Estimation and Feature Selection in Mixtures of Generalized Linear Experts Models

Bao Tuyen Huynh[a], Faicel Chamroukhi[a,*]

[a] *University of Caen, Laboratory of Mathematics LMNO, UMR CNRS Department of Mathematics and Computer Science, 14000 Caen, France.*

Abstract

Mixtures-of-Experts (MoE) are conditional mixture models that have shown their performance in modeling heterogeneity in data in many statistical learning approaches for prediction, including regression and classification, as well as for clustering. Their estimation in high-dimensional problems is still however challenging. We consider the problem of parameter estimation and feature selection in MoE models with different generalized linear experts models, and propose a regularized maximum likelihood estimation that efficiently encourages sparse solutions for heterogeneous data with high-dimensional predictors. The developed proximal-Newton EM algorithm includes proximal Newton-type procedures to update the model parameter by monotonically maximizing the objective function and allows to perform efficient estimation and feature selection. An experimental study shows the good performance of the algorithms in terms of recovering the actual sparse solutions, parameter estimation, and clustering of heterogeneous regression data, compared to the main state-of-the art competitors.

Keywords: Mixture-of-experts, Regularized maximum-likelihood, Feature selection, EM algorithm, Coordinate ascent, Proximal-Newton.

1. Introduction and related work

Mixtures-of-experts (MoE) models introduced by Jacobs et al. (1991), including hierarchical MoE Jordan and Jacobs (1994), have shown their performance in statical modeling of heterogeneous data in many statistical learning problems including regression, clustering and classification. MoE belong to the family of mixture models McLachlan

*Corresponding author
Email address: faicel.chamroukhi@unicaen.fr (Faicel Chamroukhi)

and Peel. (2000) and consist of a fully conditional mixture models where the mixing proportions and the components densities, i.e the gating network and the experts network, are functions of the inputs. This gives MoE some advantage in representing complex data distributions than the standard unconditional mixture distributions. The statistical inference and numerical computations of (hierarchical) MoE models are studied in Jordan and Jacobs (1994); Jiang and Tanner (1999a,b, 2000). MoE have been recently extended to model and cluster heterogeneous regression with possibly asymmetric and noisy observation, as in (Chamroukhi, 2016b,a; Nguyen and McLachlan, 2016; Chamroukhi, 2017). A general review of the MoE models and their applications can be found in Yuksel et al. (2012); Nguyen and Chamroukhi (2018).

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 (2007) and Bunea et al. (2008) for more details. In a MoE scenario, estimating the parameters with moderate numbers of features and mixture components using MLE is challenging. 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. 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, 2006), 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 related mixture models for simultaneous regression and clustering, including mixture of linear regressions (MLR), Khalili and Chen (2007) proposed regularized MLE techniques, including MIXLASSO, MIXHARD and MIXSCAD and provided asymptotic properties corresponding to these penalty functions. Another ℓ_1 penalization for MLR models for high-dimensional data was proposed by Städler et al. (2010), which uses an adaptive Lasso penalized estimator. Meynet (2013) provided an ℓ_1-oracle inequality for a Lasso estimator in finite mixture of Gaussian regression models. This result was a complementary result to Städler et al. (2010) by studying the ℓ_1-regularization properties of the Lasso in parameter estimation, rather than by considering it as a variable selection procedure. Other interesting approaches for feature selection in MLR with high-dimensional data can be found in Devijver (2015), Hui et al. (2015) and Lloyd-Jones et al. (2018).

In Khalili (2010), the author extended his MLR regularization to the MoE setting, provided a root-n consistent, 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, this is based on an approximated penalty function, and uses a Newton-Raphson procedure in the updates of the gating network parameters. The algorithm requires matrix inversion which can be of some cost in a high-dimensional setting.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 large scale scenario.

In this paper, we propose an efficient regularized estimation and feature selection of Mixtures-of-Experts that encourages sparse solutions and consider MoE models for three 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. The remainder of this paper is organized as follows. In Section 2, we describe the modeling with MoE for heterogeneous data and maximum-likelihood parameter estimation. Then, in Section 3, the proposed regularized maximum likelihood strategy of the MoE models and the EM-based algorithm are developed. An experimental study, carried out on simulated and real data sets, is provided in Section 4. Finally, in Section 5, we draw concluding remarks and mention future direction.

2. Mixture-of-Experts and Maximum Likelihood Estimation

Let $((\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n))$ be a random sample of n independently and identically distributed (i.i.d) pairs $(\boldsymbol{X}_i, Y_i)$, $(i = 1, \ldots, n)$ where $Y_i \in \mathcal{X} \subset \mathbb{R}$ is the ith response given some vector of $p \in \mathbb{N}$ predictors $\boldsymbol{X}_i \in \mathcal{X} \subset \mathbb{R}^p$. We consider the MoE modeling for the analysis of a heteregeneous set of such data. Let $\mathcal{D} = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n))$ be an observed data sample.

2.1. The MoE model

The mixture-of-experts model assumes that the observed pairs $(\boldsymbol{x}, y)$ are generated from $K \in \mathbb{N}$ (possibly unknown) parametric probability density 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. The generative process of the data hence

assumes the following hierarchical representation. Given the predictor or the input $\boldsymbol{x}_i$, the categorical variable Z_i is generated according to the multinomial distribution:

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

where each of the probabilities $\pi_{z_i}(\boldsymbol{x}_i; \mathbf{w}) = \mathbb{P}(Z_i = z_i|\boldsymbol{X}_i = \boldsymbol{x}_i)$ is given by the gating network. 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 its distribution is $p_{z_i}(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}_{z_i})$, that is:

$$Y_i|Z_i = z_i, \boldsymbol{X}_i = \boldsymbol{x}_i \sim p_{z_i}(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}_{z_i}) \tag{2}$$

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 density or the probability mass function of the expert z_i depending on the nature of the data $(\boldsymbol{x}, y)$ within the group z_i. The gating network which gives the probabilities in (1) is defined by the distribution of the hidden variable Z given the predictor $\boldsymbol{x}$, i.e., $\pi_k(\boldsymbol{x}; \mathbf{w}) = \mathbb{P}(Z = k|\boldsymbol{X} = \boldsymbol{x}; \mathbf{w})$, is in general given by gating softmax functions of the form:

$$\pi_k(\boldsymbol{x}_i; \mathbf{w}) = \mathbb{P}(Z_i = k|\boldsymbol{X}_i = \boldsymbol{x}_i; \mathbf{w}) = \frac{\exp(w_{k0} + \boldsymbol{x}_i^T \boldsymbol{w}_k)}{1 + \sum_{l=1}^{K-1} \exp(w_{l0} + \boldsymbol{x}_i^T \boldsymbol{w}_l)} \tag{3}$$

for $k = 1, \ldots, K-1$ with $\mathbf{w} = (\mathbf{w}_1^T, \ldots, \mathbf{w}_{K-1}^T)^T$ and $\mathbf{w}_k = (w_{k0}, \boldsymbol{w}_k^T)^T \in \mathbb{R}^{p+1}$ such that $\mathbf{w}_K = \mathbf{0}$ is set to the null vector for identifiability (Jiang and Tanner, 1999a). Hence, formally, the MoE is defined by the following semi-parametric probability density (or mass) function:

$$p(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}_i; \mathbf{w}) p_k(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k) \tag{4}$$

that is parameterized by the parameter vector defined by $\boldsymbol{\theta} = (\mathbf{w}_1^T, \ldots, \mathbf{w}_{K-1}^T, \boldsymbol{\theta}_1^T, \ldots, \boldsymbol{\theta}_K^T)^T \in \mathbb{R}^{\nu_{\boldsymbol{\theta}}}$ $(\nu_{\boldsymbol{\theta}} \in \mathbb{N})$ 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 can be refereed to Nguyen and Chamroukhi (2018).

2.2. Maximum likelihood parameter estimation

Given an an observed data sample $\mathcal{D} = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n))$ generated from the MoE model (4), the unknown parameter vector $\boldsymbol{\theta}$ is commonly estimated by maximizing

the observed data log-likelihood

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k(\boldsymbol{x}_i; \mathbf{w}) p_k(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k) \tag{5}$$

by using the EM algorithm (Dempster et al., 1977; Jacobs et al., 1991) which allows to iteratively find an appropriate local maximizer of the log-likelihood function (5). Jiang and Tanner (2000) studied statistical estimation and numerical computations in (hierarchical) MoE models.

