# Regularized Maximum-Likelihood Estimation of Mixture-of-Experts for Regression and Clustering

Faicel Chamroukhi, Bao Tuyen Huynh

# Regularized Maximum-Likelihood Estimation of Mixture-of-Experts for Regression and Clustering

Faicel Chamroukhi and Bao Tuyen Huynh

*Abstract*—**Mixture of experts (MoE) models are successful neural-network architectures for modeling heterogeneous data in many machine learning problems including regression, clustering and classification. The model learning is in general performed by maximum likelihood estimation (MLE). For high-dimensional data, a regularization is needed in order to avoid possible degeneracies or infeasibility of the MLE related to high-dimensional and possibly redundant and correlated features in a high-dimensional scenario. Regularized maximum likelihood estimation allows the selection of a relevant subset of features for prediction and thus encourages sparse solutions. The problem of variable selection is challenging in the modeling of heterogeneous data, including with MoE models. We consider the MoE for heterogeneous regression data and propose a regularized maximum-likelihood estimation with possibly high-dimensional features, based on a dedicated EM algorithm which integrates coordinate ascent updates of the parameters. Unlike state-of-the art regularized MLE for MoE, the proposed modeling does not require an approximate of the regularization. The proposed algorithm allows to automatically obtaining sparse solutions without thresholding, and includes coordinate ascent updates avoiding matrix inversion, and can thus be scalable. An experimental study shows the good performance of the algorithm in terms of recovering the actual sparse solutions, in parameter estimation, and in clustering of heterogeneous regression data.**

## I. Introduction

Mixture of experts (MoE) models are successful neural-network architectures for modeling heterogeneous data in many machine learning problems including regression, clustering and classification. They have been mostly studied Mixture-of-Experts (MoE) models introduced by [1] are widely used in statistics and machine learning. MoE 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 some input covariates. This makes MoE more capable in use than standard unconditional mixture distributions, while having a neural-network interpretation. A general review of the MoE models and their applications can be found in [2], [3]. For continuous data, which we consider here in the context of regression and clustering, MoE usually use Gaussian experts. While the MoE modeling with maximum likelihood inference is widely used, its application in high-dimensional problems is still challenging due to the known problem of the ML estimation (MLE) in such a setting, and hence there is a need to select a subset of the potentially large number of features, that really explain the problem. Indeed, in high-dimensional setting, the features can be correlated, present redundancy, etc, and thus the actual features that explain the problem lie in a low-dimensional space. This can be achieved by regularizing the objective function so that to encourage sparse solutions.

In related mixture models, including mixture of linear regressions (MLR), where the mixing proportions are constant, [4] proposed regularized ML inference, including MIXLASSO, MIXHARD and MIXSCAD and provided some asymptotic properties corresponding to these penalty functions. Another $L_1$ penalization for MLR models for high-dimensional data was proposed by [5] and an adaptive Lasso penalized estimator with oracle inequality which includes the setting $p \gg n$ was presented. [6] provided an $L_1$-oracle inequality by a Lasso estimator for finite mixture of Gaussian regression models. This result can be seen as a complementary result to [5], by studying the Lasso for its $L_1$-regularization properties rather than considering it as a variable selection procedure. This work was extended later in [7] by considering a mixture of multivariate Gaussian regression models. When the set of features can be seen as to be splitted into groups, [8] introduced the two types of penalty functions called MIXGL1 and MIXGL2 for MLR models, based on group Lasso. An MM algorithm [9] version for MLR with Lasso penalty can be found in [10]. Their method allows for an avoidance of matrix operations. In [11], the author extended his MLR regularisation to the MoE setting and provided a root-$n$ consistent and oracle properties for Lasso and SCAD penalties and developed an EM algorithm [12] for fitting the models. However, as we will discuss it in section III-A, this is based on approximated penalty function, and uses a Newton-Raphson in the updates, which requires matrix inversion.

In this paper, we consider MoE models with regularisation as in [11] and propose a regularised maximum-likelihood inference which doesn't require an approximate of the regularisation. We develop a hybrid EM and coordinate ascent algorithm for model fitting. The proposed algorithm allows to automatically select sparse solutions without thresholding, and includes coordinate ascent updates avoiding matrix inversion. The rest of this article is organized as follows. In Section II we present the regularised maximum-likelihood strategy or the MoE model and the proposed EM algorithm with coordinate ascent in section III-B. An experimental study on simulated and a real-data example are given Section IV. Finally, in Section V, we draw concluding remarks.

## II. Modeling with Mixture-of-Experts (MoE)

Let $((\boldsymbol{X}_1, \boldsymbol{Y}_1), \ldots, (\boldsymbol{X}_n, \boldsymbol{Y}_n))$ be a random sample of $n$ independently and identically distributed (i.i.d) pairs $(\boldsymbol{X}_i, \boldsymbol{Y}_i) \in \mathcal{X} \times \mathcal{Y}$, $(i = 1, \ldots, n)$ where $Y_i \in \mathcal{X} \subset \mathbb{R}^d$ is the $i$th response given some predictor vector $\boldsymbol{X}_i \in$

Faicel Chamroukhi and Bao Tuyen Huynh are with Normandie Univ, UNICAEN, CNRS, LMNO, 14000, Caen, France Contacts: {faicel.chamroukhi, bao-tuyen.huynh}@unicaen.fr

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

$\mathcal{X} \subset \mathbb{R}^p$. These data may be discrete or continuous. We consider the mixture of experts modeling framework for the analysis of a heteregeneous set of such data. Let $\mathcal{D} = ((\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n))$ be an observed data sample. The mixture of experts model assumes that the observed pairs $(\boldsymbol{x}, \boldsymbol{y})$ 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 gating network is defined by the distribution of the hidden variable $Z$ given the predictor $\boldsymbol{x}$, i.e., $\pi_k(\boldsymbol{x}; \boldsymbol{w}) = \mathbb{P}(Z = k|\boldsymbol{X} = \boldsymbol{x}; \boldsymbol{w})$, which is in general given by gating softmax functions of the form:

$$\begin{aligned}\pi_k(\boldsymbol{x}_i; \boldsymbol{w}) =& \mathbb{P}(Z_i = k|\boldsymbol{X}_i = \boldsymbol{x}_i; \boldsymbol{w}) \\ =& \frac{\exp(w_{k0} + \boldsymbol{x}_i^T \boldsymbol{w}_k)}{1 + \sum_{l=1}^{K-1} \exp(w_{l0} + \boldsymbol{x}_i^T \boldsymbol{w}_l)}\end{aligned} \quad (1)$$

for $k = 1, \ldots, K-1$ with $(w_{k0}, \boldsymbol{w}_k^T) \in \mathbb{R}^{p+1}$ and $(w_{K0}, \boldsymbol{w}_K^T) = (0, \boldsymbol{0})$ for identifiability [13]. The experts network is defined by the conditional densities $f(\boldsymbol{y}_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k)$ which is the short notation of $f(\boldsymbol{y}_i|\boldsymbol{X} = \boldsymbol{x}, Z_i = k; \boldsymbol{\theta})$. The MoE thus decomposes the probability density of the observed data as a convex sum of a finite experts weighted by a softmax gating network, and can be defined by the following semi-parametric probability density (or mass) function:

$$f(\boldsymbol{y}_i|\boldsymbol{x}_i; \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}_i; \boldsymbol{w}) f(\boldsymbol{y}_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k) \quad (2)$$

that is parameterized by the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{\nu_{\boldsymbol{\theta}}}$ ($\nu_{\boldsymbol{\theta}} \in \mathbb{N}$) defined by

$$\boldsymbol{\theta} = (\boldsymbol{w}_1^T, \ldots, \boldsymbol{w}_{K-1}^T, \boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_K^T)^T \quad (3)$$

where $\boldsymbol{\theta}_k$ ($k = 1, \ldots, K$) is the parameter vector of the $k$th expert.

