in practice. Here, to tackle the problem, we propose exponential power mixture regression models to improve the model-fit to the data by relaxing the restrictions on the shape parameters in Gaussian mixture regression models.

In this chapter, first of all, we build an exponential power mixture model to reflect grouping structures in the data. Then we develop the identifiability of the model followed by present the details on how to calculate the maximum likelihood estimators of the above model by using the EM algorithm. The novelty lies in that we convert a regression estimation problem to a least square problem. Next, we conduct three simulations to assess the performance of maximum likelihood estimation of exponential power regression in estimating and clustering, under the following scenarios: 1. the data are drawn from a Gaussian mixture regression model; 2. the data are drawn from an exponential power mixture regression model; 3. the data with clumpy correlations, which means they are neither drawn from a Gaussian mixture nor exponential power mixture of regression model. All results are compared with maximum likelihood estimation of Gaussian mixture regressions. At the end of this chapter we apply our method on the gene expression and motifs dataset.

5.1 Methodology

Similar to Chapter 4, there are 2 steps in maximum likelihood estimation(MLE) of the exponential power mixture model of regression by using the EM algorithm: In the E-step, the missing data are estimated by using the observed data and current estimate of the model parameters. In the M-step, the expected complete likelihood function of the exponential power mixture of regression model is maximized under the assumption that the missing data are known.

Suppose we have an independent sample $(y_i, \mathbf{x}_i)$, $1 \leq i \leq n$, where conditional on $\mathbf{x}_i$, y_i drawn from the following finite exponential power mixture regression model with K components,

$$f(y_i|\mathbf{x}_i, \mathbf{\Psi}) = \sum_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k).$$

where the parameter vector $\Psi = (\pi_1, \dots, \pi_{K-1}, \boldsymbol{\zeta}^T)^T$ is the vector containing all the unknown parameters in this mixture model, and $\boldsymbol{\zeta} = (\boldsymbol{\theta}_1^T, \dots, \boldsymbol{\theta}_K^T)^T$. $\boldsymbol{\theta}_k = (\boldsymbol{\beta}_k, \alpha_k, \sigma_k)^T$, $k = 1, \dots, K$.

In the exponential power mixture regression model, the density of the k^{th} component can be written as

$$f_k\left(y_i\left|\mathbf{x}_i, \boldsymbol{\theta}_k\right.\right) = \frac{\alpha_k}{2\sigma_k \Gamma\left(1/\alpha_k\right)} \exp\left(-\frac{\left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_k\right|^{\alpha_k}}{(\sigma_k^2)^{\alpha_k/2}}\right).$$

Recall that the latent variable is

$$z_{ik} = (z_i)_k = \begin{cases} 1 & y_i \in \text{the } k\text{th component} \\ 0 & y_i \notin \text{the } k\text{th component} \end{cases}.$$

In the EM framework, the observed-data $\mathbf{D} = (y_1, \mathbf{x}_1; \ldots; y_n, \mathbf{x}_n)^T$ are viewed as an incomplete data. We combine it with the indicator z_{ik} to form the complete-data

$$\mathbf{D}_c = (\mathbf{D}, \mathbf{z}).$$

The complete-data log-likelihood for $\boldsymbol{\Psi}$ can be written in the following form

$$\log L_c(\boldsymbol{\Psi}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik}\{\log \pi_k + \log f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)\}. \tag{5.1}$$

5.1.1 E step:

Similar as shown in the previous Chapter, we calculate the expectation of the complete-data log-likelihood based on the current value of $\boldsymbol{\Psi}^{(s)}$, where $\boldsymbol{\Psi}^{(s)}$ stand for the value of all parameters after the s^{th} iteration.

$$\begin{aligned} Q\left(\Psi|\Psi^{(s)}\right) &= E_{\Psi^{(s)}}\left(\log L_c(\boldsymbol{\Psi})\,|\mathbf{y}, \mathbf{X}\right) \\ &= E_{\Psi^{(s)}}\left\{\left[\sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\left(\log\left(\pi_k\right) + \log f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)\right)\right]|\mathbf{y}, \mathbf{X}\right\} \\ &= E_{\Psi^{(s)}}\left\{\sum_{i=1}^{n}\sum_{k=1}^{K}\left(z_{ik}\,|\mathbf{y}, \mathbf{X}\right)\left[\log\left(\pi_k\right) + \log f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)\right]\right\} \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} E_{\Psi^{(s)}}\left(z_{ik}\,|\mathbf{y}, \mathbf{X}\right)\left[\log\left(\pi_k\right) + \log f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)\right] \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})\left[\log\left(\pi_k\right) + \log f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k)\right] \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \log\left(\pi_k\right) + \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k\left(y_i, \Psi^{(s)}\right) \log f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k) \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \log\left(\pi_k\right) + \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k\left(y_i, \Psi^{(s)}\right) \log \frac{\alpha_k}{2\sigma_k \Gamma\left(1/\alpha_k\right)} \\
&\quad - \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \frac{\left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_k\right|^{\alpha_k}}{(\sigma_k^2)^{\alpha_k/2}} \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \log\left(\pi_k\right) + A,
\end{aligned}
$$

where

$$
A = \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \log \frac{\alpha_k}{2\sigma_k \Gamma\left(1/\alpha_k\right)} - \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \frac{\left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_k\right|^{\alpha_k}}{(\sigma_k^2)^{\alpha_k/2}}.
$$

Given the observation data $(\mathbf{y}, \mathbf{X})$, the expectation of latent vector $\mathbf{z}$

$$
\begin{aligned}
E_{\Psi^{(s)}}\left(z_{ik}\left|\mathbf{y}, \mathbf{X}\right.\right) &= P_{\Psi^{(s)}}\left\{z_{ik}\left|\mathbf{y}, \mathbf{X}\right.\right\} \\
&= \tau_k\left(y_i|\mathbf{x}_i, \Psi^{(s)}\right) \\
&= \frac{\pi_k^{(s)} f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k^{(s)})}{\sum\limits_{t=1}^{K} \pi_t^{(s)} f_t(y_i|\mathbf{x}_i, \boldsymbol{\theta}_t^{(s)})} \\
&= \frac{\pi_k^{(s)} f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k^{(s)})}{f\left(y\left|\mathbf{x}, \boldsymbol{\Psi}^{(s)}\right.\right)}.
\end{aligned}
$$

5.1.2 M step:

(?)

by calculating the maximization of the expectation of the log-likelihood. In this step we estimate the parameters Ψ_t on the $(s+1)^{th}$ iteration, based on the parameters obtained from the s^{th} iteration. For this purpose, we maximise the expectation of the complete-data log-likelihood from the last step.

Estimate π_t

To update the value of π_t, we define

$$\begin{aligned}
B &= Q\left(\Psi|\Psi^{(s)}\right) - \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right) \\
&= \sum_{i=1}^{n}\sum_{k\neq t}^{K}\tau_k(y_i|\mathbf{x}_i,\boldsymbol{\Psi}^{(s)})\log\left(\pi_k\right) + \sum_{i=1}^{n}\sum_{k\neq t}^{K}\tau_k(y_i|\mathbf{x}_i,\boldsymbol{\Psi}^{(s)})\log\frac{\alpha_k}{2\sigma_k\Gamma\left(1/\alpha_k\right)} \\
&\quad -\sum_{i=1}^{n}\sum_{k\neq t}^{K}\tau_k(y_i|\mathbf{x}_i,\boldsymbol{\Psi}^{(s)})\frac{\left|y_i-\mathbf{x}_i^T\boldsymbol{\beta}_k\right|^{\alpha_k}}{(\sigma_k^2)^{\alpha_k/2}} + \sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\boldsymbol{\Psi}^{(s)})\log\left(\pi_t\right) \\
&\quad +\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\boldsymbol{\Psi}^{(s)})\log\frac{\alpha_t}{2\sigma_t\Gamma\left(1/\alpha_t\right)} - \sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\boldsymbol{\Psi}^{(s)})\frac{\left|y_i-\mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\alpha_t}}{(\sigma_t^2)^{\alpha_t/2}} \\
&\quad -\lambda\left(\sum_{k=1}^{K}\pi_k - 1\right).
\end{aligned}$$

Then, we differentiate B with respect to π_t and set it to be zero, which gives $\lambda = n$, and $\hat{\pi}_t$ on the $(s+1)^{th}$ step is

$$\hat{\pi}_t^{(s+1)} = \frac{\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\boldsymbol{\Psi}^{(s)})}{n}. \tag{5.2}$$

Estimate $\boldsymbol{\beta}_t$

There are two ways to estimate $\boldsymbol{\beta}_t$, the first one is to convert the problem to a least square estimation problem, and the second way is to use the Newton-Raphson method. The details are as follows.

Least square approach In order to estimate $\boldsymbol{\beta}_t^{(s+1)}$, we derive the expectation of the complete-data log-likelihood function, A, with respect to $\boldsymbol{\beta}_t$ and set it equals to 0, we have

$$
\begin{aligned}
\frac{\partial A}{\partial \boldsymbol{\beta}_t} &= -\sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t^2)^{\alpha_t/2}} \frac{\partial \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\alpha_t}}{\partial \boldsymbol{\beta}_t} \\
&= -\sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t)^{\alpha_t}} \frac{\partial \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\alpha_t}}{\partial \boldsymbol{\beta}_t} \\
&= -\sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t)^{\alpha_t}} \alpha_t \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\left(\alpha_t-1\right)} (-\mathbf{x}_i) \\
&= -\sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t)^{\alpha_t}} \alpha_t \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\left(\alpha_t-2\right)} \left(\mathbf{x}_i y_i - \mathbf{x}_i\mathbf{x}_i^T\boldsymbol{\beta}_t\right) = 0.
\end{aligned}
$$

We re-arrange the equation to get:

$$
\begin{aligned}
&\sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t)^{x_t}} \alpha_t \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\left(\alpha_t-2\right)} \mathbf{x}_i y_i \\
= &\sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t)^{\alpha_t}} \alpha_t \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\left(\alpha_t-2\right)} \mathbf{x}_i\mathbf{x}_i^T\boldsymbol{\beta}_t.
\end{aligned}
$$

By solving the above equation, we get the estimated $\hat{\boldsymbol{\beta}}_t$ on the $(s+1)^{th}$ iteration

$$
\hat{\boldsymbol{\beta}}_t^{(s+1)} = \frac{\sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t)^{\alpha_t}} \alpha_t \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\left(\alpha_t-2\right)} \mathbf{x}_i y_i}{\sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t)^{\alpha_t}} \alpha_t \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\left(\alpha_t-2\right)} \mathbf{x}_i\mathbf{x}_i^T}.
$$

Let $\mathbf{W}_t$ be a square matrix where it has the entries of $\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\left(x_t-2\right)}$, $i = (1, ..., n)$, on the main diagonal and zeroes elsewhere:

$$
\mathbf{W}_t = \begin{pmatrix}
\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \left|y_1 - x_1^T\boldsymbol{\beta}_t\right|^{\left(\alpha_t-2\right)} & 0 & 0 \\
0 & \ddots & 0 \\
0 & 0 & \tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \left|y_n - x_n^T\boldsymbol{\beta}_t\right|^{\left(\alpha_t-2\right)}
\end{pmatrix}.
$$

We define $\mathbf{X}^* = \mathbf{W}^{1/2}\mathbf{X}$, and $\mathbf{y}^* = \mathbf{W}^{1/2}\mathbf{y}$, hence, we converted the generalised regression estimation problem to a least square problem, such that

$$\hat{\boldsymbol{\beta}}_t^{(s+1)} = (\mathbf{X}^{*T}\mathbf{X}^*)^{-1}\mathbf{X}^{*T}\mathbf{y}^*.$$

Newton-Raphson approach We can also use Newton-Raphson method for finding successively better approximations to $\boldsymbol{\beta}_t$ of function A.

Recall that,

$$A = \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \log \frac{\alpha_k}{2\sigma_k\Gamma\left(1/\alpha_k\right)} - \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \frac{\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_k\right|^{\alpha_k}}{(\sigma_k^2)^{\alpha_k/2}}.$$

The first derivative of the function A with respect to $\boldsymbol{\beta}_t$ is:

$$\begin{aligned}
\frac{\partial A}{\partial \boldsymbol{\beta}_t} &= -\sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t^2)^{\alpha_t/2}} \frac{\alpha_t}{2} \times 2\left(\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^2\right)^{(\alpha_t/2)-1} \left(y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right)\left(-\mathbf{x}_i^T\right) \\
&= \alpha_t \sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t^2)^{\alpha_t/2}} \left(\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^2\right)^{(\alpha_t/2)-1} \left(y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right)\left(\mathbf{x}_i^T\right),
\end{aligned}$$

and the second derivative of A is:

$$\begin{aligned}
\frac{\partial^2 A}{\partial \boldsymbol{\beta}_t \partial \boldsymbol{\beta}_t^T} &= \alpha_t \sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t^2)^{\alpha_t/2}} \left(\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^2\right)^{(\alpha_t/2)-1} \left(-\mathbf{x}_i^T\right)\left(\mathbf{x}_i^T\right) \\
&\quad + \alpha_t \sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t^2)^{\alpha_t/2}} \left(\frac{\alpha_t}{2} - 1\right) \\
&\quad \times 2\left(\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^2\right)^{(\alpha_t/2)-2} \left(y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right)^2 \left(\mathbf{x}_i^T\right)\left(-\mathbf{x}_i\right)
\end{aligned}$$

$$= \quad -\alpha_t \frac{\tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)})}{(\sigma_t^2)^{\alpha_t/2}} \sum_{i=1}^{n} \left(\left(\left| y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t \right|^2 \right)^{(\alpha_t/2)-1} \left(\mathbf{x}_i \mathbf{x}_i^T \right) \left(\mathbf{x}_t - 1 \right) \right).$$

Therefore, by implanting the Newton-Raphson method, we have:

$$\begin{aligned} \hat{\boldsymbol{\beta}}_t^{(s+1)} &= \boldsymbol{\beta}_t^{(s)} - \left(\frac{\partial^2 A}{\partial \boldsymbol{\beta}_t \partial \boldsymbol{\beta}_t^T} \right)^{-1} \left(\frac{\partial A}{\partial \boldsymbol{\beta}_t} \right) \\ &= \boldsymbol{\beta}_t^{(s)} + (\alpha_t - 1)^{-1} \left[\sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)})}{(\sigma_t^2)^{\alpha_t/2}} \left| y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t \right|^{\alpha_t - 2} \mathbf{x}_i \mathbf{x}_i^T \right]^{-1} \\ &\quad \times \sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)})}{(\sigma_t^2)^{\alpha_t/2}} \left| y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t \right|^{\alpha_t - 2} \left(y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t \right) \mathbf{x}_i. \end{aligned}$$

Both the least square and the Newton-Raphson approach give the same result of $\hat{\boldsymbol{\beta}}_t^{(s+1)}$.

Estimate σ_t^2

Similarly as above, to estimate σ_t^2, we calculate the derivative of the function A with respect to $(1/\sigma_t^2)$ and set it equals to 0.

We have:

$$\begin{aligned} \frac{\partial A}{\partial(\sigma_t^{(-2)})} &= \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \frac{\partial(\log(1/\sigma_t))}{\partial(1/\sigma_t^2)} \\ &\quad - \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left| y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t \right|^{\alpha_t} \frac{\alpha_t}{2} \left(1/\sigma_t^2 \right)^{(\alpha_t/2)-1} \\ &= \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \frac{\partial[1/2 \log(1/\sigma_t^2)]}{\partial(1/\sigma_t^2)} \\ &\quad - \frac{\alpha_t}{2} \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left| y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t \right|^{\alpha_t} \left(1/\sigma_t^2 \right)^{(\alpha_t/2)-1} \end{aligned}$$

$$\begin{aligned} &= \sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)})\sigma_t^2}{2} \\ &\quad -\frac{\alpha_t}{2}\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\alpha_t} \left(1/\sigma_t^2\right)^{(\alpha_t/2)-1} \\ &= 0. \end{aligned}$$

By re-arranging the equation, we have:

$$\begin{aligned} \sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)})}{2}\sigma_t^2 &= \frac{\alpha_t}{2}\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\alpha_t} \left(1/\sigma_t^2\right)^{(\alpha_t/2)-1} \\ \sigma_t^2(\sigma_t^2)^{\alpha_t/2-1} &= \left(\sigma_t^2\right)^{\alpha_t/2} = \frac{\alpha_t \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\alpha_t}}{\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)})}. \end{aligned}$$

Hence, the estimated σ_t^2 on the $(s+1)^{th}$ iteration, $(\sigma^2)^{(s+1)}$, can be written in the following form:

$$(\hat{\sigma}_t^2)^{(s+1)} = \left(\frac{\alpha_t \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\alpha_t}}{\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)})}\right)^{2/\alpha_t}. \tag{5.3}$$

Estimate α_t

As α_t can not take a negative value, we define a new variable $\eta_t = log(\alpha_t)$, where η can take values from $-\infty$ to ∞. We use Newton-Raphson method to update α_t as follows.

We can write A in the following form:

$$A = \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \log(\frac{\alpha_k}{2\sigma_k\Gamma\left(1/\alpha_k\right)}) - \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)})\frac{\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_k\right|^{\alpha_k}}{(\sigma_k^2)^{\alpha_k/2}}$$

$$
\begin{aligned}
= & \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \log(\frac{1}{2\sigma_k}) + \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \log(\frac{\alpha_k}{\Gamma\left(1/\alpha_i\right)}) \\
& - \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \frac{\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_k\right|^{\alpha_k}}{(\sigma_k^2)^{\alpha_k/2}}.
\end{aligned}
$$

The first derivative of the function A with respect to α_t is:

$$
\begin{aligned}
\frac{\partial A}{\partial \alpha_t} = & \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left(\frac{1}{\alpha_t} + \frac{\Gamma'\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)}\frac{1}{\alpha_t^2}\right) \\
& - \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left(\frac{\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|}{\sigma_t}\right)^{\alpha_t} \log\left(\frac{\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|}{\sigma_t}\right),
\end{aligned}
$$

and the second derivative of the function A with respect to α_t is:

$$
\begin{aligned}
\frac{\partial^2 A}{\partial \alpha_t^2} = & \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left(-\frac{1}{\alpha_t^2} - \frac{2}{\alpha_t^3}\frac{\Gamma'\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)} + \frac{1}{\alpha_t^2}\left(\frac{\Gamma''\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)}\left(-\frac{1}{\alpha_t^2}\right)\right.\right. \\
& \left.\left. + \left(\frac{\Gamma'\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)}\right)^2\left(-\frac{1}{\alpha_t^2}\right)\right)\right) \\
& - \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left(\frac{\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|}{\sigma_t}\right)^{\alpha_t} \left(\log\left(\frac{\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|}{\sigma_t}\right)\right)^2 \\
= & -\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left(\frac{1}{\alpha_t^2} + \frac{2}{\alpha_t^3}\frac{\Gamma'\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)} + \frac{1}{\alpha_t^4}\left(\frac{\Gamma''\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)} - \left(\frac{\Gamma'\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)}\right)^2\right)\right) \\
& - \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left(\frac{\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|}{\sigma_t}\right)^{\alpha_t} \left(\log\left(\frac{\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|}{\sigma_t}\right)\right)^2
\end{aligned}
$$

where

$$
\frac{\partial 1/\Gamma\left(1/\alpha_t\right)}{\partial \alpha_t} = \frac{\partial\left(\Gamma\left(1/\alpha_t\right)\right)^{-1}}{\partial \alpha_t} = \left(\Gamma\left(1/\alpha_t\right)\right)^{-2}\Gamma'\left(1/\alpha_t\right)\left(-\frac{1}{\alpha_t^2}\right).
$$

Note that

$$\alpha_t = e^{\eta_t},$$

which implies,

$$\frac{\partial \alpha_t}{\partial \eta_t} = \frac{\partial e^{\eta_t}}{\partial \eta_t} = e^{\eta_t} = \alpha_t.$$

By chain rule, the first derivative of the function A with respect to η_t is:

$$\frac{\partial A}{\partial \eta_t} = \frac{\partial A}{\partial \alpha_t}\frac{\partial \alpha_t}{\partial \eta_t} = \frac{\partial A}{\partial \alpha_t}\alpha_t,$$

and the second derivative of the function A with respect to η_t is:

$$\begin{aligned}
\frac{\partial^2 A}{\partial \eta_t^2} &= \frac{\partial}{\partial \alpha_t}\left(\frac{\partial A}{\partial \eta_t}\right)\alpha_t + \frac{\partial A}{\partial \alpha_t}\frac{\partial \alpha_t}{\partial \eta_t} \\
&= \frac{\partial}{\partial \alpha_t}\left(\frac{\partial A}{\partial \alpha_t}\frac{\partial \alpha_t}{\partial \eta_t}\right)\alpha_t + \frac{\partial A}{\partial \alpha_t}\frac{\partial \alpha_t}{\partial \eta_t} \\
&= \frac{\partial^2 A}{\partial \alpha_t^2}\frac{\partial \alpha_t}{\partial \eta_t}\alpha_t + \frac{\partial A}{\partial \alpha_t}\frac{\partial \alpha_t}{\partial \eta_t} \\
&= \frac{\partial^2 A}{\partial \alpha_t^2}\alpha_t^2 + \frac{\partial A}{\partial \alpha_t}\alpha_t \\
&= \left(\frac{\partial^2 A}{\partial \alpha_t}\alpha_t + \frac{\partial A}{\partial \alpha_t}\right)\alpha_t,
\end{aligned}$$

which implies,

$$\frac{\partial^2 A}{\partial \eta_t \partial \eta_t^T} = \begin{pmatrix} \left(\frac{\partial^2 A}{\partial \alpha_1}\alpha_1^2 + \frac{\partial A}{\partial \alpha_1}\right)\alpha_1 & \cdots & \frac{\partial^2 A}{\partial \alpha_K \partial \alpha_1}\alpha_K\alpha_1 \\ \frac{\partial^2 A}{\partial \alpha_t \partial \alpha_k}\alpha_t\alpha_k & \ddots & \frac{\partial^2 A}{\partial \alpha_t \partial \alpha_k}\alpha_t\alpha_k \\ \frac{\partial^2 A}{\partial \alpha_1 \partial \alpha_g}\alpha_1\alpha_K & \cdots & \left(\frac{\partial^2 A}{\partial \alpha_K^2}\alpha_K + \frac{\partial A}{\partial \alpha_K}\right)\alpha_K \end{pmatrix},$$

for $k \neq t$, where

$$\frac{\partial^2 A}{\partial \eta_t \partial \eta_k} = \frac{\partial^2 A}{\partial \alpha_t \partial \alpha_k} \alpha_t \alpha_k = 0.$$

By applying the Newton-Raphson method

$$\boldsymbol{\eta} = \begin{pmatrix} \eta_1 \\ \vdots \\ \eta_K \end{pmatrix}^{(s+1)} = \begin{pmatrix} \eta_1 \\ \vdots \\ \eta_K \end{pmatrix}^{(s)} - \left(\frac{\partial^2 A}{\partial \boldsymbol{\eta} \partial \boldsymbol{\eta}^T} \right) \begin{pmatrix} \frac{\partial A}{\partial \alpha_1} \alpha_1 \\ \vdots \\ \frac{\partial A}{\partial \alpha_K} \alpha_K \end{pmatrix},$$

the updated η_t is

$$\eta_t^{(s+1)} = \eta_t^s - \left(\frac{\partial^2 A}{\partial \alpha_t^2} \alpha_t + \frac{\partial A}{\partial \alpha_t} \right)^{-1} \frac{\partial A}{\partial \alpha_t} \Big|_{\alpha_t = \alpha_t^{(s)}} .$$

Hence, the t^{th} component of the estimated shape parameter α, $\hat{\alpha_t}$, after the $(s+1)^{th}$ iteration is:

$$\hat{\alpha_t}^{(s+1)} = \alpha_t^{(s)} \exp \left(- \left(\frac{\partial^2 A}{\partial \alpha_t^2} \alpha_t + \frac{\partial A}{\partial \alpha_t} \right)^{-1} \frac{\partial A}{\partial \alpha_t} \Big|_{\alpha_t = \alpha_t^{(s)}} \right),$$

for $1 \leqslant t \leqslant K$.

To avoid extreme values of α_t, we added a small constraint value r, such that

$$\alpha_t - r = e^{\eta_t}.$$

The first and second derivative of A with respect to η_t becomes:

$$\begin{aligned} \frac{\partial A}{\partial \eta_t} &= \frac{\partial A}{\partial \alpha_t} (\alpha_t - r) \\ \frac{\partial^2 A}{\partial \eta_t^2} &= \frac{\partial^2 A}{\partial \alpha_t^2} (\alpha_t - r)^2 + \frac{\partial A}{\partial \alpha_t} (\alpha_t - r), \end{aligned}$$

therefore, the updated α_t on the $(s+1)^{th}$ iteration is:

$$\begin{aligned}\hat{\alpha}_t^{(s+1)} &= r + (\alpha_t^{(s)} - r)e^{-\left(\frac{\partial^2 A}{\partial \alpha_t^2}(\alpha_t - r)^2 + \frac{\partial A}{\partial \alpha_t}(\alpha_t - r)\right)^{-1} \frac{\partial A}{\partial \alpha_t}(\alpha_t - r)\Big|_{\alpha_t = \alpha_t^{(s)}}} \\ &= r + (\alpha_t^{(s)} - r)e^{-\left(\frac{\partial^2 A}{\partial \alpha_t^2}(\alpha_t - r) + \frac{\partial A}{\partial \alpha_t}\right)^{-1} \frac{\partial A}{\partial \alpha_t}\Big|_{\alpha_t = \alpha_t^{(s)}}},\end{aligned}$$

for $1 \leqslant t \leqslant K$.

5.2 Theoretical Property

5.2.1 Identifiability

In statistics, identifiability is a property which a model must satisfy in order for inference to be possible. That means different values of the parameter must generate different probability distributions of the observations. Suppose we have an independent sample $(y_i, \mathbf{x}_i)$, $1 \leq i \leq n$, where conditional on $\mathbf{x}$, y_i drawn from a exponential power mixture regression model $\{f(y|\mathbf{x}, \boldsymbol{\beta}, \sigma, \alpha)\}$ where $\mathbf{X} = (1, x_1, ..., x_n)^T$, and $\mathbf{x}_i = (1, x_{i1}, \ldots, x_{iP})^T = (1, \mathbf{x}_i^*)^T$, $i = 1, \ldots, n$, $\boldsymbol{\beta} \in (-\infty, \infty), \sigma^2 \in (0, \infty), \alpha \in (0, \infty)$, is said identifiable if the relation is of the form

$$\sum_{k=1}^{K} \pi_k f(y|\mathbf{x}, \boldsymbol{\beta}_k, \sigma_k^2, \alpha_k) = \sum_{k=1}^{K^*} \pi_k^* f(y|\mathbf{x}, \boldsymbol{\beta}_k^*, \sigma_k^{2*}, \alpha_k^*),\ y \in \mathbb{R}^1,\ \mathbf{x} \in \mathbb{R}^{p+1} \quad (5.4)$$

where K and K^* are positive integers. Note that $\sum_{k=1}^{K} \pi_k = \sum_{k=1}^{K^*} \pi_k^* = 1$ and $\pi_k > 0$, $1 \leq k \leq K$, $\pi_k^* > 0$, $1 \leq k \leq K^*$, implies that $K = K^*$, and that there exists a permutation ν on $1, 2, ..., K$ for any $\mathbf{x}$, such that $\mathbf{x}^T \boldsymbol{\beta}_k^* = \mathbf{x}^T \boldsymbol{\beta}_{\nu(k)}$, which implies $\boldsymbol{\beta}_k^* = \boldsymbol{\beta}_{\nu(k)}$, then we have $(\pi_k^*, \boldsymbol{\beta}_k^*, \sigma_k^{2*}, \alpha_k^*) = (\pi_{\nu(k)}, \boldsymbol{\beta}_{\nu(k)}, \sigma_{\nu(k)}^2, \alpha_{\nu(k)})$.

