Estimation and Feature Selection in High-Dimensional Mixtures-of-Experts Models

Bao-Tuyen Huynh

Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité MATHEMATIQUES

Préparée au sein de l'Université de Caen Normandie

Estimation and Feature Selection in High-Dimensional Mixtures-of-Experts Models

Présentée et soutenue par
Bao Tuyen HUYNH

Thèse soutenue publiquement le 03/12/2019
devant le jury composé de

Mme FLORENCE FORBES	Directeur de recherche à l'INRIA, INRIA Grenoble	Rapporteur du jury
M. JULIEN JACQUES	Professeur des universités, Université Lumière Lyon 2	Rapporteur du jury
Mme ÉMILIE DEVIJVER	Chargé de recherche, Laboratoire LIG Grenoble	Membre du jury
M. MICHAEL FOP	Maître de conférences, University College Dublin	Membre du jury
M. FAICEL CHAMROUKHI	Professeur des universités, Université Caen Normandie	Directeur de thèse

Thèse dirigée par FAICEL CHAMROUKHI, Laboratoire de Mathématiques 'Nicolas Oresme' (Caen)

UNIVERSITÉ CAEN NORMANDIE

MISΣ

LMNO

Acknowledgements

The completion of this thesis could not have been achieved without the help and support of many friends, colleagues, teachers and my family. First, I would like to thank my advisor Faïcel Chamroukhi for his guidance with an interesting topic. I appreciate all of the time you have spent discussing to all of my ideas, and all of the effort that you have put into fixing my bad grammar. Most of all, I appreciate your friendship.

A very special thanks to Mr. Eric Ricard for his kindness and support as a director of LMNO Lab.

I would like to thank Mme. Florence Forbes and Mr. Julien Jacques for reviewing my thesis. I am also grateful to Mme. Emilie Devijver and Mr. Michael Fop for agreeing to be part of my committee.

Next, to my current and former colleagues at LMMO Lab: Angelot, Arnaud, Cesar, Guillaume, Etienne, Julien, Hung, Mostafa, Nacer, Pablo, Thanh, Thien and Tin. Many thanks to you, guys. It has been a pleasure to spend the last three years or more getting to know you.

Very special thanks to my teacher Dang Phuoc-Huy. Your enthusiasm for statistics, your wealth of knowledge, and your diligence inspires me to learn more and work harder, in an attempt to emulate your achievements. Without your help and support I cannot finalize this work. I also extend my thanks to anh Vu and cô Phuong for their kindness as my brother and sister.

Finally, to my parents, my parents-in-law, my wife Ngo Xuan Binh-An, my sisters and my brothers-in-law. Thank you for always being there for me. You make me feel warm, safe and loved. I am very fortunate to be part of our happy family. Last but not least, thank you my three little kids. You always make me feel happy and I love you with all my heart.

Caen, September 30, 2019
Huynh Bao-Tuyen

Abstract

The statistical analysis of heterogeneous and high-dimensional data is being a challenging problem, both from modeling, and inference point of views, especially with the today's big data phenomenon. This suggests new strategies, particularly in advanced analyses going from density estimation to prediction, as well as the unsupervised classification, of many kinds of such data with complex distribution. Mixture models are known to be very successful in modeling heterogeneity in data, in many statistical data science problems, including density estimation and clustering, and their elegant Mixtures-of-Experts (MoE) variety, which strengthen the link with supervised learning and hence deals furthermore with prediction from heterogeneous regression-type data, and for classification. In a high-dimensional scenario, particularly for data arising from a heterogeneous population, using such MoE models requires addressing modeling and estimation questions, since the state-of-the art estimation methodologies are limited.

This thesis deals with the problem of modeling and estimation of high-dimensional MoE models, towards effective density estimation, prediction and clustering of such heterogeneous and high-dimensional data. We propose new strategies based on regularized maximum-likelihood estimation (MLE) of MoE models to overcome the limitations of standard methods, including MLE estimation with Expectation-Maximization (EM) algorithms, and to simultaneously perform feature selection so that sparse models are encouraged in such a high-dimensional setting. We first introduce a mixture-of-experts' parameter estimation and variable selection methodology, based on ℓ_1 (lasso) regularizations and the EM framework, for regression and clustering suited to high-dimensional contexts. Then, we extend the method to regularized mixture of experts models for discrete data, including classification. We develop efficient algorithms to maximize the proposed ℓ_1-penalized observed-data log-likelihood function. Our proposed strategies enjoy the efficient monotone maximization of the optimized criterion, and unlike previous approaches, they do not rely on approximations on the penalty functions, avoid matrix inversion, and exploit the efficiency of the coordinate ascent algorithm, particularly within the proximal Newton-based approach.

Keywords: Mixture models; Mixture of Experts; Regularized Estimation; Feature Selection; Lasso; ℓ_1-regularization; Sparsity; EM algorithm; MM Algorithm; Proximal-Newton; Coordinate Ascent; Clustering; Classification; Regression; Prediction.

Contents

Chapter 1

Introduction

Contents

1.1 Scientific context

Nowadays, big data are collected and mined in almost every area of science, entertainment, business and industry. For example, medical scientists study the genomes of patients to decide the best treatments and to learn the underlying causes of their disease. Often, big data is heterogeneous, high-dimensional, unlabeled, involve missing characteristics, etc. Such complexity poses some issues on treating and analyzing this kind of data. Analyses include prediction on future data, and data exploration to summarize the main information within the data, or reveal hidden useful information, like groups of common characteristics, which can be achieved via optimized clustering techniques. Analyses also cover identifying redundant or usefulness features, in representing the data, for which feature selection methodologies can be helpful.

Such challenges impose new analysis strategies including pushing forward the modeling methodologies, along with optimized model inference algorithms. In such area of analysis, main state-of-the art methods rely either on learning automatic data analysis systems like neural networks (Bishop, 1995), support vectors machines (Vapnik, 1998), kernel machines (Scholkopf and Smola, 2001), and also provable statistical generative learning models (Friedman et al., 2001), including statistical latent variable models. Statistical analysis of data is indeed an efficient tool that takes into account the randomness part of the data, measures uncertainty, and hence provides a nice framework for density estimation, prediction and unsupervised learning from data, including clustering. Among statistical models, mixture models (Titterington et al., 1985; McLachlan and Peel., 2000) are the standard way for the analysis of heterogeneous data across a broad number of fields including bioinformatics, economics, machine learning, among many others. Such kind of models have been used for example applications including complex systems

maintenance (Chamroukhi et al., 2008, 2009a; Onanena et al., 2009; Chamroukhi et al., 2011), bioacoustics (Chamroukhi et al., 2014; Bartcus et al., 2015).

Thanks to their flexibility, mixture models can be used to estimate densities and cluster data arising from complex heterogeneous populations. They are also being used to conduct prediction analysis including regression, via their Mixtures of experts (MoE) extension Jacobs et al. (1991); Jordan and Jacobs (1994). Basically, maximum likelihood estimation (MLE) (Fisher, 1912) with EM algorithms (Dempster et al., 1977; McLachlan and Krishnan, 2008) is the common way to conduct parameter estimation of mixture models and MoE models. However, applying these methods directly on high-dimensional data set has some drawbacks. For example, estimating the covariance matrix in Gaussian mixture models in a high-dimensional scenario is "a curse of dimensionality" problem. Furthermore, a more serious problem occurs in the EM algorithm when computing the posterior probabilities in the E-step which require the inversion of the covariance matrices. The same problem can be found while applying the Newton-Raphson (Boyd and Vandenberghe, 2004) procedure in the M-step of the EM algorithm namely for mixtures-of-experts.

Actually, in a high-dimensional setting, the features can be correlated and the actual features that explain the problem reside in a low-dimensional intrinsic space. Hence, there is a need to conduct an adapted estimation and select a subset of the relevant features, that really explain the data. The Lasso or ℓ_1-regularized linear regression method introduced by Tibshirani (1996) is a successful method that has been shown to be effective ant in classical statistical analysis like regression on homogeneous data. Lasso provides two advantages: it yields sparse solution vectors, having only some coordinates that are nonzero and the convexity of the related optimization problem greatly simplifies the computation.

In this thesis, we focus on Mixtures of experts (MoE) and for generalized linear models. They go beyong density estimation and clustering of vectorial data, and provide an efficient framework to density estimation, clustering, and prediction of regression-type data, as well as supervised classification of heterogeneous data (Jacobs et al., 1991; Yuksel et al., 2012; Jiang and Tanner, 1999a,b; Grün and Leisch, 2007). MoE are used in several applications such as: predicting the daily electricity demand of France (Weigend et al., 1995), generalizing the autoregressive models for time-series data (Zeevi et al., 1997; Wong and Li, 2001), recognizing handwritten digits (Hinton et al., 1995), segmentation and clustering time series with changes in regime (Chamroukhi et al., 2009b; Samé et al., 2011; Chamroukhi and Nguyen, 2018), human activity recognition (Chamroukhi et al., 2013), etc. Unfortunately, modeling with mixtures of experts models in the case of high-dimensional predictors is still limited. Our objectives here are therefore to study the estimation and feature selection in high-dimensional mixtures-of-experts models, and to:

i) propose new estimation, and feature selection strategies, to overcome such limitations. We wish to achieve that by introducing new regularized estimation and feature selection in the MoE framework for generalized linear models based on Lasso-like penalties. We study

them for different family of mixtures of experts models, including MoE with Gaussian, Poisson and logistic expert distributions.

ii) develop efficient algorithms for estimate the parameters of these regularized models. The developed algorithms perform simultaneous feature estimation and selection, and should enjoy the capability of encouraging sparsity, and deal with some typical high-dimensional issues like by avoiding matrix inversion, so that they perform quite well in high-dimensional situations.

In the following section, we summarize the contributions of the thesis.

1.2 Contributions of the thesis

The manuscript is organized as follows. Chapter 2 is dedicated to state-of-the-art. Chapter 3 presents our first contribution to the estimation and feature selection of mixtures-of-experts models, for continuous data. Then, Chapter 4, presents our second contribution. It is on the estimation and feature selection of MoE models, for discrete data. Finally, in Chapter 5, we discuss our research, draw conclusions, and highlight some potential future avenues to pursue. Technical details related to the mathematical developments of our contributions are provided in Appendices A, B and C.

More specifically, first, in Chapter 2, we provide a substantial review of state-of-the art models and algorithms related to scientific subjects addressed in the thesis. We first focus on the general mixture modeling framework, as an appropriate choice of modeling heterogeneity in data. We describe its statistical modeling aspects and the related estimation strategies, and model selection techniques, with a particular attention given to the MLE via the EM algorithm. Then, we revisit this mixture modeling context, in the framework of regression problems on heterogeneous data, and present its extension to the framework of Mixtures of Experts models. At the next stage, we consider these models in a high-dimensional setting, including the case with Gaussian components, and describe the three main strategies to address the curse of dimensionality, i.e, the two-fold dimensionality reduction approach, the one of spectral decomposition of the model covariance matrices, and the one of feature selection via Lasso regularization techniques. We opt for this latter regularization strategy, and present it for the case of mixture of regression and MoE models. We review the regularized maximum likelihood estimation and feature selection for these models via adapted EM algorithms.

Then, in Chapter 3, we introduce a novel approach for the estimation and feature selection of mixtures-of-experts for regression with potentially high-dimensional predictors, and a heterogeneous population. The approach simultaneously performs parameter estimation, feature selection, clustering and regression on the heterogeneous regression data. It consists of regularized maximum-likelihood approach with a dedicated regularization that, on the one hand, encourages sparsity thanks to a Lasso-like regularization part, and on the other hand, efficient

to handle due to the convexity of the ℓ_1 penalty. We propose an effective hybrid Expectation-Maximization (EM) framework, to efficiently solve the resulting optimization problem which monotonically maximizes the regularized log-likelihood. It results into three hybrid algorithms for maximizing the proposed objective function, that is, a Majorization-Maximization (MM) algorithm, a coordinate ascent, and a proximal Newton-type procedure. We show that the proposed approach does not require an approximate of the regularization term, and the three developed hybrid algorithms, allow to automatically select sparse solutions without any approximation on the penalty functions. We rely on a modified BIC criterion to achieve the model selection task, including the selection of the number of components, and the regularization tuning hyper-parameters. An experimental is then considered to compare the proposed approach to the main competitive state-of-art methods for the subject. Evaluation is made to assess the performance of the approach in terms of clustering, density estimation, regression, and sparsity, of heterogeneous data by MoE models. Extensive experiments on both simulations and real data, show that the proposed approach outperforms its competitive and is very encouraging to address the high-dimensional issue. This chapter has mainly led to the journal publication (Chamroukhi and Huynh, 2019) and conferences papers.

Next, in Chapter 4, we consider another family of MoE models, the one of discrete data, including MoE for counting data and MoE for classification, and introduce our main second contribution. We present a new regularized MLE strategy to the estimation and feature selection of dedicated mixtures of generalized linear expert models in a high-dimensional setting. We develop an efficient EM algorithm, which relies on a proximal Newton approximation, to monotonically maximize the proposed penalized log-likelihood criterion. The presented strategy simultaneously performs parameter estimation, feature selection, and classification on the heterogeneous discrete data. An advantage of the introduced proximal Newton-type strategy consists in the fact that one just need to solve weighted quadratic Lasso problems to update the parameters. Efficient tools such as coordinate ascent algorithm can be used to deal with these problems. Hence, the proposed approach does not require an approximate of the regularization term, and allow to automatically select sparse solutions without thresholding. Our approach is shown to perform well including in a high-dimensional setting and to outperform competitive state of the art regularized MoE models on several experiments on simulated and real data. The main publication related to this chapter is Huynh and Chamroukhi (2019) and other communication.

Finally, in Chapter 5, we discuss our developed research, draw some conclusions and open some future directions. The thesis research publications results, including R packages of open-source codes, are given in the list of publications and communications.

A substantial summary in French is provided in Résumé long en français at the end of the manuscript.

Chapter 2

State of the art

Contents

2.1 Introduction

In this chapter, we provide an overview of the finite mixture models (FMMs) (McLachlan and Krishnan, 2008), which are widely used in statistical learning for analyzing heterogeneous data.

More specifically, we focus on FMMs for modeling, density estimation, clustering and regression. We also review the Expectation-Maximization (EM) algorithm (Dempster et al., 1977; McLachlan and Krishnan, 2008) for maximum likelihood parameter estimation (MLE) of FMMs (McLachlan and Peel., 2000). However, the problem lies in the fact that applying these EM algorithms directly on high-dimensional data sets poses some drawbacks. In such situations, we require new techniques to handle these difficulties.

For data clustering with FMMs, several propositions were introduced in literature to deal with high-dimensional data sets such as: Parsimonious Gaussian mixture models (Banfield and Raftery, 1993; Celeux and Govaert, 1995), Mixture of factor analyzers (Ghahramani and Hinton, 1996; McLachlan and Peel, 2000; McLachlan et al., 2003), etc. Recently, some authors suggested to use the regularized approaches for data clustering in high-dimensional setting. We take a survey on these approaches here.

Later, a review on regularized methods for variable selection in mixture of regression models, which form the core of this thesis, is presented. These works are mainly inspired by the Lasso regularization. We also give some discussions on the advantages and drawbacks of the approaches not only in term of modeling but also in term of parameter estimation approaches.

Most of the methods and approaches in this chapter are well-known in literature and are included to provide context for later chapters. This chapter is organized into four main parts. The first and second parts are concerned with FMMs for density estimation, clustering and regression tasks, and EM algorithms for parameter estimation. The third and last parts address the same problems but in high-dimensional scenario. Finally, we draw some conclusions and introduce our research directions in this thesis.

2.2 Finite mixture models

Let $\boldsymbol{X} \in \mathcal{X}$ be a random variable where $\mathcal{X} \subset \mathbb{R}^p$ for some $p \in \mathbb{N}$. Denote the probability density function of $\boldsymbol{X}$ by $f(\boldsymbol{x})$ and let $f_k(\boldsymbol{x})$ $(k = 1, 2, \ldots, K)$ be K probability density functions over $\mathcal{X}$. X is said to arise from a finite mixture model (FMM) if the density function of $\boldsymbol{X}$ is decomposed into a weighted linear combination of K component densities,

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

where $\pi_k > 0$, for all k and $\sum_{k=1}^{K} \pi_k = 1$. The parameters $\pi_1, \ldots, \pi_K$ are referred to as mixing proportions and $f_1, \ldots, f_K$ are referred to as component densities.
We can characterize this model via a hierarchical construction by considering a latent variable Z, where Z represents a categorical random variable which takes its values in the finite set $\mathcal{Z} = \{1, \ldots, K\}$ and $\mathbb{P}(Z = k) = \pi_k$. If we assume that the conditional density of $\boldsymbol{X}$ given

$Z = k$ equals $f_k(x)$, then the join density of $(\boldsymbol{X}, Z)$ can be written as following

$$f(\boldsymbol{x}, z) = \prod_{k=1}^{K} [\pi_k f_k(\boldsymbol{x})]^{\mathbb{I}(z=k)} \tag{2.2}$$

where $\mathbb{I}(z = k)$ is the indicator function and equals 1 when $z = k$ and 0 otherwise. Hence, the marginal density of $\boldsymbol{X}$ is

$$\begin{aligned} f(\boldsymbol{x}) &= \sum_{z=1}^{K} f(\boldsymbol{x}, z) \\ &= \sum_{z=1}^{K} \big(\prod_{k=1}^{K} [\pi_k f_k(\boldsymbol{x})]^{\mathbb{I}(z=k)} \big) \\ &= \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}). \end{aligned} \tag{2.3}$$

This model was first proposed by Pearson (1894). In his work, Pearson modeled the distribution of the forehead breadth to body length ratios for 1000 Neapolitan crabs with a mixing of $K = 2$ univariate Gaussian distributions. However, as we have been seen through literature review, there are mainly three types of FMM (McLachlan and Peel. (2000), Nguyen (2015)):

- Homogeneous-parametric FMMs: the component densities come from the same parametric family, such as Gaussian Mixture Models, t-distribution Mixture Models;
- Heterogeneous FMMs: the component densities come from the different parametric family, like zero inflated Poisson distribution (ZIP), uniform-Gaussian FMMs, etc;
- Nonparametric FMMs.

For the parametric FMMs, assuming that each of the K component densities typically consists of a relatively simple parametric model $f_k(\boldsymbol{x}; \boldsymbol{\theta}_k)$ (such as for a homogeneous Gaussian mixture model $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$).

The FMMs are widely used in a variety of applications. For those of biology, economics, genetics interested readers can refer to Titterington et al. (1985). Besides that, these models are also used in many modern facets of scientific research, most notably in bioinformatics, pattern recognition and machine learning.

In this thesis, we shall refer FMMs as homogeneous-parametric FMMs to distinguish them from heterogeneous FMMs and nonparametric FMMs. It turns out that many parameter estimation methods for FMMs have been proposed in literature, such as Pearson (1894) used the method of moments to fit a mixture of two univariate Gaussian components, Rao (1948) used Fisher′s method of scoring to estimate a mixture of two homoscedastic Gaussian components, etc. However, the more common methods are the maximum likelihood (McLachlan and Peel. (2000) and the Bayesian methods (Maximum A Posteriori (MAP)) where a prior distribution

is assumed for the model parameters (see Stephens et al. (2000)). Here, we consider the maximum likelihood framework. For more details on the method of point estimation, viewers can refer to Lehmann and Casella (2006). The optimization algorithm for performing the maximum likelihood parameter estimation is the Expectation-Maximization (EM) algorithm (see Dempster et al. (1977) and also McLachlan and Krishnan (2008)). In the next section, the use of the Expectation-Maximization algorithm for learning the parameters of FMMs will be discussed. The object is to maximize the log-likelihood as a function of the model parameters $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_{K-1}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$, over the parameter space Ω.

Assume that $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is an sample generated from K clusters and each cluster follows a probability distribution $f_k(\boldsymbol{x}_i; \boldsymbol{\theta}_k)$ (a common choice of $f_k(\boldsymbol{x}_i; \boldsymbol{\theta}_k)$ can be Gaussian distribution $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$). A hidden variable Z_i $(i = 1, \ldots, n)$ represents a multinomial random variable which takes its values in the finite set $\mathcal{Z} = \{1, ..., K\}$ and for each observation $\boldsymbol{x}_i$ the probability that $\boldsymbol{x}_i$ belongs to the kth cluster is given by $\mathbb{P}(Z_i = k) = \pi_k$. Hence, the data set $\mathbf{x}$ can be interpreted as an i.i.d sample generated from a FMMs with the probability density function $f(\boldsymbol{x}; \boldsymbol{\theta})$, where

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}; \boldsymbol{\theta}_k); \ \pi_k > 0, \ \sum_{k=1}^{K} \pi_k = 1. \tag{2.4}$$

The observed-data log-likelihood function therefore is given by

$$\begin{aligned} L(\boldsymbol{\theta}; \mathbf{x}) &= \log \prod_{i=1}^{n} f(\boldsymbol{x}_i; \boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}_i; \boldsymbol{\theta}_k). \end{aligned} \tag{2.5}$$

In this case, the log-likelihood is a nonlinear function. Therefore, there is no way to maximize it in a closed form. However, it can be locally maximized using iterative procedures such as Newton-Raphson procedure or the EM algorithm. We mainly focus on the EM algorithm, which is widely used for FMMs. The next section presents the EM algorithm for finding local maximizers of the general parametric FMMs. We will then apply this algorithm to GMMs.

2.2.1 Maximum likelihood estimation for FMMs via EM algorithm

The Expectation-Maximization (EM) algorithms are a class of iterative algorithms that were first considered in Dempster et al. (1977) and the book of McLachlan and Peel. (2000) provides a complete review on the topic. It is a broadly used for the iterative computation of the maximum likelihood estimates on the framework of latent models. In particular, an EM algorithm simplifies considerably the problem of fitting FMMs by the maximum likelihood. This can be described as follows.

EM algorithm for FMMs

In the EM framework, the observed-data vector $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is viewed as being incomplete, where each $\boldsymbol{x}_i$ is associated with the K-dimensional component label vector $\boldsymbol{z}_i$ (which is also called the latent variable) and the kth element of $\boldsymbol{z}_i$, $z_{ik} = 1$ or 0, according to whether $\boldsymbol{x}_i$ did or did not arise from the kth component of the mixture (2.4) ($i = 1, \ldots, n; k = 1 \ldots, K$).

The component-label vectors $(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$ are taken to be the realized values of the random vectors $(Z_1, \ldots, Z_n)$. Thus Z_i is distributed according to a multinomial distribution consisting of one draw on K categories with probabilities $(\pi_1, \ldots, \pi_K)$

$$Z_1, \ldots, Z_n \overset{i.i.d}{\sim} \text{Mult}(1; \pi_1, \pi_2, \ldots, \pi_K). \tag{2.6}$$

The complete-data log-likelihood function over observed variable $\mathbf{x}$ and latent variable $\mathbf{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$, governed by parameters $\boldsymbol{\theta}$ can be written as

$$\begin{aligned} L_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}) &= \log f(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) \\ &= \log \prod_{i=1}^{n} \prod_{k=1}^{K} \big(\pi_k f_k(\boldsymbol{x}_i; \boldsymbol{\theta}_k)\big)^{z_{ik}} \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \big\{\log \pi_k + \log f_k(\boldsymbol{x}_i; \boldsymbol{\theta}_k)\big\}, \end{aligned} \tag{2.7}$$

where $f(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta})$ is the complete-data probability density function.

The EM algorithm is an iterative algorithm where each iteration consists of two steps, the E-step (Expectation step) and the M-step (Maximization step). Let $\boldsymbol{\theta}^{[q]}$ denote the value of the parameter vector $\boldsymbol{\theta}$ after the qth iteration, and let $\boldsymbol{\theta}^{[0]}$ be some initialization value. The EM algorithm starts with $\boldsymbol{\theta}^{[0]}$ and iteratively alternates between the two following steps until convergence:

E-step

This step (on the $(q+1)$th iteration) consists the computation of the conditional expectation of $L_c(\boldsymbol{\theta}; \boldsymbol{X}, Z)$ given $\mathbf{x}$, using $\boldsymbol{\theta}^{[q]}$ for $\boldsymbol{\theta}$. This expectation, denoted $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{[q]})$, is given by

$$\begin{aligned} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{[q]}) &= \mathbb{E}[L_c(\boldsymbol{\theta}; \boldsymbol{X}, Z) | \mathbf{x}; \boldsymbol{\theta}^{[q]}] \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{E}[Z_{ik} | \mathbf{x}; \boldsymbol{\theta}^{[q]}] \big\{\log \pi_k + \log f_k(\boldsymbol{x}_i; \boldsymbol{\theta}_k)\big\} \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{[q]} \big\{\log \pi_k + \log f_k(\boldsymbol{x}_i; \boldsymbol{\theta}_k)\big\}, \end{aligned} \tag{2.8}$$

where

$$\tau_{ik}^{[q]} = \mathbb{E}[Z_{ik}|\mathbf{x};\boldsymbol{\theta}^{[q]}] = \mathbb{P}(Z_{ik}=1|\mathbf{x};\boldsymbol{\theta}^{[q]}) = \frac{\pi_k^{[q]} f_k(\boldsymbol{x}_i;\boldsymbol{\theta}_k^{[q]})}{\sum_{l=1}^{K}\pi_l^{[q]} f_l(\boldsymbol{x}_i;\boldsymbol{\theta}_l^{[q]})} \tag{2.9}$$

is the posterior probability that $\boldsymbol{x}_i$ belongs to the kth component density. As shown in the expression of $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[q]})$ above, the E-step simply requires computing of the conditional posterior probabilities $\tau_{ik}^{[q]}$ $(i=1,\ldots,n;\ k=1,\ldots,K)$.

M-step

The M-step (on the $(q+1)$th iteration) evaluates the estimate of $\boldsymbol{\theta}$ by the value $\boldsymbol{\theta}^{[q+1]}$ that maximizes the function $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[q]})$ with respect to $\boldsymbol{\theta}$ over the parameter space Ω:

$$\boldsymbol{\theta}^{[q+1]} = \arg\max_{\boldsymbol{\theta}\in\Omega} Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[q]}). \tag{2.10}$$

Rewrite Q in the separate form

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{[q]}\log\pi_k + \sum_{i=1}^{n}\tau_{ik}^{[q]}\log f_k(\boldsymbol{x}_i;\boldsymbol{\theta}_k). \tag{2.11}$$

The updated estimate $\boldsymbol{\pi}^{[q+1]} = (\pi_1^{[q+1]},\ldots,\pi_{K-1}^{[q+1]})^T$ of the mixing proportions π_k $(k=1,\ldots,K-1)$ is calculated independently of the updated estimate $\boldsymbol{\eta}^{[q+1]} = (\boldsymbol{\theta}_1^{[q+1]},\ldots,\boldsymbol{\theta}_K^{[q+1]})$ of the parameter vector $\boldsymbol{\eta} = (\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_K)$ containing the unknown parameters in the component densities of the FMM (2.4).

Hence, maximizing the function $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[q]})$ in (2.11) with respect to the parameter $\boldsymbol{\pi}$ can be performed using the method of Lagrange multiplier (by taking account of the constraint $\sum_{k=1}^{K}\pi_k = 1$), we obtain the updated estimates of the mixing proportions π_k

$$\pi_k^{[q+1]} = \sum_{i=1}^{n}\tau_{ik}^{[q]}/n, \tag{2.12}$$

for $k=1,\ldots,K$.

Now, in forming the estimate of the parameter vector $\boldsymbol{\eta}$ on the M-step of the $(q+1)$th iteration, it can be seen from (2.11) that $\boldsymbol{\eta}^{[q+1]}$ is obtained as an appropriate root of

$$\sum_{k=1}^{K}\sum_{i=1}^{n}\tau_{ik}^{[q]}\frac{\partial\log f_k(\boldsymbol{x}_i;\boldsymbol{\theta}_k)}{\partial\boldsymbol{\eta}} = \mathbf{0}. \tag{2.13}$$

One nice feature of the EM algorithm is that the solution of (2.13) often exists in closed form.

The E- and M-steps are alternated repeatedly until the difference between the (incomplete-data) log-likelihood function $L(\boldsymbol{\theta}^{[q+1]};\mathbf{x})$ and the log-likelihood function $L(\boldsymbol{\theta}^{[q]};\mathbf{x})$ is small enough such as

$$L(\boldsymbol{\theta}^{[q+1]};\mathbf{x}) - L(\boldsymbol{\theta}^{[q]};\mathbf{x}) < \varepsilon,$$

for a small value of some threshold ε.

Generally, the E- and M-steps have simple forms when the complete-data probability density function is from the exponential family (McLachlan and Krishnan, 2008; McLachlan and Peel., 2000). In cases where the M-step cannot be performed directly, other methods can be used to update $\boldsymbol{\theta}_k^{[q+1]}$ such as Newton-Raphson algorithm and other adapted extensions.

We now show that, after each iteration the value of the log-likelihood function $L(\boldsymbol{\theta}; \mathbf{x})$ (2.5) is not decreased, in the sense that

$$L(\boldsymbol{\theta}^{[q+1]}; \mathbf{x}) = \log g(\mathbf{x}; \boldsymbol{\theta}^{[q+1]}) \geq \log g(\mathbf{x}; \boldsymbol{\theta}^{[q]}) = L(\boldsymbol{\theta}^{[q]}; \mathbf{x}), \tag{2.14}$$

for each $q = 0, 1, \ldots$ and $g(\mathbf{x}; \boldsymbol{\theta})$ is the probability density function of the observed data vector $\mathbf{x}$. This property can be found via the results from Section 10.3 of Lange (1998) and is given in Appendix A.1.

In case that one can not find $\boldsymbol{\theta}^{[q+1]} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{[q]})$, Dempster et al. (1977) suggested to take $\boldsymbol{\theta}^{[q+1]}$ such that $Q(\boldsymbol{\theta}^{[q+1]}; \boldsymbol{\theta}^{[q]}) \geq Q(\boldsymbol{\theta}^{[q]}; \boldsymbol{\theta}^{[q]})$. The result of this is an increasing array of the log-likelihood values, i.e, $\log g(\mathbf{x}; \boldsymbol{\theta}^{[q+1]}) \geq \log g(\mathbf{x}; \boldsymbol{\theta}^{[q]})$. These algorithms are called generalized EM algorithms. The EM algorithm ensures that one will receive a monotonically increasing sequence $L(\boldsymbol{\theta}^{[q]}; \mathbf{x})$. A complete proof on the convergence properties of EM algorithm can be found in Wu (1983).

2.2.2 Gaussian mixture models

The most popular homogeneous-parametric FMMs are the Gaussian mixture models (GMMs). It is used for modeling the probability density function of random variables in $\mathbb{R}^p$. In this case, GMMs have density functions of the form

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.15}$$

where $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ denotes the multivariate Gaussian density with mean vector $\boldsymbol{\mu}_k$, covariance matrix $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\theta}$ is the vector containing all the parameters in $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_{K-1}\}$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$, $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K\}$ that represents this model. Since many natural measurements and processes tend to have Gaussian distributions, it is not hard to incorporate the use of GMMs. In these cases, the populations containing subpopulations of such measurements tends to have densities resembling GMMs. Furthermore, GMMs are among the simplest FMMs to estimate and are thus straightforward to apply in practice. Another critical property of GMMs is that, for a given continuous density function $f(\boldsymbol{x})$, one can use a GMM with K components to approximate $f(\boldsymbol{x})$. This result can be done by using the following theorem (see Section 33.1 of DasGupta (2008)).

Theorem 2.2.1 (Denseness of FMMs)**.** *Let $1 \leq p < \infty$ and $f(\boldsymbol{x})$ be a continuous density on*

$\mathbb{R}^p$. If $g(\boldsymbol{x})$ is any continuous density on $\mathbb{R}^p$, then given $\varepsilon > 0$ and a compact set $C \subset \mathbb{R}^p$, there exists an FMM of the form

$$\hat{f}(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \frac{1}{\sigma_k^p} g\Big(\frac{\boldsymbol{x} - \boldsymbol{\mu}_k}{\sigma_k}\Big),$$

such that $\sup_{\boldsymbol{x} \in C} |f(\boldsymbol{x}) - \hat{f}(\boldsymbol{x})| < \varepsilon$, for some $K \in \mathbb{N}$, where $\boldsymbol{\mu}_k \in \mathbb{R}^p$ and $\sigma_k > 0$ $(k = 1 \ldots, K)$.

By taking $g(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \mathbf{0}, \boldsymbol{\Delta})$, where $\boldsymbol{\Delta}$ is a constant $p \times p$ matrix and is symmetric positive definite, we see that GMMs with component densities from the location-scale family of g,

$$\Big\{ \frac{1}{\sigma^p} g\big(\frac{\boldsymbol{x} - \boldsymbol{\mu}}{\sigma}\big) \colon \boldsymbol{\mu} \in \mathbb{R}^p, \sigma > 0 \Big\} = \{ \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \sigma^2 \boldsymbol{\Delta}) : \boldsymbol{\mu} \in \mathbb{R}^p, \sigma > 0 \}$$

are dense in the class of continuous densities on $\mathbb{R}^p$.

Recent novel theoretical results about the approximation capabilities of mixture models are given in Nguyen et al. (2019b). In the next section, we present the EM algorithm for GMMs.

EM algorithm for GMMs

Consider the GMM case. Here, we assume that $\boldsymbol{x}_i \overset{i.i.d}{\sim} f(\boldsymbol{x}; \boldsymbol{\theta})$, with $f(\boldsymbol{x}; \boldsymbol{\theta})$ is a mixture density function of the form (2.15)

$$f(x; \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k); \ \pi_k > 0, \ \sum_{k=1}^{K} \pi_k = 1,$$

with normal components

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}_k|^{1/2}} \exp\Big(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k) \Big) \ (k = 1, \ldots, K).$$

From (2.5) and (2.7), the observed-data log-likelihood function for the GMM is given by

$$L(\boldsymbol{\theta}; \mathbf{x}) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.16}$$

and the complete-data log-likelihood is

$$L_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \big\{ \log \pi_k + \log \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \big\}. \tag{2.17}$$

With an initial parameter $\boldsymbol{\theta}^{[0]} = (\pi_1^{[0]}, \ldots, \pi_{K-1}^{[0]}, \boldsymbol{\theta}_1^{[0]}, \ldots, \boldsymbol{\theta}_K^{[0]})$ $(k = 1, \ldots, K)$ where $\boldsymbol{\theta}_k^{[0]} = (\boldsymbol{\mu}_k^{[0]}, \boldsymbol{\Sigma}_k^{[0]})$, the corresponding EM algorithm is defined as follows:

E-step This step (on the $(q+1)$th iteration) consists of computing the conditional expectation of (2.17), given the observed data $\mathbf{x}$, using the current value $\boldsymbol{\theta}^{[q]}$ for $\boldsymbol{\theta}$. In this case, we have

from (2.8) that

$$\begin{aligned} Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[q]}) &= \mathbb{E}[L_c(\boldsymbol{\theta};\boldsymbol{X},Z)|\mathbf{x};\boldsymbol{\theta}^{[q]}] \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{[q]}\log\pi_k + \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{[q]}\log\mathcal{N}(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k), \end{aligned} \tag{2.18}$$

where the posterior probabilities are given by (2.9)

$$\tau_{ik}^{[q]} = \frac{\pi_k^{[q]}\mathcal{N}(\boldsymbol{x}_i;\boldsymbol{\mu}_k^{[q]},\boldsymbol{\Sigma}_k^{[q]})}{\sum_{l=1}^{K}\pi_l^{[q]}\mathcal{N}(\boldsymbol{x}_i;\boldsymbol{\mu}_l^{[q]},\boldsymbol{\Sigma}_l^{[q]})} \ (i=1,\ldots,n;k=1,\ldots,K). \tag{2.19}$$

This step therefore computes the posterior probabilities that $\boldsymbol{x}_i$ belongs to the kth component density using the current parameter value $\boldsymbol{\theta}^{[q]}$.

M-step The M-step updates $\boldsymbol{\theta}$ by the value $\boldsymbol{\theta}^{[q+1]}$ that maximizes the function $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[q]})$ with respect to $\boldsymbol{\theta}$. Here, for the GMMs, the updated estimates of the mixing proportions π_k are as given by (2.12) and the updates of the component means $\boldsymbol{\mu}_k$ are given by

$$\boldsymbol{\mu}_k^{[q+1]} = \sum_{i=1}^{n}\tau_{ik}^{[q]}\boldsymbol{x}_i \Big/ \sum_{i=1}^{n}\tau_{ik}^{[q]}, \tag{2.20}$$

and for the problem, known as heteroscedastic, the component covariance matrices $\boldsymbol{\Sigma}_k$ are unequal. To model this, the updated estimate $\boldsymbol{\Sigma}_k^{[q+1]}$ of $\boldsymbol{\Sigma}_k$ is defined by

$$\boldsymbol{\Sigma}_k^{[q+1]} = \sum_{i=1}^{n}\tau_{ik}^{[q]}(\boldsymbol{x}_i-\boldsymbol{\mu}_k^{[q+1]})(\boldsymbol{x}_i-\boldsymbol{\mu}_k^{[q+1]})^T \Big/ \sum_{i=1}^{n}\tau_{ik}^{[q]}, \tag{2.21}$$

for $k=1,\ldots,K$.
For homoscedastic case, i.e, $\boldsymbol{\Sigma}_k=\boldsymbol{\Sigma}$ $(k=1,\ldots,K)$, then $\boldsymbol{\Sigma}$ is updated by

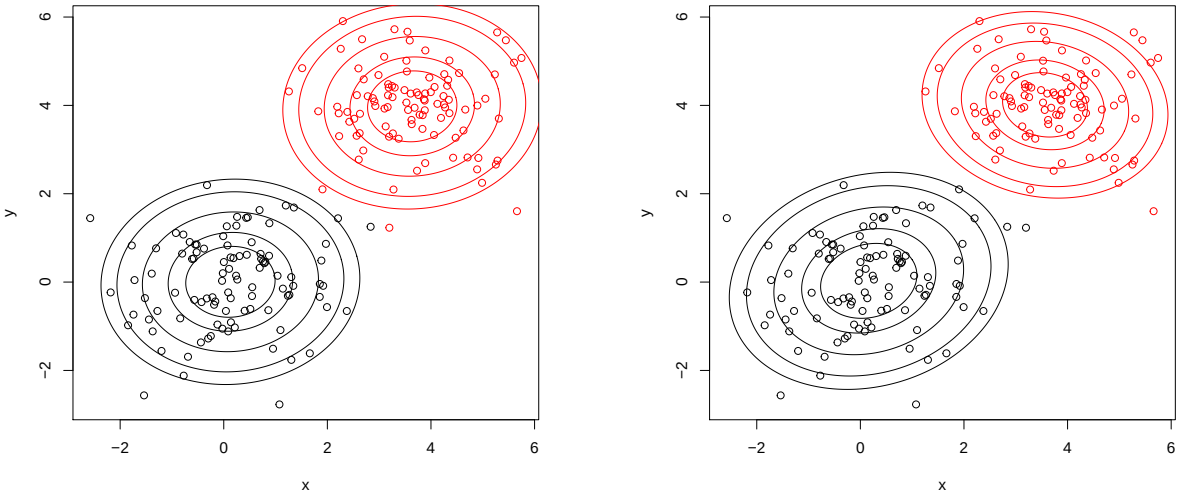
$$\boldsymbol{\Sigma}^{[q+1]} = \frac{1}{n}\sum_{k=1}^{K}\sum_{i=1}^{n}\tau_{ik}^{[q]}(\boldsymbol{x}_i-\boldsymbol{\mu}_k^{[q+1]})(\boldsymbol{x}_i-\boldsymbol{\mu}_k^{[q+1]})^T. \tag{2.22}$$

The above updated estimates of $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ can be deduced in an analogous method to deriving the maximum likelihood estimates of the mean vector and the covariance matrix (from an multivariate Gaussian distribution), that have been discussed by Anderson (2003) and given in Appendix A.2.

Example 2.2.1. *In this example, we generate* 200 *data points of a mixture of two Gaussian components with,*

$$\boldsymbol{\mu}_1=(0,0)^T,\ \boldsymbol{\mu}_2=(4,4)^T,\ \boldsymbol{\Sigma}_1=\boldsymbol{\Sigma}_2=\boldsymbol{\Sigma}=\mathbf{I}_2,\ \pi_1=\pi_2=1/2.$$

The results after using EM algoritm for homoscedastic case and heteroscedastic case are clarified



Homoscedastic case Heteroscedastic case

Figure 2.1: The results on testing data.

in figure 2.1. Here, for homoscedastic case we obtain

$$\hat{\boldsymbol{\mu}}_1 = (0.128806119, 0.005730867)^T,\ \hat{\boldsymbol{\mu}}_2 = (3.638555, 3.978157)^T,$$

$$\hat{\boldsymbol{\Sigma}} = \begin{pmatrix} 1.04168655 & 0.03520841 \\ 0.03520841 & 0.90212557 \end{pmatrix},\ \hat{\pi}_1 = 0.4992703,\ \hat{\pi}_2 = 0.5007297,\ L(\hat{\boldsymbol{\theta}}; \mathbf{x}) = -696.7831.$$

The parameter estimates for heteroscedastic case are

$$\hat{\boldsymbol{\mu}}_1 = (0.17277148, 0.03419084)^T,\ \hat{\boldsymbol{\mu}}_2 = (3.658629, 4.022911)^T,$$

$$\hat{\boldsymbol{\Sigma}}_1 = \begin{pmatrix} 1.2332897 & 0.1860267 \\ 0.1860267 & 1.0160972 \end{pmatrix},\ \hat{\boldsymbol{\Sigma}}_2 = \begin{pmatrix} 0.93025045 & -0.09940991 \\ -0.09940991 & 0.72055415 \end{pmatrix},$$

$$\hat{\pi}_1 = 0.5084509,\ \hat{\pi}_2 = 0.4915491,\ L(\hat{\boldsymbol{\theta}}; \mathbf{x}) = -693.8807.$$

However, estimating the covariance matrix of GMMs in large case ($p \gg 1$) is "a curse of dimensionality". For a p-variate GMM with K components, the maximum dimension of total number of parameters to estimate is equal to

$$(K-1) + Kp + Kp(p+1)/2,$$

where $(K-1)$, Kp and $Kp(p+1)/2$ are respectively the numbers of free parameters for the proportions, the means and the covariance matrices. Hence, the number of parameters to estimate is a quadratic function of p in the case of GMM. A large number of observations will be

necessary to correctly estimate those model parameters. Furthermore, a more serious problem occurs in the EM algorithm when computing the posterior probabilities $\tau_{ik} = \mathbb{P}[Z_i = k|\boldsymbol{x}_i; \boldsymbol{\theta}]$ which require the inversion of the covariance matrices $\boldsymbol{\Sigma}_k, k = 1, \ldots, K$. In addition, if the number of observations n is smaller than p ($n < p$) then the estimates of the covariance matrix $\hat{\boldsymbol{\Sigma}}_k$ are singular and the clustering methods cannot be used at all. We will mention some recent results that related to parameter reducing and feature selection for data clustering and classification in high dimension in Section 2.4. This also includes the GMMs for data clustering.

Alongside with GMMs, which is widely used for data clustering there has been an interest in the formulations of FMMs for random variables, utilizing non-Gaussian component densities. The multivariate t FMM was first considered in McLachlan and Peel (1998) and Peel and McLachlan (2000), for the clustering of data that arise from non-Gaussian subpopulations. To model data where the component densities have non-elliptical confidence sets some skewed densities in FMMs have been proposed such as skew-normal FMMs and skew-t FMMs. For those results related to these models one can refer to the work of (Lee and Mclachlan, 2013; Lee and McLachlan, 2013, 2014), etc. A Bayesian inference approach for skew-normal and skew-t FMMs can be found in Frühwirth-Schnatter and Pyne (2010).

2.2.3 Determining the number of components

Until now, we have considered the FMMs and an EM algorithm for estimation the parameters in a context of a known number of components K. Therefore, in practice, a natural question arises: "How to choose the number of components from the data?". This problem can be seen as a model selection problem. In FMM literature, it is one of the most interesting research topics and pays a lot of attentions; see (McLachlan and Peel., 2000, Chapter 6), Celeux and Soromenho (1996), Fraley and Raftery (1998), Fonseca and Cardoso (2007) for reviews on the topic.

In the mixture model, the log-likelihood of the sample $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ takes the form

$$L_K(\boldsymbol{\theta}; \mathbf{x}) = \sum_{i=1}^{n} \log\big[\sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}_i; \boldsymbol{\theta}_k)\big].$$

The function $L_K(\boldsymbol{\theta}; \mathbf{x})$ cannot be used as a selection criterion which balances model fit and model complexity for choosing the number K of components in the mixture. The most popular methodology for the selection of K is to use a criterion (score function) that ensures the compromise between flexibility and over-fitting. In general, a score function that is explicitly composed of two components: a component that measures the goodness of fit of the model to the data (such as the log-likelihood value, the complete-data log-likelihood value, etc), and a penalty function that governs the model complexity. This yields an overall score function of the form of

$$\text{score(model)} = \text{error(model)} + \text{penalty(model)}$$

which will be minimized.

As in general, the complexity of a model is related to the number of its free parameters ν. Thus, the penalty function then involves the number of model parameters. Popular score functions for the selection of K are information criteria (IC) which take into account the log-likelihood value. An information criterion (IC) based selection process can be described as below.

Let $f_K = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}; \boldsymbol{\theta})$ be a FMM with component density function of form $f_k(\boldsymbol{x}; \boldsymbol{\theta})$ and suppose that $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is a data sample arising from a population with density $f_{K_0}(\boldsymbol{x}; \boldsymbol{\theta}_0) = \sum_{k=1}^{K_0} \pi_k f_k(\boldsymbol{x}; \boldsymbol{\theta})$, where $K_0 \in \mathcal{K} \subset \mathbb{N}$. For each $K \in \mathcal{K}$ we estimate the MLE $\hat{\boldsymbol{\theta}}_{K,n}$ and compute the IC for n observations and K components as

$$\mathrm{IC}(n, K) = -2L(\hat{\boldsymbol{\theta}}_{K,n}) + \eta(n, \hat{\boldsymbol{\theta}}_{K,n}),$$

where $L(\hat{\boldsymbol{\theta}}_{K,n})$ is the log-likelihood value, $\eta(n, \hat{\boldsymbol{\theta}}_{K,n})$ is the penalty function that depends on n and the number of free parameters ν. After that, an estimator for K_0 is selected by

$$\hat{K}_0 = \arg\min_{K \in \mathcal{K}} \mathrm{IC}(n, K).$$

In this section, we cite the most commonly used criteria for model selection in FMM. The widely used IC are defined as follows:

- Akaike Information Criterion (AIC) (Akaike, 1974)

$$\mathrm{AIC}(n, K) = -2L(\hat{\boldsymbol{\theta}}_{K,n}) + 2\nu(\hat{\boldsymbol{\theta}}_{K,n}).$$

- Bayesian Information Criterion (BIC) (Schwarz, 1978)

$$\mathrm{BIC}(n, K) = -2L(\hat{\boldsymbol{\theta}}_{K,n}) + \nu(\hat{\boldsymbol{\theta}}_{K,n}) \log(n).$$

- Approximate weight of evidence (AWE) (Banfield and Raftery, 1993)

$$\mathrm{AWE}(n, K) = -2L(\hat{\boldsymbol{\theta}}_{K,n}) + 2\nu(\hat{\boldsymbol{\theta}}_{K,n})(\frac{3}{2} + \log(n)).$$

- Integrated Classification Likelihood (ICL) (Biernacki et al., 2000)

$$\mathrm{ICL}(n, K) = -2L_c(\hat{\boldsymbol{\theta}}_{K,n}) + \nu(\hat{\boldsymbol{\theta}}_{K,n}) \log(n),$$

  where $L_c(\hat{\boldsymbol{\theta}}_{K,n})$ is the complete-data log-likelihood for the model.

A general review and an empirical comparison of these criteria are presented in (McLachlan and Peel., 2000, Chapter 6).

2.3 Mixture models for regression data

In applied statistics, tremendous number of applications deal with relating a random response variable $\boldsymbol{Y}$ to a set of explanatory variables or covariates $\boldsymbol{X}$ through a regression-type model. One can model the relationship between $\boldsymbol{Y}$ and $\boldsymbol{X}$ via the conditional density function of $\boldsymbol{Y}$ given $\boldsymbol{X} = \boldsymbol{x}$, say, $f(\boldsymbol{y}|\boldsymbol{x})$. Similar to the density estimation, regression analysis is commonly conducted via parametric models of the form of $f(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the parameter vector. The homogeneous case assumes the regression coefficients are the same for every observation data point $(\boldsymbol{X}_1, \boldsymbol{Y}_1), \ldots, (\boldsymbol{X}_n, \boldsymbol{Y}_n)$. However, this assumption is often inadequate since parameters may change for difference subgroups of observations. Such heterogeneous data can be modeled with a mixture model for regression, as studied namely in Chamroukhi (2010, 2016d,a).

As an alternative extension of FMMs for the regression data, we suppose there is a latent random variable Z with probability mass function $\mathbb{P}(Z = k) = \pi_k,\ k = 1 \ldots, K$ and $\sum_{k=1}^{K} \pi_k = 1$. Moreover, assuming that $\boldsymbol{Y}|\boldsymbol{X} = \boldsymbol{x}, Z = k$ has the density $p_k(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}_k)$, where $\boldsymbol{\theta}_k$ is the parameter vector of the kth subgroup, then we have the mixture of regression model

$$f(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_k(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}_k) \tag{2.23}$$

by the same argument as that of (2.2).
This model was first introduced in Quandt (1972), since he studied the market for housing starts by modeling the conditional density function of Y given $\boldsymbol{X} = \boldsymbol{x}$ by a mixture of univariate two component Gaussian linear regression models, i.e,

$$f(y|\boldsymbol{x};\boldsymbol{\theta}) = \pi\mathcal{N}(y;\boldsymbol{x}^T\boldsymbol{\beta}_1, \sigma_1^2) + (1-\pi)\mathcal{N}(y;\boldsymbol{x}^T\boldsymbol{\beta}_2, \sigma_2^2), \tag{2.24}$$

where $\mathcal{N}(.;\mu,\sigma^2)$ is the Gaussian density function with mean μ and variance σ^2.

In this thesis, $\boldsymbol{Y}$ is assumed to be a univariate random variable. The next subsection presents the univariate Gaussian linear mixtures of regression models and the EM algorithm for estimating parameters.

2.3.1 Mixture of linear regression models

Let $((\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n))$ be a random sample of n independent pairs $(\boldsymbol{X}_i, Y_i)$, $(i = 1, \ldots, n)$ where $Y_i \in \mathcal{Y} \subset \mathbb{R}$ is the ith response given some vector of predictors $\boldsymbol{X}_i \in \mathcal{X} \subset \mathbb{R}^p$. Let $\mathcal{D} = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n))$ be the observed data sample.

For mixture of linear regression models (MLR), we consider the conditional density of Y given $\boldsymbol{X} = \boldsymbol{x}$ defined by a mixture of K Gaussian densities

$$f(y|\boldsymbol{x};\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y;\beta_{k0} + \boldsymbol{x}^T\boldsymbol{\beta}_k, \sigma_k^2), \tag{2.25}$$

where the vector $\boldsymbol{\beta}_k = (\beta_{k1}, \ldots, \beta_{kp})^T \in \mathbb{R}^p$, and scalars β_{k0} and σ_k^2 are the regression coefficients, intercepts and variances of the kth component density, respectively. $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^T$ is the vector of mixing propositions.

2.3.2 MLE via the EM algorithm

The observed-data log-likelihood function for this model is given by

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k, \sigma_k^2). \tag{2.26}$$

The EM algorithm can be used to obtain the updated estimates of the parameter $\boldsymbol{\theta}$ as in case of a FMM described in Subsection 2.2.1 (see also Subsection 2.2.2). We can therefore use a set $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ of latent variables where $z_{ik} \in \{0, 1\}$ in which, for each data point, all of the elements $k = 1, \ldots, K$ are 0 except for a single value of 1 indicating which regression model of the mixture was responsible for generating that data point.

The complete-data log-likelihood then takes the form

$$L_c(\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} (\log \pi_k + \log \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k, \sigma_k^2)). \tag{2.27}$$

The EM algorithm starts with an initial parameter $\boldsymbol{\theta}^{[0]}$ and repeat the following steps until convergence:

E-step (on the $(q+1)$th iteration) This step consists of computing the expectation of the complete-data log-likelihood:

$$\begin{aligned} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{[q]}) &= \mathbb{E}[L_c(\boldsymbol{\theta}) | \mathbf{x}, \mathbf{y}; \boldsymbol{\theta}^{[q]}] \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{E}[Z_{ik} | \mathbf{x}, \mathbf{y}; \boldsymbol{\theta}^{[q]}] (\log \pi_k + \log \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k, \sigma_k^2)) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{[q]} \log \pi_k + \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{[q]} \log \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k, \sigma_k^2), \end{aligned} \tag{2.28}$$

where

$$\begin{aligned} \tau_{ik}^{[q]} &= \mathbb{E}[Z_{ik} | \mathbf{x}, \mathbf{y}; \boldsymbol{\theta}^{[q]}] = \mathbb{P}[Z_{ik} = 1 | \mathbf{x}, \mathbf{y}; \boldsymbol{\theta}^{[q]}] \\ &= \frac{\pi_k^{[q]} \mathcal{N}(y_i; \beta_{k0}^{[q]} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k^{[q]}, \sigma_k^{[q]2})}{\sum_{l=1}^{K} \pi_l^{[q]} \mathcal{N}(y_i; \beta_{l0}^{[q]} + \boldsymbol{x}_i^T \boldsymbol{\beta}_l^{[q]}, \sigma_l^{[q]2})} \end{aligned} \tag{2.29}$$

is the posterior probability that $(\boldsymbol{x}_i, y_i)$ belongs to the kth component given the current parameter estimation $\boldsymbol{\theta}^{[q]}$, and $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, $\mathbf{y} = (y_1, \ldots, y_n)$.

M-step (on the $(q+1)$th iteration) In this step, the value of the parameter $\boldsymbol{\theta}$ is updated by computing the parameter $\boldsymbol{\theta}^{[q+1]}$ that maximizing the Q-function with respect to $\boldsymbol{\theta}$. The Q-function in (2.28) is decomposed as

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[q]}) = Q(\boldsymbol{\pi};\boldsymbol{\theta}^{[q]}) + \sum_{k=1}^{K} Q_k(\boldsymbol{\theta}_k;\boldsymbol{\theta}^{[q]}), \tag{2.30}$$

where

$$\begin{aligned} Q(\boldsymbol{\pi};\boldsymbol{\theta}^{[q]}) &= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}^{[q]} \log \pi_k, \\ Q_k(\boldsymbol{\theta}_k;\boldsymbol{\theta}^{[q]}) &= \sum_{i=1}^{n} \tau_{ik}^{[q]} \log \mathcal{N}(y_i;\beta_{k0}+\boldsymbol{x}_i^T\boldsymbol{\beta}_k,\sigma_k^2), \end{aligned} \tag{2.31}$$

and $\boldsymbol{\theta}_k = (\beta_{k0}, \boldsymbol{\beta}_k^T, \sigma_k^2)^T$ is the parameter vector of the kth component density function.
The maximization of $Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[q]})$ with respect to $\boldsymbol{\theta}$ is the performed by separately maximizing $Q(\boldsymbol{\pi};\boldsymbol{\theta}^{[q]})$ and $Q_k(\boldsymbol{\theta}_k;\boldsymbol{\theta}^{[q]})$ $(k = 1\ldots,K)$. Maximizing $Q(\boldsymbol{\pi};\boldsymbol{\theta}^{[q]})$ with respect to $\boldsymbol{\pi}$ subject to $\sum_{k=1}^{K}\pi_k = 1$ consists of constrained optimization problem, which is solved using Lagrange multipliers and leading to the updated estimate for π_k given by (2.12).
The maximization of $Q_k(\boldsymbol{\theta}_k;\boldsymbol{\theta}^{[q]})$ with respect to $(\beta_{k0}, \boldsymbol{\beta}_k^T)$ has the same form as weighted linear regression and the estimated regression coefficients in this case are given by

$$(\beta_{k0}^{[q+1]}, \boldsymbol{\beta}_k^{[q+1]T})^T = (\mathbf{X}^T\mathbf{R}_k^{[q]}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{R}_k^{[q]}\mathbf{y}, \tag{2.32}$$

where $\mathbf{R}_k^{[q]} = \mathrm{diag}(\tau_{1k}^{[q]},\ldots,\tau_{nk}^{[q]})$, $\mathbf{X}$ is the design matrix and $\mathbf{y}$ is the response vector.
Finally, the estimation of the variance σ_k^2 is given by maximizing

$$\sum_{i=1}^{n} \tau_{ik}^{[q]} \big[\log \frac{1}{\sqrt{2\pi}\sigma_k} - \frac{(y_i - \beta_{k0} - \boldsymbol{x}_i^T\boldsymbol{\beta}_k)^2}{2\sigma_k^2}\big]$$

with respect to σ_k^2 and takes the form

$$\sigma_k^{[q+1]2} = \sum_{i=1}^{n} \tau_{ik}^{[q]}(y_i - \beta_{k0}^{[q+1]} - \boldsymbol{x}_i^T\beta_k^{[q+1]})^2 \Big/ \sum_{i=1}^{n} \tau_{ik}^{[q]}, \qquad \text{(for heteroscedastic case)} \tag{2.33}$$

and

$$\sigma^{[q+1]2} = \frac{1}{n}\sum_{k=1}^{K}\sum_{i=1}^{n} \tau_{ik}^{[q]}(y_i - \beta_{k0}^{[q+1]} - \boldsymbol{x}_i^T\beta_k^{[q+1]})^2. \qquad \text{(for homoscedastic case)} \tag{2.34}$$

Example 2.3.1. *In this example, a simulation study was performed whereupon the data was*

generated from a two component Gaussian MLR with one covariate, with the form

$$f(y|x;\boldsymbol{\theta}) = \pi_1\mathcal{N}(y;\beta_{10}+\beta_{11}x,\sigma^2) + \pi_2\mathcal{N}(y;\beta_{20}+\beta_{21}x,\sigma^2), \tag{2.35}$$

where $\boldsymbol{\theta} = (\pi_1, \beta_{10}, \beta_{11}, \beta_{20}, \beta_{21}, \sigma^2)^T$*, and* x *is generated from a uniform distribution over* $[0, 1]$*. Here,* 300 *data points were generated with* $\pi_1 = 2/3$*,* $\beta_{10} = -1$*,* $\beta_{11} = 1.5$*,* $\beta_{20} = 0.5$*,* $\beta_{21} = -1$ *and* $\sigma = 0.1$*. Using the EM algorithm, the estimated parameters are* $\hat{\pi}_1 = 0.6664539$*,* $\hat{\beta}_{10} = -1.008368$*,* $\hat{\beta}_{11} = 1.523120$*,* $\hat{\beta}_{20} = 0.4912459$*,* $\hat{\beta}_{21} = -1.0036146$ *and* $\hat{\sigma} = 0.09725428$*. The log-likelihood is* $L(\hat{\boldsymbol{\theta}}) = 107.87553$*. Visualizations of the simulation scenario is provided in Figure 2.2.*

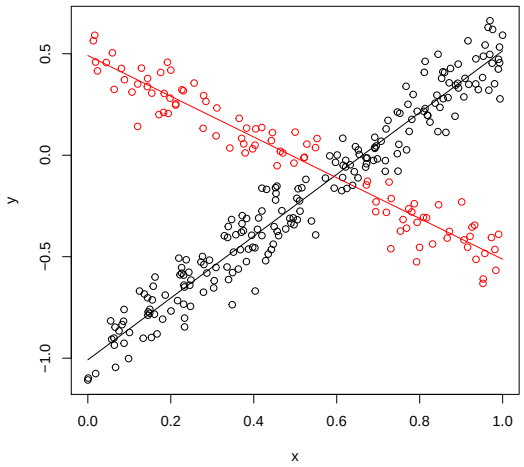


Figure 2.2: Illustration of the MLR on two-component MLR model. The black and red lines are the fitted conditional expectations of the density components 1 and 2, respectively.

2.3.3 Mixtures of experts

In a more general context, it is useful to consider that the latent variable Z also has a conditional relationship with some covariate vector $\mathbf{R} \in \mathbb{R}^p$. Commonly, $\mathbf{R}$ is also the explanatory variables vector, i.e, $\mathbf{R} = \mathbf{X}$. Jacobs et al. (1991) modeled the conditional probability of $Z = k|\mathbf{X} = \boldsymbol{x}$ via the logistic function (also called *softmax* function) and named these models as mixtures of experts.

The mixture of experts (MoE) model assumes that the observed pairs $(\boldsymbol{x}, y)$ are generated from $K \in \mathbb{N}$ (possibly unknown) tailored probability density components (the experts), governed by a hidden categorical random variable $Z \in [K] = \{1, \ldots, K\}$ that indicates the component from which a particular observed pair is drawn. The latter represents the gating network. Formally, the MoE decomposes the probability density of the observed data as a convex sum of a finite experts weighted by the gating network (typically a softmax function), and defined by

the following semi-parametric probability density (or mass) function:

$$p(y|\boldsymbol{x};\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x};\mathbf{w}) p_k(y|\boldsymbol{x};\boldsymbol{\theta}_k) \tag{2.36}$$

that is parameterized by the parameter vector defined by $\boldsymbol{\theta} = (\mathbf{w}^T, \boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_K^T)^T$, where $\mathbf{w}^T = (\mathbf{w}_1^T, \ldots, \mathbf{w}_{K-1}^T)$, and $\boldsymbol{\theta}_k$ $(k = 1, \ldots, K)$ is the parameter vector of the kth expert.

The gating network is defined by the distribution of the hidden variable Z given the predictor $\boldsymbol{x}$, i.e., $\pi_k(\boldsymbol{x};\mathbf{w}) = \mathbb{P}(Z = k|\boldsymbol{X} = \boldsymbol{x};\mathbf{w})$, which is in general given by gating softmax function of the form:

$$\pi_k(\boldsymbol{x};\mathbf{w}) = \mathbb{P}(Z = k|\boldsymbol{X} = \boldsymbol{x};\mathbf{w}) = \frac{\exp(w_{k0} + \boldsymbol{x}^T\boldsymbol{w}_k)}{1 + \sum_{l=1}^{K-1} \exp(w_{l0} + \boldsymbol{x}^T\boldsymbol{w}_l)} \tag{2.37}$$

for $k = 1, \ldots, K-1$ with $\mathbf{w}_k^T = (w_{k0}, \boldsymbol{w}_k^T) \in \mathbb{R}^{p+1}$ and $\mathbf{w}_K^T = (w_{K0}, \boldsymbol{w}_K^T) = \mathbf{0}$ for identifiability (Jiang and Tanner, 1999a). The experts network is defined by the conditional distribution $p_k(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_k)$ which is the short notation of $p(y_i|\boldsymbol{X} = \boldsymbol{x}, Z_i = k;\boldsymbol{\theta})$. The experts are chosen to sufficiently represent the data for each group k. Tailored regressors explaining the response y by the predictor $\boldsymbol{x}$ for continuous data, or multinomial experts for discrete data. For example, non-Normal MoE models (Chamroukhi, 2015), including MoE for non-symmetric data (Chamroukhi, 2016c) and robust MoE (Chamroukhi, 2017, 2016b; Nguyen and McLachlan, 2016), with public software (e.g see Chamroukhi et al. (2019a)), have been introduced. To avoid singularities and degeneracies of the MLE as highlighted namely in Stephens and Phil (1997); Fraley and Raftery (2007) one can regularize the likelihood through a prior distribution over the model parameter space. For a complete account of MoE, the types of gating networks and experts networks, reader can refere to Nguyen and Chamroukhi (2018). A comprehensive survey of the MoEs can be found in Yuksel et al. (2012).