However, it is well-known that 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. In such a context, a regularization of the MLE is needed.

3. Regularized Maximum Likelihood Estimation for the MoE model

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 modeling, one may consider both sparsity in the feature space of the gates, and of the experts. As proposed, the MoE model inferred by maximizing a regularized log-likelihood criterion and encourages sparsity for both the gating network parameters and the experts network parameters. This does not require any approximation along with performing the maximization, therefore avoid matrix inversion. The proposed regularization that combines two Lasso penalties for the experts parameters, and for the gating network is defined by:

$$PL(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) - \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1 - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1. \tag{6}$$

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) where the author proposed regularization methods 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, 2007; Bunea et al., 2008). However, the ℓ_2 norm also affect the sparsity of the models. We therefore remove this ℓ_2 penalty in our proposal model. For parameter estimation, Khalili introduced an EM algorithm follows the suggestion of Hunter and Li (2005) to approximate the penalty function in a some neighborhood by a local quadratic function. After that, a Newton-Raphson can be used to update parameters in the M-step. 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 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. Recently, Chamroukhi and Huynh (2019) suggested another approach to the estimation and feature selection in MoE by using an EM algorithm with coordinate ascent updates to overcome these limitations of Khalili's method. But this proposal still has some drawbacks since unlike (6), it maximizes a version with for it with an additional ℓ_2 term which may affect sparsity, and it may require significant computing time due to the maximization of nonsmooth univariate concave function using the Newton method. Hence, it is needed to be improved to deal with large scale data sets. 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 three different models for the experts, that is Gaussian, Poisson, and

logistic regressors.

3.1. Parameter estimation with a proximal Newton-EM algorithm

For each of the three considered GLM for the MoE models, we propose an EM algorithm to monotonically find at least local maximizers of (6). The E-step is common to the three 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 3.2. The EM algorithm for the maximization of (6) requires the construction of the penalized complete-data log-likelihood, which is, in our context, given by

$$PL_c(\boldsymbol{\theta}) = L_c(\boldsymbol{\theta}) - \sum_{k=1}^{K} \lambda_k \|\boldsymbol{\beta}_k\|_1 - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 \tag{7}$$

where

$$L_c(\boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{k=1}^{K} Z_{ik} \log \left[\pi_k(\boldsymbol{x}_i; \mathbf{w}) p_k(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k)\right] \tag{8}$$

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 (6)).

E-step:. The E-Step computes the conditional expectation of the penalized complete-data log-likelihood (7), 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:

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[q]}) = \mathbb{E}\left[PL_c(\boldsymbol{\theta})|\mathcal{D};\boldsymbol{\theta}^{[q]}\right] = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{[q]}\log\left[\pi_k(\boldsymbol{x}_i;\mathbf{w})p_k(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_k)\right] - \sum_{k=1}^{K}\lambda_k\|\boldsymbol{\beta}_k\|_1 - \sum_{k=1}^{K-1}\gamma_k\|\boldsymbol{w}_k\|_1 \quad (9)$$

where

$$\tau_{ik}^{[q]} = \mathbb{P}(Z_i = k|y_i,\boldsymbol{x}_i;\boldsymbol{\theta}^{[q]}) = \pi_k(\boldsymbol{x}_i;\mathbf{w}^{[q]})p_k(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_k^{[q]})/p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}^{[q]}) \quad (10)$$

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 (9) w.r.t $\boldsymbol{\theta}$. The Q-function can be written as:

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{[q]}) = Q(\mathbf{w};\boldsymbol{\theta}^{[q]}) + \sum_{k=1}^{K}Q_k(\boldsymbol{\theta}_k;\boldsymbol{\theta}^{[q]}) \quad (11)$$

with

$$\begin{aligned} Q(\mathbf{w};\boldsymbol{\theta}^{[q]}) &= \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{[q]}\log\pi_k(\boldsymbol{x}_i;\mathbf{w}) - \sum_{k=1}^{K-1}\gamma_k\|\boldsymbol{w}_k\|_1, \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K-1}\tau_{ik}^{[q]}(w_{k0} + \boldsymbol{x}_i^T\boldsymbol{w}_k) - \sum_{i=1}^{n}\log\Big[1 + \sum_{k=1}^{K-1}e^{w_{k0}+\boldsymbol{x}_i^T\boldsymbol{w}_k}\Big] - \sum_{k=1}^{K-1}\gamma_k\|\boldsymbol{w}_k\|_1. \end{aligned} \quad (12)$$

and

$$Q_k(\boldsymbol{\theta}_k;\boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n}\tau_{ik}^{[q]}\log p_k(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}_k^{[q]}) - \lambda_k\|\boldsymbol{\beta}_k\|_1. \quad (13)$$

The parameters $\mathbf{w}$ are therefore updated by maximizing the function (12). 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 large scale 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 Appendix A. 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.

3.2. Proximal Newton-type procedure for updating the gating network

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 (12) given by

$$I(\mathbf{w}) = \sum_{i=1}^{n}\sum_{k=1}^{K-1} \tau_{ik}^{[q]}(w_{k0} + \boldsymbol{x}_i^T \boldsymbol{w}_k) - \sum_{i=1}^{n} \log\Big[1 + \sum_{k=1}^{K-1} e^{w_{k0} + \boldsymbol{x}_i^T \boldsymbol{w}_k}\Big] \tag{14}$$

with its Taylor expansion at current estimates

$$\tilde{I}_t(\mathbf{w}) = I(\mathbf{w}^{(t)}) + \nabla I(\mathbf{w}^{(t)})^T(\mathbf{w} - \mathbf{w}^{(t)}) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^{(t)})^T \nabla^2 I(\mathbf{w}^{(t)})(\mathbf{w} - \mathbf{w}^{(t)}), \tag{15}$$

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

$$\tilde{Q}_t(\mathbf{w}) = \tilde{I}_t(\mathbf{w}) - \sum_{k=1}^{K-1} \gamma_k \|\boldsymbol{w}_k\|_1 \tag{16}$$

instead of $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$ and then searches for the updating value $\mathbf{w}^{(t+1)}$ based on the solution of (A.2) 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 (A.2) 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 $I(\mathbf{w})$ w.r.t $(w_{k0}, \boldsymbol{w}_k)$ at $\tilde{\mathbf{w}}$ is given by (see Appendix Appendix B for more details)

$$l_{I_k}(w_{k0}, \boldsymbol{w}_k) = -\frac{1}{2}\sum_{i=1}^{n} d_{ik}(c_{ik} - w_{k0} - \boldsymbol{x}_i^T \boldsymbol{w}_k)^2 + C(\tilde{\mathbf{w}}), \tag{17}$$

where

$$c_{ik} = \tilde{w}_{k0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{w}}_k + \frac{\tau_{ik}^{[q]} - \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)}{\pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)(1 - \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i))}, \tag{18}$$

$$d_{ik} = \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)(1 - \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)), \tag{19}$$

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

$$\max_{(w_{k0}, \boldsymbol{w}_k)} l_{I_k}(w_{k0}, \boldsymbol{w}_k) - \gamma_k \|\boldsymbol{w}_k\|_1. \tag{20}$$

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

$$w_{kj}^{m+1} = \frac{\mathcal{S}_{\gamma_k}(\sum_{i=1}^{n} d_{ik} u_{ikj}^m x_{ij})}{\sum_{i=1}^{n} d_{ik} x_{ij}^2}, \tag{21}$$

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

$$w_{k0}^{m+1} = \frac{\sum_{i=1}^{n} d_{ik}(c_{ik} - \boldsymbol{x}_i^T \boldsymbol{w}_k^m)}{\sum_{i=1}^{n} d_{ik}}. \tag{22}$$

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 1.