Theorem 5.1. *The above exponential power mixture regression models are identifiable.*

Proof. The poof of Theorem 5.1 follows by the theorem 1 in Zhang and Liang (2010), for mixture regression models, we use $\mathbf{x}^T\boldsymbol{\beta}$ instead of μ in (Zhang and Liang, 2010). □

5.3 Simulations

In this section, we assess the effect of data structure on the accuracy of the EM-based clustering and estimating of exponential power mixture regression models. The simulations are under the following scenarios: 1. the data are drawn from a Gaussian mixture regression model; 2. the data are drawn from a exponential power mixture regression model; 3. the data are drawn from a mixture regression model which is neither Gaussian distributed nor exponential power distributed. All results are compared with the EM-based clustering and estimating of Gaussian mixture regressions.

5.3.1 Simulation 1 (GM)

In the first simulation, we assess the accuracy of the EM-based clustering of exponential power mixture regressions for a 3-component Gaussian mixture regression model, and compare the results with the EM-based clustering of Gaussian mixture regressions.

Suppose we have a 3-component parametric mixture regression model:

$$f(y_i|\mathbf{x}_i, \Psi) = \sum_{k=1}^{3} \pi_k f_k(y_i|\mathbf{x}_i, \theta_k),$$

where each component is Gaussian distributed.

Let $\mathbf{X} = (\mathbf{x_1}, \ldots, \mathbf{x_{500}})^{\mathbf{T}}$ be the covariate matrix, and $\mathbf{x_i} = (1, x_{i1}, \ldots, x_{i3})^T = (1, \mathbf{x}_i^*)^T$, $i = 1, \ldots, 500$, where $\mathbf{x}_i^* = (x_{i1}, \ldots, x_{i3})^T$ denote the i^{th} observations on 3 covariates. We generated $\mathbf{x}_i^*$ from $N(0, 1)$. For $i = 1, \ldots, 500$, conditional on $\mathbf{x}_i$, we generated y_i from a 3-component Gaussian mixture regression model. Both $\mathbf{y}$ and the covariates in $\mathbf{X}$ were standardised. The interceptors

$\beta_{01} = \beta_{02} = \beta_{03} = 0$, and let $(\beta_{11}, \beta_{21}, \beta_{31}) \sim U[1, 2]$, $(\beta_{12}, \beta_{22}, \beta_{32}) \sim U[2, 3]$, $(\beta_{13}, \beta_{23}, \beta_{33}) \sim U[3, 4]$ respectively. The proportion of the 3 components was randomly selected with the constraint $\sum_{k=1}^{3} \pi_k = 1$, and the variance is 1 for all components. The real parameters of the model are summarised as below:

$$\begin{aligned}
(\pi_1, \pi_2, \pi_3) &= (0.509, 0.209, 0.282)\\
\begin{bmatrix} \beta_{01} & \beta_{02} & \beta_{03}\\ \beta_{11} & \beta_{12} & \beta_{13}\\ \beta_{21} & \beta_{22} & \beta_{23}\\ \beta_{31} & \beta_{32} & \beta_{33} \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0\\ 1.423 & 2.432 & 3.749\\ 1.695 & 2.380 & 3.415\\ 1.777 & 2.499 & 3.988 \end{bmatrix}\\
(\sigma_1, \sigma_2, \sigma_3) &= (1, 1, 1).
\end{aligned}$$

The Q-Q plot of observation $\mathbf{y}$ is shown in Figure 5.1. The Figure 5.1 also proved that the data is Gaussian distributed.

To set the initial values of parameters, we let $\pi_k \sim U(0, 1)$ with constraint $\sum_{k=1}^{3} \pi_k = 1$, $\beta_{jk} \sim U[1, 3]$, and the variances $\sigma_k^2 \sim U[1, 3]$, where $k = 1, 2, 3$ and $j = 1, \ldots, 4$.

Let $D = 100$ datasets. The GM implement Mclust gave a result of 2 components with proportions $\pi = (0.511, 0.489)$. By selecting the best fit, we compare BIC for the number of components $k = 1, 2, 3, 4, 5$. The results are summarised in the Table 5.1:

Table 5.1: Averaged value of AIC, BIC, EBIC and RAND indexs for data generated from GM regression model with dimension is 4.

	k	$Ave(\min BIC)$	$Ave(\min AIC)$	$Ave(\min EBIC)$	$Ave(RAND)$
GM	3	1650.273	1565.981	1563.857	0.750
	k	$Ave(\min BIC)$	$Ave(\min AIC)$	$Ave(\min EBIC)$	$Ave(RAND)$
EPD	3	1663.735	1566.799	1667, 319	0.729

From Table 5.1, we can see for the EM-based clustering of Gaussian mixture regressions, by fixing the shape parameters equal to 2, it gives the min BIC =

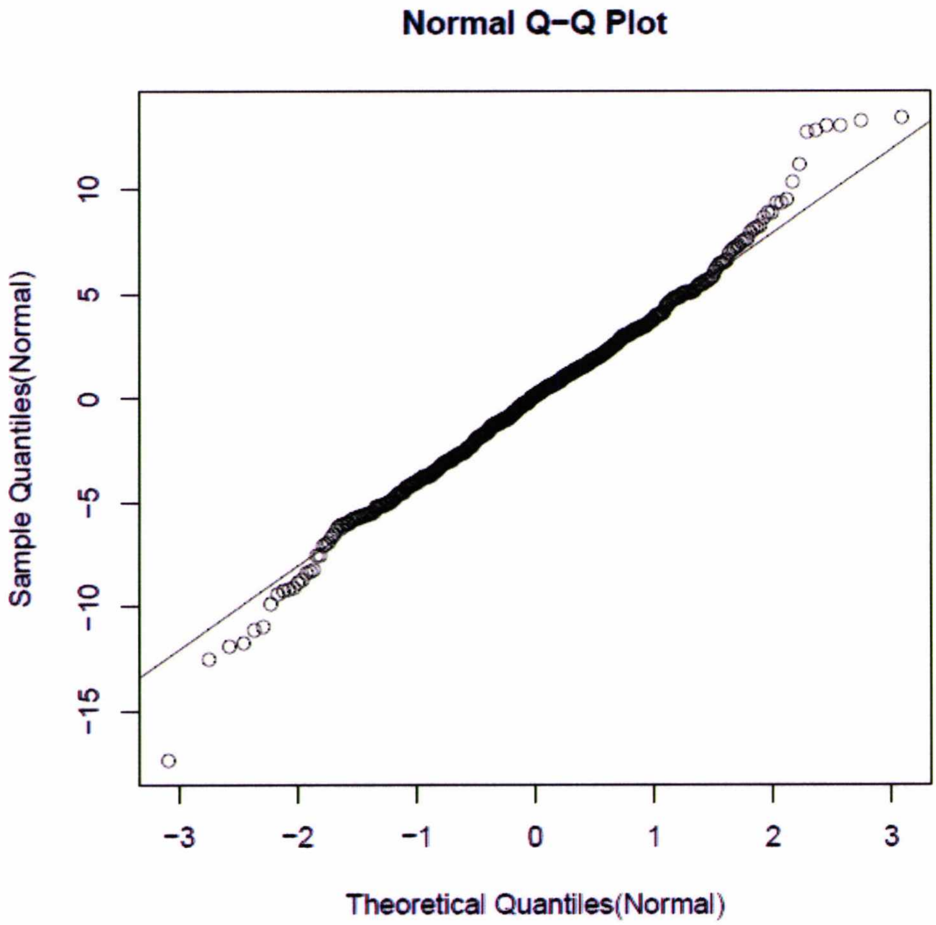

Figure 5.1: The Q-Q plot of observation **y** drawn from a GM mixture regression model.

1650.273(with corresponding $\min \text{AIC} = 1565.981$ and $\min \text{EBIC} = 1653.857$) when it has 3 components with the RAND index= 0.750. For the EM-based clustering of exponential power mixture regression, it reached its $\min BIC = 1663.735$(with corresponding $\min AIC = 1566.799$ and $\min EBIC = 1667.319$) when it has 3 components with the RAND index= 0.729. It shows that when the data are generated from a Gaussian mixture regression model, the EM-based clustering of Gaussian mixture regressions is more accurate than the EM-based clustering of exponential power mixture regressions.

We use the Q-Q plot to illustrate the residuals of each component after clustering by both EPD and GM respectively, shown in Figure 5.2 and 5.3.

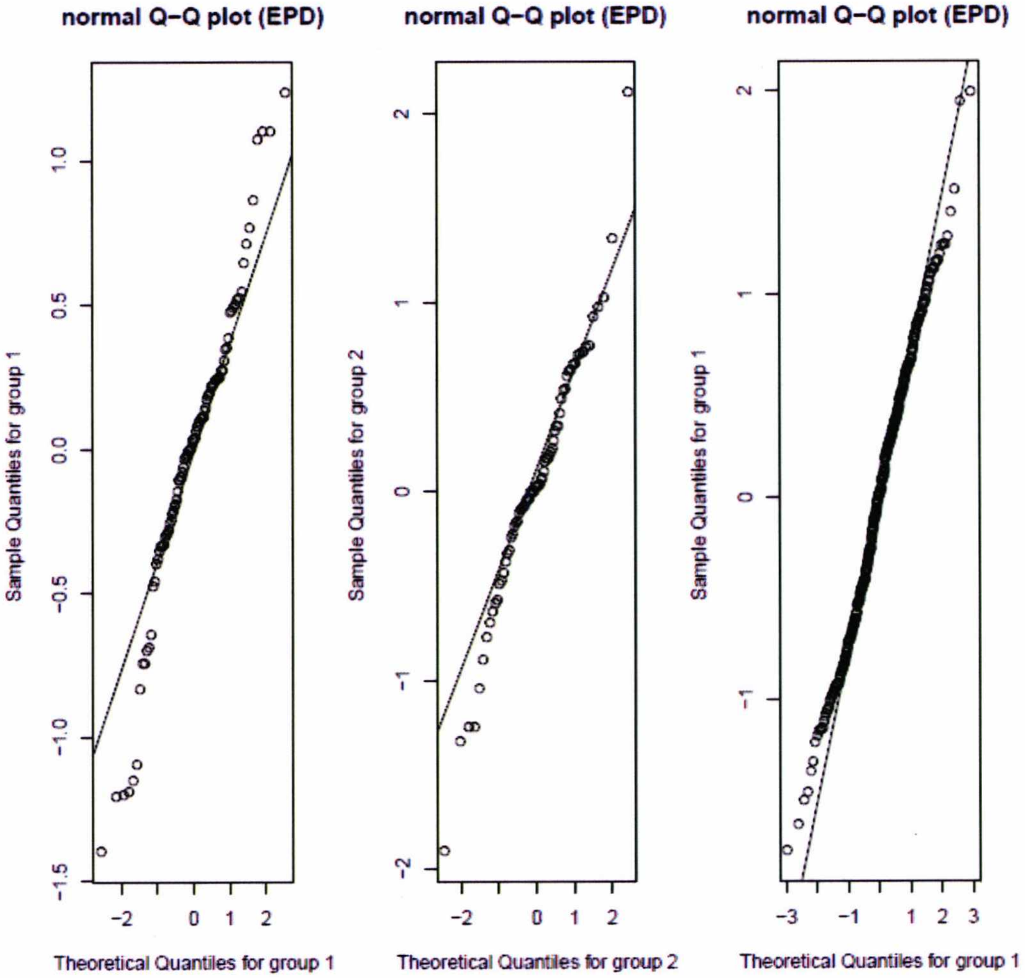

Figure 5.2: The Q-Q plot for the residuals after EPD based clustering.

Both Figure 5.2 and 5.3 show that the residuals of each component are Gaussian distributed.

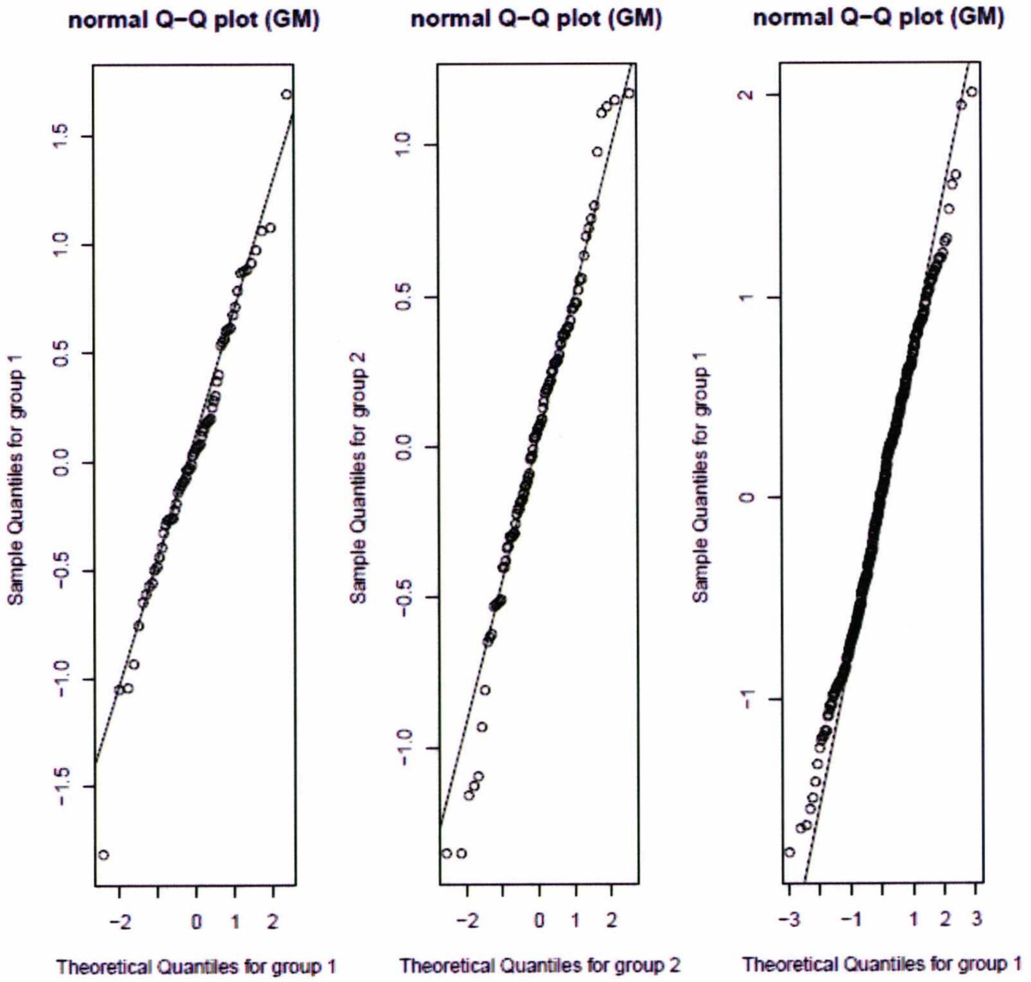

Figure 5.3: The Q-Q plot for the residuals after GM based clustering.

5.3.2 Simulation 2 (EPD)

In the second simulation, we assess the accuracy of the EM-based clustering of exponential power mixture regressions for a 2-component exponential power mixture regression model, and compare the results with the EM-based clustering of Gaussian mixture regressions.

Suppose we have a 2-component parametric mixture regression model:

$$f(y_i|\mathbf{x}_i, \mathbf{\Psi}) = \pi_1 f_1(y_i|\mathbf{x}_i, \theta_1) + \pi_2 f_2(y_i|\mathbf{x}_i, \theta_2),$$

where each component is exponential power distributed.

We use the same X that in the Simulation 1. For $i = 1, \ldots, 500$, conditional on $\mathbf{x}_i$, we generated y_i from a 2-component exponential power mixture regression model with the shape parameter $1 < \alpha_k < 2$, for $k = 1, 2$. Both $\mathbf{y}$ and the covariates in $\mathbf{X}$ were standardised. The interceptors $\beta_{01} = \beta_{02} = 0$, and let $(\beta_{11}, \beta_{21}, \beta_{31}) \sim U[-0.5, 0.5]$, $(\beta_{12}, \beta_{22}, \beta_{32}) \sim U[3, 3.5]$ respectively. The proportion of the 2 components was randomly selected with the constraint $\pi_1 + \pi_2 = 1$, and $0 < \sigma_k^2 < 0.5$ for $k = 1, 2$.

The Q-Q plot of observation $\mathbf{y}$ is shown in Figure 5.4. From the Figure 5.4 we can see that $\mathbf{y}$ is not Gaussian distributed.

To set the initial values of parameters, we let $\pi_k \sim U(0, 1)$ with constraint $\sum_{k=1}^{2} \pi_k = 1$, $\beta_{jk} \sim U[1, 3]$, $\alpha_k \sim U[1, 3]$, and the variances $\sigma_k^2 \sim U[1, 3]$, where $k = 1, 2$ and $j = 1, \ldots, 4$.

Let $D = 100$ datasets. By selecting the best fit, we compare BIC for the number of components $k = 1, 2, 3, 4, 5$. The results are summarised in the Table 5.2:

Table 5.2: Averaged value of minimum AIC, BIC, EBIC and RAND indexs for data generated from EPD mixture regressions with dimension is 4.

GM	k	$Ave(\min BIC)$	$Ave(\min AIC)$	$Ave(\min EBIC)$	$Ave(RAND)$
	2	1483.326	1428.536	1486.910	0.965
EPD	k	$Ave(\min BIC)$	$Ave(\min AIC)$	$Ave(\min EBIC)$	$Ave(RAND)$
	2	1468.776	1371.840	1472.360	0.925

From Table 5.2, we can see that for the EM-based clustering of Gaussian mixture regressions, by fixing the shape parameters equal to 2, it gives the

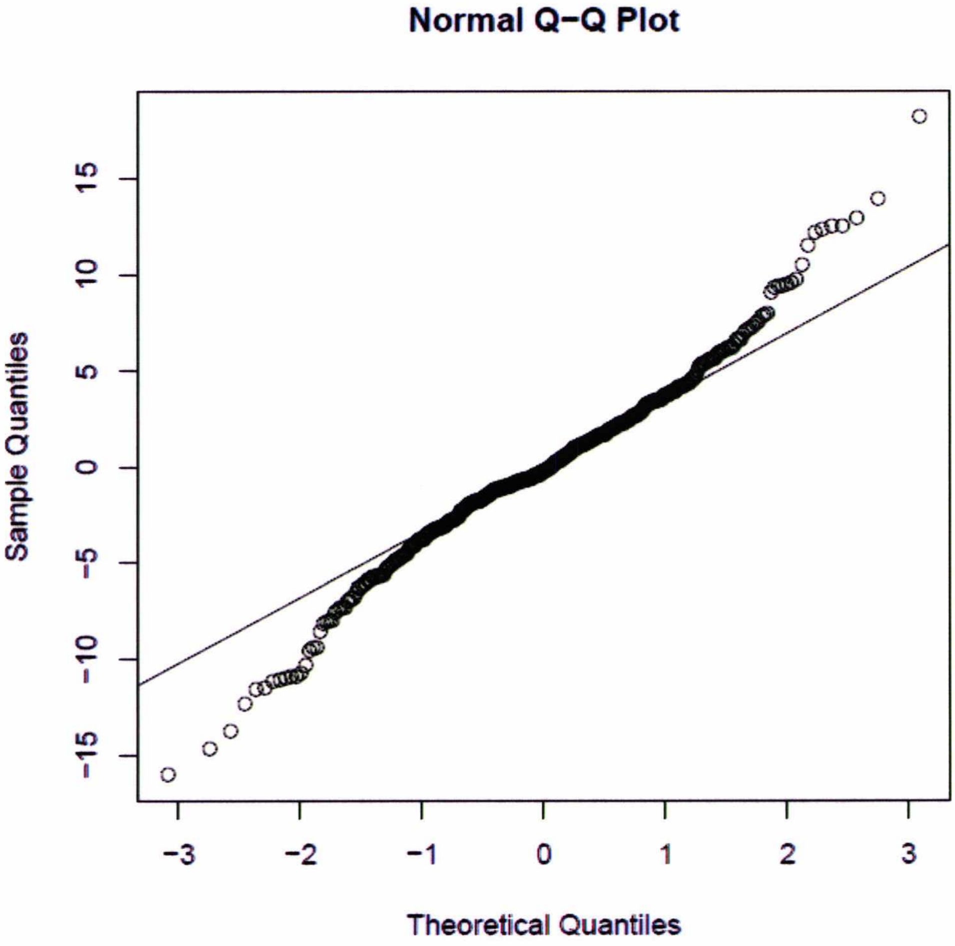

Figure 5.4: The Q-Q plot of observation $\mathbf{y}$ drawn from a exponential power mixture regression model.

averaged $\min BIC = 1483.326$ (with the corresponding $\min AIC = 1428.536$ and $\min EBIC = 1486.910$) when it has 2 components with the RAND index= 0.925. For the EM-based clustering of exponential power mixture regression, it reached its averaged $\min BIC = 1468.776$ (with the corresponding $\min AIC = 1371.840$ and $\min EBIC = 1472.360$) when it has 2 components with the RAND index= 0.965.

It shows that when the data are generated from a exponential power regression model, the EM-based clustering of Gaussian mixture regressions is less accurate than the EM-based clustering of exponential power mixture regressions. Compared to the previous simulation, we find that the prediction of clustering is more accurate when the number of components decreased from 3 to 2.

The Figure 5.5 and 5.6 illustrate the plots of the minimum BIC and the corresponding RAND index for both GM and EPD clustering derived over 100 datasets, where the red line indicates the minimum BIC of GM clustering while the blue line is for the EPD clustering: The Figures 5.5 and 5.6 proved that the EM-based clustering of EPD mixture regression have smaller BIC and higher corresponding RAND index for most dataset, when the data were drawn form a EPD mixture regression.

We also undertook the same simulation for higher dimensions data: P=6 (7 dimensions) and P=10 (11 dimensions). The results are summarized in the Table 5.3: The Table 5.3 shows the EM-based clustering of EPD mixture regression give better predictions than the EM-based clustering of GM regression under all scenarios.

Table 5.3: Averaged minimum AIC, BIC, EBIC and RAND indexes for data generated from EPD mixture regressions with dimensions are 7 and 11.

GM	k	$Ave(\min BIC)$	$Ave(\min AIC)$	$Ave(\min EBIC)$	$Ave(RAND)$
$P = 6$	2	1557.036	1485.388	1562.452	0.928
$P = 10$	2	1609.163	1503.798	1616.776	0.957
EPD	k	$Ave(\min BIC)$	$Ave(\min AIC)$	$Ave(\min EBIC)$	$Ave(RAND)$
$P = 6$	2	1542.982	1462.904	1548.398	0.965
$P = 10$	2	1590.001	1476.207	1597.614	0.965

The Figure 5.7 illustrate the plots of the minimum BIC and the corresponding RAND index for both GM and EPD clustering over 100 datasets, where the red line indicates the minimum BIC of GM clustering while the blue line is for

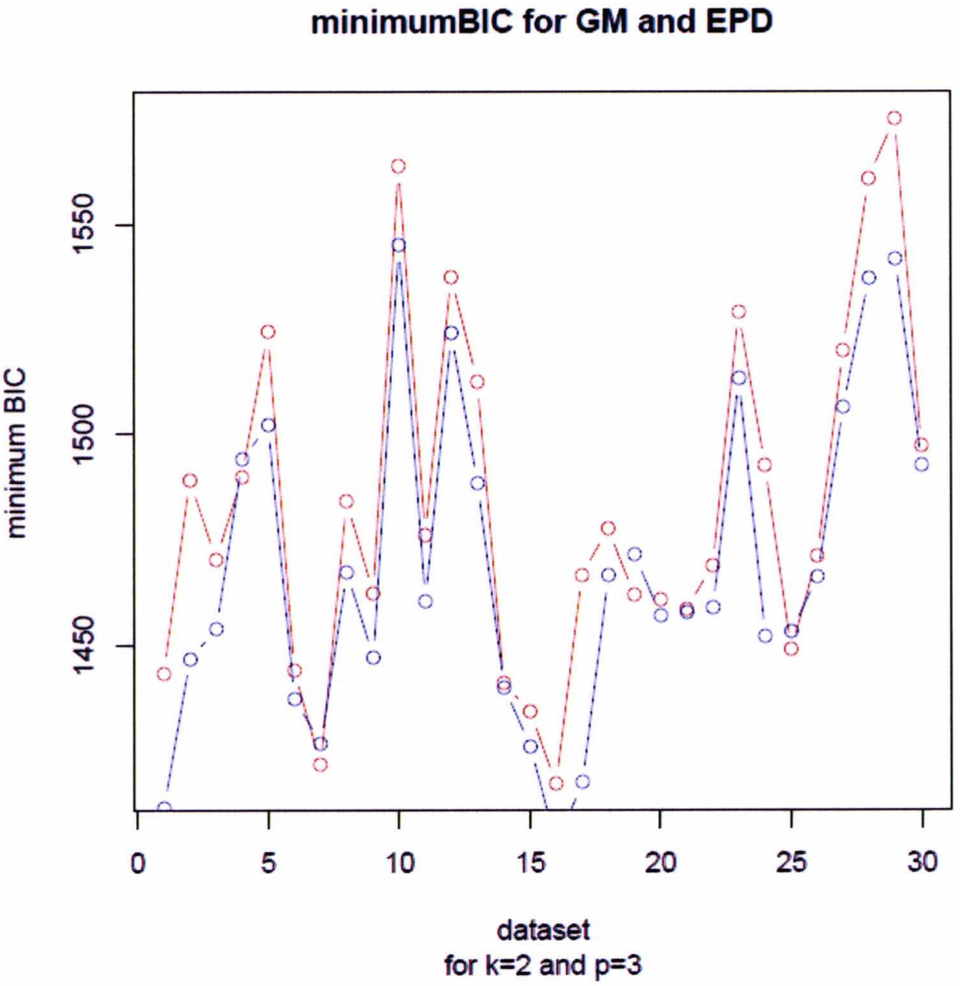

Figure 5.5: The minimum value of BIC for both GM(red) and EPD(blue) derived over 100 datasets, for data were generated from EPD mixture regressions with k=2, P=3 and $\lambda = 1/500$.

the EPD clustering:

The Figures 5.7 proved that the EM-based clustering of EPD mixture regression have smaller BIC and higher corresponding RAND index for most dataset, when the data were drawn form a EPD mixture regression.