Let $Y_1, \ldots, Y_n$ be an independent random sample, with corresponding covariates $\boldsymbol{x}_i$ $(i = 1, \ldots, n)$, arising from a distribution with density of form (2.36), and let y_i be the observed data of Y_i $(i = 1, \ldots, n)$. The generative process of the data assumes the following hierarchical representation. Given the predictor $\boldsymbol{x}_i$, the categorical variable Z_i follows the multinomial distribution:

$$Z_i|\boldsymbol{x}_i \sim \text{Mult}(1; \pi_1(\boldsymbol{x}_i;\mathbf{w}), \ldots, \pi_K(\boldsymbol{x}_i;\mathbf{w})) \tag{2.38}$$

where each of the probabilities $\pi_{z_i}(\boldsymbol{x}_i;\mathbf{w})$ is given by the multinomial logistic function (2.37). Then, conditioning on the hidden variable $Z_i = z_i$, given the covariate $\boldsymbol{x}_i$, a random variable Y_i is assumed to be generated according to the following representation

$$Y_i|Z_i = z_i, \boldsymbol{X}_i = \boldsymbol{x}_i \sim p_{z_i}(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_{z_i}) \tag{2.39}$$

where $p_{z_i}(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_{z_i}) = p(y_i|Z_i = z_i, \boldsymbol{X}_i = \boldsymbol{x}_i;\boldsymbol{\theta}_{z_i})$ is the probability density or the probability

mass function of the expert z_i depending on the nature of the data $(\boldsymbol{x}_i, y_i)$ within the group z_i.

A common choice to model the relationship between the input $\boldsymbol{x}$ and the output Y is by considering regression functions. Within each homogeneous group $Z_i = z_i$, the response Y_i, given the expert k, is modeled by the noisy linear model: $Y_i = \beta_{z_i 0} + \boldsymbol{\beta}_{z_i}^T \boldsymbol{x}_i + \sigma_{z_i}\varepsilon_i$, where the ε_i are standard i.i.d zero-mean unit variance Gaussian noise variables, the bias coefficient $\boldsymbol{\beta}_{k0} \in \mathbb{R}$ and $\boldsymbol{\beta}_k \in \mathbb{R}^p$ are the usual unknown regression coefficients describing the expert $Z_i = k$, and $\sigma_k > 0$ corresponds to the standard deviation of the noise. In such a case, the generative model (2.39) of Y becomes

$$Y_i | Z_i = z_i, \boldsymbol{x}_i \sim \mathcal{N}(.; \beta_{z_i 0} + \boldsymbol{\beta}_{z_i}^T \boldsymbol{x}_i, \sigma_{z_i}^2)\cdot \tag{2.40}$$

Assuming that, conditionally to the $\boldsymbol{x}_i$s, the Y_is are independent distributed with densities $f(y_i|\boldsymbol{x}_i; \boldsymbol{\theta})$, respectively, in which each of these densities is a MoE of K Gaussian densities,

$$f(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}_i; \mathbf{w}) \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k, \sigma_k^2),$$

where the parameter vector of the kth expert is $\boldsymbol{\theta}_k = (\beta_{k0}, \boldsymbol{\beta}_k^T, \sigma_k^2)^T$ $(k = 1, \ldots, K)$.
In the considered model for Gaussian regression, the incomplete-data log-likelihood function is given by

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log\Big[\sum_{k=1}^{K} \pi_k(\boldsymbol{x}_i; \mathbf{w}) \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{\beta}_k^T \boldsymbol{x}_i, \sigma_k^2)\Big], \tag{2.41}$$

and the complete-data log-likelihood has the form of

$$L_c(\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log\Big[\pi_k(\boldsymbol{x}_i; \mathbf{w}) \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{\beta}_k^T \boldsymbol{x}_i, \sigma_k^2)\Big]. \tag{2.42}$$

The EM algorithm for maximizing the log-likelihood function (2.41) can be described as follows (see Jacobs et al. (1991); Yuksel et al. (2012); Nguyen and Chamroukhi (2018)):

E-step This step requires computing the expectation of the complete-data log-likelihood under the parameter vector $\boldsymbol{\theta}^{[q]}$. This is simply equivalence to computing the posterior probability that $(\boldsymbol{x}_i, y_i)$ belongs to the kth expert given the current parameter $\boldsymbol{\theta}^{[q]}$. The $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{[q]})$ function has the form of

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{[q]}) = Q(\mathbf{w}; \boldsymbol{\theta}^{[q]}) + \sum_{k=1}^{K} Q_k(\boldsymbol{\theta}_k; \boldsymbol{\theta}^{[q]}), \tag{2.43}$$

where

$$Q(\mathbf{w}; \boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{[q]} \log \pi_k(\boldsymbol{x}_i; \mathbf{w}), \tag{2.44}$$

$$Q_k(\boldsymbol{\theta}_k; \boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n} \tau_{ik}^{[q]} \log \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k, \sigma_k^2), \tag{2.45}$$

with

$$\begin{aligned}\tau_{ik}^{[q]} &= \mathbb{E}[Z_{ik}|\mathbf{x},\mathbf{y};\boldsymbol{\theta}^{[q]}] \\ &= \frac{\pi_k(\boldsymbol{x}_i;\mathbf{w}^{[q]})\mathcal{N}(y_i;\beta_{k0}^{[q]}+\boldsymbol{x}_i^T\boldsymbol{\beta}_k^{[q]},\sigma_k^{[q]2})}{\sum_{l=1}^{K}\pi_l(\boldsymbol{x}_i;\mathbf{w}^{[q]})\mathcal{N}(y_i;\beta_{l0}^{[q]}+\boldsymbol{x}_i^T\boldsymbol{\beta}_l^{[q]},\sigma_l^{[q]2})},\end{aligned} \tag{2.46}$$

and $\pi_k(\boldsymbol{x}_i;\mathbf{w}^{[q]})$ is defined by (2.37).

M-step This step updates $\boldsymbol{\theta}$ with $\boldsymbol{\theta}^{[q+1]}$ by maximizing the Q-function can be done separately by maximizing $Q(\mathbf{w};\boldsymbol{\theta}^{[q]})$ and $Q_k(\boldsymbol{\theta}_k;\boldsymbol{\theta}^{[q]})$. For the experts, the formula (2.32) can be used to update $(\beta_{k0},\boldsymbol{\beta}_k)$ and the formulas (2.33), (2.34) are used to update σ_k^2, σ^2, respectively.

Updating the parameters for the gating network requires to maximize $Q(\mathbf{w};\boldsymbol{\theta}^{[q]})$. This function is concave and cannot be maximized in a closed form. The Newton-Raphson algorithm is generally used to perform the maximization as well as other gradient-based techniques (see Minka (2001)). Here, we consider the use of the Newton-Raphson algorithm to maximize $Q(\mathbf{w};\boldsymbol{\theta}^{[q]})$ with respect to $\mathbf{w}$. The Newton-Raphson algorithm is an iterative procedure, which consists of starting with an initial arbitrary solution $\mathbf{w}^{(0)}=\mathbf{w}^{[q]}$, and updating the estimation of $\mathbf{w}$ until the convergence criterion is reached. A single update is given by

$$\mathbf{w}^{(s+1)} = \mathbf{w}^{(s)} - [\nabla^2 Q(\mathbf{w}^{(s)};\boldsymbol{\theta}^{[q]})]^{-1}\nabla Q(\mathbf{w}^{(s)};\boldsymbol{\theta}^{[q]}), \tag{2.47}$$

where $\nabla^2 Q(\mathbf{w}^{(s)};\boldsymbol{\theta}^{[q]})$ is the Hessian matrix of $Q(\mathbf{w};\boldsymbol{\theta}^{[q]})$ with respect to $\mathbf{w}^{(s)}$ and $\nabla Q(\mathbf{w}^{(s)};\boldsymbol{\theta}^{[q]})$ is the gradient vector of $Q(\mathbf{w};\boldsymbol{\theta}^{[q]})$ at $\mathbf{w}^{(s)}$. For the closed form formulas of the Hessian matrix and the gradient vector one can refer to Chamroukhi (2010). The main advantage of the Newton-Raphson lies in the fact that it is quadratic convergence (Boyd and Vandenberghe (2004)).

Zeevi et al. (1998) showed that under some regularity conditions, then for any target function $f(\boldsymbol{x})$ belongs to the Sobolev class, $f\in W_p^r(L)$

$$W_p^r(L) \triangleq \left\{g(\boldsymbol{x}) : \|g\|_{W_p^r} = \sum_{|\alpha|\leq r}\|g^{(\alpha)}(\boldsymbol{x})\|_p \leq L, \boldsymbol{x}\in I^d\right\}$$

where

$$g^{(\alpha)} \equiv \frac{\partial^{\|\alpha\|_1} g}{\partial x_1^{\alpha_1} x_2^{\alpha_2}\ldots x_d^{\alpha_d}},$$

$\alpha\in\mathbb{N}^d$ and $I^d=[-1,1]^d$, then

$$\sup_{f\in W_p^r}\inf_{q_K\in\mathcal{Q}_K}\|f(\boldsymbol{x})-q_K(\boldsymbol{x})\|_{L_p(I^d,\lambda)} \leq \frac{c}{K^{r/d}},\ 1\leq p\leq\infty \tag{2.48}$$

with c is an absolute constant, the $L_p(I^d,\lambda)$ norm is defined as $\|f\|_{L_p(I^d,\lambda)}\equiv\left(\int_{I^d}|f|^p d\lambda\right)^{1/p}$ and

the manifold $\mathcal{Q}_K$ is given by

$$\mathcal{Q}_K \triangleq \Big\{ q(\boldsymbol{x}) : q(\boldsymbol{x}) = \frac{\sum_{k=1}^{K} c_k \sigma(\beta_{k0} + \boldsymbol{x}^T \boldsymbol{\beta}_k)}{\sum_{k=1}^{K} \sigma(\beta_{k0} + \boldsymbol{x}^T \boldsymbol{\beta}_k)},\ c_k, \beta_{k0} \in \mathbb{R}, \boldsymbol{\beta}_k \in \mathbb{R}^d \Big\}.$$

The ridge function $\sigma(.)$ is chosen to satisfy Assumption 2 of Zeevi et al. (1998). By choosing the exponent ridge function $\sigma(t) = e^t$, then the manifold $\mathcal{Q}_K$ is of the form

$$\mathcal{Q}_K \triangleq \Big\{ q(\boldsymbol{x}) : q(\boldsymbol{x}) = \sum_{k=1}^{K} \frac{\exp(\beta_{k0} + \boldsymbol{x}^T \boldsymbol{\beta}_k)}{\sum_{k=1}^{K} \exp(\beta_{k0} + \boldsymbol{x}^T \boldsymbol{\beta}_k)} c_k,\ c_k, \beta_{k0} \in \mathbb{R}, \beta_k \in \mathbb{R}^d \Big\}.$$

This shows that as $K \to \infty$ and under regularity conditions, for a given function f, there exists a mean function $q(\boldsymbol{x})$ of a MoE model that can approximate f, arbitrarily closely. Here,

$$q(\boldsymbol{x}) = \mathbb{E}[Y = y | \boldsymbol{X} = \boldsymbol{x}; \boldsymbol{\theta}] = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}; \boldsymbol{w})(\beta_{k0} + \boldsymbol{x}^T \boldsymbol{\beta}_k). \tag{2.49}$$

2.3.4 Mixture of generalized linear models

Until now, we have consider the cases where the component density functions of both MLR and MoE are the Gaussian densities. It is worth to consider the cases that these component density functions are described as generalized linear models (GLMs). GLMs were first introduced in Nelder and Wedderburn (1972) as a method for unifying various disparate regression methods, such as Gaussian, Poisson, logistic, binomial and gamma regressions. They consider various univariate cases, where the expectation of the response variable Y given the covariate $\boldsymbol{x}$ can be expressed as a function of a linear combination of $\boldsymbol{x}$

$$\mathbb{E}(Y | \boldsymbol{X} = \boldsymbol{x}) = h^{-1}(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta}), \tag{2.50}$$

$h(.)$ is the univariate invertible link function. In Table 2.1, we present characteristics of some common univariate GLMs which were considered in Nelder and Wedderburn (1972). For more details regarding to GLMs, see McCullagh (2018).

Model	dom(Y)	$h^{-1}(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta})$	$f(y)$	$\mathbb{E}(Y \vert \boldsymbol{X} = \boldsymbol{x})$
Gaussian	$\mathbb{R}$	$\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta}$	$\mathcal{N}(y; \mu, \sigma^2)$	μ
Binomial	$\{0, \ldots, N\}$	$\frac{N \exp(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta})}{1 + \exp(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta})}$	$\binom{N}{y} p^y (1-p)^{N-y}$	Np
Gamma	$(0, \infty)$	$-1/(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta})$	$\Gamma(y; a, b)$	a/b
Poisson	$\mathbb{N}$	$\exp(\beta_0 + \boldsymbol{x}^T \boldsymbol{\beta})$	$\frac{\exp(-\lambda)\lambda^y}{y!}$	λ

Table 2.1: Characteristics of some common univariate GLMs.

Consider the mixtures of GLMs. These models can be expressed as follows

$$f(y|\boldsymbol{x};\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k f_k(y; g(\beta_{k0} + \boldsymbol{x}^T\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k), \boldsymbol{\varphi}_k) \tag{2.51}$$

where $f_k(.)$ belongs to the class of GLMs, and $\boldsymbol{\theta}_k = (\beta_{k0}, \boldsymbol{\beta}_k^T, \boldsymbol{\varphi}_k^T)^T$ is the parameter vector of the kth component density. The characterizations of $f_k(.)$, $g(.)$ and $\boldsymbol{\varphi}_k$ for common GLMs are presented in Table 2.2.

Model	dom(Y)	$\boldsymbol{\varphi}_k$	$g(\beta_{k0} + \boldsymbol{x}^T\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k)$	$f_k(y; g(\beta_{k0} + \boldsymbol{x}^T\boldsymbol{\beta}_k, \boldsymbol{\varphi}_k), \boldsymbol{\varphi}_k)$
Gaussian	$\mathbb{R}$	σ_k^2	$\beta_{k0} + \boldsymbol{x}^T\boldsymbol{\beta}_k$	$\mathcal{N}(y; \beta_{k0} + \boldsymbol{x}^T\boldsymbol{\beta}_k, \sigma_k^2)$
Binomial	$\{0, \ldots, N\}$	None	$\frac{\exp(\beta_0+\boldsymbol{x}^T\boldsymbol{\beta})}{1+\exp(\beta_0+\boldsymbol{x}^T\boldsymbol{\beta})}$	$\binom{N}{y}\left[\frac{\exp(\beta_0+\boldsymbol{x}^T\boldsymbol{\beta})}{1+\exp(\beta_0+\boldsymbol{x}^T\boldsymbol{\beta})}\right]^y\left[\frac{1}{1+\exp(\beta_0+\boldsymbol{x}^T\boldsymbol{\beta})}\right]^{N-y}$
Gamma	$(0, \infty)$	b_k	$-\frac{b_k}{\beta_{k0}+\boldsymbol{x}^T\boldsymbol{\beta}_k}$	$\Gamma(y; -\frac{b_k}{\beta_{k0}+\boldsymbol{x}^T\boldsymbol{\beta}_k}, b_k)$
Poisson	$\mathbb{N}$	None	$\exp(\beta_0 + \boldsymbol{x}^T\boldsymbol{\beta})$	$\frac{\exp(-\exp(\beta_0+\boldsymbol{x}^T\boldsymbol{\beta}))\exp((\beta_0+\boldsymbol{x}^T\boldsymbol{\beta})y)}{y!}$

Table 2.2: Parameter vectors of some common univariate GLMs.

The fitting of mixtures of GLMs has been considered by Jansen (1993), Wedel and DeSarbo (1995) via EM algorithms. Along with the original GLMs, Wedel and DeSarbo (1995) also reported the inverse Gaussian regression models as an alternative extension to the gamma model. Other interesting works are (Aitkin, 1996, 1999). For the multivariate version of mixture of GLMs, it is interesting to consider the work of Oskrochi and Davies (1997).

Asides from the mixtures of GLMs, the MoEs based on GLMs have also been widely mentioned in literature. For example, (Jiang and Tanner, 1999a,b,c, 2000) considered the use of GLM based MoEs from various perspectives. An R package $flexmix$ for estimating the parameters of the MoEs based on GLMs is also introduced in (Grün and Leisch, 2007, 2008). Jiang and Tanner (1999a) established the approximation error when using the MoEs based on GLMs. Actually, their theorem can be interpreted as below:
Let $\Omega = [0,1]^p$, the space of the predict $\boldsymbol{x}$. Let $\mathcal{Y} \subseteq \mathbb{R}$ be the space of the response y. Let $(\mathcal{Y}, \mathcal{F}_Y, \lambda)$ be a general measurable space, $(\Omega, \mathcal{F}_\Omega, \kappa)$ be a probability space such that κ has a positive continuous density with respect to the Lebesgue measure on Ω and $(\Omega\times\mathcal{Y}, \mathcal{F}_\Omega\otimes\mathcal{F}_Y, \kappa\otimes\lambda)$ be the product measure space. Consider a random predictor-response pair $(\boldsymbol{X}, Y)$. Suppose that $\boldsymbol{X}$ has a probability measure κ, and $(\boldsymbol{X}, Y)$ has a probability density function (pdf) $\varphi(.,.)$ with respect to $\kappa \otimes \lambda$, where φ is of the form

$$\varphi(\boldsymbol{x}, y) = \pi(h(\boldsymbol{x}), y).$$

Here,

$$\pi(.,.) : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$$

has the one-parameter exponential form

$$\pi(h, y) = \exp\big\{a(h)y + b(h) + c(y)\big\}, \text{ for } y \in \mathcal{Y}, \tag{2.52}$$

such that $\int_{\mathcal{Y}} \pi(h, y)d\lambda(y) = 1$ for each $h \in \mathbb{R}$. $a(.)$ and $b(.)$ are analytic and have nonzero derivatives on $\mathbb{R}$ and $c(.)$ is $\mathcal{F}_Y$-measurable. Assume that $h \in W_\infty^2(K_0)$, where $W_\infty^2(K_0)$ is a ball with radius K_0 in a Sobolev space with sup-norm and second-order continuous differentiability. Denote the set of all probability density functions $\varphi(.,.) = \pi(h(.),.)$ as $\mathbf{\Phi}$.

Consider an approximator f in the MoE family is assumed to have the following form

$$f = f_K(\boldsymbol{x}, y; \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}; \boldsymbol{w})\pi(h_k(\boldsymbol{x}), y), \tag{2.53}$$

where $h_k(\boldsymbol{x}) = \beta_{k0} + \boldsymbol{x}^T\boldsymbol{\beta}_k$ and $\pi(.,.)$ has the form of (2.52). Then, under some conditions Jiang and Tanner (1999a) proved the following result:

Theorem 2.3.1. *(Approximation rate in KL divergence)*

$$\sup_{\varphi \in \mathbf{\Phi}} \inf_{f_K \in \mathbf{\Pi}_{K,\mathcal{S}}} KL(f_K, \varphi) \leq \frac{c}{K^{4/p}},$$

where c is a constant independent of p, $\mathbf{\Pi}_{K,\mathcal{S}}$ is a restriction on the set of the parameters of f_K, imposed by Jiang and Tanner (1999a) and $KL(f_K, \varphi)$ is the Kullback-Leibler divergence between f_K and φ defined by

$$KL(f_K, \varphi) = \int \int_{\Omega \times \mathcal{Y}} f_K(\boldsymbol{x}, y) \log\Big\{\frac{f_K(\boldsymbol{x}, y)}{\varphi(\boldsymbol{x}, y)}\Big\} d\kappa(\boldsymbol{x})d\lambda(y).$$

2.3.5 Clustering with FMMs

One of the most common applications of FMMs is to cluster heterogeneous data, i.e the so-called model-based clustering approach. Suppose that we have an observation $\boldsymbol{X}$ from a FMM of form (2.4). One can consider each of the component densities $f_k(\boldsymbol{x}; \boldsymbol{\theta}_k)$ as a subpopulation density of the overall population defined by density (2.4). Hence, it is reasonable to think about the probability that $\boldsymbol{X}$ belongs to one of the K densities as clustering criterion.

Using the characterization of a FMM, we can consider the label $Z = k$ as the true cluster that the observation $\boldsymbol{X}$ belongs to (i.e. $\boldsymbol{X}$ is generated from the kth component density $f_k(\boldsymbol{x}; \boldsymbol{\theta}_k)$). In addition, we also can use the Bayes' rule to compute the posterior probability that $Z = k$ given the observation of Y (see also (2.9))

$$\begin{aligned} \mathbb{P}(Z = k|\boldsymbol{X} = \boldsymbol{x}) &= \frac{\pi_k f_k(\boldsymbol{x}; \boldsymbol{\theta}_k)}{\sum_{l=1}^{K} \pi_l f_l(\boldsymbol{x}; \boldsymbol{\theta}_l)}, (k = 1, \ldots, K). \\ &= \tau_k(\boldsymbol{x}) \end{aligned} \tag{2.54}$$

A hard-clustering of $\boldsymbol{X}$ is given using the posterior probability (2.54) via the Bayes' assignment rule

$$\hat{z} = \arg\max_k \tau_k(\boldsymbol{x}). \tag{2.55}$$

$\hat{z}$ is considered as the cluster that $\boldsymbol{X}$ is generated from. For more details on the properties of the Bayes' rule see Wasserman (2013).

Now, assume that $\mathbf{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is a sample generated form a parametric K component FMM of form (2.4) with the parameter vector $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_{K-1}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$. Let $\hat{\boldsymbol{\theta}}$ denotes the estimated parameter vector for $\boldsymbol{\theta}$ ($\hat{\boldsymbol{\theta}}$ can be estimated via EM algorithm). Hence, the clustering task via FMMs can be described as follows:

i) For each observation $\boldsymbol{x}_i$, compute the posterior probabilities $\tau_{ik} \equiv \tau_k(\boldsymbol{x}_i; \hat{\boldsymbol{\theta}})$ via (2.9).

ii) Use the Bayes' rule (2.55) to determine the cluster label of $\boldsymbol{x}_i$.

In GMMs framework, the equivalent form of (2.9) is given by (2.19). Similar strategies can be used to cluster the data for MLR models and MoE models. In these settings, one just need to replace the formula (2.9) with (2.29) and (2.46), respectively, to compute the posterior probabilities.

2.4 Clustering and classification in high-dimensional setting

In this part, we focus on some recent methods for model-based clustering in high-dimensional setting. The classical methods basically can be split into three families: dimensionality reduction, regularization and constrained and parsimonious models see for example Bouveyron and Brunet-Saumard (2014) for a review. Aside with these methods, recent research also provides some interesting approaches to deal with high-dimensional data such as subspace clustering methods and feature selection methods. A helpful text for these methods is Bouveyron and Brunet-Saumard (2014).

2.4.1 Classical methods

Dimensionality reduction approaches

For dimensionality reduction methods, one assumes that his data set with the number p of measured variables is too large and, implicitly, there is a small subspace of order $d \ll p$ contains most of his data. In a Gaussian mixture scenario, once the data is projected in this subspace, it is possible to apply the EM algorithm on the projected observations to obtain a partition of the original data (if d is small enough).

One of the most popular linear methods used for dimensionality reduction is the principal component analysis (PCA), which was introduction by Pearson (1901). Pearson described it as a linear projection that minimizes the average projection cost. Tipping and Bishop (1999) provided a probabilistic view of PCA by assuming the observed variables $\boldsymbol{X} \in \mathbb{R}^p$, are conditionally

independent given the values of the latent (or unobserved) variables $\boldsymbol{T} \in \mathbb{R}^d$. The relationship between $\boldsymbol{X}$ and $\boldsymbol{T}$ is of the linear form

$$\boldsymbol{X} = \Lambda \boldsymbol{T} + \boldsymbol{\mu} + \varepsilon. \tag{2.56}$$

The $p \times d$ matrix Λ relates the two sets of variable, while the parameter $\boldsymbol{\mu}$ permits the model to have non-zero mean, $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_p)$. Hence, the $\boldsymbol{T}$ conditional probability distribution over $\boldsymbol{X}$-space is given by

$$\boldsymbol{X}|\boldsymbol{T} \sim \mathcal{N}(\boldsymbol{\mu} + \Lambda \boldsymbol{T}, \sigma^2 I_p). \tag{2.57}$$

Furthermore, if the marginal distribution over the latent variables $\boldsymbol{T}$ is Gaussian $\boldsymbol{T} \sim \mathcal{N}(\mathbf{0}, I_d)$, then the marginal distribution of $\boldsymbol{X}$ is also Gaussian $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Lambda\Lambda^T + \sigma^2 I_p)$. Estimates for Λ and $\boldsymbol{\mu}$ can also be obtained by iterating maximization of the log-likelihood function, such as by using EM algorithm.

One will then cluster the observations using $\boldsymbol{T}$ as the input data rather than $\boldsymbol{X}$. Ghosh and Chinnaiyan (2002) used this strategy to cluster microarray datasets. An EM algorithm for clustering the latent variable $\boldsymbol{T}$ can also be found in this work. A Bayesian version for this approach was investigated by Liu et al. (2003).

Factor analysis (FA) is another way to deal with dimensionality reduction. The only difference between PCA and FA lies in the fact that, with FA models, the distribution of ε in (2.56) with a Gaussian distribution given by

$$\varepsilon \sim \mathcal{N}(\mathbf{0}, \Psi),$$

where Ψ is a diagonal matrix, $\Psi = \text{diag}(\sigma_1^2, \ldots, \sigma_p^2)$. Hence, it is to reduce the dimensionality of the space and to keep the observed covariance structure of the data. In the same context, Tamayo et al. (2007) suggested decomposing the data matrix $\mathbf{X}$ using the nonnegative matrix factorization (Lee and Seung (1999)) to reduce the dimension before clustering the data.

However, all these approaches have a number of drawbacks. The resulting clustering is not sparse in the features, since each latent variable $T_h, h = 1, \ldots, d$ is a linear combination of the full set of p features. Moreover, there is no guarantee that the new feature T_h will contains the information that one is interested in detecting via clustering. In fact, Chang (1983) studied the effective of performing PCA to reduce the data dimension before clustering. He generated a data set from a mixture of two Gaussian distributions and found that using this procedure is not justified since the first component does not necessarily provide the best separation between subgroups.

Mixtures with regularization of the covariance matrix

As we have mentioned at the end of Section 2.2.1, it is possible to see the problem in clustering of high-dimensional data as a numerical problem in computing the matrix inversion of $\boldsymbol{\Sigma}_k$, which is

used to compute the posterior probabilities τ_{ik} and the log-likelihood function. From this point of view, a simple way to tackle this problem is to regularize the estimation of $\mathbf{\Sigma}_k$ before their inversion. This can be done by using the ridge regression, which adds a small positive quantity δ_k to the diagonal of $\hat{\mathbf{\Sigma}}_k$

$$\tilde{\mathbf{\Sigma}}_k = \hat{\mathbf{\Sigma}}_k + \delta_k I_p.$$

A general version was introduced by Hastie et al. (1995) while these authors studied the penalized discriminant analysis

$$\tilde{\mathbf{\Sigma}}_k = \hat{\mathbf{\Sigma}}_k + \delta\Omega,$$

where Ω is a $p \times p$ symmetric and nonnegative definite matrix. This approach is different from the previous one since it also penalizes the correlations between the predictors. Friedman (1989) proposed a compromise between linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA), which allows one to shrink the separate covariances of QDA to a common covariance as in LDA. The regularized covariance matrices have the form

$$\hat{\mathbf{\Sigma}}_k(\lambda) = \frac{\lambda(n_k - 1)\hat{\mathbf{\Sigma}}_k + (1-\lambda)(n-K)\hat{\mathbf{\Sigma}}}{\lambda(n_k - 1) + (1-\lambda)(n-K)}, \tag{2.58}$$

where $\hat{\mathbf{\Sigma}}$ is the pooled covariance matrix as used in LDA and $\hat{\mathbf{\Sigma}}_k$ is the covariance matrix of group kth used in QDA. $\lambda \in [0, 1]$ allows a continuum of models between LDA and QDA.

A similar strategy allows $\hat{\mathbf{\Sigma}}_k(\lambda)$ to be reduced to the scalar covariance. Combining these models leads to a more general family of covariances $\hat{\mathbf{\Sigma}}_k(\gamma, \lambda)$, indexed by a pair of parameters

$$\hat{\mathbf{\Sigma}}_k(\gamma, \lambda) = \gamma\hat{\mathbf{\Sigma}}_k(\lambda) + (1-\gamma)\frac{\mathrm{tr}[\hat{\mathbf{\Sigma}}_k(\lambda)]}{p} I_p,$$

for $\lambda \in [0, 1]$ and $\mathrm{tr}[\hat{\mathbf{\Sigma}}_k(\lambda)]/p$ is the average eigenvalue of $\hat{\mathbf{\Sigma}}_k(\lambda)$.

It is also possible to use the Moore-Penrose pseudo-inverse of $\hat{\mathbf{\Sigma}}$ instead of the usual inverse $\hat{\mathbf{\Sigma}}^{-1}$. For the generalized inverse of matrices and its applications, the reader can refer to Rao (1971). A comprehensive overview of regularization techniques in classification can be found in Mkhadri et al. (1997).

Constrained and parsimonious mixture models

In this section, we consider another way to tackle the curse of dimensionality by consider it as a problem of over-parameterized modeling. As we have discussed in Section 2.2.1, the GMM requires lots of parameters to infer the model in high-dimension setting. The use of constrained and parsimonious models is another approach to reduce the number of parameters in model-based clustering.

Constrained Gaussian mixture models. A classical method to reduce the number of free parameters of GMMs is to add some constrains on the model through their parameters. Mainly, we will

Model	Number of parameters	$K = 3, p = 50$
Full GMM	$(K-1) + Kp + Kp(p+1)/2$	3977
Homoscedastic GMM	$(K-1) + Kp + p(p+1)/2$	1427
Diag-GMM	$(K-1) + Kp + Kp$	302
Homoscedastic Diag-GMM	$(K-1) + Kp + p$	202

Table 2.3: Number of free parameters to estimate for constrained GMMs.

add some restrictions on the covariance matrices $\mathbf{\Sigma}_k$, which commonly requires $p(p+1)/2$ parameters to construct one of them. A full GMM with K components needs $Kp(p+1)/2$ real values for the covariance matrices. A simple way to reduce the parameter space is to constraint the K covariance matrices to be the same across all mixture components, i.e., using the homoscedastic case. It is also possible to assume that the variables are conditionally independent. This assumption implies that all the covariance matrices are diagonal, i.e., $\mathbf{\Sigma}_k = \text{diag}(\sigma_{k1}^2, \ldots, \sigma_{kp}^2)$ for all $k = 1, \ldots, K$, and the associated model (Diag-GMM) has a low number of free parameters. In addition, one can reduce the free parameters by assuming that $\sigma_{kj}^2 = \sigma_k^2, \ \forall j = 1, \ldots, p$. He can also consider the homoscedastic case for those models. Table 2.3 provides the number of parameters which are used for those constrained models. However, it is hard to find a real data set, in which the features are conditionally independent. Therefore, it restricts the range of applications of these models. Banfield and Raftery (1993) and Celeux and Govaert (1995) developed a different criteria that are more general than the constrained models. The key to their approach is a reparameterization of the covariance matrix $\mathbf{\Sigma}_k$ in term of its eigenvalue decomposition. Celeux and Govaert (1995) called them parsimonious Gaussian models.

Parsimonious Gaussian models. These models parametrize the covariance matrices from their eigenvalue decomposition (see (Bellman, 1960, Section 3.5))

$$\mathbf{\Sigma}_k = \lambda_k D_k A_k D_k^T, \tag{2.59}$$

where D_k is the matrix of eigenvectors determines the orientation of the group kth, A_k is a diagonal matrix with the eigenvalues of $\mathbf{\Sigma}_k$ on the diagonal that explains its shape, and the parameter λ_k determines the volume. By constraining the parameters λ_k, D_k and A_k, Celeux and Govaert (1995) enumerated 14 different models which are listed in Table A.1. The second column of Table A.1 corresponds to the name used by Raftery and Dean (2006). The parsimonious models propose a trade-off between the perfect modeling and what one can correctly estimate in practice. The reader can refer to Celeux and Govaert (1995) for more details on these models including parameter estimation methods. Model selection can be achieved using the Bayesian information criterion (BIC) Schwarz (1978).

The drawback of all the classical approaches lies in the fact that all the features are kept while clustering the data. Hence, subspace clustering methods are good alternatives to dimensionality reduction approaches. Recent works have focus on reduce data dimensionality by selecting

relevant variables for the clustering task. We take a survey on these works in next sections.

2.4.2 Subspace clustering methods

In this part, we focus on some subspace clustering methods in the point of view of model-based approach. These methods are mostly related to the factor analysis model assuming the observation variable is linked to a latent variable through a linear relationship. EM algorithms for maximum likelihood factor analysis can be given through the work of Rubin and Thayer (1982).

Mixture of factor analyzers

The idea of Mixture of factor analyzers (MFA) was introduced by Ghahramani and Hinton (1996) in which an EM algorithm was proposed for parameter estimation, and was first presented from the perspective of a method for model-based density estimation from high-dimensional data in McLachlan and Peel (2000). In PFA, for each cluster, the observed variable, $\boldsymbol{X} \in \mathbb{R}^p$ is linked with a latent variable $\boldsymbol{Y} \in \mathbb{R}^d$ $(d < p)$ through a linear relationship which are different from each cluster. The model can be described as follows:
Let $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be n independent observations of a random vector $\boldsymbol{X} \in \mathbb{R}^p$. Assume that $\boldsymbol{X}$ can be expressed from an unobserved random vector $\boldsymbol{Y} \in \mathbb{R}^d$ $(d < p)$, named the factor. Moreover, assume that there are n unobserved partition $\{z_1, \ldots, z_n\}$, which are assumed to be independent unobserved realizations of a random latent variable $Z \in \{1, \ldots, K\}$ where $z_i = k$ indicates that $\boldsymbol{x}_i$ belongs to the kth factor analyzer.

$$Z \sim \text{Mult}(1; \pi_1, \ldots, \pi_K),$$

where $\pi_k > 0$, for all k and $\sum_{k=1}^{K} \pi_k = 1$. The relationship between $\boldsymbol{X}$ and $\boldsymbol{Y}$ conditionally to Z is given by

$$\boldsymbol{X}|_{Z=k} = \boldsymbol{\mu}_k + \Lambda_k \boldsymbol{Y} + \varepsilon, \tag{2.60}$$

where $\boldsymbol{\mu}_k$ is the mean vector of the kth factor analyzer, Λ_k is a $p \times d$ matrix. $\varepsilon \in \mathbb{R}^p$ is assumed to be a center Gaussian distribution with a diagonal covariance matrix Ψ and is independent with $\boldsymbol{Y}$,

$$\varepsilon \sim \mathcal{N}(\boldsymbol{0}, \Psi).$$

Besides, as in regular factor analysis, the factors are all assumed to be $\mathcal{N}(\boldsymbol{0}, I_d)$ distributed, therefore,

$$\boldsymbol{Y}|_{Z=k} = \boldsymbol{Y} \sim \mathcal{N}(\boldsymbol{0}, I_d).$$

This implies that the conditional distribution of $\boldsymbol{X}$ given $\boldsymbol{Y}, Z$ is also Gaussian

$$\boldsymbol{X}|\boldsymbol{Y}, Z = k \sim \mathcal{N}(\boldsymbol{\mu}_k + \Lambda_k \boldsymbol{Y}, \Psi),$$

and

$$\boldsymbol{X}|Z = k \sim \mathcal{N}(\boldsymbol{\mu}_k, \Lambda_k\Lambda_k^T + \Psi).$$

The marginal distribution of $\boldsymbol{X}$ is thus a GMM

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} p(Z = k) f_k(\boldsymbol{x}|Z = k) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \Lambda_k\Lambda_k^T + \Psi). \tag{2.61}$$

For this model, we need $(K-1)$ parameters for the mixing proportions, Kp for the means $\{\boldsymbol{\mu}_k\}_{k=1}^K$. If d is chosen sufficiently smaller than p, then there are some constrains on the covariance matrices $\Lambda_k\Lambda_k^T + \Psi$ and thus reduces the number of free parameters to be estimated. In the case $d > 1$, there is an infinity of choices for Λ_k since one can replace Λ_k by $\Lambda_k C$, where C is an orthogonal matrix, i.e., $CC^T = I_d$. Hence, a way to specify Λ_k is to choose it in such that the $d \times d$ matrix

$$\Lambda_k^T \Psi \Lambda_k$$

is diagonal. Using this approach, Lawley and Maxwell (1962) showed that the number of free parameters for a regular factor analysis model (without the mean vector) is $p + pd - d(d-1)/2$. Therefore, $Kd(p-(d-1)/2) + p$ free parameters are required to estimate the component covariance matrices in equation (2.61). The model complexity is $(K-1) + Kp + Kd(p-(d-1)/2) + p$. For example with $K = 3$, $p = 50$ and $d = 5$, then 1672 parameters have to be estimated. An EM algorithm for parameter estimation is also given in Ghahramani and Hinton (1996).

This model was generalized later by McLachlan et al. (2003) who removed the constraint of the noise ε in (2.60). Hence,

$$\boldsymbol{X}|_{Z=k} = \boldsymbol{\mu}_k + \Lambda_k \boldsymbol{Y} + \varepsilon_k, \tag{2.62}$$

where $\varepsilon_k \sim \mathcal{N}(\boldsymbol{0}, \Psi_k)$, and $\Psi_k = \text{diag}(\sigma_{k1}^2, \ldots, \sigma_{kp}^2)$. In this case, the model complexity increases and takes the following value: $(K-1) + Kp + Kd(p-(d-1)/2) + Kp$. Regarding the model inference, McLachlan et al. (2003) proposed to make use of an alternating expectation-conditional maximization (AECM) algorithm (Meng and Van Dyk (1997)) to fit the mixture of factor analyzers by maximum likelihood. An interesting application of this model can be found in the work of McLachlan et al. (2002), where the authors used it to cluster gene expression microarray data.

To reduce the complexity of the MFA models, Baek et al. (2009) reparameterized the mixture model with restrictions on the means and covariance matrices, such that, for $k = 1, \ldots, K$

$$\boldsymbol{\mu}_k = A\boldsymbol{\rho}_k \tag{2.63}$$

and

$$\boldsymbol{\Sigma}_k = A\Omega_k A^T + D, \tag{2.64}$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are respectively the mean vector and covariance matrix of kth cluster, $\boldsymbol{\rho}_k$

is a d-dimensional vector, A is a $p \times d$ orthonormal matrix of loading on d unobserved factors ($A^T A = I_d$), Ω_k is a $d \times d$ positive definite symmetric matrix and D is a diagonal $p \times p$ matrix. The orthonormal condition on the matrix A is to make sure the solution $\hat{A}$ for A is unique up to postmultiplication by an orthogonal matrix (see (Baek et al., 2009, Appendix) for more details). This model is named as mixtures of common factor analyzers (MCFA) by its authors, who proof that in fact MCFA is a special case of the MFA approach. According to the MCFA assumptions, there are Kd means parameters to estimate instead of Kp in the MFA model. For the loading matrix A which is constrained to have orthonormal columns and to be common in all classes, then only $pd - d(d+1)/2$ parameters are required to estimate it. Moreover, p positive values are also required to estimate D. Finally, for the positive definite symmetric matrices Ω_k, one need to use $Kd(d+1)/2$ parameters. Hence, the complexity of the MCFA is $(K-1) + Kd + p + (pd - d(d+1)/2) + Kd(d+1)/2$. Typically, for a model with $K = 3$ classes, $p = 50$ and $d = 5$ then the complexity is equal to 302. The main advantage of the MCFA model is that the (estimated) posterior means of the factors corresponding to the observed data can be used to display the latter in low-dimensional spaces. The MCFA models can be fitted via the EM algorithm given by their authors (see (Baek et al., 2009, Appendix)).

Expanded parsimonious Gaussian mixture models

Inspired from MFA, McNicholas and Murphy (2010) proposed a family of models known as the expanded parsimonious Gaussian mixture models (EPGMM) family. Each component factor the covariance matrix $\mathbf{\Sigma}_k$ in (2.61) is given by

$$\mathbf{\Sigma}_k = \Lambda_k \Lambda_k^T + \Psi_k,$$

where Λ_k is the loading matrix and Ψ_k is the diagonal variance matrix for the noise term of the kth factor. The matrix Ψ_k can be further parameterized by writing

$$\Psi_k = \omega_k \Delta_k,$$

where $\omega_k \in \mathbb{R}$ and $\Delta_k = \text{diag}(\delta_{k1}, \ldots, \delta_{kp})$ such that $|\Delta_k| = 1$. Therefore, the covariance matrix $\mathbf{\Sigma}_k$ can be rewritten as

$$\mathbf{\Sigma}_k = \Lambda_k \Lambda_k^T + \omega_k \Delta_k.$$

By adding some constraints on the loading matrices Λ_k, the constant ω_k and the diagonal matrix Ψ_k, McNicholas and Murphy (2010) obtained twelve Gaussian mixture models named EPGMM family. Nomenclature and the corresponding covariance structure of the members of the EPGMM are given in Table A.2. Parameter estimation is also carried out using AECM algorithm. Model selection can be achieved using the BIC. It is worth to mention that their AECM algorithm could be used for inferring most of the MFA models.

Mixture of high-dimensional GMMs

Following the classical parsimonious GMMs, (Bouveyron et al., 2007a,b) constrained the GMM using the eigen-decomposition of the covariance matrix $\mathbf{\Sigma}_k$ of the kth group (see equation 2.59)

$$\mathbf{\Sigma}_k = Q_k \Lambda_k Q_k^T, \tag{2.65}$$

where Q_k is a $p \times p$ orthogonal matrix contains the eigenvectors of $\mathbf{\Sigma}_k$ and Λ_k is a diagonal matrix of the associated eigenvalue sorted in a decreasing order. The key idea of these works lies in fact that the authors reparametrize the matrix Λ_k, such that Λ_k has only $d_k + 1$ different eigenvalues

$$\Lambda_k = \text{diag}(a_{k1}, \ldots, a_{kd_k}, \underbrace{b_k, \ldots, b_k}_{p-d_k}),$$

where d_k first eigenvalues $a_{k1}, \ldots, a_{kd_k}$ parametrize the variance in the group-specific subspace and the $p - d_k$ last terms, the b_k's model the variance of the noise ($d_k < p$). The noise variance of each cluster k is isotropic and is contained in a subspace which is orthogonal to the subspace of the kth group. Bouveyron et al. (2007a) named this family the $[a_{kj} b_k Q_k d_k]$.

If the symmetric positive definite matrix $\mathbf{\Sigma}_k$ have p distinct eigenvalues then one can obtain the eigen-decomposition form (2.65). However, we now proof that if this symmetric positive definite matrix has only $d + 1$ different eigenvalues ($d < p$) with $p - d$ multiple characteristic roots then we still have the decomposition form (2.65).

Lemma 2.4.1. *Let $\mathbf{\Sigma}$ be a $p \times p$ symmetric positive definite matrix with $d+1$ different eigenvalues $\{\lambda_1, \ldots, \lambda_d, \gamma\}$ where γ has $p-d$ multiple characteristic roots. Then, there exits a $p \times p$ orthogonal matrix Q which contains the eigenvectors of $\mathbf{\Sigma}$ such that*

$$\mathbf{\Sigma} = Q \Lambda Q^T, \tag{2.66}$$

where $\Lambda = diag(\lambda_1, \ldots, \lambda_d, \underbrace{\gamma, \ldots, \gamma}_{p-d})$.

The proof of this Lemma is given in Appendix A.3.3.

The number of parameters required for the estimation of $\mathbf{\Sigma}$ (with the assumption that $\lambda_i > \gamma$ for all $i \in \{1, \ldots, d\}$) is $d(p - (d+1)/2) + d + 1$, where $d(p - (d+1)/2)$ and $(d+1)$ are the number of parameters required for the estimation of Q and $(\lambda_1, \ldots, \lambda_d, \gamma)$. This can be computed using the trick provided in Appendix A.3.4. Given some constraints on a_{kj}, b_k, Q_k and d_k, the authors proposed 28 GMMs. Among them, 16 models have the closed-form estimators. The covariance structure, number of covariance parameters for these models are given in Table A.3. The inference for 16 GMMs listed in Table A.3 can be done using the EM algorithm since update formula for each parameter is closed-form.

The subspace clustering methods posed in this section belong to a huge family of GMMs. Besides that, there exists several links between these models, which were listed in (Bouveyron and

Brunet-Saumard, 2014, Figure 4). These models in some sense can be described as a combination of the dimensionality reduction and parametric reduction methods. Some drawbacks can be found since all features still be used to construct the subspace through the loading matrices or through the component densities. In addition, the estimation procedures in some models can not be directly done using the EM algorithm because of the specific features of the latent subspace.

Among the related works, we may cite the discriminative latent mixture models, which were proposed by Bouveyron and Brunet (2012). The robust versions of the MFA models rely on t-distributions can be referred to Andrews and Mcnicholas (2011) and McLachlan et al. (2007).

2.4.3 Variable selection for clustering

In this section, we focus on some recent works which simultaneously cluster the data and select relevant variables for the clustering task. This task can be handle in two different ways. The first one tackles the problem of variable selection for model-based clustering as a model selection problem, while the other aims at introducing a penalty term in the log-likelihood function in order to obtain sparsity in the features.

Variable selection as a model selection problem

Law et al. (2004) considered the GMMs and assumed that the features are conditionally independent given the (hidden) component label, that is,

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}|\boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{p} p_j(x_j|\boldsymbol{\theta}_{kj}), \tag{2.67}$$

where $p_j(x_j|\boldsymbol{\theta}_{kj})$ is the pdf of the jth feature in the kth component. In case of GMMs, this condition is equivalent to adopting diagonal covariance matrices. In addition, for the role of each feature, they suggest that the jth feature is irrelevant if its distribution is independent of the class label, i.e., if it follows a common density, denoted by $q(x_j|\lambda_j)$. So if we denote $\Psi = (\psi_1, \ldots, \psi_p)$ be the set of binary variables, such that $\psi_j = 1$ if feature jth is relevant and $\psi_j = 0$ otherwise, then the mixture density in (2.67) can be rewritten as

$$p(\boldsymbol{x}|\Psi, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{p} [p_j(x_j|\boldsymbol{\theta}_{kj})]^{\psi_j} [q(x_j|\lambda_j)]^{1-\psi_j}. \tag{2.68}$$

Hence, the joint density of $(\boldsymbol{x}, \Psi)$ is

$$\begin{aligned} p(\boldsymbol{x}, \Psi; \boldsymbol{\theta}) &= p(\boldsymbol{x}|\Psi; \boldsymbol{\theta}) p(\Psi; \{\rho_j\}) \\ &= \Big(\sum_{k=1}^{K} \pi_k \prod_{j=1}^{p} [p_j(x_j|\boldsymbol{\theta}_{kj})]^{\psi_j} [q(x_j|\lambda_j)]^{1-\psi_j}\Big) \prod_{j=1}^{p} \rho_j^{\psi_j} (1-\rho_j)^{1-\psi_j} \\ &= \sum_{k=1}^{K} \pi_k \prod_{j=1}^{p} [\rho_j p_j(x_j|\boldsymbol{\theta}_{kj})]^{\psi_j} [(1-\rho_j) q(x_j|\lambda_j)]^{1-\psi_j}, \end{aligned}$$

where $\rho_j = \mathbb{P}(\psi_j = 1)$. Therefore, the marginal density of $\boldsymbol{x}$ is given by

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{\Psi} p(\boldsymbol{x}, \Psi; \boldsymbol{\theta}) &= \sum_{k=1}^{K} \pi_k \sum_{\Psi} \prod_{j=1}^{p} [\rho_j p_j(x_j|\boldsymbol{\theta}_{kj})]^{\psi_j} [(1-\rho_j) q(x_j|\lambda_j)]^{1-\psi_j} \\ &= \sum_{k=1}^{K} \pi_k \prod_{j=1}^{p} [\rho_j p_j(x_j|\boldsymbol{\theta}_{kj}) + (1-\rho_j) q(x_j|\lambda_j)], \end{aligned} \tag{2.69}$$

with $\boldsymbol{\theta} = \{\{\pi_k\}, \{\boldsymbol{\theta}_{kj}\}, \{\lambda_j\}, \{\rho_j\}\}$ is the set of all parameters of the model. The parameter estimation is done via the EM algorithm by treating Z (the hidden class labels) and Ψ are hidden variables.

However, assuming that the irrelevant variables are independent on both the clustering variables and the relevant variable seem to be unrealistic. To tackle these limitations, Maugis et al. (2009a) based on Raftery and Dean (2006) to split the variables into two different sets: $\mathcal{S}$ and $\mathcal{S}^c$, where $\mathcal{S}$ denotes the set of relevant variables and $\mathcal{S}^c$ is the set of irrelevant variables. In the proposed model, they assumed that the subset $\mathcal{S}^c$ of irrelevant variables can be explained by a linear regression from a subset $\mathcal{R}$ of the clustering variables $\mathcal{S}$. The model selection problem can be achieved by maximizing the quantity

$$\text{crit}(\hat{K}, \hat{m}, \hat{\mathcal{S}}, \hat{\mathcal{R}}) = \arg \max_{(K,m,\mathcal{S},\mathcal{R})} \big\{\text{BIC}_{\text{clust}}(\boldsymbol{x}^{\mathcal{S}}|K, m) + \text{BIC}_{\text{reg}}(\boldsymbol{x}^{\mathcal{S}^c}|\boldsymbol{x}^{\mathcal{R}})\big\}, \tag{2.70}$$

where

$$\text{BIC}_{\text{clust}}(\boldsymbol{x}^{\mathcal{S}}|K, m) = 2 \log\{p_{\text{clust}}(\boldsymbol{x}^{\mathcal{S}}|K, m, \hat{\boldsymbol{\theta}})\} - \lambda^{\mathcal{S}}_{(K,m)} \log n, \tag{2.71}$$

with

$$p_{\text{clust}}(\boldsymbol{x}^{\mathcal{S}}|K, m, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}^{\mathcal{S}}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_{k(m)}),$$

$\boldsymbol{\theta} = (\pi_1, \ldots, \pi_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K)$ is the parameter vector, $\lambda^{\mathcal{S}}_{(K,m)}$ is the number of free parameters of the (K, m) mixture model with card$(\mathcal{S})$ variables. The BIC_{reg} is given by

$$\text{BIC}_{\text{reg}}(\boldsymbol{x}^{\mathcal{S}^c}|\boldsymbol{x}^{\mathcal{R}}) = 2 \log\{f_{\text{reg}}(\boldsymbol{x}^{\mathcal{S}^c}|\boldsymbol{x}^{\mathcal{R}}, \hat{B}, \hat{\boldsymbol{\Omega}})\} - \nu_{(\mathcal{S},\mathcal{R})} \log n, \tag{2.72}$$

$(\hat{B}, \hat{\mathbf{\Omega}})$ are the maximum likelihood estimate of the regression parameters and

$$\nu_{(\mathcal{S},\mathcal{R})} = (\text{card}(\mathcal{R}) + 1)\text{card}(\mathcal{S}^c) + \frac{\text{card}(\mathcal{S}^c)(\text{card}(\mathcal{S}^c) + 1)}{2}$$

is the number of parameters for the regression model (in general case).

Later, Maugis et al. (2009b) proposed a general variable role modeling. They split the features into three separated subsets of variable: the relevant variables ($\mathcal{S}$), the irrelevant variables ($\mathcal{U}$) which depend on a subset $\mathcal{R}$ of the relevant ones through a linear relationship and an other part ($\mathcal{W}$) are independent of other variables. The data density can be decomposed into three parts as follows

$$p(\boldsymbol{x}|K, m, r, l, V, \boldsymbol{\theta}) = \left\{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}^{\mathcal{S}}; \boldsymbol{\mu}_k, \mathbf{\Sigma}_{k(m)}\right\} \times \mathcal{N}(\boldsymbol{x}^{\mathcal{U}}; \boldsymbol{a} + \boldsymbol{x}^{\mathcal{R}} B, \mathbf{\Omega}_{(r)}) \times \mathcal{N}(\boldsymbol{x}^{\mathcal{W}}; \boldsymbol{\gamma}, \mathbf{\Gamma}_{(l)}),$$

where $\boldsymbol{\theta}$ is the full parameter vector and $V = (\mathcal{S}, \mathcal{R}, \mathcal{U}, \mathcal{W})$. The form of the regression covariance matrix $\mathbf{\Omega}$ is denoted by r, which can be either spherical, diagonal or general. The form of the covariance matrix $\mathbf{\Gamma}$ of the independent variables $\mathcal{W}$ is denoted by l and can be spherical or diagonal. Thus, the model selection problem is solved by maximizing the following criterion (see also (Maugis et al., 2009b, Section 3))

$$\text{crit}(K, m, r, l, V) = \text{BIC}_{\text{clust}}(\boldsymbol{x}^{\mathcal{S}}|K, m) + \text{BIC}_{\text{reg}}(\boldsymbol{x}^{\mathcal{U}}|\boldsymbol{x}^{\mathcal{R}}, r) + \text{BIC}_{\text{indep}}(\boldsymbol{x}^{\mathcal{W}}|l), \tag{2.73}$$

where $\text{BIC}_{\text{clust}}$, BIC_{reg} are given in (2.71) and (2.72). $\text{BIC}_{\text{indep}}$ represents the BIC criterion of the Gaussian model with the variables $\mathcal{W}$.

For the identifiability of the SRUW model and the consistency of the variable selection, reader can refer to (Maugis et al., 2009b, Section 4). A procedure using embedded stepwise variable selection algorithms is used to identify the SRUW sets. Unfortunately, these procedures are limited as the number of variables is of the order of a few tens. Thus, it is not appropriate for high-dimensional data.

Recently, Celeux et al. (2019) proposed an alternative regularization approach of variable selection to overcome the drawback of the SRUW model. First, the variables are ranked with a Lasso-like procedure in order to avoid painfully slow stepwise algorithms. The variable roles are then determined using the stepwise procedures as in the SRUW model. In the next section, we will consider another way to combine variable selection and clustering by likelihood penalization which including the Lasso method used by Celeux et al. (2019).

Variable selection by likelihood penalization

We take a survey on the penalized likelihood methods. These approaches aim at penalizing the clustering criteria in order to yield sparsity in the features. A general form for the penalized

log-likelihood function is as follows

$$PL_p(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) - p_\lambda(\boldsymbol{\theta}), \tag{2.74}$$

where $L(\boldsymbol{\theta})$ is the log-likelihood function and $p_\lambda(\boldsymbol{\theta})$ is the penalty function. Lasso (Tibshirani, 1996), which will be described in the following Section 2.4.4, is one of the most widely used regularization method in literature. In GMM context, Pan and Shen (2007) proposed the Lasso-type penalized log-likelihood to yield the sparsity of the mean vectors

$$PL(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log\Big\{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\Big\} - \lambda \sum_{k=1}^{K} \|\mu_k\|_1, \tag{2.75}$$

where $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_p^2)$ for all $k \in \{1, \ldots, K\}$ and λ is tuning parameter which decides the level of sparsity. Note that, the observations are standardized to have zero mean and unit variance for each variable j. If $\mu_{1j} = \cdots = \mu_{Kj} = 0$, then the variable jth cannot differentiate the components, hence deemed as noninformative and automatically excluded from clustering.

In a similar approach, Xie et al. (2008a) proposed a method dealing with the case of clustering specific diagonal covariance matrices leading to the following penalty function

$$p_\lambda(\boldsymbol{\theta}) = \lambda_1 \sum_{k=1}^{K} \|\mu_k\|_1 + \lambda_2 \sum_{k=1}^{K} \sum_{j=1}^{p} |\sigma_{kj}^2 - 1|, \tag{2.76}$$

with $\boldsymbol{\Sigma}_k = \text{diag}(\sigma_{k1}^2, \ldots, \sigma_{kp}^2)$ be the covariance matrix of the kth group. In this case, a second penalty term is used to force an estimate of σ_{kj}^2 that is close to 1 to be exactly 1.

Finally, Zhou et al. (2009) considered a general covariance matrix $\boldsymbol{\Sigma}$ by relaxing the diagonal covariance matrix assumption. To facilitate estimating large and sparse covariance matrices, they employed the following penalty function

$$p_\lambda(\boldsymbol{\theta}) = \lambda_1 \sum_{k=1}^{K} \|\mu_k\|_1 + \lambda_2 \sum_{j=1}^{p} \sum_{h=1}^{p} |\boldsymbol{\Sigma}_{jh}^{-1}|. \tag{2.77}$$

This penalty function is then modified to permit varying cluster volumes and orientations

$$p_\lambda(\boldsymbol{\theta}) = \lambda_1 \sum_{k=1}^{K} \|\mu_k\|_1 + \lambda_2 \sum_{k=1}^{K} \sum_{j,h} |\boldsymbol{\Sigma}_{k;jh}^{-1}|, \tag{2.78}$$

where $\boldsymbol{\Sigma}_k$ is the covariance matrix of the kth cluster. Note that, the penalty on the mean parameter is mainly for variable selection, while that for the covariance matrices is necessary for high-dimensional data. However, this leads to a difficult problem in estimating the covariance matrices, which are said to be positive-definite.

In the same spirit, Xie et al. (2010) proposed a penalized MFA approach from the model

introduced by Ghahramani and Hinton (1996), where the noise covariance matrix is diagonal and common to all factor. By standardizing each variable to have variance at 1, one can treat these variables in a similar scale and thus penalize their mean parameters together by an ℓ_1 penalty. In addition, a variable j is irrelevant to all clusters if all $\mu_{jk}, k = 1, \ldots, K$ are 0 and all $b_{kj\cdot} = (b_{k;j1}, \ldots, b_{k;jq})$, $k = 1, \ldots, K$ are required to be $\mathbf{0}$, where $b_{k;jl}$ is the value of the loading matrix Λ_k at row jth and column lth. Hence, to realize more effective variable selection, it is natural to treat $b_{k;j1}, \ldots, b_{k;jq}$ as a group of parameters, constructing a penalty that encourages all of them to be exactly 0. Therefore, the authors regularized the log-likelihood function of (2.61) with the following penalty function

$$\begin{aligned} p_(\lambda, \gamma)(\boldsymbol{\theta}) &= \lambda p_1(\boldsymbol{\mu}) + \gamma p_2(\Lambda) \\ &= \lambda \sum_{k=1}^{K} \|\boldsymbol{\mu}_k\|_1 + \gamma \sum_{k=1}^{K} \sum_{j=1}^{p} \|b_{kj\cdot}\|_2, \end{aligned} \tag{2.79}$$

where $\|b_{kj\cdot}\|_2 = \sqrt{\sum_{l=1}^{q} b_{k;jl}^2}$. The ℓ_1 norm $p_1(\boldsymbol{\mu})$, as in Pan and Shen (2007), is use to shrink a small estimate μ_{kj} to be exactly 0, while $p_2(\Lambda)$, serving as a grouped variable penalty as in Xie et al. (2008b) is used to shrink an estimate of factor loading vector $b_{kj\cdot}$ that is close to $\mathbf{0}$ to be exactly $\mathbf{0}$.

As general, the tuning parameters of all these regularization methods are selected through a modified BIC criterion, which takes into account the level of sparsity in the model complexity term. However, the selection of the sparsity parameters is still an open issue. Among the related works, we refer the works of Witten and Tibshirani (2010) and Wang and Zhu (2008). For the first approach, Witten and Tibshirani proposed a framework for sparse clustering based on a Lasso-type penalty to select the features. The method is then used to develop sparse K-means and sparse hierarchical clustering methods. For the remaining, Wang and Zhu suggested two regularization methods. The first one replaces the ℓ_1-norm in Pan and Shen (2007) with the ℓ_∞-norm to shrink the mean vectors and obtain a sparsity model

$$p_\lambda(\boldsymbol{\theta}) = \lambda \sum_{j=1}^{p} \|(\mu_{1j}, \ldots, \mu_{Kj})\|_\infty, \tag{2.80}$$

where $\|(\mu_{1j}, \ldots, \mu_{Kj})\|_\infty = \max_k(|\mu_{1j}|, \ldots, |\mu_{Kj}|)$. Different from the ℓ_1-norm, the ℓ_∞-norm penalizes the maximum absolute value of $\mu_{kj}, k = 1, \ldots, K$, for the jth variable. If the maximum of $|\mu_{kj}|$ is reduced to 0, all μ_{kj} are automatically reduced to 0. The jth variable is therefore irrelevant. However, this ℓ_∞-norm penalty tends to reduce the μ_{kj}, $k = 1, \ldots, K$ into the same magnitude. This motivates them to propose the second penalty function. First, μ_{kj} is reparameterized in a more general form

$$\mu_{kj} = \gamma_j \eta_{kj}, \ k = 1, \ldots, K; \ j = 1, \ldots, p, \tag{2.81}$$

where $\gamma_j \geq 0$. Secondly, they considered the following hierarchically penalized GMM

$$PL(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) - \lambda_\gamma \sum_{j=1}^{p} \gamma_j - \lambda_\eta \sum_{k=1}^{K} \sum_{j=1}^{p} |\eta_{kj}|, \tag{2.82}$$

subject to $\gamma_j \geq 0$. If γ_j is reduced to zero then all μ_{kj} for the jth variable will be equal to zero. Otherwise, some of θ_{kj} hence some of the μ_{kj}, $k = 1, \ldots, K$, still have the possibility of being zero; in this sense, the hierarchical penalty keeps the flexibility of the ℓ_1-norm penalty.

2.4.4 Lasso regularization towards the mixture approach

In this section, we introduce the Lasso estimator for the linear regression problem, before considering it for the mixture of regression models. This method plays a key role in our research. A method for solving the Lasso is also mentioned at the end of this section.

The Lasso regularization

Lasso is the ℓ_1 regularization of linear regression model, in which we assume that we are given n observations $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$, where the response variable $y_i \in \mathbb{R}$ is related to the p-dimensional vector of predictors $\boldsymbol{x}_i \in \mathbb{R}^p$ via the linear combination of the predictors $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^T$ as:

$$f(\boldsymbol{x}_i) = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}. \tag{2.83}$$

Here, β_0 is the intercept term $\beta_0 \in \mathbb{R}$ and $\boldsymbol{\beta} \in \mathbb{R}^p$ is the vector of regression weights. Without loss of generality, assume that $\beta_0 = 0$.

The least-squares estimator for $\boldsymbol{\beta}$ is based on minimizing the square-error loss:

$$\arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2, \tag{2.84}$$

where $\mathbf{y} = (y_1, \ldots, y_n)^T$ and

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1p} \\ x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & \ldots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{np} \end{pmatrix}.$$

It is not hard to check that the solution of (2.84), which is called the least-squares estimator, is given by

$$\hat{\boldsymbol{\beta}}^{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{2.85}$$

However, it is well known in literature that if the square matrix $\mathbf{X}^T\mathbf{X}$ has one or more small eigenvalues, then the distance from the estimator $\hat{\boldsymbol{\beta}}^{LS}$ and the true parameter vector $\boldsymbol{\beta}$ will tend to be large. Especially, since $n < p$ the least-squares estimator poses a problem: the solution is

not unique because the rank of $\mathbf{X}^T\mathbf{X}$ is smaller than p.