Algorithm 1 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 (17) by using the current parameters.
5: Solve the penalized weighted least-square problem in (20) by using coordinate ascent algorithm and compute the solution $\tilde{\mathbf{w}}_k^{(s)}$ according to (21), (22).
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 in Chamroukhi and Huynh (2019) 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 $\pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)$ can become too small (or too close to 1), and the algorithm can get stuck while solving (20). To address this issure, 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 (20), one can solve the local quadratic model

$$\max_{(w_{k0}, \boldsymbol{w}_k)} \hat{l}_{I_k}(w_{k0}, \boldsymbol{w}_k) - \gamma_k \|\boldsymbol{w}_k\|_1. \tag{23}$$

where

$$\hat{l}_{I_k}(w_{k0}, \boldsymbol{w}_k) = -\frac{1}{8} \sum_{i=1}^{n} (\hat{c}_{ik} - w_{k0} - \boldsymbol{x}_i^T \boldsymbol{w}_k)^2 + \hat{C}(\tilde{\mathbf{w}}), \tag{24}$$

and

$$\hat{c}_{ik} = \tilde{w}_{k0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{w}}_k + 4(\tau_{ik}^{[q]} - \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)), \tag{25}$$

$\hat{C}(\tilde{\mathbf{w}})$ is a function of $\tilde{\mathbf{w}}$. Here, it is clear that this approach has some advantages. One can avoid computing the Hessian matrix and can also avoid numerical instability caused by $\pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)$. The increase of the $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$ after each loop is guaranteed, since this algorithm is a proximal Newton-type algorithm and is a specific case of the MM algorithm.

3.3. Updating the experts network

Now consider the updates of the experts models. Fortunately, the proximal Newton strategy described above can be once again used to update the expert's parameters in cases where the function $Q_k(\boldsymbol{\theta}_k; \boldsymbol{\theta}^{[q]})$ is concave. This property holds in Poisson regression and multi-logistic regression, which we consider here. For the Gaussian case, by fixing σ then $Q_k(\boldsymbol{\theta}_k; \boldsymbol{\theta}^{[q]})$ has the weighted Lasso form and the parameters can be updated using coordinate ascent with soft-threshoding operator. In this part, the expert's

parameters for three common generalized linear models are updated, including for the Gaussian experts, the Poisson experts and the multinomial logistic experts.

3.3.1. Expert network with Gaussian outputs

Consider the case of univariate continuous outputs Y_i where there is a relationship between the input $\boldsymbol{x}$ and the output Y given by regression functions. For the Gaussian case, 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 $\beta_{k0} \in \mathbb{R}$ 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) of Y becomes

$$Y_i | Z_i = z_i, \boldsymbol{x}_i \sim \mathcal{N}(.; \beta_{z_i 0} + \boldsymbol{\beta}_{z_i}^T \boldsymbol{x}_i, \sigma_{z_i}^2). \tag{26}$$

After updating the gating network parameters, the kth Gaussian expert is updated by updating the parameters $\boldsymbol{\theta}_k = (\beta_{k0}, \sigma_k^2, \boldsymbol{\beta}_k^T)^T$. This is done by updating (β_{k0}, β_k) while fixing σ_k. The coordinate ascent algorithm is used to solve this optimization problem. In this situation, the coordinate ascent algorithm was a special case of the MM algorithm. Specifically, the update of β_{kj} is performed by maximizing the $Q_k(\boldsymbol{\theta}_k; \boldsymbol{\theta}^{[q]})$ function in (13)

$$Q_k(\boldsymbol{\theta}_k; \boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n} \tau_{ik}^{[q]} \log \mathcal{N}(y_i; \beta_{k0} + \boldsymbol{\beta}_k^T \boldsymbol{x}_i, \sigma_k^2) - \lambda_k \|\boldsymbol{\beta}_k\|_1; \tag{27}$$

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

$$\beta_{kj}^{[s+1]} = \frac{\mathcal{S}_{\lambda_k \sigma_k^{[s]2}} \left(\sum_{i=1}^{n} \tau_{ik}^{[q]} r_{ikj}^{[s]} x_{ij} \right)}{\sum_{i=1}^{n} \tau_{ik}^{[q]} x_{ij}^2}, \tag{28}$$

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

$\beta_{kh}^{[s+1]} = \beta_{kh}^{[s]}$. For each iteration m, β_{k0} is updated by

$$\beta_{k0}^{[s+1]} = \frac{\sum_{i=1}^{n} \tau_{ik}^{[q]}(y_i - \boldsymbol{x}_i^T \boldsymbol{\beta}_k^{[s+1]})}{\sum_{i=1}^{n} \tau_{ik}^{[q]}}. \tag{29}$$

After updating all the vectors (β_{k0}, β_k), 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

$$\sigma_k^{2[q+2]} = \frac{\sum_{i=1}^{n} \tau_{ik}^{[q+1]}(y_i - \beta_{k0}^{[q+2]} - \boldsymbol{x}_i^T \boldsymbol{\beta}_k^{[q+2]})^2}{\sum_{i=1}^{n} \tau_{ik}^{[q+1]}}. \tag{30}$$

Each of the proposed algorithms is iterated until the change in $PL(\boldsymbol{\theta})$ is small enough. Zero coefficients can be obtained without any thresholds unlike in Khalili (2010); Hunter and Li (2005).

3.3.2. 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 (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

$$p_k(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}_k) = \mathbb{P}(y_i|\boldsymbol{x}_i; \beta_{k0}, \boldsymbol{\beta}_k) = \frac{\exp[-\exp(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k)] \exp[(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k) y_i]}{y_i!}. \tag{31}$$

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 (13), with

$$Q_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) = \underbrace{\sum_{i=1}^{n} \tau_{ik}^{[q]} \left[-\exp(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k) + y_i(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k) - \log(y_i!) \right]}_{P_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]})} - \lambda_k \|\boldsymbol{\beta}_k\|_1. \tag{32}$$

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 Appendix C.1 for more details)

$$\tilde{P}_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) = -\frac{1}{2} \sum_{i=1}^{n} a_{ik}(b_{ik} - \beta_{k0} - \boldsymbol{x}_i^T \boldsymbol{\beta}_k)^2 + D(\tilde{\beta}_{k0}, \tilde{\boldsymbol{\beta}}_k), \tag{33}$$

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

$$\max_{(\beta_{k0}, \boldsymbol{\beta}_k)} \tilde{P}_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) - \lambda_k \|\boldsymbol{\beta}_k\|_1. \tag{34}$$

Then the solution is taken in account for the next update of the proximal Newton algorithm. This can be interpreted as in Algorithm 2.

3.3.3. 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, (2) is defined by

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

Algorithm 2 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 (33) using the current parameters.
4: Solve the penalized weighted least-square problem in (34) 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.

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 (13) is written in the more explicit form

$$\begin{aligned} Q_k(\boldsymbol{\beta}_k; \boldsymbol{\theta}^{[q]}) \ = \ & \underbrace{\sum_{i=1}^{n} \tau_{ik}^{[q]} \Big[\sum_{r=1}^{R-1} u_{ir}(\beta_{kr0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr}) - \log\big(1 + \sum_{r=1}^{R-1} \exp(\beta_{kr0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr})\big) \Big]}_{I(\boldsymbol{\beta}_k)} \\ & - \sum_{r=1}^{R-1} \lambda_{kr} \|\boldsymbol{\beta}_{kr}\|_1. \end{aligned} \tag{35}$$

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 $I(\boldsymbol{\beta}_k)$ w.r.t. $(\beta_{kr0}, \boldsymbol{\beta}_{kr})$ at $\tilde{\boldsymbol{\beta}}_k$ is given by (see Appendix Appendix C.2)

$$\tilde{I}_r(\boldsymbol{\beta}_k) = -\frac{1}{2} \sum_{i=1}^{n} \tau_{ik}^{[q]} d_{ikr}(c_{ikr} - \beta_{kr0} - \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr})^2 + E(\tilde{\boldsymbol{\beta}}_k), \tag{36}$$

where

$$c_{ikr} = \tilde{\beta}_{kr0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}_{kr} + \frac{u_{ir} - \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)}{\alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)(1 - \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i))}, \tag{37}$$

$$d_{ikr} = \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)(1 - \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)), \tag{38}$$

and $E(\tilde{\boldsymbol{\beta}}_k)$ is a function of $\tilde{\boldsymbol{\beta}}_k$.
The corresponding Lasso form is described as following

$$\tilde{I}_r(\boldsymbol{\beta}_k) - \lambda_{kr}\|\boldsymbol{\beta}_{kr}\|_1. \tag{39}$$

Using a similar algorithm with Algorithm 1 by replacing the weighted Lasso in (20) with (39), 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 (39) one will maximize a simple weighted Lasso form

$$-\frac{1}{8}\sum_{i=1}^{n} \tau_{ik}^{[q]} (\hat{c}_{ikr} - \beta_{kr0} - \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr})^2 + \hat{E}(\tilde{\boldsymbol{\beta}}_k) - \lambda_{kr}\|\boldsymbol{\beta}_{kr}\|_1, \tag{40}$$

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 $\tilde{\boldsymbol{\beta}}_k$.