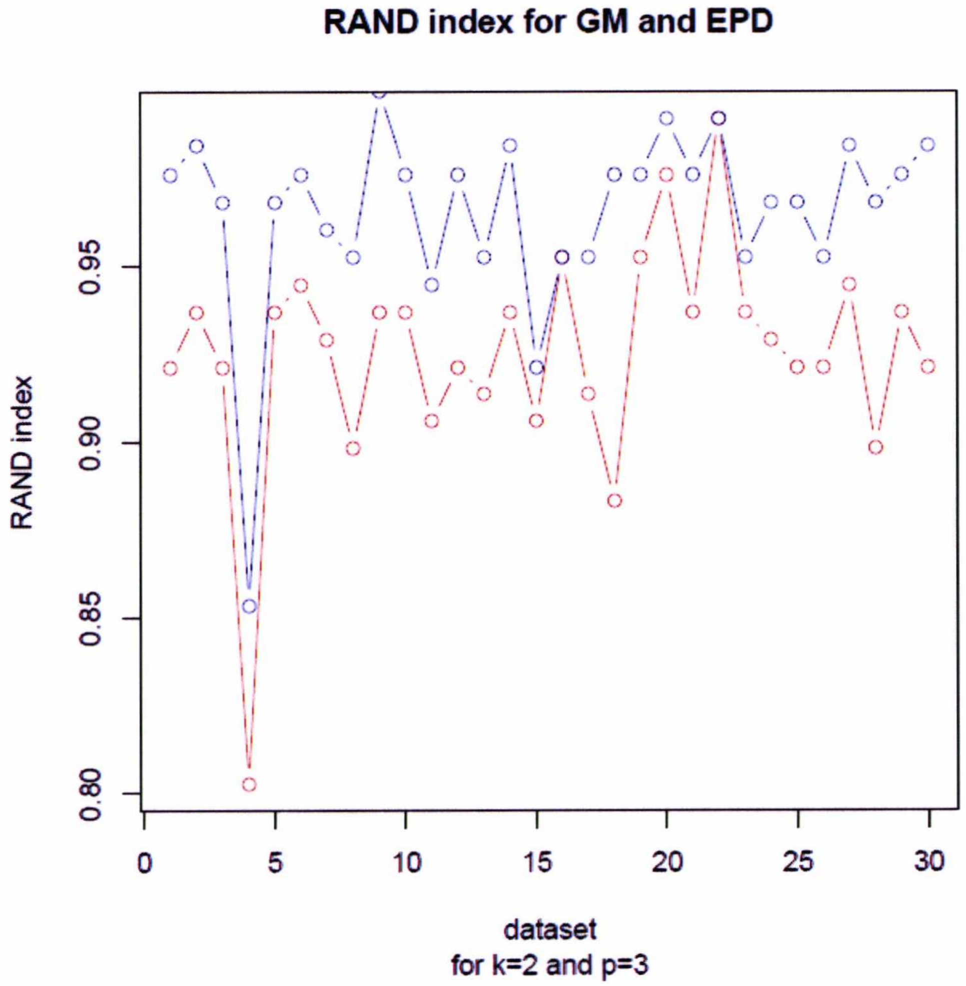

Figure 5.6: The RAND indexes for both GM(red) and EPD(blue) derived from 100 datasets, for data were generated from EPD mixture regressions with k=2, P=3 and $\lambda = 1/500$.

5.3.3 Simulation 3 (Clumpy correlations)

In the above simulations, the data were generated either from the Gaussian distribution or the exponential power distribution. In the following, we assess the accuracy of the exponential power mixture based clustering on the data which were generated from neither of the above two distributions.

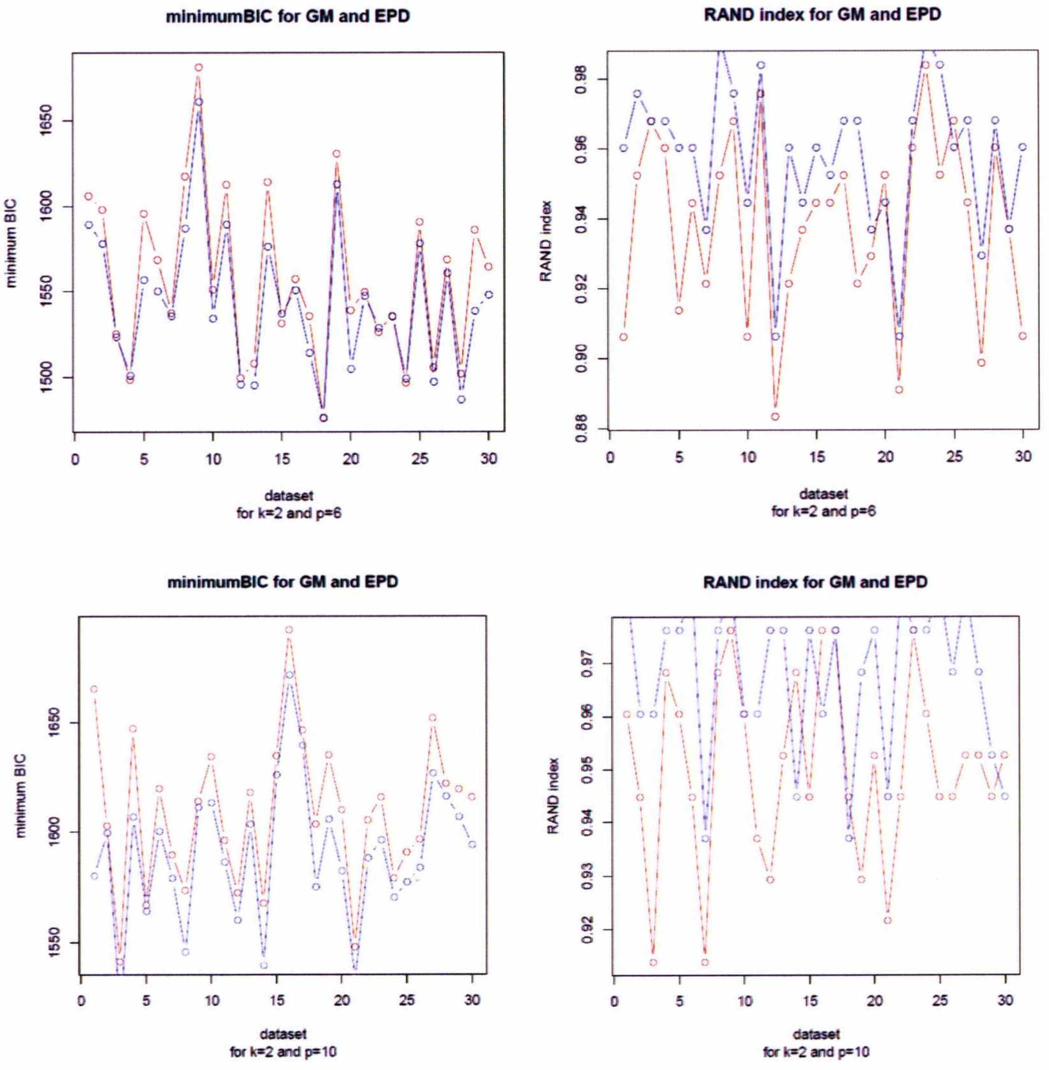

Figure 5.7: The minimum value of BIC and RAND index for both GM(red) and EPD(blue) derived from 100 datasets, where the data were generated from EPD mixture regressions with k=2 and P=6, P=10 respectively.

We let $\mathbf{X} = (\mathbf{x_1}, \ldots, \mathbf{x_{1000}})^{\mathbf{T}}$ be the covariate matrix, and $\mathbf{x_i} = (1, x_{i1}, \ldots, x_{i5})^T = (1, \mathbf{x}_i^*)^T$, where $\mathbf{x}_i^* = (x_{i1}, \ldots, x_{i5})^T$ denote the i^{th} observations on 5 covariates. We generated $\mathbf{x}_i^*$ from $N(0, \Sigma_x)$, where $\Sigma_x = (\rho^{|l-m|})_{1 \leq l,m \leq 5}$ and $0 \leq \rho \leq 1$ is a constant. For $i = 1, \ldots, n$, conditional on $\mathbf{x}_i$, the response vector $\mathbf{y}$ has two components, where $y_i = \mathbf{x}_i^T \boldsymbol{\beta}_1 + \boldsymbol{\epsilon}_i$ for $i = 1, \ldots, 500$, and $y_i = \mathbf{x}_i^T \boldsymbol{\beta}_2 + \boldsymbol{\epsilon}_i$ for $i = 501, \ldots, 1000$. Both $\mathbf{y}$ and the covariates in $\mathbf{X}$ are standardised.

Let $\boldsymbol{\beta}_1 = (0, 3, 3, 3, 3, 3)^T$ and $\boldsymbol{\beta}_2 = (0, -1, -1, -1, -1, -1)^T$. For the error term $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_{1000})^T$, ϵ_i, $i = 1, \ldots, 1000$, were generated from six independent groups: $(\epsilon_1, \ldots, \epsilon_{150}) \sim N(2, \Sigma_\epsilon)$, $(\epsilon_{151}, \ldots, \epsilon_{200}) \sim N(0, \Sigma_\epsilon)$, $(\epsilon_{201}, \ldots, \epsilon_{300}) \sim N(0, \Sigma_\epsilon)$, $(\epsilon_{301}, \ldots, \epsilon_{400}) \sim N(0, \Sigma_\epsilon)$, $(\epsilon_{401}, \ldots, \epsilon_{500}) \sim N(0, \Sigma_\epsilon)$, $(\epsilon_{501}, \ldots, \epsilon_{1000}) \sim N(-2, \Sigma_\epsilon)$, where Σ_ϵ is determined by $var(\epsilon_i) = 1$, $cov(\epsilon_l, \epsilon_m) = |l-m|^{-ak}/(2-2^{-ak})$ and $0 \leq ak \leq 1$ is a constant.

Let $D = 100$ datasets. We assess the accuracy of the EM-based clustering for both GM regressions and EPD mixture regressions by considering following combinations: $\rho = (1/4, 1/2, 3/4)$ and $ak = (1/2, 1/4, 1/8, 1/16, 1/32, 1/64)$.

Figure 5.8, 5.9 and 5.10 illustrate the box plots of RAND indexes when $\rho = 1/4, 1/2, 3/4$ respectively. In each Figure, from the left to the right the following cases are considered: $ak = 1/2$, $ak = 1/4$, $ak = 1/8$, $ak = 1/16$, $ak = 1/32$ and $ak = 1/64$.

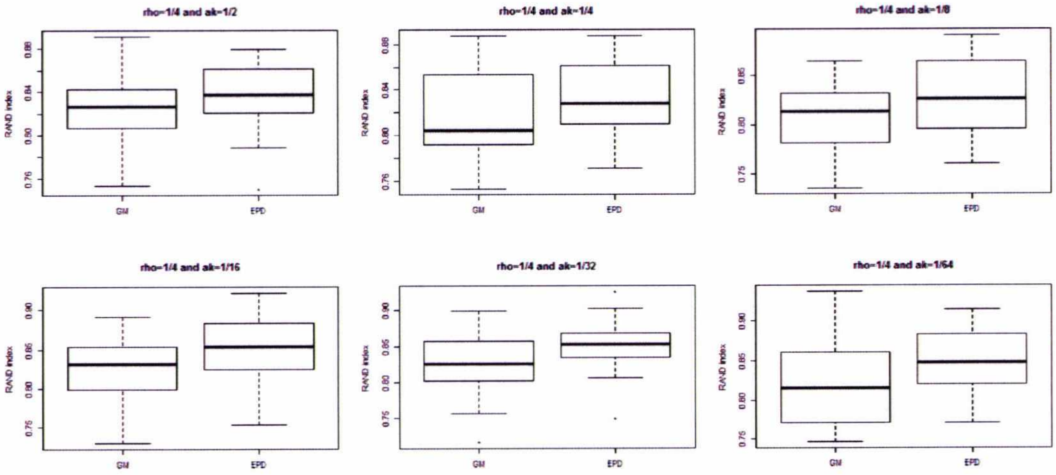

Figure 5.8: Rand indexes for GM and EPD based clustering with clumpy correlation data when $\rho = 1/4$.

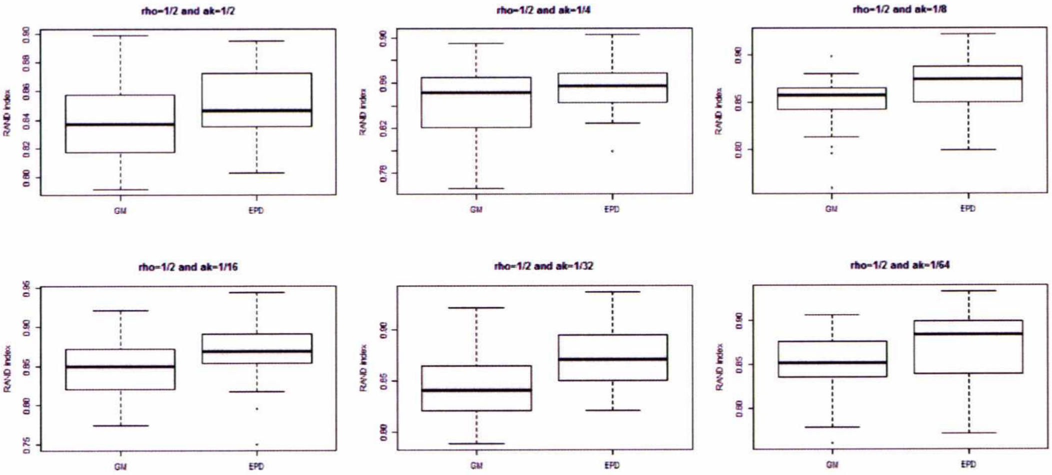

Figure 5.9: Rand indexes for GM and EPD based clustering with clumpy correlation data when $\rho = 1/2$.

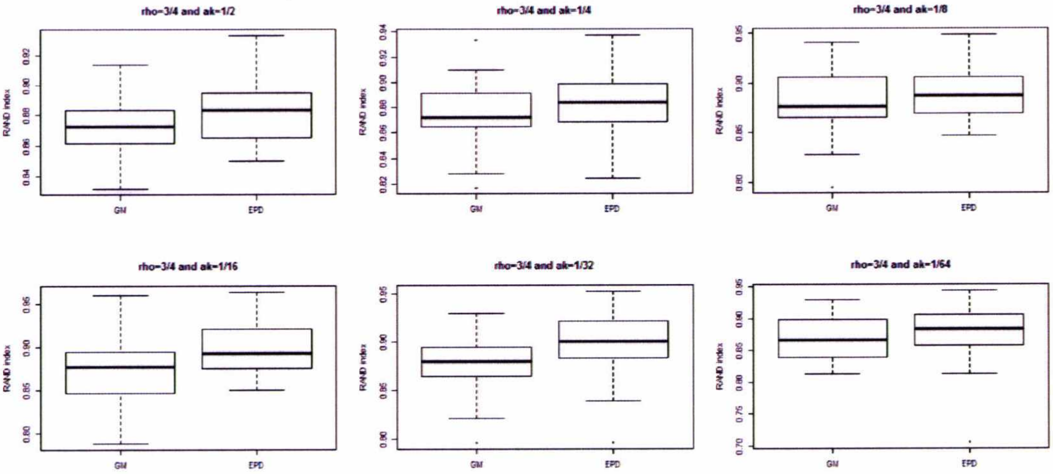

Figure 5.10: Rand indexes for GM and EPD based clustering with clumpy correlation data when $\rho = 3/4$.

The Figures 5.8, 5.9 and 5.10 prove that the EM-based clustering of EPD mixture regression have larger medians of RAND indexes in all scenarios. Which means, when the data is neither drawn form a EPD mixture regression nor GM regression model, the EM-based clustering of EPD mixture regression gives a better prediction of clustering.

5.4 Real Data Analysis

In this section, we assess the performance of the proposed procedure on real genetic datasets called *yeast stress dataset*. Gasch et al. (2000) used DNA microarrays to explore genome-wide expression patterns in the yeast Saccharomyces cerevisiae in response to diverse environmental changes. In this study the dataset contains the expression levels of 496 selected yeast genes under 173 experimental conditions (Zhang and Liang, 2010), where its columns stand for genes and its rows represent the experimental conditions. Here the size of the data $\mathbf{y}$ is 173 and we use the first row as observations on response variable and the second to the fourth rows as observations on 3 covariates.

We run both the EPD mixture regression based and GM regression based clustering procedures 100 times with initial values of $\alpha, \boldsymbol{\beta}, \sigma$ and π being randomly chosen with the constraints $\sum_{k=1}^{K} \pi_k = 1$, $0 < \sigma_k^2$ and $0 < \alpha_k$ for $k = 1, \ldots, K$. We compared the performances of the above two procedures. The averaged minimum values of AIC, BIC and EBIC are summarized in Table 5.4:

Here k is the number of components we set initially. We select the averages minimum BIC derived from 100 datasets as the best fit BIC for each number of components.

All the selection criterion shows the same results that the best fit by using the EPD mixture regression model is obtained when it has 2 components with shape index $\alpha = (0.47, 1.69)$, and the best fit by using GM regression model is obtained when it has 3 components. The minimum BIC of GM model when $k = 3$ is larger than the minimum BIC of EPD model when $k = 2$. For the best GM fit, we use the Q-Q plot to assess the the residuals of each GM com-

Table 5.4: Averaged minimum values of AIC, BIC and EBIC for the yeast dataset

	k=1	k=2	k=3	k=4
min BIC $_{EP}$	453.357	435.162	451.744	489.425
min BIC $_{GM}$	507.122	448.111	436.348	481.459
min AIC $_{EP}$	434.437	388.678	394.169	404.286
min AIC $_{GM}$	491.356	413.425	382.742	408.933
min EBIC $_{EP}$	455.554	437.359	453.941	491.622
min EBIC $_{GM}$	509.319	450.308	438.545	483.656

ponent. The Figure 5.11 prove that the residuals of each component that after GM clustering are not Gaussian distributed.

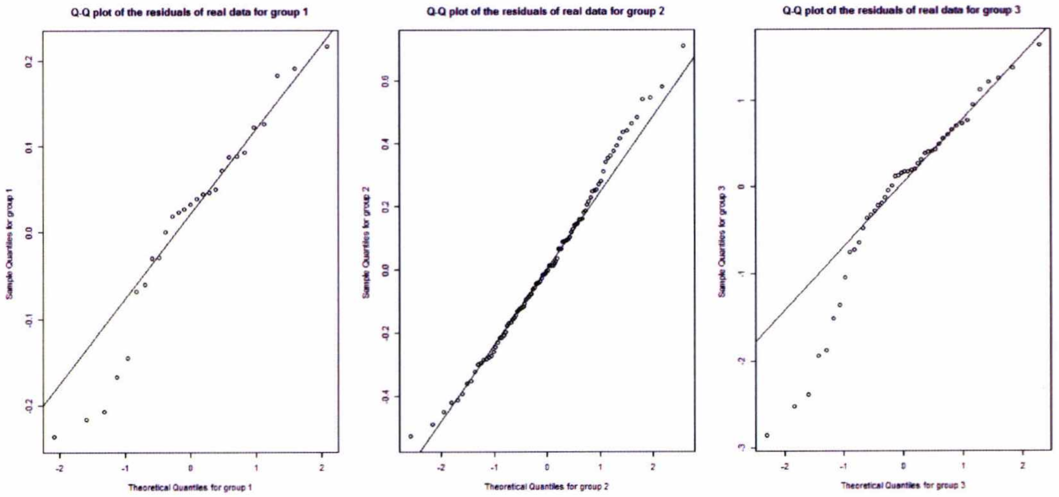

Figure 5.11: The Q-Q plot of the residuals of each components of real data after GM clustering.

We also plot the residual of each component after EPD clustering, which shows in Figure 5.12. This Figure also prove that the residuals of each component after EPD clustering are not Gaussian distributed.

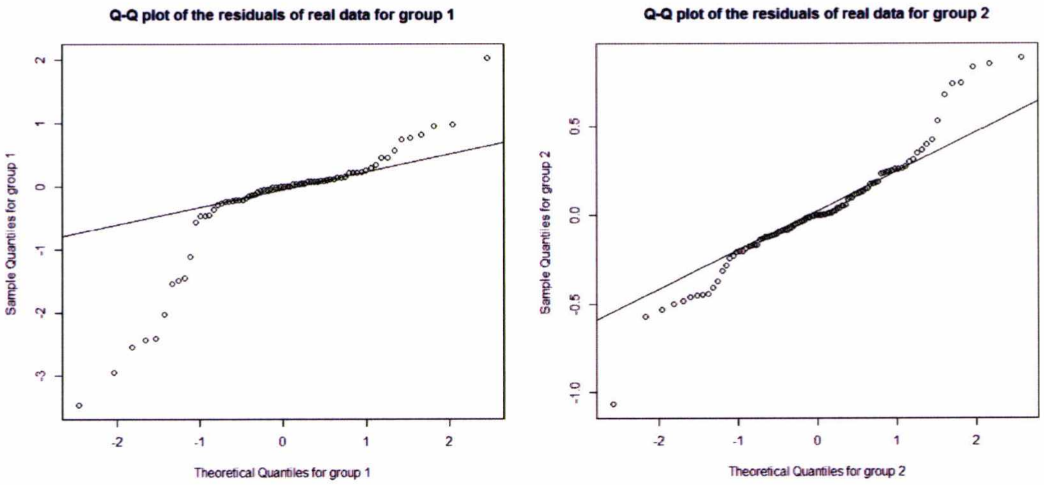

Figure 5.12: The QQ-plot of the residuals of each components real data after EPD clustering.

Chapter 6

Simultaneous Variable Selection and Clustering

In the previous chapter, we build an exponential power mixture model to reflect grouping structures in the data, where the data size is relatively small or all the covariates are related to the response vector. In modern statistics, data with unprecedented size and complexity are appearing more and more commonly in scientific fields such as genetics, engineering and finance etc. In the application of finite mixture of regression(FMR) models there are many covariates used, but the contributions to the response variables vary from each other, therefore, how to selecting the most important variables becomes a problem. There are various methods for dealing with variable selection problems for regression with many variables (Khalili, 2011). The old selection methods such as the all-subsets selection methods, the forward and backward selection methods need extensive computations. Modern studies on variable selection such as boosting which was proposed by Freund and Schapire (1997), and Lasso (Tibshirani, 1996) solved this problem. Recently, Khalili et al. (2011) proposed feature selection in finite mixture of sparse Normal linear model in high-dimensional feature space with the assumption that all data are Gaussian distributed. Here, the Gaussian assumption may be invalid. To tackle the problem, we propose exponential power mixture regression models for simultaneous variable selection and clustering in a high dimension.

In this Chapter, we introduce the LASSO type method in variable selection, and present the details on how to calculate the maximum likelihood estimators of a model by using the EM algorithm. The novelty lies in that we convert a general penalised regression estimation problem to a special L_1 penalised regression (i.e., LASSO) problem (Tibshirani, 1996). We also introduce forward selection method to reduce the size of data. We conducted three simulations to assess the accuracy of clustering by using LASSO type method and the forward selection method for variable selection, via maximum likelihood estimation of exponential power regression, under several scenarios. All results are compared with maximum likelihood estimation of Gaussian mixture regressions. At the end of this chapter we apply our method on the gene expression and motifs dataset.

6.1 Variable selection for finite exponential power mixture regression models

Suppose we have an independent sample $(y_i, \mathbf{x}_i)$, $1 \leq i \leq n$, where conditional on $\mathbf{x}_i$, y_i drawn from the following finite exponential power mixture regression model with K components,

$$f(y_i|\mathbf{x}_i, \mathbf{\Psi}) = \sum_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k).$$

where the parameter vector $\Psi = (\pi_1, \ldots, \pi_{K-1}, \boldsymbol{\zeta}^T)^T$ is the vector containing all the unknown parameters in this mixture model, and $\boldsymbol{\zeta} = (\boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_K^T)^T$. $\boldsymbol{\theta}_k = (\boldsymbol{\beta}_k, \alpha_k, \sigma_k)^T$, $k = 1, \ldots, K$.

In the sparse exponential power mixture regression model, the density of the

k^{th} component can be written as

$$f_k\left(y_i\left|\mathbf{x}_i,\boldsymbol{\theta}_k\right.\right)=\frac{\alpha_k}{2\sigma_k\Gamma\left(1/\alpha_k\right)}\exp\left(-\frac{\left|y_i-\mathbf{x}_i^T\boldsymbol{\beta}_k\right|^{\alpha_k}}{(\sigma_k^2)^{\alpha_k/2}}\right).$$

Recall that the latent variable is

$$z_{ik}=(z_i)_k=\begin{cases}1 & y_i\in\text{the } k\text{-th component}\\ 0 & y_i\notin\text{the } k\text{-th component}\end{cases}.$$

In the EM framework, the observed-data $\mathbf{D}=(y_1,\mathbf{x}_1;\ldots;y_n,\mathbf{x}_n)^T$ are viewed as an incomplete data. We combine it with the indicator z_{ik} to form the complete-data

$$\mathbf{D}_c=(\mathbf{D},\mathbf{z}).$$

The complete-data log-likelihood for $\boldsymbol{\Psi}$ can be written in the following form

$$\log L_c(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X})=\sum_{i=1}^{n}\sum_{k=1}^{K}z_{ik}\{\log\pi_k+\log f_k(y_i|\mathbf{x}_i,\boldsymbol{\theta}_k)\}. \tag{6.1}$$

where the regression coefficients is $\boldsymbol{\beta}_k=(\beta_{k0},\beta_{k1},\ldots,\beta_{kP})$, $k=1,\ldots,K$. The $\beta_{kj}=0$, $j=1,\ldots,P$, if the effect of a component of $\mathbf{x}$ is not significant. When there are too many covariates in the model, the data will be over fitted.

To avoid the over fitting problem, we add a penalty to the complete-data log-likelihood.

We consider the penalized complete-data log-likelihood

$$\begin{aligned}\tilde{l}_c(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X}) &= \log L_c(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X})-P_n(\boldsymbol{\Psi})\\ &= \sum_{i=1}^{n}\sum_{k=1}^{K}z_{ik}\{\log\pi_k+\log f_k(y_i|\mathbf{x}_i,\boldsymbol{\theta}_k)\}-n\sum_{k=1}^{K}\pi_k\sum_{j=0}^{P}P_\lambda(\frac{\mid\beta_{k,j}\mid}{\sigma_k}),\end{aligned}$$

where the penalty

$$P_\lambda(\frac{\mid \beta_{kj} \mid}{\sigma_k}) \cong P_\lambda(\frac{\mid \beta_{kj}^{(s)} \mid}{\sigma_k^{(s)}}) + P_\lambda^{'}(\frac{\mid \beta_{kj}^{(s)} \mid}{\sigma_k^{(s)}})(\frac{\mid \beta_{kj}^{(s)} \mid}{\sigma_k^{(s)}} - \frac{\mid \beta_{kj}^{(s)} \mid}{\sigma_k^{(s)}}),$$

with $\beta_{kj}^{(s)}$ and $\sigma_k^{(s)}$ were known from the s^{th} step of iteration.

The reason that the regression coefficients β_{kj} scaled by σ_k is it need to be more accurate when σ_k is small.

In the LASSO, the penalty is defined as $P_\lambda^{'}(.)$:

$$P_\lambda(\frac{\mid \beta_{k,j}^{(s)} \mid}{\sigma_k^{(s)}}) = \lambda\frac{\mid \beta_{k,j} \mid}{\sigma_k}, \; P_\lambda^{'}(\frac{\mid \beta_{k,j}^{(s)} \mid}{\sigma_k^{(s)}}) = \lambda.$$

Maximizing the penalized log-likelihood is the same as maximizing the completed log-likelihood with certain constrains. The LASSO method selects a sub-model while the estimated $\beta_{kj} = 0$ if $j \notin \mathbf{1}_k$ or $\beta_{kj} \neq 0$ if $j \in \mathbf{1}_k$. Thus it combines the variable selection and the parameter estimation into one step, which reducing computational intensity.