To tackle the ill-conditional condition of $\mathbf{X}^T\mathbf{X}$, Hoerl and Kennard (1970) proposed an ℓ_2 regularization method which is named ridge regression. Basically, the ridge regression solves the follows problem

$$\hat{\boldsymbol{\beta}}^{\text{ridge}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2, \text{ subject to } \|\boldsymbol{\beta}\|_2 \leq t. \tag{2.86}$$

Unfortunately, although ridge regression shrinks the regression coefficients but this method still poses a drawback since none of the ridge regression equals to zero. Therefore, all the features x_js still be used to construct the regression function. For more details on the ridge regression, readers can refer to Hoerl (1985), Friedman et al. (2001).

Tibshirani (1996) introduced a minutely method to overcome the drawback of least square estimation and also the ridge regression. The idea lies in the fact that replacing the Euclidean norm in constraint of the ridge regression $\|\beta\|_2 \leq t$ with the ℓ_1 constraint $\|\beta\|_1 \leq t$. This model is called Least Absolute Shrinkage and Selection Operator (Lasso). It was directly inspired by the nonnegative garrote estimator of Breiman (1995). As its name, this method provides two advantages: first it shrinks some coefficients and secondly, it sets other coefficients to 0. Therefore it retains the good features of both subset selection and ridge regression. Figure 2.3 depicts the Lasso and ridge regression for the vector $\boldsymbol{x}$ is of the size 2. Both methods find the first point, where the elliptical contours hit the constraint region. Since the Lasso constraint region has corners; if the hit point is one of these corners then one of the two parameters is equal to zero.

It is interesting to look back to the classical feature selection method based on the ℓ_0 norm as a natural approach for features selection, which is described as follows:

$$\arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2, \text{ subject to } \|\boldsymbol{\beta}\|_0 \leq t, \tag{2.87}$$

where the ℓ_0 norm is defined by

$$\|\boldsymbol{\beta}\|_0 \triangleq \text{card}\{j \in \{1, \ldots, p\} : \beta_j \neq 0\}.$$

This model has paid a lot of attentions in literature (see Foster et al. (1994)). Unfortunately, the constraint is non-convex and the ℓ_0 regularization method is proofed to be an NP-hard problem (see Natarajan (1995)). Hence, it is ineffective to use this method in application and solving (2.87) is still a challenge. Recently, Frommlet and Nuel (2016) introduce an adaptive ridge procedure (AR), where iteratively weighted ridge problems are solved whose weights are updated in such a way that the procedure converges towards selection with ℓ_0 penalties.

Going back to the ℓ_1 regularized method, the Lasso estimate requires solving an optimization problem, which can be described as

$$\arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \beta_0 - \mathbf{X}\boldsymbol{\beta}\|_2^2, \text{ subject to } \|\boldsymbol{\beta}\|_1 \leq t. \tag{2.88}$$

For convenience, assuming that all features are standardized to have mean zero and unit variance. We also assume that the outcome values y_i have been centered. By doing this, we can omit the intercept β_0. Hence, without loss of generally, the Lasso can be rewritten as

$$\arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2, \text{ subject to } \|\boldsymbol{\beta}\|_1 \leq t. \tag{2.89}$$

It is often convenient to rewrite the Lasso problem in the Lagrange form,

$$\arg\min_{\boldsymbol{\beta}} \Big\{ \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \Big\}. \tag{2.90}$$

The value of λ will decide the sparsity of the model. As proved in Section A.4.1, the smallest

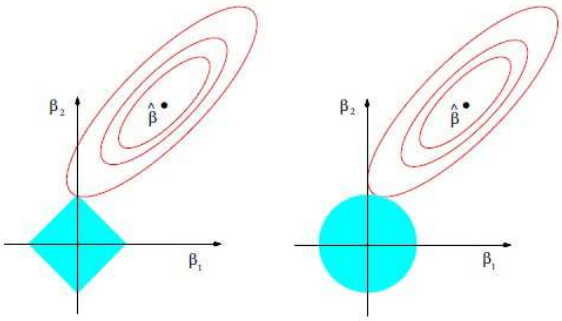


Figure 2.3: Estimation picture for the Lasso (left) and ridge regression (right) with constraint regions and the contours of the least squares error function. Picture from Friedman et al. (2001)

value of λ such that the regression coefficients estimated by Lasso (2.90) are all equal to zero is

$$\lambda_{\max} = \max_j |\langle \mathbf{y}, \mathbf{x}_j \rangle|,$$

where $\mathbf{y} = (y_1, \ldots, y_n)^T$, $\mathbf{x}_j = (x_{1j}, x_{2j}, \ldots, x_{nj})^T$. $\langle \mathbf{y}, \mathbf{x}_j \rangle$ is the inner product between $\mathbf{y}$ and $\mathbf{x}_j$, $\langle \mathbf{y}, \mathbf{x}_j \rangle = \mathbf{y}^T\mathbf{x}_j$.

The bound of t in (2.88) controls the complexity of the model: the larger values of t allow the model to adapt more closely to the training data. Conversely, the smaller values of t will restrict the parameters more and leading to sparser but the interpretable model fit the data less closely. Therefore, a natural question arises for the Lasso regression: "how to choose t or λ?". A procedure known as *cross-validation* is a common tool to decide the value of the tuning parameter λ. For more details, (Hastie et al., 2015, Sec. 2.3) is a helpful text for the cross-validation procedure.

Lemma 2.4.2 (Properties of the Lasso solutions (Tibshirani (2013)))**.** *For some $\lambda \geq 0$, suppose the $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\eta}}$ two solutions of (2.90) with common optimal value c^* then:*

- $\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}\hat{\boldsymbol{\eta}}$*, i.e, the two solutions must yield the same predicted values.*
- *If $\lambda > 0$ then $\|\hat{\boldsymbol{\beta}}\|_1 = \|\hat{\boldsymbol{\eta}}\|_1$.*

The proof of this lemma can be found in Appendix A.4.2. As shown in Lemma 2.4.2, the fitted values $\mathbf{X}\hat{\boldsymbol{\beta}}$ are unique. However, the solution $\hat{\boldsymbol{\beta}}$ may not be unique. Consider an example with two predictors $\mathbf{x}_1$, $\mathbf{x}_2$ and response $\mathbf{y}$. Assume that the Lasso solution coefficients $\hat{\boldsymbol{\beta}}$ at λ are $(\hat{\beta}_1, \hat{\beta}_2)$. If we include the third predictor $\mathbf{x}_3$ which is an identical copy of the second predictor, i.e, $\mathbf{x}_3 = \mathbf{x}_2$ and consider the Lasso problem

$$\arg \min_{(\beta_1,\beta_2,\beta_3)} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_1 x_{i1} - \beta_2 x_{i2} - \beta_3 x_{i3})^2 + \lambda \sum_{j=1}^{3} |\beta_j|. \tag{2.91}$$

This optimization problem can be rewritten as

$$\arg \min_{\{\beta\}} \Big\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_1 x_{i1} - (\beta_2 + \beta_3) x_{i2})^2 + \lambda(|\beta_j| + |\beta_2 + \beta_3|) + \lambda(|\beta_2| + |\beta_3| - |\beta_2 + \beta_3|) \Big\}. \tag{2.92}$$

It turns out that, for any $\alpha \in [0, 1]$ the triplet $(\hat{\beta}_1, \alpha\hat{\beta}_2, (1-\alpha)\hat{\beta}_2)$ is a solution of (2.91). For this model, there is an infinite family of solutions.

In general, when $\lambda > 0$, Tibshirani (2013) showed that if the matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ has columns in *general position*, then the Lasso solutions are unique. To be precise, the columns $\{\mathbf{x}_j\}_{j=1}^{p}$ of $\mathbf{X}$ are in general position if no k-dimensional subspace $L \subseteq \mathbb{R}^n$, for $k < \min\{n, p\}$, contains more than $k + 1$ elements of the set $\{\pm\mathbf{x}_1, \ldots, \pm\mathbf{x}_p\}$, excluding antipodal pairs. This means the affine span of any $k + 1$ points $\sigma_1 \mathbf{x}_{j_1}, \ldots, \sigma_{k+1}\mathbf{x}_{j_{k+1}}$ $(j_1, \ldots, j_{k+1} \in \{1, \ldots, p\})$, for arbitrary signs $\sigma_1, \ldots, \sigma_{k+1} \in \{-1, +1\}$ does not contain any element of $\{\pm\mathbf{x}_j : j \neq j_1, \ldots, j_{k+1}\}$. For the complete proof of this property see (Tibshirani, 2013, Lemma 2, 3). Hence, if the data matrix $\mathbf{X}$ are drawn from a continuous probability distribution, then with probability one the data are in general position and the Lasso solutions will be unique. The non-uniqueness of the Lasso solutions can occur only with discrete-valued data. In the previous example, the affine is generated from $\{\pm\mathbf{x}_1, \pm\mathbf{x}_2\}$ also contains $\pm\mathbf{x}_3$. Therefore, the data are not in general position.

There has been an enormous amount theoretical works analyzing the performance of the Lasso. For the related works, some references are Bickel et al. (2009); Meinshausen et al. (2006); Donoho et al. (2006); Zhao and Yu (2006); Wainwright (2009); Massart and Meynet (2011); a helpful book for these kind of results is Bühlmann and Van De Geer (2011).

Solving the Lasso optimization problem

The Lasso problem is a convex program, specifically a quadratic program with a convex constraint. As such, there are many sophisticated quadratic program methods for solving the Lasso. In this thesis, we use one of the simple but effective methods for solving the Lasso: coordinate descent algorithm. For more details on this method, viewers can the reader can be referred to Boyd and Vandenberghe (2004); Hastie et al. (2015).

Consider the convex program

$$\arg\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}),$$

with the objective function $f(\boldsymbol{\beta})$ is presented as follows

$$f(\boldsymbol{\beta}) = g(\boldsymbol{\beta}) + h(\boldsymbol{\beta})$$

where $g(.)$ is a differentiable convex function, $h(.)$ is convex but not necessarily differentiable.

Coordinate descent is an iterative algorithm that updates from $\boldsymbol{\beta}^{(t)}$ to $\boldsymbol{\beta}^{(t+1)}$ by choosing a single coordinate to update, and then performing a univariate minimization over this coordinate. If coordinate k is chosen at iteration t, then

$$\boldsymbol{\beta}_k^{(t+1)} = \arg\min_{\beta_k} f(\beta_1^{(t)}, \ldots, \beta_{k-1}^{(t)}, \beta_k, \beta_{k+1}^{(t)}, \ldots, \beta_p^{(t)}), \tag{2.93}$$

and $\beta_j^{(t+1)} = \beta_j^{(t+1)}$ for $j \neq k$.
The coordinate descent can get "stuck", and fail to reach the global minimum of f. A counter example is shown in (Hastie et al., 2015, Section 5.4). However, if the cost function f has the additive decomposition

$$f(\beta_1, \ldots, \beta_p) = g(\beta_1, \ldots, \beta_p) + \sum_{j=1}^{p} h_j(\beta_j) \tag{2.94}$$

where $h_j : \mathbb{R} \to \mathbb{R}$ are convex, then with some conditions including the separate structure of $h(\boldsymbol{\beta})$ Tseng (2001) and Wu and Lange (2008) show that the coordinate descent algorithm converges to the minimum point. For the Lasso problem, we have

$$f(\boldsymbol{\beta}) = g(\boldsymbol{\beta}) + \sum_{j=1}^{p} h(\beta_j),$$

with $g(\boldsymbol{\beta}) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2$ and $h(\beta_j) = \lambda|\beta_j|$. Updating $\beta_j^{(t+1)}$ using the coordinate descent procedure is done by solving this problem

$$\beta_j^{(t+1)} = \arg\min_{\beta_j} \frac{1}{2}\sum_{i=1}^{n}\big[\beta_j x_{ij} + (\sum_{k\neq j}\beta_k^{(t)} x_{ik} - y_i)\big]^2 + \lambda|\beta_j|. \tag{2.95}$$

Then the solution for (2.95) satisfies

$$\beta_j^{(t+1)} = \mathcal{S}_\lambda\big(\sum_{i=1}^{n} r_i^j x_{ij}\big)\Big/\sum_{i=1}^{n} x_{ij}^2, \tag{2.96}$$

with $r_i^j = y_i - \sum_{k\neq j}\beta_k^{(t)} x_{ik}$ and $\mathcal{S}_\lambda(.)$ is a *soft-thresholding* operator defined by $\mathcal{S}_\lambda(x) = \text{sign}(x)(|x| - \lambda)_+$ and $(x)_+$ a shorthand for $\max\{x, 0\}$. Other parameters are kept while updating β_j.

The coordinate descent procedure simply solves (2.95) in a cyclical way, iterating over $j = 1, 2, \ldots, p, 1, 2, \ldots$. In general, the coordinate descent algorithm for solving the Lasso can be interpreted as below:

Algorithm 1 Coordinate descent method for solving the Lasso

1: $\boldsymbol{\beta}^{(0)}$.
2: **repeat**
3: **for** $j = 1$ to p **do**
4: Update $\beta_j^{(t+1)}$ using the soft-thresholding operator in (2.96).
5: For $k \neq j$ then $\beta_k^{(t+1)} = \beta_k^{(t)}$.
6: **end for**
7: Evaluate the objective function.
8: **until** the stopping criterion is satisfied.

For the standard Lasso regression model

$$\arg\min_{(\beta_0, \boldsymbol{\beta})} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_0 - \boldsymbol{x}_i^T \boldsymbol{\beta})^2 + \lambda \|\boldsymbol{\beta}\|_1, \tag{2.97}$$

one just need to modify the updating of β_0 at each step using the value

$$\beta_0^{(t+1)} = \frac{\sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^T \boldsymbol{\beta}^{(t+1)})}{n}.$$

Example 2.4.1. *In this example,* $n = 150$ *data points were generated with* $p = 6$ *features,* $X_j \overset{i.i.d}{\sim} U[0, 1]$. *The respond variable* Y *is of the form*

$$Y = 1 + 2X_1 - X_4 + 0.5X_6 + \varepsilon,$$

where $\varepsilon \sim \mathcal{N}(0, 0.5^2)$ *and* $\lambda = 2.8$. *Starting from* $(\beta_0, \boldsymbol{\beta}^T)^T = \mathbf{0}$ *with the objective function value* 273.88630, *after* 187 *loops the coordinate descent algorithm stops at*

$$(\hat{\beta}_0(\lambda), \hat{\boldsymbol{\beta}}^T(\lambda))^T = (0.9877018, 1.7612036, 0, 0, -0.5732187, 0, 0.3337205)^T,$$

with the minimum value 26.51752. *It is clear that, in this case the Lasso estimator provides the zero coefficients coincident with the true zero parameters of the model. For the non-zero coefficients (excepts the intercept), there are biases between these parameters with the true ones cause by penalty function. The changing of the objective function after each iteration can be observed from Figure 2.4.*

Least angle regression (LARS) is another efficient algorithm for solving the Lasso, which is proposed by Efron et al. (2004). It still requires to compute the inverse matrix in each loop, hence does not scale up to large problems as well as the coordinate descent method. An MM version for solving the regularized regression problems has been investigated in Hunter and Li (2005). Chapter 5 of Hastie et al. (2015) provides a general review for other optimization

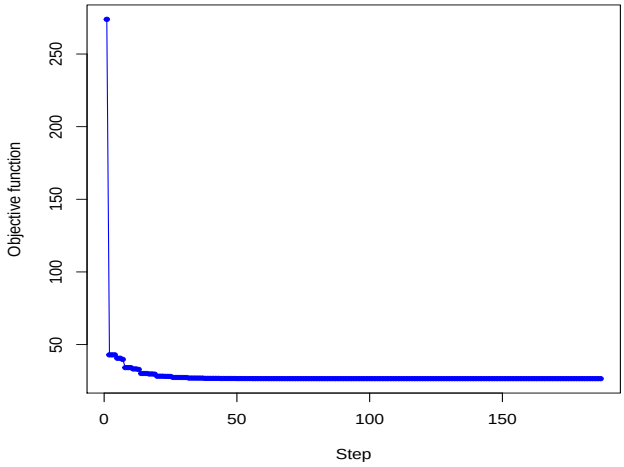


Figure 2.4: The changing of the objective function after each iteration.

methods which can be used to solve the Lasso.

2.5 Regularized mixtures of regression models

In the applications of the mixture of regression models, many features are often used, and the contribution of each feature to the response variable Y is different from one component to another of the model. As the number of features and components in the mixture models increases the cost of computing the Akiake information criterion (Akaike, 1974) and the BIC also become more expensive. Hence, these methods may not be not idea in certain cases. In this section, we take some recent works that use penalized likelihood approaches for variable selection in mixture of regression models. Some authors such as (Khalili and Chen, 2007; Städler et al., 2010a; Khalili and Lin, 2013; Hui et al., 2015) tackle the problem of variable section for mixture of regressions models using Lasso-type regularization methods. In the similar context, (Devijver, 2015) considered the Lasso-penalized methods for mixture of multivariate Gaussian regression models. For the MoEs models, several works, such as (Khalili, 2010; Peralta and Soto, 2014; Jiang et al., 2018) in particular, introduced a penalty term in the log-likelihood function in order to yield the sparsity in the features both for the experts and for the gating network. The reader can also refer to Khalili (2011), which provides a comprehensive overview of the feature selection methods in mixture of regression models.

2.5.1 Regularized mixture of regression models

Khalili and Chen (2007) is considered as the earliest variable selection method in mixture of linear regression models (MLR) using penalized likelihood approach. Let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$

be a sample of observations from the MLR model in (2.25). The (conditional) log-likelihood function of $\boldsymbol{\theta}$ is given by

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log\Big\{\sum_{k=1}^{K} \pi_k \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k, \sigma_k^2)\Big\}. \tag{2.98}$$

When the effect of a jth feature is not significant, the corresponding ordinary maximum likelihood estimate is often close to, but not equal to 0. To exclude the role of this feature, these author consider the penalized log-likelihood function as

$$PL(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) - \sum_{k=1}^{K} \pi_k \sum_{j=1}^{p} p_n(\beta_{kj}; \lambda_{nk}), \tag{2.99}$$

where $p_n(\beta_{kj}; \lambda_{nk})$ could be one of the following penalty functions:

- ℓ_1-norm penalty (The Lasso):

$$p_n(\beta_{kj}; \lambda_{nk}) = \lambda_{nk}\sqrt{n}|\beta_{kj}|. \tag{2.100}$$

- SCAD penalty (Smooth clipped absolute deviation penalty Fan and Li (2001)):

$$\frac{p_{nk}(\beta_{kj}; \lambda_{nk})}{\sqrt{n}} = \begin{cases} \lambda_{nk}|\beta_{kj}|, & |\beta_{kj}| \le \lambda_{nk} \\ -(\beta_{kj}^2 - 2a\lambda_{nk}|\beta_{kj}| + \lambda_{nk}^2)/[2(a-1)], & \lambda_{nk} < |\beta_{kj}| \le a\lambda_{nk} \\ \lambda_{nk}^2(a+1)/2, & |\beta_{kj}| > a\lambda_{nk} \end{cases}, \tag{2.101}$$

  for some constant $a > 2$.

- HARD penalty (hard thresholding penalty Antoniadis (1997)):

$$p_{nk}(\beta_{kj}) = \lambda_{nk}^2 - (\sqrt{n}|\beta_{kj}| - \lambda_{nk})^2 I(\sqrt{n}|\beta_{kj}| < \lambda_{nk}).$$

The hope is that with a proper choice of the penalty functions, if some of the regression coefficients β_{kj} are zero, then their corresponding estimator $\hat{\beta}_{kj}$ are also zero. Although the Lasso has many attractive properties, the shrinkage introduced by it results in significant bias toward 0 for large regression coefficients. The remains are nonconvex penalty functions. The SCAD has the so-called *oracle property*, i.e., in asymptotic sense, it performs as well as the oracle procedure in variable selection; namely, it works as well as if the correct submodel was known.

Note that the amount of penalty on each regression parameters β_{kj} is proportional to the mixing probability π_k, which is a common practice in relating the amount of penalty to the sample size. Under standard regularity conditions as well as the certain conditions on the penalty function, Khalili and Chen (2007) showed the estimator $\hat{\boldsymbol{\theta}}_n$ of (2.99) satisfies (see (Khalili and

Chen, 2007, Theorem 1))

$$\|\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0\| = O_p(n^{-1/2}(1 + b_n)),$$

where $\|.\|$ represents the Euclidean norm, $\boldsymbol{\theta}_0$ is the true vector of parameters in the true MLR model underlying the data and

$$b_n = \max_{k,j}\{|p'_{nk}(\beta^0_{kj})|/\sqrt{n} : \beta^0_{kj} \neq 0\}.$$

In addition, if the penalty function guarantees that, for $N_n = \{\beta : 0 < \beta \leq n^{-1/2}\log n\}$,

$$\lim_{n\to\infty} \inf_{\beta\in N_n} p'_{nk}(\beta)/\sqrt{n} = \infty,$$

then $\hat{\boldsymbol{\theta}}_n$ also provides the following properties (see also (Khalili and Chen, 2007, Theorem 2)):

- Sparsity: $\mathbb{P}(\hat{\boldsymbol{\beta}}_{k2} = \mathbf{0}) \to 1$, for $k = 1, \ldots, K$, where $\hat{\boldsymbol{\beta}}_{k2}$ is the vector of zero regression coefficients in the kth component of the model.

- Asymptotic normality:

$$\sqrt{n}\left\{\left[I(\boldsymbol{\theta}_{01}) - \frac{p''(\boldsymbol{\theta}_{01})}{n}\right](\hat{\boldsymbol{\theta}}_{n1} - \boldsymbol{\theta}_{01}) + \frac{p'(\boldsymbol{\theta}_{01})}{n}\right\} \xrightarrow{d} \mathcal{N}(\mathbf{0}, I(\boldsymbol{\theta}_{01})),$$

  where $p'(.)$ and $p''(.)$ are the first and second derivatives of the penalty function with respect to β_{kj}'s, and $I(\boldsymbol{\theta}_{01})$ is the Fisher information matrix computed under the reduced model when all zero effects are removed.

To obtain these properties, the number of mixture component K of the model is known. In applications, one can use the BIC or other criteria to select K. Another strategy of regularization to perform simulatenous parameter estimation and selection of K is presented in Chamroukhi (2013, 2016d).

Parameter estimation is done via the EM algorithm. To maximize the well-known Q-function in the M-step, the authors approximated the penalty function by a local quadratic approximation using Hunter and Li (2005) suggestion

$$p_{nk}(\beta) \approx p_{nk}(\beta_0) + \frac{p'_n k(\beta_0)}{2(\beta_0 + \epsilon)}(\beta^2 - \beta_0^2), \tag{2.102}$$

in a neighborhood of β_0. A Newton-Raphson algorithm can then be used to maximize the penalized Q-function, where each iteration updates the local quadratic approximation. The drawback lies in the fact that using Newton-Raphson algorithm for high-dimensional data (p is large) may not be an appropriate choice. In addition, one must choose a threshold δ and declares a coefficient is zero if it values smaller than δ. Therefore, the selection of ϵ in (2.102) and δ affects the sparsity of the model and should be consider carefully. One important thing to mention is that maximizing the Q-function with respect to the mixing proportions $\pi_k, k = 1, \ldots, K$ is still

a challenge. In their work, the authors keep the same update of π_k as a non penalized case and it still works quite well.

Later, Khalili and Lin (2013) proposed elastic-net type penalties for finite mixture regression models with a diverging number of covariates in each component. Their penalized log-likelihood function is as follows

$$PL_n(\boldsymbol{\theta}) = L_n(\boldsymbol{\theta}) - p_n(\boldsymbol{\theta}) \tag{2.103}$$

with the penalty function

$$p_n(\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \sum_{j=1}^{p} r_n(\beta_{kj}; \lambda_{nk}) + \frac{\tau_n}{2} \sum_{k=1}^{K} \pi_k \sum_{j=1}^{p} \beta_{kj}^2, \tag{2.104}$$

for some tuning parameter $\tau_n \geq 0$, and $r_n(\beta_{kj}; \lambda_{nk})$ is a nonnegative penalty function indexed by some tuning parameter $\lambda_{nk} \geq 0$. In fact, the authors considered three well-known regularization functions for $r_n(\beta_{kj}; \lambda_{nk})$: the Lasso and SCAD as in (2.100), (2.101) and MCP (Zhang (2010)) which is given by

$$\frac{r_n(\beta_{kj}; \lambda_{nk})}{n} = \begin{cases} \lambda_{nk}\big(|\beta_{kj}| - \frac{\beta_{kj}^2}{2\gamma\lambda_{nk}}\big), & |\beta_{kj}| < \gamma\lambda_{nk} \\ \gamma\lambda_{nk}^2/2, & |\beta_{kj}| \geq \gamma\lambda_{nk} \end{cases}. \tag{2.105}$$

The parameter γ in MCP controls the concavity of the penalty. Moreover, if $\gamma \to \infty$ the penalty becomes Lasso, if $\gamma \to 0^+$ then it becomes the ℓ_0 penalty.
Using the similar approach in Khalili and Chen (2007), the authors draw some asymptotic properties for their models. EM algorithms for parameter estimation are also given using the same strategy in Khalili and Chen (2007). The methods are then applied to study the Gaussian and Poisson finite mixture of regression models.

Similarly, Städler et al. (2010a) applied the adaptive Lasso (Zou (2006)) with one tuning parameter for the mixture of linear regression models. In addition, some asymptotic theory and oracle inequalities are also presented. Conversely to this approach, Meynet (2013) presented a complementary result to Städler et al. (2010a) by studying the Lasso for its ℓ_1-regularization properties rather than considering it as a variable selection procedure. Meynet established oracle inequality for the Lasso in finite mixture Gaussian regression models, which can be seen as an extension of Massart and Meynet (2011). Devijver (2015) extended Meynet's ℓ_1-oracle inequality for the mixture of multivariate Gaussian regression models. To exploit the grouped structure of covariates in the MLR models, Hui et al. (2015) introduced two regularization models MIXGL1 and MIXGL2 based on a modification of the group Lasso (Yuan and Lin (2006a)). With some regularity conditions, the oracle properties for their proposed models are also studied. Finally, it is interesting to present the work of Lloyd-Jones et al. (2018), who introduced an MM algorithm for maximizing the Lasso-penalized likelihood function of the MLR model.

2.5.2 Regularized mixture of experts models

A general framework for the regularized MLR model was also proposed by Khalili (2010) which focuses on MoE model. Since each mixing proportion is replaced by a softmax function of the variates, the effect of each feature appears both in the experts network and in the mixing proportion functions (the gating network). Hence, for feature selection, two extra penalty functions are applied to the ℓ_2-penalized log likelihood function. His proposal penalized log-likelihood is then given by

$$PL(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log\Big[\sum_{k=1}^{K} \pi_k(\mathbf{w}; \boldsymbol{x}_i)\mathcal{N}(y_i; \beta_{k0} + \boldsymbol{x}_i^T\boldsymbol{\beta}_k, \sigma_k^2)\Big] - p_n(\boldsymbol{\theta}), \tag{2.106}$$

with $\pi_k(\mathbf{w}; \boldsymbol{x}_i)$ is a softmax function in (2.37),and

$$p_n(\boldsymbol{\theta}) = \sum_{k=1}^{K}\sum_{j=1}^{p} r_n(\beta_{kj}; \lambda_k) + \sum_{k=1}^{K-1}\sum_{j=1}^{p} r_n(w_{kj}; \gamma_k) + \frac{\tau_n}{2}\sum_{k=1}^{K-1} \|\boldsymbol{w}_k\|_2^2, \tag{2.107}$$

where $r_n(.;.)$ is the Lasso or SCAD functions. Adding an ℓ_2-norm of the mixing probabilities to the log-likelihood function controls the large positive and negative estimates of the regression coefficients corresponding to the correlated features. This behavior can be observed in logistic/multinomial regression when the number of potential features is large and highly correlated; see Park and Hastie (2007b) and Bunea (2008). However, this also affects the sparsity of the regularization model.

By extending the theoretical developments for MLR models in Khalili and Chen (2007), a standard asymptotic theory for MoE models is established (see Theorem 2, Theorem 3 of Khalili (2010)). The asymptotic properties of the estimator $\hat{\boldsymbol{\theta}}_n$ depend on the choice of the penalty $r_n(.;.)$ and the tuning parameter τ_n accompanying the ℓ_2-norm. By taking an appropriate choice of these tuning parameters in SCAD, the estimator $\hat{\boldsymbol{\theta}}_n$ of the penalized log-likelihood is both consistent in feature selection and maintains root-n consistency. In contradiction with SCAD, for Lasso, the estimator $\hat{\boldsymbol{\theta}}_n$ cannot achieve both properties simultaneously. It is worth to point out that the proposed methodology of Khalili (2010) can be used for feature selection in the high-dimensional situations ($p \gg n$). However, the statistical properties of the method in high-dimensional context such as consistency and oracle inequality still be open questions.

For maximizing the regularized log-likelihood, an appropriate EM algorithm combines with the Newton-Raphson method was introduced. Using the suggestion of Hunter and Li (2005), the penalty functions for both the gating network and the experts network are approximated with their local quadratic functions. After that, the Newton-Raphson can be used to update these coefficients in the M-step. However, some drawbacks from the EM algorithm for regularized MLR in Khalili and Chen (2007) still appear here. One must choose a threshold δ to declare zero coefficient and an addition parameter ϵ to avoid numerical instability of the algorithm due to the small values of some of the coefficient in the demominator of the approximation (2.102).

In a similar context, Peralta and Soto (2014) studied the ℓ_1-regularized MoE for multinomial logit models. Hence, they assume that each expert component follows a multinomial distribution with R levels ($R \geq 2$), i.e.,

$$Y_i|Z_i = z_i, \boldsymbol{x}_i \sim \text{Mult}(1; \varphi_{z_i 1}(\boldsymbol{x}_i; \boldsymbol{\beta}_{z_i}), \ldots, \varphi_{z_i R}(\boldsymbol{x}_i; \boldsymbol{\beta}_{z_i}))$$

where

$$\varphi_{kr}(\boldsymbol{x}_i; \boldsymbol{\beta}_k) = \mathbb{P}(y_i = r|\boldsymbol{x}_i; z_i = k; \boldsymbol{\beta}_k) = \frac{\exp(\beta_{kr0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr})}{\sum_{l=1}^{R} \exp(\beta_{kl0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_{kl})}, \ r \in \{1, \ldots, R\}.$$

The ℓ_1 regularized observed data log-likelihood is given by

$$\begin{aligned} PL(\boldsymbol{\theta}) = &\sum_{i=1}^{n} \log\Big[\sum_{k=1}^{K} \pi_k(\mathbf{w}; \boldsymbol{x}_i) \text{Mult}(y_i; \varphi_{k1}(\boldsymbol{x}_i; \boldsymbol{\beta}_k), \ldots, \varphi_{kR}(\boldsymbol{x}_i; \boldsymbol{\beta}_k))\Big] \\ &- \lambda \sum_{k=1}^{K-1} \sum_{j=1}^{p} |w_{kj}| - \gamma \sum_{k=1}^{K} \sum_{r=1}^{R-1} \sum_{j=1}^{p} |\beta_{krj}|. \end{aligned} \tag{2.108}$$

For parameter estimation, these authors proposed an EM-based algorithm. Especially, for maximizing the penalized Q-function with respect to the gating network parameters in the M-step

$$Q(\mathbf{w}; \boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{[q]} \log \pi_k(\boldsymbol{x}_i; \mathbf{w}) - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1, \tag{2.109}$$

where $\tau_{ik}^{[q]}$ is the conditional probability that the data pair $(\boldsymbol{x}_i, y_i)$ is generated by the kth expert with respect to the current parameter vector $\boldsymbol{\theta}^{[q]}$, they approximate (2.109) by using a transformation that implies inverting the softmax function. Hence, updating $\mathbf{w}_k^{[q+1]}$ can be obtained by solving an approximated Lasso problem

$$\mathbf{w}_k^{[q+1]} = \arg\min_{\mathbf{w}_k} \sum_{i=1}^{n} (\log \tau_{ik}^{[q]} - w_{k0} - \boldsymbol{x}_i^T \boldsymbol{w}_k)^2, \text{ subject to } \|\boldsymbol{w}_k\|_1 \leq \lambda. \tag{2.110}$$

Unfortunately, this approximation does not guarantee that the penalized log-likelihood will increase after each update step. In is worth to noting that, the authors do not provide any evidence to prove this critical property of their EM algorithm.

Recently, to overcome the difficulty of updating the gating network, Jiang et al. (2018) based on the work of Xu et al. (1995) proposed a regularization method for the localized MoE models. Localized MoE assumes the latent discrete variable Z follows a multinomial distribution with

$$\mathbb{P}(Z = k) = \pi_k, \ k = 1, \ldots, K \text{ and } \sum_{k=1}^{K} \pi_k = 1.$$

Moreover, given the label variable $Z = k$, the random variate $\boldsymbol{X}$ follows a multivariate Gaussian distribution, i.e.,

$$\boldsymbol{X}|Z = k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

Then the estimation of localized MoE is based on joint distribution $p(\boldsymbol{X}, Y)$, instead of the conditional distribution $p(Y|\boldsymbol{X})$ in a standard MoE. The joint distribution $p(\boldsymbol{X}, Y)$ is as follows

$$\begin{aligned} p(\boldsymbol{X}, Y) &= \sum_{k=1}^{K} p(\boldsymbol{X}, Y, Z = k) \\ &= \sum_{k=1}^{K} p(Z = k) p(\boldsymbol{X}|Z = k) p(Y|\boldsymbol{X}, Z = k) \\ &= \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{X}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \mathcal{N}(y; \beta_{k0} + \boldsymbol{x}^T \boldsymbol{\beta}_k, \sigma_k^2). \end{aligned} \tag{2.111}$$

Hence, the mixing proportion $\mathbb{P}(Z = k|\boldsymbol{X} = \boldsymbol{x})$ has a different form as

$$\begin{aligned} \mathbb{P}(Z = k|\boldsymbol{X} = \boldsymbol{x}) &= \frac{\mathbb{P}(Z = k)\mathbb{P}(\boldsymbol{X} = \boldsymbol{x}|Z = k)}{\sum_{l=1}^{K} \mathbb{P}(Z = l)\mathbb{P}(\boldsymbol{X} = \boldsymbol{x}|Z = l)} \\ &= \frac{\pi_k \mathcal{N}(\boldsymbol{X}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\boldsymbol{X}; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}. \end{aligned} \tag{2.112}$$

Maximum likelihood estimator is done on the joint distribution

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log\Big\{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k, \sigma_k^2)\Big\}, \tag{2.113}$$

where $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_{K-1}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K, (\beta_{10}, \boldsymbol{\beta}_1), \ldots, (\beta_{K0}, \boldsymbol{\beta}_K), \sigma_1, \ldots, \sigma_K)$ is the parameter vector of the model. An EM algorithm for maximizing $L(\boldsymbol{\theta})$ in (2.113) can be referred to Xu et al. (1995), and a very recent algorithm for the regularized estimation is presented in Chamroukhi et al. (2019b).

In order to simultaneously select the number of experts and influential variables of these experts, Jiang et al. (2018) consider the following penalized log-likelihood function

$$PL(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) - P_1(\boldsymbol{\pi}) - P_2(\boldsymbol{\beta}),$$

where the first penalty $P_1(.)$ is served to select the number of component. Following Huang et al. (2017), they consider

$$P_1(\boldsymbol{\pi}) = n\lambda \sum_{k=1}^{K} \big[\log(\epsilon + p_{\lambda, a_1}(\pi_k)) - \log(\epsilon)\big],$$

and $p_{\lambda,a_1}(\pi_k))$ is the SCAD penalty with the pair (λ, a_1). $P_2(\boldsymbol{\beta})$ is also a SCAD penalty for variable selection

$$P_2(\boldsymbol{\beta}) = \sum_{k=1}^{K}\sum_{j=1}^{p} p_{\gamma,a_2}(|\beta_{kj}|),$$

with $p_{\gamma,a_2}(.)$ is the SCAD penalty with the pair (γ, a_2).

It is important to mention that, in this regularization method, each expert's covariance matrix $\boldsymbol{\Sigma}_k$ still updates as in normal. However, for a large value of the dimension p this task becomes a challenge. An efficient method for estimating covariance matrices $\boldsymbol{\Sigma}_k$ should be considered. In a similar way, Morvan et al. (2019) considered a penalized likelihood method for localized MoE models with logistic expert. Here, they assume that Y being a binary response in $\{0, 1\}$ and satisfies

$$Y|\boldsymbol{X} = \boldsymbol{x}, Z = k \sim \mathcal{B}\left(1, \frac{\exp(\beta_{k0} + \boldsymbol{x}^T\boldsymbol{\beta}_k)}{1 + \exp(\beta_{k0} + \boldsymbol{x}^T\boldsymbol{\beta}_k)}\right).$$

In their proposed model, an ℓ_1-regularized estimation strategy is adopted to obtain sparse estimates of both the precision matrices $\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_K$ of the predictors and the regression coefficients $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$, where $\boldsymbol{\Theta}_k = \boldsymbol{\Sigma}_k^{-1}$. Therefore, the penalized log-likelihood function is given by

$$PL(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) - \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1 - \sum_{k=1}^{K} \rho_k \|\boldsymbol{\Theta}_k\|_1,$$

where $\|\boldsymbol{\beta}_k\|_1 = \sum_{j=1}^{p} |\beta_{kj}|$ and $\|\boldsymbol{\Theta}_k\|_1$ denotes the sum of the absolute values of $\boldsymbol{\Theta}_k$. Note that, the penalty on the regression coefficients is mainly for variable selection, while that for the covariance matrices is necessary to have a sparse presentation of the inverse covariance matrices. However, this leads to a difficult problem in estimating these matrices, which are said to be positive-definite.

Note that, besides the regularization methods, it is also interesting to present the works of Deleforge et al. (2015); Perthame et al. (2018), who proposed the hybrid Gaussian locally-linear mapping (hybrid-GLLiM) model which relates to the localized MoE model and can be applied in high dimension with multivariate output.

2.6 Conclusion

In this chapter, we reviewed FMMs used for modeling, clustering and regression for heterogeneous data. We also consider the EM algorithms with are widely use to estimate the parameters of these models.

In high-dimensional scenario, several approaches have been proposed for model-based clustering with FMMs. Classical methods such as parsimonious Gaussian models lay a lot of assumptions on the data and with many submodels it becomes a difficult problem to decide which

model for a given data set. Also, they do not take into account that the structure of association among the variables can vary from one cluster to the other. Mixture of factor analyzers is another ways to deal with high-dimensional data. Unfortunately, this method does not focus on the feature selection task which is one of the main missions in high-dimensional setting. In such a problem, variable selection methods such as SRUW model, penalized models seem to be potential approaches. However, there are some aspects should be study more such as an efficient method to have a sparse representation of the covariance matrices. Recently, Fop et al. (2019) proposed an interesting method to follow in order to directly obtain sparse component covariance matrices.

For the regression problems, we can see that regularization methods for feature selection seem to be very promising in a MoE context. However, there are some open questions that should be answer. In the theoretical point of view, the statistical properties of the methods in high-dimensional situations require more advanced theoretical developments, which is an interesting subject. From the application point of view, estimating the parameters using the EM algorithm requires a lot of advanced results from mathematical optimization, especially for solving the M-step with non-convex penalty functions. Some common methods aim at approximating the penalty function and combine with the Newton-Raphson procedure to update the parameters. Unfortunately, Newton-Raphson procedure is not an appropriate choice in high-dimensional setting. Regularization methods for localized MoE are interesting approaches to avoid updating the gating network parameters in MoE. However, estimating the component covariance matrices $\mathbf{\Sigma}_k$ in high-dimensional situation should be studied more. In this context, a regularization approach likes Morvan et al. (2019) is an interesting suggestion. Another way to tackle this difficult is by considering the parsimonious Gaussian models (Banfield and Raftery, 1993; Celeux and Govaert, 1995) to parametrize the covariance matrices $\mathbf{\Sigma}_k$ or to represent sparse covariance matrices via graph models (Fop et al., 2019). Besides that, updating the experts network parameters with nonconvex penalty functions is not an easy task. One important thing should be reminded is that, to the best of our knowledge, until now there is no theoretical to support the regularization methods for localized MoE. Hence, establishing the statistical properties of the methods is the subject of future research.

Chapter 3

Regularized estimation and feature selection in mixture of experts for regression and clustering

Contents

3.1 Introduction

In the previous chapter, we studied the Mixture of experts (MoE) introduced by Jacobs et al. (1991) for regression, clustering and classification. MoE belong to the family of mixture models (Titterington et al., 1985; McLachlan and Peel., 2000; Frühwirth-Schnatter, 2006) and is a fully conditional mixture model where both the mixing proportions, i.e, the gating network, and the components densities, i.e, the experts network, depend on the inputs. A general review of the MoE models and their applications can be found in Nguyen and Chamroukhi (2018). While the MoE modeling with maximum likelihood estimation (MLE) is widely used, its application in high-dimensional problems is still challenging due to the well-known problem of the ML estimator in such a setting. Indeed, in high-dimensional setting, the features can be correlated and thus the actual features that explain the problem reside in a low-dimensional space. Hence, there is a need to select a subset of the potentially large number of features, that really explain the data. To avoid singularities and degeneracies of the MLE as highlighted namely in Stephens and Phil (1997); Snoussi and Mohammad-Djafari (2005); Fraley and Raftery (2007), one can regularize the likelihood through a prior distribution over the model parameter space. A better fitting can therefore be achieved by regularizing the objective function so that to encourage sparse solutions. However, feature selection by regularized inference encourages sparse solutions, while having a reasonable computational cost. Several approaches have been proposed to deal with the feature selection task, both in regression and in clustering.

For regression, the well-known Lasso method (Tibshirani, 1996) is one of the most popular and successful regularization technique which utilizes the ℓ_1 penalty to regularize the squared error function, or by equivalence the log-likelihood in Gaussian regression, and to achieve parameter estimation and feature selection. This allows to shrink coefficients toward zero, and can also set many coefficients to be exactly zero.

In related mixture models for simultaneous regression and clustering, including mixture of linear regressions (MLR), where the mixing proportions are constant, Khalili and Chen (2007) proposed regularized ML inference, including MIXLASSO, MIXHARD and MIXSCAD and provided asymptotic properties corresponding to these penalty functions. Later Khalili (2010) extended his MLR regularization to the MoE setting, and provided a root-n consistent and oracle properties for Lasso and SCAD penalties, and developed an EM algorithm for fitting the models. However, as we will discuss it in Section 3.3, this is based on approximated penalty function, and uses a Newton-Raphson procedure in the updates of the gating network parameters, and thus requires matrix inversion.

In this chapter, we consider and improve the previous regularized MoE models from Khalili (2010). We propose a new regularized maximum likelihood estimation approach with three hybrid algorithms for maximizing the proposed objective function. The proposed algorithms for fitting the model consist of an Expectation-Majorization-Maximization (EMM) algorithm, an EM algorithm with a coordinate ascent algorithm and an EM algorithm with proximal Newton-type method. The proposed approach does not require an approximate of the regularization

term, and the three developed hybrid algorithms, allow to automatically select sparse solutions without thresholding. In addition, an evaluation of the effectiveness of an ℓ_2 penalty for the gating network in Khalili (2010) is also considered.

The remainder of this chapter is organized as follows. In Section 3.2 we present the modeling with MoE for heterogeneous data. Then, in Section 3.3, we present the regularized maximum likelihood strategy of the MoE model, and the three proposed EM-based algorithms. An experimental study, carried out on simulated and two real data sets, are given in Section 3.4. In Section 3.4.6, we discuss the effectiveness of our method in dealing with moderate dimensional problems, and consider an experiment which promotes its use in high-dimensional scenarios. Finally, in Section 3.5, we draw concluding remarks and mention future direction.

3.2 Mixture of experts model for continuous data

Let $((\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n))$ be a random sample of n independently distributed pairs $(\boldsymbol{X}_i, Y_i)$, $(i = 1, \ldots, n)$ where $Y_i \in \mathcal{Y} \subset \mathbb{R}$ is the ith response given some vector of predictors $\boldsymbol{X}_i \in \mathcal{X} \subset \mathbb{R}^p$. Let $\mathcal{D} = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n))$ be an observed data sample. These data may be discrete or continuous. In this chapter, we emphasize on the continuous case and consider the MoE modeling in Section 2.3.3 for the analysis of a heterogeneous set of such data.

3.2.1 The model

The mixture of experts model assumes that the observed pairs $(\boldsymbol{x}, y)$ are generated from $K \in \mathbb{N}$ probability density components (the experts) governed by a hidden categorical random variable $Z \in [K] = \{1, \ldots, K\}$ that indicates the component from which a particular observed pair is drawn. The latter represents the gating network. Formally, the gating network is defined by the distribution of the hidden variable Z given the predictor $\boldsymbol{x}$, i.e., $\pi_k(\boldsymbol{x}; \mathbf{w}) = \mathbb{P}(Z = k|\boldsymbol{X} = \boldsymbol{x}; \mathbf{w})$, which is in general given by gating softmax functions of the form:

$$\pi_k(\boldsymbol{x}; \mathbf{w}) = \mathbb{P}(Z = k|\boldsymbol{X} = \boldsymbol{x}; \mathbf{w}) = \frac{\exp(w_{k0} + \boldsymbol{x}^T \boldsymbol{w}_k)}{1 + \sum_{l=1}^{K-1} \exp(w_{l0} + \boldsymbol{x}^T \boldsymbol{w}_l)} \tag{3.1}$$

for $k = 1, \ldots, K-1$ with $\mathbf{w}_k^T = (w_{k0}, \boldsymbol{w}_k^T) \in \mathbb{R}^{p+1}$ and $\mathbf{w}_K^T = (w_{K0}, \boldsymbol{w}_K^T) = (0, \mathbf{0})$ for identifiability (Jiang and Tanner, 1999a). The experts network is defined by the conditional densities $f(y|\boldsymbol{x}; \boldsymbol{\theta}_k)$ which is the short notation of $f(y|\boldsymbol{X} = \boldsymbol{x}, Z = k; \boldsymbol{\theta})$. Specially, for the regression problem in this chapter, the expert network is modeled by the noisy linear model:

$$f(y|\boldsymbol{x}; \boldsymbol{\theta}_k) = \mathcal{N}(y; \beta_{k0} + \boldsymbol{x}^T \boldsymbol{\beta}_k, \sigma_k^2),$$

where the bias coefficient $\beta_{k0} \in \mathbb{R}$ and $\boldsymbol{\beta}_k \in \mathbb{R}^p$ are the regression coefficients describing the kth expert, and $\sigma_k > 0$ corresponds to the standard deviation of the noise. Thus, in a MoE model for Gaussian regression with K components, the conditional density function of Y given $\boldsymbol{x}$ can

be defined by the following semi-parametric probability density function:

$$\begin{aligned} f(y|\boldsymbol{x};\boldsymbol{\theta}) &= \sum_{k=1}^{K} \pi_k(\boldsymbol{x};\mathbf{w}) f(y|\boldsymbol{x};\boldsymbol{\theta}_k) \\ &= \sum_{k=1}^{K} \pi_k(\boldsymbol{x};\mathbf{w}) \mathcal{N}(y;\beta_{k0}+\boldsymbol{x}^T\boldsymbol{\beta}_k,\sigma_k^2) \end{aligned} \tag{3.2}$$

that is parameterized by the parameter vector defined by $\boldsymbol{\theta} = (\mathbf{w}_1^T,\ldots,\mathbf{w}_{K-1}^T,\boldsymbol{\theta}_1^T,\ldots,\boldsymbol{\theta}_K^T)^T$ where $\boldsymbol{\theta}_k = (\beta_{k0},\boldsymbol{\beta}_k^T,\sigma_k^2)^T$ $(k=1,\ldots,K)$ is the parameter vector of the kth expert.

3.2.2 Maximum likelihood parameter estimation

Assume that, $\mathcal{D} = ((\boldsymbol{x}_1,y_1),\ldots,(\boldsymbol{x}_n,y_n))$ is an observed data sample generated from the MoE (3.2) with unknown parameter $\boldsymbol{\theta}$. The parameter vector $\boldsymbol{\theta}$ is commonly estimated by maximizing the observed data log-likelihood $L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k(\boldsymbol{x}_i;\mathbf{w}) f(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_k)$ by using the EM algorithm which allows to iteratively find an appropriate local maximizer of the log-likelihood function (see Section 2.3.3 and Dempster et al. (1977); Jacobs et al. (1991) for more details). In the considered model for Gaussian regression, the log-likelihood function is given by

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log\Big[\sum_{k=1}^{K} \pi_k(\boldsymbol{x}_i;\mathbf{w}) \mathcal{N}(y_i;\beta_{k0}+\boldsymbol{\beta}_k^T\boldsymbol{x}_i,\sigma_k^2)\Big]. \tag{3.3}$$

However, it is well-known that the MLE may be unstable of even infeasible in high-dimension namely due to possibly redundant and correlated features. In such a context, a regularization of the MLE is needed.

3.3 Regularized maximum likelihood parameter estimation of the MoE

Regularized maximum likelihood estimation allows the selection of a relevant subset of features for prediction and thus encourages sparse solutions. In mixture of experts modeling, one may consider both sparsity in the feature space of the gates, and of the experts. We propose to infer the MoE model by maximizing a regularized log-likelihood criterion, which encourages sparsity for both the gating network parameters and the experts network parameters, and does not require any approximation, along with performing the maximization, so that to avoid matrix inversion. The proposed regularization combines a Lasso penalty for the experts parameters, and an elastic net like penalty for the gating network, defined by:

$$PL(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) - \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1 - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 - \frac{\rho}{2} \sum_{k=1}^{K-1} \|\boldsymbol{w}_k\|_2^2. \tag{3.4}$$

A similar strategy has been proposed in Khalili (2010) where the author proposed a regularized ML function like (3.4) but which is then approximated in the EM algorithm of the model inference. The EM algorithm for fitting the model follows indeed the suggestion of Fan and Li (2001) to approximate the penalty function in a some neighborhood by a local quadratic function. Therefore, a Newton-Raphson can be used to update parameters in the M-step. The weakness of this scheme is that once a feature is set to zero, it may never reenter the model at a later stage of the algorithm. To avoid this numerical instability of the algorithm due to the small values of some of the features in the denominator of this approximation, Khalili (2010) replaced that approximation by an ϵ-local quadratic function. Unfortunately, these strategies have some drawbacks. First, by approximating the penalty functions with (ϵ-)quadratic functions, none of the components will be exactly zero. Hence, a threshold should be considered to declare a coefficient is zero, and this threshold affects the degree of sparsity. Secondly, using Newton-Raphson procedure for maximizing a concave function with large dimension p is not an appropriate choice related to the required Hessian matrix inversion. In addition, one has also to choose ϵ as an additional tuning parameter in practice. Our proposal overcomes these limitations.

The ℓ_2 term penalty is added in our model to take into account possible strong correlation between the features x_j which could be translated especially on the coefficients of the gating network $\mathbf{w}$ because they are related between the different experts, contrary to the regression coefficients $\boldsymbol{\beta}$. The resulting combination of ℓ_1 and ℓ_2 for $\mathbf{w}$ leads to an elastic net-like regularization, which enjoys similar sparsity of representation as the ℓ_1 penalty. The ℓ_2 term is not however essential especially when the main goal is to retrieve the sparsity, rather than to perform prediction.

3.3.1 Parameter estimation and feature selection with a dedicated block-wise EM

We propose three block-wise EM algorithms to monotonically find at least local maximizers of (3.4). The E-step is common to these algorithms, while in the M-step, three different algorithms are proposed to update the model parameters. More specifically, the first one relies on a MM algorithm, and the second one uses a coordinate ascent to update the gating network $\mathbf{w}$ parameters. Meanwhile, the last one uses proximal Newton-type procedure to update $\mathbf{w}$. All these algorithm use the same coordinate ascent method to update the experts network $\boldsymbol{\beta}$' parameters. The EM algorithm for the maximization of (3.4) firstly requires the construction of the penalized complete-data log-likelihood

$$PL_c(\boldsymbol{\theta}) = L_c(\boldsymbol{\theta}) - \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1 - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 - \frac{\rho}{2} \sum_{k=1}^{K-1} \|\boldsymbol{w}_k\|_2^2 \tag{3.5}$$

where $L_c(\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log \left[\pi_k(\boldsymbol{x}_i; \mathbf{w}) f(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k)\right]$ is the standard complete-data log-likelihood, z_{ik} is an indicator binary-valued variable such that $z_{ik} = 1$ if $Z_i = k$ (i.e., if the ith pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$

is generated from the kth expert component) and $z_{ik} = 0$ otherwise. Thus, the EM algorithm for the RMoE in its general form runs as follows. After starting with an initial solution $\boldsymbol{\theta}^{[0]}$, it alternates between the two following steps until convergence (e.g., when there is no longer a significant change in the relative variation of the regularized log-likelihood).

3.3.2 E-step

The E-Step computes the conditional expectation of the penalized complete-data log-likelihood (3.5), given the observed data $\mathcal{D}$ under the current parameter vector $\boldsymbol{\theta}^{[q]}$, q being the current iteration number of the block-wise EM algorithm:

$$\begin{aligned} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{[q]}) &= \mathbb{E}\left[PL_c(\boldsymbol{\theta})|\mathcal{D}; \boldsymbol{\theta}^{[q]}\right] \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}^{[q]} \log\left[\pi_k(\boldsymbol{x}_i; \mathbf{w}) f_k(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k)\right] - \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1 - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 - \frac{\rho}{2}\sum_{k=1}^{K-1} \|\boldsymbol{w}_k\|_2^2 \end{aligned} \tag{3.6}$$

where

$$\tau_{ik}^{[q]} = \mathbb{P}(Z_i = k|y_i, \boldsymbol{x}_i; \boldsymbol{\theta}^{[q]}) = \frac{\pi_k(\boldsymbol{x}_i; \mathbf{w}^{[q]})\mathcal{N}(y_i; \beta_{k0}^{[q]} + \boldsymbol{x}_i^T\boldsymbol{\beta}_k^{[q]}, \sigma_k^{[q]2})}{\sum_{l=1}^{K} \pi_l(\boldsymbol{x}_i; \mathbf{w}^{[q]})\mathcal{N}(y_i; \beta_{l0}^{[q]} + \boldsymbol{x}_i^T\boldsymbol{\beta}_l^{[q]}, \sigma_l^{[q]2})} \tag{3.7}$$

is the conditional probability that the data pair $(\boldsymbol{x}_i, y_i)$ is generated by the kth expert. This step therefore only requires the computation of the conditional component probabilities $\tau_{ik}^{[q]}$ $(i = 1, \ldots, n)$ for each of the K experts.

3.3.3 M-step

The M-Step updates the parameters by maximizing the Q function (3.6), which can be written as

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{[q]}) = Q(\mathbf{w}; \boldsymbol{\theta}^{[q]}) + Q(\boldsymbol{\beta}, \sigma; \boldsymbol{\theta}^{[q]}) \tag{3.8}$$

with

$$Q(\mathbf{w}; \boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}^{[q]} \log \pi_k(\boldsymbol{x}_i; \mathbf{w}) - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 - \frac{\rho}{2}\sum_{k=1}^{K-1} \|\boldsymbol{w}_k\|_2^2, \tag{3.9}$$

and

$$Q(\boldsymbol{\beta}, \sigma; \boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}^{[q]} \log \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{x}_i^T\boldsymbol{\beta}_k, \sigma_k^2) - \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1. \tag{3.10}$$

The parameters $\mathbf{w}$ are therefore separately updated by maximizing the function

$$Q(\mathbf{w}; \boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n}\sum_{k=1}^{K-1} \tau_{ik}^{[q]}(w_{k0} + \boldsymbol{x}_i^T\boldsymbol{w}_k) - \sum_{i=1}^{n} \log\Big[1 + \sum_{k=1}^{K-1} e^{w_{k0} + \boldsymbol{x}_i^T\boldsymbol{w}_k}\Big] - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 - \frac{\rho}{2}\sum_{k=1}^{K-1} \|\boldsymbol{w}_k\|_2^2. \tag{3.11}$$

We propose and compare three approaches for maximizing (3.9) based on a MM algorithm, a coordinate ascent algorithm and proximal Newton-type method. These approaches have some advantages since they do not use any approximate for the penalty function, and have a separate structure which avoid matrix inversion.

MM algorithm for updating the gating network

In this part, we construct a MM algorithm to iteratively update the gating network parameters $(w_{k0}, \boldsymbol{w}_k)$. At each iteration step s of the MM algorithm, we maximize a minorizing function of the initial function (3.9). We begin this task by giving the definition of a minorizing function.

Definition 3.3.1. *(see Lange (2013)) Let $F(x)$ be a function of x. A function $G(x|x_m)$ is called a minorizing function of $F(x)$ at x_m if and only if*

$$F(x) \geq G(x|x_m) \text{ and } F(x_m) = G(x_m|x_m), \ \forall x.$$

In the maximization step of the MM algorithm, we maximize the surrogate function $G(x|x_m)$, rather than the function $F(x)$ itself. If x_{m+1} is the maximum of $G(x|x_m)$, then we can show that the MM algorithm forces $F(x)$ uphill, because

$$F(x_m) = G(x_m|x_m) \leq G(x_{m+1}|x_m) \leq F(x_{m+1}).$$

By doing so, we can find a local maximizer of $F(x)$. If $G(x_m|x_m)$ is well constructed, then we can avoid matrix inversion when maximizing it. Next, we derive the surrogate function for $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$. We start by the following lemma.

Lemma 3.3.1. *If $x > 0$, then the function $f(x) = -\ln(1+x)$ can be minorized by*

$$g(x|x_m) = -\ln(1+x_m) - \frac{x - x_m}{1+x_m}, \quad \text{at } x_m > 0.$$

By applying this lemma and following (Lange, 2013, page 211) we have

Theorem 3.3.1. *The function $I_1(\mathbf{w}) = -\sum_{i=1}^{n} \log\Big[1 + \sum_{k=1}^{K-1} e^{w_{k0} + \boldsymbol{x}_i^T \boldsymbol{w}_k}\Big]$ is a majorizer of*

$$G_1(\mathbf{w}|\mathbf{w}^{[s]}) = \sum_{i=1}^{n}\Big[-\sum_{k=1}^{K-1} \frac{\pi_k(\boldsymbol{x}_i; \mathbf{w}^{[s]})}{p+1} \sum_{j=0}^{p} e^{(p+1)x_{ij}(w_{kj} - w_{kj}^{[s]})} - \log C_i^m + 1 - \frac{1}{C_i^m}\Big],$$

where $C_i^m = 1 + \sum_{k=1}^{K-1} e^{w_{k0}^{[s]} + \boldsymbol{x}_i^T \boldsymbol{w}_k^{[s]}}$ and $x_{i0} = 1$.