3.4. Algorithm tuning and model selection

In practice, the appropriate values of the tuning parameters (λ, γ) should be chosen. To select the tuning parameters, a modified BIC with a grid search scheme, as an extension of the criterion used in Städler et al. (2010) 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, a grid of the tuning parameters is chosen. Consider grids of values $\{\lambda_1, \ldots, \lambda_{M_1}\}$, $\{\gamma_1, \ldots, \gamma_{M_2}\}$ in the size of $\sqrt{n}$. For a given triplet (K, λ_i, γ_j), the maximal penalized log-likelihood estimators $\widehat{\boldsymbol{\theta}}_{K,\lambda,\gamma}$ is selected using each

of the hybrid EM algorithms presented above. The following modified BIC criterion,

$$\text{BIC}(K, \lambda, \gamma) = L(\widehat{\boldsymbol{\theta}}_{K,\lambda,\gamma}) - DF(\lambda, \gamma)\frac{\log n}{2}, \tag{41}$$

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 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.

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 (MoE). Several evaluation criteria are used to assess the performance of the models, including sparsity, parameters estimation and clustering criteria.

The R packages of codes of the developed algorithms and the documentation are publicly available on this link[1].

4.1. Evaluation criteria

The results of all the models are compared based on 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, the ratio of the estimated zero/nonzero coefficients to the true number of zero/nonzero coefficients of the true parameter is computed 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. We present the average proportion of these criteria computed over 100 different data sets. To deal with the label switching before calculating these criteria, we

[1] https://github.com/fchamroukhi/prEMME

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 for both the penalized parameters and the non penalized parameters and compare 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 $\|\theta_j - \hat{\theta}_j\|_2^2$.

For the clustering criterion, once the parameters are estimated and permuted, the provided conditional component probabilities $\hat{\tau}_{ik}$ defined in (10) represent a soft partition of the data. A hard partition of the data is given by applying the Bayes's allocation rule

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

where $\hat{z}_i$ represents the estimated cluster label for the ith observation. Given the estimated and true cluster labels, the correct classification rate and the Adjusted Rand Index (ARI) are computed.

4.2. Simulation study

For each data set, 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, a logistic model with two classes and a Poisson model of $K = 2$ expert components with the following regression coefficients:

• Parameters for the normal MoE model:

$$
\begin{aligned}
(\beta_{10}, \boldsymbol{\beta}_1)^T &= (0, 0, 1.5, 0, 0, 0, 1)^T; \\
(\beta_{20}, \boldsymbol{\beta}_2)^T &= (0, 1, -1.5, 0, 0, 2, 0)^T; \\
(w_{10}, \boldsymbol{w}_1)^T &= (1, 2, 0, 0, -1, 0, 0)^T; \\
\sigma_1 = \sigma_2 &= \sigma = 1.
\end{aligned}
$$

• Parameters for the Poisson model:

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

• Parameters for the multinomial-logistic model ($R = 2$):

$$
\begin{aligned}
(\beta_{110}, \boldsymbol{\beta}_{11})^T &= (0, -1, 2, 0, 0, 1.5, 0)^T; \\
(\beta_{210}, \boldsymbol{\beta}_{21})^T &= (0, 1, 0, 0, -2, 0, 0)^T; \\
(w_{10}, \boldsymbol{w}_1)^T &= (1, 0, 0, 1, 0, 0, -1.5)^T.
\end{aligned}
$$

100 data sets were generated for each simulation. The results will be presented in the following sections.

4.2.1. Sensitivity/specificity criteria

Table 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 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, a local maximum parameter that closed to the true value for the MoE of logistic model is not found (see Table 2). 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
Gaussian	0.700	1.000	0.790	1.000	0.748	0.995
Poisson	0.717	1.000	0.818	1.000	0.835	1.000
Logistic	0.693	0.960	0.835	0.805	0.780	0.980

Table 1: Sensitivity (S_1) and specificity (S_2) results.

True value			Lasso method			MLE 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 2: Estimated parameters for a logistic model data set.

4.2.2. Parameter estimation

The boxplots of all estimated parameters are shown in Figures 1, 2 and 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 MoE 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 3, Table 4 and Table 5, notice that the models using standard MoE give better results than the Lasso. This is because the Lasso can cause bias to the estimated parameters since the penalty functions are added to the log-likelihood function. On the other hand, the Lasso provide better results than MoE for estimating the zero coefficients in term of average mean squared error.

4.2.3. 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 6. The Lasso models provide a result for clustering data as good as MoE models. The difference between Lasso-MoE and standard MoE is smaller than 1%.

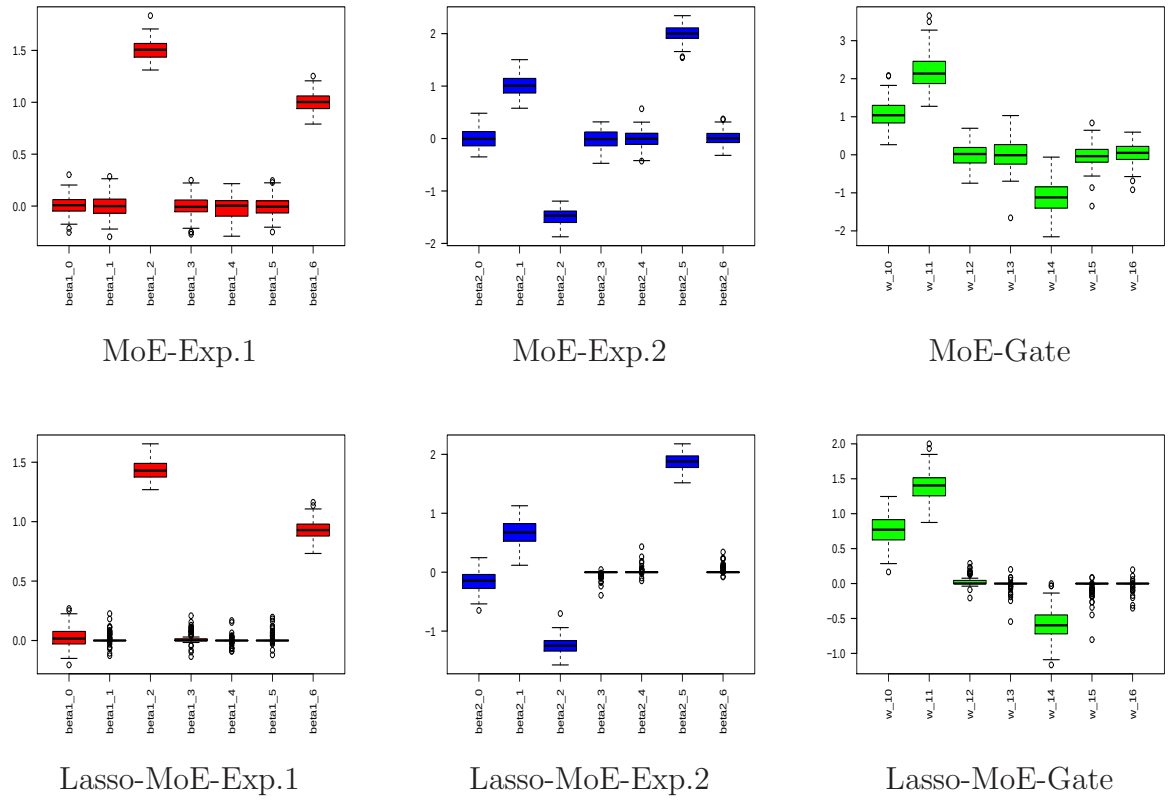

Figure 1: Boxplots of MoE and Lasso-MoE for Gaussian regression.

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 Lasso works as well as MoE models for the Gaussian and Poisson models. For logistic model, the Lasso is successful in retrieving the actual parameters used for the model, while the non regularized method failed in this task.