In the following section, we use a Lasso type method for variable selection in the finite mixture of regression models.

6.2 Lasso type method

In this section, we present the details on how to calculate the maximum likelihood estimators $\boldsymbol{\Psi}$ by using the EM algorithm. The novelty of this research is that solving the regression estimation problem for β_t is converted to solving a problem of the Lasso for β_t.

6.2.1 EM algorithm

Similar to the previous two chapters, there are two steps in maximum likelihood estimation(MLE) of the exponential power mixture model of regression by using the EM algorithm: In the E-step, the missing data are estimated by using the observed data and current estimate of the model parameters. In the M-step, the expected complete log-likelihood function of the exponential power mixture of regression model is maximized under the assumption that the missing data are known.

Suppose we have an independent sample $(y_i, \mathbf{x}_i)$, $1 \leq i \leq n$, where conditional on $\mathbf{x}_i$, y_i drawn from the following finite exponential power mixture regression model with K components,

$$f(y_i|\mathbf{x}_i, \boldsymbol{\Psi}) = \sum_{k=1}^{K} \pi_k f_k(y_i|\mathbf{x}_i, \boldsymbol{\theta}_k).$$

In this model, the regression coefficients $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k)^T$, for $1 \leq k \leq K$, where $\boldsymbol{\beta}_k = (\beta_{k0}, \beta_{k1}, \ldots, \beta_{kP})^T$. The dispersion and shape parameters are $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_K)^T$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)^T$ respectively. The predictive vector $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})^T$ where $\mathbf{x_i} = (1, x_1, \ldots, x_P)^T$ for $i = (1, \ldots, n)$, while the response variable $\mathbf{y} = (y_1, \ldots, y_n)$.

E step

In this step, we calculate the expectation of the penalised complete-data log-likelihood based on the current value of $\boldsymbol{\Psi}^{(s)}$, where $\boldsymbol{\Psi}^{(s)}$ stand for the value of all the parameters after the s^{th} iteration.

$$Q\left(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(s)}\right) = E_{\boldsymbol{\Psi}^{(s)}}\left(\tilde{l}_c(\boldsymbol{\Psi}|\mathbf{y}, \mathbf{X})\,|\mathbf{y}, \mathbf{X}\right)$$

$$
\begin{aligned}
&= E_{\boldsymbol{\Psi}^{(s)}} \Bigg\{ \Bigg[\sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \left(\log\left(\pi_k\right) + \log f_k \left(y_i \left| \mathbf{x_i}^T \boldsymbol{\beta}_k, \sigma_k^2 \alpha_k \right.\right)\right) \\
&\quad -n \sum_{k=1}^{K} \pi_k \sum_{j=0}^{P} P_\lambda \left(\frac{|\beta_{kj}|}{\sigma_k} \right) \Bigg] |\mathbf{y}, \mathbf{X} \Bigg\} \\
&= E_{\boldsymbol{\Psi}^{(s)}} \Bigg\{ \sum_{i=1}^{n} \sum_{k=1}^{K} \left(z_{ik} \left| \mathbf{y}, \mathbf{X} \right.\right) \left[\log\left(\pi_k\right) + \log f_k \left(y_i \left| \mathbf{x}_i^T \boldsymbol{\beta}_k, \sigma_k^2, \alpha_k \right.\right) \right] \\
&\quad -n \sum_{k=1}^{K} \pi_k \sum_{j=0}^{P} P_\lambda \left(\frac{|\beta_{kj}|}{\sigma_k} \right) \Bigg\} \\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} E_{\boldsymbol{\Psi}^{(s)}} \left(z_{ik} \left| \mathbf{y}, \mathbf{X} \right.\right) \left[\log\left(\pi_k\right) + \log f_k \left(y_i \left| \mathbf{x}_i^T \boldsymbol{\beta}_k, \sigma_k^2, \alpha_k \right.\right) \right] \\
&\quad -n \sum_{k=1}^{K} \pi_k \sum_{j=0}^{P} P_\lambda \left(\frac{|\beta_{kj}|}{\sigma_k} \right) \\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \left[\log\left(\pi_k\right) + \log f_k \left(y_i \left| \mathbf{x}_i^T \boldsymbol{\beta}_k, \sigma_k^2, \alpha_k \right.\right) \right] \\
&\quad -n \sum_{k=1}^{K} \pi_k \sum_{j=0}^{P} P_\lambda \left(\frac{|\beta_{kj}|}{\sigma_k} \right) \\
&\cong \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \log\left(\pi_k\right) \\
&\quad + \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \log \frac{\alpha_k}{2\sigma_k \Gamma\left(1/\alpha_k\right)} \\
&\quad - \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \frac{\left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_k\right|^{\alpha_k}}{(\sigma_k^2)^{\alpha_k/2}} \\
&\quad -n \sum_{k=1}^{K} \pi_k \sum_{j=0}^{P} P'_{\lambda k} \left(\frac{\left|\beta_{kj}^{(s)}\right|}{\sigma_k^{(s)}} \right) \frac{\left|\beta_{kj}^{(s)}\right|}{\sigma_k^{(s)}} \\
&\quad -n \sum_{k=1}^{K} \pi_k \sum_{j=0}^{P} \left\{ P_{\lambda k} \left(\frac{\left|\beta_{kj}^{(s)}\right|}{\sigma_k^{(s)}} \right) - P'_{\lambda k} \left(\frac{\left|\beta_{kj}^{(s)}\right|}{\sigma_k^{(s)}} \right) \frac{\left|\beta_{kj}^{(s)}\right|}{\sigma_k^{(s)}} \right\}.
\end{aligned}
$$

M step

In this step we estimate the parameters Ψ_t on the $(s+1)^{th}$ iteration, based on the parameters obtained from the s^{th} iteration. For this purpose, we maximise the expectation of the penalised complete-data log-likelihood from the last step.

Estimate $\boldsymbol{\beta}_t$

To update the value of $\boldsymbol{\beta}_t$, we derivative the function $Q(.)$ w.r.t $\boldsymbol{\beta}_t$:

$$\begin{aligned}
\frac{\partial Q\left(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(s)}\right)}{\partial \boldsymbol{\beta}_t} &= -\sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t^2)^{\alpha_t/2}} \frac{\partial \left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|^{\alpha_t}}{\partial \boldsymbol{\beta}_t} - n\pi_t \frac{\partial \sum_{j=0}^{P} P'_{\lambda t}\left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right) \frac{|\beta_{tj}|}{\sigma_t}}{\partial \boldsymbol{\beta}_t} \\
&= -\sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t)^{\alpha_t}} \frac{\partial (y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t)^{2*\alpha_t}}{\partial \boldsymbol{\beta}_t} - \frac{n\pi_t}{\sigma_t} \sum_{j=0}^{P} P'_{\lambda t}\left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right) \frac{\partial |\beta_{tj}|}{\partial \boldsymbol{\beta}_t} \\
&= \sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t)^{\alpha_t}} \alpha_t \left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|^{(\alpha_t - 2)} \left(\mathbf{x}_i y_i - \mathbf{x}_i \mathbf{x}_i^T \boldsymbol{\beta}_t\right) \\
&\quad - \frac{n\lambda\pi_t}{\sigma_t} \begin{pmatrix} P'_{\lambda t}\left(\frac{\left|\beta_{t1}^{(s)}\right|}{\sigma_t^{(s)}}\right) \frac{sgn(\beta_{t1})}{\lambda} \\ \vdots \\ P'_{\lambda t}\left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right) \frac{sgn(\beta_{tj})}{\lambda} \end{pmatrix} \\
&= \sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t)^{\alpha_t}} \alpha_t \left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|^{(\alpha_t - 2)} \mathbf{x}_i y_i \\
&\quad - \sum_{i=1}^{n} \frac{\tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)})}{(\sigma_t)^{\alpha_t}} \alpha_t \left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|^{(\alpha_t - 2)} \mathbf{x}_i \mathbf{x}_i^T \boldsymbol{\beta}_t \\
&\quad - \frac{n\lambda\pi_t}{\sigma_t} \begin{pmatrix} P'_{\lambda t}\left(\frac{\left|\beta_{t1}^{(s)}\right|}{\sigma_t^{(s)}}\right) \frac{sgn(\beta_{t1})}{\lambda} \\ \vdots \\ P'_{\lambda t}\left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right) \frac{sgn(\beta_{tj})}{\lambda} \end{pmatrix}.
\end{aligned}$$

We can write the above equation in a simpler form:

$$\frac{\partial Q}{\partial \boldsymbol{\beta}_t} = \mathbf{X}^T\mathbf{W}_t\mathbf{y} - (\mathbf{X}^T\mathbf{W}_t\mathbf{X})\boldsymbol{\beta}_t - n\lambda\mathbf{V}_t\text{sgn}(\boldsymbol{\beta}_t) = 0 \tag{6.2}$$

where

$$W_t = \tfrac{\alpha_t}{(\sigma_t^2)^{\alpha_t/2}} \begin{pmatrix} \phi(\tau_{t1}) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \phi(\tau_{tn}) \end{pmatrix},$$

with $\phi(\tau_{ti}) = \tau_{ti}(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\left(\alpha_t - 2\right)}$, $i = 1, \ldots, n$,

$$V_t = \tfrac{\pi_t}{\sigma_t} \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & \frac{P'_\lambda\left(\frac{\left|\beta_{t1}^{(s)}\right|}{\sigma_t^{(s)}}\right)}{\lambda} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{P'_\lambda\left(\frac{\left|\beta_{tp}^{(s)}\right|}{\sigma_t^{(s)}}\right)}{\lambda} \end{pmatrix}, \text{ and}$$

$$sgn(\boldsymbol{\beta}_t) = \begin{pmatrix} sgn(\beta_{t1}) \\ \vdots \\ sgn(\beta_{tp}) \end{pmatrix}.$$

We define $\mathbf{X}^* = \mathbf{W}^{1/2}\mathbf{X}$, and $\mathbf{y}^* = \mathbf{W}^{1/2}\mathbf{y}$, the equation(6.2) can be write into the following form:

$$\begin{aligned} \frac{\partial Q}{\partial \boldsymbol{\beta}_t} &= X^TW_tY - (X^TW_tX)\boldsymbol{\beta}_t - n\lambda V_t sgn(\boldsymbol{\beta}_t) \\ &= \left(W_t^{(V)1/2}XV_t^{(V)(-1)}\right)^T \left(W_t^{(V)1/2}y\right) \\ &\quad - \left(W_t^{(V)1/2}XV_t^{(V)(-1)}\right)^T \left(W_t^{(V)1/2}XV_t^{(V)(-1)}\right)(V_t^{(V)} + e_1)\boldsymbol{\beta}_t \\ &\quad -n\lambda sgn((V_t^{(V)} + e_1)\boldsymbol{\beta}_t) \\ &= X^{*T}Y^* - X^{*T}X^*\boldsymbol{\beta}_t^* - n\lambda\text{sgn}(\boldsymbol{\beta}_t^*) = 0, \end{aligned} \tag{6.3}$$

for $1 \leq t \leq K$, where $e_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix}_{(p+1)\times(p+1)}$,

$$V_t^{(s)} + e_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \left(\frac{\pi_t^{(s)}}{\sigma_t^{(V)}}\right)\frac{P'_\lambda\left(\frac{|\beta_{t1}^{(s)}|}{\sigma_t^{(s)}}\right)}{\lambda} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \left(\frac{\pi_t^{(s)}}{\sigma_t^{(V)}}\right)\frac{P'_\lambda\left(\frac{|\beta_{tp}^{(s)}|}{\sigma_t^{(s)}}\right)}{\lambda} \end{pmatrix},$$

and

$$\boldsymbol{\beta}_t^* = (V_t^{(V)} + e_1)\boldsymbol{\beta}_t.$$

.

To solve equation(6.3) is the same as to optimize $\hat{\boldsymbol{\beta}}_t^*$ for each t, where

$$\hat{\boldsymbol{\beta}}_t^* = \arg\min_{\boldsymbol{\beta}_t^*}\{1/2\sum_{i=1}^{n}(y_i^* - \mathbf{x}_{\mathbf{i}}^{*T})^2 + n\lambda\sum_{j=0}^{P}|\beta_{tj}^*|\},$$

for $1 \leq t \leq K$.

We have successfully converted the problem of solving the equation(6.2) into a Lasso problem. Now we can apply the LARS algorithm to solve the above optimization problem for each t, thus

$$\hat{\boldsymbol{\beta}}_t = (V_t^{-(s)} + e_1)\hat{\boldsymbol{\beta}}_t^*, \tag{6.4}$$

for $1 \leq t \leq K$.

Estimate σ_t

We update σ_t^2, for $1 \leq t \leq K$, by derivative $Q(.)$ w.r.t $1/\sigma_t^2$ and let it equals to 0, we obtain:

$$\begin{aligned}
\frac{\partial Q\left(\boldsymbol{\Psi}|\boldsymbol{\Psi}^{(s)}\right)}{\partial(\sigma_t^{(-2)})} &= \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \frac{\partial(\log(1/\sigma_t))}{\partial(1/\sigma_t^2)} \\
&\quad - \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|^{\alpha_t} \frac{\alpha_t}{2} \left(1/\sigma_t^2\right)^{(\alpha_t/2)-1} \\
&\quad - n\pi_t \sum_{j=0}^{P} P'_\lambda \left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right) |\beta_{tj}| \frac{\partial(1/\sigma_t)}{\partial(1/\sigma_t^2)} \\
&= \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \frac{\partial[1/2 \log(1/\sigma_t^2)]}{\partial(1/\sigma_t^2)} \\
&\quad - \frac{\alpha_t}{2} \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|^{\alpha_t} \left(1/\sigma_t^2\right)^{(\alpha_t/2)-1} \\
&\quad - n\pi_t \sum_{j=0}^{P} P'_\lambda \left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right) |\beta_{tj}| \frac{\partial(1/\sigma_t^2)^{\frac{1}{2}}}{\partial(1/\sigma_t^2)} \\
&= \frac{1}{2} \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \sigma_t^2 \\
&\quad - \frac{\alpha_t}{2} \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|^{\alpha_t} \left(1/\sigma_t^2\right)^{(\alpha_t/2)-1} \\
&\quad - \frac{1}{2} n\pi_t \sum_{j=0}^{P} P'_\lambda \left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right) |\beta_{tj}| \left(1/\sigma_t^2\right)^{-\frac{1}{2}} \\
&= \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \sigma_t^2 - \alpha_t \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \boldsymbol{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|^{\alpha_t} \sigma_t^{-\alpha_t+2} \\
&\quad - n\pi_t \sum_{j=0}^{P} P'_\lambda \left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right) |\beta_{tj}| \, \sigma_t
\end{aligned}$$

$$= \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) - \frac{\alpha_t}{\sigma_t^{\alpha_t}} \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|^{\alpha_t}$$
$$-n\pi_t \sum_{j=0}^{P} P'_\lambda \left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}} \right) \frac{|\beta_{tj}|}{\sigma_t} = 0. \qquad (6.5)$$

Consider a special case of σ_t^2, if $\alpha_t = 1$, we can solve the above equation directly as follows:

$$\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)})$$
$$= \frac{1}{\sigma_t} \left(\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|^{\alpha_t} - n\pi_t \sum_{j=0}^{P} P'_\lambda \left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}} \right) |\beta_{tj}| \right),$$

which implies,

$$\frac{1}{\sigma_t} = \frac{\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)})}{\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|^{\alpha_t} - n\pi_t \sum_{j=0}^{P} P'_\lambda \left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}} \right) |\beta_{tj}|}$$

$$\hat{\sigma}_t^2 = \left(\frac{\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|^{\alpha_t} - n\pi_t \sum_{j=0}^{P} P'_\lambda \left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}} \right) |\beta_{tj}|}{\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)})} \right)^2.$$

For another special case, if $\alpha_t = 2$, the equation(6.5) becomes

$$\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) - \frac{2}{\sigma_t^2} \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|^2 - n\pi_t \sum_{j=0}^{P} P'_\lambda \left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}} \right) \frac{|\beta_{tj}|}{\sigma_t} = 0 \qquad (6.6)$$

Solving this equation by using the quadratic formula and only considering the positive root as $\frac{1}{\sigma} > 0$, we obtain:

$$\sigma_t^2 = \frac{2(\frac{4\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})\left|y_i-\mathbf{x}_i^T\boldsymbol{\beta}_t\right|^2}{\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})})}{\left(\frac{n\pi_t\sum_{j=0}^{P}P'_\lambda\left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right)|\beta_{tj}|}{\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})}\right)^2+\frac{8\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})\left|y_i-\mathbf{x}_i^T\boldsymbol{\beta}_t\right|^2}{\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})}-\frac{n\pi_t\sum_{j=0}^{P}P'_\lambda\left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right)\frac{|\beta_{tj}|}{\sigma_t}}{\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})}}.$$

If $\lambda = 0$, i.e. no penalty is applied, we obtain

$$\hat{\sigma}_t^{2(s+1)} = \frac{2\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\,\mathbf{\Psi}^{(s)})\left|y_i-\mathbf{x}_i^T\boldsymbol{\beta}_t\right|^2}{\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\,\mathbf{\Psi}^{(s)})}. \tag{6.7}$$

From the above equations, it can be seen that equation(6.5) can not be solved directly, therefore the Newton-Raphson method is applied. We calculate the first and second derivative of function $Q(.)$ with respect to $1/\sigma_t^2$ and let them equal to 0.

$$\begin{aligned}\frac{\partial Q\left(\mathbf{\Psi}|\mathbf{\Psi}^{(s)}\right)}{\partial\sigma_t^{(-2)}} &= \frac{1}{2}\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\,\mathbf{\Psi}^{(s)})\sigma_t^2 \\ &\quad -\frac{\alpha_t}{2}\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\,\mathbf{\Psi}^{(s)})\left|y_i-\mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\alpha_t}\left(1/\sigma_t^2\right)^{(\alpha_t/2)-1} \\ &\quad -\frac{1}{2}n\pi_t\sum_{j=0}^{P}P'_\lambda\left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right)|\beta_{tj}|\,(1/\sigma_t^2)^{-\frac{1}{2}},\end{aligned}$$

and

$$\frac{\partial^2 Q\left(\mathbf{\Psi}|\mathbf{\Psi}^{(s)}\right)}{\partial\left(\sigma_t^{(-2)}\right)^2} = \frac{1}{2}\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\,\mathbf{\Psi}^{(s)})\frac{\partial\left(\sigma_t^{(-2)}\right)^{-1}}{\partial\left(\sigma_t^{(-2)}\right)}$$

$$
\begin{aligned}
& -\frac{\alpha_t}{2}\left(\frac{\alpha_t}{2}-1\right)\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})\left|y_i-\mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\alpha_t}\left(1/\sigma_t^2\right)^{(\alpha_t/2)-2}\\
& -\frac{1}{2}n\pi_t\sum_{j=0}^{P}P'_\lambda\left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right)|\beta_{tj}|\frac{\partial(1/\sigma_t^2)^{-\frac{1}{2}}}{\partial(1/\sigma_t^2)}\\
= & -\frac{1}{2}\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})(1/\sigma_t^2)^{-2}\\
& -\frac{\alpha_t}{2}\left(\frac{\alpha_t}{2}-1\right)\sum_{i=1}^{n}\tau_t(y_i|\mathbf{x}_i,\mathbf{\Psi}^{(s)})\left|y_i-\mathbf{x}_i^T\boldsymbol{\beta}_t\right|^{\alpha_t}\left(1/\sigma_t^2\right)^{(\alpha_t/2)-2}\\
& +\frac{1}{4}n\pi_t\sum_{j=0}^{P}P'_\lambda\left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right)|\beta_{tj}|\,(1/\sigma_t^2)^{-\frac{3}{2}}.
\end{aligned}
$$

As $1/\sigma_t^2$ can not take negative values, we define a new variable $\eta_t = log(\sigma_t^2)$, where η can take values from $-\infty$ to ∞. By implant the Newton-Raphson method, we obtain

i.e.

$$
\boldsymbol{\eta}=\begin{pmatrix}\eta_1\\ \vdots\\ \eta_K\end{pmatrix}^{s+1}=\begin{pmatrix}\eta_1\\ \vdots\\ \eta_K\end{pmatrix}^{s}-\left(\frac{\partial^2 Q}{\partial\boldsymbol{\eta}\partial\boldsymbol{\eta}^T}\right)^{-1}\begin{pmatrix}\frac{\partial Q}{\partial\sigma_1}\sigma_1\\ \vdots\\ \frac{\partial Q}{\partial\sigma_K}\sigma_K\end{pmatrix},
$$

where,

$$
\begin{aligned}
\eta_t^{(s+1)} &= \eta_t^{(s)}-\left(\frac{\partial^2 Q}{\partial\left(1/\sigma_t^2\right)^2}\left(1/\sigma_t^2\right)+\frac{\partial Q}{\partial\left(1/\sigma_t^2\right)}\right)^{-1}\frac{\partial Q}{\partial\left(1/\sigma_t^2\right)}\\
\text{as } 1/\sigma_t^2 &= e^{\eta_t}.
\end{aligned}
$$

We can write the estimated σ_t into the following form:

$$
\left(1/\hat{\sigma_t}^2\right)^{(s+1)}=\left(1/\sigma_t^2\right)^{(s)}e^{-\left(\frac{\partial^2 Q}{\partial(1/\sigma_t^2)^2}\left(1/\sigma_t^2\right)+\frac{\partial Q}{\partial(1/\sigma_t^2)}\right)^{-1}\frac{\partial Q}{\partial(1/\sigma_t^2)}}. \tag{6.8}
$$

Estimate α_t

As α_t can not take a negative value, we define a new variable $\eta_t = log(\alpha_t)$, where η can take values from $-\infty$ to ∞. We use Newton-Raphson method to update α_t as follows.

We can write A in the following form:

$$\begin{aligned} A &= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \log(\frac{\alpha_k}{2\sigma_k\Gamma\left(1/\alpha_k\right)}) - \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \frac{\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_k\right|^{\alpha_k}}{(\sigma_k^2)^{\alpha_k/2}} \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \log(\frac{1}{2\sigma_k}) + \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \log(\frac{\alpha_k}{\Gamma\left(1/\alpha_i\right)}) \\ &\quad - \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_k(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \frac{\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_k\right|^{\alpha_k}}{(\sigma_k^2)^{\alpha_k/2}}. \end{aligned}$$

The first derivative of the function A with respect to α_t is:

$$\begin{aligned} \frac{\partial A}{\partial \alpha_t} &= \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left(\frac{1}{\alpha_t} + \frac{\Gamma'\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)}\frac{1}{\alpha_t^2}\right) \\ &\quad - \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left(\frac{\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|}{\sigma_t}\right)^{\alpha_t} \log\left(\frac{\left|y_i - \mathbf{x}_i^T\boldsymbol{\beta}_t\right|}{\sigma_t}\right), \end{aligned}$$

and the second derivative of the function A with respect to α_t is:

$$\begin{aligned} \frac{\partial^2 A}{\partial \alpha_t^2} &= \sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left(-\frac{1}{\alpha_t^2} - \frac{2}{\alpha_t^3}\frac{\Gamma'\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)} + \frac{1}{\alpha_t^2}\left(\frac{\Gamma''\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)}\left(-\frac{1}{\alpha_t^2}\right)\right.\right. \\ &\quad \left.\left. + \left(\frac{\Gamma'\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)}\right)^2\left(-\frac{1}{\alpha_t^2}\right)\right)\right) \end{aligned}$$

$$\begin{aligned}
& -\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left(\frac{\left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|}{\sigma_t}\right)^{\alpha_t} \left(\log\left(\frac{\left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|}{\sigma_t}\right)\right)^2 \\
= & -\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left(\frac{1}{\alpha_t^2} + \frac{2}{\alpha_t^3} \frac{\Gamma'\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)} + \frac{1}{\alpha_t^4}\left(\frac{\Gamma"\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)} - \left(\frac{\Gamma'\left(1/\alpha_t\right)}{\Gamma\left(1/\alpha_t\right)}\right)^2\right)\right) \\
& -\sum_{i=1}^{n} \tau_t(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left(\frac{\left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|}{\sigma_t}\right)^{\alpha_t} \left(\log\left(\frac{\left|y_i - \mathbf{x}_i^T \boldsymbol{\beta}_t\right|}{\sigma_t}\right)\right)^2
\end{aligned}$$

where

$$\frac{\partial 1/\Gamma\left(1/\alpha_t\right)}{\partial \alpha_t} = \frac{\partial\left(\Gamma\left(1/\alpha_t\right)\right)^{-1}}{\partial \alpha_t} = \left(\Gamma\left(1/\alpha_t\right)\right)^{-2} \Gamma'\left(1/\alpha_t\right)\left(-\frac{1}{\alpha_t^2}\right).$$

Note that

$$\alpha_t = e^{\eta_t},$$

which implies,

$$\frac{\partial \alpha_t}{\partial \eta_t} = \frac{\partial e^{\eta_t}}{\partial \eta_t} = e^{\eta_t} = \alpha_t.$$

By chain rule, the first derivative of the function A with respect to η_t is:

$$\frac{\partial A}{\partial \eta_t} = \frac{\partial A}{\partial \alpha_t}\frac{\partial \alpha_t}{\partial \eta_t} = \frac{\partial A}{\partial \alpha_t}\alpha_t,$$

and the second derivative of the function A with respect to η_t is:

$$\begin{aligned}
\frac{\partial^2 A}{\partial \eta_t^2} &= \frac{\partial}{\partial \alpha_t}\left(\frac{\partial A}{\partial \eta_t}\right)\alpha_t + \frac{\partial A}{\partial \alpha_t}\frac{\partial \alpha_t}{\partial \eta_t} \\
&= \frac{\partial}{\partial \alpha_t}\left(\frac{\partial A}{\partial \alpha_t}\frac{\partial \alpha_t}{\partial \eta_t}\right)\alpha_t + \frac{\partial A}{\partial \alpha_t}\frac{\partial \alpha_t}{\partial \eta_t}
\end{aligned}$$

$$\begin{aligned}
&= \frac{\partial^2 A}{\partial \alpha_t^2}\frac{\partial \alpha_t}{\partial \eta_t}\alpha_t + \frac{\partial A}{\partial \alpha_t}\frac{\partial \alpha_t}{\partial \eta_t} \\
&= \frac{\partial^2 A}{\partial \alpha_t^2}\alpha_t^2 + \frac{\partial A}{\partial \alpha_t}\alpha_t \\
&= \left(\frac{\partial^2 A}{\partial \alpha_t}\alpha_t + \frac{\partial A}{\partial \alpha_t}\right)\alpha_t,
\end{aligned}$$

which implies,

$$\frac{\partial^2 A}{\partial \eta_t \partial \eta_t^T} = \begin{pmatrix} \left(\frac{\partial^2 A}{\partial \alpha_1}\alpha_1^2 + \frac{\partial A}{\partial \alpha_1}\right)\alpha_1 & \cdots & \frac{\partial^2 A}{\partial \alpha_K \partial \alpha_1}\alpha_K \alpha_1 \\ \frac{\partial^2 A}{\partial \alpha_t \partial \alpha_k}\alpha_t \alpha_k & \ddots & \frac{\partial^2 A}{\partial \alpha_t \partial \alpha_k}\alpha_t \alpha_k \\ \frac{\partial^2 A}{\partial \alpha_1 \partial \alpha_g}\alpha_1 \alpha_K & \cdots & \left(\frac{\partial^2 A}{\partial \alpha_K^2}\alpha_K + \frac{\partial A}{\partial \alpha_K}\right)\alpha_K \end{pmatrix},$$

for $k \neq t$, where

$$\frac{\partial^2 A}{\partial \eta_t \partial \eta_k} = \frac{\partial^2 A}{\partial \alpha_t \partial \alpha_k}\alpha_t \alpha_k = 0.$$

By applying the Newton-Raphson method

$$\boldsymbol{\eta} = \begin{pmatrix} \eta_1 \\ \vdots \\ \eta_K \end{pmatrix}^{(s+1)} = \begin{pmatrix} \eta_1 \\ \vdots \\ \eta_K \end{pmatrix}^{(s)} - \left(\frac{\partial^2 A}{\partial \boldsymbol{\eta} \partial \boldsymbol{\eta}^T}\right) \begin{pmatrix} \frac{\partial A}{\partial \alpha_1}\alpha_1 \\ \vdots \\ \frac{\partial A}{\partial \alpha_K}\alpha_K \end{pmatrix},$$

the updated η_t is

$$\eta_t^{(s+1)} = \eta_t^s - \left(\frac{\partial^2 A}{\partial \alpha_t^2}\alpha_t + \frac{\partial A}{\partial \alpha_t}\right)^{-1} \frac{\partial A}{\partial \alpha_t}\Big|_{\alpha_t = \alpha_t^{(s)}}.$$

Hence, the t^{th} component of the estimated shape parameter α, $\hat{\alpha_t}$, after the $(s+1)^{th}$ iteration is:

$$\hat{\alpha}_t^{(s+1)} = \alpha_t^{(s)} \exp\left(- \left(\frac{\partial^2 A}{\partial \alpha_t^2}\alpha_t + \frac{\partial A}{\partial \alpha_t}\right)^{-1} \frac{\partial A}{\partial \alpha_t}\Big|_{\alpha_t = \alpha_t^{(s)}} \right),$$

for $1 \leqslant t \leqslant K$.