Proof. Using Lemma 3.3.1, $I_{1i}(\mathbf{w}) = -\log\Big[1 + \sum_{k=1}^{K-1} e^{w_{k0}+\boldsymbol{x}_i^T\boldsymbol{w}_k}\Big]$ can be minorized by

$$\begin{aligned} G_i(\mathbf{w}|\mathbf{w}^{[s]}) &= -\log\Big[1 + \sum_{k=1}^{K-1} e^{w_{k0}^{[s]}+\boldsymbol{x}_i^T\boldsymbol{w}_k^{[s]}}\Big] - \frac{\sum_{k=1}^{K-1}(e^{w_{k0}+\boldsymbol{x}_i^T\boldsymbol{w}_k} - e^{w_{k0}^{[s]}+\boldsymbol{x}_i^T\boldsymbol{w}_k^{[s]}})}{1 + \sum_{k=1}^{K-1} e^{w_{k0}^{[s]}+\boldsymbol{x}_i^T\boldsymbol{w}_k^{[s]}}} \\ &= -\log C_i^m + 1 - \frac{1}{C_i^m} - \sum_{k=1}^{K-1} \frac{e^{w_{k0}^{[s]}+\boldsymbol{x}_i^T\boldsymbol{w}_k^{[s]}}}{C_i^m} e^{(w_{k0}+\boldsymbol{x}_i^T\boldsymbol{w}_k)-(w_{k0}^{[s]}+\boldsymbol{x}_i^T\boldsymbol{w}_k^{[s]})}. \end{aligned}$$

Now, by using arithmetic-geometric mean inequality then

$$e^{(w_{k0}+\boldsymbol{x}_i^T\boldsymbol{w}_k)-(w_{k0}^{[s]}+\boldsymbol{x}_i^T\boldsymbol{w}_k^{[s]})} = \prod_{j=0}^{p} e^{x_{ij}(w_{kj}-w_{kj}^{[s]})} \leq \frac{\sum_{j=0}^{p} e^{(p+1)x_{ij}(w_{kj}-w_{kj}^{[s]})}}{p+1}. \tag{3.12}$$

When $(w_{k0}, \boldsymbol{w}_k) = (w_{k0}^{[s]}, \boldsymbol{w}_k^{[s]})$ the equality holds.
Thus, $I_{1i}(\mathbf{w})$ can be minorized by

$$\begin{aligned} G_{1i}(\mathbf{w}|\mathbf{w}^{[s]}) &= -\sum_{k=1}^{K-1} \frac{e^{w_{k0}^{[s]}+\boldsymbol{x}_i^T\boldsymbol{w}_k^{[s]}}}{(p+1)C_i^m} \sum_{j=0}^{p} e^{(p+1)x_{ij}(w_{kj}-w_{kj}^{[s]})} - \log C_i^m + 1 - \frac{1}{C_i^m} \\ &= -\sum_{k=1}^{K-1} \frac{\pi_k(\boldsymbol{x}_i;\mathbf{w}^{[s]})}{p+1} \sum_{j=0}^{p} e^{(p+1)x_{ij}(w_{kj}-w_{kj}^{[s]})} - \log C_i^m + 1 - \frac{1}{C_i^m}. \end{aligned}$$

This leads us to the minorizing function $G_1(\mathbf{w}|\mathbf{w}^{[s]})$ for $I_1(\mathbf{w})$

$$G_1(\mathbf{w}|\mathbf{w}^{[s]}) = \sum_{i=1}^{n}\Big[-\sum_{k=1}^{K-1} \frac{\pi_k(\boldsymbol{x}_i;\mathbf{w}^{[s]})}{p+1} \sum_{j=0}^{p} e^{(p+1)x_{ij}(w_{kj}-w_{kj}^{[s]})} - \log C_i^m + 1 - \frac{1}{C_i^m}\Big].$$

□

Therefore, the minorizing function $G^{[q]}(\mathbf{w}|\mathbf{w}^{[s]})$ for $Q(\mathbf{w};\boldsymbol{\theta}^{[q]})$ is given by

$$G^{[q]}(\mathbf{w}|\mathbf{w}^{[s]}) = \sum_{i=1}^{n}\sum_{k=1}^{K-1} \tau_{ik}^{[q]}(w_{k0} + \boldsymbol{x}_i^T\boldsymbol{w}_k) + G_1(\mathbf{w}|\mathbf{w}^{[s]}) - \sum_{k=1}^{K-1}\gamma_k \sum_{j=1}^{p}|w_{kj}| - \frac{\rho}{2}\sum_{k=1}^{K-1}\sum_{j=1}^{p} w_{kj}^2.$$

Now, let us separate $G^{[q]}(\mathbf{w}|\mathbf{w}^{[s]})$ into each parameter for all $k \in \{1,\ldots,K-1\},\ j \in \{1,\ldots,p\}$, we have:

$$G^{[q]}(w_{k0}|\mathbf{w}^{[s]}) = \sum_{i=1}^{n} \tau_{ik}^{[q]} w_{k0} - \sum_{i=1}^{n} \frac{\pi_k(\boldsymbol{x}_i;\mathbf{w}^{[s]})}{p+1} e^{(p+1)(w_{k0}-w_{k0}^{[s]})} + A_k(\mathbf{w}^{[s]}), \tag{3.13}$$

and

$$G^{[q]}(w_{kj}|\mathbf{w}^{[s]}) = \sum_{i=1}^{n} \tau_{ik}^{[q]} x_{ij} w_{kj} - \sum_{i=1}^{n} \frac{\pi_k(\boldsymbol{x}_i; \mathbf{w}^{[s]})}{p+1} e^{(p+1)x_{ij}(w_{kj} - w_{kj}^{[s]})} - \gamma_k |w_{kj}| - \frac{\rho}{2} w_{kj}^2 + B_{kj}(\mathbf{w}^{[s]}), \tag{3.14}$$

where $A_k(\mathbf{w}^{[s]})$ and $B_{kj}(\mathbf{w}^{[s]})$ are only functions of $\mathbf{w}^{[s]}$.
The update of $w_{k0}^{[s]}$ is straightforward by maximizing (3.13) and given by

$$w_{k0}^{[s+1]} = w_{k0}^{[s]} + \frac{1}{p+1} \ln \left(\frac{\sum_{i=1}^{n} \tau_{ik}^{[q]}}{\sum_{i=1}^{n} \pi_k(\boldsymbol{x}_i; \mathbf{w}^{[s]})} \right). \tag{3.15}$$

The function $G^{[q]}(w_{kj}|\mathbf{w}^{[s]})$ is a concave function. Moreover, it is a univariate function with respect to w_{kj}. We can therefore maximize it globally and with respect to each coeffcient w_{kj} separately and thus avoid matrix inversion. Indeed, let us denote by

$$F_{kjm}^{[q]}(w_{kj}) = \sum_{i=1}^{n} \tau_{ik}^{[q]} x_{ij} w_{kj} - \sum_{i=1}^{n} \frac{\pi_k(\boldsymbol{x}_i; \mathbf{w}^{[s]})}{p+1} e^{(p+1)x_{ij}(w_{kj} - w_{kj}^{[s]})} - \frac{\rho}{2} w_{kj}^2 + B_{kj}(\mathbf{w}^{[s]}),$$

hence, $G^{[q]}(w_{kj}|\mathbf{w}^{[s]})$ can be rewritten as

$$G^{[q]}(w_{kj}|\mathbf{w}^{[s]}) = \begin{cases} F_{kjm}^{[q]}(w_{kj}) - \gamma_k w_{kj} & , \text{ if } w_{kj} > 0 \\ F_{kjm}^{[q]}(0) & , \text{ if } w_{kj} = 0 \\ F_{kjm}^{[q]}(w_{kj}) + \gamma_k w_{kj} & , \text{ if } w_{kj} < 0 \end{cases}.$$

We therefore have both $F_{kjm}^{[q]}(w_{kj}) - \gamma_k w_{kj}$ and $F_{kjm}^{[q]}(w_{kj}) + \gamma_k w_{kj}$ are smooth concave functions. Thus, one can use one-dimensional Newton-Raphson algorithm to find the global maximizers of these functions and compare with $F_{kjm}^{[q]}(0)$ in order to update $w_{kj}^{[s]}$ by

$$w_{kj}^{[s+1]} = \arg\max_{w_{kj}} G^{[q]}(w_{kj}|\mathbf{w}^{[s]}).$$

The update of w_{kj} can then be computed by a one-dimensional generalized Newton-Raphson (NR) algorithm, which updates, after starting from and initial value $w_{kj}^{[0]} = w_{kj}^{[s]}$, at each iteration t of the NR, according to the following updating rule:

$$w_{kj}^{[t+1]} = w_{kj}^{[t]} - \left(\frac{\partial^2 G^{[q]}(w_{kj}|\mathbf{w}^{[s]})}{\partial^2 w_{kj}} \right)^{-1} \Big|_{w_{kj}^{[t]}} \frac{\partial G^{[q]}(w_{kj}|\mathbf{w}^{[s]})}{\partial w_{kj}} \Big|_{w_{kj}^{[t]}},$$

where the first and the scalar gradient and hessian are respectively given by:

$$\frac{\partial G^{[q]}(w_{kj}|\mathbf{w}^{[s]})}{\partial w_{kj}} = \begin{cases} U(w_{kj}) - \gamma_k & , G^{[q]}(w_{kj}|\mathbf{w}^{[s]}) = F_{kjm}^{[q]}(w_{kj}) - \gamma_k w_{kj} \\ U(w_{kj}) + \gamma_k & , G^{[q]}(w_{kj}|\mathbf{w}^{[s]}) = F_{kjm}^{[q]}(w_{kj}) + \gamma_k w_{kj} \end{cases},$$

and

$$\frac{\partial^2 G^{[q]}(w_{kj}|\mathbf{w}^{[s]})}{\partial^2 w_{kj}} = -(p+1)\sum_{i=1}^{n} x_{ij}^2 \pi_k(\boldsymbol{x}_i;\mathbf{w}^{[s]}) e^{(p+1)x_{ij}(w_{kj}-w_{kj}^{[s]})} - \rho,$$

with

$$U(w_{kj}) = \sum_{i=1}^{n} \tau_{ik}^{[q]} x_{ij} - \sum_{i=1}^{n} x_{ij}\pi_k(\boldsymbol{x}_i;\mathbf{w}^{[s]}) e^{(p+1)x_{ij}(w_{kj}-w_{kj}^{[s]})} - \rho w_{kj}.$$

Unluckily, while this method allows to compute separate univariate updates by globally maximizing concave functions, it has some drawbacks. First, we found the same behavior of the MM algorithm for this non-smooth function setting as in Hunter and Li (2005): once a coefficient is set to be zero, it may never reenter the model at a later stage of the algorithm. Second, the MM algorithm can stuck on non-optimal points of the objective function. Schifano et al. (2010) made an interesting study on the convergence of the MM algorithms for nonsmoothly penalized objective functions, in which they proof that with some conditions on the minorizing function (see Theorem 2.1 of Schifano et al. (2010)), then the MM algorithm will converge to the optimal value. One of these conditions requires the minorizing function must be strickly positive, which is not guaranteed in our method, since we use the arithmetic-geometric mean inequality in (3.12) to construct our surrogate function. Hence, we just ensure that the value of $Q(\mathbf{w};\boldsymbol{\theta}^{[q]})$ will not decrease in our algorithm. In the next section, we propose updating $\mathbf{w}_k = (w_{k0}, \boldsymbol{w}_k)$ by using coordinate ascent algorithm. This approach overcomes this weakness of the MM algorithm.

Coordinate ascent algorithm for updating the gating network

We now consider another approach for updating $(w_{k0}, \boldsymbol{w}_k)$ by using coordinate ascent algorithm. Indeed, based on Tseng (1988, 2001), with regularity conditions, then the coordinate ascent algorithm is successful in updating $\mathbf{w}$. Thus, the $\mathbf{w}$ parameters are updated in a cyclic way, where a coefficient w_{kj} $(j \neq 0)$ is updated at each time, while fixing the other parameters to their previous values. Hence, at each iteration one just needs to update only one parameter. With this setting, the update of w_{kj} is performed by maximizing the component (k, j) of (3.9) given by

$$Q(w_{kj};\boldsymbol{\theta}^{[q]}) = F(w_{kj};\boldsymbol{\theta}^{[q]}) - \gamma_k|w_{kj}|, \tag{3.16}$$

where

$$F(w_{kj};\boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n} \tau_{ik}^{[q]}(w_{k0} + \boldsymbol{w}_k^T\boldsymbol{x}_i) - \sum_{i=1}^{n}\log\Big[1 + \sum_{l=1}^{K-1} e^{w_{l0}+\boldsymbol{w}_l^T\boldsymbol{x}_i}\Big] - \frac{\rho}{2}w_{kj}^2. \tag{3.17}$$

Hence, $Q(w_{kj};\boldsymbol{\theta}^{[q]})$ can be rewritten as

$$Q(w_{kj};\boldsymbol{\theta}^{[q]}) = \begin{cases} F(w_{kj};\boldsymbol{\theta}^{[q]}) - \gamma_k w_{kj} & , \text{ if } w_{kj} > 0 \\ F(0;\boldsymbol{\theta}^{[q]}) & , \text{ if } w_{kj} = 0 \\ F(w_{kj};\boldsymbol{\theta}^{[q]}) + \gamma_k w_{kj} & , \text{ if } w_{kj} < 0 \end{cases}.$$

Again, both $F(w_{kj};\boldsymbol{\theta}^{[q]})-\gamma_k w_{kj}$ and $F(w_{kj};\boldsymbol{\theta}^{[q]})+\gamma_k w_{kj}$ are smooth concave functions. Thus, one can use one-dimensional generalized Newton-Raphson algorithm with initial value $w_{kj}^{[0]}=w_{kj}^{[q]}$ to find the maximizers of these functions and compare with $F(0;\boldsymbol{\theta}^{[q]})$ in order to update $w_{kj}^{[s]}$ by

$$w_{kj}^{[s+1]}=\arg\max_{w_{kj}} Q(w_{kj};\boldsymbol{\theta}^{[q]}),$$

where s denotes the sth loop of the coordinate ascent algorithm. The update of w_{kj} is therefore computed iteratively after starting from and initial value $w_{kj}^{[0]}=w_{kj}^{[s]}$ following the update equation

$$w_{kj}^{[t+1]}=w_{kj}^{[t]}-\Big(\frac{\partial^2 Q(w_{kj};\boldsymbol{\theta}^{[q]})}{\partial^2 w_{kj}}\Big)^{-1}\Big|_{w_{kj}^{[t]}}\frac{\partial Q(w_{kj};\boldsymbol{\theta}^{[q]})}{\partial w_{kj}}\Big|_{w_{kj}^{[t]}}, \tag{3.18}$$

where t in the inner NR iteration number, and the one-dimensional gradient and hessian functions are respectively given by

$$\frac{\partial Q(w_{kj};\boldsymbol{\theta}^{[q]})}{\partial w_{kj}}=\begin{cases}U(w_{kj})-\gamma_k & \text{, if } Q(w_{kj};\boldsymbol{\theta}^{[q]})=F(w_{kj};\boldsymbol{\theta}^{[q]})-\gamma_k w_{kj}\\ U(w_{kj})+\gamma_k & \text{, if } Q(w_{kj};\boldsymbol{\theta}^{[q]})=F(w_{kj};\boldsymbol{\theta}^{[q]})+\gamma_k w_{kj}\end{cases}, \tag{3.19}$$

and

$$\frac{\partial^2 Q(w_{kj};\boldsymbol{\theta}^{[q]})}{\partial^2 w_{kj}}=-\sum_{i=1}^{n}\frac{x_{ij}^2 e^{w_k 0+x_i^T\boldsymbol{w}_k}(C_i(w_{kj})-e^{w_{k0}+x_i^T\boldsymbol{w}_k})}{C_i^2(w_{kj})}-\rho.$$

with

$$U(w_{kj})=\sum_{i=1}^{n}x_{ij}\tau_{ik}^{[q]}-\sum_{i=1}^{n}\frac{x_{ij}e^{w_{k0}+\boldsymbol{x}_i^T\boldsymbol{w}_k}}{C_i(w_{kj})}-\rho w_{kj},$$

and

$$C_i(w_{kj})=1+\sum_{l\neq k}e^{w_{l0}+x_i^T\boldsymbol{w}_l}+e^{w_{k0}+x_i^T\boldsymbol{w}_k},$$

is a univariate function of w_{kj} when fixing other parameters. For other parameter we set $w_{lh}^{[s+1]}=w_{lh}^{[s]}$.

Similarly, for the update of w_{k0}, a univariate Newton-Raphson algorithm with initial value $w_{k0}^{[0]}=w_{k0}^{[q]}$ can be used to provide the update $w_{k0}^{[s]}$ given by

$$w_{k0}^{[s+1]}=\arg\max_{w_{k0}} Q(w_{k0};\boldsymbol{\theta}^{[q]}),$$

where $Q(w_{k0};\boldsymbol{\theta}^{[q]})$ is a univariate concave function given by

$$Q(w_{k0};\boldsymbol{\theta}^{[q]})=\sum_{i=1}^{n}\tau_{ik}^{[q]}(w_{k0}+\boldsymbol{x}_i^T\boldsymbol{w}_k)-\sum_{i=1}^{n}\log\Big[1+\sum_{l=1}^{K-1}e^{w_{l0}+\boldsymbol{x}_i^T\boldsymbol{w}_l}\Big], \tag{3.20}$$

with

$$\frac{\partial Q(w_{k0};\boldsymbol{\theta}^{[q]})}{\partial w_{k0}}=\sum_{i=1}^{n}\tau_{ik}^{[q]}-\sum_{i=1}^{n}\frac{e^{w_{k0}+\boldsymbol{x}_i^T\boldsymbol{w}_k}}{C_i(w_{k0})} \tag{3.21}$$

and

$$\frac{\partial^2 Q(w_{k0};\boldsymbol{\theta}^{[q]})}{\partial^2 w_{k0}} = -\sum_{i=1}^{n} \frac{e^{w_{k0}+x_i^T\boldsymbol{w}_k}(C_i(w_{k0}) - e^{w_{k0}+x_i^T\boldsymbol{w}_k})}{C_i^2(w_{k0})}. \tag{3.22}$$

The other parameters are fixed while updating w_{k0}. By using the coordinate ascent algorithm, we have univariate updates, and make sure that the parameters w_{kj} may change during the algorithm even after they shrink to zero at an earlier stage of the algorithm.

In the next section, we propose a new procedure for updating the gating network parameters.

Proximal Newton-type procedure for updating the gating network

Indeed, the developed MM and coordinate ascent algorithms for the estimation of the parameters of our model could be slow in a high-dimensional setting since we do not have the closed-form updates of the parameters of the gating network $\mathbf{w}$ at each step of the EM algorithm; while a univariate Newton-Raphson is derived to avoid matrix inversion operations, it is still slow in high-dimension. However, this difficulty can be overcome by proximal Newton-type method. In this part, we propose two approaches for updating the gating network parameters $\mathbf{w} = \{(w_{k0}, \boldsymbol{w}_k)\}$ by maximizing $Q(\mathbf{w};\boldsymbol{\theta}^{[q]})$ based on the proximal Newton and the proximal Newton-type method. The proximal Newton method approximates only the smooth part of (3.9) given by

$$I(\mathbf{w}) = \sum_{i=1}^{n}\sum_{k=1}^{K-1} \tau_{ik}^{[q]}(w_{k0} + \boldsymbol{x}_i^T\boldsymbol{w}_k) - \sum_{i=1}^{n} \log\Big[1 + \sum_{k=1}^{K-1} e^{w_{k0}+\boldsymbol{x}_i^T\boldsymbol{w}_k}\Big] \tag{3.23}$$

with its Taylor expansion at current estimates

$$\tilde{I}_t(\mathbf{w}) = I(\mathbf{w}^{(t)}) + \nabla I(\mathbf{w}^{(t)})^T(\mathbf{w} - \mathbf{w}^{(t)}) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^{(t)})^T\nabla^2 I(\mathbf{w}^{(t)})(\mathbf{w} - \mathbf{w}^{(t)}), \tag{3.24}$$

where $\nabla I(\mathbf{w}^{(t)})$, $\nabla^2 I(\mathbf{w}^{(t)})$ are corresponding the gradient vector and the Hessian matrix of $I(\mathbf{w})$ at $\mathbf{w}^{(t)}$. After that, the problem can be solved by an iterative algorithm with initial value $\mathbf{w}^{(0)}$ where, at step $(t+1)$, it minimizes the proximal function

$$\tilde{Q}_t(\mathbf{w}) = \tilde{I}_t(\mathbf{w}) - \sum_{k=1}^{K-1} \gamma_k\|\boldsymbol{w}_k\|_1 - \frac{\rho}{2}\sum_{k=1}^{K-1}\|\boldsymbol{w}_k\|_2^2 \tag{3.25}$$

instead of $Q(\mathbf{w};\boldsymbol{\theta}^{[q]})$ and then searches for the updating value $\mathbf{w}^{(t+1)}$ based on the solution of (3.25) that improves the Q-function, i.e., $Q(\mathbf{w}^{(t)};\boldsymbol{\theta}^{[q]}) < Q(\mathbf{w}^{(t+1)};\boldsymbol{\theta}^{[q]})$ until the algorithm converges. This strategy has some advantages especially since $I(.)$ does not have a quadratic form. First, by approximating I with its local quadratic form, several good methods can be used to solve (3.25) such as coordinate ascent, where updating one parameter in each step will avoid computing the inverse of a matrix. Second, one can obtain the closed-form update for each parameter at each iteration of the algorithm, hence, reduce the computational time of the algorithm. Finally, for searching $\mathbf{w}^{(t+1)}$, one can use the efficient backtracking line search

strategy (see Boyd and Vandenberghe (2004)) which is easy to setup.

However, the $K-1$ vectors for the gating network will not approximate $I(\mathbf{w})$ with its Taylor expansion. Here, partial Newton steps are performed by forming a partial quadratic approximation to $Q(\mathbf{w};\boldsymbol{\theta}^{[q]})$ (Taylor expansion at the current estimates), allowing only $(w_{k0},\boldsymbol{w}_k)$ to vary for a single class at a time. This algorithm is similar to the one in Friedman et al. (2010) except the fact that here after each outer loop that cycles over k, a backtracking line search is performed over the step size parameter $t\in[0,1]$. The partial quadratic approximation to $I(\mathbf{w})$ with respect to $(w_{k0},\boldsymbol{w}_k)$ at $\tilde{\mathbf{w}}$ is given by (see Appendix C.2 for more details)

$$l_{I_k}(w_{k0},\boldsymbol{w}_k)=-\frac{1}{2}\sum_{i=1}^{n}d_{ik}(c_{ik}-w_{k0}-\boldsymbol{x}_i^T\boldsymbol{w}_k)^2+C(\tilde{\mathbf{w}}), \tag{3.26}$$

where

$$c_{ik}=\tilde{w}_{k0}+\boldsymbol{x}_i^T\tilde{\boldsymbol{w}}_k+\frac{\tau_{ik}^{[q]}-\pi_k(\tilde{\mathbf{w}};\boldsymbol{x}_i)}{\pi_k(\tilde{\mathbf{w}};\boldsymbol{x}_i)(1-\pi_k(\tilde{\mathbf{w}};\boldsymbol{x}_i))}, \tag{3.27}$$

$$d_{ik}=\pi_k(\tilde{\mathbf{w}};\boldsymbol{x}_i)(1-\pi_k(\tilde{\mathbf{w}};\boldsymbol{x}_i)), \tag{3.28}$$

and $C(\tilde{\mathbf{w}})$ is a function of $\tilde{\mathbf{w}}$. After calculating the partial quadratic approximation $l_{I_k}(w_{k0},\boldsymbol{w}_k)$ about the current parameters $\tilde{\mathbf{w}}$, a coordinate ascent algorithm is used to solve the penalized weighted least-square problem

$$\max_{(w_{k0},\boldsymbol{w}_k)} l_{I_k}(w_{k0},\boldsymbol{w}_k)-\gamma_k\|\boldsymbol{w}_k\|_1-\frac{\rho}{2}\|\boldsymbol{w}_k\|_2^2. \tag{3.29}$$

Using the soft-thresholding operator (see (Hastie et al., 2015, sec. 5.4)), one can obtain the closed-form update for w_{kj} as follows

$$w_{kj}^{m+1}=\frac{\mathcal{S}_{\gamma_k}(\sum_{i=1}^{n}d_{ik}u_{ikj}^m x_{ij})}{\sum_{i=1}^{n}d_{ik}x_{ij}^2+\rho}, \tag{3.30}$$

with $u_{ikj}^m=c_{ik}-w_{k0}^m-\boldsymbol{x}_i^T\boldsymbol{w}_k^m+w_{kj}^m x_{ij}$ and $\mathcal{S}_{\gamma_k}(.)$ is a soft-thresholding operator defined by $[\mathcal{S}_\gamma(u)]_j=\text{sign}(u_j)(|u_j|-\gamma)_+$ and $(x)_+$ a shorthand for $\max\{x,0\}$. Here, m is defined as the mth step of the coordinate ascent algorithm. Note that, for each iteration of the coordinate ascent algorithm one parameter is updated while other are kept fixed, that means for $h\neq j$, $w_{kh}^{m+1}=w_{kh}^m$. For w_{k0}, the closed-form update is given by

$$w_{k0}^{m+1}=\frac{\sum_{i=1}^{n}d_{ik}(c_{ik}-\boldsymbol{x}_i^T\boldsymbol{w}_k^m)}{\sum_{i=1}^{n}d_{ik}}. \tag{3.31}$$

Once the coordinate ascent algorithm converges, the new values of $(w_{k0}, \boldsymbol{w}_k)$ are taken into account for the next loop of the proximal Newton algorithm. Overall, the algorithm is summarized by pseudo-code 2.

Algorithm 2 Proximal Newton method for updating the gating network

1: $\mathbf{w}^{(0)} = \mathbf{w}^{[q]}$.
2: **repeat**
3: **for** $k = 1$ to $K - 1$ **do**
4: Update the quadratic approximation $l_{I_k}(w_{k0}, \boldsymbol{w}_k)$ in (3.26) by using the current parameters.
5: Solve the penalized weighted least-square problem in (3.29) by using coordinate ascent algorithm and compute the solution $\tilde{\mathbf{w}}_k^{(s)}$ according to (3.30), (3.31).
6: Update $(w_{k0}, \boldsymbol{w}_k)$ by the new values.
7: **end for**
8: Set $\mathbf{w}^{(s+1)} = (1 - t)\mathbf{w}^{(s)} + t\tilde{\mathbf{w}}^{(s)}$, where t is found using a backtracking line-search.
9: Evaluate the objective function $Q(.; \boldsymbol{\theta}^{[q]})$ at $\mathbf{w}^{(s+1)}$.
10: **until** the stopping criterion is satisfied.

The initial values for $(w_{k0}, \boldsymbol{w}_k)$ in this EM algorithm are set to $\mathbf{0}$ and the backtracking line-search is needed for algorithm to converge to the optimal solution. The proximal Newton method presented here can overcome the drawback of the coordinate ascent algorithm since at each step has a closed-form update update for each parameter. Hence, it improves the running time of the algorithm.

Even though in some cases the values of the probabilities $\pi_k(\boldsymbol{x}_i; \tilde{\mathbf{w}})$ can become too small (or too close to 1), and the algorithm can get stuck while solving (3.29). To address this issue, we consider proximal Newton-type method as a proper choice for this situation. Proximal Newton-type methods use a symmetric negative definite matrix $\mathbf{B} \approx \nabla^2 I(\tilde{\mathbf{w}}_k)$ to model the curvature of $I(\mathbf{w})$ at $(w_{k0}, \boldsymbol{w}_k)$. In this case, one can follow the suggestions of (Lange, 2013, sec. 8.7) and Gormley et al. (2008) by choosing a constant negative definite matrix $\mathbf{B}$ such as $\nabla^2 I(\tilde{\mathbf{w}}_k) > \mathbf{B}$. The proximal Newton-type algorithm here can be interpreted as a special case of the MM algorithm (Hunter and Lange, 2004). Specifically it is a minorize-maximize algorithm for updating the gating network and also the expert network in multinomial outputs case.

Since,

$$\frac{\partial^2 I(\mathbf{w})}{\partial w_{kj} \partial w_{kh}} = -\sum_{i=1}^{n} x_{ij} x_{ih} \pi_k(\boldsymbol{x}_i; \mathbf{w})(1 - \pi_k(\boldsymbol{x}_i; \mathbf{w})), \ \forall j, h,$$

then, using the fact that $\pi(1 - \pi) \leq 1/4$, we can take $\mathbf{B} = -1/4 \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^T$. Thus, instead of solving (3.29), one can solve the local quadratic model

$$\max_{(w_{k0}, \boldsymbol{w}_k)} \hat{l}_{I_k}(w_{k0}, \boldsymbol{w}_k) - \gamma_k \|\boldsymbol{w}_k\|_1. \tag{3.32}$$

where

$$\hat{l}_{I_k}(w_{k0}, \boldsymbol{w}_k) = -\frac{1}{8}\sum_{i=1}^{n}(\hat{c}_{ik} - w_{k0} - \boldsymbol{x}_i^T \boldsymbol{w}_k)^2 + \hat{C}(\tilde{\mathbf{w}}), \tag{3.33}$$

and

$$\hat{c}_{ik} = \tilde{w}_{k0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{w}}_k + 4(\tau_{ik}^{[q]} - \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)), \tag{3.34}$$

$\hat{C}(\tilde{\mathbf{w}})$ is a function of $\tilde{\mathbf{w}}$. Here, it is clear that this approach has some advantages. Firstly, we can avoid computing the Hessian matrix and can also avoid numerical instability caused by $\pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)$. Secondly, the backtracking line search step is not necessary in this case. The increase of the $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$ after each loop is also guaranteed, since this algorithm is a proximal Newton-type algorithm and is a specific case of the MM algorithm.

Updating the experts network

Now once we have these three methods to update the gating network parameters, we move on updating the experts network parameters ($\{\boldsymbol{\beta}, \sigma^2\}$). To do that, we first perform the update for (β_{k0}, β_k), while fixing σ_k. This corresponds to solving K separated weighted Lasso problems. Hence, we choose to use a coordinate ascent algorithm to deal with this. Actually, in this situation the coordinate ascent algorithm can be seen as a special case of the MM algorithm; hence, this updating step is common to both of the proposed algorithms. More specifically, the update of β_{kj} is performed by maximizing

$$Q(\boldsymbol{\beta}, \sigma; \boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{[q]} \log \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{\beta}_k^T \boldsymbol{x}_i, \sigma_k^2) - \sum_{k=1}^{K}\lambda_k \|\boldsymbol{\beta}_k\|_1; \tag{3.35}$$

using a coordinate ascent algorithm, with initial values $(\beta_{k0}^{[0]}, \boldsymbol{\beta}_k^{[0]}) = (\beta_{k0}^{[q]}, \boldsymbol{\beta}_k^{[q]})$. We obtain closed-form coordinate updates that can be computed for each component following the results in (Hastie et al., 2015, sec. 5.4), and are given by

$$\beta_{kj}^{[s+1]} = \frac{\mathcal{S}_{\lambda_k \sigma_k^{(s)2}}\left(\sum_{i=1}^{n}\tau_{ik}^{[q]} r_{ikj}^{[s]} x_{ij}\right)}{\sum_{i=1}^{n}\tau_{ik}^{[q]} x_{ij}^2}, \tag{3.36}$$

with $r_{ikj}^{[s]} = y_i - \beta_{k0}^{[s]} - \boldsymbol{\beta}_k^{[s]T}\boldsymbol{x}_i + \beta_{kj}^{[s]} x_{ij}$ and $\mathcal{S}_{\lambda_k \sigma_k^{(s)2}}(.)$ is a soft-thresholding operator defined by $[\mathcal{S}_\gamma(u)]_j = \text{sign}(u_j)(|u_j| - \gamma)_+$ and $(x)_+$ a shorthand for $\max\{x, 0\}$. For $h \neq j$, we set $\beta_{kh}^{[s+1]} = \beta_{kh}^{[s]}$. At each iteration m, β_{k0} is updated by

$$\beta_{k0}^{[s+1]} = \frac{\sum_{i=1}^{n}\tau_{ik}^{[q]}(y_i - \boldsymbol{\beta}_k^{[s+1]T}\boldsymbol{x}_i)}{\sum_{i=1}^{n}\tau_{ik}^{[q]}}. \tag{3.37}$$

In the next step, we take $(w_{k0}^{[q+2]}, \boldsymbol{w}_k^{[q+2]}) = (w_{k0}^{[q+1]}, \boldsymbol{w}_k^{[q+1]})$, $(\beta_{k0}^{[q+2]}, \boldsymbol{\beta}_k^{[q+2]}) = (\beta_{k0}^{[q+1]}, \boldsymbol{\beta}_k^{[q+1]})$, rerun the E-step, and update σ_k^2 according to the standard update of a weighted Gaussian

regression

$$\sigma_k^{2[q+2]} = \frac{\sum_{i=1}^n \tau_{ik}^{[q+1]}(y_i - \beta_{k0}^{[q+2]} - \boldsymbol{\beta}_k^{[q+2]^T}\boldsymbol{x}_i)^2}{\sum_{i=1}^n \tau_{ik}^{[q+1]}}. \tag{3.38}$$

Each of these proposed algorithms is iterated until the change in $PL(\boldsymbol{\theta})$ is small enough. Moreover, we can directly get zero coefficients without any thresholding unlike in Khalili (2010); Hunter and Li (2005). These algorithms increase the penalised log-likelihood function (3.4) as shown in Appendix B.

3.3.4 Algorithm tuning and model selection

In practice, appropriate values of the tuning parameters (λ, γ, ρ) should be chosen. To select the tuning parameters, we propose a modified BIC with a grid search scheme, as an extension of the criterion used in Städler et al. (2010b) for regularized mixture of regressions. First, assume that $K_0 \in \{K_1, \ldots, K_M\}$ whereupon K_0 is the true number of expert components. For each value of K, we choose a grid of the tuning parameters. Consider grids of values $\{\lambda_1, \ldots, \lambda_{M_1}\}$, $\{\gamma_1, \ldots, \gamma_{M_2}\}$ in the size of $\sqrt{n}$ and a small enough value of $\rho \approx O(\log n)$ for the ridge turning parameter. $\rho = 0.1 \log n$ can be used in practice. For a given triplet (K, λ_i, γ_j), we select the maximal penalized log-likelihood estimators $\widehat{\boldsymbol{\theta}}_{K,\lambda,\gamma}$ using each of our hybrid EM algorithms presented above. Then, the following modified BIC criterion,

$$\text{BIC}(K, \lambda, \gamma) = L(\widehat{\boldsymbol{\theta}}_{K,\lambda,\gamma}) - DF(\lambda, \gamma)\frac{\log n}{2}, \tag{3.39}$$

where $DF(\lambda, \gamma)$ is the estimated number of non-zero coefficients in the model, is computed. Finally, the model with parameters $(K, \lambda, \gamma) = (\tilde{K}, \tilde{\lambda}, \tilde{\gamma})$ which maximizes the modified BIC value, is selected. While the problem of choosing optimal values of the tuning parameters for penalized MoE models is still an open research, the modified BIC performs reasonably well in our experiments.

3.4 Experimental study

We study the performance of our methods on both simulated data and real data. The results are compared among the regularized MoE using three proposed hybrid EM algorithms with Minorization-Maximization (MoE-Lasso+ℓ_2 (MM)), coordinate ascent (MoE-Lasso+ℓ_2 (CA)) and proximal Newton method (MoE-Lasso+ℓ_2(PN)) with the following four methods: *i)* the standard non-penalized MoE (MoE), *ii)* the MoE with ℓ_2 regularization (MoE+ℓ_2), *iii)* the mixture of linear regressions with Lasso penalty (MIXLASSO), and *iv)* the MoE with BIC penalty for feature selection (MoE-BIC). We consider several evaluation criteria to assess the performance of the models, including sparsity, parameters estimation and clustering criteria.

In addition, to evaluate the affect of the ℓ_2 norm to the sparsity of the model, we also compare our results with the Lasso-regularized model. We use proximal Newton method (MoE-

Lasso (PN)) to estimate parameters in this case.

3.4.1 Evaluation criteria

We compare the results of all the models for three different criteria: sensitivity/specificity, parameters estimation, and clustering performance for simulation data. The sensitivity/specificity is defined by

- *Sensitivity:* proportion of correctly estimated zero coefficients;
- *Specificity:* proportion of correctly estimated nonzero coefficients.

In this way, we compute the ratio of the estimated zero/nonzero coefficients to the true number of zero/nonzero coefficients of the true parameter for each component. In our simulation, the proportion of correctly estimated zero coefficients and nonzero coefficients have been calculated for each data set for the experts parameters and the gating parameters, and we present the average proportion of these criteria computed over 100 different data sets. Also, to deal with the label switching before calculating these criteria, we permuted the estimated coefficients based on an ordered between the expert parameters. If the label switching happens, one can permute the expert parameters and the gating parameters then replace the kth gating network vector with $\mathbf{w}_k^{per} = \mathbf{w}_k - \mathbf{w}_K$. By doing so, we ensure that the log-likelihood will not change, that means $L(\hat{\boldsymbol{\theta}}) = L(\hat{\boldsymbol{\theta}}^{per})$ and these parameters satisfy the initialized condition $\mathbf{w}_K^{per} = \mathbf{0}$. However, the penalized log-likelihood value can be different from the one before permutation. So this may result in misleading values of the sparsity criterion of the model when we permute the parameters. The regularized method tends to choose the model with small absolute values of the gating network. However, for $K = 2$, the log-likelihood function and the penalized log-likelihood function will not change since we have $\mathbf{w}_1^{per} = -\mathbf{w}_1$.
For the second criterion of parameter estimation, we compute the mean and standard deviation of both penalized parameters and non penalized parameters in comparison with the true value $\boldsymbol{\theta}$. We also consider the mean squared error (MSE) between each component of the true parameter vector and the estimated one, which is given by $\|\theta_j - \hat{\theta}_j\|^2$.
For the clustering criterion, once the parameters are estimated and permuted, the provided conditional component probabilities $\hat{\tau}_{ik}$ defined in (3.7) represent a soft partition of the data. A hard partition of the data is given by applying the Bayes's allocation rule

$$\hat{z}_i = \arg\max_{k=1}^{K} \tau_{ik}(\widehat{\boldsymbol{\theta}}), \tag{3.40}$$

where $\hat{z}_i$ represents the estimated cluster label for the ith observation. Given the estimated and true cluster labels, we therefore compute the correct classification rate and the Adjusted Rand Index (ARI).
Also, we note that for the standard MoE with BIC penalty, we consider a pool of $5 \times 4 \times 5 = 100$ submodels. Our EM algorithm with coordinate ascent has been used with zero penalty coefficients and without updating the given zero parameters in the experts and the gating network

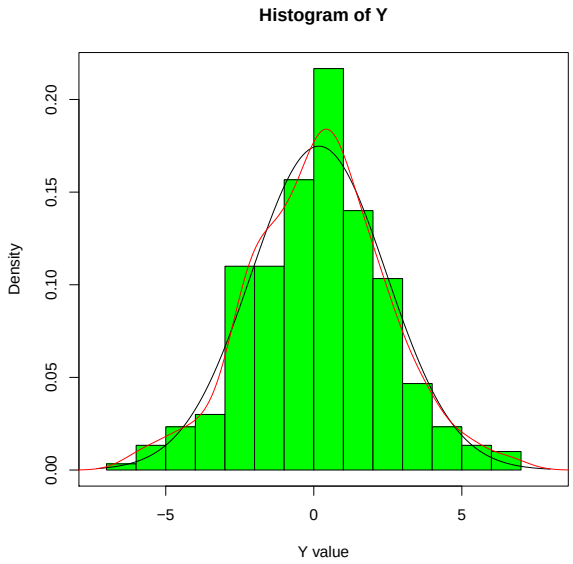


Figure 3.1: Histogram of the response variable Y for a MoE simulation data. The red line represents the estimated density using non-parametric kernel density estimation and the black line shows the estimated density using a GMM.

to obtain the (local) MLE of each submodel. After that, the BIC criterion in (3.39) was used to choose the best submodel among 100 model candidates.

3.4.2 Simulation study

For each data set, we consider $n = 300$ predictors $\boldsymbol{x}$ generated from a multivariate Gaussian distribution with zero mean and correlation defined by $\text{corr}(x_{ij}, x_{ij'}) = 0.5^{|j-j'|}$. The response $Y|\boldsymbol{x}$ is generated from a normal MoE model of $K = 2$ expert components as defined by (2.38) and (2.40), with the following regression coefficients:

$$
\begin{aligned}
(\beta_{10}, \boldsymbol{\beta}_1)^T &= (0, 0, 1.5, 0, 0, 0, 1)^T; \\
(\beta_{20}, \boldsymbol{\beta}_2)^T &= (0, 1, -1.5, 0, 0, 2, 0)^T; \\
(w_{10}, \boldsymbol{w}_1)^T &= (1, 2, 0, 0, -1, 0, 0)^T;
\end{aligned}
$$

and $\sigma_1 = \sigma_2 = \sigma = 1$. Histogram of a typical simulation data set can be seen from Figure 3.1. 100 data sets were generated for this simulation. The results will be presented in the following sections.

Sensitivity/specificity criteria

Table 4.1 presents the sensitivity (S_1) and specificity (S_2) values for the experts 1 and 2 and the gates for each of the considered models. As it can be seen in the obtained results that the ℓ_2 and MoE models cannot be considered as model selection methods since their sensitivity almost

surely equals zero. However, it is obvious that the MoE-Lasso+ℓ_2, with the MM, the coordinate ascent and proximal Newton algorithms, performs quite well for experts 1 and 2. The difference between the methods are not significant. The feature selection becomes more difficult for the gate $\pi_k(\boldsymbol{x}; \boldsymbol{w})$ since there is correlation between features. While mixed Lasso+ℓ_2 regularized model using MM (MoE-Lasso+ℓ_2 (MM)) may get trouble in detecting non-zero coefficients in the gating network, the model using coordinate ascent (MoE-Lasso+ℓ_2 (CA)), proximal Newton (MoE-Lasso+ℓ_2 (PN)), and the Lasso regularized model (MoE-Lasso (PN)) perform quite well. There is a slightly difference between MoE-Lasso+ℓ_2 (PN) and MoE-Lasso (PN) in detecting the zero coefficients in the gating network. In this case, the model without the quadratic penalty performs better in retrieving the zero parameters in the gating network. The MIXLASSO, can detect the zero coefficients in the experts but it will be shown in the later clustering results that this model has a poor result when clustering the data. Note that for the MIXLASSO we do not have gates, so variable "N/A" is mentioned in the results. Finally, while the BIC provides the best results in general, it is hard to apply BIC in reality since the number of submodels may be huge.

Method	Expert 1		Expert 2		Gate	
	S_1	S_2	S_1	S_2	S_1	S_2
MoE-BIC	0.920	1.000	0.930	1.000	0.850	1.000
MoE-Lasso+ℓ_2 (MM)	0.720	1.000	0.777	1.000	0.815	0.615
MoE-Lasso+ℓ_2 (CA)	0.700	1.000	0.803	1.000	0.853	0.945
MoE-Lasso+ℓ_2 (PN)	0.698	1.000	0.797	1.000	0.728	0.995
MoE-Lasso (PN)	0.700	1.000	0.790	1.000	0.748	0.995
MIXLASSO	0.775	1.000	0.693	1.000	N/A	N/A

Table 3.1: Sensitivity (S_1) and specificity (S_2) results for our proposed methods compared with the MoE model using BIC criterion for feature selection and the mixture of linear regression model (MLR).

Parameter estimation

The boxplots of all estimated parameters are shown in Figures 3.2, 3.3 and 3.4. It turns out that the MoE and MoE+ℓ_2 models could not be considered as model selection methods. Besides that, by adding the ℓ_2 penalty functions, we can reduce the variance of the parameters in the gate. The BIC, MoE-Lasso+ℓ_2 (MM), MoE-Lasso+ℓ_2 (CA), MoE-Lasso+ℓ_2 (PN) and MoE-Lasso (PN) provide sparse results for the model, not only in the experts, but also in the gates. However, the method using MM algorithm in this situation forces the nonzero parameter w_{14} toward zero, and this effects the clustering result. The MIXLASSO can also detect zero coefficients in the experts, but since this model does not have a mixture proportions that depend on the inputs, it is least competitive than others.

For the mean and standard derivation results shown in Table 3.2 and Table 3.3, we can see that the model using BIC for selection, the non penalized MoE, and the MoE with ℓ_2

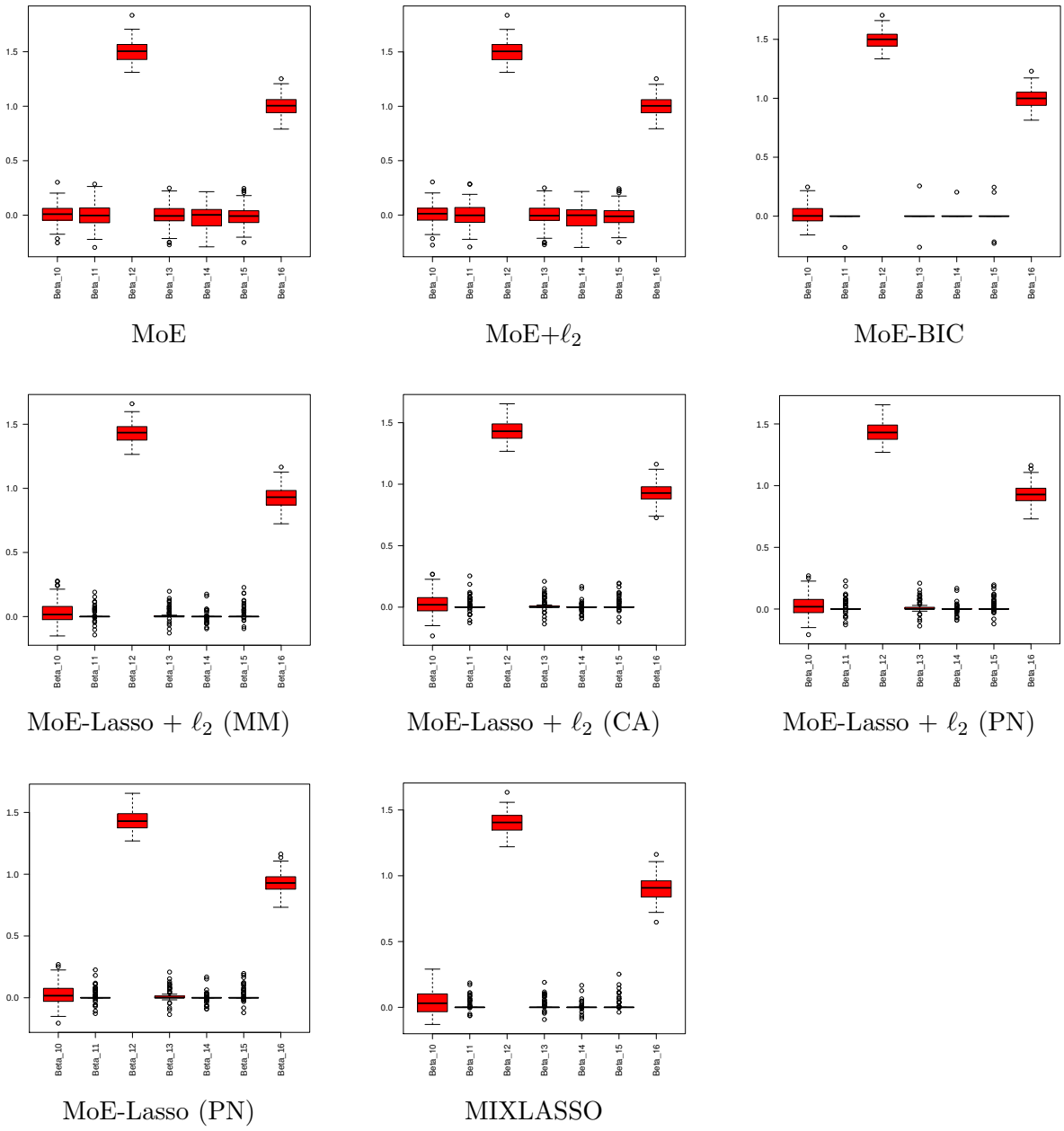


Figure 3.2: Boxplots of the expert 1's parameter estimates obtained by, respectively, Mixture-of-Experts (MoE) with standard MLE, MoE with ℓ_2 regularization, MoE with BIC procedure for model selection, and then our proposed methods: MoE with mixed ℓ_1 (Lasso) and ℓ_2 regularization using the EM algorithm with Minorization-Maximization (MM) updates, Coordinate Ascent (CA) updates, and Proximal Newton (PN) updates; MoE with Lasso regularization with hybrid EM/Proximal Newton method for parameter estimation. Finally, the standard MIXLASSO (Mixture of regressions with Lasso regularization) is also considered.

penalty have better results, while regularized MoE models and MIXLASSO can cause bias to the estimated parameters, since the penalty functions are added to the log-likelihood function.

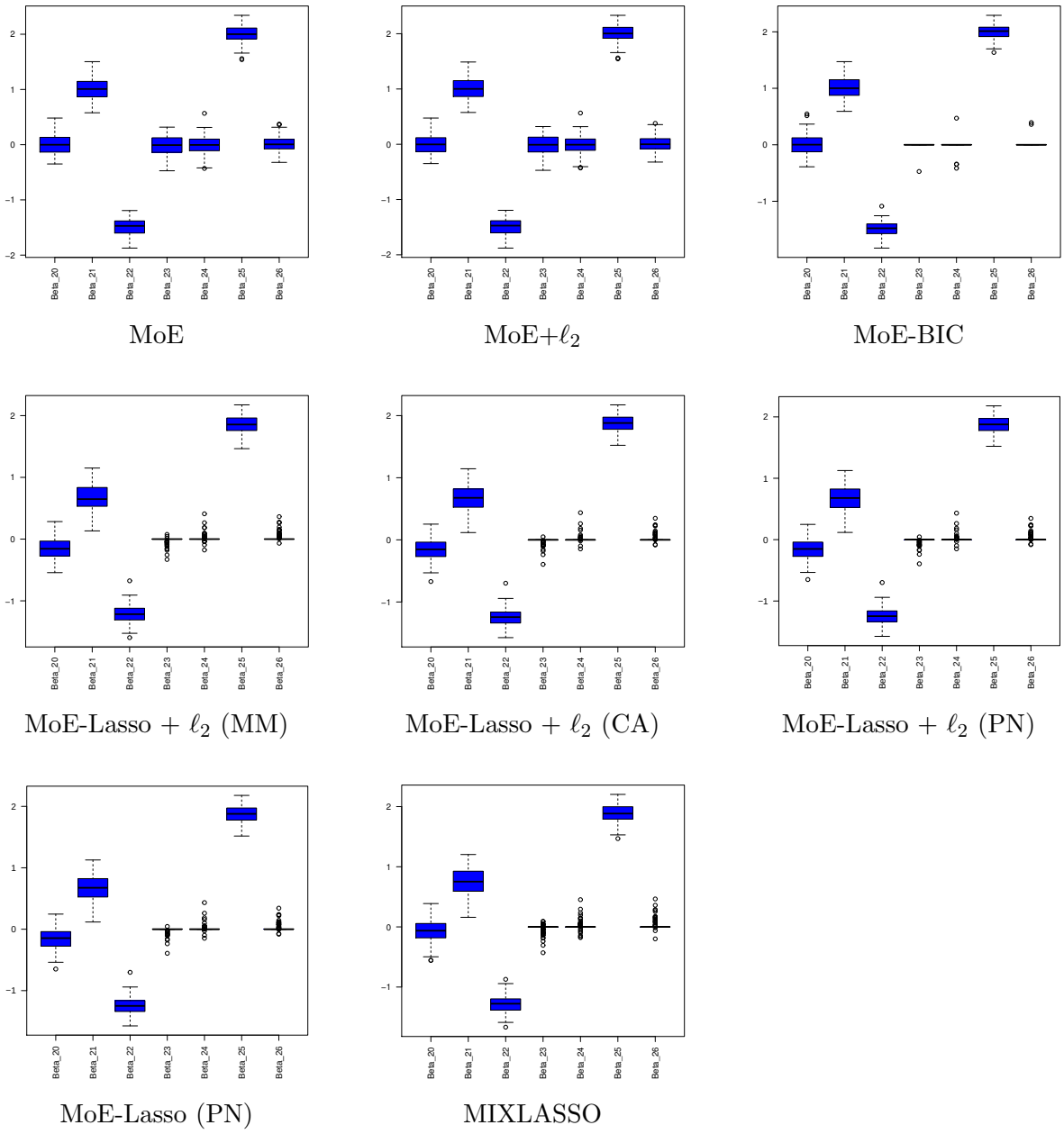
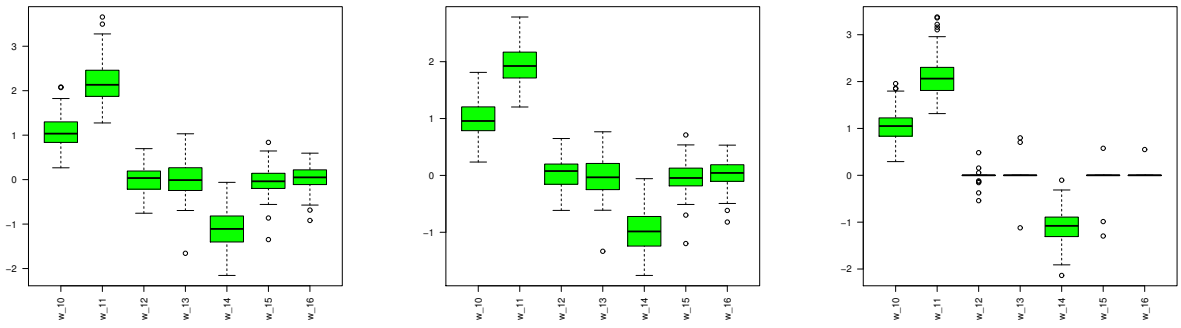


Figure 3.3: Boxplots of the expert 2's parameter estimates obtained by, respectively, Mixture-of-Experts (MoE) with standard MLE, MoE with ℓ_2 regularization, MoE with BIC procedure for model selection, and the proposed methods: MoE with mixed ℓ_1 (Lasso) and ℓ_2 regularization using EM algorithm with Minorization-Maximization (MM) updates, Coordinate Ascent (CA) updates, and Proximal Newton (PN) updates; MoE with Lasso regularization. In addition, the standard MIXLASSO (Mixture of regressions with Lasso regularization) is also considered.

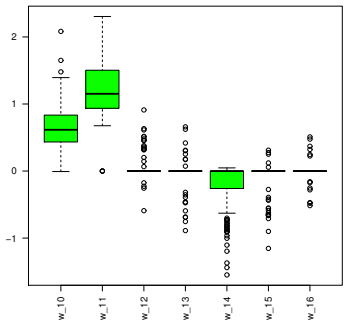
In contrast, from Table 3.4 and Table 3.5, in terms of average mean squared error, the penalized MoE model and MIXLASSO provide a better result than MoE and the MoE with ℓ_2 penalty for estimating the zero coefficients. The BIC still provides the best result, but as we commented



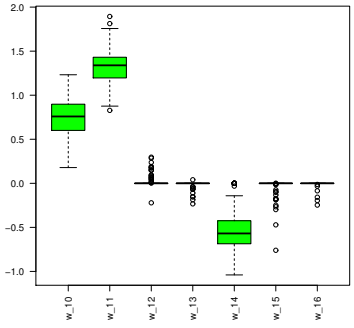
MoE

MoE+ℓ_2

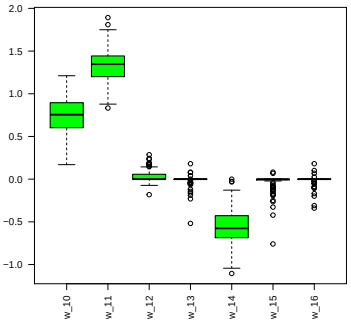
MoE-BIC



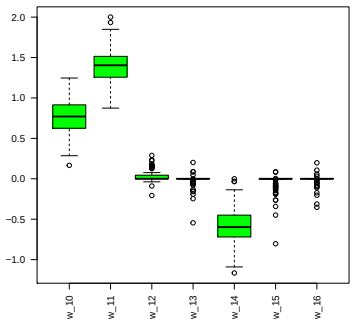
MoE-Lasso + ℓ_2 (MM)



MoE-Lasso + ℓ_2 (CA)



MoE-Lasso + ℓ_2 (PN)



MoE-Lasso (PN)

Figure 3.4: Boxplots of the gate's parameter estimates obtained by Mixture-of-Experts (MoE) with standard MLE, MoE with ℓ_2 regularization, and with BIC procedure for model selection compare with boxplots of the proposed methods: MoE with mixed ℓ_1 (Lasso) and ℓ_2 regularization with Minorization-Maximization (MM) updates, Coordinate Ascent (CA) updates, and Proximal Newton (PN) updates; MoE with Lasso regularization bases on Proximal Newton updates.

before, it is hard to apply BIC in reality especially for high-dimensional data, since this involves a huge collection of model candidates.

Comp.	True value	MoE	MoE+ℓ_2	MoE-BIC	MIXLASSO
Exp.1	0	$0.010_{(.096)}$	$0.009_{(.097)}$	$0.014_{(.083)}$	$0.043_{(.093)}$
	0	$-0.002_{(.106)}$	$-0.002_{(.107)}$	$-0.003_{(.026)}$	$0.011_{(.036)}$
	1.5	$1.501_{(.099)}$	$1.502_{(.099)}$	$1.495_{(.075)}$	$1.404_{(.086)}$
	0	$0.000_{(.099)}$	$0.001_{(.099)}$	$0.000_{(.037)}$	$0.013_{(.036)}$
	0	$-0.022_{(.102)}$	$-0.022_{(.102)}$	$0.002_{(.020)}$	$0.003_{(.027)}$
	0	$-0.001_{(.097)}$	$-0.003_{(.097)}$	$0.000_{(.045)}$	$0.013_{(.040)}$
	1	$1.003_{(.090)}$	$1.004_{(.090)}$	$0.998_{(.077)}$	$0.903_{(.088)}$
Exp.2	0	$0.006_{(.185)}$	$0.005_{(.184)}$	$0.002_{(.178)}$	$-0.063_{(.188)}$
	1	$1.007_{(.188)}$	$1.006_{(.188)}$	$1.002_{(.187)}$	$0.755_{(.220)}$
	-1.5	$-1.492_{(.149)}$	$-1.494_{(.149)}$	$-1.491_{(.129)}$	$-1.285_{(.146)}$
	0	$-0.011_{(.159)}$	$-0.012_{(.158)}$	$-0.005_{(.047)}$	$-0.023_{(.071)}$
	0	$-0.010_{(.172)}$	$-0.008_{(.171)}$	$-0.006_{(.079)}$	$0.016_{(.075)}$
	2	$2.004_{(.169)}$	$2.005_{(.169)}$	$2.003_{(.128)}$	$1.891_{(.159)}$
	0	$0.008_{(.139)}$	$0.007_{(.140)}$	$0.008_{(.053)}$	$0.031_{(.086)}$
Gate	1	$1.095_{(.359)}$	$1.008_{(.306)}$	$1.055_{(.328)}$	N/A
	2	$2.186_{(.480)}$	$1.935_{(.344)}$	$2.107_{(.438)}$	
	0	$0.007_{(.287)}$	$0.038_{(.250)}$	$-0.006_{(.086)}$	
	0	$-0.001_{(.383)}$	$-0.031_{(.222)}$	$0.004_{(.1.55)}$	
	-1	$-1.131_{(.413)}$	$-0.991_{(.336)}$	$-1.078_{(.336)}$	
	0	$-0.022_{(.331)}$	$-0.033_{(.281)}$	$-0.017_{(.172)}$	
	0	$0.025_{(.283)}$	$0.016_{(.246)}$	$0.005_{(.055)}$	
σ	1	$0.965_{(.045)}$	$0.961_{(.045)}$	$0.978_{(.046)}$	$1.000_{(.053)}$

Table 3.2: The true value, mean and standard derivation for each coefficient of the estimated parameter vector via MoE, MoE+ℓ_2, MoE-BIC and the MIXLASSO.

Clustering of heterogeneous regression data

We calculate the accuracy of clustering of all these mentioned models for each data set. The results in terms of ARI and correct classification rate values are provided in Table 4.5. We can see that the MoE-Lasso+ℓ_2 (PN) model provides a good result for clustering data. The obtain results from this method is as good as the non penalized model. However, as we can see, the model without ℓ_2 penalty (MoE-Lasso (PN)) provides better result than MoE-Lasso+ℓ_2 (PN). The MoE-BIC model gives the best result but always with a very significant computational load. The difference between MoE-Lasso+ℓ_2 (CA) and MoE-BIC is smaller than 1%, while the MIXLASSO provides a poor result in terms of clustering. Here, we also see that the MoE-Lasso+ℓ_2 (MM) estimates the parameters in the experts quite well. However, the MM algorithm for updating the gate's parameter causes bad effect, since this approach forces the non-zero coefficient w_{14} toward zero. Hence, this may decrease the clustering performance.
Overall, we can clearly see the MoE-Lasso+ℓ_2 (CA), MoE-Lasso+ℓ_2 (PN) algorithm perform quite well to retrieve the actual sparse support; the sensitivity and specificity results are quite reasonable for the proposed Lasso+ℓ_2 regularization. If the data is not highly correlated, one can remove the ℓ_2 penalty to improve the sparsity of the gating network and also improve the accuracy of clustering of the regularization models. While the penalty function will cause bias to the

Comp.	True value	MoE-Lasso+ℓ_2 (MM)	MoE-Lasso+ℓ_2 (CA)	MoE-Lasso+ℓ_2 (PN)	MoE-Lasso (PN)
Exp.1	0	$0.031_{(.091)}$	$0.026_{(.089)}$	$0.026_{(.089)}$	$0.026_{(.089)}$
	0	$0.009_{(.041)}$	$0.011_{(.046)}$	$0.010_{(.045)}$	$0.010_{(.045)}$
	1.5	$1.435_{(.080)}$	$1.435_{(.080)}$	$1.435_{(.080)}$	$1.434_{(.080)}$
	0	$0.012_{(.042)}$	$0.013_{(.044)}$	$0.013_{(.045)}$	$0.013_{(.044)}$
	0	$0.001_{(.031)}$	$0.000_{(.032)}$	$0.000_{(.031)}$	$0.000_{(.032)}$
	0	$0.013_{(.044)}$	$0.012_{(.043)}$	$0.012_{(.043)}$	$0.012_{(.043)}$
	1	$0.930_{(.082)}$	$0.930_{(.082)}$	$0.931_{(.082)}$	$0.931_{(.082)}$
Exp.2	0	$-0.158_{(.183)}$	$-0.162_{(.177)}$	$-0.163_{(.175)}$	$-0.165_{(.175)}$
	1	$0.661_{(.209)}$	$0.675_{(.202)}$	$0.675_{(.200)}$	$0.675_{(.200)}$
	-1.5	$-1.216_{(.152)}$	$-1.242_{(.139)}$	$-1.243_{(.138)}$	$-1.243_{(.137)}$
	0	$-0.018_{(.055)}$	$-0.018_{(.055)}$	$-0.018_{(.055)}$	$-0.018_{(.055)}$
	0	$0.013_{(.061)}$	$0.011_{(.059)}$	$0.011_{(.060)}$	$0.012_{(.060)}$
	2	$1.856_{(.150)}$	$1.876_{(.149)}$	$1.876_{(.148)}$	$1.876_{(.148)}$
	0	$0.022_{(.062)}$	$0.020_{(.060)}$	$0.020_{(.060)}$	$0.019_{(.059)}$
Gate	1	$0.651_{(.331)}$	$0.759_{(.221)}$	$0.759_{(.218)}$	$0.778_{(.224)}$
	2	$1.194_{(.403)}$	$1.332_{(.208)}$	$1.333_{(.206)}$	$1.400_{(.225)}$
	0	$0.058_{(.193)}$	$0.024_{(.068)}$	$0.033_{(.069)}$	$0.028_{(.067)}$
	0	$-0.025_{(.214)}$	$-0.011_{(.039)}$	$-0.014_{(.068)}$	$-0.014_{(.072)}$
	-1	$-0.223_{(.408)}$	$-0.526_{(.253)}$	$-0.555_{(.211)}$	$-0.584_{(.223)}$
	0	$-0.082_{(.243)}$	$-0.032_{(.104)}$	$-0.040_{(.106)}$	$-0.039_{(.111)}$
	0	$-0.002_{(.132)}$	$-0.007_{(.036)}$	$-0.012_{(.060)}$	$-0.012_{(.062)}$
σ	1	$1.000_{(.052)}$	$0.989_{(.050)}$	$0.989_{(.050)}$	$0.989_{(.050)}$

Table 3.3: The true value, mean and standard derivation for each coefficient of the estimated parameter vector via the regularized models: MoE-Lasso+ℓ_2 (MM), MoE-Lasso+ℓ_2 (CA), MoE-Lasso+ℓ_2 (PN) and MoE-Lasso (PN).

parameters, as shown in the results of the MSE, the algorithm can perform parameter density estimation with an acceptable loss of information due to the bias induced by the regularization. In terms of clustering, the methods using coordinate ascent and proximal Newton method work as well as two other MoE models and MoE-BIC, better than the MoE-Lasso+ℓ_2 (MM), MIXLASSO models. One important thing should be mention is that, with the same tuning parameters $\lambda = 10$, $\gamma = 5$ and $\rho = 0.1 \log n$ the regularized model using proximal Newton-type algorithm MoE-Lasso+ℓ_2 (PN) converges to better values of the penalized log-likelihood than those of the method using coordinate ascent MoE-Lasso+ℓ_2 (CA), $-585.140_{(12.990)}$ compares with $-585.410_{(13.029)}$.