4.3. Applications to real data sets

In this part, five real data sets are analyzed as a further test of the proposal methodology. Two data sets are for the Gaussian model, two for the logistic model and one for Poisson model. The obtain results are compared with other methods, which provided by Khalili (2010) and Peralta and Soto (2014). The comparison are based upon three different criteria: the average mean squared error (MSE) between observation values and

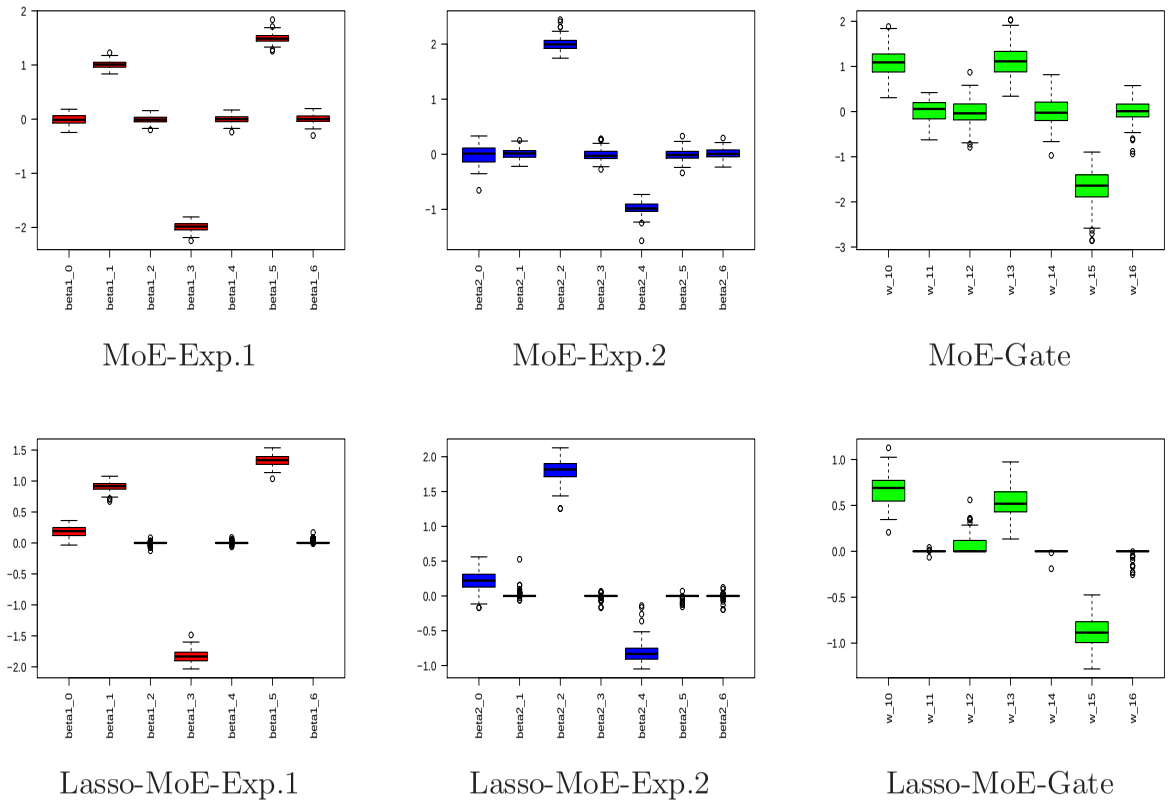

Figure 2: Boxplots of MoE and Lasso-MoE for Poisson regression.

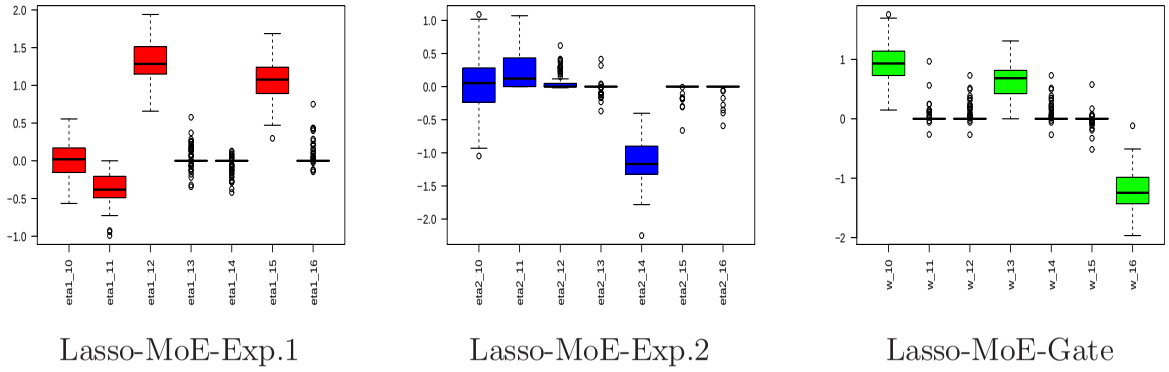

Figure 3: Boxplots of Lasso-MoE for logistic regression.

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,

Comp.	True value	Mean		Mean squared error	
		MoE	Lasso	MoE	Lasso
Exp.1	0	$0.010_{(.096)}$	$0.026_{(.088)}$	$0.0093_{(.015)}$	$0.0085_{(.014)}$
	0	$-0.002_{(.106)}$	$0.010_{(.045)}$	$0.0112_{(.016)}$	$0.0021_{(.006)}$
	1.5	$1.501_{(.099)}$	$1.434_{(.080)}$	$0.0098_{(.014)}$	$0.0107_{(.012)}$
	0	$0.000_{(.099)}$	$0.013_{(.044)}$	$0.0099_{(.016)}$	$0.0021_{(.006)}$
	0	$-0.022_{(.102)}$	$0.000_{(.032)}$	$0.0108_{(.015)}$	$0.0010_{(.004)}$
	0	$-0.001_{(.097)}$	$0.012_{(.043)}$	$0.0094_{(.014)}$	$0.0020_{(.006)}$
	1	$1.003_{(.090)}$	$0.931_{(.082)}$	$0.0081_{(.012)}$	$0.0114_{(.015)}$
Exp.2	0	$0.006_{(.185)}$	$-0.165_{(.175)}$	$0.0342_{(.042)}$	$0.0579_{(.077)}$
	1	$1.007_{(.188)}$	$0.675_{(.200)}$	$0.0355_{(.044)}$	$0.1455_{(.146)}$
	-1.5	$-1.492_{(.149)}$	$-1.243_{(.137)}$	$0.0222_{(.028)}$	$0.0851_{(.086)}$
	0	$-0.011_{(.159)}$	$-0.018_{(.055)}$	$0.0253_{(.032)}$	$0.0034_{(.017)}$
	0	$-0.010_{(.172)}$	$0.012_{(.060)}$	$0.0296_{(.049)}$	$0.0037_{(.020)}$
	2	$2.004_{(.169)}$	$1.876_{(.148)}$	$0.0286_{(.040)}$	$0.0374_{(.050)}$
	0	$0.008_{(.139)}$	$0.019_{(.059)}$	$0.0195_{(.029)}$	$0.0039_{(.015)}$
Gate	1	$1.095_{(.359)}$	$0.778_{(.224)}$	$0.1379_{(.213)}$	$0.0994_{(.122)}$
	2	$2.186_{(.480)}$	$1.400_{(.225)}$	$0.2650_{(.471)}$	$0.4111_{(.269)}$
	0	$0.007_{(.287)}$	$0.028_{(.067)}$	$0.0825_{(.116)}$	$0.0053_{(.013)}$
	0	$-0.001_{(.383)}$	$-0.014_{(.072)}$	$0.1466_{(.302)}$	$0.0054_{(.031)}$
	-1	$-1.131_{(.413)}$	$-0.584_{(.223)}$	$0.1875_{(.263)}$	$0.2226_{(.213)}$
	0	$-0.022_{(.331)}$	$-0.039_{(.111)}$	$0.1101_{(.217)}$	$0.0137_{(.068)}$
	0	$0.025_{(.283)}$	$-0.012_{(.062)}$	$0.0806_{(.121)}$	$0.0039_{(.017)}$
σ	1	$0.965_{(.045)}$	$0.989_{(.050)}$	$0.0033_{(.004)}$	$0.0027_{(.003)}$

Table 3: Estimated parameter vector of MoE and Lasso for Gaussian model.

the following value under the estimated model

$$\hat{Y} = \text{mode } p_k(Y|\boldsymbol{x}; z = k) = \text{mode } p_k(Y|\boldsymbol{x}; \hat{\boldsymbol{\theta}}_k),$$

is used as a predicted value for Y.