To avoid extreme values of α_t, we added a small constraint value r, such that

$$\alpha_t - r = e^{\eta_t}.$$

The first and second derivative of A with respect to η_t becomes:

$$\begin{aligned}
\frac{\partial A}{\partial \eta_t} &= \frac{\partial A}{\partial \alpha_t}(\alpha_t - r) \\
\frac{\partial^2 A}{\partial \eta_t^2} &= \frac{\partial^2 A}{\partial \alpha_t^2}(\alpha_t - r)^2 + \frac{\partial A}{\partial \alpha_t}(\alpha_t - r),
\end{aligned}$$

therefore, the updated α_t on the $(s+1)^{th}$ iteration is:

$$\begin{aligned}
\hat{\alpha}_t^{(s+1)} &= r + (\alpha_t^{(s)} - r)e^{-\left(\frac{\partial^2 A}{\partial \alpha_t^2}(\alpha_t - r)^2 + \frac{\partial A}{\partial \alpha_t}(\alpha_t - r)\right)^{-1} \frac{\partial A}{\partial \alpha_t}(\alpha_t - r)\Big|_{\alpha_t = \alpha_t^{(s)}}} \\
&= r + (\alpha_t^{(s)} - r)e^{-\left(\frac{\partial^2 A}{\partial \alpha_t^2}(\alpha_t - r) + \frac{\partial A}{\partial \alpha_t}\right)^{-1} \frac{\partial A}{\partial \alpha_t}\Big|_{\alpha_t = \alpha_t^{(s)}}},
\end{aligned}$$

for $1 \leqslant t \leqslant K$.

Estimate π_t

We update the weight parameter π_t, $1 \leq t \leq K$ by the direct approach. We know that

$$\tau_t(y_i | \mathbf{x}_i, \mathbf{\Psi}^{(s)})^{(s+1)} = \frac{\pi_t^{(s)} f_t\left(y_i \middle| \mathbf{x}_i^T \boldsymbol{\beta}_t^{(s)}, \sigma_t^{2^{(s)}}, \alpha_t^{(s)}\right)}{\sum_{t=1}^K \pi_t^{(s)} f_t\left(y_i \middle| \mathbf{x}_i^T \boldsymbol{\beta}_t^{(s)}, \sigma_t^{2^{(s)}}, \alpha_t^{(s)}\right)}. \tag{6.9}$$

By rearranging equation (6.9), we have an estimation of the weight parameter,

$$\hat{\pi}_t^{(s+1)} = \frac{1}{n}\sum_{i=1}^n \tau_t(y_i | \mathbf{x}_i, \mathbf{\Psi}^{(s)}). \tag{6.10}$$

To complete the process, we iterate the above two steps until a stopping criterion is attained.

The structure of the EM algorithm is rather straight forward and relatively simple for programming as can be seen, but there are still some technical issues that need to be taken care of. The first issue is focused on how to choose the initial values, as the bad initial values may lead the results to a local maximum rather than the global maximum. Another issue that needs to be considered is the stopping criterion, we currently let the program stop when the log-likelihood is lack of improvement.

By selecting the best fit, we still use AIC, BIC and EBIC criterion, which add the penalty on the complete log-likelihood and balanced it with the number of free parameters. The total number of parameters need to be calculated in all selection criterion. For the GM regression model, the number of parameters of π is $k-1$, σ^2 is k, $\boldsymbol{\beta}$ is $k + \sum_{k=1}^{K}\sum_{j=1}^{P} I(|\hat{\beta}_{kj}|\neq 0)$, where $I(.)$ is an indicator function. By fixing its shape parameters equal to 2, it gives the

$$\begin{aligned}
\text{AIC} &= -2\tilde{l}_c(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X}) + 2(3K - 1 + \sum_{k=1}^{K}\sum_{j=1}^{P} I_{(|\hat{\beta}_{kj}|\neq 0)})\log(n), \\
\text{BIC} &= -2\tilde{l}_c(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X}) + (3K - 1 + \sum_{k=1}^{K}\sum_{j=1}^{P} I_{(|\hat{\beta}_{kj}|\neq 0)})\log(n), \\
\text{EBIC} &= -2\tilde{l}_c(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X}) + (3K - 1 + \sum_{k=1}^{K}\sum_{j=1}^{P} I_{(|\hat{\beta}_{kj}|\neq 0)})\log(n) + 2\gamma\log\kappa(S_j),
\end{aligned}$$

for $0 \leq \gamma \leq 1$.

Similarly to the EPD mixture regression model, the number of parameters of π is $K-1$, σ^2 is K, $\boldsymbol{\beta}$ is $K + \sum_{k=1}^{K}\sum_{j=1}^{P} I(|\hat{\beta}_{kj}|\neq 0)$, α is K, we obtain

$$\begin{aligned}
\text{AIC} &= -2\tilde{l}_c(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X}) + 2(4K-1+\sum_{k=1}^{K}\sum_{j=1}^{P} I_{(|\hat{\beta}_{kj}|\neq 0)})\log(n),\\
\text{BIC} &= -2\tilde{l}_c(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X}) + (4K-1+\sum_{k=1}^{K}\sum_{j=1}^{P} I_{(|\hat{\beta}_{kj}|\neq 0)})\log(n),\\
\text{EBIC} &= -2\tilde{l}_c(\boldsymbol{\Psi}|\mathbf{y},\mathbf{X}) + (4K-1+\sum_{k=1}^{K}\sum_{j=1}^{P} I_{(|\hat{\beta}_{kj}|\neq 0)})\log(n) + 2\gamma\log\kappa(S_j),
\end{aligned}$$

for $0 \leq \gamma \leq 1$.

There are a number of different penalties which are commonly used nowadays. They all have different strengths:

$$\begin{aligned}
&\text{LASSO: } P_\lambda(z) = \lambda z, P'\left(\frac{\left|\beta_{tp}^{(s)}\right|}{\sigma_t^{()}}\right) = \lambda\\
&\text{Adaptive LASSO: } P_\lambda(z) = \lambda\log z, P'\left(\frac{\left|\beta_{tp}^{(s)}\right|}{\sigma_t^{(s)}}\right) = \lambda\frac{\sigma_t^{(s)}}{\left|\beta_{tp}^{(s)}\right|}\\
&\text{SCAD: } P'_\lambda(|z|) = \lambda I(|z| \leq \lambda) + \tfrac{(a\lambda-|z|)_+}{a-1} I(|z| > \lambda) \quad a = 3.7\\
&\qquad P'\left(\frac{\left|\beta_{tp}^{(s)}\right|}{\sigma_t^{(s)}}\right) = \lambda I(\left|\beta_{tp}^{(s)}\right| \leq \lambda\sigma_t^{(s)}) + \frac{(a\lambda-\frac{\left|\beta_{tp}^{(s)}\right|}{\sigma_t^{(s)}})_+}{a-1} I(\frac{\left|\beta_{tp}^{(s)}\right|}{\sigma_t^{(s)}} > \lambda)\\
&\text{Bridge: } P_\lambda(z) = \lambda z^T, P'\left(\frac{\left|\beta_{tp}^{(s)}\right|}{\sigma_t^{(s)}}\right) = \lambda r\left(\frac{\left|\beta_{tp}^{(s)}\right|}{\sigma_t^{(s)}}\right)^{r-1}, 0 < r < 1
\end{aligned}$$

6.2.2 Simulations

In this section, we assess the accuracy of the exponential power distribution mixture(EPD) based regression clustering of a two-component sparse model, and compare the results with Gaussian mixture(GM) model based regression clustering. The following scenarios are considered:

1. different number of dimensions;
2. different proportion weight of each component;
3. different correlation structure of covariates;
4. equal and unequal component variances;
5. different sample size.

We let $\mathbf{X} = (\mathbf{x_1}, \ldots, \mathbf{x_n})^{\mathbf{T}}$ be the covariate matrix, and $\mathbf{x_i} = (1, x_{i1}, \ldots, x_{iP})^T = (1, \mathbf{x}_i^*)^T$, where $\mathbf{x}_i^* = (x_{i1}, \ldots, x_{iP})^T$ denote the i^{th} observations on P covariates. We generated $\mathbf{x}_i^*$ from $N(0, \Sigma_x)$, where $\Sigma_x = (\rho^{|l-m|})_{1 \le l,m \le P}$ and $0 \le \rho \le 1$ is a constant. For $i = 1, \ldots, n$, conditional on $\mathbf{x}_i$, y_i be an observation drawn from the model

$$f(y_i|\mathbf{x}_i, \mathbf{\Psi}) = \pi_1 f_1(y_i|\mathbf{x}_i, \theta_1) + \pi_2 f_2(y_i|\mathbf{x}_i, \theta_2),$$

where $y \in \mathbb{R}^1$ and $\mathbf{x} \in \mathbb{R}^{\mathbf{P+1}}$.

We let the shape parameters be randomly chosen from a uniform distribution with boundaries 1 and 2, then we have $(\alpha_1, \alpha_2) = (1.05, 1.12)$. Let the number of non-zero regression coefficients in $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are 4 and 6 receptively. They are randomly generated from $(-1)^u(2.5+z)$, where u is a random variable from the Bernoulli distribution with parameter 0.4 and z from a standard Gaussian distribution, similar to Fan and Lv(2008). Hence we denote the non-zero coefficients by

$$\boldsymbol{\beta}_{1*} = \{-2.25, 3.71, -2.95, -3.17\}, \boldsymbol{\beta}_{2*} = \{3.15, 2.29, 3.28, -3.04, 2.74, -4.09\}. \quad (6.11)$$

We consider different combination of regression coefficients with the rest parameter: n : $(300, 400)$, ρ : $(0.5, 0.75)$, P : $(10, 100)$, (σ_1, σ_2): $(1, 1)$, $(2, 2)$, $(1.5, 2.5)$, $(1.5, 3.5)$, (π_1, π_2): $(0.5, 0.5)$, $(0.3, 0.5)$.

The averaged minimum BICs, AIC, EBIC, RAND indexs and standard errors for GM-based clustering and EPD-based clustering are derived over 100 replications in Table 6.1 and Table 6.2.

Table 6.1: Averaged minimum AIC, BIC, EBIC, and RAND indexes of GM-based clustering and EPD-based clustering when the variance are equal. (the numbers in the parenthesis are the standard errors)

Equal variance cases: $n = 100$ and $p = 10, 100$

Cases			$\pi = 0.5$			
			$p = 10$		$p = 100$	
ρ	σ		GM	EPD	GM	EPD
0.50	1	RAND	$0.869_{(0.004)}$	$0.899_{(0.003)}$	$0.742_{(0.003)}$	$0.773_{(0.004)}$
		BIC	$1416.291_{(2.027)}$	$1400.501_{(2.036)}$	$1809.944_{(2.132)}$	$1768.973_{(3.388)}$
		AIC	$1323.596_{(2.024)}$	$1300.506_{(2.033)}$	1754.387(2.135)	$1706.009_{(3.382)}$
		EBIC	$1423.804_{(2.027)}$	$1408.121_{(2.035)}$	$1826.958_{(2.134)}$	$1785.987_{(3.398)}$
	2	RAND	$0.783_{(0.010)}$	$0.815_{(0.010)}$	$0.706_{(0.004)}$	$0.720_{(0.004)}$
		BIC	$1595.397_{(3.903)}$	$1576.975_{(4.255)}$	$1926.769_{(3.321)}$	$1891.749_{(4.237)}$
		AIC	$1502.802_{(3.912)}$	$1476.973_{(4.322)}$	$1871.212_{(3.332)}$	$1828.785_{(3.988)}$
		EBIC	$1603.010_{(3.908)}$	$1584.588_{(4.312)}$	$1943.783_{(3.232)}$	$1908.763_{(4.386)}$
0.75	1	RAND	$0.873_{(0.003)}$	$0.879_{(0.003)}$	$0.762_{(0.003)}$	$0.833_{(0.003)}$
		BIC	$1412.683_{(1.868)}$	$1400.731_{(1.946)}$	$1778.587_{(3.313)}$	$1759.399_{(2.998)}$
		AIC	$1320.088_{(1.876)}$	$1300.729_{(1.943)}$	$1723.030_{(3.321)}$	$1696.435_{(2.956)}$
		EBIC	$1420.296_{(1.867)}$	$1408.344_{(1.899)}$	$1795.601_{(3.132)}$	$1776.413_{(3.134)}$
	2	RAND	$0.808_{(0.004)}$	$0.829_{(0.003)}$	$0.607_{(0,003)}$	$0.751_{(0.003)}$
		BIC	$1580.381_{(2.236)}$	$1568.245_{(2.120)}$	$1881.648_{(4.211)}$	$1876.819_{(3.179)}$
		AIC	$1487.786_{(2.027)}$	$1468.243_{(2.136)}$	$1826.091_{(4.132)}$	$1813.855_{(3.332)}8$
		EBIC	$1587.994_{(2.232)}$	$1575.858_{(2.037)}$	$1898.662_{(4.167)}$	$1893.833_{(3.334)}$

Cases			$\pi = 0.3$			
			$p = 10$		$p = 100$	
ρ	σ		GM	EPD	GM	EPD
0.50	1	RAND	$0.928_{(0.002)}$	$0.932_{(0.003)}$	$0.821_{(0.003)}$	$0.847_{(0.003)}$
		BIC	$1375.105_{(1.736)}$	$1363.751_{(2.015)}$	$1759.180_{(1.340)}$	$1736.271_{(2.271)}$
		AIC	$1282.510_{(1.677)}$	$1263.749_{(2.016)}$	$1703.623_{(1.235)}$	$1673.307_{(2.136)}$
		EBIC	$1382.718_{(11.878)}$	$1371.364_{(2.031)}$	$1776.194_{(1.356)}$	$1753.285_{(2.648)}$
	2	RAND	$0.882_{(0.003)}$	$0.892_{(0.003)}$	$0.716_{(0.004)}$	$0.809_{(0.005)}$
		BIC	$1540.099_{(2.438)}$	$1527.678_{(2.246)}$	$1870.817_{(3.432)}$	$1837.471_{(4.113)}$
		AIC	$1447.504_{(2.317)}$	$1427.676_{(2.147)}$	$1815.260_{(3.132)}$	$1774.507_{(3.364)}$
		EBIC	$1547.712_{(2.507)}$	$1535.291_{(2.245)}$	$1887.831_{(3.652)}$	$1854.485_{(4.128)}$
0.75	1	RAND	$0.890_{(0.003)}$	$0.917_{(0.003)}$	$0.846_{0.003}$	$0.846_{0.003}$
		BIC	$1391.807_{(1.666)}$	$1359.227_{(2.511)}$	$1707.693_{(2.271)}$	$1702.403_{(1.339)}$
		AIC	$1299.212_{(1.538)}$	$1259.225_{(2.431)}$	$1652.136_{(2.132)}$	$1639.439_{(1.388)}$
		EBIC	$1399.420_{(1.703)}$	$1366.840_{(2.536)}$	$1724.707_{(2.342)}$	$1719.417_{(1.393)}$
	2	RAND	$0.862_{(0.003)}$	$0.870_{(0.003)}$	$0.672_{(0.003)}$	$0.728_{(0.003)}$
		BIC	$1529.881_{(1.807)}$	$1525.886_{(1.932)}$	$1854.786_{(3.317)}$	$1833.108_{(2.229)}$
		AIC	$1437.286_{(1.708)}$	$1425.884_{(1.896)}$	$1799.229_{(3.132)}$	$1770.411_{(2.213)}$
		EBIC	$1537.494_{(1.877)}$	$1533.499_{(2.036)}$	$1871.800_{(3.154)}$	$1850.122_{(2.764)}$

For Table 6.1 and 6.2, generally speaking, the EPD-based clustering give better prediction than GM-based clustering under all scenarios. Both EPD and GM based clustering are more accurate when the dimensions are small than when the dimensions are large. When the ρ is large, or the variance are small, we also obtain better clustering results for both method, compare to the

Table 6.2: Averaged minimum value of AIC, BIC, EBIC and RAND indexes for GM-based clustering and EPD-based clustering when the variacne are unequal. (the numbers in the parenthesis are the standard errors)

Unequal variance cases: $n = 400$ and $p = 10, 100$							
Cases				$\pi = 0.5$			
				$p = 10$		$p = 100$	
ρ	σ_1	σ_2		GM	EPD	GM	EPD
0.5	1.5	2.5	RAND	$0.818_{(0.004)}$	$0.847_{(0.004)}$	$0.749_{(0.002)}$	$0.770_{(0.004)}$
			BIC	$1571.848_{(1.948)}$	$1556.641_{(1.927)}$	$2054.470_{(2.044)}$	$1901.772_{(3.246)}$
			AIC	$1472.061_{(1.848)}$	$1448.871_{(1.924)}$	$1994.598_{(2.034)}$	$1833.917_{(3.241)}$
			EBIC	$1579.461_{(1.965)}$	$1564.254_{(1.957)}$	$2071.484_{(2.047)}$	$1918.786_{(3.266)}$
	1.5	3.5	RAND	$0.812_{(0.003)}$	$0.849_{(0.004)}$	$0.762_{(0.003)}$	$0.783_{(0.003)}$
			BIC	$1623.789_{(2.215)}$	$1608.422_{(2.185)}$	$2183.134_{(2.548)}$	$2156.227_{(2.922)}$
			AIC	$1524.002_{(2.157)}$	$1500.652_{(2.148)}$	$2123.262_{(2.544)}$	$2088.372_{(2.297)}$
			EBIC	$1631.402_{(2.264)}$	$1616.035_{(2.235)}$	$2200.148_{(2.765)}$	$2173.241_{(3.227)}$
0.75	1.5	2.5	RAND	$0.819_{(0.004)}$	$0.841_{(0.004)}$	$0.772_{(0.002)}$	$0.799_{(0.002)}$
			BIC	$1570.168_{(2.258)}$	$1552.433_{(2.036)}$	$1892.174_{(2.577)}$	$1861.298_{(2.380)}$
			AIC	$1470.381_{(1.946)}$	$1444.663_{(1.921)}$	$1832.302_{(2.042)}$	$1793.443_{(2.264)}$
			EBIC	$1577.781_{(2.323)}$	$1560.046_{(2.027)}$	$1909.188_{(2.579)}$	$1878.312_{(3.244)}$
	1.5	3.5	RAND	$0.819_{(0.004)}$	$0.846_{(0.004)}$	$0.786_{(0.003)}$	$0.803_{(0.003)}$
			BIC	$1606.433_{(2.655)}$	$1594.771_{(2.907)}$	$2112.425_{(2.441)}$	$2082.483_{(2.084)}$
			AIC	$1505.646_{(2.555)}$	$1487.001_{(2.902)}$	$2052.553_{(2.431)}$	$2014.628_{(2.057)}$
			EBIC	$1614.046_{(2.659)}$	$1602.384_{(2.943)}$	$2129.439_{(2.465)}$	$2099.497_{(2.098)}$
Cases				$\pi = 0.3$			
				$p = 10$		$p = 100$	
ρ	σ_1	σ_2		GM	EPD	GM	EPD
0.5	1.5	2.5	RAND	$0.873_{(0.002)}$	$0.902_{(0.002)}$	$0.811_{(0.003)}$	$0.852_{(0.003)}$
			BIC	$1551.088_{(2.086)}$	$1541.228_{(1.520)}$	$2049.518_{(3.017)}$	$1900.989_{(2.460)}$
			AIC	$1451.301_{(2.055)}$	$14733.458_{(1.430)}$	$1989.646_{(3.003)}$	$1833.134_{(2.264)}$
			EBIC	$1558.701_{(2.135)}$	$1548.841_{(1.640)}$	$2066.532_{(3.232)}$	$1918.003_{(2.478)}$
	1.5	3.5	RAND	$0.813_{(0.002)}$	$0.829_{(0.002)}$	$0.739_{(0.002)}$	$0.784_{(0.002)}$
			BIC	$1626.796_{(1.886)}$	$1611.954_{(2.013)}$	$2288.485_{(2.537)}$	$2073.683_{(2.166)}$
			AIC	$1527.009_{(1.763)}$	$1504.184_{(2.007)}$	$2228.613_{(2.441)}$	$2005.408_{(2.024)}$
			EBIC	$1634.409_{(1.827)}$	$1619.567_{(2.347)}$	$2305.499_{(2.568)}$	$2090.277_{(2.254)}$
0.75	1.5	2.5	RAND	$0.846_{(0.008)}$	$0.866_{(0.004)}$	$0.801_{(0.003)}$	$0.827_{(0.003)}$
			BIC	$1580.306_{(3.869)}$	$1561.640_{(2.302)}$	$1938.593_{(2.234)}$	$1901.948_{(2.711)}$
			AIC	$1480.519_{(3.655)}$	$1453.870_{(2.093)}$	$1878.721_{(2.131)}$	$1834.093_{(2.643)}$
			EBIC	$1587.919_{(3.926)}$	$1569.253_{(2.453)}$	$1955.607_{(2.448)}$	$1918.962_{(2.884)}$
	1.5	3.5	RAND	$0.789_{(0.011)}$	$0.805_{(0.008)}$	$0.832_{(0.002)}$	$0.851_{(0.003)}$
			BIC	$1629.619_{(4.285)}$	$1611.077_{(3.522)}$	$2199.284_{(3.172)}$	$2173.128_{(2.017)}$
			AIC	$1529.832_{(4.654)}$	$1503.307_{(3.246)}$	$2139.412_{(2.984)}$	$2105.273_{(2.002)}$
			EBIC	$1637.232_{(4.755)}$	$1618.690_{(3.907)}$	$2216.298_{(3.487)}$	$2190.142_{(2.084)}$

scenario that the ρ is small or the variance are large. The accuracy of clustering prediction is also effect by the variety of variance, we obtain larger RAND index for both methods when the variance are equal compare to the scenario that the variance of the two component are unequal.

6.3 Forward Selection

For the dataset with very high dimension and limited sample size, it is very difficult to cluster the data in full feature space, hence the forward selection method is a good way to pre-screen the data before we apply variable selection method in later stage. Forward selection method includes forward stepwise selection and forward stagewise selection which is similar to forward stepwise selection,but adding some constrains. We will discuss both methods in the next two sections and following with some simulation studies.

6.3.1 Forward stepwise regression

Forward selection also called forward stepwise regression is a classic model selection method, which predicts variables by carrying out an automatic procedure (Hocking, 1976; Weisberg, 1980; Draper and Smith, 1981). The first algorithm was proposed in Efroymson's procedure which combined both forward and backward stepwise regression (Efroymson, 1960). It starts with no predictor variables in the model, and then begins by selecting a single variable which give the best fit by finding the smallest residual sum of squares. Then we add another variable and choose the one which improves the model most in combination with the first. After that, we add a third variable, and choose the one which improves the model most in combination with the first two. We repeat this procedure until a certain stopping criteria is reached, such as the model lacks further improvement or a certain number of predictors in the model is reached.

For a simple regression model,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{6.12}$$

where $\mathbf{y}$ = is the response vector of length n, the covariate matrix $\mathbf{X}$ is an n by $(P+1)$ matrix such that $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n})^{\mathrm{T}}$, and for $i = 1, 2, ..., n$ we have $\mathbf{x_i} = (1, x_{i1}, x_{i2}, ..., x_{iP})^T$, and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_P)$ is a regression coefficient vector of length $P + 1$. The corresponding regression coefficient will be set to 0 when a covariate is not selected. The error vector ϵ satisfied that $E(\epsilon) = 0$, which implies $E(y_i|\mathbf{x}_i) = \beta_0+\beta_1\mathbf{x}_{i1}+\beta_2\mathbf{x}_{i2}+...+\beta_P\mathbf{x}_{iP}$, and that $Var(\epsilon) = \sigma^2 I_n$ with I_n is an $n \times n$ unite matrix.

Among the covariates, we select the one with the largest absolute correlation with y, say $\mathbf{x_1}$, then we have

$$\mathbf{y} = \mathbf{x_1}\beta_1 + \epsilon_1, \tag{6.13}$$

then we consider the new response with the residual orthogonal to $\mathbf{x_1}$, i.e

$$\mathbf{y}^* = \mathbf{y} - \mathbf{x_1}\beta_1 + \epsilon_1, \tag{6.14}$$

we find another predictor which is orthogonal to $\mathbf{x_1}$ and repeat these selection steps. After k steps, we have a predictor set $\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_k}$, then we have

$$\mathbf{y} = \mathbf{x_1}\beta_1 + \ldots + \mathbf{x_k}\beta_k + \boldsymbol{\epsilon}. \tag{6.15}$$

Forward stepwise regression is a greedy method, it makes the best change for each step rather than the whole picture, and a small change in the data may have a big impact on the result of the selection. The advantage of forward stepwise regression is when it selects the best combination of two predictors it does not have to include the best isolation predictor from the first step. The disadvantage of this selection method is that it contains a bias due to its greedy nature for each step.

6.3.2 Forward stagewise regression

Forward stagewise regression is similar to forward stepwise regression but with more constraints, it has very similar behaviour with LASSO, and there is also a strong connection between forward stagewise regression and the boosting algorithm in machine learning, for example (Efron et al., 2004; Hasite et al., 2001).

It contains thousands of tiny steps before it reaches the final model, and provides the motivation for the LARS(Least Angle Regression) algorithm which allows forward stagewise to use relatively large steps. Stagewise regression select the first predictor variable in the same way as stepwise regression, but the corresponding coefficient only changes a small amount. The next step is take the variable with the highest correlation with the current residual and take a small step for that variable and repeat this procedure until a stopping criterion is reached (Hesterbery et al., 2008).