The experts are chosen to sufficiently represent the data for each group $k$, for example tailored regressors explaining the response $\boldsymbol{y}$ by the predictor $\boldsymbol{x}$ for continuous data, or multinomial experts for discrete data. For example, MoE for non-asymmetric data [14] and robust MoE [15], [16], [17] have been introduced. For a complete account of MoE, types of gating networks and experts networks, the reader is refereed to [2].

The generative process of the data described before assumes the following hierarchical representation. First, given the predictor $\boldsymbol{x}_i$, the categorical variable $Z_i$ follows the multinomial distribution:

$$Z_i|\boldsymbol{x}_i \sim \text{Mult}(1; \pi_1(\boldsymbol{x}_i; \boldsymbol{w}), \ldots, \pi_K(\boldsymbol{x}_i; \boldsymbol{w})) \quad (4)$$

where each of the probabilities $\pi_{z_i}(\boldsymbol{x}_i; \boldsymbol{w}) = \mathbb{P}(Z_i = z_i|\boldsymbol{x}_i)$ is given by the multinomial logistic function (1). Then, conditional 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

$$\boldsymbol{Y}_i|Z_i = z_i, \boldsymbol{X}_i = \boldsymbol{x}_i \sim p(\boldsymbol{y}_i|\boldsymbol{x}_i; \boldsymbol{\theta}_{z_i}) \quad (5)$$

where $p(\boldsymbol{y}_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k) = p(\boldsymbol{y}_i|Z_i = z_i, \boldsymbol{X}_i = \boldsymbol{x}_i; \boldsymbol{\theta}_{z_i})$ is the probability density or the probability mass function of the expert $z_i$ depending on the nature of the data $(\boldsymbol{x}, \boldsymbol{y})$ within the group $z_i$. In the following, we consider MoE models for regression and clustering of continuous data.

### A. MoE for regression and clustering

Consider the case of univariate continuous outputs $Y_i$. A common choice to model the relationship between the input $\boldsymbol{x}$ and the output $Y$ is by considering regression functions. Thus, within each homogeneous group $Z_i = z_i$, the response $Y_i$, given the expert $k$, is modeled by the following noisy linear model:

$$Y_i = \beta_{z_i 0} + \boldsymbol{\beta}_{z_i}^T \boldsymbol{x}_i + \sigma_{z_i} \varepsilon_i, \quad (6)$$

where the $\varepsilon_i$ 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 case, (6) $Y$ is equivalent to

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

### B. Maximum likelihood parameter estimation

Assume that, $\mathcal{D} = ((\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n))$ is an observed data sample generated from the MoE (2) with unknown parameter $\boldsymbol{\theta}$. The parameter vector $\boldsymbol{\theta}$ is commonly estimated by maximizing the observed data log-likelihood

$$\log L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k(\boldsymbol{x}_i; \boldsymbol{w}) f(\boldsymbol{y}_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k) \quad (7)$$

by using the EM algorithm [12], [1] which allows to iteratively find an appropriate local maximizer of the log-likelihood function. In the considered model for Gaussian regression, the maximized log-likelihood is given by

$$\log L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \Big[ \sum_{k=1}^{K} \pi_k(\boldsymbol{x}_i; \boldsymbol{w}) \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{\beta}_k^T \boldsymbol{x}_i, \sigma_k^2) \Big]. \quad (8)$$

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

## III. Regularised MoE modeling (RMoE)

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 expert parameters and does not require any approximation, along with performing the maximization by coordinate ascent, so that to avoid matrix inversion.

### A. *Regularised maximum-likelihood estimation of the MoE*

The proposed regularization combines a Lasso penalty for the experts parameters, and an Elastic-Net like penalty for the gating network, defined by:

$$PL(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) - \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1 - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 - \frac{\rho}{2} \sum_{k=1}^{K-1} \|\boldsymbol{w}_k\|_2^2. \tag{9}$$

A similar strategy were proposed in [11] where the author proposed a regularized ML function like (9) but which is then approximated in the model inference algorithm. The devoloped EM algorithm for fitting the model follows indeed the suggestion of [18] to approximate the penalty function in a some neighbourhood by a local quadratic function. Therefore, the Newton-Raphson method could be used to update parameters in the M-step. The weakness of this design 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, [11] replaced that approximation by an $\epsilon$-local quadratic function. Unfortunately, these strategies have some drawbacks. First, by approximating the penalty functions with ($\epsilon$-)quadratic functions, almost surely 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, it cannot guarantee the non-decreasing property of the EM algorithm of the penalized objective function. Thus, the convergence of the EM algorithm cannot be ensured. Finally, one has to choose $\epsilon$, which becomes an additional tuning parameter in practice. Our propoal gives and answer to overcome these limitations.

### B. *Parameter estimation with a block-wise EM algorithm*

We propose a block-wise EM algorithm, which integrates a coordinate ascent algorithm for updating the model parameters, to monotonically find local maximizers of (9). More specifically, in the M-step of our method, we propose using coordinate ascent algorithm to update $\boldsymbol{w}$ and the $\boldsymbol{\beta}$' parameters. The EM algorithm for the maximization of (9) firstly requires the construction of, in this case, the penalized complete-data log-likelihood

$$\log PL_c(\boldsymbol{\theta}) = \log L_c(\boldsymbol{\theta}) - \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1 - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 - \frac{\rho}{2} \sum_{k=1}^{K-1} \|\boldsymbol{w}_k\|_2^2 \tag{10}$$

where

$$\log L_c(\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{k=1}^{K} Z_{ik} \log\left[\pi_k(\boldsymbol{x}_i; \boldsymbol{w}) f(\boldsymbol{y}_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k)\right] \tag{11}$$

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 $i$th pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is generated from the $k$th 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).

*1) E-step:* The E-Step computes the conditional expectation of the penalized complete-data log-likelihood (10), given the observed data $\mathcal{D}$ and a current parameter vector $\boldsymbol{\theta}^{(s)}$:

$$\begin{aligned} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(s)}) =& \mathbb{E}\left[\log PL_c(\boldsymbol{\theta}) | \mathcal{D}; \boldsymbol{\theta}^{(s)}\right] \\ =& \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{(s)} \log\left[\pi_k(\boldsymbol{x}_i; \boldsymbol{w}) f_k(\boldsymbol{y}_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k)\right] \\ & - \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1 - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 - \frac{\rho}{2} \sum_{k=1}^{K-1} \|\boldsymbol{w}_k\|_2^2 \end{aligned} \tag{12}$$

where

$$\begin{aligned} \tau_{ik}^{(s)} =& \mathbb{P}(Z_i = k | \boldsymbol{y}_i, \boldsymbol{x}_i; \boldsymbol{\theta}^{(s)}) \\ =& \frac{\pi_k(\boldsymbol{x}_i; \boldsymbol{w}^{(s)}) f(\boldsymbol{y}_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k^{(s)})}{\sum_{l=1}^{K} \pi_l(\boldsymbol{x}_i; \boldsymbol{w}^{(s)}) f(\boldsymbol{y}_i|\boldsymbol{x}_i; \boldsymbol{\theta}_l^{(s)})} \end{aligned} \tag{13}$$

is the posterior probability that the data pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ is generated by the $k$th expert, with

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

This step therefore only requires the computation of the posterior component memberships $\tau_{ik}^{(s)}$ $(i = 1, \ldots, n)$ for each of the $K$ experts.