4.3.1. MoE model with Gaussian outputs

The regularized MoE for Gaussian model are tested on two real data sets: the housing data and the residential building data described on the website UC Irvine Machine Learning Repository. This was done to provide a comparison with the experiment of Khalili (2010) on housing data.

The housing data set concerns houses' value in the suburbs of Boston. It has 506 observations and 13 features that may affect the houses' value. The columns of X were

Comp.	True value	Mean		Mean squared error	
		MoE	Lasso	MoE	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: Estimated parameter vector of MoE and Lasso for Poisson model.

standardized to have the mean equal to 0 and the variance equal to 1. The response variable of interest is the median value of owner occupied homes in $\$1000's$, MEDV. Based on the histogram of Y = MEDV/sd(MEDV), where sd(MEDV) is the standard deviation of MEDV, Khalili separated Y into two groups of houses with "low" and "high" values. Hence, a MoE model is used to fit the response

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

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 estimated parameter of the MoE models obtained by Lasso and MLE are given in Table 7. These results are compared with Khalili's results. In Table 8, the results are provided in terms of average MSE and the correlation between the true observation value Y and its prediction $\hat{Y}$. A few parameters in both methods have the same value. The MSE and the correlation from the proposed method

Comp.	True value	Mean	Mean squared error
Exp.1	0	$0.008_{(.250)}$	$0.0623_{(.079)}$
	-1	$-0.370_{(.229)}$	$0.4494_{(.287)}$
	2	$1.315_{(.266)}$	$0.5403_{(.376)}$
	0	$0.020_{(.116)}$	$0.0138_{(.041)}$
	0	$-0.031_{(.092)}$	$0.0094_{(.027)}$
	1.5	$1.057_{(.249)}$	$0.2587_{(.250)}$
	0	$0.041_{(.124)}$	$0.0171_{(.066)}$
Exp.2	0	$0.029_{(.402)}$	$0.1624_{(.242)}$
	1	$0.228_{(.271)}$	$0.6687_{(.347)}$
	0	$0.068_{(.129)}$	$0.0213_{(.053)}$
	0	$-0.010_{(.078)}$	$0.0062_{(.025)}$
	-2	$-1.126_{(.324)}$	$0.8690_{(.575)}$
	0	$-0.023_{(.086)}$	$0.0079_{(.046)}$
	0	$-0.019_{(.084)}$	$0.0075_{(.041)}$
Gate	1	$0.934_{(.289)}$	$0.0881_{(.128)}$
	0	$0.025_{(.122)}$	$0.0154_{(.098)}$
	0	$0.046_{(.131)}$	$0.0193_{(.068)}$
	1	$0.628_{(.293)}$	$0.2236_{(.255)}$
	0	$0.046_{(.131)}$	$0.0193_{(.068)}$
	0	$-0.008_{(.092)}$	$0.0085_{(.043)}$
	-1.5	$-1.230_{(.358)}$	$0.2014_{(.272)}$

Table 5: Estimated parameter vector of Lasso for logistic model.

Criterion	Correct classification rate		ARI	
Model	MoE	Lasso	MoE	Lasso
Gaussian	$89.57\%_{(1.65\%)}$	$89.56\%_{(1.66\%)}$	$0.6226_{(.053)}$	$0.6222_{(.053)}$
Poisson	$88.85\%_{(2.04\%)}$	$88.96\%_{(2.03\%)}$	$0.5965_{(.063)}$	$0.6004_{(.063)}$
Logistic	N/A	$82.06\%_{(2.93\%)}$	N/A	$0.3985_{(.078)}$

Table 6: Average of the accuracy of clustering (correct classification rate and Adjusted Rand Index).

are better than those in Khalili (2010).

Considering the case $K = 3$ as an extension. The estimated parameters, the average MSE, and the correlation between the true observation value Y and its prediction $\hat{Y}$ for this case can be found in Table 9 and Table 10. It turns out that this model provides better results than those with $K = 2$ 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.

To evaluate the algorithm in a situation that has a moderate number of features, the Residential Building Data Set (UCI Machine Learning Repository) is used for further

Features	Lasso+ℓ_2 (Khalili), $\sigma = 0.352$			Lasso, $\sigma = 0.353$		
	Exp.1	Exp.2	Gate	Exp.1	Exp.2	Gate
x_0	2.16	2.84	1.04	2.18859	2.82834	1.00241
x_1	-0.09	-	-	-0.08818	-	-
x_2	-	0.07	-	-	0.06312	-
x_3	-	-	0.67	-	-	0.58559
x_4	-	0.05	-	0.04189	0.05606	-
x_5	-	-	-	-0.06550	-	-
x_6	-	0.60	-0.27	-	0.58868	-0.20882
x_7	-	-	-	-0.03640	-	-
x_8	-	-0.20	-	-	-0.19447	-
x_9	-	0.55	-	-	0.54518	-
x_{10}	-	-	-	-0.00329	-	-
x_{11}	-	-	0.54	-0.08641	-0.06184	0.39455
x_{12}	0.05	-	-	0.05058	-	-
x_{13}	-0.29	-0.49	1.56	-0.29022	-0.50688	1.36238

Table 7: Fitted models for housing data.

	Lasso+ℓ_2 (Khalili)	Lasso
R^2	0.8698	0.8832
MSE	$0.1371_{(.286)}$	$0.1178_{(.282)}$

Table 8: Results for Housing data set.

testing of the proximal Newton method in high-dimensional setting. This data set has 372 observations and 108 features, with the two response variables (V-9 and V-10), representing the sale prices and construction costs respectively. The V-9 variable (sale prices) is chosen as the response variable to be predicted. As usual, all the features are standardized to have zero-mean and unit-variance. The results of this algorithm with $K = 3$ expert components, $\lambda = 15$ and $\gamma = 5$ is provided. The estimated parameters are given in Table 11 and Table 12. The correlation and the mean squared error between the true value V-9 with its prediction can be found in Table 13. These results show that the proximal Newton method performs well in this setting, in which it provides a sparse model and competitive criteria in prediction and clustering.

4.3.2. 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 Cleverland Clinic Foundation heart disease data

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 9: Fitted models for housing data ($K = 3$).

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

Table 10: Results for Housing data set ($K = 3$).

set that available at the website UC Irvine Machine Learning Repository. This data set includes 13 features and 297 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 14. 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.

Features	Expert, $\sigma = 0.0255$			Gating network	
	Exp.1	Exp.2	Exp.3	Gate.1	Gate.2
x_0	-0.05023	-0.01755	0.01057	-1.57198	1.56379
x_1	-	-	-	-	-
x_2	-	-0.00780	-	-0.19781	-
x_3	-	-	-	-	-
x_4	0.01739	0.00404	-0.00951	-	-
x_5	-0.04247	-0.00595	-0.00545	-	0.25492
x_6	-	0.00275	-0.00424	-	-
x_7	-	-0.00312	-	-	-
x_8	0.02188	-0.00679	0.02428	0.13486	-
x_9	0.03075	-	0.06598	0.00636	-
x_{10}	0.00948	0.00201	-0.02975	-	-0.00683
x_{11}	0.05284	0.03829	0.12256	-	-0.71689
x_{12}	0.76636	1.00291	1.10888	-	-0.69369
x_{13}	-	-0.00492	-	-	-0.09245
x_{14}	-	0.00039	-	-	-
x_{15}	-	0.00208	-	-	-
x_{16}	-	-0.00848	-0.02100	-	-
x_{17}	-	-	-	-	-
x_{18}	-	-0.01647	-0.00209	-	-
x_{19}	-	0.05936	-	-	-
x_{20}	-	0.00926	0.02583	-	-
x_{21}	-	0.03881	0.04756	-	-
x_{22}	-0.00516	-0.00686	0.00243	-	-
x_{23}	-	-0.04237	-0.02182	-	-
x_{24}	-	-0.06031	-	-	-
x_{25}	-	-0.01020	0.02282	-	-
x_{26}	-	-0.00739	-0.00397	-	-
x_{27}	-	-	-	-	-
x_{28}	-	-	-	-	-
x_{29}	0.00192	-	-	-	-
x_{30}	-	-	-0.01254	-	-
x_{31}	0.02209	0.00543	-	-	-
x_{32}	0.01718	-0.00391	-	-	-
x_{33}	-	-	-	-	-
x_{34}	-	-	-	-	-
x_{35}	-	0.02850	-	-	-
x_{36}	-	-	-	-	-
x_{37}	-	-	0.01279	0.33964	-
x_{38}	-	-	-	-	-
x_{39}	-	0.00984	-0.03000	-	-
x_{40}	-	0.01759	0.10789	-	-0.03362
x_{41}	-0.01562	0.00276	0.04670	-	-
x_{42}	-	-0.05141	-	-	-
x_{43}	-	0.00320	-	-	-
x_{44}	-	0.00865	-	-	-
x_{45}	-	-	-	-	-
x_{46}	-	-	-	-	-
x_{47}	-	-	-	-	-
x_{48}	0.00193	0.02204	-0.02285	-	-
x_{49}	-	-	-	-	-
x_{50}	-	0.04639	-	-	-
x_{51}	0.00375	-	0.01495	-	-
x_{52}	-	-	-	-	-
x_{53}	-	-	-	-	-