The advantage of forward stagewise regression is that it is stable and it takes a number of steps before a variable is clearly selected, but its disadvantage is the computational burden as a result of too many steps during the model selection. The forward stagewise regression and LASSO have very similar behaviour in many cases such as the orthogonal predictor case, etc.

6.3.3 Simulations

In this section we assess the performance of the model with two simulations. In the first simulation, we divide the predict matrix $\mathbf{X}$ into two parts: $\mathbf{X}_1$ is the matrix where observation y is conditional on, and the $\mathbf{X}_2$ is the noise matrix, we generate $\mathbf{X}_1$ and $\mathbf{X}_{noise}$ independently. In the second simulation, we let the two parts of the prediction matrix X_1 and X_{noise} be dependent of each other.

Simulation 1

We let $\mathbf{X_1} = (\mathbf{x_1}, \ldots, \mathbf{x_{100}})^{\mathbf{T}}$ be the covariate matrix, and $\mathbf{x_i} = (1, x_{i1}, \ldots, x_{i10})^T = (1, \mathbf{x}_i^*)^T$, where $\mathbf{x}_i^* = (x_{i1}, \ldots, x_{i10})^T$ denote the i^{th} observations on 10 covariates. We generated $\mathbf{x}_i^*$ from $N(0, \Sigma_x)$, where $\Sigma_x = (\rho^{|l-m|})_{1\leq l,m \leq P}$, $\rho = (0.5, 0.75)$. For $i = 1, \ldots, n$, conditional on $\mathbf{x}_i$, y_i be an observation drawn from the model

$$f(y_i|\mathbf{x}_i, \mathbf{\Psi}) = \pi_1 f_1(y_i|\mathbf{x}_i, \theta_1) + \pi_2 f_2(y_i|\mathbf{x}_i, \theta_2),$$

where $y \in \mathbb{R}^1$ and $\mathbf{x} \in \mathbb{R}^{(\mathbf{p+1})}$.

with the total sample size $n = 100$. A standard multivariate Gaussian $N(0, I_{P-10})$ was used for the generation the noise part of the predict matrix X_{noise}. We let $P_{noise} = 200$, hence the total number of dimension of $\mathbf{X}$ is $P = 211$. For the first 10 rows of the corresponding covariates matrix $\boldsymbol{\beta}$, the number of non-zero coefficients in $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are 4 and 6 respectively. The non-zero regression coefficients are randomly generated from $(-1)^u(2.5 + z)$, where u is a random variable of Bernoulli distribution with parameter 0.4 and z from a standard Gaussian distribution, similar to Fan and Lv(2008). Hence we have non-zero coefficients for the first ten rows of corresponding covariates which are

$$\boldsymbol{\beta}_{1*} = \{-2.25, 3.71, -2.95, -3.17\}, \boldsymbol{\beta}_{2*} = \{3.15, 2.29, 3.28, -3.04, 2.74, -4.09\}, \tag{6.16}$$

and the rest entries of the regression coefficient $\boldsymbol{\beta}$ are zeros.

We let the shape parameters be randomly chosen from a uniform distribution with boundaries 1 and 2, then we have $(\alpha_1, \alpha_2) = (1.05, 1.12)$. We also consider the variance : $\sigma_1 = \sigma_2 = 1$. The proportion parameter π is assigned to be 0.5.

In this simulation, we reduce the number of variables by using one-step forward selection to $N_v = 10$, which is about 5% of the number of the sample size. In each of the forward selection iterations, we choose the best 10 combinations of variables with the maximum penalised likelihood. In the results, we consider 3 parameters:

$$\begin{aligned} r_1 &= \frac{\text{the number of true variables selected}}{\text{total number of true variables}}; \\ r_2 &= \frac{\text{the number of true variables that missed}}{\text{total number of true variables}}; \\ r_3 &= \frac{\text{the number of noisy variables selected}}{\text{total number of noisy variables}}. \end{aligned}$$

as $N_v \to n$, we expect $r_1 \to 1$, $r_2 \to 0$ and $r_3 \to 0$.

We let $D = 20$ datasets, the results show in Table 6.3:

From Table 6.3 we can see that in this simulation if we select the number of variables to be approximately 5% of the sample size, more than 80% of true variables are selected.

Table 6.3: The truth variables selected rate, the missing variables rate and the noise variables selected rate (the numbers in the parenthesis are the standard errors).

r_1	r_2	r_3
$0.817_{(0.178)}$	$0.183_{(0.212)}$	$0.010_{(0.196)}$

The Table 6.4 presents the best combinations for the first 10 iterations of forward selection. The best combinations are chosen by calculating their maximum penalised likelihood. The result shows that about 80% of the true variables were selected within the first 10 steps.

Table 6.4: Best combinations for the first 10 iterations of forward selection

	combination 1	combination 2	combination 3	combination 4	combination 5	combination 6	combination 7	combination 8	combination 9	combination 10
step 1	4	4	4	4	4	4	4	4	4	4
step 2	7	7	7	7	7	7	7	7	7	7
step 3	11	11	11	11	11	11	11	11	11	11
step 4	6	6	6	6	6	6	6	6	6	6
step 5	5	5	5	5	5	5	5	5	5	5
step 6	3	3	3	3	3	3	3	3	3	3
step 7	2	2	2	2	2	2	2	2	2	2
step 8	49	161	49	161	49	161	161	49	49	49
step 9	8	158	8	158	8	158	158	8	53	8
step 10	9	71	53	186	50	26	65	28	47	45

Simulation 2

In the second simulation, we let the predict matrix X_1 and the noise matrix X_{noise} are dependent on each other, we let X_1 and X_{noise} generated together from a multivariate Gaussian distribution $N(0, \Sigma)$, where $\Sigma = (\rho^{|l-m|})_{1 \leq l,m \leq P}$, for $i \neq j$ and $\rho = 0.5$, we let the dimension of X equals to 211.

For the first 10 rows of the corresponding covariates matrix $\boldsymbol{\beta}$, the number of non-zero coefficients in $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are 4 and 6 respectively. The non-zero regression coefficients are randomly generated from $(-1)^u(2.5+z)$, where u is a random variable of Bernoulli distribution with parameter 0.4 and z from a standard Gaussian distribution, similar to Fan and Lv(2008). Hence we have the non-zero coefficients of the first ten rows of corresponding covariates are

$$\boldsymbol{\beta}_{1*} = \{-2.25, 3.71, -2.95, -3.17\}, \boldsymbol{\beta}_{2*} = \{3.15, 2.29, 3.28, -3.04, 2.74, -4.09\},$$

and the remainder of the entries of $\boldsymbol{\beta}$ are zeros. Therefore, the observation vector y is only dependent on the non-zero elements of the corresponding covariates matrix $\boldsymbol{\beta}$. All the other parameters are selected same as in simulation 6.3.3.

Similar to simulation 6.3.3, we reduce the number of variables, N_v, to 10, which is about 5% of the sample size. In each forward selection iteration, we choose the best 10 combinations of the variables with maximum penalised likelihood.

We let $D = 20$ datasets, the results are shown in Table 6.5:

From Table 6.5 we can see that, in this simulation we select the number of variables to be about 5% of the sample size, about 70% of the true variables are selected.

Table 6.5: Truth variables selected rate, the missing variables rate and noise variables selected rate for the dependent predictor matrix X (the numbers in the parenthesis are the standard errors of the survival rates).

r_1	r_2	r_3
$0.676_{(0.208)}$	$0.324_{(0.221)}$	$0.016_{(0.163)}$

Compared to simulation 1, it is more difficult to select the true variables if X_1 and X_noise are dependent, Nevertheless, the forward selection method still provides a good opportunity to select most of the true variables within a few steps.

Stopping point When working with the real data with unknown parameters, the timing to stop the iterations becomes a vital step. We can only gain a experience from simulation studies. From both simulations above we found that when we selected most of the true variables, the difference of the penalised log likelihoods between the two steps by adding one noise variable converged to zero as the number of steps tended to the size of its dimensions. From our experience we know that as the number of steps gets larger, the slop of the difference of log-likelihood between the two steps become more stable. We realise that when the increasing rate of step $s+1$ is less than 0.01, the true variables are almost selected. Therefore, we let the standardised stopping point be $sp = 0.01$.

6.4 Real data analysis

In this real data analysis, we are interested in exploring the relative expression of a gene by the binding strengths of a subset of the P candidate motifs to the regulatory region of the gene. This dataset contained the expression levels of 4443 genes and 2155 of motif-matching scores of candidate motifs from Saccharomyces cerevisiae (Conlon et al., 2003). We take a logarithm of our response

variable $\mathbf{y}$ which states the level of genes with $n = 4443$, and the respective matrix, $\mathbf{X_{nP}}$, is a n by P matrix taken by the candidates of the motif-matching scores of motifs. In this dataset, the columns stand for genes while the rows stand for motifs, it is used to show that the BIC value of the GM model can be improved by the EPD approach.
Although in this dataset, the number of dimension P is not larger than the size of genes n, it is still a very high dimension, thus the task of variable selection remains difficult due to the potential accumulation for the noise and the interpret ability to cluster the genes in the full feature space, and as such a pre-screen might be very helpful for the variable selection in later stage. We screen the data by using the one step forward selection, we use single regression model to find the minimum BIC of both GM and EPD estimation for each candidate motif with respect to the response variables, and we fit the models with the number of components $K = 2, 3, 4$ and sorted the minimum BICs in increasing order, thus we have $\text{BICgm} = \{\text{BICgm}_1, \ldots, \text{BICgm}_{2155}\}$ and $\text{BICepd} = \{\text{BICepd}_1, \ldots, \text{BICepd}_{2155}\}$ respectively. Then, we looked for a number of p, such that the variance of the first p BICs with the weight of 95% of the total variance of 2155 BIC values, i.e for GM clustering,

$$var_{gm} = \frac{var(\text{BICgm}_1, \ldots, \text{BICgm}_{p_{gm}})}{var(\text{BICgm}_1, \ldots, \text{BICgm}_P)} = 0.95 \tag{6.17}$$

and for EPD clustering,

$$var_{epd} = \frac{var(\text{BICepd}_1, \ldots, \text{BICepd}_{p_{epd}})}{var(\text{BICepd}_1, \ldots, \text{BICepd}_P)} = 0.95. \tag{6.18}$$

After the calculations we find that, to satisfy equation(6.17) we have $p_{gm} = 155$, and when satisfy equation(6.18) we have $p_{epd} = 176$, which means we now have a much more reduced matrix with 4443 rows and 155 columns for GM clustering and with 4443 rows and 176 columns for EPD clustering. See Figure 6.1:

We apply EPD and GM on the first group of 155 selected motifs with

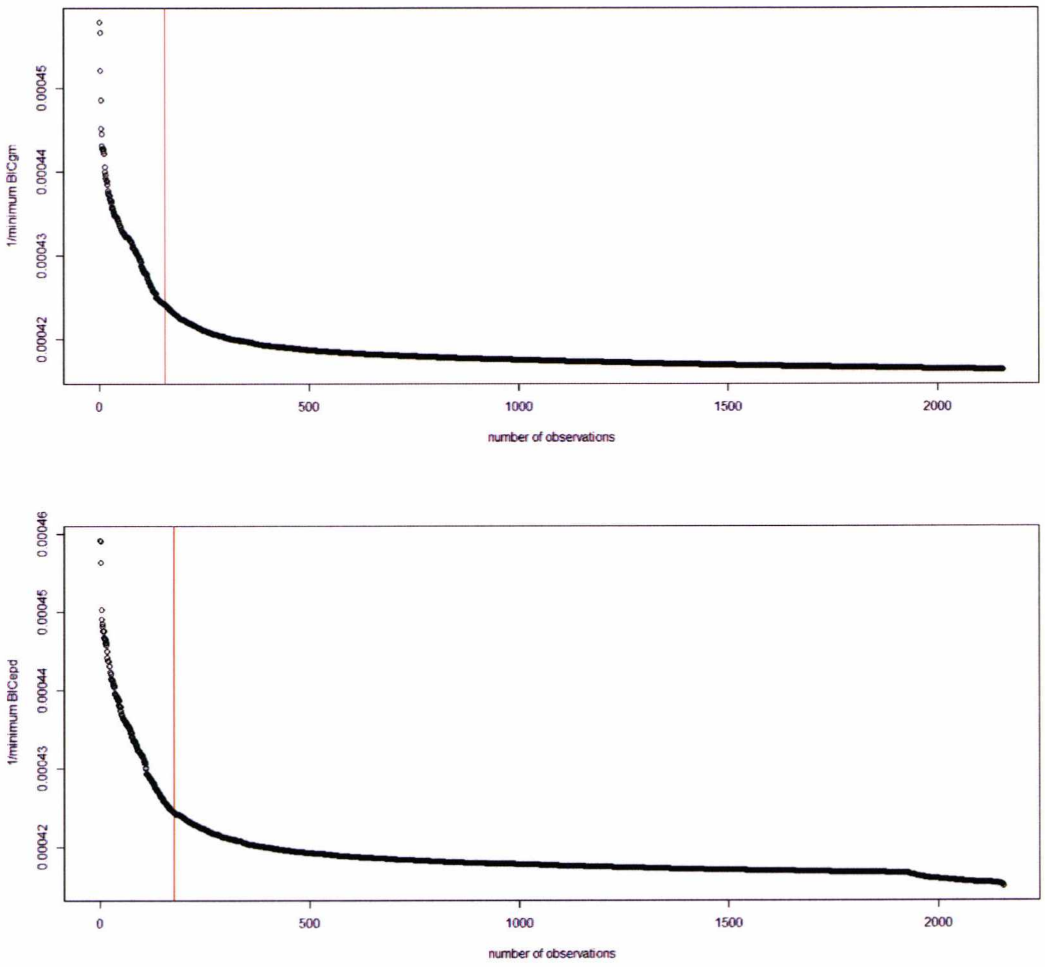

Figure 6.1: The top panel is the plot of the 1/BICgm values, where the red line is the turning point when the variance of the first p 1/BICgm's attains with the 95% of the total variance. The bottom panel is the plot of 1/BICepd values, where the red line is the turning point when the variance of the first p 1/BICepd's attains with the 95% of the total variance.

$K = 1, 2, 3, 4$, the best fit of GM model obtained when the selection criteria are $\min \text{BICgm} = 3141.524$, $\min \text{AICgm} = 1125.812$ and $\min \text{EBICgm} = 3160.298$ respectively, and the best fit of EPD model obtained when $\min \text{BICepd} = 2704.046$, $\min \text{AICepd} = 675.536$ and $\min \text{EBICepd} = 2722.820$. The number of groups is $K = 2$ in all cases based on selection criterion.

For the best fit EPD model, there are 81 variables have been selected, the names of those selected motifs are shown Table B.1 in Appendix B. Table B.2 and Table B.3 list the name of the motifs which have been selected to group 1 and group 2 respectively. After EPD mixture regression based clustering, there are 361 out of 4443 genes have been allocated in group 1, and the rest ones in group 2. The Table B.4 and B.5 list the name of 361 genes in group 1 in Appendix B.

To test whether the result is obtained by chance or not, we use bootstrapping for its justification based on BIC criterion.

Parametric Bootstrapping the BICs for GM and EPD clustering

1. Fit the finite mixture model of regression with GM clustering and EPD clustering to the data of response variable $\mathbf{y}$ and the first group of 155 selected motifs, which leads to the EM estimates $\hat{\mathbf{\Psi}}_{GM}$ and $\hat{\mathbf{\Psi}}_{EPD}$ respectively.

2. Calculate the observed BIC difference, denote this by $\text{DBIC}_o = \min \text{BIC}_{GM} - \min \text{BIC}_{EPD}$, here we have $\text{DBIC}_o = 3141.524 - 2704.046 = 437.478$. The null hypothesis, H_o: GM is true.

3.Simulate a data set of size n from GM distribution with estimated parameters $\hat{\mathbf{\Psi}}_{GM}$, we denote this sample data $y_1^*, \dots, y_n^*$.

4. Fit a finite mixture model with GM and EPD clustering to the simulated data, and calculate the corresponding bootstrap minimum BIC respectively, then find the difference of BICs, $\min \text{BIC}_{GM}^* - \min \text{BIC}_{EPD}^*$, denoted this value by DBIC^*.

5.Repeat step 3 and 4 100 times to generate the bootstrap sampling distribution of the difference between the minimum BICs, the results using a histogram are shown in Figure 6.2.

The histogram shows the results of $\min \text{BIC}_{GM}^* - \min \text{BIC}_{EPD}^*$, while the red vertical line shows where the observed BIC difference of group 1 is.

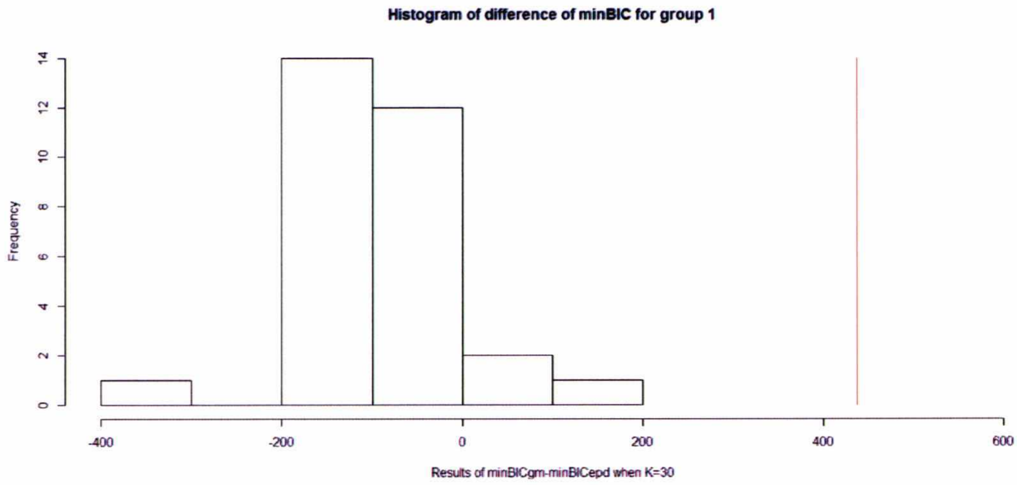

Figure 6.2: The bootstrap sampling distribution of minBICgm-minBICepd with GM fit derived over 100 times, where the red line is the observed value of minBICgm-minBICepd for group 1.

6.The bootstrap p-value is

$$P = \frac{1}{100}\sum_{i=1}^{100} I\{\text{DBIC}_o \leq \text{DBIC}^*\} = 0. \tag{6.19}$$

From the results of the bootstrap we can see that the hypothesis H_o: GM is true is rejected, which means the first dataset we selected from real data is not Gaussian distributed, even when each motif is chosen by the minimum BICs of Gaussian linear regression.

We also undertook another two bootstraps to test the second dataset, firstly we apply EPD and GM clustering to the second group of 176 selected motifs with $K = 1, 2, 3, 4$, the best fit by using GM models is when the selection criterion are $\min \text{BICgm} = 2173.629$,$\min \text{AICgm} = -1256.281$ and $\min \text{EBICgm} = 2192.913$ with the number of groups is $K = 3$, the best fit by using EPD models is when $\min \text{BICepd} = 2079.561$, $\min \text{AICepd} = -217.711$ and $\min \text{EBICepd} = 2098.846$ respectively with the number of groups is $K = 2$ based on BIC selection criterion.

To test whether the result is by chance, we used bootstrapping for its justification based on the BIC criterion, where the data are generated with the parameter of the best GM fit.

1. Fit the finite mixture model of regression with GM clustering and EPD clustering to the data of response variable $\mathbf{y}$ and the second group of 176 selected motifs, which lead to the EM estimates $\hat{\boldsymbol{\Psi}}_{GM}$ and $\hat{\boldsymbol{\Psi}}_{EPD}$ respectively.
2. Calculate the observed BIC difference, denote this by $\text{DBIC}_o = \min \text{BIC}_{GM} - \min \text{BIC}_{EPD}$, here we have $\text{DBIC}_o = 2173.629 - 2079.561 = 94.068$. The null hypothesis, H_o: the GM is true.
3.Simulate a data set of size n from the GM distribution with estimated parameters $\hat{\boldsymbol{\Psi}}_{GM}$, we denote this sample data $y_1^*, \ldots, y_n^*$.
4. Fit a finite mixture model with GM and EPD clustering to the simulated data, and calculate the corresponding bootstrap minimum BIC respectively, then find the difference of BICs, $\min \text{BIC}_{GM}^* - \min \text{BIC}_{EPD}^*$, denote this value by DBIC^*.
5.Repeat step 3 and 4 100 times to generate the bootstrap sampling distribution of the difference between the minimum BICs, the results using a histogram are shown in Figure 6.3.

The histogram shows the results of $\min \text{BIC}_{GM}^* - \min \text{BIC}_{EPD}^*$, while the red vertical line shows where the observed BIC difference of group 2 is.
6.The bootstrap p-value is

$$P = \frac{1}{100} \sum_{i=1}^{100} I\{\text{DBIC}_o \leq \text{DBIC}^*\} = 0. \tag{6.20}$$

The second bootstrap suggests us that we reject hypothesis H_o that EPD is true, which means the second dataset we selected was also not generated from Gaussian mixture.

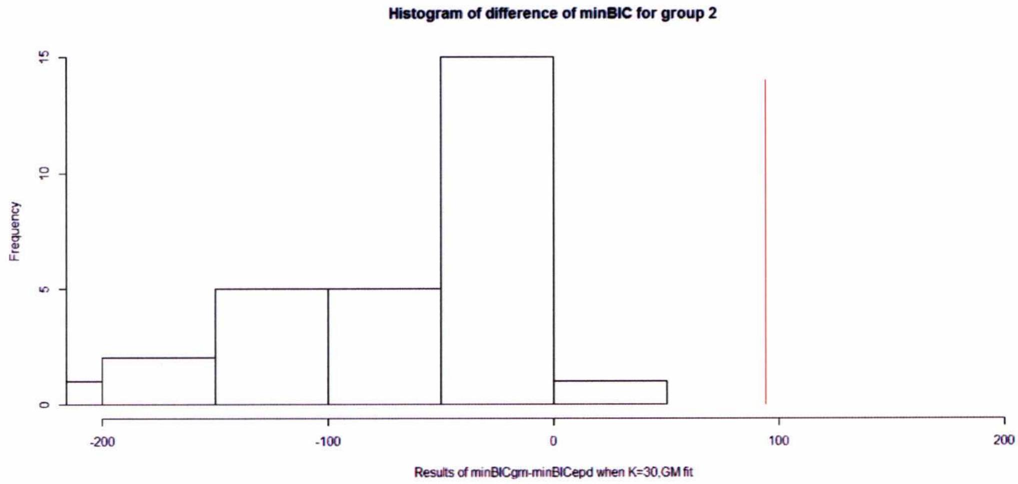

Figure 6.3: The bootstrap sampling distribution of minBICgm-minBICepd with GM fit derived over 100 times, where the red line is the observed BIC difference for group 2.

Secondly we also apply EPD and GM on the second group of 176 selected motifs with $K = 1, 2, 3, 4$, based on BIC selection criterion, $\min \text{BICgm} = 2173.629$ when $K = 3$ and $\min \text{BICepd} = 2079.561$ when $K = 2$. But this time, we use the best EPD fit to generalise the observations to test if the real data come from EPD mixture of regression.

1. Fit the finite mixture model of regression with GM clustering and EPD clustering to the data of the response variable $\mathbf{y}$ and the second group of 176 selected motifs, which leads to the EM estimates $\hat{\mathbf{\Psi}}_{GM}$ and $\hat{\mathbf{\Psi}}_{EPD}$ respectively.
2. Calculate the observed BIC difference, denote this by $\text{DBIC}_o = \min \text{BIC}_{GM} - \min \text{BIC}_{EPD}$, here we have $\text{DBIC}_o = 2173.629 - 2079.561 = 94.068$. The null hypothesis, H_o: EPD is true.
3.Simulate a data set of size n from EPD distribution with estimated parameters $\hat{\mathbf{\Psi}}_{EPD}$, we denote this sample data $y_1^*, \ldots, y_n^*$.
4. Fit a finite mixture model with GM and EPD clustering to the simulated data, and calculate the corresponding bootstrap minimum BIC respectively,

then find the difference of BICs, $\min \mathrm{BIC}^*_{GM} - \min \mathrm{BIC}^*_{EPD}$, denote this value by DBIC*.

5.Repeat steps 3 and 4 100 times to generate the bootstrap sampling distribution of the difference between the minimum BICs, the results are shown using histogram in Figure 6.4.

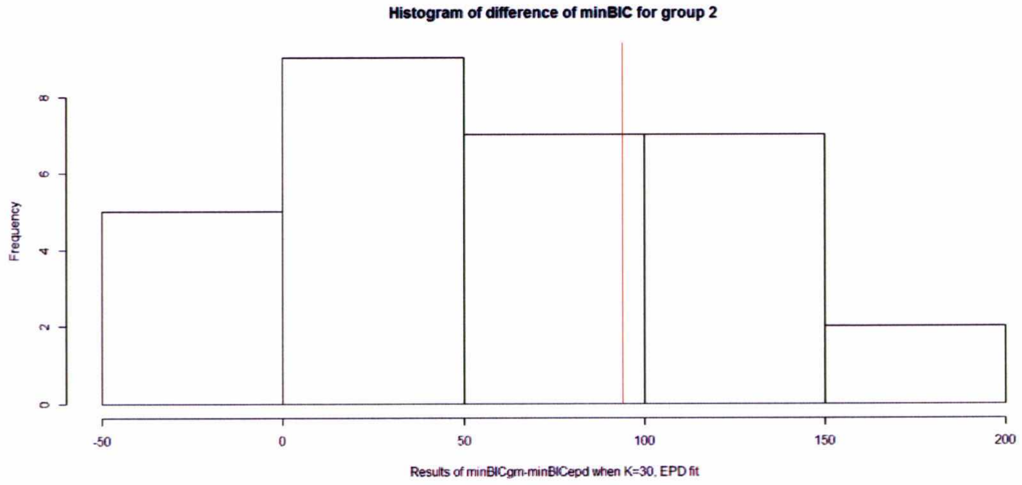

Figure 6.4: The bootstrap sampling distribution of minBICgm-minBICepd with EPD fit derived over 100 times, where the red line is the observed BIC difference for group 2.

The histogram shows the results of $\min \mathrm{BIC}^*_{GM} - \min \mathrm{BIC}^*_{EPD}$, and the red vertical line shows where the observed BIC difference of group 2 is.

6.The bootstrap p-value is

$$P = \frac{1}{100}\sum_{i=1}^{100} I\{\mathrm{DBIC}_o \leq \mathrm{DBIC}^*\} = 0.3. \tag{6.21}$$

The third bootstrap suggests that the hypothesis H_o: EPD is true and can be accepted, which means the second dataset we selected is generalised from the EPD mixture of regression.

From the above two bootstrap test we can tell that the real data was not Gaussian distributed and therefore it is not wise to use Gaussian mixture to fit the data. The EPD model provides a better fit with varying shape parameters rather than fixing it equals to 2 as in the Gaussian mixture fit.