*2) M-step:* The M-Step updates the parameters by maximizing the $Q$ function (12), which can be written as

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(s)}) = Q(\boldsymbol{w}; \boldsymbol{\theta}^{(s)}) + Q(\boldsymbol{\beta}, \sigma; \boldsymbol{\theta}^{(s)}) \tag{14}$$

with

$$\begin{aligned} Q(\boldsymbol{w}; \boldsymbol{\theta}^{(s)}) =& \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{(s)} \log \pi_k(\boldsymbol{x}_i; \boldsymbol{w}) \\ & - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 - \frac{\rho}{2} \sum_{k=1}^{K-1} \|\boldsymbol{w}_k\|_2^2, \end{aligned} \tag{15}$$

and

$$\begin{aligned} Q(\boldsymbol{\beta}, \sigma; \boldsymbol{\theta}^{(s)}) =& \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik}^{(s)} \log \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{x}_i^{\top} \boldsymbol{\beta}_k, \sigma_k^2) \\ & - \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1. \end{aligned} \tag{16}$$

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

$$\begin{aligned} Q(\boldsymbol{w}; \boldsymbol{\theta}^{(s)}) =& \sum_{i=1}^{n} \sum_{k=1}^{K-1} \tau_{ik}^{(s)} (w_{k0} + \boldsymbol{x}_i^T \boldsymbol{w}_k) - \sum_{i=1}^{n} \log\Big[1 + \sum_{k=1}^{K-1} e^{w_{k0} + \boldsymbol{x}_i^T \boldsymbol{w}_k}\Big] \\ & - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 - \frac{\rho}{2} \sum_{k=1}^{K-1} \|\boldsymbol{w}_k\|_2^2. \end{aligned} \tag{17}$$

*3) Coordinate ascent algorithm for solving the M-Step:* For that, we use a coordinate ascent algorithm. Indeed, based on [19], [20] with regularity conditions, then the coordinate ascent algorithm is successful in updating $\boldsymbol{w}$. Thus, the $\boldsymbol{w}$ 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. The update of $w_{kj}$ is performed by maximizing

$$Q(w_{kj}; \boldsymbol{\theta}^{(s)}) = F(w_{kj}; \boldsymbol{\theta}^{(s)}) - \gamma_k |w_{kj}|, \tag{18}$$

where

$$F(w_{kj};\boldsymbol{\theta}^{(s)})=\sum_{i=1}^{n}\tau_{ik}^{(s)}(w_{k0}+\boldsymbol{w}_k^T\boldsymbol{x}_i)-\sum_{i=1}^{n}\log\Big[1+\sum_{l=1}^{K-1}e^{w_{l0}+\boldsymbol{w}_l^T\boldsymbol{x}_i}\Big] -\frac{\rho}{2}w_{kj}^2. \qquad (19)$$

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

$$G^{(s)}(w_{kj}|\boldsymbol{w}^m)=\begin{cases}F(w_{kj};\boldsymbol{\theta}^{(s)})-\gamma_k w_{kj} & ,w_{kj}>0\\ F(0;\boldsymbol{\theta}^{(s)}) & ,w_{kj}=0\\ F(w_{kj};\boldsymbol{\theta}^{(s)})+\gamma_k w_{kj} & ,w_{kj}<0\end{cases}.$$

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

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

where $m$ is denoted for the $m^{th}$ loop of the coordinate ascent algorithm. In fact, updating the estimation of $w_{kj}$ in Newton-Raphson loop with initial value $w_{kj}^{(0)}=w_{kj}^m$ as follows

$$w_{kj}^{(t+1)}=w_{kj}^{(t)}-\frac{\partial Q(w_{kj};\boldsymbol{\theta}^{(s)})}{\partial w_{kj}}\Big|_{w_{kj}^{(t)}}\Big(\frac{\partial^2 Q(w_{kj};\boldsymbol{\theta}^{(s)})}{\partial^2 w_{kj}}\Big)^{-1}\Big|_{w_{kj}^{(t)}}, \qquad (20)$$

where

$$\frac{\partial Q(w_{kj};\boldsymbol{\theta}^{(s)})}{\partial w_{kj}}=\begin{cases}U(w_{kj})-\gamma_k & ,G^{(s)}(w_{kj}|\boldsymbol{w}^m)=F(w_{kj};\boldsymbol{\theta}^{(s)})-\gamma_k w_{kj}\\ U(w_{kj})+\gamma_k & ,G^{(s)}(w_{kj}|\boldsymbol{w}^m)=F(w_{kj};\boldsymbol{\theta}^{(s)})+\gamma_k w_{kj}\end{cases}, \qquad (21)$$

with

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

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.

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

For other parameter we set $w_{lh}^{m+1}=w_{lh}^m$.
Similarity, for $w_{k0}$ a univariate Newton-Raphson algorithm with initial value $w_{k0}^0=w_{k0}^{(s)}$ can be used to update $w_{k0}^m$ by

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

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

$$Q(w_{k0};\boldsymbol{\theta}^{(s)})=\sum_{i=1}^{n}\tau_{ik}^{(s)}(w_{k0}+\boldsymbol{x}_i^T\boldsymbol{w}_k)-\sum_{i=1}^{n}\log\Big[1+\sum_{l=1}^{K-1}e^{w_{l0}+\boldsymbol{x}_i^T\boldsymbol{w}_l}\Big], \qquad (22)$$

with

$$\frac{\partial Q(w_{k0};\boldsymbol{\theta}^{(s)})}{\partial w_{k0}}=\sum_{i=1}^{n}\tau_{ik}^{(s)}-\sum_{i=1}^{n}\frac{e^{w_{k0}+\boldsymbol{x}_i^T\boldsymbol{w}_k}}{C_i(w_{k0})} \qquad (23)$$

and

$$\frac{\partial^2 Q(w_{k0};\boldsymbol{\theta}^{(s)})}{\partial^2 w_{k0}}=-\sum_{i=1}^{n}\frac{e^{w_k 0+x_i^T\boldsymbol{w}_k}(C_i(w_{k0})-e^{w_{k0}+x_i^T\boldsymbol{w}_k})}{C_i^2(w_{k0})}. \qquad (24)$$