Table 11: Fitted model parameters for residential building data (part 1).

Features	Expert, $\sigma = 0.0255$			Gating network	
	Exp.1	Exp.2	Exp.3	Gate.1	Gate.2
x_{54}	-	-0.00111	-0.00626	-	-
x_{55}	-	0.00284	-	-	-
x_{56}	-0.12790	-	-0.01529	-	-
x_{57}	-	0.00580	-0.00855	-	-
x_{58}	0.00159	0.00103	0.02203	-	-0.54361
x_{59}	0.06458	0.05364	0.10600	-	-0.43888
x_{60}	-	0.00651	-	-	-
x_{61}	-	-	-	-	-
x_{62}	-	-	-	-	-
x_{63}	0.00024	0.00001	-	-	-
x_{64}	-	-	-	-	-
x_{65}	-	-	-	-	-
x_{66}	-	-	-	-	-
x_{67}	0.00146	-	-0.02747	-	-
x_{68}	-	0.00402	-	-	-
x_{69}	-	0.00134	-	-	-
x_{70}	0.03668	0.01152	0.04296	-	-
x_{71}	-	-	-	-	-
x_{72}	-	-0.01570	-	-	-
x_{73}	-	0.00463	0.00396	-	-
x_{74}	-	-	-	-	-
x_{75}	-	-0.02087	-0.02719	-	-
x_{76}	-0.07538	-0.19186	-0.08124	0.11610	-
x_{77}	-	0.02078	0.00215	-	-
x_{78}	-	0.00191	-0.03891	-	-0.15319
x_{79}	-	-	-0.00612	-0.06685	-0.73955
x_{80}	-	-	-0.02227	-	-
x_{81}	-	0.00094	-0.01800	-	-
x_{82}	0.04180	-	-	-	-
x_{83}	-	-0.00472	-	-	-
x_{84}	-	-	-	-	-
x_{85}	-	-	-	-	-
x_{86}	-	0.01642	-0.02963	-	-
x_{87}	-	-0.00049	-	-	-
x_{88}	-	0.04240	-	-	-
x_{89}	-	0.01521	-0.03813	-	-
x_{90}	-	-	-	-	-
x_{91}	-	-	-	-	-
x_{92}	-0.01141	-0.01110	-	-	-
x_{93}	-	0.00248	-	-	-
x_{94}	-0.02270	0.01467	-	-	-
x_{95}	-	-0.01235	-	-	-
x_{96}	0.00044	0.01334	-	-	-
x_{97}	0.01150	0.01022	-	-	-
x_{98}	0.01755	-	-	-0.05581	-
x_{99}	-	0.01284	-	-	-
x_{100}	-	0.00051	-	-	-
x_{101}	0.04029	0.00930	-	-	-
x_{102}	-	-0.00513	-	-	-
x_{103}	-	-	-	-	-
x_{104}	-	-	-	-	-
x_{105}	0.02698	0.02557	-	-	-
x_{106}	-	-	-	-	-
x_{107}	-	-	-	-	-

Table 12: Fitted model parameters for residential building data (part 2).

Method	Predictive criteria		Number of observations		
	R^2	MSE	Class 1	Class 2	Class 3
Proximal Newton	0.9994	$0.00062_{(.0019)}$	18	287	67

Table 13: Results for clustering the residential building data set.

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 14: Fitted models for heart disease data.

4.3.3. 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).

The parameter estimates of the MoE models obtained by Lasso are given in Table 15 and Table 16, 17. The classification accuracy and percentage of features reduction results between the proposal with Peralta's work are found in Table 18. 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 dimension 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.

5. Conclusion and future work

In this work, 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.

Acknowledgements

This research is supported by Agence Nationale de la Recherche (ANR) grant SMILES ANR-18-CE40-0014 and by Région Normandie grant RIN AStERiCs.

Appendix A. Proximal Newton-type methods

Assume that we want to solve an optimization problem given by

$$\min_{x \in \mathbb{R}^n} f(x) = g(x) + h(x), \tag{A.1}$$

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 15: Fitted models for 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 16: Fitted models for 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 17: Fitted models for Musk-1 data (part 2).

Dataset name	Classification accuracy		Dimensionality reduction	
	Lasso (Peralta)	Lasso	Lasso (Peralta)	Lasso
Ionosphere	84.1%	96.6%	21.9%	73.7%
Musk-1	80.0%	93.3%	79.6%	90.0%

Table 18: Classification accuracy and percentage of features reduction results.

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:

$$\hat{f}_k(x) = g(x_k) + \nabla g(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T H_k(x - x_k) + h(x), \qquad \text{(A.2)}$$

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 (A.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 3 (see Lee et al. (2014)).

Algorithm 3 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.

Appendix B. Partial quadratic approximation for the gating network

The $Q(\mathbf{w}; \boldsymbol{\theta}^{[q]})$ function in (12) 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

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

By taking the first and second derivatives of $I(\mathbf{w})$ w.r.t $(w_{k0}, \boldsymbol{w}_k)$

$$\frac{\partial I(\mathbf{w})}{\partial w_{kj}} = \sum_{i=1}^{n} (\tau_{ik}^{[q]} - \pi_k(\boldsymbol{x}_i; \mathbf{w})) x_{ij}, \tag{B.1}$$

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

for $j, h \in \{0, 1, \ldots, p\}$ with $x_{i0} = 1$, then the partial quadratic approximation to $I(\mathbf{w})$ w.r.t $(w_{k0}, \boldsymbol{w}_k)$ at $(\tilde{w}_{k0}, \tilde{\boldsymbol{w}}_k)$ is given by

$$l_{I_k}(w_{k0}, \boldsymbol{w}_k) = -\frac{1}{2} \sum_{i=1}^{n} d_{ik} (c_{ik} - w_{k0} - \boldsymbol{x}_i^T \boldsymbol{w}_k)^2 + C(\tilde{\mathbf{w}}), \tag{B.3}$$

and

$$c_{ik} = \tilde{w}_{k0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{w}}_k + \frac{\tau_{ik}^{[q]} - \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)}{\pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)(1 - \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i))}, \tag{B.4}$$

$$d_{ik} = \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)(1 - \pi_k(\tilde{\mathbf{w}}; \boldsymbol{x}_i)), \tag{B.5}$$

$C(\tilde{\mathbf{w}})$ is a function of $\tilde{\mathbf{w}}$.

Appendix C. Quadratic approximation for the experts network

Appendix C.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 (13) is constructed using Taylor expansion. This function is given by

$$Q_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) = P_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) - \lambda_k \|\boldsymbol{\beta}_k\|_1, \tag{C.1}$$

where $P_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]})$ is a concave, continuously differentiable function and

$$P_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) = \sum_{i=1}^{n} \tau_{ik}^{[q]} \big[-\exp(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k) + y_i(\beta_{k0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_k) - \log(y_i!) \big]. \tag{C.2}$$

The first and second derivatives of $P_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]})$ w.r.t $(\beta_{k0}, \boldsymbol{\beta}_k)$ 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

$$\tilde{P}_k(\{\beta_{k0}, \boldsymbol{\beta}_k\}; \boldsymbol{\theta}^{[q]}) = -\frac{1}{2} \sum_{i=1}^{n} a_{ik} (b_{ik} - \beta_{k0} - \boldsymbol{x}_i^T \boldsymbol{\beta}_k)^2 + D(\tilde{\beta}_{k0}, \tilde{\boldsymbol{\beta}}_k), \tag{C.3}$$

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 $(\tilde{\beta}_{k0}, \tilde{\boldsymbol{\beta}}_k)$.