Chapter 7

Classification

Classification is an important method for statistical data analysis especially for multivariate data. It is a technique to separate distinct sets of training data whose category membership is known and to identify a new observation into one of the previously defined groups. A one-group classification problem was applied by (Jackson, 1956) to a bivariate graphical implementation of multivariate quality control procedure suggusted by (Hotelling, 1947) and also the classification problem on the two-group case and multi-group case later on. This classification problem has received lots of attention in many fields such as finance, medication, biometrics etc. e.g. (Gordon, 1981; Gnanadesikan, 1997; Anderson, 2003; Johnson and Wichern, 2007). The classification indicate us how some observations belong to a particular group. It is very useful in many situations, for example when there is incomplete knowledge about future performance or the "perfect" information can only be obtained by destroying the objective, not even mention sometimes the information is very expensive or even unavailable to obtain.

In this chapter we apply the exponential power distribution on the classification instead of Gaussian distribution, and we focus classification on the two group situation. Hence as we have the non-Gaussian training data have two groups and the target is to find a rule that can be used to optimally assign new observations into one of the two groups.

7.1 Classification for two populations

We label the two groups g_1 and g_2 for convenience. Suppose we have regression data $R = (y_1, x_1; \ldots; y_n, x_n)$ which comes from two groups, the two populations can be described with the probability density function $f_1(y|x)$ and $f_2(y|x)$ for groups 1 and 2 respectively. For randomly selected regression training data, the first dataset is $(y_{11}, x_{11}; \ldots; y_{1n_1}, x_{1n_1})$ which come from g_1 and the second dataset is $(y_{21}, x_{21}; \ldots; y_{2n_2}, x_{2n_2})$ comes from g_2, we fit a regression distribution to each dataset and obtain two probability density functions $\hat{f}_1(y|x)$ and $\hat{f}_2(y|x)$. Now given a new observation (y, x), it is allocated in g_1 if it falls in the first group, and it is allocated in g_2 if it falls in the second group.

The classification rules can also lead to existence of errors such as assigning an observation in g_1 when its actually comes from g_2 or assigning an observation in g_2 when it is actually from g_1, which is because the distinction between the two populations is not very clear.

Let p_1 be the prior probability of g_1 and p_2 be the prior probability of g_2, where $p_1 + p_2 = 1$. We denote $c(1|2)$ to be the cost of misclassification for the case when an observation is allocated in g_1 when its actually comes from g_2, and $c(2|1)$ to be the cost of misclassification for the case when allocated an observation is allocated in g_2 when it's actually comes from g_1.

Now if we have a new observation (y, x), we assign it to group 1 if

$$\frac{\hat{f}_1(y|x)}{\hat{f}_2(y|x)} \geq (\frac{c(1|2)}{c(2|1)})(\frac{p_2}{p_1}), \tag{7.1}$$

we assign the new observation to group 2 if

$$\frac{\hat{f}_1(y|x)}{\hat{f}_2(y|x)} < (\frac{c(1|2)}{c(2|1)})(\frac{p_2}{p_1}), \tag{7.2}$$

by using decision theory. Where $\frac{\hat{f}_1(y|x)}{\hat{f}_2(y|x)}$ is the density ratio, $\frac{c(1|2)}{c(2|1)}$ is the cost ratio and $\frac{p_2}{p_1}$ is the prior probability ratio.

There are some special cases in the classification:

1. When the two prior probabilities are equal, i.e $p_2/p_1 = 1$.
The prior probabilities are often taken to be equal when they are unknown; hence the ratio of probability density is only compared with the ratio of the misclassification cost.
Hence, we allocate the new observation to g_1 if

$$\frac{\hat{f}_1(y|x)}{\hat{f}_2(y|x)} \geq \frac{c(1|2)}{c(2|1)},$$

we allocate the new observation to group 2 if

$$\frac{\hat{f}_1(y|x)}{\hat{f}_2(y|x)} < \frac{c(1|2)}{c(2|1)}.$$

2. When the two misclassification costs are equal, i.e. $\frac{c(1|2)}{c(2|1)} = 1$.
The misclassification are often taken to be unity when they are indeterminate, hence the ratio of the probability density is only compared to the ratio of the prior.
In this case, we allocate the new observation to g_1 if

$$\frac{\hat{f}_1(y|x)}{\hat{f}_2(y|x)} \geq \frac{p_2}{p_1},$$

we assign the new observation to group 2 if

$$\frac{\hat{f}_1(y|x)}{\hat{f}_2(y|x)} < \frac{p_2}{p_1}.$$

3. When both the two prior probabilities and the two misclassification costs

are equal, i.e $\frac{p_2}{p_1} = 1$ and $\frac{c(1|2)}{c(2|1)} = 1$. In this case, the classification is simply by comparing the value of two density functions. We use this case as an assumption in simulation and real data analysis.
In this case, we allocate the new observation to g_1 if

$$\frac{\hat{f}_1(y|x)}{\hat{f}_2(y|x)} \geq 1,$$

and we assign the new observation to group 2 if

$$\frac{\hat{f}_1(y|x)}{\hat{f}_2(y|x)} < 1.$$

For example: Assume we have a new observation (y, x) which lets $\hat{f}_1(y|x) = 0.5$ and $\hat{f}_2(y|x) = 0.2$, we find $\frac{c(1|2)}{c(2|1)} = \frac{1}{4}$ and $\frac{p_2}{p_1} = \frac{0.4}{0.6}$. Hence we have

$$\frac{\hat{f}_1(y|x)}{\hat{f}_2(y|x)} = 2.5,$$

$$(\frac{c(1|2)}{c(2|1)})(\frac{p_2}{p_1}) = (\frac{3}{4})(\frac{0.4}{0.6}) = 0.5,$$

Therefore we assign the observation (y, x) in g_1 as

$$\frac{\hat{f}_1(y|x)}{\hat{f}_2(y|x)} \geq (\frac{c(1|2)}{c(2|1)})(\frac{p_2}{p_1}).$$

7.2 Classification with two exponential power populations

Classification is normally based on the Gaussian distribution due to its simplicity and high efficiency. However, as we mentioned in the previous chapters, most real data are not perfectly Gaussian distributed; hence, there is a bias when using the Gaussian classification. Here we use a more general classification distribution, the exponential power distribution classification, instead of the widely used Gaussian classification.

Suppose we have a training data where the dataset 1 is $(y_{11}, x_{11}; \ldots; y_{1n_1}, x_{1n_1})$ which comes from group 1 and dataset 2 $(y_{21}, x_{21}; \ldots; y_{2n_2}, x_{2n_2})$ comes from group 2, we fit an exponential power regression distribution to each dataset. We obtain two exponential power density functions, say $\hat{f}_1(y|x)$ and $\hat{f}_2(y|x)$, where

$$\begin{aligned}
\hat{f}_1(y|x) &= \frac{\alpha_1}{2\sigma_1\Gamma\left(1/\alpha_1\right)} \exp\left(-\frac{\left|y_j - x_j^T\boldsymbol{\beta}_1\right|^{\alpha_1}}{(\sigma_1^2)^{\alpha_1/2}}\right) \\
\hat{f}_2(y|x) &= \frac{\alpha_2}{2\sigma_2\Gamma\left(1/\alpha_2\right)} \exp\left(-\frac{\left|y_j - x_j^T\boldsymbol{\beta}_2\right|^{\alpha_2}}{(\sigma_2^2)^{\alpha_2/2}}\right).
\end{aligned}$$

From Equation 7.1 and 7.2, we assign the new observation (y, x) to group 1 if

$$\frac{\frac{\alpha_1}{2\sigma_1\Gamma(1/\alpha_1)} \exp\left(-\frac{\left|y_j - x_j^T\boldsymbol{\beta}_1\right|^{\alpha_1}}{(\sigma_1^2)^{\alpha_1/2}}\right)}{\frac{\alpha_2}{2\sigma_2\Gamma(1/\alpha_2)} \exp\left(-\frac{\left|y_j - x_j^T\boldsymbol{\beta}_2\right|^{\alpha_2}}{(\sigma_2^2)^{\alpha_2/2}}\right)} \geq (\frac{c(1|2)}{c(2|1)})(\frac{p_2}{p_1}), \tag{7.3}$$

we assign the new observation to group 2 if

$$\frac{\frac{\alpha_1}{2\sigma_1\Gamma(1/\alpha_1)} \exp\left(-\frac{\left|y_j - x_j^T\boldsymbol{\beta}_1\right|^{\alpha_1}}{(\sigma_1^2)^{\alpha_1/2}}\right)}{\frac{\alpha_2}{2\sigma_2\Gamma(1/\alpha_2)} \exp\left(-\frac{\left|y_j - x_j^T\boldsymbol{\beta}_2\right|^{\alpha_2}}{(\sigma_2^2)^{\alpha_2/2}}\right)} < (\frac{c(1|2)}{c(2|1)})(\frac{p_2}{p_1}). \tag{7.4}$$

We can rewrite the Equation 7.3 and 7.4 into the following form respectively:

$$\frac{\frac{\alpha_1}{2\sigma_1\Gamma(1/\alpha_1)}}{\frac{\alpha_2}{2\sigma_2\Gamma(1/\alpha_2)}} \exp\left(-\frac{\left|y_j - x_j^T\boldsymbol{\beta}_1\right|^{\alpha_1}}{(\sigma_1^2)^{\alpha_1/2}} + \frac{\left|y_j - x_j^T\boldsymbol{\beta}_2\right|^{\alpha_2}}{(\sigma_2^2)^{\alpha_2/2}}\right) \geq (\frac{c(1|2)}{c(2|1)})(\frac{p_2}{p_1}),$$

$$\frac{\frac{\alpha_1}{2\sigma_1\Gamma(1/\alpha_1)}}{\frac{\alpha_2}{2\sigma_2\Gamma(1/\alpha_2)}} \exp\left(-\frac{\left|y_j - x_j^T\boldsymbol{\beta}_1\right|^{\alpha_1}}{(\sigma_1^2)^{\alpha_1/2}} + \frac{\left|y_j - x_j^T\boldsymbol{\beta}_2\right|^{\alpha_2}}{(\sigma_2^2)^{\alpha_2/2}}\right) < (\frac{c(1|2)}{c(2|1)})(\frac{p_2}{p_1}).$$

7.3 Simulation

In this simulation, we randomly generate (y, x) from groups 1 and 2, where we know the parameters and where they come from. Then we apply the above procedure for the exponential power distribution classification and Gaussian classification for 100 times to see how many times(percentage) we assign a data point to a wrong group and how many times(percentage) we assign a data point to a correct group. We denote α to be the shape parameter, β is the regression coefficient, and π is the proportion weight.

7.3.1 Simulation 1 (Two-component simple EPD regression model)

Suppose that we have a training dataset with sample size $n = 300$: dataset 1 $(y_1, x_1; \ldots; y_{n1}, x_{n1})$,from group 1 and dataset 2 $(y_{n1+1}, x_{n1+1}; \ldots; y_n, x_n)$, from group 2. We have $(x_1, \ldots, x_{n1}) \sim N(2, \Sigma_1)$ and $(x_{n1+1}, \ldots, x_n) \sim N(0, \Sigma_2)$, where the correlation structures of covariates is $\Sigma_1 = corr(x_l, x_m) = (\rho^{|l-m|})_{1\le l,m\le n1}$ and $\Sigma_2 = corr(x_l, x_m) = (\rho^{|l-m|})_{1\le l,m\le n2}$. For the response vector we have: $y_i \sim EPD(x_i^T\beta_1, \sigma_1, \alpha_1)$ for $i = 1, \ldots, n1$, and $y_i \sim EPD(x_i^T\beta_2, \sigma_2, \alpha_2)$ for $i = (n1+1), \ldots, n$. Let the shape parameter is $\alpha = (2.7, 1.3)$ and $\pi = (0.5, 0.5)$.

The distinction between regression coefficients are not clear

In this simulation we compare the performance of classification by using Gaussian classification and EPD classification in which case that the distinction between the regression coefficients are not clear. We let $\beta = (-0.5, 0.5)$ and the value of the rest parameters we chose are shown in Table 7.1.

We generate a new observation (y, x), then apply the above procedure of exponential power distribution classification and Gaussian classification for 100 times. Let "F" be the percentage we assign a wrong group and "T" be the percentage we assign correctly. The results are shown in Table 7.1.

Table 7.1: Compare the performance of classification by using Gaussian classification and EPD classification in the case that the distinction between the regression coefficients are not clear

Equal variance cases

Cases		T		F	
ρ	σ	EPD	GM	EPD	GM
0.50	1	0.69	0.65	0.31	0.35
	2	0.67	0.63	0.32	0.37
0.75	1	0.71	0.68	0.29	0.32
	2	0.73	0.69	0.27	0.31

Unequal variance cases

Cases			T		F	
ρ	σ_1	σ_2	EPD	GM	EPD	GM
0.5	1.5	2.5	0.71	0.70	0.29	0.30
	1.5	3.5	0.74	0.72	0.26	0.28
0.75	1.5	2.5	0.74	0.72	0.26	0.28
	1.5	3.5	0.70	0.69	0.30	0.31

From the Table 7.1, it can be seen that the the percentage we assign a data into a correct group by using EPD classification is higher than using GM classification under all conditions. Hence, the results shows that the EPD classification has better performance than GM classification.

The distinction between regression coefficients are very clear

In this simulation we compare the performance of classification by using Gaussian classification and EPD classification with the case that the distinction between the regression coefficients are very clear. We let $\beta = (-0.5, 2.5)$ and the value of the rest parameters we chose are shown in Table 7.2.

We generate a new observation (y, x), then apply the above procedure of exponential power distribution classification and Gaussian classification for 100 times. Let "F" be the percentage we assign a wrong group and "T" be the percentage we assign correctly. The results are shown in Table 7.2.

Table 7.2: Compare the performance of classification by using Gaussian based classification and EPD based classification in the case that the distinction between the regression coefficients are very clear

Equal variance cases

Cases		T		F	
ρ	σ	EPD	GM	EPD	GM
0.50	1	0.87	0.86	0.13	0.12
	2	0.80	0.80	0.20	0.20
0.75	1	0.89	0.89	0.11	0.11
	2	0.83	0.82	0.17	0.18

Unequal variance cases

Cases			T		F	
ρ	σ_1	σ_2	EPD	GM	EPD	GM
0.5	1.5	2.5	0.84	0.83	0.16	0.17
	1.5	3.5	0.86	0.85	0.14	0.15
0.75	1.5	2.5	0.81	0.80	0.19	0.20
	1.5	3.5	0.82	0.81	0.18	0.19

From the Table 7.2, it can be seen that the the percentage we assign a data into a correct group by using EPD classification is higher than or equal to using GM classification under all scenarios we considered. Although the advantage of using EPD classification is less obvious, compared with the case when the distinction between two regression coefficients are not clear, the results still suggest that the EPD classification performs better or equal to GM classification.

7.3.2 Simulation 2 (Two-component sparse EPD regression model)

Dimension smaller than sample size

In this simulation, we test the performance of classification by using Gaussian classification and EPD classification in a more complexed model. The training data still come from 2 groups and all the setting is similar as in the last simulation, unless we let the dimension is 100 here instead the dimension is 1 in the last simulation. Suppose that we have a training dataset with sample size $n = 300$: dataset 1 $(y_1, x_1; \dots; y_{n1}, x_{n1})$,from group 1 and dataset 2 $(y_{n1+1}, x_{n1+1}; \dots; y_n, x_n)$, from group 2. For sparse regressions models, we let the number of non-zero value in both regression coefficients vector $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are 4. We denote non-zero coefficients by

$$\boldsymbol{\beta}_{1*} = \{-3.5, -0.5, -3.5, -0.5\}, \boldsymbol{\beta}_{2*} = \{0.5, 3.5, 0.5, -3.5\},$$

and the value of the rest parameters we chose are shown in Table 7.3.

We generate a new observation (y, x), then apply the above procedure of exponential power distribution classification and Gaussian classification for 300 times. Let "F" be the percentage we assign a wrong group and "T" be the percentage we assign correctly. The results are shown in Table 7.3.

From the Table 7.3,it can be seen that the percentage we assign a data into a correct group by using EPD classification is higher than by using GM

Table 7.3: Compare the performance of classification by using Gaussian based classification and EPD based classification for sparse regression model

Equal variance cases					
Cases		T		F	
ρ	σ	EPD	GM	EPD	GM
0.50	1	0.95	0.94	0.05	0.04
	2	0.9	0.93	0.05	0.07
0.75	1	0.96	0.95	0.04	0.05
	2	0.94	0.93	0.06	0.07

Unequal variance cases						
Cases			T		F	
ρ	σ_1	σ_2	EPD	GM	EPD	GM
0.5	1.5	2.5	0.95	0.93	0.05	0.07
	1.5	3.5	0.94	0.93	0.06	0.07
0.75	1.5	2.5	0.94	0.93	0.06	0.07
	1.5	3.5	0.94	0.93	0.06	0.07

classification under all conditions in sparse regression model. But the advantage of using EPD classification is becoming weaker as the dimension is becoming larger. After all, the results still shows that the EPD classification has better or equal performance compare to GM classification in sparse regression model.

Dimension larger than sample size

Similar as in the last simulation, we test the performance of classification by using Gaussian classification and EPD classification in sparse EPD regression model. But in this simulation we let the dimension equals to 400 which is larger than the sample size $n = 300$. The results shown in Table 7.4.

It can be seen that in Table 7.4, the performance of EPD classification is still better than GM classification but the advantage is quiet small as the dimension is very large.

Table 7.4: Compare the performance of classification by using Gaussian classification and EPD classification for sparse regression model with dimension larger than sample size.

Equal variance cases

Cases		T		F	
ρ	σ	EPD	GM	EPD	GM
0.50	1	0.952	0.950	0.048	0.050
	2	0.928	0.923	0.072	0.077
0.75	1	0.959	0.956	0.041	0.044
	2	0.937	0.932	0.063	0.068

Unequal variance cases

Cases			T		F	
ρ	σ_1	σ_2	EPD	GM	EPD	GM
0.5	1.5	2.5	0.940	0.937	0.060	0.063
	1.5	3.5	0.940	0.938	0.060	0.062
0.75	1.5	2.5	0.940	0.935	0.060	0.065
	1.5	3.5	0.941	0.938	0.059	0.062

Chapter 8

Conclusion and Future Work

8.1 Overview

The mixture of regression model is an important technique used in statistical modelling to investigate the relationship between variables. It is applied in many fields such as genetics, finance and biology. Here we focus on its application to genetic data. We considered two real genetic data in this study, the first one is the yeast stress dataset of (Gasch et al., 2000) which explores genome-wide expression patterns in the yeast Saccharomyces cerevisiae in response to diverse environmental changes. This dataset containing the expression levels of 496 selected yeast genes under 173 experimental conditions, where the columns stand for genes and the rows stand for experimental conditions. We want to investigate the correlations between the genes. For the second data we are interested in exploring the relative expression of a gene by the binding strengths of a subset of the P candidate motifs to the regulatory region of the gene. This dataset containing the expression levels of 4443 genes and 2155 of motif-matching scores of candidate motifs from Saccharomyces cerevisiae (Conlon et al., 2003). We intend to find the motifs related to each gene. As we know gene expression data normally contains unknown correlation structures even after normalization, hence it raises a great challenge for the existing clustering methods such as the Gaussian mixture model and k-mean. Motivated by the work of (Zhang and Liang, 2010), we have introduced the mixture of exponential power distribution models for robustly clustering of gene expression data. One of the reason of using EPD instead of Gaussian distribution is that the EPD is more flexible than Gaussian distribution. The second reason is that the assumptions in the Gaussian mixture model may be invalid in some applications.

We have introduced and developed our method based on two different aspects of multiple regression with random errors distributed according to the EPD. The first aspect is estimation: we use both the EM and Newton-Raphson methods to estimate the parameters of the mixture of EPD regression model. The model selection criterion such as AIC, BIC and EBIC were derived for both EPD and GM models. We have examined different simulations approaches for the performance of the EPD mixture model and GM. It has suggested that the GM performs better than EPD if the data are generated from GM model, but in the cases when the data come from the EPD mixture model or data points are clumpy correlated, the EPD mixture model has shown a better performance in terms of AIC, BIC, EBIC, and RAND index and produced more accurate estimation of parameters than GM model. We have also fitted both models to yeast stress dataset, it indicated that the model of mixture EPD can give a better clustering result than GM model.

The second aspect is variable selection: with the development of the technology, scientists allow to collect data with large amount of information. For example, the gene expression data can contain very high dimension with limited sample size. Hence, the idea of reducing the dimensionality of the problem by only select the related variables of the response variable has been introduced. The novelty of this research is that we have converted each penalised regression estimation problem to a LASSO problem. Here we have applied EPD with LASSO method to identify the important variables in the large dataset especially for high dimensional data. The simulations in Chapter 6 have illustrated the performance of our method in two scenarios that with sample size $n = 300$ and $n = 400$ and with relatively large dimension $P = 10$ and $P = 100$. In both scenarios, we have obtained higher RAND indexes when we used EPD than when used GM. We have also applied EPD in forward selection to select the most relative variables. We have conducted simulations for both independent and dependent data with dimensions larger than their sample sizes. The best combination of the first 10 iterations in the result has contained the majority of relative variables. We have combined the forward selection with LASSO method to deal with the situations that the dimension is larger than its sample size. The procedure has worked well in our simulation study. For the real data analysis, we have used Saccharomyces cerevisiae micro-array experiments data. Although in this dataset, the number of dimension P is not larger than the size of genes n, but it is still a very high dimension. Thus the task of variable selection remains difficult, this is because there is the potential accumulation of noise and the interpret ability to cluster the genes in the full feature space, hence we use forward selection method as a pre-screen before apply the variable selection later on. Then we use bootstrap method to exam the reliability of our results, it seems the mixture of EPD models still do better job than GM models.

Finally, we apply exponential power distribution on classification instead of Gaussian distribution, and we focus classification on the two group situation. Hence our goal is to sort the non-Gaussian training data into two groups and the emphasis is to find a rule that can be used to optimally assign new observations into one of the two groups. Similar as before, we also did few simulations to illustrate the performance of our method. The EPD classification shows a better ability to distinct the two groups, but the advantage is not very obvious compare to Gaussian classification.

8.2 Future work

On future work on the subject, further exploration on the ultra high dimensional data is a natural path we shall follow. Working in higher dimensions is remarkably challenging, as we shown in real data analysis, it is difficult to apply our methodology in the entire set of genes. Because for large dataset with high dimensions, we have to face problems such as a large number of iterations and estimation of many hyperparameters.

How to choose the initial values could also be an interesting point to explore. Because the bad initial values may lead our results to a local maximum value rather than the global maximum, hence, run a simulation or real data analysis start with reasonable initial values could give us more reliable results.

Another extension to our models presented in this thesis could be use Bayesian methodology for mixture of EPD model of regression. Bayesian statistics gained great attention over the last decades, we could investigate the performance to the combination task of estimation and variable selection by using the mixture EPD model within the Bayesian framework.

Finally, we can apply other penalties on our variable selection method such as adaptive LASSO, SCAD and Bridge. These have different strength to analysis different dataset.

Appendix A

Supporting Calculations for Chapter 6

To get more calculation details, we can explore equation(6.2) part by part.

Let

$$\xi_i = \tau_{ti}(y_i|\mathbf{x}_i, \mathbf{\Psi}^{(s)}) \left|y_i - x_i^T \boldsymbol{\beta}_t\right|^{(\alpha_t - 2)}. \tag{A.1}$$

The first part of the right hand side of equation (6.2) can be written as:

$$\begin{aligned}
\mathbf{X}^T\mathbf{W}_t\mathbf{y} &= (\mathbf{x}_1, \cdots, \mathbf{x}_n) \left[\frac{\alpha_t}{(\sigma_t^2)^{\alpha_t/2}} \begin{pmatrix} \xi_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \xi_n \end{pmatrix} \right] \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \\
&= \mathbf{x}_1 \frac{\alpha_t}{(\sigma_t^2)^{\alpha_t/2}} (\xi_1, \cdots, \mathbf{x}_n \xi_n) \begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{pmatrix} \\
&= \mathbf{x}_1 \frac{\alpha_t}{(\sigma_t^2)^{\alpha_t/2}} (\xi_1 y_1 + \cdots + \mathbf{x}_n \xi_n y_n) \\
&= \sum_{i=1}^{n} \frac{\alpha_t}{(\sigma_t^2)^{\alpha_t/2}} \xi_i (\mathbf{x}_i \mathbf{x}_i)
\end{aligned}$$

Now we explore $W_t^{(s)}$ to matrix form,

$$W_t^{(s)} = \frac{\alpha_t}{\sigma_t^{\alpha_t}} \begin{pmatrix} (\xi_1 - Co)_+ + Co & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & (\xi_n - C_0) + Co \end{pmatrix}$$

where Co is a small positive constant, and

$$x_+ = \begin{cases} x, \ x > 0 \\ 0, \ x \leq 0 \end{cases}.$$

Note that

$$V_t^{(s)} = \frac{\pi_t^{(s)}}{\sigma_t^{(s)}} \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & \frac{P'_\lambda\left(\frac{\left|\beta_{t1}^{(s)}\right|}{\sigma_t^{(s)}}\right)}{\lambda} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{P'_\lambda\left(\frac{\left|\beta_{tP}^{(s)}\right|}{\sigma_t^{(s)}}\right)}{\lambda} \end{pmatrix},$$

the inverse matrix of V_t^s is

$$\left(V_t^{(s)}\right)^{-1} = \frac{\pi_t^{(s)}}{\sigma_t^{(s)}} \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & \frac{\lambda}{P'_\lambda\left(\frac{\left|\beta_{t1}^{(s)}\right|}{\sigma_t^{(s)}}\right)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{\lambda}{P'_\lambda\left(\frac{\left|\beta_{tP}^{(s)}\right|}{\sigma_t^{(s)}}\right)} \end{pmatrix}.$$

Consider $X^T W_t Y - (X^T W_t X)\boldsymbol{\beta}_t - n\lambda V_t sgn(\boldsymbol{\beta}_t) = 0$, for $1 \leq t \leq k$, where $P'_\lambda\left(\frac{\left|\beta_{tj}^{(s)}\right|}{\sigma_t^{(s)}}\right) > 0$, for $1 \leq j \leq P$.