The other parameters are fixed while updating $w_{k0}$.
Next, we fix $\sigma_k$, and update $\beta_{kj}$ in

$$Q(\boldsymbol{\beta},\sigma;\boldsymbol{\theta}^{(s)})=\sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(s)}\log\mathcal{N}(y_i;\beta_{k0}+\boldsymbol{\beta}_k^T\boldsymbol{x}_i,\sigma_k^2) -\sum_{k=1}^{K}\lambda_k\|\boldsymbol{\beta}_k\|_1; \qquad (25)$$

using a coordinate ascent algorithm, with initial values $(\beta_{k0}^0,\boldsymbol{\beta}_k^0)=(\beta_{k0}^{(s)},\boldsymbol{\beta}_k^{(s)})$. We obtain closed-form coordinate updates which can be computed for each component following the results in [21, sec. 5.4] and are given by

$$\beta_{kj}^{m+1}=\frac{\mathcal{S}_{\lambda_k\sigma_k^{(s)2}}\big(\sum_{i=1}^{n}\tau_{ik}^{(s)}r_{ikj}^m x_{ij}\big)}{\sum_{i=1}^{n}\tau_{ik}^{(s)}x_{ij}^2}, \qquad (26)$$

with $r_{ikj}^m=y_i-\beta_{k0}^m-\boldsymbol{\beta}_k^T\boldsymbol{x}_i^m+\beta_{kj}^m x_{ij}$ 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 $\beta_{kh}^{m+1}=\beta_{kh}^m$. At each iteration $m$, $\beta_{k0}$ is updated by

$$\beta_{k0}^{m+1}=\frac{\sum_{i=1}^{n}\tau_{ik}^{(s)}(y_i-\boldsymbol{\beta}_k^T\boldsymbol{x}_i^{m+1})}{\sum_{i=1}^{n}\tau_{ik}^{(s)}}. \qquad (27)$$

In the next step, we take $(w_{k0}^{(s+2)},\boldsymbol{w}_k^{(s+2)})=(w_{k0}^{(s+1)},\boldsymbol{w}_k^{(s+1)})$, $(\beta_{k0}^{(s+2)},\boldsymbol{\beta}_k^{(s+2)})=(\beta_{k0}^{(s+1)},\boldsymbol{\beta}_k^{(s+1)})$, rerun the E-step, and update $\sigma_k^2$ as follows

$$\sigma_k^{2(s+2)}=\frac{\sum_{i=1}^{n}\tau_{ik}^{(s+1)}(y_i-\beta_{k0}^{(s+2)}-\boldsymbol{\beta}_k^{(s+2)^T}\boldsymbol{x}_i)^2}{\sum_{i=1}^{n}\tau_{ik}^{(s+1)}}. \qquad (28)$$

The algorithm is iterated until the change in $PL(\boldsymbol{\theta})$ is small enough. The proposed algorithm, at each iteration, clearly guarantees to improve the optimised penalised log-likelihood function (9); Also we can directly get zero coefficients without any thresholding like in [11], [22].

### C. Algorithm tuning and model selection

In practical, appropriate values of the turning parameters $(\lambda,\gamma,\rho)$ should be chosen. To select the turning parameters , we use a modified BIC (Bayesian information criterion) with a grid search scheme. First, assume that $K_0\in\{K_1,\ldots,K_M\}$ whereupon $K_0$ is the true number of components. For each value of $K$, we choose a grid of tuning parameters. Consider grids of values $\{\lambda_1,\ldots,\lambda_{M_1}\}$, $\{\gamma_1,\ldots,\gamma_{M_2}\}$ scaled by $\sqrt{n}$ and $\rho\approx O(\log n)$. In practical, the value $\rho=0.1\log n$ is used for the ridge turning parameter in the simulations. For a given triad $(K,\lambda_i,\gamma_j)$, we obtain the maximal penalized log-likelihood estimators $\widehat{\boldsymbol{\theta}}_{K,\lambda,\gamma}$ using our EM algorithm presented about, then compute the modified BIC criterion

$$\text{BIC}(K,\lambda,\gamma)=-2L(\widehat{\boldsymbol{\theta}}_{K,\lambda,\gamma})+DF(\lambda,\gamma)\log n, \qquad (29)$$

where $DF(\lambda,\gamma)$ is the number of non-zero coefficients in the model. The BIC rule for tuning parameters selection is to set $(K,\lambda,\gamma)=(\tilde{K},\tilde{\lambda},\tilde{\gamma})$ which minimizes the BIC value. A criterion for choosing an optimal values of the tuning parameters for penalized MoE model is still an open research, however the modified BIC performs reasonably well in our simulation.

### *D. Statistical inference*

We study the asymptotic properties of our penalized MoE. Mainly, these results come from [11]. Our simulation study confirms these properties. First, let $\boldsymbol{V}_i = (\boldsymbol{X}_i, Y_i),\ i = 1, \ldots, n$ be a random sample from a density function $f(\boldsymbol{v}; \boldsymbol{\theta})$. Give an appropriate density function $f(\boldsymbol{x}_i)$, we can write the joint density of $\boldsymbol{V}_i$ as

$$f(\boldsymbol{v}_i; \boldsymbol{\theta}) = f(\boldsymbol{x}_i) \sum_{k=1}^{K} \pi_k(\boldsymbol{x}_i; \boldsymbol{w}) p(\boldsymbol{y}_i | \boldsymbol{x}_i; \boldsymbol{\theta}_k).$$

Assume that $\boldsymbol{\theta}_0$ is the true value of the population parameter. Considereing Theorem 2 of [11], we can state the following for the proposed regularized model and the proposed hybrid EM-Coordinate ascent algorithm. Assume that $f(\boldsymbol{v}; \boldsymbol{\theta})$ satisfies some regularity conditions (see conditions $R_1 - R_5$ in [11]). Let $(\boldsymbol{X}_i, Y_i),\ i = 1, \ldots, n$ be a random sample from this density function, where $f(\boldsymbol{x}_i)$ is well-behaved and $\rho/\sqrt{n} \to 0$ as $n \to \infty$. Then, there exists a local maximizer $\widehat{\boldsymbol{\theta}}_n$ of the regularized log-likelihood function $PL(\boldsymbol{\theta})$ for which

$$\|\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0\| = O\{n^{-1/2}(1 + q_{1n}^* + q_{1n})\},$$

where

$$q_{1n}^* = \max_{k,j}\{\lambda_k/\sqrt{n} : \beta_{kj}^0 \neq 0\};\ q_{1n} = \max_{k,j}\{\gamma_k/\sqrt{n} : w_{kj}^0 \neq 0\}.$$

The estimator $\widehat{\boldsymbol{\theta}}_n$ is root-$n$ consistent since $q_{1n}^* = O(1)$ and $q_{1n} = O(1)$. Unfortunately, we cannot select sequences $\lambda_k$ and $\gamma_k$ such that $\widehat{\boldsymbol{\theta}}$ is both sparse and root-$n$ consistent simultaneously since the Lasso penalized functions do not satisfy condition $\mathcal{C}_2$ in Theorem 3, [11]. That means we can choose $\lambda_k$ and $\gamma_k$ to attain consistence in feature selection but it also causes bias to the estimators of the true nonzero coefficients.

## IV. Experimental study

We study the performance of our method for both simulated data and real data in this section. Our result is compared with the non-penalized MoE, the MoE with $L_2$ regularisation and the mixture of linear regressions with Lasso penalty (MIXLASSO) in several criteria including the sparsity, parameters estimation and clustering criteria.