3.4.3 Lasso paths for the regularized MoE parameters

In this section, Lasso paths for the experts network are given based on one typical simulation data set. To do this two fix values of the tuning parameters γ are chosen $\gamma \in \{0, 5\}$, $\rho = 0.1 \log n$ as usual and a grid of values for λ is also provided $\lambda \in \{0, 1, \ldots, 23\}$. For $\lambda \geq 24$, a local solution of the regularized MoE model that closes to the true parameters cannot detected. EM/Proximal

Comp.	True value	Mean squared error			
		MoE	MoE+ℓ_2	MoE-BIC	MIXLASSO
Exp.1	0	$0.0093_{(.015)}$	$0.0094_{(.015)}$	$0.0070_{(.011)}$	$0.0106_{(.016)}$
	0	$0.0112_{(.016)}$	$0.0114_{(.017)}$	$0.0007_{(.007)}$	$0.0014_{(.005)}$
	1.5	$0.0098_{(.014)}$	$0.0098_{(.015)}$	$0.0057_{(.007)}$	$0.0166_{(.019)}$
	0	$0.0099_{(.016)}$	$0.0099_{(.016)}$	$0.0013_{(.009)}$	$0.0015_{(.005)}$
	0	$0.0108_{(.015)}$	$0.0109_{(.016)}$	$0.0004_{(.004)}$	$0.0007_{(.003)}$
	0	$0.0094_{(.014)}$	$0.0094_{(.014)}$	$0.0020_{(.010)}$	$0.0017_{(.008)}$
	1	$0.0081_{(.012)}$	$0.0082_{(.012)}$	$0.0059_{(.009)}$	$0.0172_{(.021)}$
Exp.2	0	$0.0342_{(.042)}$	$0.0338_{(.042)}$	$0.0315_{(.049)}$	$0.0392_{(.059)}$
	1	$0.0355_{(.044)}$	$0.0354_{(.044)}$	$0.0350_{(.044)}$	$0.1084_{(.130)}$
	-1.5	$0.0222_{(.028)}$	$0.0221_{(.028)}$	$0.0166_{(.240)}$	$0.0672_{(.070)}$
	0	$0.0253_{(.032)}$	$0.0252_{(.031)}$	$0.0022_{(.022)}$	$0.0056_{(.022)}$
	0	$0.0296_{(.049)}$	$0.0294_{(.049)}$	$0.0063_{(.032)}$	$0.0059_{(.023)}$
	2	$0.0286_{(.040)}$	$0.0287_{(.040)}$	$0.0163_{(.023)}$	$0.0371_{(.051)}$
	0	$0.0195_{(.029)}$	$0.0195_{(.029)}$	$0.0028_{(.020)}$	$0.0083_{(.028)}$
Gate	1	$0.1379_{(.213)}$	$0.0936_{(.126)}$	$0.1104_{(.178)}$	N/A
	2	$0.2650_{(.471)}$	$0.1225_{(.157)}$	$0.2035_{(.371)}$	
	0	$0.0825_{(.116)}$	$0.0641_{(.086)}$	$0.0075_{(.040)}$	
	0	$0.1466_{(.302)}$	$0.1052_{(.196)}$	$0.0239_{(.147)}$	
	-1	$0.1875_{(.263)}$	$0.1129_{(.148)}$	$0.1189_{(.191)}$	
	0	$0.1101_{(.217)}$	$0.0803_{(.164)}$	$0.0299_{(.195)}$	
	0	$0.0806_{(.121)}$	$0.0610_{(.095)}$	$0.0030_{(.030)}$	
σ	1	$0.0033_{(.004)}$	$0.0035_{(.004)}$	$0.0026_{(.003)}$	$0.0028_{(.003)}$

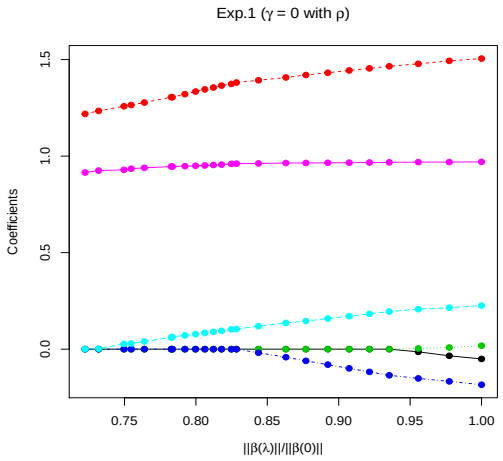
Table 3.4: The true value and mean squared error for each coefficient of the estimated parameter vector via MoE, MoE+ℓ_2, MoE-BIC and the MIXLASSO.

Newton method (MoE-Lasso+ℓ_2 (PN)) is used to obtain the coefficients. The Figure 3.5 gives the Lasso paths for the experts network and also the changing of the gating network coefficients with different values of λ.

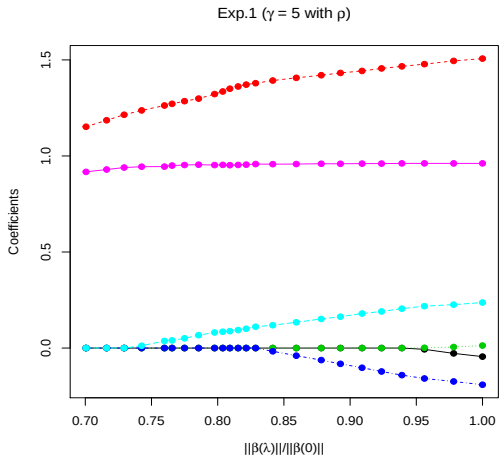
From Figure 3.5, one can see that the Lasso paths for each expert component perform quite consistent between different values of γ. Similar with the classical Lasso path for linear regression these entire Lasso paths are also piecewise-linear, excepts the trajectory of the intercept. However, the performance of the gating network coefficients with difference values of λ is hard to predict. Especially, the changing of λ also effect to the sparsity of the gating network in this case. This behavior can be found in the case $\gamma = 5$. At $\gamma = 5$ and $\lambda \geq 10$ the third coefficient of the gating network is difference from it true value 0. Hence, it is interesting to provide theoretical results to explain this behavior of the regularized MoE.

3.4.4 Evaluation of the model selection via BIC

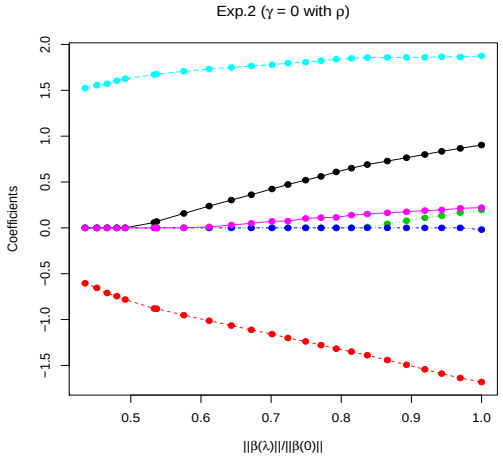
A small evaluation of the BIC criterion is given in this section. Here, we consider the effect of the BIC criterion to the sparsity of the model, the accuracy of clustering and the adjust rank index criteria. Also the effect of the ℓ_2 norm is also mention. For this reason, two grid of



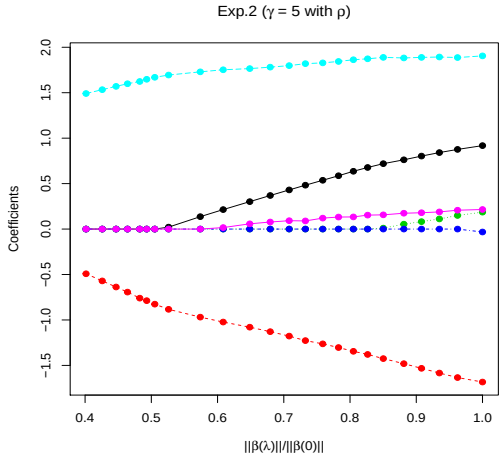
Lasso path for Exp.1 with $\gamma = 0$



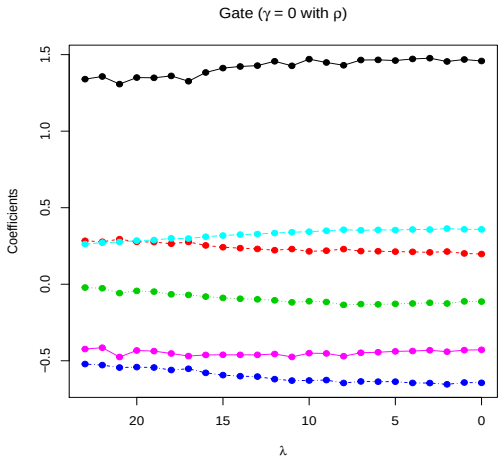
Lasso path for Exp.1 with $\gamma = 5$



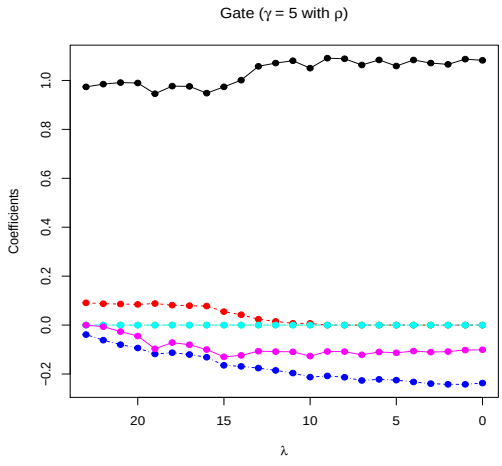
Lasso path for Exp.2 with $\gamma = 0$



Lasso path for Exp.2 with $\gamma = 5$



Changing of the gating network with $\gamma = 0$



Changing of the gating network with $\gamma = 5$

Figure 3.5: Lasso paths for the experts network and the changing of the gating network.

Comp.	True value	Mean squared error			
		MoE-Lasso+ℓ_2 (MM)	MoE-Lasso+ℓ_2 (CA)	MoE-Lasso+ℓ_2 (PN)	MoE-Lasso (PN)
Exp.1	0	$0.0092_{(.015)}$	$0.0087_{(.014)}$	$0.0085_{(.014)}$	$0.0085_{(.014)}$
	0	$0.0018_{(.005)}$	$0.0022_{(.008)}$	$0.0021_{(.007)}$	$0.0021_{(.007)}$
	1.5	$0.0106_{(.012)}$	$0.0107_{(.012)}$	$0.0106_{(.012)}$	$0.0107_{(.012)}$
	0	$0.0019_{(.005)}$	$0.0021_{(.006)}$	$0.0022_{(.006)}$	$0.0022_{(.006)}$
	0	$0.0010_{(.004)}$	$0.0010_{(.004)}$	$0.0010_{(.004)}$	$0.0010_{(.004)}$
	0	$0.0021_{(.007)}$	$0.0020_{(.006)}$	$0.0020_{(.006)}$	$0.0020_{(.006)}$
	1	$0.0117_{(.015)}$	$0.0116_{(.015)}$	$0.0115_{(.015)}$	$0.0114_{(.015)}$
Exp.2	0	$0.0585_{(.072)}$	$0.0575_{(.079)}$	$0.0575_{(.077)}$	$0.0580_{(.077)}$
	1	$0.1583_{(.157)}$	$0.1465_{(.148)}$	$0.1456_{(.146)}$	$0.1455_{(.146)}$
	-1.5	$0.1034_{(.098)}$	$0.0860_{(.087)}$	$0.0851_{(.086)}$	$0.0851_{(.086)}$
	0	$0.0033_{(.013)}$	$0.0034_{(.017)}$	$0.0034_{(.017)}$	$0.0034_{(.017)}$
	0	$0.0039_{(.019)}$	$0.0037_{(.020)}$	$0.0037_{(.020)}$	$0.0037_{(.020)}$
	2	$0.0432_{(.056)}$	$0.0375_{(.050)}$	$0.0374_{(.050)}$	$0.0374_{(.050)}$
	0	$0.0043_{(.017)}$	$0.0040_{(.015)}$	$0.0039_{(.015)}$	$0.0039_{(.015)}$
Gate	1	$0.2315_{(.240)}$	$0.1067_{(.125)}$	$0.1055_{(.124)}$	$0.0994_{(.122)}$
	2	$0.8123_{(.792)}$	$0.4890_{(.277)}$	$0.4881_{(.275)}$	$0.4111_{(.269)}$
	0	$0.0404_{(.032)}$	$0.0052_{(.015)}$	$0.0059_{(.014)}$	$0.0053_{(.013)}$
	0	$0.0501_{(.050)}$	$0.0017_{(.007)}$	$0.0049_{(.028)}$	$0.0054_{(.031)}$
	-1	$0.7703_{(.760)}$	$0.2885_{(.295)}$	$0.2428_{(.212)}$	$0.2226_{(.213)}$
	0	$0.0656_{(.066)}$	$0.0120_{(.062)}$	$0.0128_{(.061)}$	$0.0137_{(.068)}$
	0	$0.0175_{(.018)}$	$0.0013_{(.008)}$	$0.0038_{(.016)}$	$0.0039_{(.017)}$
σ	1	$0.0027_{(.003)}$	$0.0027_{(.003)}$	$0.0027_{(.003)}$	$0.0027_{(.003)}$

Table 3.5: The true value and mean squared error for each coefficient of the estimated parameter vector of the regularized models: MoE-Lasso+ℓ_2 (MM), MoE-Lasso+ℓ_2 (CA), MoE-Lasso+ℓ_2 (PN) and MoE-Lasso (PN).

Model	C.rate	ARI
MoE	$89.57\%_{(1.65\%)}$	$0.6226_{(.053)}$
MoE+ℓ_2	$89.62\%_{(1.63\%)}$	$0.6241_{(.052)}$
MoE-BIC	$90.05\%_{(1.65\%)}$	$0.6380_{(.053)}$
MoE-Lasso+ℓ_2 (MM)	$87.76\%_{(2.19\%)}$	$0.5667_{(.067)}$
MoE-Lasso+ℓ_2 (CA)	$89.46\%_{(1.76\%)}$	$0.6190_{(.056)}$
MoE-Lasso+ℓ_2 (PN)	$89.53\%_{(1.65\%)}$	$0.6210_{(.052)}$
MoE-Lasso (PN)	$89.56\%_{(1.66\%)}$	$0.6222_{(.053)}$
MIXLASSO	$82.89\%_{(1.92\%)}$	$0.4218_{(.050)}$

Table 3.6: Average of the accuracy of clustering (correct classification rate and Adjusted Rand Index) comparison among the non-penalized MoE model, MoE with ℓ_2 regularization, MoE with the BIC criterion for model selection and the proposed regularized models. Finally, the results for the standard MIXLASSO (Mixture of regressions with Lasso regularization) is also given.

three difference values of λ and γ are used $\lambda \in \{9, 10, 11\}$, $\gamma \in \{4, 5, 6\}$. The tuning parameter $\rho = 0.1 \log n$ as in our simulation. These criteria are given on 100 simulation data sets.

From Table 3.7 it shows that the model with highest BIC value with respect to $(\lambda, \gamma, \rho) =$

Model (λ, γ, ρ)	BIC	Sparsity criteria						Clustering criteria	
		Exp.1		Exp.2		Gate		C.rate	ARI
		S_1	S_2	S_1	S_2	S_1	S_2		
$(9, 4, \rho)$	$-512.75_{(13.67)}$	0.6475	1.000	0.7500	1.000	0.6425	0.995	$89.61\%_{(1.65\%)}$	$0.6236_{(.052)}$
$(9, 5, \rho)$	$-512.78_{(13.88)}$	0.6450	1.000	0.7533	1.000	0.7250	0.995	$89.50\%_{(1.68\%)}$	$0.6202_{(.053)}$
$(9, 6, \rho)$	$-513.28_{(13.70)}$	0.6475	1.000	0.7533	1.000	0.7825	0.990	$89.38\%_{(1.63\%)}$	$0.6160_{(.052)}$
$(10, 4, \rho)$	$-512.94_{(13.67)}$	0.6925	1.000	0.7900	1.000	0.6425	0.995	$89.59\%_{(1.68\%)}$	$0.6231_{(.053)}$
$(10, 5, \rho)$	$-512.81_{(13.86)}$	0.6975	1.000	0.7967	1.000	0.7275	0.995	$89.53\%_{(1.65\%)}$	$0.6210_{(.052)}$
$(10, 6, \rho)$	$-513.17_{(13.71)}$	0.6950	1.000	0.8000	1.000	0.7800	0.990	$89.42\%_{(1.65\%)}$	$0.6174_{(.053)}$
$(11, 4, \rho)$	$-513.25_{(13.88)}$	0.7425	1.000	0.8333	1.000	0.6300	0.990	$89.56\%_{(1.69\%)}$	$0.6221_{(.053)}$
$(11, 5, \rho)$	$-513.19_{(13.95)}$	0.7400	1.000	0.8300	1.000	0.7250	0.995	$89.49\%_{(1.71\%)}$	$0.6198_{(.054)}$
$(11, 6, \rho)$	$-513.54_{(13.70)}$	0.7350	1.000	0.8300	1.000	0.7850	0.990	$89.44\%_{(1.68\%)}$	$0.6183_{(.054)}$
$(10, 5, 0)$	$-512.11_{(13.75)}$	0.7000	1.000	0.7900	1.000	0.7475	0.995	$89.56\%_{(1.66\%)}$	$0.6222_{(.053)}$
$(10, 6, 0)$	$-512.53_{(13.73)}$	0.7025	1.000	0.7967	1.000	0.7875	0.990	$89.49\%_{(1.70\%)}$	$0.6199_{(.054)}$

Table 3.7: Sparsity and average of the accuracy of clustering criteria on different tuning parameters.

$(9, 4, 0.1 \log n)$ provides the best result in term of clustering even better that the non penalized model. However, the Sensitive/Specificity criteria of this model is not as good as the others. The model which is used in our simulation $(\lambda, \gamma, \rho) = (10, 5, 0.1 \log n)$ provides balance results between sparsity criteria and accuracy of clustering criteria. In general, BIC is a good criterion for selecting model but base on the BIC criterion cannot ensure that one would obtain a best model.

Another thing can be observed form Table 3.7 is that adding an ℓ_2 norm in this case can reduce the sparsity and accuracy of clustering criteria. It seems that adding the ℓ_2 penalty will work quite well for the non-penalized model due to the fact that maximizing the log-likelihood function leads to using large positive and negative estimates for the regression coefficients, especially the gating network parameters (see Park and Hastie (2007b); Bunea (2008) for the discussion on logistic regression). However, when the correlations of some features are not strong, and γ not deceases to zero the effect of the quadratic penalty with a small ρ is negligible.

3.4.5 Applications to real data sets

We analyze two real data sets as a further test of the methodology. Here, we investigate the housing data described on the website UC Irvine Machine Learning Repository and baseball salaries from the Journal of Statistics Education (www.amstat.org/publications/jse). This was done to provide a comparison with the work of Khalili (2010), Khalili and Chen (2007). While in Khalili and Chen (2007) the authors used Lasso-penalized mixture of linear regression (MLR) models, we still apply penalized mixture of experts (to better represent the data than when using MRL models). We compare the results of each model-based upon two different criteria: the average mean squared error (MSE) between observation values of the response variable and the predicted values of this variable; we also consider the correlation of these values. After the parameters are estimated, we use two following values as predicted values for Y:

- **Method 1**: This method uses the following expected value under the estimated model

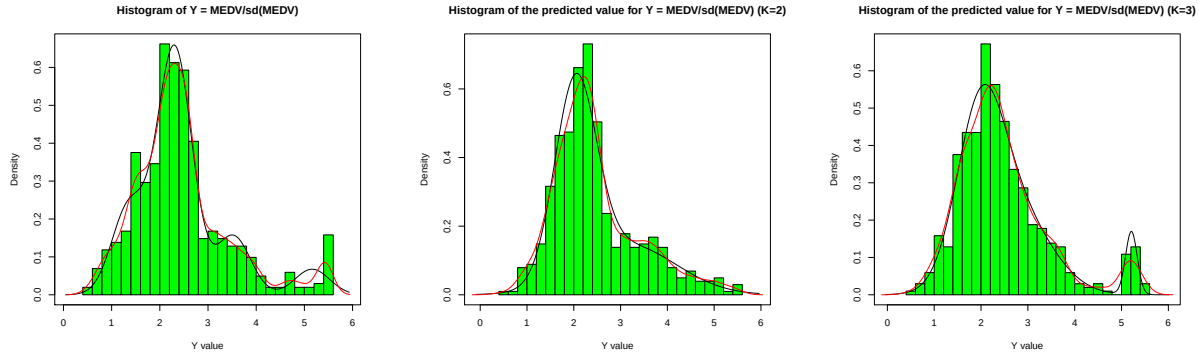
$$\begin{aligned}\hat{Y} = \mathbb{E}_{\hat{\boldsymbol{\theta}}}(Y|\boldsymbol{x}) &= \sum_{k=1}^{K} \pi_k(\boldsymbol{x}; \hat{\mathbf{w}}) \mathbb{E}_{\hat{\boldsymbol{\theta}}}(Y|Z=k, \boldsymbol{x}) \\ &= \sum_{k=1}^{K} \pi_k(\boldsymbol{x}; \hat{\mathbf{w}})(\hat{\beta}_{k0} + \boldsymbol{x}^T \hat{\boldsymbol{\beta}}_k).\end{aligned}$$

- **Method 2**: We use the Bayes's rule in (3.40) to cluster the data and the following value to predict Y

$$\hat{Y} = \mathbb{E}_{\hat{\boldsymbol{\theta}}}(Y|\boldsymbol{x}, \hat{z}=k) = \hat{\beta}_{k0} + \boldsymbol{x}^T \hat{\boldsymbol{\beta}}_k.$$

Note that, for the real data sets we do not consider the MoE model with BIC selection since it is computationally expensive.

Housing data



Histogram of Y | Histogram of $\hat{Y}(K=2)$ | Histogram of $\hat{Y}(K=3)$

Figure 3.6: Histograms of Y = MEDV/sd(MEDV) and its predicted value for housing data. The red lines represent the estimated densities using kernel density estimation and the black lines show the estimated densities using a GMM.

This data set concerns houses' value in the suburbs of Boston. It contains 506 observations and 13 features that may affect the house value. These features are: Per capita crime rate by town (x_1); proportion of residential land zoned for lots over $25,000$ sq.ft. (x_2); proportion of non-retail business acres per town (x_3); Charles River dummy variable ($= 1$ if tract bounds river; 0 otherwise) (x_4); nitric oxides concentration (parts per 10 million) (x_5); average number of rooms per dwelling (x_6); proportion of owner-occupied units built prior to 1940 (x_7); weighted distances to five Boston employment centers (x_8); index of accessibility to radial highways (x_9); full-value property-tax rate per $\$10,000$ (x_{10}); pupil-teacher ratio by town (x_{11}); $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town (x_{12}); % lower status of the population (x_{13}). The columns of X were standardized to have mean 0 and variance 1. The response homes in variable of interest is the median value of owner occupied homes in $\$1000's$, MEDV. Based on the

Features	MLE $\hat{\sigma} = 0.320$			MoE-Lasso+ℓ_2 (Khalili, 2010) $\hat{\sigma} = 0.352$		
	Exp.1	Exp.2	Gate	Exp.1	Exp.2	Gate
x_0	2.23	3.39	19.17	2.16	2.84	1.04
x_1	-0.12	3.80	-4.85	-0.09	-	-
x_2	0.07	0.04	-5.09	-	0.07	-
x_3	0.05	-0.03	7.74	-	-	0.67
x_4	0.03	-0.01	-1.46	-	0.05	-
x_5	-0.18	-0.16	9.39	-	-	-
x_6	-0.01	0.63	1.36	-	0.60	-0.27
x_7	-0.06	-0.07	-8.34	-	-	-
x_8	-0.20	-0.21	8.81	-	-0.20	-
x_9	0.02	0.31	0.96	-	0.55	-
x_{10}	-0.19	-0.33	-0.45	-	-	-
x_{11}	-0.14	-0.18	7.06	-	-	0.54
x_{12}	0.06	0.01	-6.17	0.05	-	-
x_{13}	-0.32	-0.73	36.27	-0.29	-0.49	1.56

Table 3.8: Fitted models for housing data with MLE and MoE-Lasso+ℓ_2 (Khalili, 2010).

histogram of $Y = \text{MEDV}/\text{sd}(\text{MEDV})$ in Figure 3.6, where sd(MEDV) is the standard deviation of MEDV, Khalili (2010) separated Y into two groups of houses with "low" and "high" values. Hence, a MoE model is used to fit the response

$$Y \sim \pi_1(\boldsymbol{x}; \mathbf{w})\mathcal{N}(y; \beta_{10} + \boldsymbol{x}^T\boldsymbol{\beta}_1, \sigma^2) + (1 - \pi_1(\boldsymbol{x}; \mathbf{w}))\mathcal{N}(y; \beta_{20} + \boldsymbol{x}^T\boldsymbol{\beta}_2, \sigma^2),$$

where $\pi_1(\boldsymbol{x}; \boldsymbol{w}) = \dfrac{e^{w_{10}+\boldsymbol{x}^T\boldsymbol{w}_1}}{1 + e^{w_{10}+\boldsymbol{x}^T\boldsymbol{w}_1}}$.

The parameter estimates of the MoE models obtained by MoE-Lasso+ℓ_2 (CA), MoE-Lasso+ℓ_2 (PN), MoE-Lasso (PN) and standard MLE are given in Table 3.8, Table 3.9. We compare our results with those of Khalili (2010) and the non-penalized MoE. In Table 3.10, we provide the result in terms of average MSE and the correlation between the true observation value Y and its predictions $\hat{Y}$. Our result provides a least sparse model than the one in Khalili (2010). Some parameters in these methods have the same value. However, the MSE and the correlation from our methods are better than those Khalili (2010) Hence, in application one would consider the sparsity and the prediction of each estimated parameters. The Lasso +ℓ_2 (PN) and Lasso (PN) are successful in detecting the non-zero coefficient with respect to x_6 in the gating network similar with the result from the method in Khalili (2010) and there is a slightly difference in the gating network between MoE-Lasso +ℓ_2 (PN) and MoE-Lasso (PN) due to the addition of the ℓ_2 norm. With the same tuning parameters $\lambda = 42$, $\gamma = 10$ and $\rho = 0.1 \log n$, the MoE-Lasso +ℓ_2 (PN) provides a better value of the penalized log-likelihood that the MoE-Lasso +ℓ_2 (CA), -372.377 compares with -374.489. All the regularization methods give comparative results with the MLE.

Features	MoE-Lasso+ℓ_2 (CA) $\hat{\sigma} = 0.346$			MoE-Lasso+ℓ_2 (PN) $\hat{\sigma} = 0.352$			MoE-Lasso (PN) $\hat{\sigma} = 0.353$		
	Exp.1	Exp.2	Gate	Exp.1	Exp.2	Gate	Exp.1	Exp.2	Gate
x_0	2.20	2.82	0.79	2.19	2.83	0.99	2.19	2.83	1.00
x_1	-0.09	-	-	-0.09	-	-	-0.09	-	-
x_2	-	0.07	-	-	0.06	-	-	0.06	-
x_3	-	-	0.41	-	-	0.58	-	-	0.59
x_4	0.05	0.06	-	0.04	0.06	-	0.04	0.06	-
x_5	-0.08	-	-	-0.07	-	-	-0.07	-	-
x_6	-	0.56	-	-	0.59	-0.25	-	0.59	-0.21
x_7	-0.05	-	-	-0.04	-	-	-0.04	-	-
x_8	-0.03	-0.19	-	-	-0.20	-	-	-0.19	-
x_9	-	0.60	-	-	0.54	-	-	0.55	-
x_{10}	-0.01	-	-	- 0.003	-	-	- 0.003	-	-
x_{11}	-0.10	-0.08	0.28	-0.09	-0.06	0.39	-0.09	-0.06	0.39
x_{12}	0.05	-	-	0.05	-	-	0.05	-	-
x_{13}	-0.29	-0.57	1.05	-0.29	-0.51	1.26	-0.29	-0.51	1.36

Table 3.9: Fitted models for housing data with our proposed algorithms: MoE-Lasso+ℓ_2 (CA), MoE-Lasso+ℓ_2 (PN) and MoE-Lasso (PN).

Model	Predicted criteria			
	Method 1		Method 2	
	R^2	MSE	R^2	MSE
MoE	0.8457	$0.1544_{(.577)}$	0.8981	$0.1019_{(.236)}$
MoE-Lasso+ℓ_2 (Khalili, 2010)	0.8094	$0.2044_{(.709)}$	0.8698	$0.1371_{(.286)}$
MoE-Lasso+ℓ_2 (CA)	0.8221	$0.1989_{(.619)}$	0.8905	$0.1104_{(.254)}$
MoE-Lasso+ℓ_2 (PN)	0.8180	$0.1903_{(.717)}$	0.8839	$0.1172_{(.281)}$
MoE-Lasso (PN)	0.8204	$0.1878_{(.702)}$	0.8832	$0.1178_{(.282)}$

Table 3.10: Results for Housing data set with a two-component MoE model.

Considering the case $K = 3$ and $\rho = 0$ as an extension. The estimated parameters, the average MSE, and the correlation between the true observation value Y and its prediction $\hat{Y}$ (based on Method 2) for this case can be found in Table 3.11 and Table 3.12. It turns out that this model provides better results than those with $K = 2$ (with $\rho = 0$) in term of prediction. The BIC criterion with $K = 3$ is also better than the case with $K = 2$, -246.844 compares with -292.822.

The histograms of $\hat{Y}$ (using Method 2) for $K = 2$ and $K = 3$ are also compared with the histogram of Y (see Figure 3.6). It turns out that with $K = 3$ the histogram of $\hat{Y}$ provides better result in approximating the histogram of the true value.

Features	Expert, $\sigma = 0.261$			Gating network	
	Exp.1	Exp.2	Exp.3	Gate.1	Gate.2
x_0	2.14331	5.01278	2.50307	-0.27941	-2.96191
x_1	-0.09202	-	-	0.01695	-
x_2	-	0.03392	0.01033	-	-
x_3	-	-	-0.03802	-	-
x_4	0.05261	0.01517	0.00950	-	0.12079
x_5	-0.12082	-	-	-	-
x_6	-0.08837	0.12770	0.67982	-	0.97405
x_7	-	-	-0.17057	0.27293	-
x_8	-0.08727	-	-0.12630	-	-0.27807
x_9	0.04286	-	0.11111	-	-
x_{10}	-0.06967	0.21112	-0.13565	0.42344	-
x_{11}	-0.08817	-	-0.11758	0.01711	-0.02419
x_{12}	0.03348	-	-	-0.22068	-
x_{13}	-0.34326	-	-	1.01512	-

Table 3.11: Fitted models for housing data with MoE-Lasso (PN) method ($K = 3$).

Method	Criteria (Method 2)		Number of observations		
	R^2	MSE	Class 1	Class 2	Class 3
MoE-Lasso (PN)	0.9372	$0.0629_{(.106)}$	195	28	283

Table 3.12: Results for Housing data set obtained by our MoE-Lasso (PN) ($K = 3$).

Histogram of Y Histogram of $\hat{Y}$

Figure 3.7: Histograms of the log of salary and its predicted value for baseball data. The red lines represent the estimated densities using non-parametric kernel density estimation, and the black lines represents the estimated densities using a GMM.

Baseball salaries data

We now consider baseball salaries data set from the Journal of Statistics Education (see also Khalili and Chen (2007)) as a further test of the methodology. This data set contains 337 observations and 33 features. We compare our results with the non-penalized MoE models and the MIXLASSO models (see Khalili and Chen (2007)). Khalili and Chen (2007) used this data set in the analysis, which included an addition of 16 interaction features, making in total 32 predictors. The columns of $\boldsymbol{X}$ were standardized to have mean 0 and variance 1. Histogram of the log of salary (see Figure 3.7) shows multimodality making it a good candidate for the response variable under the MoE model with two components:

$$Y = \log(salary) \sim \pi_1(\boldsymbol{x}; \mathbf{w})\mathcal{N}(y; \beta_{10} + \boldsymbol{x}^T\boldsymbol{\beta}_1, \sigma^2) + (1 - \pi_1(\boldsymbol{x}; \mathbf{w}))\mathcal{N}(y; \beta_{20} + \boldsymbol{x}^T\boldsymbol{\beta}_2, \sigma^2).$$

By taking all the tuning parameters to zero, we obtain the maximum likelihood estimator of the model. We also compare our result with MIXLASSO from Khalili and Chen (2007). Table 3.13 and Table 3.14 present the estimated parameters for baseball salary data and Table 3.15 shows the results in terms of MSE, and R^2 between the true value of Y and its predicted values. These results suggest that the proposed algorithms with the Lasso+ℓ_2 penalty also shrink some parameters to zero and have acceptable results compared to MoE. It also shows that the proposed models provide better results than that of the MIXLASSO model. The two algorithms MoE-Lasso+ℓ_2 (CA) and MoE-Lasso+ℓ_2 (PN) converge to similar sets of parameters and the same penalized log-likelihood values, -292.9301 and -292.9294. Hence, they all provide similar results based on MSE and R^2 criteria. However, the method using proximal Newton-type procedure takes few seconds to obtain the estimate parameters, while for method using coordinate ascent few minutes are required to estimate these coefficients. We compare the CPU times of these methods in the next section.

3.4.6 CPU times and discussion for the high-dimensional setting

CPU times

For the CPU times, we compare two methods: the coordinate ascent algorithm (MoE-Lasso+ℓ_2 (CA)) and the proximal Newton algorithm (MoE-Lasso+ℓ_2 (PN)). We test these algorithms on different data sets. The first data set is one of 100 data sets used for the simulation study. With this data set, we run these algorithms 10 times with different number of clusters $K = 2$ and $K = 3$. The second data set is the baseball salaries. Finally, we also consider the residential building data set (UCI Machine Learning Repository) as a further comparison. The results for this data set is given in the next section. The computer used for this work has CPU Intel i5-6500T 2.5GHz with 16GB RAM.

The obtained results are given in Table 3.16. We can see that the algorithm for the residential data which has a quite high number of features, requires only few minutes and is thus has a very reasonable speed, and for moderate dimensional problems, is very fast.

Features	MLE, $\widehat{\sigma} = 0.277$			MIXLASSO, $\widehat{\sigma} = 0.25$	
	Exp.1	Exp.2	Gate	Exp.1	Exp.2
x_0	6.0472	6.7101	-0.3958	6.41	7.00
x_1	-0.0073	-0.0197	0.1238	-	-0.32
x_2	-0.0283	0.1377	0.1315	-	0.29
x_3	0.0566	-0.4746	1.5379	-	-0.70
x_4	0.3859	0.5761	-1.9359	0.20	0.96
x_5	-0.2190	-0.0170	-0.9687	-	-
x_6	-0.0586	0.0178	0.4477	-	-
x_7	-0.0430	0.0242	-0.3682	-0.19	-
x_8	0.3991	0.0085	1.7570	0.26	-
x_9	-0.0238	-0.0345	-1.3150	-	-
x_{10}	-0.1944	0.0412	0.6550	-	-
x_{11}	0.0726	0.1152	0.0279	-	-
x_{12}	0.0250	-0.0823	0.1383	-	-
x_{13}	-2.7529	1.1153	-7.0559	0.79	0.70
x_{14}	2.3905	-1.4185	5.6419	0.72	-
x_{15}	-0.0386	1.1150	-2.8818	0.15	0.50
x_{16}	0.2380	0.0917	-7.9505	-	-0.36
$x_1 * x_{13}$	3.3338	-0.8335	8.7834	-0.21	-
$x_1 * x_{14}$	-2.4869	2.5106	-7.1692	0.63	-
$x_1 * x_{15}$	0.4946	-0.9399	2.6319	0.34	-
$x_1 * x_{16}$	-0.4272	-0.4151	7.9715	-	-
$x_3 * x_{13}$	0.7445	0.3201	0.5622	-	-
$x_3 * x_{14}$	-0.0900	-1.4934	0.1417	0.14	-0.38
$x_3 * x_{15}$	-0.2876	0.4381	-0.9124	-	-
$x_3 * x_{16}$	-0.2451	-0.2242	-5.6630	-0.18	0.74
$x_7 * x_{13}$	0.7738	0.1335	4.3174	-	-
$x_7 * x_{14}$	-0.1566	1.2809	-3.5625	-	-
$x_7 * x_{15}$	-0.0104	0.2296	-0.4348	-	0.34
$x_7 * x_{16}$	0.5733	-0.2905	3.2613	-	-
$x_8 * x_{13}$	-1.6898	-0.0091	-8.7320	0.29	-0.46
$x_8 * x_{14}$	0.7843	-1.3341	6.2614	-0.14	-
$x_8 * x_{15}$	0.3711	-0.4310	0.8033	-	-
$x_8 * x_{16}$	-0.2158	0.7790	2.6731	-	-

Table 3.13: Fitted models for baseball salary data with the standard MoE using MLE and MIXLASSO (Khalili and Chen, 2007).

Discussion for the high-dimensional setting

To evaluate the algorithm in a situation in which we have a high number of features, we consider the Residential Building Data Set (UCI Machine Learning Repository). This data set contains 372 and 108 features with the two response variables (V-9 and V-10), which represent the sale prices and construction costs. All the features are standardized to have zero-mean and unit-variance. We choose the V-9 variable (sale prices) as the response variable to be predicted.

Features	MoE-Lasso+ℓ_2 (CA), $\widehat{\sigma} = 0.345$			MoE-Lasso+ℓ_2 (PN), $\widehat{\sigma} = 0.345$		
	Exp.1	Exp.2	Gate	Exp.1	Exp.2	Gate
x_0	5.9580	6.9297	0.0046	5.9570	6.9295	0.0037
x_1	-0.0122	-	-	-0.0117	-	-
x_2	-0.0064	-	-	-0.0068	-	-
x_3	-	-	-	-	-	-
x_4	0.4521	0.0749	-	0.4512	0.0751	-
x_5	-	-	-	-	-	-
x_6	-0.0051	-	-	-0.0050	-	-
x_7	-	-	-	-	-	-
x_8	-	0.0088	-	-	0.0066	-
x_9	0.0135	0.0192	-	0.0140	0.0204	-
x_{10}	-0.1146	-	-	-0.1143	-	-
x_{11}	-0.0108	0.0762	-	-0.0107	0.0769	-
x_{12}	-	-	-	-	-	-
x_{13}	-	0.3855	-0.3946	-	0.3837	-0.3926
x_{14}	0.0927	-0.0550	-	0.0932	-0.0540	-
x_{15}	0.3268	0.3179	-	0.3273	0.3179	-
x_{16}	-	-	-	-	-	-
$x_1 * x_{13}$	0.3218	-	-	0.3203	-	-
$x_1 * x_{14}$	-	-	-	-	-	-
$x_1 * x_{15}$	-	-	-	-	-	-
$x_1 * x_{16}$	-0.0319	-	-	-0.0321	-	-
$x_3 * x_{13}$	-	0.0284	-0.5828	-	0.0254	-0.5876
$x_3 * x_{14}$	-0.0883	-	-	-0.0878	-	-
$x_3 * x_{15}$	-	-	-	-	-	-
$x_3 * x_{16}$	-	-	-	-	-	-
$x_7 * x_{13}$	-	0.004	-	-	0.014	-
$x_7 * x_{14}$	-0.1362	0.0245	-	-0.1365	0.0260	-
$x_7 * x_{15}$	-	-	-	-	-	-
$x_7 * x_{16}$	-	-	-	-	-	-
$x_8 * x_{13}$	-	0.2727	-0.3628	-	0.2801	-0.3621
$x_8 * x_{14}$	-	0.0133	-	-	0.0109	-
$x_8 * x_{15}$	0.3154	-	-	0.3151	-	-
$x_8 * x_{16}$	0.0157	-	-	0.0160	-	-

Table 3.14: Fitted models for baseball salary data with our proposed algorithms: MoE-Lasso + ℓ_2 (CA) and MoE-Lasso + ℓ_2 (PN).

Histogram of this variable is given in Figure 3.8. A MoE model with $K = 3$ expert components is chosen to fix this data set. The MoE-Lasso +ℓ_2 (PN) method is used to estimate the parameters. We provide the results of our algorithm with $K = 3$ and $\lambda = 15$, $\gamma = 5$. The estimated parameters are given in Table 3.17, Table 3.18 and Table 3.19.

The correlation and the mean squared error between the true value V-9 with its prediction can be found in Table 3.20. These results show that the proximal Newton method performs well

Model	Predicted criteria			
	Method 1		Method 2	
	R^2	MSE	R^2	MSE
MoE	0.8099	$0.2625_{(.758)}$	0.9474	$0.0726_{(.113)}$
MoE-Lasso+ℓ_2 (CA)	0.8020	$0.2821_{(.633)}$	0.9249	$0.1044_{(.169)}$
MoE-Lasso+ℓ_2 (PN)	0.8020	$0.2822_{(.634)}$	0.9249	$0.1044_{(.169)}$
MIXLASSO	0.4252	$1.1858_{(2.792)}$	0.6268	$0.6549_{(1.532)}$

Table 3.15: Results for Baseball salaries data set.

Data	No. features	No. observations	No. experts	CA	PN
Simulation	7	300	2	$45.34_{(14.28)}$ (s)	$5.03_{(1.09)}$ (s)
Simulation	7	300	3	$7.94_{(13.22)}$ (m)	$20.52_{(9.23)}$ (s)
Baseball salaries	33	337	2	$17.9_{(15.87)}$ (m)	$46.76_{(21.02)}$ (s)
Residential Data	108	372	3	N/A	$3.63_{(0.58)}$ (m)

Table 3.16: Results for CPU times.

in this setting, in which it provides a sparse model and competitive criteria in prediction and clustering. We also provide the correlation and the mean squared error between those values after clustering the data in Table 3.21.

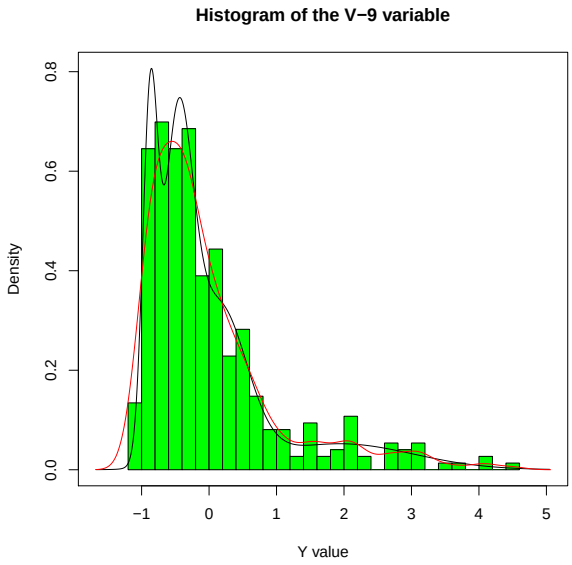
Method	Predictive criteria		Number of zero coefficients				
	R^2	MSE	Exp.1	Exp.2	Exp.3	Gate.1	Gate.2
Proximal Newton	0.991	$0.0093_{(.059)}$	71	38	75	97	101

Table 3.20: Results for residential building data set by using the clustering method 1.

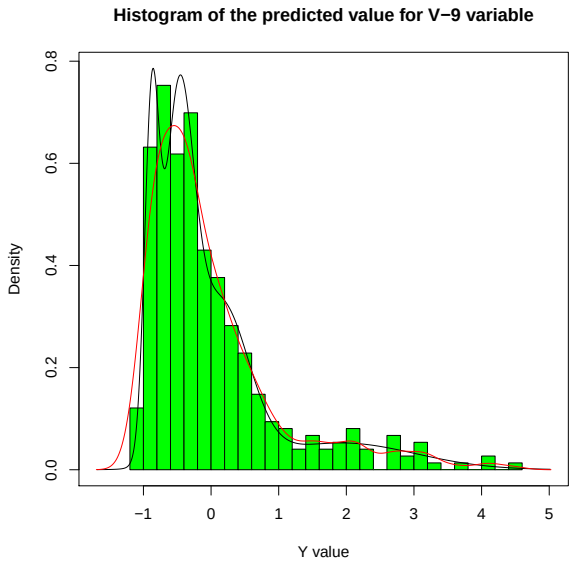
Method	Predictive criteria		Number of observations		
	R^2	MSE	Class 1	Class 2	Class 3
Proximal Newton	0.9994	$0.00064_{(.0018)}$	59	292	21

Table 3.21: Results for the residential building data set by using the clustering method 2.

An experiment for $p > n$: To consider the high-dimensional setting, we take the first $n = 90$ observations of the residential building data with all the $p = 108$ features. Histogram of the variable V-9 in this case is provided in Figure 3.9. We use a mixture of three experts and obtain the results by applying the MoE-Lasso+ℓ_2 (PN) algorithm. The parameter estimation results are provided in Table 3.24, Table 3.25 and Table 3.26. The results in terms of correlation and the mean squared error between the true value V-9 and its predictions are given in Table 3.22 and Table 3.23. From these Tables we can see that, in this high-dimensional setting, one still obtains acceptable results for the regularized MoE models and the EM algorithm using the proximal Newton method (MoE-Lasso+ℓ_2 (PN)) is an efficient tool for the parameter estimation. The running time in this experiment is about only few (~ 8) minutes and the algorithm is quite effective in this setting.

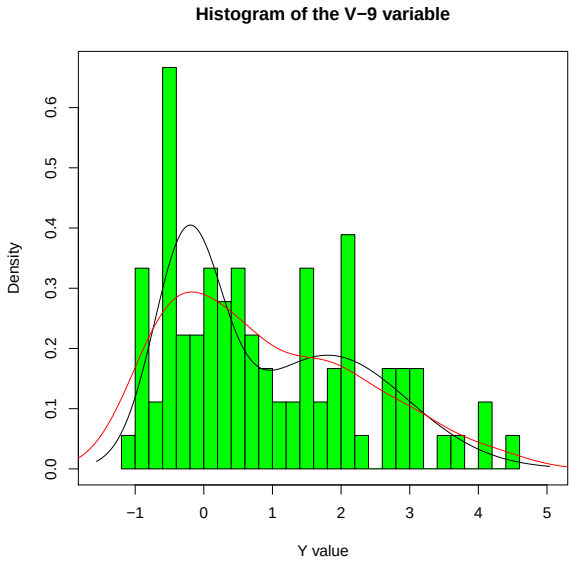


Histogram of Y

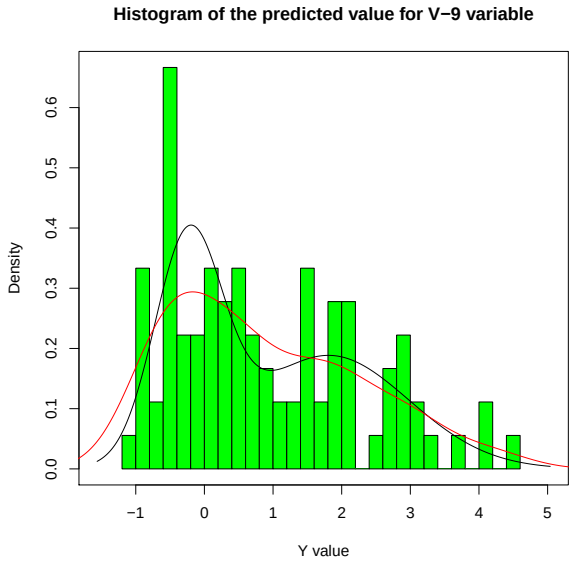


Histogram of $\hat{Y}$

Figure 3.8: Histograms of the variable Y (V-9) and its predicted value for residential building data. The red lines represent the estimated densities using non-parametric kernel density estimation, and the black lines show the densities estimated using a GMM.



Histogram of Y



Histogram of $\hat{Y}$

Figure 3.9: Histograms of the variable Y (V-9) and its predicted value for the subset of residential building data. The red lines are the estimated densities using non-parametric kernel density estimation, and the black lines show the estimated densities using a GMM.

Features	Expert, $\sigma = 0.0255$			Gating network	
	Exp.1	Exp.2	Exp.3	Gate.1	Gate.2
x_0	-0.00631	-0.01394	-0.07825	0.43542	2.40874
x_1	-	-	0.00599	-	-
x_2	0.02946	-0.00442	-	-	-
x_3	-	-	0.00849	-	-
x_4	-0.00776	0.00406	0.01485	-	-
x_5	-0.00619	-0.00759	-0.04185	-0.23943	-
x_6	0.00125	0.02581	-	-	-
x_7	-	-0.01823	0.00233	-	-
x_8	0.02271	-0.01962	0.01964	-0.04267	-
x_9	0.06822	0.00274	0.02101	-	-
x_{10}	-0.03166	-0.00008	-	-	-
x_{11}	0.12789	0.05117	0.03515	-	-0.91114
x_{12}	1.10946	1.00213	0.78915	0.22049	-0.71761
x_{13}	0.00878	-0.00647	-	0.41648	-
x_{14}	-	-	-	-	-
x_{15}	-	-	-	-	-
x_{16}	-0.01495	-0.00103	0.03774	-	-
x_{17}	-	-	-	-	-
x_{18}	-	-0.03344	-	-	-
x_{19}	-	0.06296	-	-	-
x_{20}	0.04560	0.02466	-	-	-
x_{21}	0.02368	0.03210	-	-	-
x_{22}	-	-0.00546	-0.00398	-	-
x_{23}	-	-0.03934	-	-	-
x_{24}	-	-0.04612	-	-	-
x_{25}	0.01205	-0.00352	-	-	-
x_{26}	-	-	-	-	-
x_{27}	-	0.00409	-	-	-
x_{28}	-	-	-	-	-
x_{29}	-	-	0.00047	-	-
x_{30}	-	-	-	-	-
x_{31}	-	0.03494	0.04131	-	-
x_{32}	-	-0.00003	0.02288	-	-
x_{33}	-	-	-	-	-
x_{34}	-	-	-	-	-
x_{35}	-	0.01468	-0.01095	-	-

Table 3.17: Fitted model parameters for residential building data obtained by our MoE-Lasso $+\ell_2$ (PN) algorithm (part 1).

	Predictive criteria		Number of zero coefficients				
Method	R^2	MSE	Exp.1	Exp.2	Exp.3	Gate.1	Gate.2
Proximal Newton	0.9895	$0.0204_{(.056)}$	31	60	55	106	104

Table 3.22: Results for the subset of the residential building data set by using the clustering method 1.

Features	Expert, $\sigma = 0.0255$			Gating network	
	Exp.1	Exp.2	Exp.3	Gate.1	Gate.2
x_{36}	-	-	-	-	-
x_{37}	-	0.00899	-	-	-
x_{38}	-	0.00061	-	-	-
x_{39}	-0.01694	-0.00559	-	-	-
x_{40}	0.10214	0.02533	-	0.07086	-
x_{41}	0.03770	-	-	-	-
x_{42}	-	-0.04162	-	-	-
x_{43}	-	-	-	-	-
x_{44}	-	0.00561	0.01148	-	-
x_{45}	-	0.00770	-	-	-
x_{46}	-	-	-	-	-
x_{47}	-	-	-	-	-
x_{48}	-0.07316	0.03138	-	-	-
x_{49}	-	0.00493	-0.00183	-	-
x_{50}	-	0.01320	-	-	-
x_{51}	-0.00076	-0.00041	-	-	0.03819
x_{52}	-	-	-	-	-
x_{53}	-	-	-	-	-
x_{54}	-0.00854	0.00077	-	-	-
x_{55}	-	0.00039	-	-	-
x_{56}	-	-	-0.11177	-	-
x_{57}	-	0.00334	-	-	-
x_{58}	0.04779	0.00405	0.00733	0.35226	-
x_{59}	0.06726	0.03743	0.02988	0.08489	-0.20694
x_{60}	0.02520	0.00128	0.01473	-	-
x_{61}	-	0.00843	-	-	-
x_{62}	-	0.00034	-	-	-
x_{63}	-	-0.00920	0.01184	-	-
x_{64}	-	0.00002	-	-	-
x_{65}	-	-	-	-	-
x_{66}	-	-	-	-	-
x_{67}	-0.03840	-	0.02505	-	-
x_{68}	-	0.00234	0.00238	-	-
x_{69}	-	-	-	-	-
x_{70}	0.06026	0.01750	0.05879	-	-
x_{71}	-	-	-	-	-

Table 3.18: Fitted model parameters for residential building data obtained by our MoE-Lasso $+\ell_2$ (PN) algorithm (part 2).

	Predictive criteria		Number of observations		
Method	R^2	MSE	Class 1	Class 2	Class 3
Proximal Newton	0.9999	$0.00025_{(.0014)}$	63	11	16

Table 3.23: Results for the subset of the residential building data set by using the clustering method 2.

Features	Expert, $\sigma = 0.0255$			Gating network	
	Exp.1	Exp.2	Exp.3	Gate.1	Gate.2
x_{72}	-	-0.03636	-	-	-
x_{73}	-	-	-0.02932	-	-
x_{74}	-	-	-	-	-
x_{75}	-0.02725	-0.02474	-	-	-
x_{76}	-0.01399	-0.16005	-0.08654	-	-
x_{77}	-	0.00526	-	-	-
x_{78}	-0.05816	0.02821	-	0.01303	-0.35566
x_{79}	-	-0.00358	-	1.12522	-
x_{80}	-0.05416	-	-	-	-
x_{81}	-	-	-	-	-
x_{82}	-	-	0.04329	-	-
x_{83}	-	-	-	-	-
x_{84}	-	-	-	-	-
x_{85}	-	-	-	-	-
x_{86}	-	0.00783	-	-	-
x_{87}	-	-	0.01463	-	-
x_{88}	0.02337	0.03903	-	-	-
x_{89}	-0.04720	0.00909	-	-	-
x_{90}	-	-	-	-	-
x_{91}	-	-	-	-	-
x_{92}	-0.00070	-0.00626	-0.00458	-	-
x_{93}	-	-	-	-	-
x_{94}	-0.00067	0.00309	-	-	-
x_{95}	-	-0.00925	-	-	-
x_{96}	-0.00705	-0.00656	-	-	0.03610
x_{97}	-	-0.00406	-	-	-
x_{98}	-	0.00714	0.01911	0.06610	-
x_{99}	-	0.00364	-	-	-
x_{100}	-	0.00327	-	-	-
x_{101}	-	0.02858	0.03974	-	-
x_{102}	0.01623	-0.01236	-	-	-
x_{103}	-	-	-	-	-
x_{104}	-	-	-	-	-
x_{105}	-	0.00215	-	-	-
x_{106}	-0.00006	-0.00129	-	-	-
x_{107}	-	0.00851	-	-	-

Table 3.19: Fitted model parameters for residential building data obtained by our MoE-Lasso $+\ell_2$ (PN) algorithm (part 3).

3.5 Conclusion and future work

In this chapter, we proposed a regularized MLE for the MoE model which encourages sparsity, and developed three versions of a blockwise EM algorithm to monotonically maximize this

Features	Expert, $\sigma = 0.0159$			Gating network	
	Exp.1	Exp.2	Exp.3	Gate.1	Gate.2
x_0	0.09048	0.21992	0.05460	0.73646	-0.54048
x_1	-	-	-	-	-
x_2	0.00837	-	0.00112	-	-
x_3	-	-	-	-	-
x_4	0.04498	0.07325	0.00001	-	-
x_5	0.08075	0.00807	0.00010	-	-
x_6	-0.00836	-	-0.02235	0.02205	-
x_7	0.01337	-0.00009	-0.00922	-	-
x_8	0.02375	0.00443	0.00668	-	-
x_9	0.02194	0.00379	-0.03344	-	-
x_{10}	-0.01305	-0.00079	0.00560	-	-
x_{11}	0.12763	0.01256	0.08537	-	0.16264
x_{12}	1.08977	0.72843	1.04263	-	-
x_{13}	0.00171	0.09792	-	-	-
x_{14}	-0.03158	-	-	-	-
x_{15}	-	-	-0.00001	-	-
x_{16}	-0.02218	0.00987	-0.00527	-	-
x_{17}	-	-	-	-	-
x_{18}	-	-	-0.10258	-	-
x_{19}	-0.06036	-	-	-	-
x_{20}	0.03513	-	-0.00602	-	-
x_{21}	0.01947	0.12495	0.07810	-	-
x_{22}	-0.00347	0.01317	-	-	-
x_{23}	-0.03255	-0.00125	-	-	-
x_{24}	-0.06659	-0.00007	-	-	-
x_{25}	0.03478	-	0.01314	-	-
x_{26}	0.01209	0.03787	-0.00287	-	-
x_{27}	-	-	-	-	-
x_{28}	-	-	-	-	-
x_{29}	0.06476	0.02369	-0.00461	-	-
x_{30}	-0.01017	-0.00813	0.01805	-	-
x_{31}	0.03331	-	-	-	-
x_{32}	-0.03870	0.01708	-	-	-
x_{33}	-	-	-	-	-
x_{34}	-	-	-	-	-
x_{35}	0.02278	-0.02794	0.01933	-	-

Table 3.24: Fitted model parameters for the subset of residential building data obtained by our MoE-Lasso $+\ell_2$ (PN) algorithm (part 1).

regularized objective towards at least a local maximum. The proposed regularization does not require using approximations as in standard MoE regularization. The proposed algorithms are based on univariate updates of the model parameters via an MM, coordinate ascent, proximal Newton method, which allow to tackle matrix inversion problems and obtain sparse solutions.

Features	Expert, $\sigma = 0.0159$			Gating network	
	Exp.1	Exp.2	Exp.3	Gate.1	Gate.2
x_{36}	-	-	-	-	-
x_{37}	-0.09359	-	-0.06125	-	-
x_{38}	-	-	-0.00356	-	-
x_{39}	-0.11611	-	-0.01973	-	-
x_{40}	0.21178	0.06134	0.13879	-	-
x_{41}	0.09095	-	-	-	-
x_{42}	-0.03243	-	-	-	-
x_{43}	-0.00032	-	-0.01455	-	-
x_{44}	-0.01643	-	-	-	-
x_{45}	-0.03152	0.01812	-0.02303	-	-
x_{46}	-	-	-	-	-
x_{47}	-	-	-	-	-
x_{48}	0.13661	0.00862	-	-	-
x_{49}	0.04914	0.06704	-	-	-
x_{50}	0.00424	-	-0.02954	-	-
x_{51}	0.04225	0.05518	-0.01411	-	-
x_{52}	-	-	-	-	-
x_{53}	-0.01697	-	-	-	-
x_{54}	0.02922	0.00057	-0.00501	-	-
x_{55}	-	-	-	-	-
x_{56}	-	-0.02272	0.00131	-	-
x_{57}	-	-	-	-	-
x_{58}	0.11223	-	0.05349	-	-
x_{59}	0.23868	-0.00711	0.07830	-	-
x_{60}	-0.07807	-0.05727	-	-	-0.02819
x_{61}	-0.06729	-	-	-	-
x_{62}	-0.02121	-	-	-	-
x_{63}	-0.01886	0.04294	0.00548	-	-
x_{64}	-0.01265	0.02236	-	-	-
x_{65}	-	-	-	-	-
x_{66}	-	-	-	-	-
x_{67}	-0.03609	-	-	-	-
x_{68}	-0.07929	0.01190	-0.00001	-	-
x_{69}	-	-	-	-	-
x_{70}	0.09774	-0.01388	0.01683	-	-
x_{71}	-	-	-	-	-

Table 3.25: Fitted model parameters for the subset of residential building data obtained by our MoE-Lasso $+\ell_2$ (PN) algorithm (part 2).

The results in terms of parameter estimation, the estimation of the actual support of the sparsity, and clustering accuracy, obtained on simulated and three real data sets, confirm the effectiveness of our proposal at least for problems of moderate dimension. Namely, the model sparsity does not include significant bias in terms of parameter estimation nor in terms of recovering the actual

Features	Expert, $\sigma = 0.0159$			Gating network	
	Exp.1	Exp.2	Exp.3	Gate.1	Gate.2
x_{72}	-0.08791	-	-	-	-
x_{73}	-0.06590	-0.13467	0.03526	-	-
x_{74}	0.05718	-	-	-	-
x_{75}	-0.14786	-0.03133	-	-	-
x_{76}	-0.12865	-0.07620	-0.09485	-	-
x_{77}	0.04578	0.04694	-	-	-
x_{78}	0.01510	0.01860	0.08887	-	-
x_{79}	-0.00755	0.00441	0.01526	-	-0.56947
x_{80}	-0.06835	-	-	-	-
x_{81}	-	-	-0.00166	-	-
x_{82}	-0.07267	-	-	-	-
x_{83}	-0.00061	0.02782	-	-	-
x_{84}	-	-	-	-	-
x_{85}	-	-	-	-	-
x_{86}	-0.02223	0.02194	0.03417	-	-
x_{87}	0.00029	-	-	-	-
x_{88}	-	-	-	-	-
x_{89}	-0.06311	0.03682	-0.00977	-	-
x_{90}	-	-	-	-	-
x_{91}	-	-	-	-	-
x_{92}	0.06938	-0.03040	-0.00542	-	-
x_{93}	-	-	-	-	-
x_{94}	0.05246	-	-0.00793	-	-
x_{95}	-0.01214	-	-0.00345	-	-
x_{96}	-	-0.06544	-0.00007	-	-
x_{97}	0.03763	-	-	-	-
x_{98}	0.04560	0.04346	0.00717	-	-
x_{99}	0.03892	-	-0.01578	-	-
x_{100}	0.01633	-	-0.01509	-	-
x_{101}	0.04869	0.01218	0.00076	-	-
x_{102}	-0.01996	-	-	-	-
x_{103}	-	-	-	-	-
x_{104}	-	-	-	-	-
x_{105}	-0.00248	-	-	-	-
x_{106}	-0.00344	-0.03221	0.01461	-	-
x_{107}	-0.00779	-0.01415	0.00106	-	-

Table 3.26: Fitted model parameters for the subset of the residential building data obtained by our MoE-Lasso $+\ell_2$ (PN) algorithm (part 3).

clusters of the heterogeneous data. The obtained models with the proposed approach are sparse which promote its scalability to high-dimensional problems. The hybrid EM/MM algorithm is a potential approach. However, this model should be considered carefully, especially for non-smooth penalty functions. The coordinate ascent approach for maximizing the M-step, however,

works quite well although, while we do not have the closed form update in this situation. A proximal Newton extension is possible to obtain closed form solutions for an approximate of the M-step as an efficient method that is promoted to deal with high-dimensional data sets. First experiments on an example of a quite high-dimensional scenario with a subset of real data containing 90 observations and 108 features provide encouraging results. The next chapter will consist of investigating more the high-dimensional setting as well as considering MoE for discrete data.

For further research directions, recently, Jiang et al. (2018) proposed a regularized approach for localized MoE models (Xu et al., 1995), in which the covariate variable $\boldsymbol{X}$ is assumed to have a mixture of Gaussian distributions and corresponds to each component the respond variable Y is described by a Gaussian regression function. The penalized part emphasizes on feature selection in the regression parts. Thus, it restricts their model since the covariance matrix of each mixture component cannot be estimated efficient in high-dimensional scenario. We can consider a sparse representations of these covariance matrices via the work of Fop et al. (2019). The method introduce in this chapter can directly applied for the hierarchical MoE models (Jordan and Jacobs, 1994). However, theoritical results to support the penalized method in hierarchical MoE framework should be studied more.

This chapter has lead to the following publications Huynh and Chamroukhi (2018); Chamroukhi and Huynh (2018, 2019).

Chapter 4

Regularized mixtures of experts models for discrete data

Contents

4.1 Introduction

The work presented in this chapter involves regularized MoE models for discrete data, including classification. Specifically, we consider the MoE models for counting data and classification data. While the MoE fitting by maximum likelihood (MLE) is widely used, the study of MoE in high-dimensional problems is still challenging due to the well-known problems of the ML

estimator in such a setting. Indeed, when the number of features in the data becomes being large, the features can be correlated and therefore the number of actual predictors/features that explain the problem are smaller. Additionally, numerical instability can also arise in the MLE of a MoE model in high-dimensional setting. For example in regression, maximizing the log-likelihood function leads to using large positive and negative estimates for the regression coefficients, corresponding to the correlated features when the number of features is moderate or large and highly correlated. This behavior can be observed in logistic regression; see Park and Hastie (2007b) and Bunea (2008) for more details. In a MoE scenario, estimating the parameters with moderate numbers of features and mixture components using MLE is challenging. A better fitting can indeed be achieved by regularizing the objective function so that to encourage sparse solutions. Feature selection by regularized inference encourages sparse solutions, with a reasonable computational cost.

Several approaches have been proposed to deal with the feature selection task. The well-known Lasso method Tibshirani (1996) is one of the most popular and successful regularization technique that encourages sparsity, which utilizes the ℓ_1 penalty to regularize the squared error function and achieve parameter estimation and feature selection. Extensions of the Lasso, based on penalized log-likelihood criteria with convex and nonconvex penalty functions has been proposed, including elastic net (Zou and Hastie, 2005), group Lasso (Yuan and Lin, 2006b), adaptive Lasso (Zou, 2006), smoothly clipped absolute deviation (SCAD) (Fan and Li, 2001), minimax concave penalty (MCP) (Zhang, 2010). Each method has its own advantages. The convex penalty functions are easy to handle due to the existence of efficient techniques from convex optimization to fit the models, while the nonconvex penalty functions involve practical challenges in fitting these models.