We have

$$\begin{aligned} & \left(W_t^{(V)1/2} X V_t^{(V)(-1)}\right)^T \left(W_t^{(V)1/2} y\right) \\ & - \left(W_t^{(V)1/2} X V_t^{(V)(-1)}\right)^T \left(W_t^{(V)1/2} X V_t^{(V)(-1)}\right) (V_t^{(V)} + e_1)\boldsymbol{\beta}_t \\ & - n\lambda \mathrm{sgn}((V_t^{(V)} + e_1)\boldsymbol{\beta}_t) \\ = \ & 0, \end{aligned}$$

for $1 \leq t \leq K$, where

$$e_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 \end{pmatrix}_{(P+1)\times(P+1)}, \text{ and}$$

$$V_t^{(s)} + e_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \left(\frac{\pi_t^{(s)}}{\sigma_t^{(V)}}\right) \frac{P'_\lambda\left(\frac{\left|\beta_{t1}^{(s)}\right|}{\sigma_t^{(s)}}\right)}{\lambda} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \left(\frac{\pi_t^{(s)}}{\sigma_t^{(V)}}\right) \frac{P'_\lambda\left(\frac{\left|\beta_{tP}^{(s)}\right|}{\sigma_t^{(s)}}\right)}{\lambda} \end{pmatrix}.$$

Hence, the first part of $\hat{\boldsymbol{\beta}}$ can be written as:

$$\begin{aligned}
&\left(W_t^{(V)1/2} X V_t^{(V)(-1)}\right)^T \left(W_t^{(V)1/2} y\right) \\
&= \left\{ \left(\frac{\alpha_t}{(\sigma_t^2)^{\alpha_t/2}}\right)^{\frac{1}{2}} \begin{pmatrix} \xi_1^{\frac{1}{2}} & \xi_1^{\frac{1}{2}} x_{11} & \cdots & \xi_1^{\frac{1}{2}} x_{P1} \\ \xi_2^{\frac{1}{2}} & \xi_2^{\frac{1}{2}} x_{12} & \cdots & \xi_2^{\frac{1}{2}} x_{P2} \\ \vdots & & \ddots & \vdots \\ \xi_n^{\frac{1}{2}} & \xi_n^{\frac{1}{2}} x_{1n} & \cdots & \xi_n^{\frac{1}{2}} x_{Pn} \end{pmatrix} V_t^{(V)(-1)} \right\} \left(W_t^{(V)1/2} y\right) \\
&= \left(\frac{\alpha_t}{(\sigma_t^2)^{\alpha_t/2}}\right)^{\frac{1}{2}} \left(\frac{\alpha_t}{\pi_t}\right) \begin{pmatrix} 0 & \xi_1^{\frac{1}{2}} x_{11} \frac{\lambda}{P'_\lambda\left(\frac{\left|\beta_{t1}^{(s)}\right|}{\sigma_t^{(s)}}\right)} & \cdots & \xi_1^{\frac{1}{2}} x_{P1} \frac{\lambda}{P'_\lambda\left(\frac{\left|\beta_{tP}^{(s)}\right|}{\sigma_t^{(s)}}\right)} \\ 0 & \xi_2^{\frac{1}{2}} x_{12} \frac{\lambda}{P'_\lambda\left(\frac{\left|\beta_{t1}^{(s)}\right|}{\sigma_t^{(s)}}\right)} & \cdots & \xi_2^{\frac{1}{2}} x_{P2} \frac{\lambda}{P'_\lambda\left(\frac{\left|\beta_{tP}^{(s)}\right|}{\sigma_t^{(s)}}\right)} \\ \vdots & \vdots & \ddots & \vdots \\ 0_n & \xi_n^{\frac{1}{2}} x_{1n} \frac{\lambda}{P'_\lambda\left(\frac{\left|\beta_{t1}^{(s)}\right|}{\sigma_t^{(s)}}\right)} & \cdots & \xi_n^{\frac{1}{2}} x_{Pn} \frac{\lambda}{P'_\lambda\left(\frac{\left|\beta_{tP}^{(s)}\right|}{\sigma_t^{(s)}}\right)} \end{pmatrix}^T
\end{aligned}$$

$$\times \left(\frac{\alpha_t}{(\sigma_t^2)^{\alpha_t/2}}\right)^{\frac{1}{2}} \begin{pmatrix} \xi_1^{\frac{1}{2}} y_1 \\ \xi_2^{\frac{1}{2}} y_2 \\ \vdots \\ \xi_n^{\frac{1}{2}} y_n \end{pmatrix}_{n\times 1}$$

$$= \left(\frac{\alpha_t}{(\sigma_t^2)^{\alpha_t/2}}\right)^{\frac{1}{2}} \left(\frac{\alpha_t}{\pi_t}\right) \begin{pmatrix} 0 \\ \sum_{i=1}^n \xi_i \frac{\lambda}{P'_\lambda\left(\frac{\left|\beta_{t1}^{(s)}\right|}{\sigma_t^{(s)}}\right)} x_{i1} y_i \\ \vdots \\ \sum_{i=1}^n \xi_i \frac{\lambda}{P'_\lambda\left(\frac{\left|\beta_{tP}^{(s)}\right|}{\sigma_t^{(s)}}\right)} x_{iP} y_i \end{pmatrix}_{(P+1)\times 1} .$$

Appendix B

Selected Genes and Motifs in Chapter 6

In the real data analysis of Chapter 6, we use LASSO method to select the relative motifs. For the best EPD fit, we select 81 motifs from the total of 176 motifs.

The name of the selected motifs are list in Table B.1, from left to right with increasing order of the absolute value of correlation coefficients:

Table B.1: List of names of 81 selected motifs.

Motif.N1.6.13 -1.98631485	Motif.N1.8.3 0.83470318	Motif.P1.6.1 -0.14711098	Motif.P1.7.2 -0.39753278	Motif.N1.11.5 -0.22162536
Motif.P1.10.3 -0.79153627	Motif.P1.11.3 -0.80268264	Motif.N1.11.4 -0.48371419	Motif.N1.11.10 0.08520930	Motif.P1.8.2 -0.32390556
Motif.N1.12.3 -1.12012978	Motif.N1.11.5 0.18443424	Motif.N1.10.3 -0.52923331	Motif.N1.11.2 -0.61390895	Motif.P1.9.8 -0.34747851
Motif.N1.5.1 -0.18794934	Motif.P1.5.1 -0.20180962	Motif.N1.7.4 -0.83448023	Motif.N1.5.3 0.08473147	Motif.N1.12.4 -0.05269894
Motif.P1.11.5 -1.09267018	Motif.N1.11.7 -0.89739114	Motif.N1.9.4 0.19402792	Motif.N1.11.3 -0.53896102	Motif.N1.6.1 -1.91988705
Motif.N1.6.2 0.04433817	Motif.N1.6.3 -0.22232244	Motif.N1.10.2 -0.42315136	Motif.N1.6.2 -0.62357463	Motif.N1.7.9 -0.14754249
Motif.P1.10.4 0.37432483	Motif.N1.6.1 -0.88560767	Motif.N1.12.2 -0.86427729	Motif.N1.10.1 -1.52032762	Motif.N1.11.11 -1.32834896
Motif.P1.9.3 -0.84260586	Motif.N1.8.2 0.43412053	Motif.N1.9.6 -0.77006987	Motif.P1.12.1 -0.83337200	Motif.P1.7.1 0.11925958
Motif.N1.8.4 -0.82212396	Motif.N1.8.5 -1.70739894	Motif.N1.9.2 0.13365569	Motif.N1.9.2 -0.49696780	Motif.N1.10.4 -0.41664891
Motif.N1.12.1 -1.08987876	Motif.P1.8.3 -0.75313856	Motif.N1.9.5 0.04788381	Motif.N1.6.1 0.68183372	Motif.N1.9.2 -0.78533679
Motif.N1.7.11 -1.34798440	Motif.P1.9.1 -0.60317017	Motif.P1.12.3 -0.77606171	Motif.N1.10.14 -0.41528582	Motif.N1.8.1 -0.99565205
Motif.P1.10.1 -0.59745921	Motif.N1.7.5 0.31183043	Motif.N1.7.1 -0.47952442	Motif.P1.9.3 -0.18431995	Motif.N1.9.5 -0.04403038
Motif.N1.6.2 0.67666290	Motif.N1.7.3 -0.76610623	Motif.N1.9.3 -0.56329600	Motif.N1.8.3 0.33784313	Motif.N1.11.7 -0.65125406
Motif.N1.7.4 0.92680194	Motif.P1.11.3 -0.25685018	Motif.N1.7.6 0.19222513	Motif.N1.12.3 -0.64954378	Motif.P1.10.2 -0.92841767
Motif.P1.11.1 -1.40458585	Motif.P1.11.15 0.20135993	Motif.N1.7.2 1.27840128	Motif.P1.6.1 -0.35814744	Motif.N1.11.1 -0.32648068
Motif.N1.10.1 -0.99747357	Motif.N1.11.4 -0.66613440	Motif.N1.9.1 0.18216754	Motif.P1.7.1 -0.13895433	Motif.N1.8.1 -0.99828680
Motif.N1.8.1 0.54208671				

The Table B.2 and B.3 list the name of the motifs which have been selected in the group 1 and group 2 respectively.

Table B.2: List of names of the motifs selected for group 1.

Motif.N1.8.2 -0.64133935	Motif.N1.6.3 -0.22232244	Motif.N1.12.2 -0.86427729	Motif.N1.12.3 -0.72112019	Motif.N1.6.1 -1.91988705
Motif.N1.8.2 0.43412053	Motif.P1.12.2 0.05462189	Motif.N1.11.1 0.13663764	Motif.N1.11.5 0.18443424	Motif.N1.7.6 -0.78364314
Motif.N1.9.5 0.04788381	Motif.N1.10.3 -0.90718696	Motif.N1.9.2 -1.76735603	Motif.P1.11.1 -1.40458585	Motif.N1.7.6 -1.42970594
Motif.N1.7.1 -0.95396217	Motif.N1.6.1 -0.51995838	Motif.P1.10.2 -0.69585839	Motif.N1.8.1 -0.99828680	Motif.P1.12.3 -0.77606171
Motif.P1.10.14 -0.72946548	Motif.P1.8.3 -0.75313856	Motif.P1.12.1 -0.83031052	Motif.N1.7.11 0.41372746	Motif.N1.9.2 0.13365569
Motif.N1.8.3 0.33784313	Motif.N1.9.4 -1.54584945	Motif.N1.12.4 -0.05269894	Motif.P1.9.3 -0.84260586	Motif.N1.8.1 -0.72545769
Motif.N1.9.5 -0.37099085	Motif.N1.7.11 -1.34798440	Motif.P1.11.15 0.20135993	Motif.P1.12.3 -0.55571714	Motif.P1.5.1 -0.28513338
Motif.P1.8.1 -0.99711952	Motif.N1.6.2 -0.62357463	Motif.N1.11.2 -0.61390895	Motif.P1.7.2 -0.43995422	Motif.P1.7.2 -0.39753278
Motif.P1.11.5 -1.09267018	Motif.N1.11.4 -0.48371419	Motif.N1.11.11 -1.32834896	Motif.N1.10.1 0.54661810	Motif.N1.9.1 0.18216754
Motif.N1.7.2 1.27840128	Motif.N1.11.3 -0.01006387	Motif.P1.7.1 0.15707986	Motif.P1.8.1 -0.46384682	Motif.P1.9.8 -0.34747851
Motif.N1.11.7 -0.89739114	Motif.P1.8.1 -0.40060299	Motif.P1.12.3 -0.33008944	Motif.N1.12.1 -1.08987876	Motif.N1.12.3 -1.12012978
Motif.P1.11.1 -0.75808060				

Table B.3: List of names of the motifs selected for group 2.

Motif.P1.10.3 -0.79153627	Motif.N1.7.1 -0.12321588	Motif.P1.12.3 -0.77606171	Motif.N1.9.2 -1.76735603	Motif.P1.9.3 -0.85890769
Motif.N1.7.8 -1.81570622	Motif.N1.9.3 0.07965436	Motif.N1.9.4 -1.54584945	Motif.N1.12.3 -0.72112019	Motif.N1.9.5 -0.04403038
Motif.N1.11.10 0.08520930	Motif.N1.11.4 -0.48371419	Motif.N1.12.1 0.62486316	Motif.P1.7.2 -0.43995422	Motif.P1.11.3 -0.25685018
Motif.P1.9.8 -0.34747851	Motif.N1.7.2 1.27840128	Motif.N1.6.2 0.04433817	Motif.N1.12.4 -0.05269894	Motif.N1.7.6 -0.78364314
Motif.N1.9.5 -0.37099085	Motif.P1.7.1 0.15707986	Motif.N1.10.2 0.47976356	Motif.P1.11.15 0.20135993	Motif.N1.8.1 -0.99565205
Motif.N1.9.1 0.59945174	Motif.N1.12.1 -1.08987876	Motif.P1.9.3 -0.18431995	Motif.P1.11.3 -0.80268264	Motif.N1.8.1 -0.99828680
Motif.N1.5.1 -2.02911479	Motif.P1.6.1 -0.14711098	Motif.P1.9.2 -0.24329791	Motif.P1.11.1 -1.40458585	Motif.N1.11.1 0.13663764
Motif.P1.8.3 -0.75313856	Motif.N1.11.2 -0.17186360	Motif.N1.11.1 -1.07842663	Motif.P1.8.1 -0.99711952	Motif.N1.9.3 -0.56329600
Motif.N1.11.1 -0.37942041				

After EPD mixture regression based clustering, there are 361 out of 4443 genes have been allocated in group 1, and the rest ones in group 2. The Table B.4 and B.5 list the names of 361 genes which have been allocated in group 1.

Table B.4: List of names of 361 genes that have been allocated in gourp 1.

YCL055W 2.95	YPL192C 2.89	YCL027W 2.47	YGL028C 2.45	YIL117C 2.41	YDL037C 2.38	YDR461W 2.36	YBL016W 2.31
YDR085C 2.17	YJL157C 2.12	YNL145W 2.06	YNL192W 2.04	YJL107C 1.95	YIL079C 1.90	YKL189W 1.89	YKR058W 1.87
YKL128C 1.83	YCR089W 1.79	YGL053W 1.79	YGL052W 1.72	YOR385W 1.67	YJL108C 1.61	YKL109W 1.60	YBR067C 1.57
YHR030C 1.46	YMR137C 1.45	YLR332W 1.40	YKR042W 1.25	YNL280C 1.24	YKL104C 1.22	YOR344C 1.20	YOR220W 1.19
YDL048C 1.19	YLR286C 1.18	YNL208W 1.17	YER130C 1.17	YLR120C 1.13	YBR183W 1.11	YLR194C 1.11	YBR238C 1.11
YHR143W 1.09	YGL055W 1.03	YLR345W 1.01	YHR005C 0.99	YDL023C 0.98	YHR011W 0.97	YAR009C 0.96	YOL002C 0.96
YGL193C 0.95	YLR280C 0.95	YER124C 0.92	YGL051W 0.90	YPL008W 0.87	YBR023C 0.87	YER158C 0.85	YBR223C 0.84
YBR225W 0.81	YNL053W 0.79	YNL141W 0.78	YML046W 0.75	YOR238W 0.74	YFL027C 0.73	YPL177C 0.69	YJL159W 0.69
YPL149W 0.67	YBR182C 0.67	YIL083C 0.66	YOR137C 0.65	YOR193W 0.64	YKL209C 0.64	YML130C 0.63	YHR084W 0.62
YBR153W 0.61	YDR309C 0.60	YNL106C 0.59	YHL039W 0.58	YOR347C 0.58	YOR012W 0.57	YNL107W 0.57	YLR119W 0.56
YCR032W 0.55	YBR034C 0.55	YCL018W 0.54	YNR053C 0.54	YEL058W 0.53	YLR040C 0.53	YGR234W 0.53	YIL010W 0.52
YLR433C 0.52	YGR149W 0.52	YIL121W 0.52	YBR214W 0.51	YMR103C 0.51	YDL063C 0.51	YOR212W 0.51	YGR143W 0.51
YLR417W 0.51	YLR414C 0.51	YDR223W 0.51	YOR273C 0.50	YHR142W 0.50	YER179W 0.50	YCR007C 0.50	YHR209W 0.50
YOR229W 0.49	YMR291W 0.49	YMR164C 0.49	YDR383C 0.49	YPL183C 0.49	YGR122W 0.48	YOR177C 0.48	YEL021W 0.47
YGL074C 0.47	YBR295W 0.47	YHR195W 0.47	YJR054W 0.47	YOR062C 0.46	YLR281C 0.46	YKR013W 0.46	YOR208W 0.45
YIL060W 0.45	YPR172W 0.45	YJL219W 0.45	YOR095C 0.45	YGL029W 0.45	YLR251W 0.45	YER184C 0.45	YOR036W 0.44
YPR143W 0.44	YLR217W 0.44	YNL210W 0.44	YOL010W 0.44	YEL059W 0.44	YLR250W 0.44	YDL072C 0.44	YOL101C 0.43
YKL152C 0.43	YCR100C 0.43	YIL104C 0.43	YDL129W 0.43	YIL014W 0.42	YGR120C 0.42	YGL144C 0.42	YLR257W 0.42
YNL281W 0.42	YDR111C 0.42	YLR283W 0.42	YLL011W 0.42	YBR257W 0.41	YBR016W 0.41	YER144C 0.41	YER057C 0.41
YGR161C 0.41	YDR233C 0.41	YGR072W 0.41	YEL068C 0.40	YPL062W 0.40	YCR072C 0.40	YGR070W 0.40	YJL084C 0.39
YBR226C 0.38	YIR032C 0.38	YER188W 0.38	YNL050C 0.38	YMR238W 0.38	YHR065C 0.38	YPL070W 0.38	YGR268C 0.38
YMR244W 0.38	YBR157C 0.37	YGL076C 0.37	YBL101W-B 0.37	YBR297W 0.37	YLR024C 0.37	YHR068W 0.37	YJL033W 0.37
YJL050W 0.37	YFL003C 0.37	YGL047W 0.37	YMR268C 0.36	YLR435W 0.36	YGR227W 0.36	YKL120W 0.36	YER079W 0.36
YGL171W 0.36	YGL078C 0.36	YCR057C 0.36	YPR016C 0.36	YLR401C 0.36	YGR294W 0.36	YIL058W 0.36	YPR142C 0.36
YIL040W 0.36	YHL021C 0.35	YJL032W 0.35	YHR201C 0.35	YJL130C 0.35	YFL006W 0.35	YJL151C 0.35	YMR192W 0.35
YIR040C 0.35	YDR174W 0.35	YNL178W 0.35	YDL111C 0.35	YLR425W 0.35	YDR358W 0.35	YDR041W 0.35	YGR103W 0.35

Table B.5: Continue of Table B.4

YOR267C 0.35	YML045W 0.34	YGL166W 0.34	YIL056W 0.34	YMR049C 0.34	YJR051W 0.34	YKL018W 0.34	YOR155C 0.34
YCR027C 0.34	YCL067C 0.34	YMR008C 0.34	YHL002W 0.33	YHR183W 0.33	YFR017C 0.33	YGR100W 0.33	YMR160W 0.33
YPL146C 0.33	YAR028W 0.33	YKL136W 0.33	YNL283C 0.33	YBR240C 0.33	YDR185C 0.33	YFR055W 0.33	YER126C 0.33
YML004C 0.33	YPL231W 0.32	YMR034C 0.32	YMR035W 0.32	YKL216W 0.32	YBL066C 0.32	YIL167W 0.32	YMR051C 0.32
YGR243W 0.32	YER015W 0.31	YOR224C 0.31	YGL050W 0.31	YDR059C 0.31	YML088W 0.31	YGR001C 0.31	YDR074W 0.31
YCR035C 0.31	YLR202C 0.31	YMR088C 0.31	YGR036C 0.31	YDL086W 0.31	YMR272C 0.31	YGR080W 0.31	YMR284W 0.31
YPL107W 0.31	YBR234C 0.31	YLR241W 0.30	YBR174C 0.30	YPR014C 0.30	YDR368W 0.30	YML062C 0.30	YMR020W 0.30
YML110C 0.30	YPL181W 0.30	YHL036W 0.30	YEL053C 0.30	YML121W 0.30	YOR079C 0.29	YOL121C 0.29	YPL226W 0.29
YOR169C 0.29	YMR135C 0.29	YMR182C 0.29	YLR239C 0.29	YOR219C 0.29	YKR106W 0.29	YGR255C 0.29	YGR290W 0.29
YIR042C 0.29	YLL049W 0.29	YIR011C 0.29	YJL001W 0.29	YDR080W 0.29	YGR145W 0.29	YFL055W 0.29	YNR012W 0.28
YMR081C 0.28	YLR130C 0.28	YER080W 0.28	YLR451W 0.28	YMR217W 0.28	YNL022C 0.28	YLL018C 0.28	YGR076C 0.28
YOR014W 0.28	YDL147W 0.28	YPL012W 0.28	YGR190C 0.28	YPL259C 0.27	YOR167C 0.27	YKL110C 0.27	YPL121C 0.27
YDR280W 0.27	YER132C 0.27	YGL039W 0.27	YJL109C 0.27	YJL148W 0.27	YGL048C 0.27	YCL049C 0.27	YHL006C 0.27
YBR274W 0.27	YHR128W 0.27	YGR233C 0.26	YFR052W 0.26	YDR424C 0.26	YDR178W 0.26	YNR046W 0.26	YDL027C 0.26
YIR037W 0.26	YLR011W 0.26	YGR216C 0.26	YBR012W-B 0.26	YJL067W 0.22	YNL096C 0.21	YNL241C 0.21	YMR208W 0.19
YNL069C 0.19	YNL226W 0.18	YNR061C 0.15	YEL018W 0.10	YIR035C 0.06	YDR038C 0.05	YDR160W 0.03	YPR024W 0.03
YJR080C 0.03	YOL003C 0.00	YLR066W 0.00	YPR029C -0.01	YJL017W -0.02	YGL173C -0.02	YMR267W -0.04	YNL236W -0.07
YJL024C -0.07	YMR221C -0.09	YKR079C -0.10	YDR029W -0.12	YPR150W -0.15	YPL040C -0.17	YMR165C -0.20	YHR191C -0.20
YNL011C -0.21	YNL169C -0.27	YGR065C -0.32	YCR048W -0.34	YMR149W -0.35	YDR208W -0.36	YFL038C -0.38	YAL060W -0.69
YHR127W -0.83							

Bibliography

H. Akaike. A new look at the statistical model identification. *IEEE*, AC19: 716–723, 12 1974.

T. W. Anderson. *An introduction to multivariate statistical analysis(3rd ed.)*. John Wiley, 2003.

T. Benagliz, D. Chauveau, D. R. Hunter, and D. S. Young. mixtools: An r package for analyzing finite mixture models. *Journal of Statistical Software*, 32:1–29, 2009.

G. E. P. Box and G. C. Tiao. *Bayesian inference in statistical analysis*. Addison-Wesley Publishing Co., 1973.

L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, Nov 1995.

T. Cederborg, M. Li, A. Baranes, and P. Oudeyer. Incremental local online gaussian mixture regression for imitation learning of multiple tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 267–274, Taipei, Oct. 2010.

J. Chen and Z. Chen. Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95:759–771, 2008.

E. Conlon, X. Liu, J. Lieb, and J. Liu. Interarating regulatory motif discovery and genome-wide expression analysis. *Proceedings of the National Academy of Sciences of the United States of America*, 100, 2003.

P. Dellaportas. Bayesian classification of neolithic tools. *Applied Statistics*, 47: 279–297, 1998.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

N. Draper and H. Smith. *Applied Regression Analysis*. Jonh Wiley Son, New York, 1981.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(407-451), 2004.

M.A. Efroymson. *Multiple regression analysis*. Mathematical Methods for Digital Computers. Wiley, 1960.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of American Statistical Association*, 91:1348–1360, 2001.

I.E. Frank and J.H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35:109–148, 1993.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of computer and system sciences*, 55:119–139, 1997.

W. J. Fu. Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7:397–416, 1998.

S. Gaffney and P. Smyth. Trajectory clustering with mixtures of regression models. Technical Report 99-15, University of California,Irvine, 1999.

A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel andM. B. Eisen, G. Storz, D. Botstein, and P. O. Brown. Genomic expression program in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11:4241–4257, 2000.

A. E. Gelfand and A. F. M. Smith. Sampling based approaches to calculationg marginal densities. *J. Am Statist*, 85:398–409, 1990.

W. R. Gilks, L. Oldfield, and A. Rutherford. *Statistical analysis. In Leucocyte Typing IV: White Cell Differentiation Antigens.* Oxford University Press, Oxford, 1989.

R. Gnanadesikan. *Methods for statistical data analysis of multivariate observations*, chapter 4. John Wiley Sons, Inc., 1997.

A. D. Gordon. *Classification-Monographs on applied probability and statistics.* Chapman and Hall Ltd, 1981.

B. Grun and F. Leisch. Finite mixtures of generalized linear regression models. Technical Report 013, University of Munich, 2007.

T. Hasite, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer Verlag, 2001.

V. Hasselblad. Estimation of parameters for a mixture of normal frequency distributions. *Technometrics*, 8, 1966.

T. Hesterbery, N. H. Choi, L. Meier, and C. Fraley. Least angle and l1 penalized regression: A review. *Statistics Survey*, 2:61–93, 2008.

R. R. Hocking. The analysis and selection of variables in linear regression. *Biometrics*, 32:1–49, 1976.

H. Hotelling. *Multivariate quality control, illustrate by the air testing of sample bombsight. In Selected Techniques of Statistical Analysis*, pages 111–184. McGraw Hill, 1947.

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

J. E. Jackson. Quality control methods for two related variables. *Ind. Qual. Control*, 12:2–6, 1956.

R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis(6th ed.)*, chapter 11. Pearson Prentice Hall, 2007.

A. Khalili. New estimation and feature selection methods in mixture-of-experts models. *The Canadian Journal of Statistics*, 38:519–539, 2010.

A. Khalili. An overview of the new feature selection methods in finite mixture of regression models. *Journal of the Iranian Statistical Society*, 10:201–235, 2011.

A. Khalili, J. Chen, and S. Lin. Feature selection in finite mixture of sparse normal linear models in high-dimensional feature space. *Biostatistics*, 12: 156–172, 2011.

M. Liu and H. Bozdogan. Multivariate regression models with power exponential random errors and subset selection using genetic algorithms with information complexity. *European Journal of Pure and Applied Mathematics*, 1: 4–37, 2008.

X.S. Liu, D.L. Brutlag, and J.S. Liu. An algorithm for finding protein–dna binding sites with applications to chromatin-immunoprecipitation microarray experiments. *Nature biotechnology*, 20:835–839, 2002.

G. Mclachlan and D. Peel. *Finite Mixture Models*. Wiley series in probability and statistics. John Wiley Sons, New York, 2001.

M.R Osborne, B. Presnell, and B. Turlach. On the lasso and its dual. *Journal of Computational and Graphical Statistics*, 9:319–337, 2000.

K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society*, A:71–110, 1894.

S. T. Rachev and S. Mittnik. Stable paretian models in finance. *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, 2000.

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

C. R. Rao. Large sample tests of statistical hypotheses concerning several parameters with applications to problems of estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 44:50–57, 1948.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6: 461–464, 1978.

N. Staedler, P. Buehlman, and S. Van de Geer. 1-penalization for mixture regression models(with discussion). *Test*, 19:209–285, 2010.

M. T. Subbotin. On the law of frequency of errors. *Matematicheskii Sbornik*, 31:296–300, 1923.

H. G. Sung. *Gaussian mixture of regression and classification.* PhD thesis, Houston, Texas, 2004.

P. Theodossiou. Financial data and the skewed generalized t distribution. *Management Science*, 44:1650–1661, 1998.

R. Tibshirani. Regrssion shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58:267–288, 1996.

D. M. Titterington, A. F. M Smith, and U. E Markov. *Statistical Analysis of Finite Mixture Distributions.* Wiley, 1985.

J. Toyli, K. Kaski, and A. Kanto. On the shape of asset return distribution. *Communications in Statistics-Simulation and Computation*, 31:489–521, 2002.

P. Vounatsou, T. Smith, and A.F.M. Smith. Bayesian analysis of two-component mixture distributions applied to estimationg malaria attributable fractions. *Applied Statistics*, 47:575–587, 1998.

S. Weisberg. *Applied Linear Regression.* Wiley, New York, 1980.

C. K. I. Williams and C. E. Rasmussen. *Gaussian processes for regression.* The MIT Press, 1996.

R. Zeckhauser and M. Thompson. Linear regression with non-normal error term. *Review of Economics and Statistics*, 52:280–286, 1970.

J. Zhang and F. Liang. Robust clustering using exponential power mixtures. *Biometrics*, 66:1078–1086, 2010.