### *A. Simulation study*

In this section, a simulation was performed to assess the sample performance of the regularization MoE. Covariate variables $\{\boldsymbol{x}_i, i = 1, \ldots, n\}$ were generated from a multivariate Gaussian distribution with mean zero and correlation structure $\text{corr}(x_{ij}, x_{ij'}) = 0.5^{|j-j'|}$. After that, the response $Y$ is generated from a normal MoE model with $K = 2$, $p = 6$ and $n = 300$. The parameter $\sigma = 1$ is treated as unknown, other regression coefficients for this simulation is given as following:

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

100 data sets were generated for this simulation. For the results, we evaluate the performance of the penalized MoE compare with MoE with ridge penalty function for the gate, nonpenalized MoE and MIXLASSO (see [4]) in three different criteria: *sensitivity/specificity*, *parameters estimation* and *clustering*. Here, the *sensitivity/specificity* is defined by

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

In our simulations, the proportion of correctly estimated zero coefficients and nonzero coefficients was calculated for each data set for expert's parameters and gating's parameters and we present the average proportion of these criteria in Table I. 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 gating parameters $\boldsymbol{w}_k^{per}$ with $\boldsymbol{w}_k^{per} - \boldsymbol{w}_K^{per}$. By doing so, we can ensure that the log-likelihood will not change, that means $L(\hat{\boldsymbol{\theta}}) = L(\hat{\boldsymbol{\theta}}^{per})$ and these parameters satisfy initialized condition $\boldsymbol{w}_K^{per} = \mathbf{0}$. Unfortunately, the penalized log-likelihood value can be different from the old value. This also affects to the sparsity property of the model when we permute the parameters. However, for $K = 2$ both log-likelihood function and penalized log-likelihood function will not change since $\boldsymbol{w}_1^{per} = -\boldsymbol{w}_1$.

For the second criterion, we compute the mean and standard deviation of both penalized parameters and non-penalized parameters compare with the true value $\boldsymbol{\theta}$. We also consider the mean square error (MSE) between each component of the true parameter vector and the estimated one, which is given by $\|\theta_j - \hat{\theta}_j\|_2^2$. The square errors are averaged on 100 trials.

For *clustering* criterion, once the parameters are estimated and permuted, the provided posterior component memberships $\hat{\tau}_{ik}$ defined in (13) represent a soft partition of the data. A hard partition of the data is given by applying the optimal Bayes's rule

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

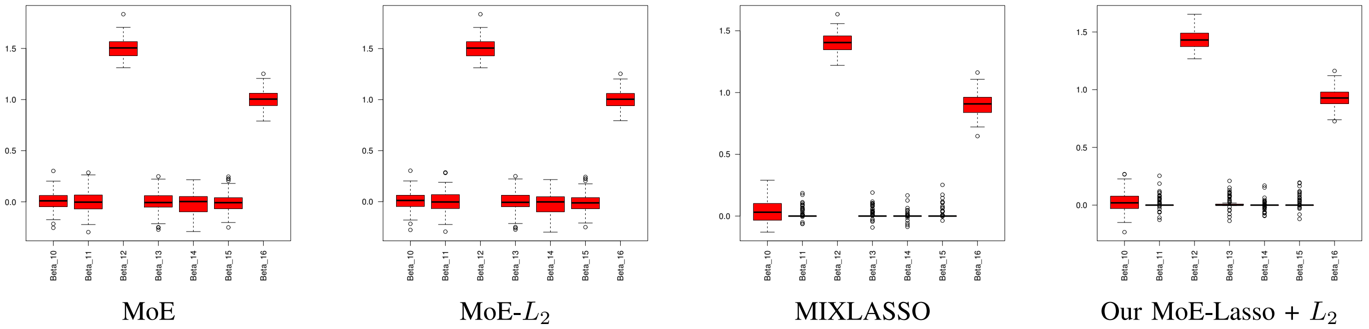
where $\hat{z}_i$ represents the estimated cluster label for the $i$th observation. We therefore compare the average ratio of true cluster labels of all observations by considering four models.

*1) Sensitivity/specificity criteria:* Table I contains the sensitivity ($S_1$) and specificity ($S_2$) values for the experts 1 and 2 of the model and also the gate. This criteria is computed for three models: Lasso+$L_2$, $L_2$ and MoE. Actually, the $L_2$ and MoE models cannot be considered as model selection methods since their sensitivity criterion almost surely equal zero. From Table I, one can clearly see that the Lasso+$L_2$ performs quite well in terms of experts 1 and 2. Feature selection becomes more difficult for the gate $\pi_k(\boldsymbol{x}; \boldsymbol{w})$ since there are correlations between features. However, the Lasso+$L_2$ performs reasonably well in term of detecting zero coefficient both in the experts and the gating network although it can fail in model selection for the gating network, meaning that, it also shrinks the non-zero values toward zero. The MIXLASSO, in some sense can detect the parameters in the experts. However, it will show that, this model have a poor result when clustering the data.

| Method | Expert 1 | | Expert 2 | | Gate | |
|---|---|---|---|---|---|---|
| | $S_1$ | $S_2$ | $S_1$ | $S_2$ | $S_1$ | $S_2$ |
| MoE | 0.0000 | 1.0000 | 0.0000 | 1.0000 | 0.0000 | 1.0000 |
| $L_2$ | 0.0000 | 1.0000 | 0.0000 | 1.0000 | 0.0000 | 1.0000 |
| Lasso+$L_2$ | 0.7000 | 1.0000 | 0.8033 | 1.0000 | 0.8525 | 0.9450 |
| MIXLASSO | 0.7750 | 1.0000 | 0.6933 | 1.0000 | N/A | N/A |

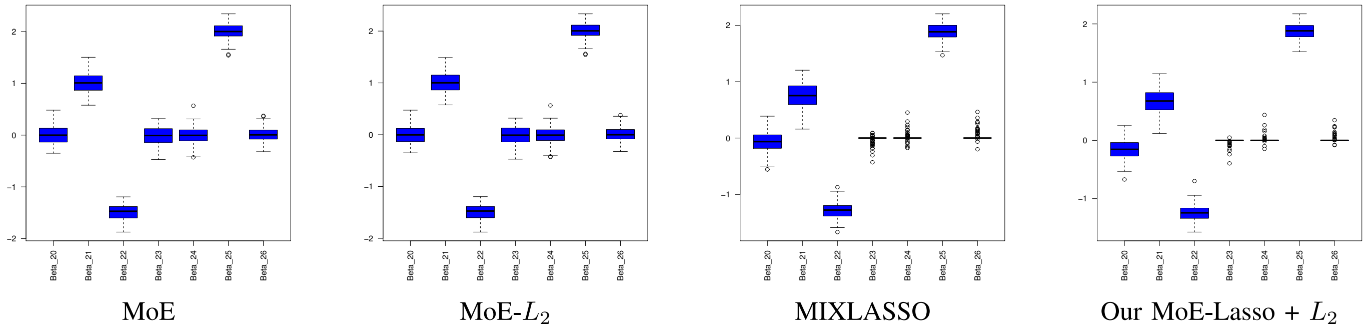
TABLE I. Sensitivity ($S_1$) and specificity ($S_2$) summaries.

*2) Parameter estimation:* The box plots of all estimated parameters are given in Figure 1, 2 and 3. For the mean and standard derivation assess, Table II shows that, the non penalized MoE and the MoE with $L_2$ penalized have a best result while $L_2$+Lasso and MIXLASSO can cause bias to the estimated parameters since the penalty functions are added in the log-likelihood function. However, from Table III in term of average mean square error, the $L_2$+Lasso and MIXLASSO provide a best result for the zero coefficients.