In a related model, Peralta and Soto (2014) considered MoE with logistic regression model for the experts and proposed an EM algorithm based on inverting the soft-max function to estimate their Lasso regularized logistic MoE model. Unfortunately, the authors did not give any evidence that their EM algorithm improves the objective function after each iteration loop. To tackle the difficulty of updating the coefficients of the gating network, Jiang et al. (2018) introduced a penalized likelihood method for the localized MoE models (Xu et al., 1995). One limitation of their method lies in the fact that the local covariance matrix is updated normally in the M-step. Thus, it poses some disadvantages if one would like to apply their method in high-dimensional scenario.

In this chapter, we propose an efficient regularized estimation and feature selection of Mixtures-of-Experts that encourages sparse solutions and consider MoE models for two common generalized linear models. We develop a proximal Newton-EM algorithm to maximize the proposed ℓ_1-penalized log-likelihood function, in which a proximal Newton-type method for maximizing the M-step is used. An advantage of using proximal Newton-type method lies in the fact that one just need to solve weighted quadratic Lasso problems to update the parameters. Efficient tools such as coordinate ascent algorithm can be used to deal with these problems.

Hence, the proposed approach does not require an approximate of the regularization term, and allow to automatically select sparse solutions without thresholding. Our approach is shown to perform well including in a high-dimensional setting and to outperform competitive state of the art regularized MoE models on several experiments on simulated and real data.

This chapter is organized as follows. Section 4.2 describes the modeling with MoE for heterogeneous data and maximum-likelihood parameter estimation. Then, in Section 4.3, the proposed regularized maximum likelihood strategy of the MoE models and the EM-based algorithm are developed. Our method is applied on simulated and real data sets in Section 4.4. Finally, we also discuss the effectiveness of our method in dealing with moderate dimensional problems and provide an experiment study to promotes our method in high-dimensional setting in Section 4.5.

4.2 Mixture of experts and maximum likelihood estimation for discrete data

Let $((\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n))$ be a random sample of n independently distributed pairs $(\boldsymbol{X}_i, Y_i)$, $(i = 1, \ldots, n)$ where $Y_i \in \mathcal{Y} \subset \mathbb{N}$ is the ith discrete response variable given some vector of $p \in \mathbb{N}$ predictors $\boldsymbol{X}_i \in \mathcal{X} \subset \mathbb{R}^p$, and let $\mathcal{D} = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n))$ be an observed data sample. We consider the MoE modeling for the analysis of a heteregeneous set of such data (see Section 2.3.3). The model can be summarized as follows:

4.2.1 The mixture of experts model for discrete data

Assume that the observed pairs $(\boldsymbol{x}, y)$ are generated from $K \in \mathbb{N}$ parametric probability mass components (the experts) $p_z(y|\boldsymbol{x}; \boldsymbol{\theta})$, $z \in [K] = \{1, \ldots, K\}$, governed by a gating network $\pi_z(\boldsymbol{x}; \mathbf{w})$ represented by a hidden categorical random variable $Z \in [K]$ that indicates the expert to which a particular observed pair belongs. In MoE framework, these gating network functions $\pi_z(\boldsymbol{x}; \mathbf{w})$ are modeled by logistic functions. Hence, given the predictor or the input $\boldsymbol{x}_i$, the categorical variable Z_i is generated according to the multinomial distribution:

$$Z_i|\boldsymbol{x}_i \sim \text{Mult}(1; \pi_1(\boldsymbol{x}_i; \mathbf{w}), \ldots, \pi_K(\boldsymbol{x}_i; \mathbf{w})) \tag{4.1}$$

where

$$\pi_k(\boldsymbol{x}_i; \mathbf{w}) = \mathbb{P}(Z_i = k|\boldsymbol{X}_i = \boldsymbol{x}_i) = \frac{\exp(w_{k0} + \boldsymbol{x}_i^T \boldsymbol{w}_k)}{1 + \sum_{l=1}^{K-1} \exp(w_{l0} + \boldsymbol{x}_i^T \boldsymbol{w}_l)},$$

such that $\mathbf{w}_k = (w_{k0}, \boldsymbol{w}_k^T)^T$ and $\mathbf{w}_K = \mathbf{0}$ is set to the null vector for identifiability (Jiang and Tanner, 1999a). Then, conditional on the hidden variable $Z_i = z_i$ and $\boldsymbol{x}_i$, the observed random variable Y_i is assumed to be generated from the expert z_i distribution $p_{z_i}(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}_{z_i})$, that is:

$$Y_i|Z_i = z_i, \boldsymbol{X}_i = \boldsymbol{x}_i \sim p_{z_i}(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}_{z_i}) \tag{4.2}$$

where $p_{z_i}(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_{z_i}) = p(y_i|Z_i = z_i, \boldsymbol{X}_i = \boldsymbol{x}_i;\boldsymbol{\theta}_{z_i})$ is the probability mass function of the expert z_i depending on the nature of the data $(\boldsymbol{x}, y)$ within the group z_i. Hence, formally, the MoE is defined by the following semi-parametric probability mass function:

$$p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}_i;\mathbf{w}) p_k(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_k) \tag{4.3}$$

with the parameter vector defined by $\boldsymbol{\theta} = (\mathbf{w}_1^T, \ldots, \mathbf{w}_{K-1}^T, \boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_K^T)^T$ where $\boldsymbol{\theta}_k$ $(k = 1, \ldots, K)$ is the parameter vector of the kth expert.

For a complete account of MoE, types of gating networks and expert networks, the reader is referred to Nguyen and Chamroukhi (2018).

4.2.2 Maximum likelihood parameter estimation

For an observed data sample $\mathcal{D} = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n))$ generated from the MoE model (4.3), the unknown parameter vector $\boldsymbol{\theta}$ is commonly estimated by maximizing the observed data log-likelihood

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k(\boldsymbol{x}_i;\mathbf{w}) p_k(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_k), \tag{4.4}$$

by using the EM algorithm. Unfortunately, the MLE can be unstable or even infeasible in high-dimension due to possibly redundant and correlated features. In some cases, such as multi-logistic model, this task becomes a challenge since the log-likelihood function becomes singular. Indeed, Rosset et al. (2004) showed that in logistic regression if the data are separable by the predictors the maximum likelihood solutions are undefined. In such cases, a regularization of the MLE is needed. For example, Park and Hastie (2007a) considered adding a small amount of quadratic penalization, and let the coefficients converge to the ℓ_2 penalized logistic regression solutions instead of infinity. Here, we consider the regularized models with take into account the well-known Lasso penalty.

4.3 Regularized maximum likelihood estimation

Regularized MLE allows the selection of a relevant subset of features for prediction and thus encourages sparse solutions. This approach also bounds the norm of the estimated parameters. Hence, it avoids the singularity of the penalized log-likelihood. In mixture-of-experts context, one may consider both sparsity in the feature space of the gates, and of the experts. We utilize Lasso penalty for the experts, and for detecting zero and nonzero coefficients in the mixing functions another Lasso penalty is added for the gating parameters. This approach has some advantages since the Lasso penalty functions are convex and convenient to handle. Moreover, by adding these penalty functions, we also force some parameters, not only in the experts but also in the gating networks, toward zero, to obtain a sparse model. The proposed regularization that combines two Lasso penalties for the experts parameters, and for the gating network is

defined by:

$$PL(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) - \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1 - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1. \tag{4.5}$$

where $\|\boldsymbol{v}\|_1 = \sum_{j=1}^{p} |v_j|$ is the ℓ_1 norm of a vector $\boldsymbol{v} \in \mathbb{R}^p$, $\lambda_k \geq 0$ for all $k = 1, \ldots, K$ and $\gamma_k \geq 0$ for all $k = 1, \ldots, (K-1)$. The regularization parameters λ_k and γ_k control the amount of shrinkage on the parameters $\boldsymbol{\beta}_k$ and $\boldsymbol{w}_k$. A similar strategy has been proposed in Khalili (2010) for Gaussian regression based on two well-known penalized techniques: Lasso (Tibshirani, 1996) and SCAD (Fan and Li, 2001) which are then approximated in the EM algorithm of the model inference. An ℓ_2 penalty function for the gating network is added to avoid wildly large positive and negative estimates of the regression coefficients corresponding to the mixing proportions. This behavior can be observed in logistic/multinomial regression when the number of potential features is large and they are highly correlated (Park and Hastie, 2007b; Bunea, 2008). However, the ℓ_2 norm also affect the sparsity of the models (see Section 3.4). We therefore remove this ℓ_2 penalty in our proposal model. For parameter estimation, Khalili (2010) introduced an EM algorithm follows the suggestion of Hunter and Li (2005). Unfortunately, this approach draws some inadequacies (see discussion in Section 3.3) which can be tackled by our proposals in Chapter 3.

In a similar scenario, Peralta and Soto (2014) suggested an EM algorithm for the regularized MoE of logistic regression, in which using a transformation that implies inverting the soft-max function. However, there is no evidence to ensure the increasing of their penalized log-likelihood values and this leads to the poor results from their approach. In our approach presented here, we propose and EM algorithm which relies on proximal Newton-type procedures in the M-step to overcome these limitations. We consider that in mixture of experts with two different models for the experts, that is Poisson, and logistic regressors.

4.3.1 Parameter estimation and feature selection via a proximal Newton-EM

For each of the two considered GLM for the MoE models, we propose an EM algorithm to monotonically find at least local maximizers of (4.5). The E-step is common to these models. For the M-step, two different algorithms are proposed to update the model parameters. Specifically, the first one relies on proximal Newton method, while the second one uses a proximal Newton-type method to update the gating network and expert's parameters. The difference between these algorithms is that the proximal Newton-type method we construct here to update the gating network can avoid the numerical instability of the proximal Newton method due to the small value of the mixing proportions. We discuss this difference in Section 4.3.2. The EM algorithm for the maximization of (4.5) requires the construction of the penalized complete-data log-likelihood, which is, in our context, given by

$$PL_c(\boldsymbol{\theta}) = L_c(\boldsymbol{\theta}) - \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1 - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 \tag{4.6}$$

where

$$L_c(\boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik} \log\left[\pi_k(\boldsymbol{x}_i;\mathbf{w}) p_k(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_k)\right] \tag{4.7}$$

is the standard complete-data log-likelihood for the MoE model where z_{ik} an indicator binary-valued variable such that $z_{ik} = 1$ if $Z_i = k$ (i.e., if the ith pair $(\boldsymbol{x}_i, y_i)$ is generated from the kth expert component) and $z_{ik} = 0$ otherwise. Thus, the proposed EM algorithm for the regularized MoE model in its general form runs as follows. After starting with an initial solution $\boldsymbol{\theta}^{[0]}$, it alternates between the two following steps until convergence (e.g., when there is no longer a significant change in the relative variation of (4.5)).

E-step: The E-Step computes the conditional expectation of the penalized complete-data log-likelihood (4.6), given the observed data $\mathcal{D}$ and a current parameter vector $\boldsymbol{\theta}^{[q]}$, q being the current iteration number of the block-wise EM algorithm:

$$\begin{aligned} Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[q]}) &= \mathbb{E}\left[PL_c(\boldsymbol{\theta})|\mathcal{D};\boldsymbol{\theta}^{[q]}\right] \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}^{[q]} \log\left[\pi_k(\boldsymbol{x}_i;\mathbf{w}) p_k(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_k)\right] - \sum_{k=1}^{K}\lambda_k\|\boldsymbol{\beta}_k\|_1 - \sum_{k=1}^{K-1}\gamma_k\|\boldsymbol{w}_k\|_1 \end{aligned} \tag{4.8}$$

where

$$\tau_{ik}^{[q]} = \mathbb{P}(Z_i = k|y_i, \boldsymbol{x}_i;\boldsymbol{\theta}^{[q]}) = \pi_k(\boldsymbol{x}_i;\mathbf{w}^{[q]}) p_k(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_k^{[q]}) / p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}^{[q]}) \tag{4.9}$$

is the conditional probability that the data pair $(\boldsymbol{x}_i, y_i)$ is generated by the kth expert. This step only requires the computation of the conditional component probabilities $\tau_{ik}^{[q]}$ $(i = 1, \ldots, n)$ for each of the K experts.

M-step: The M-Step updates the parameters by maximizing the Q function (4.8) with respect to $\boldsymbol{\theta}$. The Q-function can be written as:

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[q]}) = Q(\mathbf{w};\boldsymbol{\theta}^{[q]}) + \sum_{k=1}^{K} Q_k(\boldsymbol{\theta}_k;\boldsymbol{\theta}^{[q]}) \tag{4.10}$$

with

$$\begin{aligned} Q(\mathbf{w};\boldsymbol{\theta}^{[q]}) &= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}^{[q]} \log \pi_k(\boldsymbol{x}_i;\mathbf{w}) - \sum_{k=1}^{K-1}\gamma_k\|\boldsymbol{w}_k\|_1, \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K-1} \tau_{ik}^{[q]}(w_{k0} + \boldsymbol{x}_i^T\boldsymbol{w}_k) - \sum_{i=1}^{n}\log\Big[1 + \sum_{k=1}^{K-1} e^{w_{k0}+\boldsymbol{x}_i^T\boldsymbol{w}_k}\Big] - \sum_{k=1}^{K-1}\gamma_k\|\boldsymbol{w}_k\|_1. \end{aligned} \tag{4.11}$$

and

$$Q_k(\boldsymbol{\theta}_k;\boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n} \tau_{ik}^{[q]} \log p_k(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_k) - \lambda_k\|\boldsymbol{\beta}_k\|_1. \tag{4.12}$$

The parameters $\mathbf{w}$ are therefore updated by maximizing the function (4.11). Here, the composite function $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$ is concave and does not have the weighted Lasso form. One can use coordinate ascent algorithm to update $\mathbf{w}$ since the penalty part has a separate structure (see Tseng (2001) for more details). However, this approach requires a lot of computing and is not suitable for high-dimensional data (see Chamroukhi and Huynh (2019)). In this case, proximal Newton algorithm and proximal Newton-type algorithm are good choices to overcome these drawbacks. The principle of these methods are described in Appendix C.1. The idea of these approaches lies in the fact that they approximate the smooth part of $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$ with a local quadratic function. After that, one will solve a weighted Lasso regression problem, which has a closed-form update. The solution of this weighted Lasso regression a direction that one can choose to improve the value of $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$ using backtracking line search.

The methods for updating the gating network's parameters using proximal Newton, and proximal Newton-type method are described in the next section.

4.3.2 Proximal Newton-type procedure for updating the gating network

For updating the gating network, the similar approaches which were used to update these parameters for Gaussian outputs in Chapter 3 can be applied once again. Hence, the same proximal Newton and proximal Newton-type algorithms in Section 3.3.3 are the key methods to update $\mathbf{w}$ in this chapter. A little adjustment is provided by setting $\rho = 0$ in this case to have a closed-form update for w_{kj} after each loop of the coordinate ascent algorithm. This is done since we remove the ℓ_2 the penalty in our model.

4.3.3 Proximal Newton-type procedure for updating the experts network

Expert network with Poisson outputs

In this case we consider the situation in which the response Y_i is a count variable and the conditional probability distribution of Y_i, given $\boldsymbol{X}_i$ and Z_i is described as a Poisson distribution. Therefore, the generative model (4.2) of Y is the one of Poisson expert regressor and is given by

$$Y_i|Z_i = z_i, \boldsymbol{x}_i \sim \mathcal{P}_0(.; e^{\beta_{z_i 0} + \boldsymbol{\beta}_{z_i}^T \boldsymbol{x}_i}).$$

Hence, the expert's distribution $p_k(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k)$ becomes

$$p_k(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k) = \mathbb{P}(y_i|\boldsymbol{x}_i; \beta_{k0}, \boldsymbol{\beta}_k) = \frac{\exp[-\exp(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k)] \exp[(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k) y_i]}{y_i!}. \tag{4.13}$$

If the count data Y is such that the probability of zero is large then the zero-inflated Poisson (ZIP) regression model should be considered. For the regularized zero-inflated regression models, we refer the reader to (Buu et al., 2011; Wang et al., 2014; Tang et al., 2014).

Updating the parameter vector for the kth Poisson regressor expert requires the maximization

of the function $Q_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]})$ in (4.12), with

$$Q_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) = \underbrace{\sum_{i=1}^{n} \tau_{ik}^{[q]} \big[-\exp(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k) + y_i(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k) - \log(y_i!) \big]}_{P_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]})} - \lambda_k \|\boldsymbol{\beta}_k\|_1. \tag{4.14}$$

This composite function is concave, nonsmooth and has a non quadratic form. Therefore, the proximal Newton method can be used to update $\boldsymbol{\beta}_k$. Following the strategy that was used to update the gating network, one needs to compute the quadratic approximation $\tilde{P}_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]})$ of $P_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]})$ at $(\tilde{\beta}_{k0}, \tilde{\boldsymbol{\beta}}_k)$. This function is given by (see Appendix C.3.1 for more details)

$$\tilde{P}_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) = -\frac{1}{2} \sum_{i=1}^{n} a_{ik}(b_{ik} - \beta_{k0} - \boldsymbol{x}_i^T \boldsymbol{\beta}_k)^2 + D(\tilde{\beta}_{k0}, \tilde{\boldsymbol{\beta}}_k), \tag{4.15}$$

with

$$\begin{aligned} a_{ik} &= \tau_{ik}^{[q]} \exp(\tilde{\beta}_{k0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}_k); \\ b_{ik} &= \frac{y_i}{\exp(\tilde{\beta}_{k0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}_k)} - 1 + \tilde{\beta}_{k0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}_k; \end{aligned}$$

and $D(\tilde{\beta}_{k0}, \tilde{\boldsymbol{\beta}}_k)$ is a function of $(\tilde{\beta}_{k0}, \tilde{\boldsymbol{\beta}}_k)$.
After that, the coordinate ascent algorithm with soft-thresholding operator is used to maximizing the penalized weighted least-square

$$\max_{(\beta_{k0}, \boldsymbol{\beta}_k)} \tilde{P}_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) - \lambda_k \|\boldsymbol{\beta}_k\|_1. \tag{4.16}$$

Then the solution is taken in account for the next update of the proximal Newton algorithm. This can be interpreted as in Algorithm 3.

Algorithm 3 Proximal Newton method for Poisson model

1: $(\beta_{k0}^{(0)}, \boldsymbol{\beta}_k^{(0)}) = (\beta_{k0}^{[q]}, \boldsymbol{\beta}_k^{[q]})$.
2: **repeat**
3: Update the quadratic approximation $\tilde{P}_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]})$ in (4.15) using the current parameters.
4: Solve the penalized weighted least-square problem in (4.16) by using coordinate ascent algorithm and let $(\tilde{\beta}_{k0}^{(s)}, \tilde{\boldsymbol{\beta}}_k^{(s)})$ be the solution.
5: Set $(\beta_{k0}^{(s+1)}, \boldsymbol{\beta}_k^{(s+1)}) = (1-t)(\beta_{k0}^{(s)}, \boldsymbol{\beta}_k^{(s)}) + t(\tilde{\beta}_{k0}^{(s)}, \tilde{\boldsymbol{\beta}}_k^{(s)})$, where t is found using a backtracking line-search.
6: Evaluate the objective function $Q_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]})$ at $(\beta_{k0}^{(s+1)}, \boldsymbol{\beta}_k^{(s+1)})$.
7: **until** the stopping criterion is satisfied.

Expert network with Multinomial outputs

Finally, for MoE for classification, assuming that each expert part is governed by a multinomial distribution with R (≥ 2) levels and the probability distribution of Y_i given $\boldsymbol{x}_i$ and z_i becomes a multinomial-logistic distribution, i.e, (4.2) is defined by

$$Y_i|Z_i = z_i, \boldsymbol{x}_i \sim \text{Mult}(1; \alpha_{z_i 1}(\boldsymbol{x}_i; \boldsymbol{\beta}_{z_i}), \ldots, \alpha_{z_i R}(\boldsymbol{x}_i; \boldsymbol{\beta}_{z_i}))$$

where

$$\alpha_{kr}(\boldsymbol{x}_i; \boldsymbol{\beta}_k) = \mathbb{P}(y_i = r|\boldsymbol{x}_i; z_i = k) = \frac{\exp(\beta_{kr0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr})}{1 + \sum_{l=1}^{R-1} \exp(\beta_{kl0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_{kl})}, \; r \in \{1, \ldots, R\}$$

with $(\beta_{kR0}, \boldsymbol{\beta}_{kR}) = \mathbf{0}$. Denote by U the $n \times R$ indicator response matrix with elements $u_{ir} = \mathbb{I}(y_i = r)$. Then $Q_k(\boldsymbol{\beta}_k; \boldsymbol{\theta}^{[q]})$ in (4.12) is written in the more explicit form

$$Q_k(\boldsymbol{\beta}_k; \boldsymbol{\theta}^{[q]}) = \underbrace{\sum_{i=1}^{n} \tau_{ik}^{[q]} \Big[\sum_{r=1}^{R-1} u_{ir}(\beta_{kr0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr}) - \log\big(1 + \sum_{r=1}^{R-1} \exp(\beta_{kr0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr})\big) \Big]}_{I(\boldsymbol{\beta}_k)} - \sum_{r=1}^{R-1} \lambda_{kr} \|\boldsymbol{\beta}_{kr}\|_1. \tag{4.17}$$

The same strategy for updating the gating network by using proximal Newton method can be applied in this case. It is not hard to show that the local quadratic approximation $\tilde{I}_r(\boldsymbol{\beta}_k)$ of $I(\boldsymbol{\beta}_k)$ with respect to $(\beta_{kr0}, \boldsymbol{\beta}_{kr})$ at $\tilde{\boldsymbol{\beta}}_k$ is given by (see Appendix C.3.2)

$$\tilde{I}_r(\boldsymbol{\beta}_k) = -\frac{1}{2} \sum_{i=1}^{n} \tau_{ik}^{[q]} d_{ikr}(c_{ikr} - \beta_{kr0} - \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr})^2 + E(\tilde{\boldsymbol{\beta}}_k), \tag{4.18}$$

where

$$c_{ikr} = \tilde{\beta}_{kr0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}_{kr} + \frac{u_{ir} - \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)}{\alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)(1 - \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i))}, \tag{4.19}$$

$$d_{ikr} = \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)(1 - \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)), \tag{4.20}$$

and $E(\tilde{\boldsymbol{\beta}}_k)$ is a function of $\tilde{\boldsymbol{\beta}}_k$.
The corresponding Lasso form is described as following

$$\tilde{I}_r(\boldsymbol{\beta}_k) - \lambda_{kr} \|\boldsymbol{\beta}_{kr}\|_1. \tag{4.21}$$

Using a similar algorithm with Algorithm 2 by replacing the weighted Lasso in (3.29) with (4.21), one can obtain the kth expert's parameter vector.

The proximal Newton-type method can be suggested by replacing the Hessian matrix with the constant matrix $\mathbf{B} = -1/4 \sum_{i=1}^{n} \tau_{ik}^{[q]} \boldsymbol{x}_i \boldsymbol{x}_i^T$ to avoid possible numerical instability. In such a case, instead of maximizing the weighted Lasso in (4.21) one will maximize a simple weighted

Lasso form

$$-\frac{1}{8}\sum_{i=1}^{n}\tau_{ik}^{[q]}(\hat{c}_{ikr}-\beta_{kr0}-\boldsymbol{x}_i^T\boldsymbol{\beta}_{kr})^2+\hat{E}(\tilde{\boldsymbol{\beta}}_k)-\lambda_{kr}\|\boldsymbol{\beta}_{kr}\|_1, \tag{4.22}$$

where

$$\hat{c}_{ikr}=\tilde{\beta}_{kr0}+\boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}_{kr}+4(u_{ir}-\alpha_{kr}(\tilde{\boldsymbol{\beta}}_k;\boldsymbol{x}_i)),$$

and $\hat{E}(\tilde{\boldsymbol{\beta}}_k)$ is a function of $\tilde{\boldsymbol{\beta}}_k$.

4.3.4 Algorithm tuning and model selection

The appropriate values of the tuning parameters (λ,γ) and the number of clusters K should be chosen carefully in practice. Here, we suggest to use a modified BIC with a grid search scheme introduced by Städler et al. (2010b) for this difficult task. This strategy was mentioned in Section 3.3.4. However, it is worth to note that choosing optimal values of the tuning parameters and the number of components for penalized MoE models is still an open research. Though, the BIC performs quite well for our simulation study.

4.4 Experimental study

The performance of these methods is studied on both simulated data and real data. The results of these algorithms are compared to the standard non-penalized MoE for the Poisson model (denoted by PMoE) and MoE with ℓ_2 regularization (LMoE+ℓ_2) for the logistic model since local maximum parameters that closed to the true value for the MoE of logistic model cannot found in some cases. Several evaluation criteria are used to assess the performance of the models, including sparsity, parameters estimation and clustering criteria. Our penalized models are denoted by PMoE-Lasso and LMoE-Lasso for the Poisson and logistic cases, respectively.

The R packages of codes of the developed algorithms and the documentation are publicly available on this link[1].

4.4.1 Evaluation criteria

To evaluate our methods, the results of all the models are compared based on three different criteria which were mentioned in Section 3.4.1. These criteria are: sensitivity/specificity, parameters estimation, and clustering performance. To deal with the label switching before calculating these criteria, we permuted the estimated coefficients based on an ordered between the expert parameters then replace the kth gating network vector with $\mathbf{w}_k^{per}=\mathbf{w}_k-\mathbf{w}_K$. Specially, for $K=2$, the log-likelihood function and the penalized log-likelihood function will not change since we have $\mathbf{w}_1^{per}=-\mathbf{w}_1$.

[1] `https://github.com/fchamroukhi/prEMME`

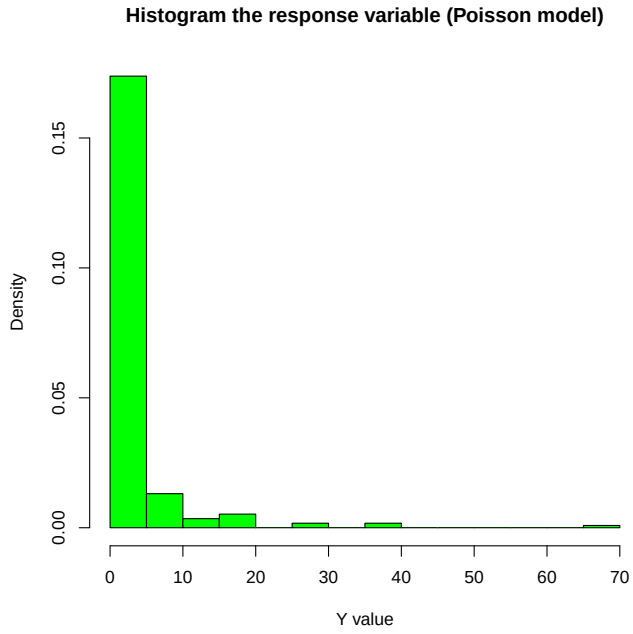


Figure 4.1: Histogram of the response variable Y of the Poisson model.

4.4.2 Simulation study

For this simulation, we generate $n = 300$ predictors $\boldsymbol{x}$ from a multivariate Gaussian distribution with zero mean and correlation structure given by $\text{corr}(x_{ij}, x_{ij'}) = 0.5^{|j-j'|}$. Meanwhile, the response $Y|\boldsymbol{x}$ is generated from a logistic model with two classes and a Poisson model of $K = 2$ expert components with the following regression coefficients:

• Simulation parameters for the Poisson model:

$$(\beta_{10}, \boldsymbol{\beta}_1)^T = (0, 1, 0, -2, 0, 1.5, 0)^T;$$
$$(\beta_{20}, \boldsymbol{\beta}_2)^T = (0, 0, 2, 0, -1, 0, 0)^T;$$
$$(w_{10}, \boldsymbol{w}_1)^T = (1, 0, 0, 1, 0, -1.5, 0)^T.$$

• Simulation parameters for the multinomial-logistic model ($R = 2$):

$$(\beta_{110}, \boldsymbol{\beta}_{11})^T = (0, -1, 2, 0, 0, 1.5, 0)^T;$$
$$(\beta_{210}, \boldsymbol{\beta}_{21})^T = (0, 1, 0, 0, -2, 0, 0)^T;$$
$$(w_{10}, \boldsymbol{w}_1)^T = (1, 0, 0, 1, 0, 0, -1.5)^T.$$

100 data sets were generated for each simulation. The results will be presented in the following sections. Histogram of the response variable for a typical simulated data in Poisson MoE model is presented in Figure 4.1.

Sensitivity/specificity criteria

Table 4.1 presents the sensitivity (S_1), specificity (S_2) values for the experts 1 and 2, and the gates for each of the considered models. The non-penalized MoE models cannot be considered as model selection methods since their sensitivity almost surely equals zero, hence the results for these models are not provided. Especially, the estimated parameters for the logistic model with the standard MoE becomes challenging and unstable. For a typical data set, local maximum parameters that closed to the true value for the MoE of logistic model cannot found (see Table 4.2). Actually, in case of logistic regression Rosset et al. (2004) showed that if the data are separable by the predictors the maximum likelihood solutions are undefined. In such cases, by adding a small amount of quadratic penalization, Park and Hastie (2007a) let the coefficients converge to the ℓ_2 penalized logistic regression solutions instead of infinity as the Lasso tuning parameter λ approaches zero. Based on these ideas, we add two small quadratic penalizations for the MoE model with logistic outputs

$$\frac{\rho}{2}\sum_{k=1}^{K}\sum_{r=1}^{R-1}\|\boldsymbol{\beta}_{kr}\|_2^2 \text{ and } \frac{\rho}{2}\sum_{k=1}^{K-1}\|\boldsymbol{w}_k\|_2^2$$

where $\rho = 0.1\log(n)$ as in Chapter 3. Hence, the MoE with Lasso regularization model in the logistic case is compared with the ℓ_2 regularized MoE model. Here, the Lasso performs quite well for detecting non-zero coefficients both in the experts and in the gating network. By adding the penalty term, one can avoid the instability of the estimators. In the case with high correlation between features, one can consider adding ℓ_2 penalties for the experts and the gating network.

Model	Expert 1		Expert 2		Gate	
	S_1	S_2	S_1	S_2	S_1	S_2
PMoE-Lasso	0.717	1.000	0.818	1.000	0.835	1.000
LMoE-Lasso	0.693	0.960	0.835	0.805	0.780	0.980

Table 4.1: Sensitivity (S_1) and specificity (S_2) results of the Lasso regularized MoE models for Poisson outputs (PMoE-Lasso) and logistic ouputs (LMoE-Lasso).

True value			LMoE-Lasso			MoE method		
Exp. 1	Exp. 2	Gate	Exp. 1	Exp. 2	Gate	Exp. 1	Exp. 2	Gate
0	0	1	-0.1184	-0.1470	0.5604	-2.5467	**49.4886**	0.4417
-1	1	0	-0.6242	0	0	-1.8442	**31.0822**	-0.0505
2	0	0	1.3393	0	0.0411	3.7090	**-30.1612**	-0.0523
0	0	1	0	0	0.7802	-0.3482	**48.1645**	0.3263
0	-2	0	0	-1.5576	0	0.9839	**-66.4277**	0.6738
1.5	0	0	1.2773	0	-0.1194	2.7540	**-9.4606**	-0.7398
0	0	-1.5	0.2138	0	-0.9343	-0.5401	**-6.1314**	-0.7966

Table 4.2: Estimated parameters for a typical data set obtained with Lasso regularized MoE (LMoE-Lasso) and MLE for MoE with logistic observations.

Parameter estimation

The boxplots of all estimated parameters are shown in Figures 4.2 and 4.3. The boxplots are not provided for standard logistic model since the estimating parameter for this model is unstable in this case. It turns out that the PMoE and LMoE+ℓ_2 could not be considered as model selection methods. The Lasso provides sparse results for the model, both in the experts and in the gates. These Lasso models work quite well in detecting non-zero coefficients. However, in the logistic case, this becomes more challenging in the experts and in the gating network.

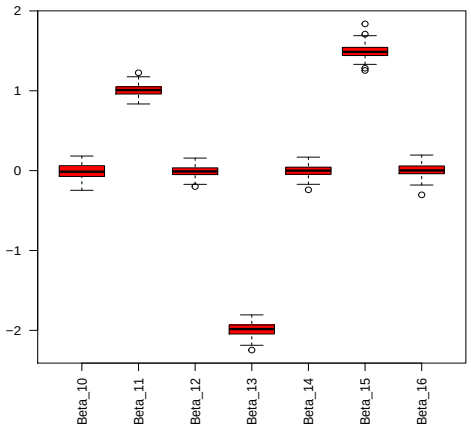
For the mean and standard derivation shown in Table 4.3 and Table 4.4. Notice that, in case of Poisson outputs the model using standard MoE give better results than PMoE-Lasso. This is because the regularized model can cause bias to the estimated parameters since the penalty functions are added to the log-likelihood function. On the other hand, the PMoE-Lasso provides better results than PMoE for estimating the zero coefficients in term of average mean squared error.

For the logistic model, it turns out that by adding two small quadratic penalty functions one can avoid the instability of estimation the MLE parameters. The LMoE+ℓ_2 has better results than the LMoE-Lasso for estimating the non-zero coefficients. Conversely, as in the Poisson model the Lasso is good at detecting zero coefficients and provides small MSE than LMoE+ℓ_2 at the zero parameters.

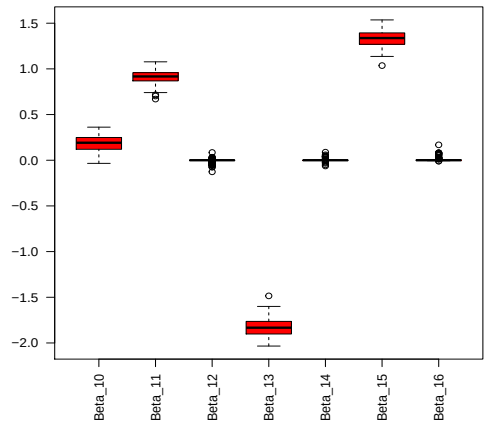
Clustering

The accuracy of clustering for all these mentioned models are calculated for each data set. The results in terms of ARI and correct classification rate values are provided in Table 4.5. For the Poisson case, the PMoE-Lasso model provides a better result than the PMoE model in accuracy of clustering term. However, the difference between these model is smaller than 1%. For the logistic case, it can be seen that, with LMoE-Lasso model we can obtain better results than LMoE+ℓ_2 model in this simulation.

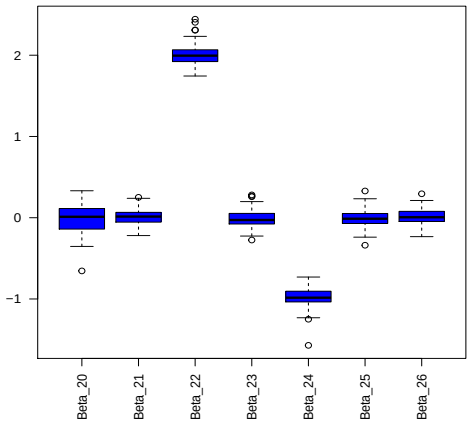
In summary, it is clear that the regularized methods perform quite well in retrieving the actual sparse support; the sensitivity and specificity results are quite reasonable for the proposed models. Although the penalty function will cause bias to the parameters, as shown in the results of the MSE, the algorithm can perform parameter density estimation with an acceptable loss of information due to the bias induced by the regularization. In terms of clustering, the PMoE-Lasso works as well as PMoE models for the Poisson model. For logistic model, the LMoE-Lasso is successful in retrieving the actual parameters used for the model, while the non regularized method failed in this task. Hence, adding quadratic penalty functions in this case is necessary. A comparison between the LMoE-Lasso and the LMoE+ℓ_2 model shows that Lasso provides better results not only in terms of sparsity but also in terms of clustering.



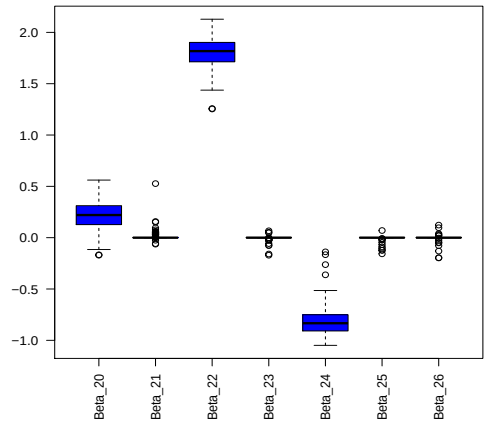
PMoE-Exp.1



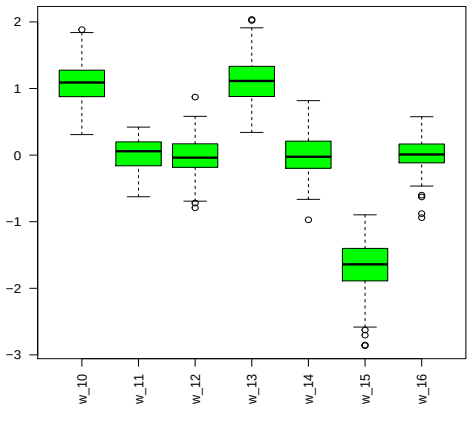
PMoE-Lasso-Exp.1



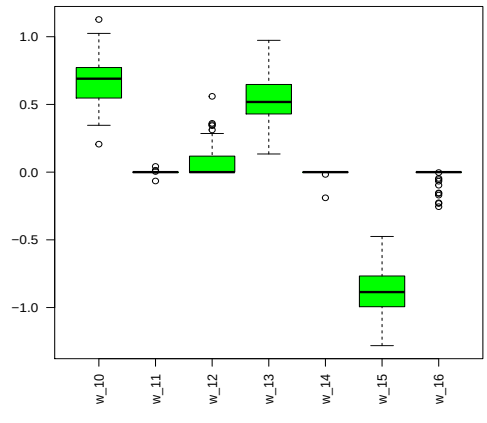
PMoE-Exp.2



PMoE-Lasso-Exp.2

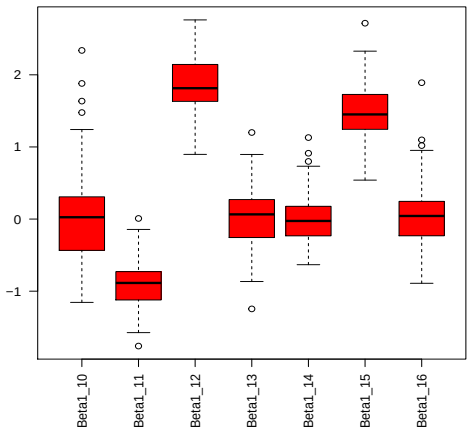


PMoE-Gate

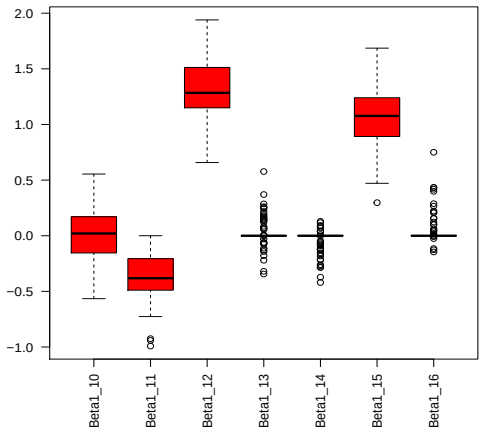


PMoE-Lasso-Gate

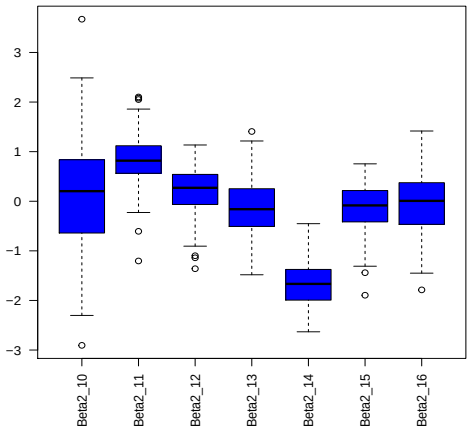
Figure 4.2: Boxplots of non-penalized MoE model (PMoE) and the proposed Lasso penalized MoE model (PMoE-Lasso) for Poisson case.



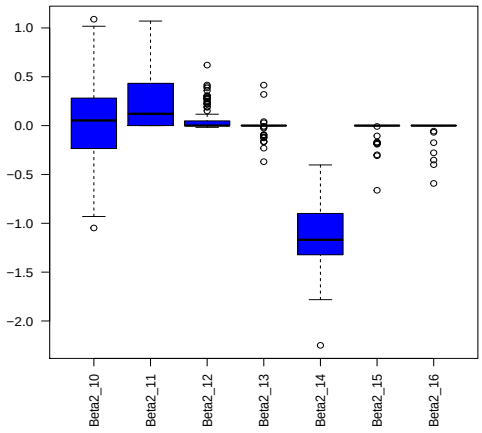
LMoE+ℓ_2-Exp.1



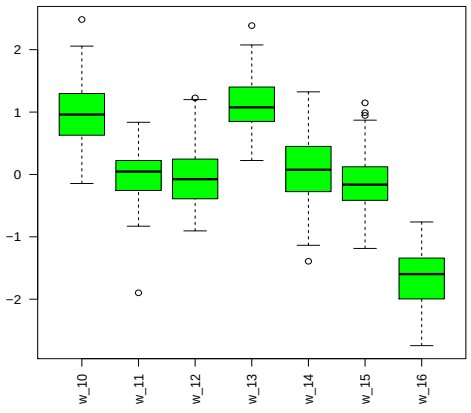
LMoE-Lasso-Exp.1



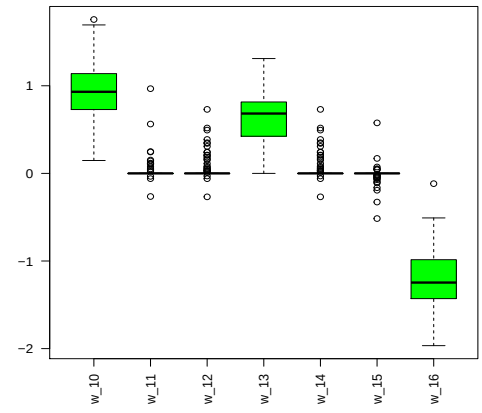
LMoE+ℓ_2-Exp.2



LMoE-Lasso-Exp.2



LMoE+ℓ_2-Gate



LMoE-Lasso-Gate

Figure 4.3: Boxplots of ℓ_2 regularized MoE model (LMoE+ℓ_2) and the proposed Lasso penalized MoE model (LMoE-Lasso) for logistic case.

Comp.	True value	Mean		Mean squared error	
		PMoE	PMoE-Lasso	PMoE	PMoE-Lasso
Exp.1	0	$-0.008_{(.094)}$	$0.190_{(.092)}$	$0.0089_{(.011)}$	$0.0445_{(.036)}$
	1	$1.006_{(.076)}$	$0.905_{(.077)}$	$0.0059_{(.009)}$	$0.0150_{(.021)}$
	0	$-0.009_{(.067)}$	$-0.006_{(.024)}$	$0.0046_{(.007)}$	$0.0006_{(.002)}$
	-2	$-1.989_{(.088)}$	$-1.825_{(.100)}$	$0.0079_{(.011)}$	$0.0407_{(.043)}$
	0	$-0.004_{(.067)}$	$0.003_{(.017)}$	$0.0045_{(.008)}$	$0.0003_{(.001)}$
	1.5	$1.492_{(.089)}$	$1.325_{(.089)}$	$0.0080_{(.015)}$	$0.0386_{(.037)}$
	0	$0.004_{(.077)}$	$0.012_{(.027)}$	$0.0059_{(.011)}$	$0.0009_{(.003)}$
Exp.2	0	$-0.014_{(.178)}$	$0.218_{(.138)}$	$0.0317_{(.051)}$	$0.0669_{(.062)}$
	0	$0.004_{(.091)}$	$0.015_{(.059)}$	$0.0082_{(.012)}$	$0.0037_{(.028)}$
	2	$2.002_{(.130)}$	$1.796_{(.149)}$	$0.0169_{(.030)}$	$0.0638_{(.093)}$
	0	$-0.013_{(.107)}$	$-0.005_{(.028)}$	$0.0117_{(.017)}$	$0.0008_{(.004)}$
	-1	$-0.984_{(.118)}$	$-0.808_{(.157)}$	$0.0142_{(.035)}$	$0.0614_{(.120)}$
	0	$-0.008_{(.111)}$	$-0.007_{(.029)}$	$0.0123_{(.020)}$	$0.0009_{(.003)}$
	0	$0.013_{(.093)}$	$-0.004_{(.036)}$	$0.0089_{(.014)}$	$0.0013_{(.006)}$
Gate	1	$1.092_{(.301)}$	$0.673_{(.174)}$	$0.0992_{(.154)}$	$0.1371_{(.121)}$
	0	$0.011_{(.252)}$	$0.000_{(.008)}$	$0.0636_{(.078)}$	$0.0001_{(.000)}$
	0	$-0.025_{(.282)}$	$0.071_{(.106)}$	$0.0804_{(.132)}$	$0.0163_{(.040)}$
	1	$1.136_{(.336)}$	$0.528_{(.165)}$	$0.1312_{(.201)}$	$0.2496_{(.156)}$
	0	$-0.001_{(.314)}$	$-0.002_{(.019)}$	$0.0986_{(.147)}$	$0.0004_{(.004)}$
	-1.5	$-1.699_{(.415)}$	$-0.885_{(.173)}$	$0.2121_{(.355)}$	$0.4079_{(.217)}$
	0	$-0.002_{(.265)}$	$-0.015_{(.049)}$	$0.0703_{(.135)}$	$0.0027_{(.011)}$

Table 4.3: Estimated parameter vector obtained by PMoE and our PMoE-Lasso for Poisson outputs.

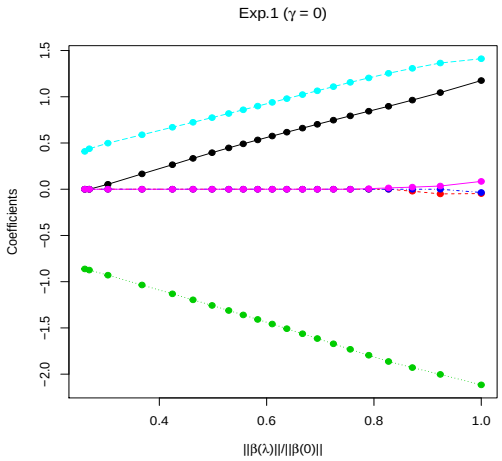
4.4.3 Lasso paths for the regularized MoE parameters

In this part, we provide the Lasso paths for the experts network with Poisson and logistic outputs based on typical simulation data sets. To do this two fix values for the tuning parameters γ are chosen, and a grid of values for λ is also provided.

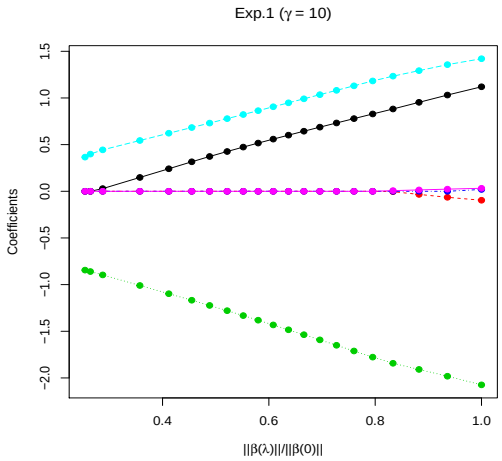
Lasso paths for the case of Poisson outputs

For the regularized MoE with Poisson regression, two fix values for γ are chosen 0 and 10; $\lambda \in \{0, 5, 10, \ldots, 100\}$. The Figure 4.4 gives the Lasso paths for the experts network and also the changing of the gating network coefficients with different values of λ.

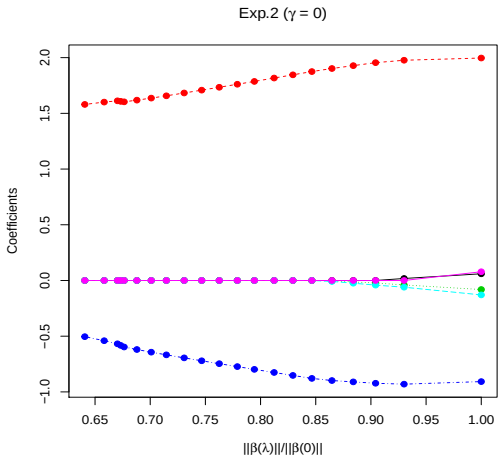
It turns out that the Lasso paths for each expert component perform quite consistent between different values of γ. The performance of the gating network coefficients with difference values of λ is quite similar between $\gamma = 0$ and $\gamma = 10$. The difference lies in the fact that with $\gamma = 10$ some parameters which are close to 0 are set to be 0. However, the changing of λ also effect to the sparsity of the gating network similar with the Gaussian model in Chapter 3. This behavior can be found in the case $\gamma = 10$, since $\lambda \geq 65$ the third coefficient of the gating network is difference from it true value 0.



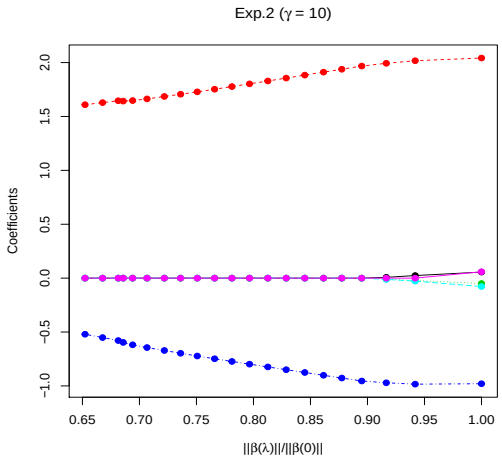
Lasso path for Exp.1 with $\gamma = 0$



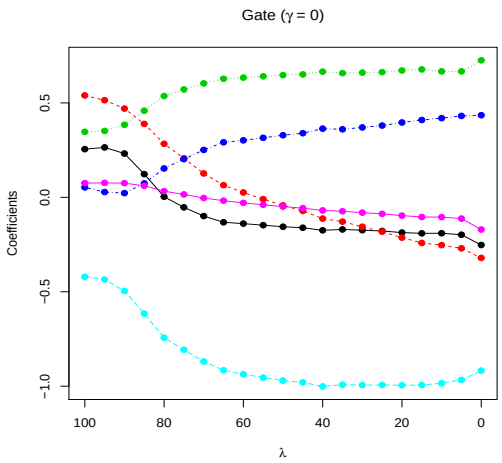
Lasso path for Exp.1 with $\gamma = 10$



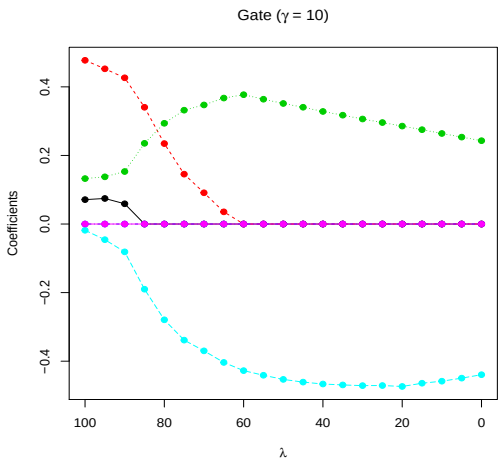
Lasso path for Exp.2 with $\gamma = 0$



Lasso path for Exp.2 with $\gamma = 10$



Changing of the gating network with $\gamma = 0$



Changing of the gating network with $\gamma = 10$

Figure 4.4: Lasso paths for the experts network and the changing of the gating network for regularized MoE model (PMoE-Lasso) with Poisson outputs.

Comp.	True value	Mean		Mean squared error	
		LMoE+ℓ_2	LMoE-Lasso	LMoE+ℓ_2	LMoE-Lasso
Exp.1	0	$0.037_{(.627)}$	$0.008_{(.250)}$	$0.3956_{(.758)}$	$0.0623_{(.079)}$
	-1	$-0.897_{(.317)}$	$-0.370_{(.229)}$	$0.1110_{(.172)}$	$0.4494_{(.287)}$
	2	$1.855_{(.374)}$	$1.315_{(.266)}$	$0.1611_{(.215)}$	$0.5403_{(.376)}$
	0	$0.032_{(.411)}$	$0.020_{(.116)}$	$0.1702_{(.267)}$	$0.0138_{(.041)}$
	0	$-0.004_{(.340)}$	$-0.031_{(.092)}$	$0.1156_{(.188)}$	$0.0094_{(.027)}$
	1.5	$1.496_{(.392)}$	$1.057_{(.249)}$	$0.1539_{(.230)}$	$0.2587_{(.250)}$
	0	$0.042_{(.449)}$	$0.041_{(.124)}$	$0.2031_{(.423)}$	$0.0171_{(.066)}$
Exp.2	0	$0.098_{(1.151)}$	$0.029_{(.402)}$	$1.3347_{(2.054)}$	$0.1624_{(.242)}$
	1	$0.824_{(.532)}$	$0.228_{(.271)}$	$0.3138_{(.612)}$	$0.6687_{(.347)}$
	0	$0.186_{(.484)}$	$0.068_{(.129)}$	$0.2687_{(.336)}$	$0.0213_{(.053)}$
	0	$-0.147_{(.542)}$	$-0.010_{(.078)}$	$0.3152_{(.421)}$	$0.0062_{(.025)}$
	-2	$-1.657_{(.469)}$	$-1.126_{(.324)}$	$0.3376_{(.457)}$	$0.8690_{(.575)}$
	0	$-0.143_{(.501)}$	$-0.023_{(.086)}$	$0.2717_{(.498)}$	$0.0079_{(.046)}$
	0	$-0.037_{(.643)}$	$-0.019_{(.084)}$	$0.4148_{(.541)}$	$0.0075_{(.041)}$
Gate	1	$0.979_{(.467)}$	$0.934_{(.289)}$	$0.2189_{(.337)}$	$0.0881_{(.128)}$
	0	$-0.001_{(.388)}$	$0.025_{(.122)}$	$0.1503_{(.376)}$	$0.0154_{(.098)}$
	0	$-0.055_{(.467)}$	$0.046_{(.131)}$	$0.2212_{(.305)}$	$0.0193_{(.068)}$
	1	$1.121_{(.422)}$	$0.628_{(.293)}$	$0.1924_{(.293)}$	$0.2236_{(.255)}$
	0	$0.100_{(.538)}$	$0.046_{(.131)}$	$0.2993_{(.423)}$	$0.0193_{(.068)}$
	0	$-0.114_{(.474)}$	$-0.008_{(.092)}$	$0.2376_{(.308)}$	$0.0085_{(.043)}$
	-1.5	$-1.654_{(.462)}$	$-1.230_{(.358)}$	$0.2372_{(.335)}$	$0.2014_{(.272)}$

Table 4.4: Estimated parameter vector obtained by LMoE+ℓ_2 and the proposed LMoE-Lasso models for logistic outputs.

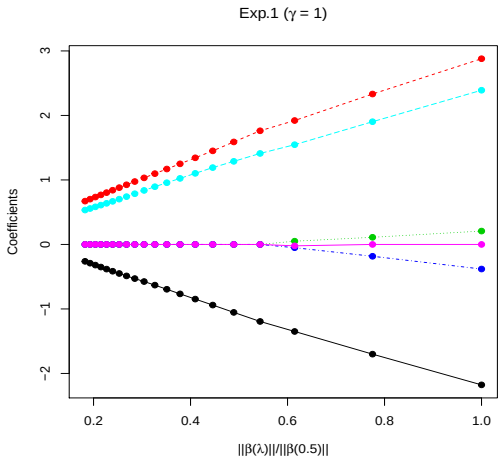
Model	C.rate	ARI
PMoE	$88.85\%_{(2.04\%)}$	$0.5965_{(.063)}$
PMoE-Lasso	$88.96\%_{(2.03\%)}$	$0.6004_{(.063)}$
LMoE+ℓ_2	$81.02\%_{(4.00\%)}$	$0.3813_{(.092)}$
LMoE-Lasso	$82.06\%_{(2.93\%)}$	$0.3985_{(.078)}$

Table 4.5: Average of the accuracy of clustering (correct classification rate and Adjusted Rand Index) results for the Poisson outputs with the non-penalized model (PMoE) and the proposed Lasso regularized model (PMoE-Lasso). The results are also considered for the logistic case with ℓ_2 penalization (LMoE+ℓ_2) and Lasso regularization (LMoE-Lasso).

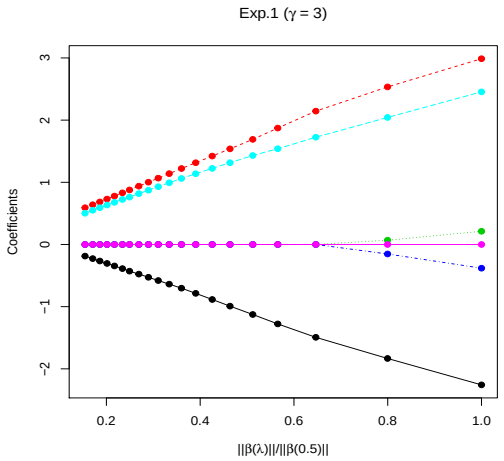
Lasso paths for the case of logistic outputs

Finally, for the regularized MoE with logistic regression, we choose two non-zero values for γ, $\gamma \in \{1, 3\}$ and a grid of 20 different values for λ, $\lambda \in \{0.5, 1.0, 1.5, \ldots, 10\}$. Figure 4.5 gives the Lasso paths for the experts network and also the changing of the gating network coefficients with different values of λ.

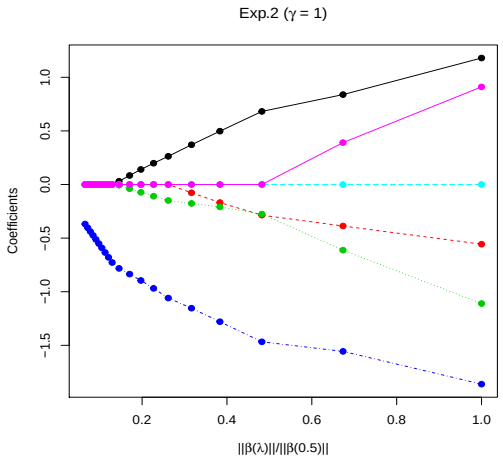
Here, the Lasso paths for each expert part performs are almost identical with different values of γ. However, as λ decreases to zero, some coefficients of the experts network can grow to infinity. We therefore add two small quadratic penalizations for the regularized model with



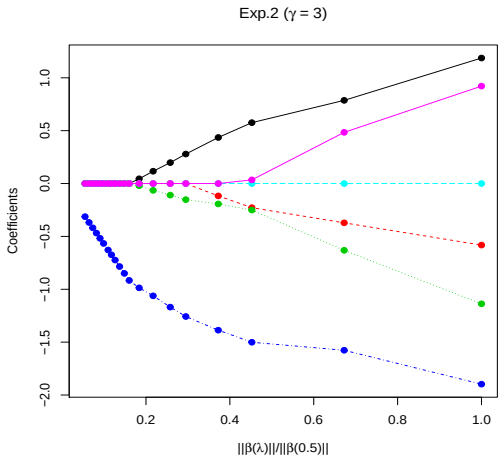
Lasso path for Exp.1 with $\gamma = 1$



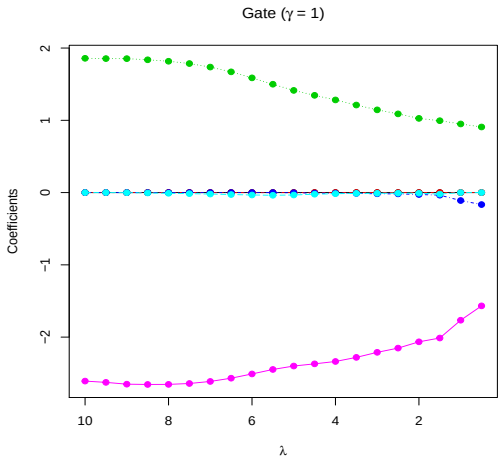
Lasso path for Exp.1 with $\gamma = 3$



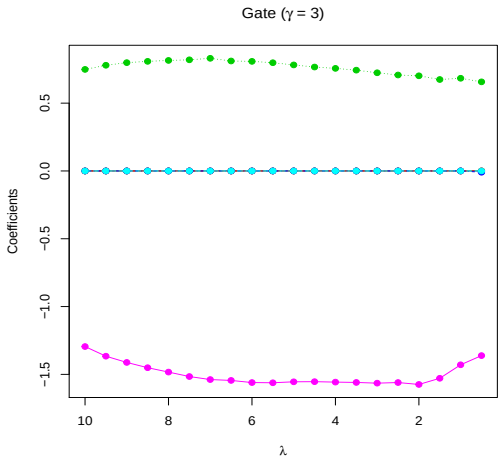
Lasso path for Exp.2 with $\gamma = 1$



Lasso path for Exp.2 with $\gamma = 3$



Changing of the gating network with $\gamma = 1$



Changing of the gating network with $\gamma = 3$

Figure 4.5: Lasso paths for the experts network and the changing of the gating network for regularized MoE model (LMoE-Lasso) with logistic outputs.

logistic outputs

$$\frac{\rho}{2}\sum_{k=1}^{K}\sum_{r=1}^{R-1}\|\boldsymbol{\beta}_{kr}\|_2^2 \text{ and } \frac{\rho}{2}\sum_{k=1}^{K-1}\|\boldsymbol{w}_k\|_2^2$$

where $\rho = 0.1\log(n)$ and consider addition elastic net paths in this case. This regularized model is denoted by LMoE-ElasticNet. Hence, the update of the gating network is done by using a similar proximal Newton-type procedure in Chapter 3. For the experts network, the equation (4.21) is replaced by

$$\tilde{I}_r(\boldsymbol{\beta}_k) - \lambda_{kr}\|\boldsymbol{\beta}_{kr}\|_1 - \frac{\rho}{2}\|\boldsymbol{\beta}_{kr}\|_2^2. \tag{4.23}$$

to update the coefficients. The elastic net paths are given in Figure 4.6 for $\gamma \in \{0, 3\}$, $\lambda \in \{0, 0.5, \ldots, 10\}$ and $\rho = 0.1\log(n)$.

It is interesting to see that the elastic net paths in this situation have similar shapes with the Lasso, except the ℓ_2 penalized solution is obtained instead of infinity as λ approaches zero. Thus, the instability of the solution can avoid. For $\lambda > 1$, the effect of the quadratic penalty with a small ρ is not appreciable. In application, depend on the data such as the correlations between features and the value of tuning parameters λ, γ, one can consider adding two small amount of quadratic penalizations or not. Otherwise, the model works quite well with just ℓ_1 penalizations.

Overall, the performance of Lasso paths for the experts network in both models seem to be consistence with different values of γ. Some open questions should be study more such as the effect of λ to the sparsity of the gating network.