Appendix C.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 (13), where as before

$$Q_k(\boldsymbol{\beta}_k; \boldsymbol{\theta}^{[q]}) = I(\boldsymbol{\beta}_k) - \sum_{r=1}^{R-1} \lambda_{kr} \|\boldsymbol{\beta}_{kr}\|_1, \tag{C.4}$$

$I(\boldsymbol{\beta}_k)$ is a concave, continuously differentiable function and

$$I(\boldsymbol{\beta}_k) = \sum_{i=1}^{n} \tau_{ik}^{[q]} \Big[\sum_{r=1}^{R-1} u_{ir}(\beta_{kr0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr}) - \log\big(1 + \sum_{r=1}^{R-1} \exp(\beta_{kr0} + \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr})\big) \Big]. \tag{C.5}$$

The first and second derivatives of $I(\boldsymbol{\beta}_k)$ w.r.t $(\beta_{kr0}, \boldsymbol{\beta}_{kr})$ are

$$\frac{\partial I(\boldsymbol{\beta}_k)}{\partial \beta_{krj}} = \sum_{i=1}^{n} \tau_{ik}^{[q]} x_{ij}(u_{ir} - \alpha_{kr}(\boldsymbol{\beta}_k; \boldsymbol{x}_i)), \tag{C.6}$$

$$\frac{\partial^2 I(\boldsymbol{\beta}_k)}{\partial \beta_{krj} \partial \beta_{krh}} = -\sum_{i=1}^{n} \tau_{ik}^{[q]} x_{ij} x_{ih} \alpha_{kr}(\boldsymbol{\beta}_k; \boldsymbol{x}_i)(1 - \alpha_{kr}(\boldsymbol{\beta}_k; \boldsymbol{x}_i)), \tag{C.7}$$

for $j, h \in \{0, 1, \ldots, p\}$ and $x_{i0} = 1$. Hence, the partial quadratic approximation $\tilde{I}_r(\boldsymbol{\beta}_k)$ of $I(\boldsymbol{\beta}_k)$ w.r.t. $(\beta_{kr0}, \boldsymbol{\beta}_{kr})$ at $\tilde{\boldsymbol{\beta}}_k$ can be described as following

$$\tilde{I}_r(\boldsymbol{\beta}_k) = -\frac{1}{2} \sum_{i=1}^{n} \tau_{ik}^{[q]} d_{ikr}(c_{ikr} - \beta_{kr0} - \boldsymbol{x}_i^T \boldsymbol{\beta}_{kr})^2 + E(\tilde{\boldsymbol{\beta}}_k), \tag{C.8}$$

with

$$c_{ikr} = \tilde{\beta}_{kr0} + \boldsymbol{x}_i^T \tilde{\boldsymbol{\beta}}_{kr} + \frac{u_{ir} - \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)}{\alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)(1 - \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i))}, \tag{C.9}$$

$$d_{ikr} = \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)(1 - \alpha_{kr}(\tilde{\boldsymbol{\beta}}_k; \boldsymbol{x}_i)), \tag{C.10}$$

$E(\tilde{\boldsymbol{\beta}}_k)$ is a function of $\tilde{\boldsymbol{\beta}}_k$.

References

References

Boyd, S., Vandenberghe, L., 2004. Convex optimization. Cambridge university press.

Bunea, F., et al., 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., 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.

Chamroukhi, F., 2016a. Robust mixture of experts modeling using the t distribution. Neural Networks 79, 20–36.

Chamroukhi, F., 2016b. Skew-normal mixture of experts. In: Neural Networks (IJCNN), 2016 International Joint Conference on Neuron Networks. IEEE, pp. 3000–3007.

Chamroukhi, F., 2017. Skew t mixture of experts. Neurocomputing - Elsevier 266, 390–408.

Chamroukhi, F., Huynh, B.-T., 2019. Regularized maximum likelihood estimation and feature selection in mixtures-of-experts models. Journal de la SFdS 160 (1), 57–85.

Dempster, A. P., Laird, N. M., Rubin, D. B., 1977. Maximum likelihood from incomplete data via the em algorithm. J. of the royal statistical society. Series B, 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.

Fan, J., Li, R., 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American statistical Association 96 (456), 1348–1360.

Fraley, C., 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., Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. Journal of statistical software 33 (1), 1.

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.

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

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., Lange, K., 2004. A tutorial on mm algorithms. The American Statistician 58 (1), 30–37.

Hunter, D. R., Li, R., 2005. Variable selection using mm algorithms. Annals of statistics 33 (4), 1617.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., Hinton, G. E., 1991. Adaptive mixtures of local experts. Neural computation 3 (1), 79–87.

Jiang, W., Tanner, M. A., 1999a. Hierarchical mixtures-of-experts for exponential family regression models: approximation and maximum likelihood estimation. Annals of Statistics, 987–1011.

Jiang, W., 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., 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., Qinghua, J., 2018. Model selection for the localized mixture of experts models. Journal of Applied Statistics 45 (11), 1994–2006.

Jordan, M. I., 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., Chen, J., 2007. Variable selection in finite mixture of regression models. Journal of the American Statistical association 102 (479), 1025–1038.

Lange, K., 2013. Optimization (2nd edition). Springer.

Lee, J. D., Sun, Y., Saunders, M. A., 2014. Proximal newton-type methods for minimizing composite functions. SIAM Journal on Optimization 24 (3), 1420–1443.

Lee, S.-I., Lee, H., Abbeel, P., Ng, A. Y., 2006. Efficient l_1 regularized logistic regression. In: AAAI. Vol. 6. pp. 401–408.

Lloyd-Jones, L. R., Nguyen, H. D., McLachlan, G. J., 2018. A globally convergent algorithm for lasso-penalized mixture of linear regression models. Computational Statistics & Data Analysis 119, 19–38.

McLachlan, G. J., Peel., D., 2000. Finite mixture models. New York: Wiley.

Meynet, C., 2013. An ℓ_1-oracle inequality for the lasso in finite mixture gaussian regression models. ESAIM: Probability and Statistics 17, 650–671.

Nguyen, H. D., Chamroukhi, F., Feb 2018. Practical and theoretical aspects of mixture-of-experts modeling: An overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, e1246–n/a.
URL http://dx.doi.org/10.1002/widm.1246

Nguyen, H. D., McLachlan, G. J., 2016. Laplace mixture of linear experts. Computational Statistics & Data Analysis 93, 177–191.

Park, M. Y., Hastie, T., 2007. Penalized logistic regression for detecting gene interactions. Biostatistics 9 (1), 30–50.

Peralta, B., Soto, A., 2014. Embedded local feature selection within mixture of experts. Information Sciences 269, 176–187.

Städler, N., Bühlmann, P., Van De Geer, S., 2010. $l1$-penalization for mixture regression models. Test 19 (2), 209–256.

Stephens, M., Phil, D., 1997. Bayesian methods for mixtures of normal distributions.

Tang, Y., Xiang, L., 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, 267–288.

Tseng, P., 2001. Convergence of a block coordinate descent method for nondifferentiable minimization. Journal of optimization theory and applications 109 (3), 475–494.

Wang, Z., Ma, S., Wang, C.-Y., Zappitelli, M., Devarajan, P., 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.

Xu, L., Jordan, M. I., Hinton, G. E., 1995. An alternative model for mixtures of experts. In: Advances in neural information processing systems. pp. 633–640.

Yuan, M., Lin, Y., 2006. 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., Gader, P. D., 2012. Twenty years of mixture of experts. IEEE transactions on neural networks and learning systems 23 (8), 1177–1193.

Zhang, C.-H., 2010. Nearly unbiased variable selection under minimax concave penalty. The Annals of statistics 38 (2), 894–942.

Zou, H., 2006. The adaptive lasso and its oracle properties. Journal of the American statistical association 101 (476), 1418–1429.

Zou, H., 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.