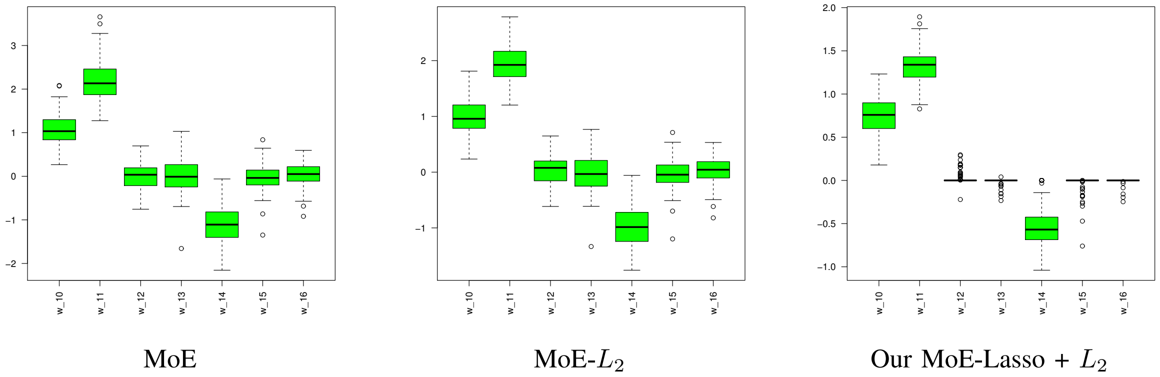
MoE MoE-$L_2$ MIXLASSO Our MoE-Lasso + $L_2$

Fig. 1. Boxplots of the expert 1's parameter $(\beta_{10}, \boldsymbol{\beta}_1)^T = (0, 0, 1.5, 0, 0, 0, 1)^T$.



MoE MoE-$L_2$ MIXLASSO Our MoE-Lasso + $L_2$

Fig. 2. Boxplots of the expert 2's parameter $(\beta_{20}, \boldsymbol{\beta}_2)^T = (0, 1, -1.5, 0, 0, 2, 0)^T$.



MoE MoE-$L_2$ Our MoE-Lasso + $L_2$

Fig. 3. Boxplots of the gate's parameter $(w_{10}, \boldsymbol{w}_1)^T = (1, 2, 0, 0, -1, 0, 0)^T$.

*3) Clustering:* We calculate the accuracy of clustering of four models for each data set. First, we permute the parameters corresponding to the true parameter and then we compute the ratio of true estimated cluster label for every observation in each data set and taking the average. The results in terms of Adjusted rand index (ARI) values are provided by Table IV. In this example, we see that the Lasso +$L_2$ model provides a good model for clustering data. $L_2$ model gives the best result. The difference between these models is smaller than $2\%$, while the MIXLASSO provides a poor result in term of clustering. Overall, we can clearly see the algorithm performs very well to retrieve the actual sparse support; the sensitivity and specificity results are better for the proposed Lasso+$L_2$ regularization. The MIXLASSO can identified the parameters in the regression components, however, this model gives a worst results while clustering the observation into clusters. The specificity and the specificity for the gating function of the proposed model is quite well. But the penalty function will cause bias to the parameters. This result can be observed for the MSE which means that the algorithm can also perform density estimation with a reasonable loss of information due to the bias induced by the regularization. In term of clustering, the Lasso+$L_2$ works as good as two other MoE models and better than the MIXLASSO model.

## B. *Applications to real data*

We now analyze a real data set consisting of baseball salaries from the Journal of Statistics Education (see also [4]) as a further test of the methodology. We compare our results with the non-penalized MoE models and the MIXLASSO models (see [4]) in different criteria: the average mean square error (MSE) between observation values of the response variable and the predicted values of this variable; we also consider the correlation of these values.

| Component | True value | MoE | $L_2$ | Lasso+$L_2$ | MIXLASSO |
|---|---|---|---|---|---|
| Expert 1 | $0$ | $0.010_{(.096)}$ | $0.009_{(.097)}$ | $0.026_{(.089)}$ | $0.043_{(.093)}$ |
| | $0$ | $-0.002_{(.106)}$ | $-0.002_{(.107)}$ | $0.011_{(.046)}$ | $0.011_{(.036)}$ |
| | $1.5$ | $1.501_{(.099)}$ | $1.502_{(.099)}$ | $1.435_{(.080)}$ | $1.404_{(.086)}$ |
| | $0$ | $0.000_{(.099)}$ | $0.001_{(.099)}$ | $0.013_{(.044)}$ | $0.013_{(.036)}$ |
| | $0$ | $-0.022_{(.102)}$ | $-0.022_{(.102)}$ | $0.000_{(.032)}$ | $0.003_{(.027)}$ |
| | $0$ | $-0.001_{(.097)}$ | $-0.003_{(.097)}$ | $0.012_{(.043)}$ | $0.013_{(.040)}$ |
| | $1$ | $1.003_{(.090)}$ | $1.004_{(.090)}$ | $0.930_{(.082)}$ | $0.903_{(.088)}$ |
| Expert 2 | $0$ | $0.006_{(.185)}$ | $0.005_{(.184)}$ | $-0.162_{(.177)}$ | $-0.063_{(.188)}$ |
| | $1$ | $1.007_{(.188)}$ | $1.006_{(.188)}$ | $0.675_{(.202)}$ | $0.755_{(.220)}$ |
| | $-1.5$ | $-1.492_{(.149)}$ | $-1.494_{(.149)}$ | $-1.242_{(.139)}$ | $-1.285_{(.146)}$ |
| | $0$ | $-0.011_{(.159)}$ | $-0.012_{(.158)}$ | $-0.018_{(.055)}$ | $-0.023_{(.071)}$ |
| | $0$ | $-0.010_{(.172)}$ | $-0.008_{(.171)}$ | $0.011_{(.059)}$ | $0.016_{(.075)}$ |
| | $2$ | $2.004_{(.169)}$ | $2.005_{(.169)}$ | $1.876_{(.149)}$ | $1.891_{(.159)}$ |
| | $0$ | $0.008_{(.139)}$ | $0.007_{(.140)}$ | $0.020_{(.060)}$ | $0.031_{(.086)}$ |
| Gating | $1$ | $1.095_{(.359)}$ | $1.008_{(.306)}$ | $0.759_{(.221)}$ | N/A |
| | $2$ | $2.186_{(.480)}$ | $1.935_{(.344)}$ | $1.332_{(.208)}$ | |
| | $0$ | $0.007_{(.287)}$ | $0.038_{(.250)}$ | $0.024_{(.068)}$ | |
| | $0$ | $-0.001_{(.383)}$ | $-0.031_{(.222)}$ | $-0.011_{(.039)}$ | |
| | $-1$ | $-1.131_{(.413)}$ | $-0.991_{(.336)}$ | $-0.526_{(.253)}$ | |
| | $0$ | $-0.022_{(.331)}$ | $-0.033_{(.281)}$ | $-0.032_{(.104)}$ | |
| | $0$ | $0.025_{(.283)}$ | $0.016_{(.246)}$ | $-0.007_{(.036)}$ | |
| $\sigma$ | $1$ | $0.965_{(.045)}$ | $0.961_{(.045)}$ | $0.989_{(.050)}$ | $1.000_{(.053)}$ |

TABLE II. MEAN AND STANDARD DERIVATION OF EACH COMPONENT.