4.4.4 Applications to real data sets

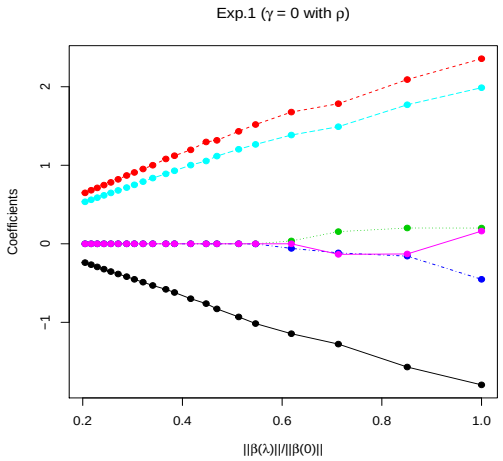
In this part, three real data sets are analyzed as a further test of the proposal methodology. Two data sets are for the logistic model and one for Poisson model. The obtain results are compared with other methods, which provided by Peralta and Soto (2014). The comparison are based upon three different criteria: the average mean squared error (MSE) between observation values and the predicted values of the response variable, the sparsity of each result, and the correlation of these values. After the parameters are estimated and the data are clustered, the following value under the estimated model

$$\hat{Y} = \text{mode } p_k(y|\boldsymbol{x}; \hat{z} = k) = \text{mode } p_k(y|\boldsymbol{x}; \hat{\boldsymbol{\theta}}_k),$$

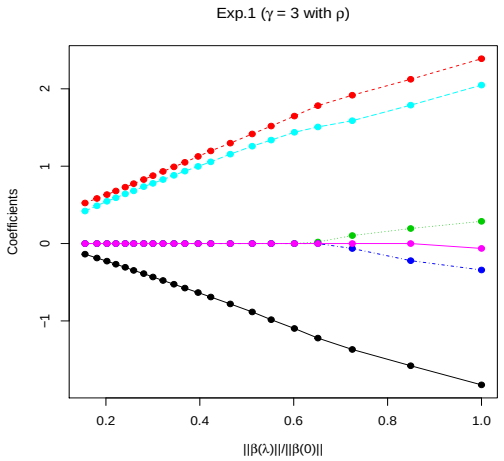
is used as a predicted value for Y.

Regularized MoE model with Poisson outputs

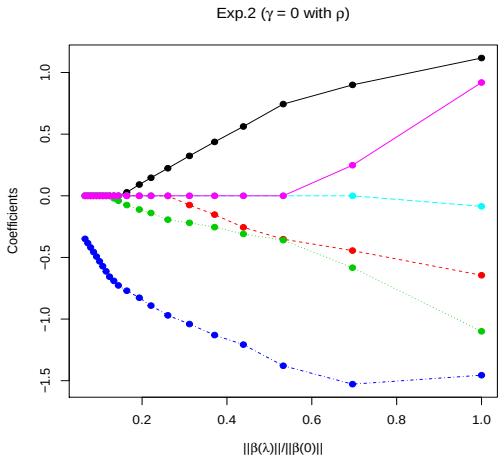
A data set is used here to illustrate for the proposed regularized MoE of Poisson regression experts. The study used Cleveland Clinic Foundation heart disease data set that available at the website UC Irvine Machine Learning Repository. This data set includes 13 features and 297



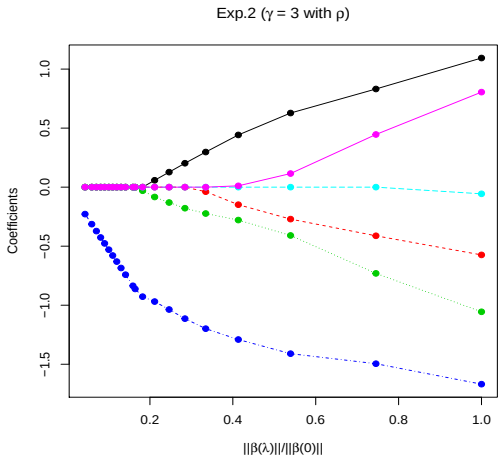
Lasso path for Exp.1 with $\gamma = 0$



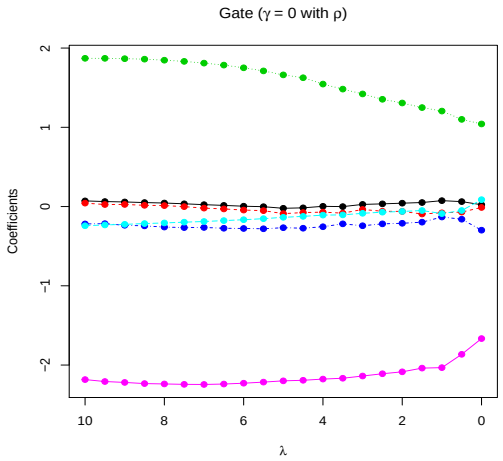
Lasso path for Exp.1 with $\gamma = 3$



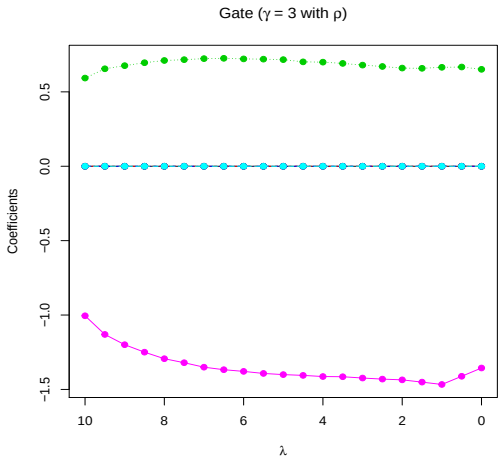
Lasso path for Exp.2 with $\gamma = 0$



Lasso path for Exp.2 with $\gamma = 3$



Changing of the gating network with $\gamma = 0$



Changing of the gating network with $\gamma = 3$

Figure 4.6: Elastic net paths for the experts network and the changing of the gating network for regularized MoE model (LMoE-ElasticNet) with logistic outputs.

Feature	Exp.1	Exp.2	Gate
x_0	0.51211	-1.38996	-0.71073
x_1	-	-	-
x_2	-	-	0.54763
x_3	0.06753	-	0.54110
x_4	0.00959	0.09146	-
x_5	-	-	-
x_6	-	-	-
x_7	0.07229	-	0.10834
x_8	-	-	-0.62335
x_9	-	0.50573	-
x_{10}	0.05960	0.33149	0.03440
x_{11}	0.11976	0.01285	-
x_{12}	0.05649	-	1.54824
x_{13}	0.04244	0.46287	0.64450

Table 4.6: Fitted regularized MoE model (PMoE-Lasso) to the heart disease data.

observations. 160 observations among them have zero response value. Generally, an appropriate approach for this type of data is to use the zero inflated Poisson regression model (ZIP model). However, the regularized MoE of the Poisson regression is tested and observed on its behavior with this type of data. Taking $K = 2$ and focusing on the regularized MoE for Poisson regression, the model's estimated parameters are provided in Table 4.6. There are two components, the first one has 108 objects and the second one has 189 objects. The second class contains 156 over 160 observations that have zero response value. In this case, it looks like the data is splitted into two parts, with one part contains mainly zero response value similar with the approach of ZIP. In term of prediction, 65% of observations have the same values between their predictions and their response values. It is worth to consider the regularized MoE for ZIP model as an extended approach for this type of data. The changing of the penalized log-likelihood after each iteration is given in Figure 4.7.

Histogram of the variable Y and its predicted value for this data set is given in Figure 4.8.

Regularized MoE model with Multinomial outputs

For the logistic case, we consider the two data sets that were used by Peralta and Soto (2014) in their work and compare the results between our approach with their method. We investigate the Ionosphere data and Musk-1 data which are described on the website UC Irvine Machine Learning Repository. The Ionosphere data contains 351 observations and 33 features. The Musk-1 data has 486 observations and 168 features. The variables with zero variance are removed. Hence, the Musk-1 data set remains with 167 features. Both data sets have two classes. All features are standardized to have mean zero and unit variance. $K = 2$ is taken as in Peralta and Soto (2014).

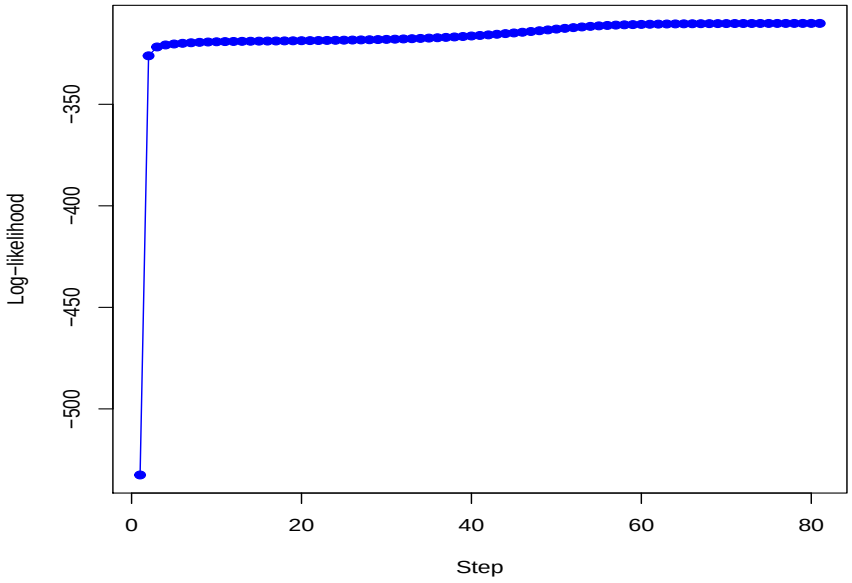
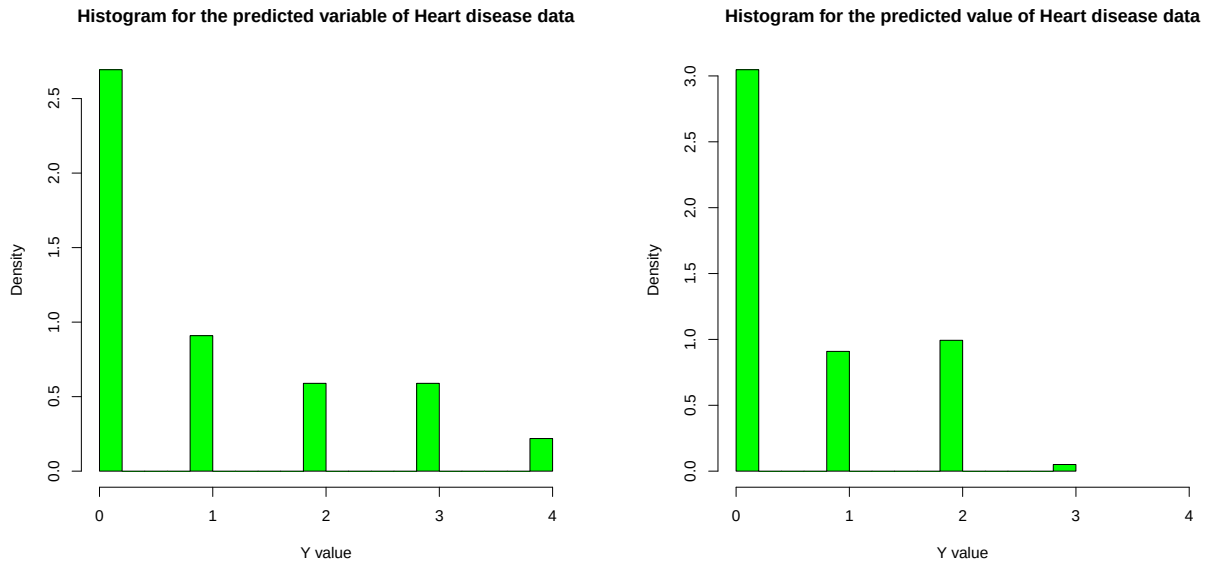


Figure 4.7: The penalized log-likelihood during the EM iterations when fitting the PMoE-Lasso model to the Cleveland Clinic Foundation heart disease data.



Histogram of Y Histogram of $\hat{Y}$

Figure 4.8: Histograms of the variable Y and its predicted value for the Cleveland Clinic Foundation heart disease data.

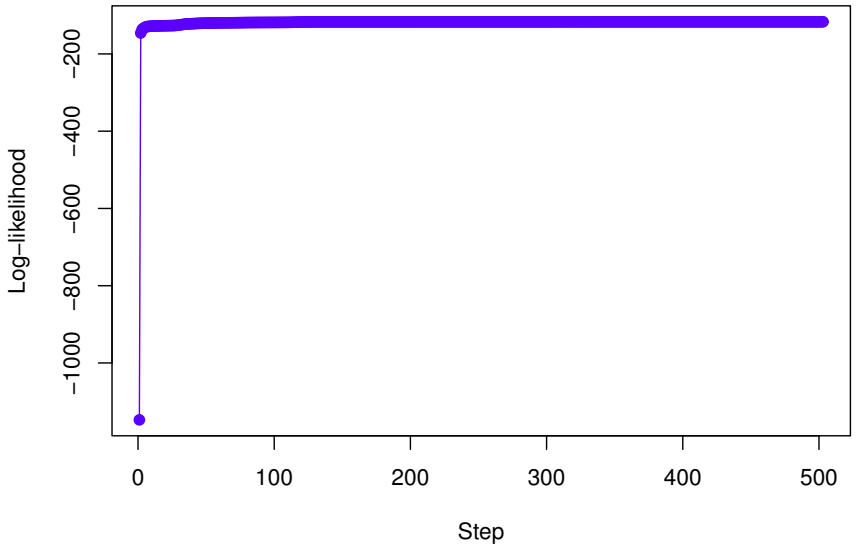


Figure 4.9: The penalized log-likelihood during the EM iterations when fitting the LMoE-Lasso model to the Ionosphere data.

The parameter estimates of the MoE models obtained by Lasso are given in Table 4.7 and Table 4.8, 4.9. The classification accuracy and percentage of features reduction results between the proposal with Peralta's work are found in Table 4.10. These results suggest that the proposed algorithm with Lasso provide better results than the remain method in term of data classification and features reduction. For Ionosphere dataset, Peralta used on average 78.1% of all dimensions while our approach just need 26.3%. For the Musk-1 dataset, the proposed Lasso method also increases the ratio of dimensionality reduction up to 10%. Consider the classification rate, on both data sets the proposal method increases this ratio up to 12% since comparing with Peralta's. One of the reasons for this improvement is that the approach of Peralta does not guarantee the increase of the penalized log-likelihood values after each loop of their EM algorithm. The changing of the penalized log-likelihood functions on these data set can be referred from Figure 4.9 and Figure 4.10.

Dataset name	Classification accuracy		Dimensionality reduction	
	Lasso (Peralta)	LMoE-Lasso	Lasso (Peralta)	LMoE-Lasso
Ionosphere	84.1%	96.6%	21.9%	73.7%
Musk-1	80.0%	93.3%	79.6%	90.0%

Table 4.10: Classification accuracy and percentage of features reduction results compared between Peralta's method (Peralta and Soto, 2014) and our proposed method (LMoE-Lasso).

Feature	Exp.1	Exp.2	Gate
x_0	-1.64671	-1.25999	0.34349
x_1	-1.04171	-0.79945	-
x_2	-0.94925	-0.64691	-
x_3	-	-	-
x_4	-	-1.81555	0.94631
x_5	-0.05046	-0.20732	-
x_6	-0.45212	-0.27119	-
x_7	-0.85935	-0.18387	-
x_8	-0.04429	-	-
x_9	-0.75204	-	-0.28020
x_{10}	-	-	-
x_{11}	-	-	-
x_{12}	-	-	-
x_{13}	-	-	-
x_{14}	-	-	-
x_{15}	-	-0.15926	-
x_{16}	-	-	-
x_{17}	-	-0.29576	-
x_{18}	-	-	-
x_{19}	-	-	-
x_{20}	-	-	-
x_{21}	0.41903	-	-
x_{22}	-	-	-
x_{23}	-1.36138	1.48880	-1.83610
x_{24}	-0.41763	-	-
x_{25}	-	-	-
x_{26}	-	0.20319	-
x_{27}	-	-	-
x_{28}	-	-	-
x_{29}	-	-0.02892	-
x_{30}	-	-	-
x_{31}	-	-	-
x_{32}	-	-	-
x_{33}	0.99009	-0.21365	-

Table 4.7: Fitted regularized MoE model (LMoE-Lasso) to Ionosphere data.

Feature	Exp.1	Exp.2	Gate	Feature	Exp.1	Exp.2	Gate
x_0	0.06922	0.17778	0.12277	x_{42}	-	-	-
x_1	-	-	-	x_{43}	-	-0.32513	-
x_2	-	-	-	x_{44}	-	-	-
x_3	-	-	-	x_{45}	-	-	-
x_4	-	-	-	x_{46}	-	-	-
x_5	-	-	-	x_{47}	0.10696	0.13833	-
x_6	-	-	-1.15153	x_{48}	-0.70925	-	-
x_7	-	-	-	x_{49}	-	0.05006	-
x_8	-	-	-0.73044	x_{50}	-0.10448	-0.20221	-
x_9	-	-	-	x_{51}	-	-	-
x_{10}	-	-	-	x_{52}	-	-	-
x_{11}	-	-	-	x_{53}	-	-	-
x_{12}	-	-	-	x_{54}	-	-	-
x_{13}	-	-	-	x_{55}	-0.10431	-	-
x_{14}	-	0.35940	-	x_{56}	-0.53456	-	-
x_{15}	-	-	-	x_{57}	-	-	-
x_{16}	-	-	-	x_{58}	-	-	-
x_{17}	-	-	-	x_{59}	-0.07893	-	-
x_{18}	-	-	-	x_{60}	-	-	-
x_{19}	-	-	-	x_{61}	0.00010	-	-
x_{20}	-	-	-	x_{62}	-	-	-
x_{21}	-	-	-	x_{63}	-	-	-
x_{22}	-	-	-	x_{64}	-	-	-
x_{23}	-	-	-	x_{65}	-	-	-
x_{24}	-0.31879	-	-	x_{66}	-	-	-
x_{25}	-	-	-	x_{67}	-	-	-
x_{26}	-	-	-	x_{68}	-	-	-
x_{27}	-	-	-	x_{69}	-	-	-
x_{28}	-	-	-	x_{70}	0.18476	-	-
x_{29}	-	-	-	x_{71}	-	-	-
x_{30}	-	-	-	x_{72}	-	-	-
x_{31}	-	0.56436	-	x_{73}	-	-	-
x_{32}	-	-	-	x_{74}	-	-	-
x_{33}	-	-	-	x_{75}	-	-	-
x_{34}	-	-	-	x_{76}	0.08573	0.45813	-
x_{35}	-	-	-	x_{77}	-	-	-
x_{36}	0.22055	0.31051	-	x_{78}	-	-	-
x_{37}	-	0.41421	-	x_{79}	-	-	-
x_{38}	-	-	-	x_{80}	-	-	-
x_{39}	-	-	-	x_{81}	-	-	-
x_{40}	-	-	-	x_{82}	-	-	-
x_{41}	-	-	-	x_{83}	-0.88481	-	-

Table 4.8: Fitted regularized MoE model (LMoE-Lasso) to Musk-1 data (part 1).

Feature	Exp.1	Exp.2	Gate	Feature	Exp.1	Exp.2	Gate
x_{84}	-0.03139	0.55857	-1.21692	x_{126}	0.36082	-	-
x_{85}	-	-	-	x_{127}	-	-	-
x_{86}	-	-	-	x_{128}	-	-	-
x_{87}	-	-	-	x_{129}	-0.57213	-	-
x_{88}	-	0.20919	-	x_{130}	-	-	-
x_{89}	-	-	-	x_{131}	-	-	-
x_{90}	-	-	-	x_{132}	0.02409	-	-
x_{91}	-	-	-	x_{133}	-	-	-
x_{92}	0.25523	0.03731	-	x_{134}	-	-	-
x_{93}	-	-	-	x_{135}	-	-	-
x_{94}	-	-	-	x_{136}	0.34955	-	-
x_{95}	-	-	-	x_{137}	-	-	-
x_{96}	-	-	-	x_{138}	-	-	-
x_{97}	-	0.36352	-	x_{139}	-	-	-
x_{98}	-	-	-	x_{140}	-	-	-
x_{99}	-	-	-	x_{141}	-0.18019	-	-
x_{100}	-	-	-	x_{142}	-	-	-
x_{101}	-	-	-	x_{143}	-	-	-
x_{102}	0.20188	-	-	x_{144}	-	-	-
x_{103}	-	-	-	x_{145}	-	-	-
x_{104}	-	-	-	x_{146}	-	-	-
x_{105}	-	-	-	x_{147}	0.20336	0.51844	-
x_{106}	-	-	-0.88963	x_{148}	-	-	-
x_{107}	-	-	-	x_{149}	-	-	-
x_{108}	-	-	-	x_{150}	-	-	-
x_{109}	0.13949	-	-	x_{151}	0.56270	-	-
x_{110}	-	-	-	x_{152}	-	-	-
x_{111}	-	-	-	x_{153}	-	-	-
x_{112}	-	-	-	x_{154}	-	-	-
x_{113}	-	-	-	x_{155}	-	-	-
x_{114}	-	-	-	x_{156}	-	-	-
x_{115}	-	-	-	x_{157}	-	0.23666	-
x_{116}	-0.21509	-0.39766	-	x_{158}	-	-	-
x_{117}	-	-	-	x_{159}	-	-	-
x_{118}	-	-	-	x_{160}	-	-	-
x_{119}	-	-	-	x_{161}	-	-	-
x_{120}	-	-	-	x_{162}	0.33300	0.62605	-
x_{121}	-	-	-	x_{163}	-	0.14212	-
x_{122}	-0.28134	-	-	x_{164}	0.28869	-	-
x_{123}	-	-	-	x_{165}	-	-0.66940	-
x_{124}	-	-	-	x_{166}	-	-	-
x_{125}	-	-	-				

Table 4.9: Fitted regularized MoE model (LMoE-Lasso) to Musk-1 data (part 2).

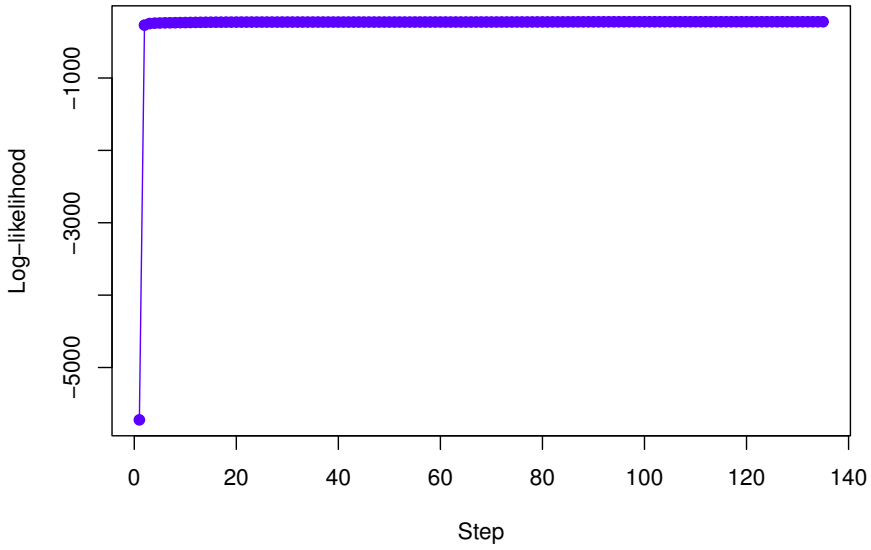


Figure 4.10: The penalized log-likelihood during the EM iterations when fitting the LMoE-Lasso model to the Musk-1 data.

4.5 Discussion for the high-dimensional setting

To evaluate our methods in high-dimensional setting $p > n$, we take 100 random observations of the Musk-1 data set with all 166 features. A MoE model with logistic outputs is used to modeling this data. Parameter estimation results are provided in Table 4.11 and Table 4.12. A computer with CPU Intel i7-5500U 2.40GHz and 8GB RAM needs few ($\sim$ 17) minutes to estimate all the parameters for the model with $\lambda = 1.5$ and $\gamma = 1.5$. The classification accuracy and percentage of features reduction results are given in Table 4.13.

From these tables, it turns out that the proposed algorithms perform quite well in this setting. We still obtain acceptable results for the regularized MoE models and the EM algorithm using the proximal Newton and proximal Newton-type methods are good tool for the parameter estimation. However, comparing with the parameters of the full data set there are big differences. For the full data set, 255 observations are clustered into the Class 1 and 221 observations belong to the Class 2. Meanwhile, in the subset data 89 observations belongs to Class 1 and the remaining is from Class 2. For the affect of each feature to the experts network and gating network, one can easily find that some features such as x_{47}, x_{50}, x_{76}, x_{83}, x_{92}, and x_{147} still keep their roles in the first expert part for both data set. While just only the feature x_{84} appears in the gating network on two parameters estimation sets. Hence, to interpret the model in high-dimensional setting is an interesting question and should need more improvement in theoretical results.

Dataset name	Classification accuracy	Nb. of zero coefficients			Nb. of observations	
		Exp.1	Exp.2	Gate	Class 1	Class 2
Musk-1 (Subset)	100%	145	165	161	89	11

Table 4.13: Classification accuracy and percentage of features reduction results for the subset of Musk-1 data using the proposed method LMoE-Lasso.

4.6 Conclusion and future work

In this chapter, we proposed a regularized MLE for the MoE model which encourages sparsity, and developed EM-based algorithms to monotonically maximize this regularized objective towards at least a local maximum, while they do not require using approximations as in standard MoE regularization. The proposed algorithms are based on proximal Newton-type methods and univariate updates of the model parameters via coordinate ascent, which allows to tackle matrix inversion problems and obtain sparse solutions. The results on the simulated and the real data sets in terms of parameter estimation, the estimation of the actual support of the sparsity, and clustering accuracy, confirm the effectiveness of this proposal, at least for problems with moderate dimension. The model sparsity does not include significant bias in terms of parameter estimation nor in terms of recovering the actual clusters of the heterogeneous data. A proximal Newton-type approach is possible to obtain closed form solutions for an approximate of the M-step as an efficient method that is promoted to deal with high-dimensional data sets.

A future work may consist of investigating more model selection experiments and considering hierarchical MoE of generalized linear models. Our proposed in this chapter can extend for the hierarchical MoE models with Lasso-like regularizations. As it can be seen through out this chapter, the Lasso regularized MoE models perform quite efficient in high-dimensional setting when $p > n$. However, theoretical results should be studied more to explain the behavior of the penalized methods in this context. In a different aspect, directly applying the regularized models for huge-scale data is not a reasonable approach. One should need more dimension reduction first, and our proposals should be consider as the terminal method to modeling these heterogeneous data. Recently work of Celeux et al. (2019) is an open suggestion for dimension reduction. A hybrid method can be an efficient way to deal with these kinds of data.

In a similar spirit, Morvan et al. (2019) proposed a penalized mixture of logistic regression models. Actually, this is a regularized version of the localized MoE models (Xu et al., 1995) with logistic regression components. They suggested to penalize the experts network with the Lasso penalizations for feature selection task and another Lasso penalty function for sparse representation of the inverse component covariance matrices. The difficulty lies in the fact that they must guarantee that the estimate matrices are symmetric positive definite. In addition, the explanatory variable $\boldsymbol{X}$ can lied in a subspace for each mixture component. Hence, a method to interpret the distribution of $\boldsymbol{X}$ should be taken in account.

This chapter has led to the following publication Huynh and Chamroukhi (2019).

Feature	Exp.1	Exp.2	Gate	Feature	Exp.1	Exp.2	Gate
x_0	-0.33669	4.34170	3.17863	x_{42}	-	-	-
x_1	-	-	-	x_{43}	-	-	-
x_2	-	-	-	x_{44}	-	-	-
x_3	-	-	-	x_{45}	-	-	-
x_4	-	-	-	x_{46}	-	-	-
x_5	-	-	-	x_{47}	0.08585	-	-
x_6	-	-	-	x_{48}	-	-	-
x_7	-	-	-	x_{49}	-	-	-
x_8	-	-	-	x_{50}	-0.69865	-	-
x_9	-	-	-	x_{51}	-0.46418	-	-
x_{10}	-	-	-	x_{52}	-	-	-
x_{11}	-	-	-	x_{53}	-	-	-
x_{12}	-	-	-	x_{54}	-	-	-
x_{13}	-	-	-	x_{55}	-	-	-
x_{14}	-	-	-	x_{56}	-	-	-
x_{15}	-	-	-	x_{57}	-	-	-
x_{16}	-	-	-	x_{58}	0.00085	-	-
x_{17}	-	-	0.42661	x_{59}	-	-	-
x_{18}	-	-	-	x_{60}	-	-	-
x_{19}	-	-	-	x_{61}	-	-	-
x_{20}	-	-	-	x_{62}	-	-	-
x_{21}	-0.63760	-	-	x_{63}	-	-	-
x_{22}	-	-	-	x_{64}	-	-	-
x_{23}	-	-	-	x_{65}	-	-	-
x_{24}	-	-	-	x_{66}	-	-	-
x_{25}	-	-	-	x_{67}	0.22870	-	-
x_{26}	-	-	-	x_{68}	-	-	-
x_{27}	-	-	-	x_{69}	-	-	-
x_{28}	-	-	-	x_{70}	-	-	-
x_{29}	-	-	-	x_{71}	-	-	-
x_{30}	-	-	-	x_{72}	-	-	-
x_{31}	-	-	-	x_{73}	-	-	-
x_{32}	-	-	-	x_{74}	-	-	-
x_{33}	-	-	-	x_{75}	-	-	-
x_{34}	-0.70692	-	-	x_{76}	0.49465	-	-
x_{35}	-	-	-	x_{77}	-	-	-
x_{36}	-	-	-	x_{78}	-	-	-
x_{37}	-	-	-1.49579	x_{79}	-	-	-
x_{38}	-	-	-	x_{80}	0.28250	-	-
x_{39}	-	-	-	x_{81}	-	-	-
x_{40}	-	-	-	x_{82}	-	-	-
x_{41}	-	-	-	x_{83}	-0.87708	-	-

Table 4.11: Fitted regularized MoE model (LMoE-Lasso) to the subset of Musk-1 data (part 1).

Feature	Exp.1	Exp.2	Gate	Feature	Exp.1	Exp.2	Gate
x_{84}	-	-	-0.27199	x_{126}	-	-	-
x_{85}	-	-	-	x_{127}	-	-	-
x_{86}	-	-	-	x_{128}	-	-	-
x_{87}	-	-	-	x_{129}	-	-	-
x_{88}	0.07382	-	-	x_{130}	-	-	-
x_{89}	-0.11238	-	-	x_{131}	-	-	-
x_{90}	-	-	-	x_{132}	-	-	-
x_{91}	0.25618	-	-	x_{133}	-	-	-
x_{92}	0.32106	-	-	x_{134}	-	-	-
x_{93}	-	-	-	x_{135}	-	-	-
x_{94}	-0.46848	-	-	x_{136}	-	-	-
x_{95}	-	-	-	x_{137}	-	-	-
x_{96}	0.40657	-	-	x_{138}	-	-	-
x_{97}	-	-	-	x_{139}	-	-	-
x_{98}	-	-	-	x_{140}	-	-	-
x_{99}	-	-	-	x_{141}	-	-	-
x_{100}	-	-	-	x_{142}	-	-	-
x_{101}	-	-	-	x_{143}	-	-	-
x_{102}	-	-	-	x_{144}	-	-	-
x_{103}	-	-	-	x_{145}	-	-0.45413	-0.45352
x_{104}	-	-	-	x_{146}	0.44840	-	-
x_{105}	-	-	-	x_{147}	0.53837	-	-
x_{106}	-	-	-	x_{148}	-	-	-
x_{107}	-	-	-	x_{149}	-	-	-
x_{108}	-	-	-	x_{150}	-	-	-
x_{109}	-	-	-	x_{151}	-	-	-
x_{110}	-	-	-	x_{152}	-	-	-
x_{111}	-	-	-	x_{153}	-	-	-
x_{112}	-	-	-	x_{154}	-	-	-
x_{113}	-	-	-	x_{155}	-	-	-
x_{114}	-	-	-	x_{156}	-	-	-
x_{115}	-	-	-	x_{157}	0.38850	-	-
x_{116}	-	-	-	x_{158}	-	-	-
x_{117}	-	-	-	x_{159}	-	-	-
x_{118}	-	-	-	x_{160}	-	-	-
x_{119}	-	-	-	x_{161}	1.00947	-	-
x_{120}	-	-	-	x_{162}	-	-	-
x_{121}	-	-	-	x_{163}	-	-	-0.16575
x_{122}	-	-	-	x_{164}	-	-	-
x_{123}	-	-	-	x_{165}	-	-	-
x_{124}	-	-	-	x_{166}	-	-	-
x_{125}	1.03320	-	-				

Table 4.12: Fitted regularized MoE model (LMoE-Lasso) to the subset of Musk-1 data (part 2).

Chapter 5

Conclusions and future directions

5.1 Conclusions

This thesis has been written regarding the regularized MoE models for regression data with various types of outputs. The developed approaches are particularly based on the Lasso-like penalty which simultaneously performs parameter estimation and feature selection on the heterogeneous regression data.

In Chapter 3, we first presented the MoE based for modeling heterogeneous regression data with continuous outputs. More specifically, we used MoE with Gaussian regressor experts for these data. Then, a regularized model, which contains two Lasso-like penalty functions, was proposed. These penalty parts concomitantly select the feature from the gating network as well as the experts network. For parameter estimation, three hybrid EM algorithms were introduced. Different from the state-of-art, our proposed methods avoid matrix inversion and also do not require any approximation on the penalty functions. Experiments on simulated data and real data showed the effectiveness of our approach. Furthermore, we believe that the proposed algorithm based on EM and proximal Newton-type procedure is a potential approach to deal with a high-dimensional data set. This method also can handle with the situation in which $p > n$, as discussed in Section 3.4.6.

In Chapter 4, we extended this regularized MoE modeling approach to model regression data with discrete outputs including counting data and classification data. The regularized models are obtained by adding two Lasso penalty functions: one the logistic mixing functions and the other for the regression components. By doing so, one can attain a sparse MoE model which can represent the data better than a classical model. Regularized MLE is done via hybrid EM algorithms with proximal Newton-type method. The algorithms inherit all the best characteristics of the previous one in Chapter 3, i.e, guarantee the monotonicity of the penalized log-likelihood after each iteration and also avoid computing the inverse matrices, and encourage sparsity. Experiments conducted on simulated data and real data demonstrated the relevance of the proposed approaches in terms of feature selection, clustering and prediction as compared to other alternatives.

Finally, the R packages for the proposed algorithms are constructed and can be downloaded via the link in List of Publications and Communications.

5.2 Future directions

Future work may consist of developing an extension of the regularized model for hierarchical MoE of generalized linear models (Jordan and Jacobs, 1994; Jiang and Tanner, 1999a). Here, the architecture is a tree in which the gating networks sit at the nonterminals of the tree and the expert networks sit at the leaves of the tree. Each expert produces an output vector for each input vector. These output vectors proceed up the tree, being blended by the gating network outputs. Hence, a regularized approach should be studied for this framework in preventing overfitting and to have a sparse representation model due to the large size of the input vector. Our proposals in this thesis can be seen as a potential approach for this problem.

In addition, penalized MoE models with multivariate response variables are also an interesting topic and has good theoretical properties (Nguyen et al., 2019a). To the best of our known, there is no research focus on these cases from the regularized estimation perspective.

For the localized MoE models (Xu et al., 1995), it is interesting to study more these models namely with multiple outputs. On the one hand, we should provide a sparse interpret in the experts networks and on the other hand, it also needed to have a sparse structure of the covariance matrices. A suggestion can be given by combining our methods with the work of Fop et al. (2019).

Finally, as it has shown in experimental study, our methods work quite well with high-dimensional data, even for the case $p > n$. However, theoretical results to support our methods need to study more. Moreover, directly applying our methods for huge-scale data seems to be not a correct way. In such cases, the data should be reduce its dimension using other methods such as: SRUW, rough sets, etc, and then our approaches can be considered as a terminal step of the process.

Appendix A

Mathematics materials for State of the art

A.1 Monotonicity of the EM algorithm

In this section, we provide a proof for the monotonicity of the EM algorithm based on Information inequality theorem (Lange, 1998, Section 10.3).

Theorem A.1.1 (Information Inequality)**.** *Let f and g be probability densities with respect to a measure μ. Suppose $f > 0$ and $g > 0$ almost everywhere relative to μ. If $\mathbb{E}_f$ denotes expectation with respect to the probability measure $f d\mu$, then $\mathbb{E}_f(\log f) \geq \mathbb{E}_f(\log g)$, with equality only if $f = g$ almost everywhere relative to μ.*

Proof. Because $-\log(.)$ is a strictly convex function on $(0, \infty)$, we can use Jensen′s inequality for the random variable g/f and get

$$\begin{aligned}\mathbb{E}_f(\log f) - \mathbb{E}_f(\log g) &= \mathbb{E}_f(-\log \frac{g}{f}) \\ &\geq -\log \mathbb{E}_f(\frac{g}{f}) = -\log \int \frac{g}{f} f d\mu = -\log \int g d\mu = 0.\end{aligned}$$

Equality hold only if $g/f = \mathbb{E}_f(g/f)$ almost every where relative to μ. But $\mathbb{E}_f(g/f) = \int g d\mu = 1$, i.e., $g = f$ μ-almost everywhere. □

Now, let

$$h(\mathbf{z}|\mathbf{x}; \boldsymbol{\theta}) = \frac{f(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta})}{g(\mathbf{x}; \boldsymbol{\theta})}$$

be the conditional density of Z given $\boldsymbol{X} = \mathbf{x}$. Using Theorem A.1.1 we have

$$\begin{aligned}
Q(\boldsymbol{\theta}^{[q]}; \boldsymbol{\theta}^{[q]}) - L(\boldsymbol{\theta}^{[q]}; \mathbf{x}) &= \mathbb{E}_{\boldsymbol{\theta}^{[q]}}[L_c(\boldsymbol{\theta}^{[q]}; \boldsymbol{X}, Z)] - \log g(\mathbf{x}; \boldsymbol{\theta}^{[q]}) \\
&= \mathbb{E}_{\boldsymbol{\theta}^{[q]}}\Big[\log \frac{f(\boldsymbol{X}, Z; \boldsymbol{\theta}^{[q]})}{g(X; \boldsymbol{\theta}^{[q]})}\Big|\boldsymbol{X} = \mathbf{x}\Big] \\
&= \mathbb{E}_{\boldsymbol{\theta}^{[q]}}[\log h(Z|\boldsymbol{X}; \boldsymbol{\theta}^{[q]})|\boldsymbol{X} = \mathbf{x}] \\
&\geq \mathbb{E}_{\boldsymbol{\theta}^{[q]}}[\log h(Z|\boldsymbol{X}; \boldsymbol{\theta}^{[q+1]})|\boldsymbol{X} = \mathbf{x}] \\
&= \mathbb{E}_{\boldsymbol{\theta}^{[q]}}\Big[\log \frac{f(\boldsymbol{X}, Z; \boldsymbol{\theta}^{[q+1]})}{g(\boldsymbol{X}; \boldsymbol{\theta}^{[q+1]})}\Big|\boldsymbol{X} = \mathbf{x}\Big] \\
&= Q(\boldsymbol{\theta}^{[q+1]}; \boldsymbol{\theta}^{[q]}) - L(\boldsymbol{\theta}^{[q+1]}; \mathbf{x}).
\end{aligned}$$

Therefore,

$$L(\boldsymbol{\theta}^{[q+1]}; \mathbf{x}) - L(\boldsymbol{\theta}^{[q]}; \mathbf{x}) \geq Q(\boldsymbol{\theta}^{[q+1]}; \boldsymbol{\theta}^{[q]}) - Q(\boldsymbol{\theta}^{[q]}; \boldsymbol{\theta}^{[q]}) \geq 0,$$

because $\boldsymbol{\theta}^{[q+1]} = \arg\max_{\boldsymbol{\theta} \in \Omega} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{[q]})$.

A.2 Updated formulas for the convariance matrices of GMMs

To see these updated formulas, we can rewrite the second term on the right-hand side of (2.18) above as

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}^{[q]} \log \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = &- \frac{np}{2}\log(2\pi) - \frac{1}{2}\sum_{k=1}^{K}(\sum_{i=1}^{n}\tau_{ik}^{[q]})\log|\boldsymbol{\Sigma}_k| \\
&- \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{n}\tau_{ik}^{[q]}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k), \qquad \text{(A.1)}
\end{aligned}$$

and for heteroscedastic case, note that by taking $\boldsymbol{\mu}_k^{[q+1]}$ as in (2.20) we have

$$\begin{aligned}
&\sum_{i=1}^{n}\tau_{ik}^{[q]}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k) = \mathrm{tr}\Big[\boldsymbol{\Sigma}_k^{-1}(\sum_{i=1}^{n}\tau_{ik}^{[q]}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T)\Big] \\
&= \mathrm{tr}\Big[\boldsymbol{\Sigma}_k^{-1}(\sum_{i=1}^{n}\tau_{ik}^{[q]}(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{[q+1]})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{[q+1]})^T + (\sum_{i=1}^{n}\tau_{ik}^{[q]})(\boldsymbol{\mu}_k^{[q+1]} - \boldsymbol{\mu}_k)(\boldsymbol{\mu}_k^{[q+1]} - \boldsymbol{\mu}_k)^T)\Big] \\
&= \mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}A_k) + (\sum_{i=1}^{n}\tau_{ik}^{[q]})(\boldsymbol{\mu}_k^{[q+1]} - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k^{[q+1]} - \boldsymbol{\mu}_k), \qquad \text{(A.2)}
\end{aligned}$$

where $\mathrm{tr}(A)$ represents for the trace of a square matrix A, and the sum $\sum_{i=1}^{n}\tau_{ik}^{[q]}(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{[q+1]})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{[q+1]})^T$ is denoted as A_k (for $k = 1, \ldots, K$).

We now substitute (A.2) into (A.1) to give

$$\sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{[q]}\log\mathcal{N}(\boldsymbol{x}_i;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) = -\frac{np}{2}\log(2\pi)+\frac{1}{2}\sum_{k=1}^{K}\Big\{(-\sum_{i=1}^{n}\tau_{ik}^{[q]})\log|\boldsymbol{\Sigma}_k| - \mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}A_k)\Big\} - \frac{1}{2}\sum_{k=1}^{K}(\sum_{i=1}^{n}\tau_{ik}^{[q]})(\boldsymbol{\mu}_k^{[q+1]}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k^{[q+1]}-\boldsymbol{\mu}_k). \tag{A.3}$$

Next, we use the following lemma (with a slightly change) (see Lemma 3.2.2 in Anderson (2003)).

Lemma A.2.1. *Let D be a symmetric positive definite matrix of order p, and α a positive real number. Then the maximum of*

$$f(G) = -\alpha\log|G| - tr(G^{-1}D) \tag{A.4}$$

with respect to symmetric positive definite matrices G exists, occurs at $G=(1/\alpha)D$, and has the value $f[(1/\alpha)D] = p\alpha\log\alpha - \alpha\log|D| - p\alpha$.

To get the updated estimates of $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ $(k=1,\ldots,K)$, we apply Lemma A.2.1 with $\alpha=\sum_{i=1}^{n}\tau_{ik}^{[q]}$, $G=\boldsymbol{\Sigma}_k$ and $D=A_k$ (for each $k=1,\ldots,K$), and using the fact that $(\boldsymbol{\mu}_k^{[q+1]}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\mu}_k^{[q+1]}-\boldsymbol{\mu}_k)\geq 0$ and is 0 if and only if $\boldsymbol{\mu}_k=\boldsymbol{\mu}_k^{[q+1]}$ $(k=1,\ldots,K)$. Then the maximum of the function (A.3) (with respect to $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ $(k=1,\ldots,K)$) occurs at $\boldsymbol{\mu}_k=\boldsymbol{\mu}_k^{[q+1]}$ and $\boldsymbol{\Sigma}_k=\boldsymbol{\Sigma}_k^{[q+1]}$, where $\boldsymbol{\Sigma}_k^{[q+1]}$ is determined by (2.21).

A.3 Covariance structure of GMMs in high-dimensional setting

A.3.1 Covariance structure of the expanded parsimonious Gaussian models

Model	Name	Number of parameters	$K=3, p=50$
$\lambda_k D_k A_k D_k^T$	VVV	$(K-1)+Kp+Kp(p+1)/2$	3977
$\lambda D_k A_k D_k^T$	EVV	$(K-1)+Kp+Kp(p+1)/2-(K-1)$	3975
$\lambda_k D_k A D_k^T$	VEV	$(K-1)+Kp+Kp(p+1)/2-(K-1)(p-1)$	3879
$\lambda D_k A D_k^T$	EEV	$(K-1)+Kp+Kp(p+1)/2-(K-1)p$	3877
$\lambda_k D A_k D^T$	VVE	$(K-1)+Kp+p(p+1)/2+(K-1)p$	1527
$\lambda D A_k D^T$	EVE	$(K-1)+Kp+p(p+1)/2+(K-1)(p-1)$	1525
$\lambda_k D A D^T$	VEE	$(K-1)+Kp+p(p+1)/2+(K-1)$	1429
$\lambda D A D^T$	EEE	$(K-1)+Kp+p(p+1)/2$	1427
$\lambda_k B_k$	VVI	$(K-1)+Kp+Kp$	302
λB_k	EVI	$(K-1)+Kp+Kp-(K-1)$	300
$\lambda_k B$	VEI	$(K-1)+Kp+p+(K-1)$	204
λB	EEI	$(K-1)+Kp+p$	202
$\lambda_k I_p$	VII	$(K-1)+Kp+K$	155
λI_p	EII	$(K-1)+Kp+1$	153

Table A.1: Number of free parameters for parsimonious GMMs with K components and p variable.

A.3.2 Covariance structure of the parsimonious Gaussian mixture models

$\Lambda_k = \Lambda$	$\Delta_k = \Delta$	$\omega_k = \omega$	$\Delta_k = I_p$	Covariance structure	Number of Covariance Parameters
C	C	C	C	$\boldsymbol{\Sigma}_k = \Lambda\Lambda^T + \omega I_p$	$[pd - d(d-1)/2)] + 1$
C	C	U	C	$\boldsymbol{\Sigma}_k = \Lambda\Lambda^T + \omega_k I_p$	$[pd - d(d-1)/2)] + K$
U	C	C	C	$\boldsymbol{\Sigma}_k = \Lambda_k\Lambda_k^T + \omega I_p$	$K[pd - d(d-1)/2)] + 1$
U	C	U	C	$\boldsymbol{\Sigma}_k = \Lambda_k\Lambda_k^T + \omega_k I_p$	$K[pd - d(d-1)/2)] + K$
C	C	C	U	$\boldsymbol{\Sigma}_k = \Lambda\Lambda^T + \omega\Delta$	$[pd - d(d-1)/2)] + p$
C	C	U	U	$\boldsymbol{\Sigma}_k = \Lambda\Lambda^T + \omega_k\Delta$	$[pd - d(d-1)/2)] + [K + (p-1)]$
U	C	C	U	$\boldsymbol{\Sigma}_k = \Lambda_k\Lambda_k^T + \omega\Delta$	$K[pd - d(d-1)/2)] + p$
U	C	U	U	$\boldsymbol{\Sigma}_k = \Lambda_k\Lambda_k^T + \omega\Delta_k$	$K[pd - d(d-1)/2)] + [1 + K(p-1)]$
C	U	C	U	$\boldsymbol{\Sigma}_k = \Lambda\Lambda^T + \omega\Delta_k$	$[pd - d(d-1)/2)] + [1 + K(p-1)]$
C	U	U	U	$\boldsymbol{\Sigma}_k = \Lambda\Lambda^T + \omega_k\Delta_k$	$[pd - d(d-1)/2)] + Kp$
U	U	C	U	$\boldsymbol{\Sigma}_k = \Lambda_k\Lambda_k^T + \omega\Delta_k$	$K[pd - d(d-1)/2)] + [1 + K(p-1)]$
U	U	U	U	$\boldsymbol{\Sigma}_k = \Lambda_k\Lambda_k^T + \omega_k\Delta_k$	$K[pd - d(d-1)/2)] + Kp$

Table A.2: The covariance structure, number of covariance parameters for each member of the EPGMM family McNicholas and Murphy (2010) (C = constrained, U = unconstrained).

A.3.3 Proof of Lemma 2.4.1

Denote $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d$ are d eigenvectors associated with $\lambda_1, \ldots, \lambda_d$ such that $\boldsymbol{x}_i^T\boldsymbol{x}_i = 1$ for all $i \in \{1, \ldots, d\}$. Let $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{p-d}\}$ be $p-d$ linearly independent eigenvectors corresponding to the eigenvalue γ. Note that, for $i \neq j$ then $\boldsymbol{x}_i^T\boldsymbol{x}_j = 0$, and $\boldsymbol{x}_i^T\boldsymbol{y}_j = 0$ for all $i \in \{1, \ldots, d\}$, $j \in \{1, \ldots, p-d\}$.

We now construct $p-d$ eigenvectors $\{\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_{p-d}\}$ associated with γ from the system $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{p-d}\}$ which satisfy $\tilde{\boldsymbol{y}}_i^T\tilde{\boldsymbol{y}}_j = 0$ $(i \neq j)$. This can be done by using the Gram-Schmidt orthonormalization procedure.

$$\tilde{\boldsymbol{y}}_1 = \boldsymbol{y}_1; \tag{A.5}$$

$$\tilde{\boldsymbol{y}}_j = \boldsymbol{y}_j - \sum_{i=1}^{j-1} \frac{\langle \boldsymbol{y}_j, \tilde{\boldsymbol{y}}_i \rangle}{\langle \tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{y}}_i \rangle} \tilde{\boldsymbol{y}}_i, \text{ for } 2 \leq j \leq p-d. \tag{A.6}$$

Hence,

$$\begin{aligned} \langle \tilde{\boldsymbol{y}}_2, \tilde{\boldsymbol{y}}_1 \rangle &= \langle \boldsymbol{y}_2, \tilde{\boldsymbol{y}}_1 \rangle - \frac{\langle \boldsymbol{y}_2, \tilde{\boldsymbol{y}}_1 \rangle}{\langle \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_1 \rangle} \langle \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_1 \rangle \\ &= \langle \boldsymbol{y}_2, \tilde{\boldsymbol{y}}_1 \rangle - \langle \boldsymbol{y}_2, \tilde{\boldsymbol{y}}_1 \rangle = 0. \end{aligned}$$

In addition,

$$\begin{aligned}\boldsymbol{\Sigma}\tilde{\boldsymbol{y}}_2 &= \boldsymbol{\Sigma}\boldsymbol{y}_2 - \frac{\langle \boldsymbol{y}_2, \tilde{\boldsymbol{y}}_1\rangle}{\langle \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_1\rangle}\boldsymbol{\Sigma}\tilde{\boldsymbol{y}}_1 = \gamma\boldsymbol{y}_2 - \frac{\langle \boldsymbol{y}_2, \tilde{\boldsymbol{y}}_1\rangle}{\langle \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_1\rangle}\gamma\tilde{\boldsymbol{y}}_1 \\ &= \gamma(\boldsymbol{y}_2 - \frac{\langle \boldsymbol{y}_2, \tilde{\boldsymbol{y}}_1\rangle}{\langle \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_1\rangle}\tilde{\boldsymbol{y}}_1) = \gamma\tilde{\boldsymbol{y}}_2.\end{aligned}$$

Thus, $\tilde{\boldsymbol{y}}_2$ is an eigenvector associated with γ and $\langle \tilde{\boldsymbol{y}}_2, \tilde{\boldsymbol{y}}_1\rangle = 0$.
For all $l < j$, we have

$$\begin{aligned}\langle \tilde{\boldsymbol{y}}_j, \tilde{\boldsymbol{y}}_l\rangle &= \Big\langle \boldsymbol{y}_j - \sum_{i=1}^{j-1}\frac{\langle \boldsymbol{y}_j, \tilde{\boldsymbol{y}}_i\rangle}{\langle \tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{y}}_i\rangle}\tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{y}}_l\Big\rangle \\ &= \langle \boldsymbol{y}_j, \tilde{\boldsymbol{y}}_l\rangle - \sum_{i=1}^{j-1}\frac{\langle \boldsymbol{y}_j, \tilde{\boldsymbol{y}}_i\rangle}{\langle \tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{y}}_i\rangle}\langle \tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{y}}_l\rangle.\end{aligned}$$

Since $\langle \tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{y}}_l\rangle = 0$, $\forall\ i, l < j\ (i \neq l)$, then

$$\langle \tilde{\boldsymbol{y}}_j, \tilde{\boldsymbol{y}}_l\rangle = \langle \boldsymbol{y}_j, \tilde{\boldsymbol{y}}_l\rangle - \frac{\langle \boldsymbol{y}_j, \tilde{\boldsymbol{y}}_l\rangle}{\langle \tilde{\boldsymbol{y}}_l, \tilde{\boldsymbol{y}}_l\rangle}\langle \tilde{\boldsymbol{y}}_l, \tilde{\boldsymbol{y}}_l\rangle = 0.$$

Moreover,

$$\begin{aligned}\boldsymbol{\Sigma}\tilde{\boldsymbol{y}}_j &= \boldsymbol{\Sigma}\boldsymbol{y}_j - \sum_{i=1}^{j-1}\frac{\langle \boldsymbol{y}_j, \tilde{\boldsymbol{y}}_i\rangle}{\langle \tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{y}}_i\rangle}\boldsymbol{\Sigma}\tilde{\boldsymbol{y}}_i = \gamma\boldsymbol{y}_j - \sum_{i=1}^{j-1}\frac{\langle \boldsymbol{y}_j, \tilde{\boldsymbol{y}}_i\rangle}{\langle \tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{y}}_i\rangle}\gamma\tilde{\boldsymbol{y}}_i \\ &= \gamma(\boldsymbol{y}_j - \sum_{i=1}^{j-1}\frac{\langle \boldsymbol{y}_j, \tilde{\boldsymbol{y}}_i\rangle}{\langle \tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{y}}_i\rangle}\tilde{\boldsymbol{y}}_i) = \gamma\tilde{\boldsymbol{y}}_i.\end{aligned}$$

Therefore, $\tilde{\boldsymbol{y}}_j$ is an eigenvector associated with γ and $\langle \tilde{\boldsymbol{y}}_i, \tilde{\boldsymbol{y}}_j\rangle = 0$, $\forall\ i < j$. Normalized the system $\{\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_{p-d}\}$, we obtain $p - d$ eigenvectors $\{\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_{p-d}\}$ corresponded with γ such that

$$\langle \hat{\boldsymbol{y}}_j, \hat{\boldsymbol{y}}_j\rangle = 1, \text{ and } \langle \hat{\boldsymbol{y}}_j, \hat{\boldsymbol{y}}_i\rangle = 0,$$

for all $i, j \in \{1, \ldots, p - d\}$ and $i \neq j$. Now, consider the $p \times p$ matrix Q, where

$$Q = [\boldsymbol{x}_1\ \ldots\ \boldsymbol{x}_d\ \hat{\boldsymbol{y}}_1\ \ldots\ \hat{\boldsymbol{y}}_{p-d}].$$

It is not hard to prove that Q is an orthogonal matrix and

$$\boldsymbol{\Sigma} = Q\Lambda Q^T \text{ with } \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_d, \underbrace{\gamma, \ldots, \gamma}_{p-d}).$$

A.3.4 Covariance complexity of the $[a_{kj}b_kQ_kd_k]$ family

First, we rewrite $\mathbf{\Sigma}$ as

$$\begin{aligned}\mathbf{\Sigma} &= Q\Lambda Q^T \\ &= Q[\Lambda - \gamma I_p]Q^T + \gamma Q I_p Q^T \\ &= Q[\Lambda - \gamma I_p]Q^T + \gamma I_p \\ &= [Q\Gamma][Q\Gamma]^T + \gamma I_p, \end{aligned} \tag{A.7}$$

where the $p \times d$ matrix Γ is given by

$$\Gamma = \begin{pmatrix} \boxed{\begin{matrix} \sqrt{\lambda_1 - \gamma} & & 0 \\ & \ddots & \\ 0 & & \sqrt{\lambda_d - \gamma} \end{matrix}} \\ \\ \mathbf{0} \end{pmatrix}.$$

Then, the Equation (A.7) has the same formula which is used to construct the covariance matrices of the MFA in (2.61). Hence, based on the work of Lawley and Maxwell (1962), the complexity of $\mathbf{\Sigma}$ is $d(p-(d+1)/2)+d+1$, where $d(p-(d+1)/2)$ and $(d+1)$ are the number of parameters required for the estimation of Q and $(\lambda_1, \ldots, \lambda_d, \gamma)$.

A.3.5 Covariance structure of the mixture of high-dimensional GMMs

Model name	Number of covariance parameters
$[a_{kj}b_kQ_kd_k]$	$\sum_{k=1}^K d_k[p-(d_k+1)/2] + \sum_{k=1}^K d_k + 2K$
$[a_{kj}bQ_kd_k]$	$\sum_{k=1}^K d_k[p-(d_k+1)/2] + \sum_{k=1}^K d_k + 1 + K$
$[a_kb_kQ_kd_k]$	$\sum_{k=1}^K d_k[p-(d_k+1)/2] + 3K$
$[ab_kQ_kd_k]$	$\sum_{k=1}^K d_k[p-(d_k+1)/2] + 1 + 2K$
$[a_kbQ_kd_k]$	$\sum_{k=1}^K d_k[p-(d_k+1)/2] + 1 + 2K$
$[abQ_kd_k]$	$\sum_{k=1}^K d_k[p-(d_k+1)/2] + 2 + K$
$[a_{kj}b_kQ_kd]$	$Kd[p-(d+1)/2] + Kd + K + 1$
$[a_jb_kQ_kd]$	$Kd[p-(d+1)/2] + d + K + 1$
$[a_{kj}bQ_kd]$	$Kd[p-(d+1)/2] + Kd + 2$
$[a_jbQ_kd]$	$Kd[p-(d+1)/2] + d + 2$
$[a_kb_kQ_kd]$	$Kd[p-(d+1)/2] + 2K + 1$
$[ab_kQ_kd]$	$Kd[p-(d+1)/2] + K + 2$
$[a_kbQ_kd]$	$Kd[p-(d+1)/2] + K + 2$
$[abQ_kd]$	$Kd[p-(d+1)/2] + 3$
$[a_jbQd]$	$d[p-(d+1)/2] + d + 2$
$[abQd]$	$d[p-(d+1)/2] + 3$

Table A.3: Nomenclature and number of covariance parameters for the members of the $[a_{kj}b_kQ_kd_k]$ family (Bouveyron et al., 2007a).

A.4 Mathematics materials for Lasso regression

A.4.1 Smallest value of λ to obtain all zero coefficients in Lasso regression

Consider the Lasso problem

$$\arg\min_{\boldsymbol{\beta}}\Big\{\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_1\Big\}. \tag{A.8}$$

We have

$$\begin{aligned}\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_1 &= \frac{1}{2}\sum_{i=1}^{n}(y_i-\boldsymbol{x}_i^T\boldsymbol{\beta})^2+\lambda\|\boldsymbol{\beta}\|_1\\ &= \frac{1}{2}\sum_{i=1}^{n}y_i^2+\frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i^T\boldsymbol{\beta})^2-\sum_{j=1}^{p}\langle\mathbf{y},\mathbf{x}_j\rangle\beta_j+\lambda\sum_{j=1}^{p}|\beta_j|,\end{aligned}$$

Hence, if $\lambda\geq\max_j|\langle\mathbf{y},\mathbf{x}_j\rangle|$, then

$$\begin{aligned}\lambda\sum_{j=1}^{p}|\beta_j|-\sum_{j=1}^{p}\langle\mathbf{y},\mathbf{x}_j\rangle\beta_j &\geq \lambda\sum_{j=1}^{p}|\beta_j|-\sum_{j=1}^{p}|\langle\mathbf{y},\mathbf{x}_j\rangle||\beta_j|\\ &\geq \sum_{j=1}^{p}|\beta_j|(\lambda-\max_j|\langle\mathbf{y},\mathbf{x}_j\rangle|)\\ &\geq 0.\end{aligned}$$

Therefore, with $\lambda\geq\max_j|\langle\mathbf{y},\mathbf{x}_j\rangle|$

$$\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_1\geq\frac{1}{2}\sum_{i=1}^{n}y_i^2.$$

The equality holds for $\boldsymbol{\beta}=\mathbf{0}$.

A.4.2 Proof of the properties of the Lasso solutions

The proof of Lemma 2.4.2 is given in this section.

Assume that $\mathbf{X}\hat{\boldsymbol{\beta}}>\mathbf{X}\hat{\boldsymbol{\eta}}$. Consider $\hat{\boldsymbol{\gamma}}=(\hat{\boldsymbol{\beta}}+\hat{\boldsymbol{\eta}})/2$. Using the fact that the function $f(a)=(y-a)^2$ is strictly convex, i.e., if $a\neq b$ then $f(ta+(1-t)b)<tf(a)+(1-t)f(b)$, $\forall\ t\in(0,1)$, we have

$$\begin{aligned}\frac{1}{2}\|\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\gamma}}\|_2^2 &= \frac{1}{2}\Big\|\mathbf{y}-\frac{\mathbf{X}(\hat{\boldsymbol{\beta}}+\hat{\boldsymbol{\eta}})}{2}\Big\|_2^2\\ &< \frac{\frac{1}{2}\|\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}\|_2^2+\frac{1}{2}\|\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\eta}}\|_2^2}{2}.\end{aligned} \tag{A.9}$$

Asides from this, the function $g(a) = \lambda|a|$ ($\lambda \geq 0$) is also convex. Thus,

$$\begin{aligned}\lambda\|\hat{\boldsymbol{\gamma}}\|_1 &= \lambda\Big\|\frac{\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\eta}}}{2}\Big\|_1 \\ &\leq \frac{\lambda\|\hat{\boldsymbol{\beta}}\|_1 + \lambda\|\hat{\boldsymbol{\eta}}\|_1}{2}.\end{aligned} \tag{A.10}$$

From (A.9) and (A.10), we obtain this inequality

$$\begin{aligned}\frac{1}{2}\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\gamma}}\|_2^2 + \lambda\|\hat{\boldsymbol{\gamma}}\|_1 &< \frac{\frac{1}{2}\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|_2^2 + \frac{1}{2}\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\eta}}\|_2^2}{2} + \frac{\lambda\|\hat{\boldsymbol{\beta}}\|_1 + \lambda\|\hat{\boldsymbol{\eta}}\|_1}{2} \\ &< \frac{\frac{1}{2}\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|_2^2 + \lambda\|\hat{\boldsymbol{\beta}}\|_1}{2} + \frac{\frac{1}{2}\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\eta}}\|_2^2 + \lambda\|\hat{\boldsymbol{\eta}}\|_1}{2} \\ &< c^*.\end{aligned} \tag{A.11}$$

This contradicts with the assumption that $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\eta}}$ are two solutions with optimal value c^*. Hence,

$$\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}\hat{\boldsymbol{\eta}}.$$

The last property is apparent.

Appendix B

Monotonicity of the penalized log-likelihood for Gaussian MoE

The proposed EMM algorithm maximizes the penalised log-likelihood function (4.5). To show that the penalized log-likelihood is monotonically improved, that is

$$PL(\boldsymbol{\theta}^{[q+1]}) \geq PL(\boldsymbol{\theta}^{[q]}), \tag{B.1}$$

we need to show that

$$Q(\boldsymbol{\theta}^{[q+1]}, \boldsymbol{\theta}^{[q]}) \geq Q(\boldsymbol{\theta}^{[q]}, \boldsymbol{\theta}^{[q]}). \tag{B.2}$$

Indeed, as in the standard EM algorithm algorithm for the non-penalised maximum likelihood estimation, by applying Bayes theorem we have

$$PL(\boldsymbol{\theta}) = PL_c(\boldsymbol{\theta}) - \log p(\mathbf{z}|\mathcal{D}; \boldsymbol{\theta}), \tag{B.3}$$

and by taking the conditional expectation with respect to the latent variables $\mathbf{z}$, given the observed data $\mathcal{D}$ and the current parameter estimation $\boldsymbol{\theta}^{[q]}$, the conditional expectation of the penalised completed-data log-likelhood is given by:

$$\mathbb{E}\left[PL(\boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\theta}^{[q]}\right] = \mathbb{E}\left[PL_c(\boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\theta}^{[q]}\right] - \mathbb{E}\left[\log p(\mathbf{z}|\mathcal{D}; \boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\theta}^{[q]}\right]. \tag{B.4}$$

Since the penalised log-likelihood function $PL(\boldsymbol{\theta})$ does not depend on the variables $\mathbf{z}$, its expectation with respect to $\mathbf{z}$ therefore still unchanged and we get the following relation:

$$PL(\boldsymbol{\theta}) = \underbrace{\mathbb{E}\left[PL_c(\boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\theta}^{[q]}\right]}_{Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{[q]})} - \underbrace{\mathbb{E}\left[\log p\big(\mathbf{z}|\mathcal{D}; \boldsymbol{\theta}\big)|\mathcal{D}, \boldsymbol{\theta}^{[q]}\right]}_{H(\boldsymbol{\theta}, \boldsymbol{\theta}^{[q]})}. \tag{B.5}$$

Thus, the value of change of the penalised log-likelihood function between two successive iterations is given by:

$$PL(\boldsymbol{\theta}^{[q+1]}) - PL(\boldsymbol{\theta}^{[q]}) = \Big(Q(\boldsymbol{\theta}^{[q+1]}, \boldsymbol{\theta}^{[q]}) - Q(\boldsymbol{\theta}^{[q]}, \boldsymbol{\theta}^{[q]})\Big) - \Big(H(\boldsymbol{\theta}^{[q+1]}, \boldsymbol{\theta}^{[q]}) - H(\boldsymbol{\theta}^{[q]}, \boldsymbol{\theta}^{[q]})\Big). \tag{B.6}$$

As in the standard EM algorithm, it can be easily shown, by using Jensen' inequality, that the second term $H(\boldsymbol{\theta}^{[q+1]}, \boldsymbol{\theta}^{[q]}) - H(\boldsymbol{\theta}^{[q]}, \boldsymbol{\theta}^{[q]})$ in the r.h.s of (B.6) is negative and we therefore just need to show that the first term $Q(\boldsymbol{\theta}^{[q+1]}, \boldsymbol{\theta}^{[q]}) - Q(\boldsymbol{\theta}^{[q]}, \boldsymbol{\theta}^{[q]})$ is positive.

In the following, we show that $Q(\boldsymbol{\theta}^{[q+1]}, \boldsymbol{\theta}^{[q]}) \geq Q(\boldsymbol{\theta}^{[q]}, \boldsymbol{\theta}^{[q]})$. First, the Q-function is decomposed as

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{[q]}) = Q(\mathbf{w}; \boldsymbol{\theta}^{[q]}) + Q(\{\boldsymbol{\beta}_k, \sigma_k^2\}; \boldsymbol{\theta}^{[q]}) \tag{B.7}$$

and is accordingly maximized separately with respect to $\mathbf{w}$, $\{\boldsymbol{\beta}_k\}$ and $\{\sigma_k^2\}$.

To update $\mathbf{w}$, first we use a univariate MM algirthm to iteratively maximize the minorizing function which satisfies

$$Q(\mathbf{w}; \boldsymbol{\theta}^{[q]}) \geq G(\mathbf{w}|\mathbf{w}^{[q]}), \forall \mathbf{w} \tag{B.8}$$

and

$$Q(\mathbf{w}^{[q]}; \boldsymbol{\theta}^{[q]}) = G(\mathbf{w}^{[q]}|\mathbf{w}^{[q]}). \tag{B.9}$$

In our situation, the minorizing function is concave and has a separate structure. We thus use a one-dimensional Newton Raphson algorithm to maximize it. Thus, the solution $\mathbf{w}^{[q+1]}$ guarantees

$$G(\mathbf{w}^{[q+1]}|\mathbf{w}^{[q]}) \geq G(\mathbf{w}^{[q]}|\mathbf{w}^{[q]}) \tag{B.10}$$

and hence we have

$$Q(\mathbf{w}^{[q+1]}; \boldsymbol{\theta}^{[q]}) \geq G(\mathbf{w}^{[q+1]}|\mathbf{w}^{[q]}) \geq G(\mathbf{w}^{[q]}|\mathbf{w}^{[q]}) = Q(\mathbf{w}^{[q]}; \boldsymbol{\theta}^{[q]}). \tag{B.11}$$

Hence, the MM algorithm leads to the improvement of the value of the $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$ function.

For the second version of the EM algorithm which uses the coordinate ascent algorithm to update $\mathbf{w}$, we rely on the work of Tseng (1988) and Tseng (2001), where it is proved that, if the nonsmooth part of $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$ has a separate structure, the coordinate ascent algorithm is successful in finding the $\mathbf{w}^{[q+1]} = \arg\max_{\mathbf{w}} Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$. At each step of the coordinate ascent algorithm, within the M-step of the EM algorithm, we iteratively update the jth component, while fixing the other parameters to their previous values:

$$w_{kj}^{[q,s+1]} = \arg\max_{w_{kj}} Q(w_{kj}; \boldsymbol{\theta}^{[q,s]}), \tag{B.12}$$

s being the current iteration of the coordinate ascent algorithm. The function $Q(w_{kj}; \boldsymbol{\theta}^{[q]})$ is concave, and the used iterative procedure to find $w_{kj}^{[q+1]}$ is the Newton Raphson algorithm. Hence, the coordinate ascent leads to the improvement of the function $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$, that is

$$Q(\mathbf{w}^{[q+1]}; \boldsymbol{\theta}^{[q]}) \geq Q(\mathbf{w}^{[q]}; \boldsymbol{\theta}^{[q]}). \tag{B.13}$$

For the proximal Newton-type method, we based on the works of Lee et al. (2014) and Lee et al. (2006) to show that the value of $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$ is always increase after each loop. Specially, the proximal Newton-type method which uses a constant matrix $\mathbf{B}$ to approximate the Hessian matrix is proofed to be a special case of the MM algorithm (see (Lange, 2013, Sec. 8.7) and Gormley et al. (2008)). Hence, it always improve the values of $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$. For more details see Appendix C.1.

Finally, the updates of the experts' parameters $\{\boldsymbol{\beta}\}$ and $\{\sigma^2\}$ are performed by separate maximizations of $Q(\boldsymbol{\beta}, \sigma^2; \boldsymbol{\theta}^{[q]})$. This function is concave and has the quadratic form. Hence, the coordinate ascent algorithm with soft-thresholding operator is successful to provide the updates

$$\boldsymbol{\beta}^{[q+1]} = \arg\max_{\boldsymbol{\beta}} Q(\boldsymbol{\beta}, \sigma^{[q]}; \boldsymbol{\theta}^{[q]}), \tag{B.14}$$

and

$$\sigma^{[q+1]} = \arg\max_{\sigma} Q(\boldsymbol{\beta}^{[q+1]}, \sigma; \boldsymbol{\theta}^{[q]}) \tag{B.15}$$

and thus we have

$$Q(\boldsymbol{\beta}^{[q+1]}; \boldsymbol{\theta}^{[q]}) \geq Q(\boldsymbol{\beta}; \boldsymbol{\theta}^{[q]}) \geq Q(\boldsymbol{\beta}^{[q]}; \boldsymbol{\theta}^{[q]}), \tag{B.16}$$

and

$$Q(\sigma^{[q+1]}; \boldsymbol{\theta}^{[q]}) \geq Q(\sigma; \boldsymbol{\theta}^{[q]}) \geq Q(\sigma^{[q]}; \boldsymbol{\theta}^{[q]}). \tag{B.17}$$

Equations (B.11), (B.13), (B.16), and (B.17) and the increasing property of the proximal Newton-type methods show that $(B.2)$ holds; hence, the penalised log-likelihood in monotonically increased by the proposed algorithms.

Appendix C

Proximal Newton-type methods

C.1 Proximal Newton-type methods

Assume that we want to solve an optimization problem given by

$$\min_{x\in\mathbb{R}^n} f(x) = g(x) + h(x), \tag{C.1}$$

with a composite function $f(x)$ where g is a convex, continuously differentiable loss function, and h is a convex but non differentiable penalty function. Such problems include the Lasso, elastic net, etc. Proximal Newton-type methods approximate only the smooth part g with a local quadratic function of the form:

$$\hat{f}_k(x) = g(x_k) + \nabla g(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T H_k(x - x_k) + h(x), \tag{C.2}$$

where $\nabla g(x_k)$ is the gradient vector of g at x_k and H_k is an approximation to the Hessian matrix $\nabla^2 g(x_k)$. If we choose $H_k = \nabla^2 g(x_k)$, we obtain the *proximal Newton method*. In this method, one uses an iterative algorithm with initial value x_0 and in which at step k minimizes the proximal function $\hat{f}_k(x)$ instead of f and then searches for the next value x_{k+1} based on the solution of (C.2) that will improve the value of f, i.e., $f(x_{k+1}) < f(x_k)$ by using a back tracking line search until the algorithm converges. Lee et al. (2014) and Lee et al. (2006) studied convergence properties of proximal Newton methods. A generic proximal Newton-type method can be listed as in Algorithm 4 (see Lee et al. (2014)).

C.2 Partial quadratic approximation for the gating network

The $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$ function in (3.9) is given as following

$$Q(\mathbf{w}; \boldsymbol{\theta}^{[q]}) = I(\mathbf{w}) - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1,$$

where the concave, continuously differentiable function $I(\mathbf{w})$ is

$$I(\mathbf{w}) = \sum_{i=1}^{n}\sum_{k=1}^{K-1} \tau_{ik}^{[q]}(w_{k0} + \boldsymbol{x}_i^T \boldsymbol{w}_k) - \sum_{i=1}^{n} \log\Big[1 + \sum_{k=1}^{K-1} e^{w_{k0} + \boldsymbol{x}_i^T \boldsymbol{w}_k}\Big]$$

Algorithm 4 A generic proximal Newton-type procedure

1: Starting point $x_0 \in \text{dom} f$.
2: **repeat**
3: Choose H_k, a positive definite approximation to the Hessian.
4: Solve the subproblem for a search direction:

$$\triangle x_k \leftarrow \arg\min_d \nabla g(x_k)^T d + \frac{1}{2} d^T H_k d + h(x_k + d).$$

5: Select t_k with a backtracking line search.
6: Update: $x_{k+1} \leftarrow x_k + t_k \triangle x_k$.
7: **until** a stopping condition is satisfied.

By taking the first and second derivatives of $I(\mathbf{w})$ with respect to $(w_{k0}, \boldsymbol{w}_k)$

$$\frac{\partial I(\mathbf{w})}{\partial w_{kj}} = \sum_{i=1}^{n} (\tau_{ik}^{[q]} - \pi_k(\boldsymbol{x}_i; \mathbf{w})) x_{ij}, \tag{C.3}$$

$$\frac{\partial^2 I(\mathbf{w})}{\partial w_{kj} \partial w_{kh}} = -\sum_{i=1}^{n} x_{ij} x_{ih} \pi_k(\boldsymbol{x}_i; \mathbf{w})(1 - \pi_k(\boldsymbol{x}_i; \mathbf{w})), \tag{C.4}$$

for $j, h \in \{0, 1, \ldots, p\}$ with $x_{i0} = 1$, then the partial quadratic approximation to $I(\mathbf{w})$ with respect to $(w_{k0}, \boldsymbol{w}_k)$ at $(\tilde{w}_{k0}, \tilde{\boldsymbol{w}}_k)$ is given by

$$l_{I_k}(w_{k0}, \boldsymbol{w}_k) = -\frac{1}{2} \sum_{i=1}^{n} d_{ik} (c_{ik} - w_{k0} - \boldsymbol{x}_i^T \boldsymbol{w}_k)^2 + C(\tilde{\mathbf{w}}), \tag{C.5}$$

and

$$c_{ik} = \tilde{w}_{k0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{w}}_k + \frac{\tau_{ik}^{[q]} - \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)}{\pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)(1 - \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i))}, \tag{C.6}$$

$$d_{ik} = \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)(1 - \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)), \tag{C.7}$$

$C(\tilde{\mathbf{w}})$ is a function of $\tilde{\mathbf{w}}$.

C.3 Quadratic approximation for the experts network

C.3.1 Quadratic approximation for the Poisson outputs

In this part, the quadratic approximation for the function $Q_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]})$ of the Poisson model in (4.12) is constructed using Taylor expansion. This function is given by

$$Q_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) = P_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) - \lambda_k \|\boldsymbol{\beta}_k\|_1, \tag{C.8}$$

where $P_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]})$ is a concave, continuously differentiable function and

$$P_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n} \tau_{ik}^{[q]} \big[-\exp(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k) + y_i(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k) - \log(y_i!) \big]. \tag{C.9}$$

The first and second derivatives of $P_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]})$ with respect to $(\beta_{k0}, \boldsymbol{\beta}_k)$ can easily obtained. It is not hard to show that

$$\begin{aligned} \frac{\partial P_k}{\partial \beta_{kj}} &= \sum_{i=1}^{n} \tau_{ik}^{[q]} \big[y_i x_{ij} - x_{ij} \exp(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k) \big]; \\ \frac{\partial^2 P_k}{\partial \beta_{kj} \partial \beta_{kh}} &= -\sum_{i=1}^{n} \tau_{ik}^{[q]} x_{ij} x_{ih} \exp(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k); \end{aligned}$$

for $j, h \in \{0, \ldots, p\}$ and $x_{i0} = 1$.
Thus the quadratic approximation of $P_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]})$ at $(\tilde{\beta}_{k0}, \tilde{\boldsymbol{\beta}}_k)$ is given as following

$$\tilde{P}_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) = -\frac{1}{2} \sum_{i=1}^{n} a_{ik}(b_{ik} - \beta_{k0} - \boldsymbol{x}_i^T \boldsymbol{\beta}_k)^2 + D(\tilde{\beta}_{k0}, \tilde{\boldsymbol{\beta}}_k), \tag{C.10}$$

with

$$\begin{aligned} a_{ik} &= \tau_{ik}^{[q]} \exp(\tilde{\beta}_{k0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}_k); \\ b_{ik} &= \frac{y_i}{\exp(\tilde{\beta}_{k0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}_k)} - 1 + \tilde{\beta}_{k0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}_k; \end{aligned}$$

and $D(\tilde{\beta}_{k0}, \tilde{\boldsymbol{\beta}}_k)$ is a function of $(\tilde{\beta}_{k0}, \tilde{\boldsymbol{\beta}}_k)$.

C.3.2 Partial quadratic approximation for the Multinomial outputs

Finally, we construct the quadratic approximation for the function $Q_k(\boldsymbol{\beta}_k; \boldsymbol{\theta}^{[q]})$ in (4.12), where as before

$$Q_k(\boldsymbol{\beta}_k; \boldsymbol{\theta}^{[q]}) = I(\boldsymbol{\beta}_k) - \sum_{r=1}^{R-1} \lambda_{kr} \|\boldsymbol{\beta}_{kr}\|_1, \tag{C.11}$$

$I(\boldsymbol{\beta}_k)$ is a concave, continuously differentiable function and

$$I(\boldsymbol{\beta}_k) = \sum_{i=1}^{n} \tau_{ik}^{[q]} \Big[\sum_{r=1}^{R-1} u_{ir}(\beta_{kr0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr}) - \log\big(1 + \sum_{r=1}^{R-1} \exp(\beta_{kr0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr})\big) \Big]. \tag{C.12}$$

The first and second derivatives of $I(\boldsymbol{\beta}_k)$ with respect to $(\beta_{kr0}, \boldsymbol{\beta}_{kr})$ are

$$\frac{\partial I(\boldsymbol{\beta}_k)}{\partial \beta_{krj}} = \sum_{i=1}^{n} \tau_{ik}^{[q]} x_{ij}(u_{ir} - \alpha_{kr}(\boldsymbol{\beta}_k; \boldsymbol{x}_i)), \tag{C.13}$$

$$\frac{\partial^2 I(\boldsymbol{\beta}_k)}{\partial \beta_{krj} \partial \beta_{krh}} = -\sum_{i=1}^{n} \tau_{ik}^{[q]} x_{ij} x_{ih} \alpha_{kr}(\boldsymbol{\beta}_k; \boldsymbol{x}_i)(1 - \alpha_{kr}(\boldsymbol{\beta}_k; \boldsymbol{x}_i)), \tag{C.14}$$

for $j, h \in \{0, 1, \ldots, p\}$ and $x_{i0} = 1$. Hence, the partial quadratic approximation $\tilde{I}_r(\boldsymbol{\beta}_k)$ of $I(\boldsymbol{\beta}_k)$ with respect to $(\beta_{kr0}, \boldsymbol{\beta}_{kr})$ at $\tilde{\boldsymbol{\beta}}_k$ can be described as following

$$\tilde{I}_r(\boldsymbol{\beta}_k) = -\frac{1}{2}\sum_{i=1}^{n} \tau_{ik}^{[q]} d_{ikr}(c_{ikr} - \beta_{kr0} - \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr})^2 + E(\tilde{\boldsymbol{\beta}}_k), \tag{C.15}$$

with

$$c_{ikr} = \tilde{\beta}_{kr0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}_{kr} + \frac{u_{ir} - \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)}{\alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)(1 - \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i))}, \tag{C.16}$$

$$d_{ikr} = \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)(1 - \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)), \tag{C.17}$$

$E(\tilde{\boldsymbol{\beta}}_k)$ is a function of $\tilde{\boldsymbol{\beta}}_k$.

List of Publications and Communications

Journal papers

- Chamroukhi, F. and Huynh, B.-T. (2019). Regularized maximum likelihood estimation and feature selection in mixtures-of-experts models. *Journal de la SFdS*, 160(1) : 57 − 85.

- Huynh, B. T. and Chamroukhi, F. (2019). Estimation and feature selection in mixtures of generalized linear experts models. *arXiv preprint arXiv:1907.06994*, (Submitted).

Conference papers and presentations

- Huynh, B. T. and Chamroukhi, F. (2018). Regularised maximum-likelihood inference of mixture of experts for regression and clustering. In *26th European Symposium on Artificial Neural Networks Bruges, Belgium, April 25-26-27 (ESANN)*, pages 473 − 478.

- Chamroukhi, F. and Huynh, B. T. (2018). Regularized maximum-likelihood estimation and feature selection in mixtures-of-experts models. In 50^e *Journées de Statistique de la SFdS, Paris-Saclay, 28 mai* - 1^{er} *juin, 2018*.

- Chamroukhi, F. and Huynh, B. T. (2018). Regularized maximum-likelihood estimation of mixture-of-experts for regression and clustering. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1 − 8. IEEE.

- Huynh, B. T. and Chamroukhi, F. (2018). Regularized maximum-likelihood estimation and feature selection in mixtures-of-experts models. In *The 9th Vietnam Mathematical Congress, Nha Trang, 14 - 18 August, 2018*.

Scientific talks

- Huynh, B.T. (Febuary, 2017). Statistical learning in large-scale scenarios: A state of the art. Talk at LMNO Lab.

- Huynh, B. T. (June, 2018). Regularised Maximum Likelihood Estimation of Mixtures-of-Experts. Talk at S4D 2018 Research Summer School.

- Huynh, B.T. (May, 2019). Estimation and feature selection in mixtures of generalized linear experts models. Talk at LMNO Lab.

Software

- R package of open-source codes available at `https://github.com/fchamroukhi/prEMME`:
  - RMoE: regularized MoE with Gaussian outputs;
  - PoissonRMoE: regularized MoE with Poisson outputs;
  - LogisticRMoE: regularized MoE with multinomial outputs.

Bibliography

Aitkin, M. (1996). A general maximum likelihood analysis of overdispersion in generalized linear models. *Statistics and computing*, 6(3):251–262.

Aitkin, M. (1999). A general maximum likelihood analysis of variance components in generalized linear models. *Biometrics*, 55(1):117–128.

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

Anderson, T. W. (2003). *An introduction to multivariate statistical analysis (3rd edition)*. John Wiley & Sons, Inc.

Andrews, J. L. and Mcnicholas, P. D. (2011). Extending mixtures of multivariate t-factor analyzers. *Statistics and Computing*, 21(3):361–373.

Antoniadis, A. (1997). Wavelets in statistics: a review. *Journal of the Italian Statistical Society*, 6(2):97–130.

Baek, J., McLachlan, G. J., and Flack, L. K. (2009). Mixtures of factor analyzers with common factor loadings: Applications to the clustering and visualization of high-dimensional data. *IEEE transactions on pattern analysis and machine intelligence*, 32(7):1298–1309.

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

Bartcus, M., Chamroukhi, F., and Glotin, H. (2015). Hierarchical dirichlet process hidden markov model for unsupervised bioacoustic analysis. In *The International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland.

Bellman, R. (1960). *Introduction to matrix analysis*. McGraw-Hill book company, Inc.

Bickel, P. J., Ritov, Y., Tsybakov, A. B., et al. (2009). Simultaneous analysis of Lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732.

Biernacki, C., Celeux, G., and Govaert, G. (2000). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence*, 22(7):719–725.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

Bouveyron, C. and Brunet, C. (2012). Simultaneous model-based clustering and visualization in the fisher discriminative subspace. *Statistics and Computing*, 22(1):301–324.

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

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

Bouveyron, C., Girard, S., and Schmid, C. (2007b). High-dimensional discriminant analysis. *Communications in Statistics − Theory and Methods*, 36(14):2607–2623.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Breiman, L. (1995). Better subset selection using the non-negative garotte. *Technometrics*, 37(4):738–754.

Bühlmann, P. and Van De Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

Bunea, F. (2008). Honest variable selection in linear and logistic regression models via ℓ_1 and $\ell_1 + \ell_2$ penalization. *Electronic Journal of Statistics*, 2:1153–1194.

Buu, A., Johnson, N. J., Li, R., and Tan, X. (2011). New variable selection methods for zero-inflated count data with applications to the substance abuse field. *Statistics in medicine*, 30(18):2326–2340.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern recognition*, 28(5):781–793.

Celeux, G., Maugis-Rabusseau, C., and Sedki, M. (2019). Variable selection in model-based clustering and discriminant analysis with a regularization approach. *Advances in Data Analysis and Classification*, 13(1):259–278.

Celeux, G. and Soromenho, G. (1996). An entropy criterion for assessing the number of clusters in a mixture model. *Journal of classification*, 13(2):195–212.

Chamroukhi, F. (13 december 2010). *Hidden process regression for curve modeling, classification and tracking*. Ph.D. thesis, Université de Technologie de Compiègne.

Chamroukhi, F. (2013). Robust EM algorithm for model-based curve clustering. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN), IEEE*, pages 1–8, Dallas, Texas.

Chamroukhi, F. (2015). Non-normal mixtures of experts. *arXiv:1506.06707*.

Chamroukhi, F. (2016a). Piecewise regression mixture for simultaneous functional data clustering and optimal segmentation. *Journal of Classification*, 33(3):374–411.

Chamroukhi, F. (2016b). Robust mixture of experts modeling using the t-distribution. *Neural Networks*, 79:20–36.

Chamroukhi, F. (2016c). Skew-normal mixture of experts. In *Neural Networks (IJCNN), 2016 International Joint Conference on Neuron Networks*, pages 3000–3007. IEEE.

Chamroukhi, F. (2016d). Unsupervised learning of regression mixture models with unknown number of components. *Journal of Statistical Computation and Simulation*, 86:2308 – 2334.

Chamroukhi, F. (2017). Skew t mixture of experts. *Neurocomputing - Elsevier*, 266:390–408.

Chamroukhi, F., Bartcus, M., and Glotin, H. (2014). Bayesian non-parametric parsimonious gaussian mixture for clustering. In *Proceedings of 22nd International Conference on Pattern Recognition (ICPR)*, Stockholm.

Chamroukhi, F. and Huynh, B. T. (2018). Regularized maximum-likelihood estimation of mixture-of-experts for regression and clustering. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Chamroukhi, F. and Huynh, B.-T. (2019). Regularized maximum likelihood estimation and feature selection in mixtures-of-experts models. *Journal de la SFdS*, 160(1):57–85.

Chamroukhi, F., Lecocq, F., and Bartcus, M. (2019a). *MEteorits: Mixtures-of-ExperTs modEling for cOmplex non-noRmal dIsTributions*. R package version 0.1.0.

Chamroukhi, F., Lecocq, F., and Nguyen, H. D. (2019b). Regularized estimation and feature selection in mixtures of gaussian-gated experts models. *arXiv:1909.05494*.

Chamroukhi, F. and Nguyen, H. D. (2018). Model-based clustering and classification of functional data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. DOI: 10.1002/widm.1298.

Chamroukhi, F., Samé, A., and Aknin, P. (2008). Switch mechanism diagnosis using a pattern recognition approach. In *The 4th IET International Conference on Railway Condition Monitoring RCM (IEEE)*, pages 1–4, Derby, UK. IEEE.

Chamroukhi, F., Samé, A., and Aknin, P. (2009a). A probabilistic approach for the classification of railway switch operating states. In *The VIth International Conference on Condition Monitoring and Machinery Failure Prevention Technologies*, Dublin, UK. IEEE.

Chamroukhi, F., Samé, A., Govaert, G., and Aknin, P. (2009b). Time series modeling by a regression approach based on a latent process. *Neural Networks*, 22(5-6):593–602.

Chamroukhi, F., Samé, A., Govaert, G., and Aknin, P. (2011). A dynamic probabilistic modeling of railway switches operating states. In *Proccedings of the 9th World Congress on Railway Research (WCRR)*, Lille.

Chamroukhi, F., Trabelsi, D., Mohammed, S., Oukhellou, L., and Amirat, Y. (2013). Joint segmentation of multivariate time series with hidden process regression for human activity recognition. *Neurocomputing*, 120:633–644.

Chang, W.-C. (1983). On using principal components before separating a mixture of two multivariate normal distributions. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 32(3):267–275.

DasGupta, A. (2008). *Asymptotic theory of statistics and probability*. Springer Science & Business Media.

Deleforge, A., Forbes, F., and Horaud, R. (2015). High-dimensional regression with gaussian mixtures and partially-latent response variables. *Statistics and Computing*, 25(5):893–911.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. of the royal statistical society. Series B*, pages 1–38.

Devijver, E. (2015). An ℓ_1-oracle inequality for the Lasso in multivariate finite mixture of multivariate gaussian regression models. *ESAIM: Probability and Statistics*, 19:649–670.

Donoho, D. L., Elad, M., and Temlyakov, V. N. (2006). Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on information theory*, 52(1):6–18.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of statistics*, 32(2):407–499.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360.

Fisher, R. A. (1912). On an absolute criterion for fitting frequency curves. *Messenger of Mathematics*, 41:155–156.

Fonseca, J. R. and Cardoso, M. G. (2007). Mixture-model cluster analysis using information theoretical criteria. *Intelligent Data Analysis*, 11(2):155–173.

Fop, M., Murphy, T. B., and Scrucca, L. (2019). Model-based clustering with sparse covariance matrices. *Statistics and Computing*, 29(4):791–819.

Foster, D. P., George, E. I., et al. (1994). The risk inflation criterion for multiple regression. *The Annals of Statistics*, 22(4):1947–1975.

Fraley, C. and Raftery, A. E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, 41(8):578–588.

Fraley, C. and Raftery, A. E. (2007). Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of classification*, 24(2):155–181.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American statistical association*, 84(405):165–175.

Frommlet, F. and Nuel, G. (2016). An adaptive ridge procedure for l_0 regularization. *PloS one*, 11(2):e0148620.

Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models (Springer Series in Statistics)*. Springer Verlag, New York.

Frühwirth-Schnatter, S. and Pyne, S. (2010). Bayesian inference for finite mixtures of univariate and multivariate skew-normal and skew-t distributions. *Biostatistics*, 11(2):317–336.

Ghahramani, Z. and Hinton, G. E. (1996). The EM algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto.

Ghosh, D. and Chinnaiyan, A. M. (2002). Mixture modelling of gene expression data from

microarray experiments. *Bioinformatics*, 18(2):275–286.

Gormley, I. C., Murphy, T. B., et al. (2008). A mixture of experts model for rank data with applications in election studies. *The Annals of Applied Statistics*, 2(4):1452–1477.

Grün, B. and Leisch, F. (2007). Fitting finite mixtures of generalized linear regressions in r. *Computational Statistics & Data Analysis*, 51(11):5247–5252.

Grün, B. and Leisch, F. (2008). Flexmix version 2: finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, 28(4).

Hastie, T., Buja, A., and Tibshirani, R. (1995). Penalized discriminant analysis. *The Annals of Statistics*, pages 73–102.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Taylor & Francis.

Hinton, G. E., Revow, M., and Dayan, P. (1995). Recognizing handwritten digits using mixtures of linear models. In *Advances in neural information processing systems*, pages 1015–1022.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hoerl, R. W. (1985). Ridge analysis 25 years later. *The American Statistician*, 39(3):186–192.

Huang, T., Peng, H., and Zhang, K. (2017). Model selection for gaussian mixture models. *Statistica Sinica*, pages 147–169.

Hui, F. K., Warton, D. I., Foster, S. D., et al. (2015). Multi-species distribution modeling using penalized mixture of regressions. *The Annals of Applied Statistics*, 9(2):866–882.

Hunter, D. R. and Lange, K. (2004). A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37.

Hunter, D. R. and Li, R. (2005). Variable selection using MM algorithms. *Annals of statistics*, 33(4):1617–1642.

Huynh, B. T. and Chamroukhi, F. (2018). Regularised maximum-likelihood inference of mixture of experts for regression and clustering. In *26th European Symposium on Artificial Neural Networks Bruges, Belgium, April 25-26-27 (ESANN)*, pages 473–478.

Huynh, B. T. and Chamroukhi, F. (2019). Estimation and feature selection in mixtures of generalized linear experts models. *arXiv preprint arXiv:1907.06994*.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Jansen, R. (1993). Maximum likelihood in a generalized linear finite mixture model by using the EM algorithm. *Biometrics*, pages 227–231.

Jiang, W. and Tanner, M. A. (1999a). Hierarchical mixtures-of-experts for exponential family regression models: approximation and maximum likelihood estimation. *Annals of Statistics*, pages 987–1011.

Jiang, W. and Tanner, M. A. (1999b). On the approximation rate of hierarchical mixtures-of-

experts for generalized linear models. *Neural computation*, 11(5):1183–1198.

Jiang, W. and Tanner, M. A. (1999c). On the identifiability of mixtures-of-experts. *Neural Networks*, 12(9):1253–1258.

Jiang, W. and Tanner, M. A. (2000). On the asymptotic normality of hierarchical mixtures-of-experts for generalized linear models. *IEEE Transactions on Information Theory*, 46(3):1005–1013.

Jiang, Y., Conglian, Y., and Qinghua, J. (2018). Model selection for the localized mixture of experts models. *Journal of Applied Statistics*, 45(11):1994–2006.

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, 6(2):181–214.

Khalili, A. (2010). New estimation and feature selection methods in mixture-of-experts models. *Canadian Journal of Statistics*, 38(4):519–539.

Khalili, A. (2011). An overview of the new feature selection methods in finite mixture of regression models. *Journal of The Iranian Statistical Society*, 10(2):201–235.

Khalili, A. and Chen, J. (2007). Variable selection in finite mixture of regression models. *Journal of the American Statistical association*, 102(479):1025–1038.

Khalili, A. and Lin, S. (2013). Regularization in finite mixture of regression models with diverging number of parameters. *Biometrics*, 69(2):436–446.

Lange, K. (1998). *Numerical analysis for statisticians*. Springer-Verlag.

Lange, K. (2013). *Optimization (2nd edition)*. Springer.

Law, M. H., Figueiredo, M. A., and Jain, A. K. (2004). Simultaneous feature selection and clustering using mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 26(9):1154–1166.

Lawley, D. N. and Maxwell, A. E. (1962). Factor analysis as a statistical method. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 12(3):209–229.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788.

Lee, J. D., Sun, Y., and Saunders, M. A. (2014). Proximal newton-type methods for minimizing composite functions. *SIAM Journal on Optimization*, 24(3):1420–1443.

Lee, S. and McLachlan, G. J. (2014). Finite mixtures of multivariate skew t-distributions: some recent and new results. *Statistics and Computing*, 24(2):181–202.

Lee, S.-I., Lee, H., Abbeel, P., and Ng, A. Y. (2006). Efficient l_1 regularized logistic regression. In *AAAI*, volume 6, pages 401–408.

Lee, S. X. and McLachlan, G. J. (2013). Model-based clustering and classification with non-normal mixture distributions. *Statistical Methods & Applications*, 22(4):427–454.

Lee, S. X. and Mclachlan, G. J. (2013). On mixtures of skew normal and skew t-distributions. *Advances in Data Analysis and Classification*, 7(3):241–266.

Lehmann, E. L. and Casella, G. (2006). *Theory of point estimation*. Springer Science & Business Media.

Liu, J. S., Zhang, J. L., Palumbo, M. J., and Lawrence, C. E. (2003). Bayesian clustering with variable and transformation selections. *Bayesian statistics*, 7:249–275.

Lloyd-Jones, L. R., Nguyen, H. D., and McLachlan, G. J. (2018). A globally convergent algorithm for Lasso-penalized mixture of linear regression models. *Computational Statistics & Data Analysis*, 119:19–38.

Massart, P. and Meynet, C. (2011). The Lasso as an ℓ_1-ball model selection procedure. *Electronic Journal of Statistics*, 5:669–687.

Maugis, C., Celeux, G., and Martin-Magniette, M.-L. (2009a). Variable selection for clustering with gaussian mixture models. *Biometrics*, 65(3):701–709.

Maugis, C., Celeux, G., and Martin-Magniette, M.-L. (2009b). Variable selection in model-based clustering: A general variable role modeling. *Computational Statistics & Data Analysis*, 53(11):3872–3882.

McCullagh, P. (2018). *Generalized linear models*. Routledge.

McLachlan, G. and Krishnan, T. (2008). *The* EM *algorithm and extensions (2nd edition)*. New Jersey: Wiley.

McLachlan, G. and Peel, D. (2000). Mixtures of factor analyzers. In *In Proceedings of the Seventeenth International Conference on Machine Learning*. Citeseer.

McLachlan, G. J., Bean, R., and Jones, L. B.-T. (2007). Extension of the mixture of factor analyzers model to incorporate the multivariate t-distribution. *Computational Statistics & Data Analysis*, 51(11):5327–5338.

McLachlan, G. J., Bean, R., and Peel, D. (2002). A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18(3):413–422.

McLachlan, G. J. and Peel, D. (1998). Robust cluster analysis via mixtures of multivariate t-distributions. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 658–666. Springer.

McLachlan, G. J. and Peel., D. (2000). *Finite mixture models*. New York: Wiley.

McLachlan, G. J., Peel, D., and Bean, R. (2003). Modelling high-dimensional data by mixtures of factor analyzers. *Computational Statistics & Data Analysis*, 41(3-4):379–388.

McNicholas, P. D. and Murphy, T. B. (2010). Model-based clustering of microarray expression data via latent gaussian mixture models. *Bioinformatics*, 26(21):2705–2712.

Meinshausen, N., Bühlmann, P., et al. (2006). High-dimensional graphs and variable selection with the Lasso. *The annals of statistics*, 34(3):1436–1462.

Meng, X.-L. and Van Dyk, D. (1997). The EM algorithm-an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(3):511–

567.

Meynet, C. (2013). An ℓ_1-oracle inequality for the Lasso in finite mixture gaussian regression models. *ESAIM: Probability and Statistics*, 17:650–671.

Minka, T. (2001). Algorithms for maximum-likelihood logistic regression. Technical report, CMU, Department of Statistics, TR 758.

Mkhadri, A., Celeux, G., and Nasroallah, A. (1997). Regularization in discriminant analysis: an overview. *Computational Statistics & Data Analysis*, 23(3):403–423.

Morvan, M., Devijver, E., Giacofci, M., and Monbet, V. (2019). Prediction of the nash through penalized mixture of logistic regression models. *HAL-02151564*.

Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234.

Nelder, J. A. and Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384.

Nguyen, D. H. (2015). *Finite mixture models for regression problems*. PhD thesis, The University of Queensland, Australia.

Nguyen, H. D. and Chamroukhi, F. (2018). Practical and theoretical aspects of mixture-of-experts modeling: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, pages e1246–n/a.

Nguyen, H. D., Chamroukhi, F., and Forbes, F. (2019a). Approximation results regarding the multiple-output mixture of linear experts model. *Neurocomputing*, doi:10.1016/j.neucom.2019.08.014.

Nguyen, H. D. and McLachlan, G. J. (2016). Laplace mixture of linear experts. *Computational Statistics & Data Analysis*, 93:177–191.

Nguyen, T.-T., Nguyen, D. H., Chamroukhi, F., and McLachlan, G. (2019b). Approximation by finite mixtures of continuous density functions that vanish at infinity. *Submitted*. Submitted.

Onanena, R., Chamroukhi, F., Oukhellou, L., Candusso, D., Aknin, P., and Hissel, D. (2009). Estimation of fuel cell life time using latent variables in regression context. In *Proceeding of the Eighth IEEE International Conference on Machine Learning and Applications (IEEE ICMLA)*, pages 632–637, Miami, Flordia, USA. IEEE Computer Society.

Oskrochi, G. and Davies, R. (1997). An EM-type algorithm for multivariate mixture models. *Statistics and Computing*, 7(2):145–151.

Pan, W. and Shen, X. (2007). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, 8(May):1145–1164.

Park, M. Y. and Hastie, T. (2007a). L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677.

Park, M. Y. and Hastie, T. (2007b). Penalized logistic regression for detecting gene interactions.

Biostatistics, 9(1):30–50.

Pearson, K. (1894). Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110.

Pearson, K. (1901). On lines and planes of closest fit to systems of point in space. *Philosophical Magazine*, 2(11):559–572.

Peel, D. and McLachlan, G. J. (2000). Robust mixture modelling using the t distribution. *Statistics and computing*, 10(4):339–348.

Peralta, B. and Soto, A. (2014). Embedded local feature selection within mixture of experts. *Information Sciences*, 269:176–187.

Perthame, E., Forbes, F., and Deleforge, A. (2018). Inverse regression approach to robust nonlinear high-to-low dimensional mapping. *Journal of Multivariate Analysis*, 163:1–14.

Quandt, R. E. (1972). A new approach to estimating switching regressions. *Journal of the American statistical association*, 67(338):306–310.

Raftery, A. E. and Dean, N. (2006). Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178.

Rao, C. R. (1948). The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203.

Rao, C. R. (1971). *Generalized inverse of matrices and its applications*. John Wiley & Sons, Inc.

Rosset, S., Zhu, J., and Hastie, T. (2004). Boosting as a regularized path to a maximum margin classifier. *Journal of Machine Learning Research*, 5(Aug):941–973.

Rubin, D. B. and Thayer, D. T. (1982). EM algorithms for ML factor analysis. *Psychometrika*, 47(1):69–76.

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

Schifano, E. D., Strawderman, R. L., Wells, M. T., et al. (2010). Majorization-minimization algorithms for nonsmoothly penalized objective functions. *Electronic Journal of Statistics*, 4:1258–1299.

Scholkopf, B. and Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

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

Snoussi, H. and Mohammad-Djafari, A. (2005). Degeneracy and likelihood penalization in multivariate gaussian mixture models. *Univ. of Technology of Troyes, Troyes, France, Tech. Rep. UTT*.

Städler, N., Bühlmann, P., and Van De Geer, S. (2010a). ℓ_1-penalization for mixture regression models. *Test*, 19(2):209–256.

Städler, N., Bühlmann, P., and Van De Geer, S. (2010b). $l1$-penalization for mixture regression models. *Test*, 19(2):209–256.

Stephens, M. et al. (2000). Bayesian analysis of mixture models with an unknown number of componentsan alternative to reversible jump methods. *The Annals of Statistics*, 28(1):40–74.

Stephens, M. and Phil, D. (1997). Bayesian methods for mixtures of normal distributions.

Tamayo, P., Scanfeld, D., Ebert, B. L., Gillette, M. A., Roberts, C. W., and Mesirov, J. P. (2007). Metagene projection for cross-platform, cross-species characterization of global transcriptional states. *Proceedings of the National Academy of Sciences*, 104(14):5959–5964.

Tang, Y., Xiang, L., and Zhu, Z. (2014). Risk factor selection in rate making: EM adaptive Lasso for zero-inflated poisson regression models. *Risk Analysis*, 34(6):1112–1127.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B*, pages 267–288.

Tibshirani, R. J. (2013). The Lasso problem and uniqueness. *Electronic Journal of Statistics*, 7:1456–1490.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622.

Titterington, D. M., Smith, A. F., and Makov, U. E. (1985). *Statistical analysis of finite mixture distributions*. Wiley, New York.

Tseng, P. (1988). Coordinate ascent for maximizing nondifferentiable concave functions. Technical report.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494.

Vapnik, V. (1998). *Statistical learning theory*. Wiley New York.

Wainwright, M. J. (2009). Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE transactions on information theory*, 55(5):2183–2202.

Wang, S. and Zhu, J. (2008). Variable selection for model-based high-dimensional clustering and its application to microarray data. *Biometrics*, 64(2):440–448.

Wang, Z., Ma, S., Wang, C.-Y., Zappitelli, M., Devarajan, P., and Parikh, C. (2014). EM for regularized zero-inflated regression models with applications to postoperative morbidity after cardiac surgery in children. *Statistics in medicine*, 33(29):5192–5208.

Wasserman, L. (2013). *All of statistics: a concise course in statistical inference*. Springer Science & Business Media.

Wedel, M. and DeSarbo, W. S. (1995). A mixture likelihood approach for generalized linear models. *Journal of Classification*, 12(1):21–55.

Weigend, A. S., Mangeas, M., and Srivastava, A. N. (1995). Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting. *International Journal of Neural Systems*,

6(04):373–399.

Witten, D. M. and Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726.

Wong, C. S. and Li, W. K. (2001). On a logistic mixture autoregressive model. *Biometrika*, 88(3):833–846.

Wu, C. J. (1983). On the convergence properties of the EM algorithm. *The Annals of statistics*, pages 95–103.

Wu, T. T. and Lange, K. (2008). Coordinate descent algorithms for Lasso penalized regression. *The Annals of Applied Statistics*, 2(1):224–244.

Xie, B., Pan, W., and Shen, X. (2008a). Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables. *Electronic journal of statistics*, 2:168–212.

Xie, B., Pan, W., and Shen, X. (2008b). Variable selection in penalized model-based clustering via regularization on grouped parameters. *Biometrics*, 64(3):921–930.

Xie, B., Pan, W., and Shen, X. (2010). Penalized mixtures of factor analyzers with application to clustering high-dimensional microarray data. *Bioinformatics*, 26(4):501–508.

Xu, L., Jordan, M. I., and Hinton, G. E. (1995). An alternative model for mixtures of experts. In *Advances in neural information processing systems*, pages 633–640.

Yuan, M. and Lin, Y. (2006a). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67.

Yuan, M. and Lin, Y. (2006b). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

Yuksel, S. E., W., J. N., and Gader, P. D. (2012). Twenty years of mixture of experts. *IEEE transactions on neural networks and learning systems*, 23(8):1177–1193.

Zeevi, A. J., Meir, R., and Adler, R. J. (1997). Time series prediction using mixtures of experts. In *Advances in neural information processing systems*, pages 309–318.

Zeevi, A. J., Meir, R., and Maiorov, V. (1998). Error bounds for functional approximation and estimation using mixtures of experts. *IEEE Transactions on Information Theory*, 44(3):1010–1025.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942.

Zhao, P. and Yu, B. (2006). On model selection consistency of Lasso. *Journal of Machine learning research*, 7(Nov):2541–2563.

Zhou, H., Pan, W., and Shen, X. (2009). Penalized model-based clustering with unconstrained covariance matrices. *Electronic journal of statistics*, 3:1473–1496.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal*

of the Royal Statistical Society: Series B (Statistical Methodology), 67(2):301–320.

List of Figures

List of Tables

Résumé long en français

Résumé court

L'analyse statistique de données hétérogènes de grande dimension pose certaines difficultés, tant du point de vue de la modélisation que du point de vue de l'estimation, en particulier dans le contexte actuel lié au phénomène des données massives. Cela suggère de nouvelles stratégies, en particulier pour les analyses avancées allant de l'estimation de densité à la prédiction, en passant par la classification non supervisée, de telles situations de données à distributions complexes. Les modèles de mélange se sont affirmés comme l'approche de choix dans la modélisation de données hétérogènes, dans de nombreux problèmes de l'approche statistique de la science des données, y compris en estimation de densité et en classification, et leur élégante extension des mélanges d'experts (MoE), renforce le lien avec l'apprentissage supervisé et traitent donc en outre de la prédiction à partir de données hétérogènes de type régression, ainsi que de la classification, supervisée ou non. Dans un scénario à grande dimension, en particulier pour les données provenant d'une population hétérogène, l'utilisation de tels modèles de mélanges d'experts suggère de répondre à des questions de modélisation et d'estimation, car les méthodes d'estimation de l'état de l'art sont limitées.

Cette thèse traite de la modélisation et de l'estimation de modèles de mélanges d'experts de grande dimension, en vue d'efficaces estima-

tion de densité, prédiction et classification de telles données complexes car hétérogènes et de grande dimension. Nous proposons de nouvelles stratégies basées sur l'estimation par maximum de vraisemblance régularisé des modèles pour pallier aux limites des méthodes standards, y compris l'EMV avec les algorithmes d'espérance-maximisation (EM), et pour effectuer simultanément la sélection des variables pertinentes afin d'encourager des solutions parcimonieuses dans un contexte à grande dimension. Nous introduisons d'abord une méthode d'estimation régularisée des paramètres et de sélection de variables d'un mélange d'experts, basée sur des régularisations ℓ_1 (lasso) et le cadre de l'algorithme EM, pour la régression et la classification adaptés aux contextes de la grande dimension. Ensuite, nous étendons la stratégie à un mélange régularisé de modèles d'experts pour les données discrètes, y compris pour la classification. Nous développons des algorithmes efficaces pour maximiser la fonction de log-vraisemblance ℓ_1-pénalisée des données observées. Nos stratégies proposées jouissent de la maximisation monotone efficace du critère optimisé, et contrairement aux approches précédentes, ne s'appuient pas sur des approximations des fonctions de pénalité, évitent l'inversion de matrices et exploitent l'efficacité de l'algorithme de montée de coordonnées, particulièrement dans l'approche proximale par montée de coordonnées.

Mots-clés Modèles de mélange; Mélange d'experts; Estimation régularisée; Sélection de variables; Lasso; Régularisation; Parcimonie; Algorithme EM; Algorithme MM; Proximal-Newton; Montée de coordonnées; Clustering; Classification; Régression; Prédiction.

Contexte scientifique

De nos jours, des données massives sont recueillies et exploitées dans presque tous les domaines de la science, de la technologie, et de l'industrie. Cela concerne de très nombreux objectifs allant de l'optimisation de la performance en entreprise à partir d'indicateurs métier et de données client, jusqu'à la médecine personnalisée, en passant par l'exploration et la visualisation optimisée de données hétérogènes de grande dimension, ou encore le développement de systèmes d'intelligence artificielle comme en recherche d'information ou en robotique.

Très souvent, ces masses de données sont hétérogènes, de grande dimension, non étiquetées, incomplètes, etc. Une telle complexité des données pose de nouveaux défis quant à leur traitement et leur analyse. Les analyses peuvent concerner la prédiction de données futures à partir de connaissances antérieures de celles-ci, via des méthodes d'apprentissage supervisé. Elles peuvent également être dans un but exploratoire afin de résumer les principales informations contenues dans un échantillon, révéler des informations utiles cachées, comme des groupes ou une hiérarchie de groupes de caractéristiques communes, à travers des méthodes d'apprentissage non-supervisé, comme des techniques optimisées de classification. Les analyses couvrent également l'identification de variables redondantes et/ou les plus pertinentes à la représentation des données, par des méthodologies de sélection de variables.

De tels défis imposent de nouvelles stratégies de modélisation, d'estimation et de sélection, et l'optimisation des algorithmes d'inférence et de sélection associés. Les principales méthodes réputées reposent soit sur des ystèmes d'apprentissage automatique de données tels que les réseaux neuronaux (Bishop, 1995), les machines à vecteurs de support (Vapnik, 1998), les

machines à noyaux (Scholkopf and Smola, 2001), et aussi les méthode d'apprentissage statistique de modèles génératifs (Friedman et al., 2001), en particulier les modèles à variables latentes.

L'analyse statistique des données qui suppose que celles-ci sont des réalisations de variables aléatoires dont il s'agira de déterminer les distributions les plus adaptées aux données observées, est tient ainsi compte de la partie aléatoire des données et mesure l'incertitude, fournit un cadre agréable pour l'estimation de densité, la prédiction et l'apprentissage non supervisé, y compris en clustering. Parmi les modèles statistiques on distingue les modèles de mélange de densités (Titterington et al., 1985; McLachlan and Peel., 2000) qui représentent le moyen commode pour l'analyse de données hétérogènes en statistique et en apprentissage machine, dans différents domaines d'application allant de la bioinformatique, à l'économie. Grâce à leur flexibilité, les modèles de mélange peuvent être utilisés pour l'estimation de densités et la classification de données provenant de populations hétérogènes complexes. Ils sont également utilisés pour effectuer des analyses de prédiction, y compris des analyses de régression, par le biais des mélanges d'experts Jacobs et al. (1991); Jordan and Jacobs (1994), l'une de leur extensions les plus flexibles. La question finale d'analyse sur la base de tels modèles, passe d'abord par l'ajustement de ces modèles aux données observées, c'est la question de l'estimation. L'estimation par maximum de vraisemblance (EMV) (Fisher, 1912) avec les algorithmes EM (Dempster et al., 1977; McLachlan and Krishnan, 2008) est le moyen courant d'estimer les paramètres des modèles de mélange et des modèles de mélange d'experts. Toutefois, l'application directe de ces méthodes à des ensembles de données de grande dimension présente certains inconvénients. Par exemple, il devient difficile d'estimer la matrice de covariance dans les modèles de mélange gaussiens avec un grand nom-

bre de variables, et c'est encore plus difficile dans un scénario à grande dimension. Cela devient en effet plus limitant avec l'algorithme EM dont le calcul de l'étape E nécessite l'inversion des matrices de covariance. Le même problème se pose lors de l'application de l'algorithme de Newton-Raphson (Boyd and Vandenberghe, 2004) dans l'étape M de l'algorithme EM, pour notamment les mélanges d'experts.

Dans un contexte de grande dimension, les variables peuvent être corrélées, et celles réellement nécessaires à l'analyse car intrinsèques au problème étudié, résident très souvent dans un espace de dimension réduite. Il est donc nécessaire de procéder à une estimation adaptée et de sélectionner le sous-ensemble des variables pertinentes qui représentent aux mieux les données. Le Lasso, ou régression linéaire ℓ_1-régularisée, introduite par Tibshirani (1996), est une méthode efficace en analyse statistique classique comme la régression sur données homogènes. Le Lasso présente l'avantage d'encourager des solutions parcimonieuses, n'ayant que quelques variables non nulles et la convexité du problème d'optimisation qui lui est associé simplifie le calcul algorithmique. Les méthodes dérivées dans cette thèse, s'appuie sur ce type de régularisation.

Méthodologie et objectifs

Dans cette thèse, nous nous concentrons sur les mélanges d'experts (MoE) y compris pour les modèles linéaires généralisés. Les mélanges d'experts vont au delà de l'estimation de densité et de la classification de données vectorielles, et fournissent un cadre adapté à l'estimation de densité, la classification automatique et la prédiction de données liées de type régression, ainsi que pour la classification supervisée de données hétérogènes (Jacobs et al., 1991; Yuksel et al., 2012; Jiang and Tanner, 1999a,b; Grün and Leisch, 2007). Les mélanges d'experts sont utilisés dans plusieurs applications

telles que: prédiction de la demande quotidienne d'électricité (Weigend et al., 1995), généralisation des modèles autorégressifs pour les séries temporelles (Zeevi et al., 1997; Wong and Li, 2001), reconnaissance de chiffres manuscrits (Hinton et al., 1995), segmentation et classification de séries temporelles avec changements de régime (Chamroukhi et al., 2009b; Samé et al., 2011; Chamroukhi and Nguyen, 2018), reconnaissance d'activités humaines (Chamroukhi et al., 2013), etc. Malheureusement, la modélisation à l'aide des modèles de mélanges d'experts dans le cas des prédicteurs de grande dimension reste encore limitée. Nos objectifs ici sont donc d'étudier l'estimation et la sélection des variables dans des modèles de mélanges d'experts de grande dimension, et de:

i) proposer de nouvelles stratégies d'estimation et de sélection de variables pour surmonter ces limites. Pour y parvenir, nous souhaitons introduire une nouvelle estimation régularisée et une sélection de variables dans le cadre de mélanges d'experts pour les modèles linéaires généralisés, basées sur des pénalités de type Lasso. Nous les étudions pour différentes familles d'experts, y compris le mélanges d'experts avec des distributions d'experts gaussiennes, de Poisson et logistiques.

ii)] développer des algorithmes efficaces pour estimer les paramètres de ces modèles régularisés. Les algorithmes développés effectuent simultanément l'estimation et la sélection de variables et devraient avoir la capacité d'encourager la parcimonie, et traiter certaines questions typiques du problème de grande dimension, en évitant l'inversion de matrice, de sorte qu'ils fonctionnent assez bien dans des situations de grande dimension.

Dans la section suivante, nous résumons les contributions de la thèse.

Contributions de la thèse

Le manuscrit est organisé comme suit.

Contribution du Chapitre 1

Le chapitre 2 est consacré à l'état de l'art. Le chapitre 3 présente notre première contribution à l'estimation et à la sélection de variables des modèles de mélanges d'experts, pour des données continues. Ensuite, le chapitre 4, présente notre deuxième contribution. Il porte sur l'estimation et la sélection des variables des modèles de mélanges d'experts, pour les données discrètes. Enfin, dans le chapitre 5, nous discutons de notre recherche, tirons des conclusions et soulignons quelques potentielles pistes pour l'avenir. Les détails techniques relatifs aux développements mathématiques de nos contributions sont fournis dans les annexes A, B et C.

Contribution du Chapitre 2

Plus précisément, tout d'abord, dans le chapitre 2, nous présentons une revue substantielle des modèles et algorithmes liés aux sujets scientifiques traités dans la thèse. Nous nous concentrons d'abord sur le cadre général des modèles de mélanges, en tant que choix approprié pour la modélisation de l'hétérogénéité des données. Nous décrivons les aspects de modélisation statistique et les stratégies d'estimation associées, ainsi que les techniques de sélection de modèles, en accordant une attention particulière à l'EMV via l'algorithme EM. Ensuite, nous revisitons ce contexte de modélisation par mélange de densités, dans le cadre de problèmes de régression sur des données hétérogènes, et présentons son extension au cadre des modèles de mélanges d'experts. Dans l'étape suivante, nous examinons ces modèles dans un contexte de grande dimension, y compris le cas d'experts gaussiens,

et décrivons les trois stratégies principales pour pallier au problème de la malédiction de la dimension, c'est-à-dire l'approche de réduction de dimension, celle de décomposition spectrale des matrices de covariance du modèle, et celle de sélection des variables par régularisation Lasso. Nous optons pour cette dernière stratégie de régularisation et la présentons dans le cas du mélange de modèles de régressions et du mélange d'experts. Nous passons en revue l'estimation et la sélection de variables par maximum de vraisemblance régularisé pour ces modèles au moyen d'algorithmes EM adaptés.

Contribution du Chapitre 3

Puis, dans le chapitre 3, nous introduisons une nouvelle approche pour l'estimation et la sélection de variables de mélanges d'experts pour la régression avec des prédicteurs potentiellement de grande dimension, et une population hétérogène. L'approche effectue simultanément l'estimation des paramètres, la sélection des variables, le clustering et la régression sur des données de régression hétérogènes. Il s'agit d'une approche de maximum de vraisemblance régularisé avec une régularisation dédiée qui, d'une part, encourage la parcimonie grâce à une partie de régularisation de type Lasso, et d'autre part, un calcul facilité grâce à la convexité de la pénalité ℓ_1. Nous proposons un cadre hybride efficace d'espérance-maximisation (EM) pour résoudre efficacement le problème d'optimisation qui en résulte, et qui maximise de façon monotone la log-vraisemblance régularisée. Il en résulte trois algorithmes hybrides pour maximiser la fonction objective proposée, i.e un algorithme de Majorisation-Maximisation (MM), une montée de coordonnées, et une procédure de type Newton proximal. Nous montrons que l'approche proposée ne nécessite pas d'approximation du terme de régularisation, et les trois algorithmes hybrides développés permettent de

sélectionner automatiquement des solutions parcimonieuses sans approximation sur les fonctions de pénalisation. Nous nous appuyons sur un critère de type BIC pour réaliser la tâche de sélection du modèle, y compris la sélection du nombre de composantes du mélange et les hyper-paramètres de régularisation. Une expérimentation est alors menée pour comparer l'approche proposée aux principales méthodes concurrentes en la matière. L'évaluation vise à évaluer la performance de l'approche en termes de classification, d'estimation de densité, de régression des données hétérogènes et de parcimonie des modèles de mélanges d'experts. Des expériences approfondies, tant sur des simulations que sur des données réelles, montrent que l'approche proposée surpasse ses concurrents et qu'elle est très encourageante pour s'attaquer au problème de la grande dimension. Ce chapitre a principalement mené à la publication de l'article de journal (Chamroukhi and Huynh, 2019) et à des articles de conférences.

Contribution du Chapitre 4

Ensuite, au chapitre 4, nous examinons une autre famille de modèles de mélanges d'experts, celle pour des données discrètes, y compris le mélange d'experts pour des données de comptage et le le mélange d'experts pour la classification, et présentons notre deuxième contribution principale. Nous présentons une nouvelle stratégie régularisée d'EMV pour l'estimation et la sélection de variables de mélanges spécifiques de modèles linéaires d'experts généralisés dans un environnement de grande dimension. Nous développons un algorithme EM efficace, qui s'appuie sur une approximation proximale de Newton, pour maximiser de façon monotone le critère de log-vraisemblance pénalisée proposé. La stratégie présentée effectue simultanément l'estimation des paramètres, la sélection des variables et la classification des données discrètes hétérogènes. Un avantage de la stratégie

de type Newton proximal introduite consiste dans le fait qu'il suffit de résoudre des problèmes de Lasso quadratique pondéré pour mettre à jour les paramètres. Des outils efficaces tels que l'algorithme de montée des coordonnées peuvent être utilisés pour résoudre ces problèmes. L'approche proposée ne nécessite pas non plus d'approximation du terme de régularisation et permet de sélectionner automatiquement des solutions parcimonieuses sans seuillage. Notre approche s'est avérée performante, y compris dans un environnement à grande dimension, et surpasse les modèles récents régularisés du mélange d'experts sur plusieurs expériences portant sur des données simulées et réelles. La principale publication liée à ce chapitre est Huynh and Chamroukhi (2019) ainsi que d'autres communications.

Contribution du Chapitre 5

Enfin, dans le chapitre 5, nous discutons de notre recherche développée, tirons quelques conclusions et ouvrons quelques orientations futures. Les résultats sous formes de publications de recherche de la thèse, y compris les codes sources libres R, sont présentés dans la liste des publications et communications à la fin du manuscrit.

Estimation et sélection de variables dans les modèles de mélange d'experts de grande dimension

Résumé: Cette thèse traite de la modélisation et de l'estimation de modèles de mélanges d'experts de grande dimension, en vue d'efficaces estimation de densité, prédiction et classification de telles données complexes car hétérogènes et de grande dimension. Nous proposons de nouvelles stratégies basées sur l'estimation par maximum de vraisemblance régularisé des modèles pour pallier aux limites des méthodes standards, y compris l'EMV avec les algorithmes d'espérance-maximisation (EM), et pour effectuer simultanément la sélection des variables pertinentes afin d'encourager des solutions parcimonieuses dans un contexte haute dimension. Nous introduisons d'abord une méthode d'estimation régularisée des paramètres et de sélection de variables d'un mélange d'experts, basée sur des régularisations ℓ_1 (lasso) et le cadre de l'algorithme EM, pour la régression et la classification adaptés aux contextes de la grande dimension. Ensuite, nous étendons la stratégie un mélange régularisé de modèles d'experts pour les données discrètes, y compris pour la classification. Nous développons des algorithmes efficaces pour maximiser la fonction de log-vraisemblance ℓ_1-pénalisée des données observées. Nos stratégies proposées jouissent de la maximisation monotone efficace du critère optimisé, et contrairement aux approches précédentes, ne s'appuient pas sur des approximations des fonctions de pénalité, évitent l'inversion de matrices et exploitent l'efficacité de l'algorithme de montée de coordonnées, particulièrement dans l'approche proximale par montée de coordonnées.

Mots-clés: Modèles de mélange; Mélange d'experts; Estimation régularisée; Sélection de variables; Lasso; Régularisation; Parcimonie; Algorithme EM; Algorithme MM; Proximal-Newton; Montée de coordonnées; Clustering; Classification; Régression; Prédiction.

***

Estimation and Feature Selection in High-Dimensional Mixtures-of-Experts Models

Abstract: This thesis deals with the problem of modeling and estimation of high-dimensional MoE models, towards effective density estimation, prediction and clustering of such heterogeneous and high-dimensional data. We propose new strategies based on regularized maximum-likelihood estimation (MLE) of MoE models to overcome the limitations of standard methods, including MLE estimation with Expectation-Maximization (EM) algorithms, and to simultaneously perform feature selection so that sparse models are encouraged in such a high-dimensional setting. We first introduce a mixture-of-experts' parameter estimation and variable selection methodology, based on ℓ_1 (lasso) regularizations and the EM framework, for regression and clustering suited to high-dimensional contexts. Then, we extend the method to regularized mixture of experts models for discrete data, including classification. We develop efficient algorithms to maximize the proposed ℓ_1-penalized observed-data log-likelihood function. Our proposed strategies enjoy the efficient monotone maximization of the optimized criterion, and unlike previous approaches, they do not rely on approximations on the penalty functions, avoid matrix inversion, and exploit the efficiency of the coordinate ascent algorithm, particularly within the proximal Newton-based approach.

Keywords: Mixture models; Mixture of Experts; Regularized Estimation; Feature Selection; Lasso; ℓ_1-regularization; Sparsity; EM algorithm; MM Algorithm; Proximal-Newton; Coordinate Ascent; Clustering; Classification; Regression; Prediction.