| Component | True value | Mean square error | | | |
|---|---|---|---|---|---|
| | | MoE | $L_2$ | Lasso+$L_2$ | MIXLASSO |
| Expert 1 | $0$ | $0.0093_{(.015)}$ | $0.0094_{(.015)}$ | $\mathbf{0.0087}_{(\mathbf{.014})}$ | $0.0106_{(.016)}$ |
| | $0$ | $0.0112_{(.016)}$ | $0.0114_{(.017)}$ | $0.0022_{(.008)}$ | $\mathbf{0.0014}_{(\mathbf{.005})}$ |
| | $1.5$ | $\mathbf{0.0098}_{(\mathbf{.014})}$ | $0.0098_{(.015)}$ | $0.0107_{(.012)}$ | $0.0166_{(.019)}$ |
| | $0$ | $0.0099_{(.016)}$ | $0.0099_{(.016)}$ | $0.0021_{(.006)}$ | $\mathbf{0.0015}_{(\mathbf{.005})}$ |
| | $0$ | $0.0108_{(.015)}$ | $0.0109_{(.016)}$ | $\mathbf{0.0001}_{(\mathbf{.004})}$ | $0.0007_{(.003)}$ |
| | $0$ | $0.0094_{(.014)}$ | $0.0094_{(.014)}$ | $0.0020_{(.006)}$ | $\mathbf{0.0017}_{(\mathbf{.008})}$ |
| | $1$ | $\mathbf{0.0081}_{(\mathbf{.012})}$ | $0.0082_{(.012)}$ | $0.0116_{(.015)}$ | $0.0172_{(.021)}$ |
| Expert 2 | $0$ | $0.0342_{(.042)}$ | $\mathbf{0.0338}_{(\mathbf{.042})}$ | $0.0575_{(.079)}$ | $0.0392_{(.059)}$ |
| | $1$ | $0.0355_{(.044)}$ | $\mathbf{0.0354}_{(\mathbf{.044})}$ | $0.1465_{(.148)}$ | $0.1084_{(.130)}$ |
| | $-1.5$ | $0.0222_{(.028)}$ | $\mathbf{0.0221}_{(\mathbf{.028})}$ | $0.0860_{(.087)}$ | $0.0672_{(.070)}$ |
| | $0$ | $0.0253_{(.032)}$ | $0.0252_{(.031)}$ | $\mathbf{0.0034}_{(\mathbf{.017})}$ | $0.0056_{(.022)}$ |
| | $0$ | $0.0296_{(.049)}$ | $0.0294_{(.049)}$ | $\mathbf{0.0037}_{(\mathbf{.020})}$ | $0.0059_{(.023)}$ |
| | $2$ | $\mathbf{0.0286}_{(\mathbf{.040})}$ | $0.0287_{(.040)}$ | $0.0375_{(.050)}$ | $0.0371_{(.051)}$ |
| | $0$ | $0.0195_{(.029)}$ | $0.0195_{(.029)}$ | $\mathbf{0.0040}_{(\mathbf{.015})}$ | $0.0083_{(.028)}$ |
| Gating | $1$ | $0.1379_{(.213)}$ | $\mathbf{0.0936}_{(\mathbf{.126})}$ | $0.1067_{(.125)}$ | N/A |
| | $2$ | $0.2650_{(.471)}$ | $\mathbf{0.1225}_{(\mathbf{.157})}$ | $0.4890_{(.277)}$ | |
| | $0$ | $0.0825_{(.116)}$ | $0.0641_{(.086)}$ | $\mathbf{0.0052}_{(\mathbf{.015})}$ | |
| | $0$ | $0.1466_{(.302)}$ | $0.1052_{(.196)}$ | $\mathbf{0.0017}_{(\mathbf{.007})}$ | |
| | $-1$ | $0.1875_{(.263)}$ | $\mathbf{0.1129}_{(\mathbf{.148})}$ | $0.2885_{(.295)}$ | |
| | $0$ | $0.1101_{(.217)}$ | $0.0803_{(.164)}$ | $\mathbf{0.0120}_{(\mathbf{.062})}$ | |
| | $0$ | $0.0806_{(.121)}$ | $0.0610_{(.095)}$ | $\mathbf{0.0013}_{(\mathbf{.008})}$ | |
| $\sigma$ | $1$ | $0.0033_{(.004)}$ | $0.0035_{(.004)}$ | $\mathbf{0.0027}_{(\mathbf{.003})}$ | $0.0028_{(.003)}$ |

TABLE III. MEAN SQUARE ERROR BETWEEN EACH COMPONENT OF THE ESTIMATED PARAMETER VECTOR OF LASSO+$L_2$, $L_2$, MOE AND THE ACTUAL ONE.

| Model | MoE | $L_2$ | Lasso+$L_2$ | MIXLASSO |
|---|---|---|---|---|
| C. rate | $89.57\%_{(1.65\%)}$ | $89.62\%_{(1.63\%)}$ | $89.46\%_{(1.76\%)}$ | $82.89\%_{(1.92\%)}$ |
| ARI | $0.6226_{(.053)}$ | $0.6241_{(.052)}$ | $0.6190_{(.056)}$ | $0.4218_{(.050)}$ |

TABLE IV. AVERAGE OF THE ACCURACY OF CLUSTERING (CORRECT CLASSIFICATION RATE AND ADUJUSTED RAND INDEX).

[4] 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 standardised to have mean 0 and variance 1. Histogram of the log of salary shows multi-modality making it a good candidate for the response variable under the MoE model with two components.

$$\begin{aligned} Y = \log(salary) \sim & \pi_1(\boldsymbol{x};\boldsymbol{w})\mathcal{N}(y;\beta_{10}+\boldsymbol{x}^T\boldsymbol{\beta}_1,\sigma^2) \\ & +(1-\pi_1(\boldsymbol{x};\boldsymbol{w}))\mathcal{N}(y;\beta_{20}+\boldsymbol{x}^T\boldsymbol{\beta}_2,\sigma^2), \end{aligned} \quad (30)$$

where $\pi_1(\boldsymbol{x};\boldsymbol{w}) = \frac{e^{w_{10}+\boldsymbol{x}^T\boldsymbol{w}_1}}{1+e^{w_{10}+\boldsymbol{x}^T\boldsymbol{w}_1}}$. By taking all the tuning parameters equal zero, we obtain the maximum likelihood estimator of the model. We also compare our result with MIXLASSO from [4]. Table VI presents the parameter estimates for baseball salary data. Table VI provides the results in term of parameters estimation and Table V shows the results in terms of MSE, and $R^2$. These results clearly suggest that the proposed algorithm with the Lasso+$L_2$ penalty also shrinks some parameters to zero and have an acceptable results when comparing with MoE, it also shows that this model provides a better results than the MIXLASSO model.

| | MoE | Lasso+$L_2$ | MIXLASSO |
|---|---|---|---|
| $R^2$ | 0.8099 | 0.8020 | 0.4252 |
| MSE | $0.2625_{(.758)}$ | $0.2821_{(.633)}$ | $1.1858_{(2.792)}$ |

TABLE V. RESULTS FOR BASEBALL SALARIES DATA SET.

## V. CONCLUSION AND FUTURE WORK

In this paper we proposed a regularized MLE for the MoE model which encourages sparsity, and developed a blockwise EM algorithm which monotonically maximizes this regularized objective towards at least a local maximum. The proposed regularization does not require using approximations as in standard MoE regularization. The proposed algorithm is based on univariate updates of the model parameters via coordinate ascent, which allows to tackle problems in high-dimensional computation by avoiding

| Features | MLE, $\widehat{\sigma} = 0.277$ | | | Lasso+$L_2$, $\widehat{\sigma} = 0.345$ | | | MIXLASSO, $\widehat{\sigma} = 0.25$ | |
|---|---|---|---|---|---|---|---|---|
| | Exp.1 | Exp.2 | Gating | Exp.1 | Exp.2 | Gating | Exp.1 | Exp.2 |
| $x_0$ | 6.0472 | 6.7101 | -0.3958 | 5.9580 | 6.9297 | 0.0046 | 6.41 | 7.00 |
| $x_1$ | -0.0073 | -0.0197 | 0.1238 | -0.0122 | - | - | - | -0.32 |
| $x_2$ | -0.0283 | 0.1377 | 0.1315 | -0.0064 | - | - | - | 0.29 |
| $x_3$ | 0.0566 | -0.4746 | 1.5379 | - | - | - | - | -0.70 |
| $x_4$ | 0.3859 | 0.5761 | -1.9359 | 0.4521 | 0.0749 | - | 0.20 | 0.96 |
| $x_5$ | -0.2190 | -0.0170 | -0.9687 | - | - | - | - | - |
| $x_6$ | -0.0586 | 0.0178 | 0.4477 | -0.0051 | - | - | - | - |
| $x_7$ | -0.0430 | 0.0242 | -0.3682 | - | - | - | -0.19 | - |
| $x_8$ | 0.3991 | 0.0085 | 1.7570 | - | 0.0088 | - | 0.26 | - |
| $x_9$ | -0.0238 | -0.0345 | -1.3150 | 0.0135 | 0.0192 | - | - | - |
| $x_{10}$ | -0.1944 | 0.0412 | 0.6550 | -0.1146 | - | - | - | - |
| $x_{11}$ | 0.0726 | 0.1152 | 0.0279 | -0.0108 | 0.0762 | - | - | - |
| $x_{12}$ | 0.0250 | -0.0823 | 0.1383 | - | - | - | - | - |
| $x_{13}$ | -2.7529 | 1.1153 | -7.0559 | - | 0.3855 | -0.3946 | 0.79 | 0.70 |
| $x_{14}$ | 2.3905 | -1.4185 | 5.6419 | 0.0927 | -0.0550 | - | 0.72 | - |
| $x_{15}$ | -0.0386 | 1.1150 | -2.8818 | 0.3268 | 0.3179 | - | 0.15 | 0.50 |
| $x_{16}$ | 0.2380 | 0.0917 | -7.9505 | - | - | - | - | -0.36 |
| $x_1 * x_{13}$ | 3.3338 | -0.8335 | 8.7834 | 0.3218 | - | - | -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 | -0.0319 | - | - | - | - |
| $x_3 * x_{13}$ | 0.7445 | 0.3201 | 0.5622 | - | 0.0284 | -0.5828 | - | - |
| $x_3 * x_{14}$ | -0.0900 | -1.4934 | 0.1417 | -0.0883 | - | - | 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 | - | 0.004 | - | - | - |
| $x_7 * x_{14}$ | -0.1566 | 1.2809 | -3.5625 | -0.1362 | 0.0245 | - | - | - |
| $x_7 * x_{15}$ | -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 | - | 0.2727 | -0.3628 | 0.29 | -0.46 |
| $x_8 * x_{14}$ | 0.7843 | -1.3341 | 6.2614 | - | 0.0133 | - | -0.14 | - |
| $x_8 * x_{15}$ | 0.3711 | -0.4310 | 0.8033 | 0.3154 | - | - | - | - |
| $x_8 * x_{16}$ | -0.2158 | 0.7790 | 2.6731 | 0.0157 | - | - | - | - |

TABLE VI. FITTED MODELS FOR BASEBALL SALARY DATA.

matrix inversion and to promote its scalability. The results on both simulations and a real-data example confirm the effectiveness of the proposal. This was observed in terms of parameter estimation, the estimation of the actual support of the sparsity, and clustering accuracy. Namely, 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 future work would consist in performing additional model selection experiments and considering hierarchical MoE and MoE for discrete data.

## REFERENCES

[1] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[2] H. D. Nguyen and F. Chamroukhi, "An introduction to the practical and theoretical aspects of mixture-of-experts modeling," *ArXiv preprint arXiv:1707.03538v1*, Jul 2017. [Online]. Available: https://arxiv.org/abs/1707.03538v1

[3] S. E. Yuksel, J. N. W., and P. D. Gader, "Twenty years of mixture of experts," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 8, pp. 1177–1193, 2012.

[4] A. Khalili and J. Chen, "Variable selection in finite mixture of regression models," *Journal of the American Statistical association*, vol. 102, no. 479, pp. 1025–1038, 2007.

[5] N. Städler, P. Bühlmann, and S. Van De Geer, "$l1$-penalization for mixture regression models," *Test*, vol. 19, no. 2, pp. 209–256, 2010.

[6] C. Meynet, "An $\ell_1$-oracle inequality for the lasso in finite mixture gaussian regression models," *ESAIM: Probability and Statistics*, vol. 17, pp. 650–671, 2013.

[7] E. Devijver, "An $\ell_1$-oracle inequality for the lasso in multivariate finite mixture of multivariate gaussian regression models," *ESAIM: Probability and Statistics*, vol. 19, pp. 649–670, 2015.

[8] F. K. Hui, D. I. Warton, S. D. Foster *et al.*, "Multi-species distribution modeling using penalized mixture of regressions," *The Annals of Applied Statistics*, vol. 9, no. 2, pp. 866–882, 2015.

[9] K. Lange, *Optimization (2nd edition)*. Springer, 2013.

[10] L. R. Lloyd-Jones, H. D. Nguyen, and G. J. McLachlan, "A globally convergent algorithm for lasso-penalized mixture of linear regression models," *arXiv:1603.08326*, 2016.

[11] A. Khalili, "New estimation and feature selection methods in mixture-of-experts models," *Canadian Journal of Statistics*, vol. 38, no. 4, pp. 519–539, 2010.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. of the royal statistical society. Series B*, pp. 1–38, 1977.

[13] W. Jiang and M. A. Tanner, "On the approximation rate of hierarchical mixtures-of-experts for generalized linear models," *Neural computation*, vol. 11, no. 5, pp. 1183–1198, 1999.

[14] F. Chamroukhi, "Skew-normal mixture of experts," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 3000–3007.

[15] ——, "Skew t mixture of experts," *Neurocomputing*, vol. 266, pp. 390 – 408, 2017.

[16] ——, "Robust mixture of experts modeling using the t distribution," *Neural Networks*, vol. 79, pp. 20–36, 2016.

[17] H. D. Nguyen and G. J. McLachlan, "Laplace mixture of linear experts," *Computational Statistics & Data Analysis*, vol. 93, pp. 177–191, 2016.

[18] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[19] P. Tseng, "Coordinate ascent for maximizing nondifferentiable concave functions," 1988.

[20] ——, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of optimization theory and applications*, vol. 109, no. 3, pp. 475–494, 2001.

[21] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*. Taylor & Francis, 2015.

[22] D. R. Hunter and R. Li, "Variable selection using mm algorithms," *Annals of statistics*, vol. 33, no. 4, p. 1617, 2005.