# Model selection for mixture of experts using group fused lasso

**ARTICLE HISTORY**
Compiled August 10, 2019

**ABSTRACT**
The mixture of experts, or mixture of learners in general, is a popular and powerful machine learning model in which each expert learns to handle a different region of the covariate space. However, it is crucial to choose an appropriate number of experts to avoid overfitting or underfitting. We introduce a group fused lasso term to the model with the goal of making the coefficients of experts and gating network closer together. By varying the strength of the penalization, we can avoid overspecialization of each expert and choose the optimal set of parameters at the same time. An efficient algorithm to optimize the problem is developed using block-wise coordinate descent in the dual counterpart. Numerical results on simulated and real world datasets show that the penalized model outperforms the unpenalized one and performs on par with many well-known machine learning models.

**KEYWORDS**
Mixture of experts; group lasso; fused lasso; blockwise coordinate descent; duality; convex optimization

Word count: 3231

## 1. Introduction

The decision tree has been one of the most popular and widely used predictive models not only in statistics but also in almost all other scientific fields. The reasons for its success are its robustness and easy interpretability. Decision trees can be fitted to almost every kind of data [1] and their clear structure allows everyone to interpret the model easily. However, decision trees have the huge drawback that they are greedy algorithms, meaning each split is made to optimize a splitting criterion, without consideration of latter nodes. Trees also use the hard-split rule, which is non-smooth and cannot be trained using maximum likelihood [2]. There have been a few modifications to the original tree to make it non-greedy [3], [4], [5]. Nonetheless, these models are relatively time-consuming to train and difficult to understand since they do not use the "user-friendly" likelihood function.

Jordan and Jacobs [6] and Jacobs et al. [7] introduced a tree-structured architecture called Mixture of Experts (ME) which shares the same clear representation with decision trees. ME uses the same divide-and-conquer strategy like the decision tree and multivariate adaptive regression splines [8], but after we divide the input space into many subspaces, we fit a linear regression (the authors call it an expert) to each subspace. The beauty of ME is that the splits between subspaces are soft, implying that observations get assigned to all nodes or subspaces, with some probabilities which sum to 1, instead of to just one node or subspace as with decision trees. This smoothness

allows the model to be estimated easily using maximum likelihood. However, ME can underfit or overfit if we choose two few or too many experts. If we choose too few linear experts, we may not have enough experts to cover complicated covariate regions. On the other hand, if there are too many experts in the model, some experts may start to specialize in noise regions, limiting generalization. Until now, the literature for model selection for ME consists mainly of three parts: growing the structure [9],[10]; pruning the structure [11],[12]; and using Bayesian techniques [13],[14],[15]. The first method builds the structure slowly by adding one level or expert at a time. On the other hand, the second method begins with a large model and then tries to reduce the model complexity. The goal is to keep only the most-used branches and remove the least-used ones. The stopping time for both of the above methods is typically chosen by cross validation. Lastly, Bayesian techniques impose sparsity-promoting priors on the parameters so that the model has a smaller number of non-zero weights.

In this paper we propose another method for pruning the structure of ME. Specifically, we want to use the group fused lasso [16] to accomplish this task. Initializing from a large structured ME, we use a fusion penalty to penalize the difference between coefficients of different components of the gating network and experts so that we will have some identical experts with identical weights as well. We then can merge these similar experts and their weights together to simplify the structure.

The paper is organized as follows: In Section 2 we will give a review about Mixture of Experts and the penalty term that we propose. Details about how to fit the new penalized model are included in Section 3. Numerical results on simulation and real-world datasets will be presented in Section 4.

## 2. Penalized Mixture of Experts

We first introduce and give some background on the Mixture of Experts (ME) model. Then we introduce our fused lasso penalty.

### *2.1. Mixture of Experts*

In the ME model, a fixed number of experts and a gating network, which assigns weights to the experts, work together to solve a nonlinear supervised learning problem. In this paper we will consider only the Normal regression case, but generalization to cases of classification and Poisson regression is straightforward. The gating network's job is to divide the covariate space into many small subspaces by making soft splits of the whole space. On the other hand, the job of each expert is to specialize in one of the subspaces and learn the pattern in that particular subspace. Due to soft-splitting, the model has a smooth transition from one subspace to another and the predictions in those transitioning regions are stable.

Let $\{Y, X\}$ be our sample, where $Y$ is a vector of length $N$ and $X$ is a covariate matrix of dimension $N \times P$ (including the intercept column). For now we assume we work with low-dimensional data $N > P$. Modification for high-dimensional data $N < P$ will be discussed in Section 3.3.

If there are $K$ experts in the ME model, we denote by $\beta_k$ $(k = 1, ...K)$ the coefficient vector for the $k$th expert and by $\gamma_k$ $(k = 1, ...K)$ the set of coefficients governing how the gating network assigns weights to the $K$ experts. We also define $\beta = (\beta_1^T, \beta_2^T, \ldots, \beta_K^T)^T, \gamma = (\gamma_1^T, \gamma_2^T, \ldots, \gamma_K^T)^T$, and $\theta = (\beta^T, \gamma^T)^T$.

For a data point $(X^{(n)}, Y^{(n)})$, let $P(Y^{(n)}|k, X^{(n)}, \beta_k)$ be the conditional pdf of $Y^{(n)}$ given $X^{(n)}$ according to the $k^{th}$ expert and let $P(k|X^{(n)}, \gamma)$ be the weight assigned to the expert $k$, for $k = 1, \ldots, K$. Then the conditional pdf of $Y^{(n)}$ given $X^{(n)}$, according to the ME model is given by

$$P(Y^{(n)}|X^{(n)}, \theta) = \sum_{k=1}^{K} P(k|X^{(n)}, \gamma) P(Y^{(n)}|k, X^{(n)}, \beta_k).$$

As in the original ME paper [7], we define the weights of the gating network using the softmax function such that

$$P(k|X^{(n)}, \gamma) = \frac{\exp(\gamma_k^T X^{(n)})}{\sum_{l=1}^{K} \exp(\gamma_l^T X^{(n)})}, \qquad k = 1, \ldots, K \text{ and } n = 1, \ldots, N.$$

We note that for the purpose of identifiability, we force $\gamma_K$, the coefficients of the last gating network's component, to be 0, just as in the case of multinomial logistic regression.

This choice of weight function for the gating network guarantees a positive weight for each expert across the entire covariate space (so there is soft-splitting between experts, unlike in regression trees), and the weights assigned to the experts at any point in the covariate space sum to 1. Since we consider the Normal regression case, the conditional pdf $P(Y^{(n)}|k, X^{(n)}, \beta_k)$ is the pdf of a Normal distribution with mean ${X^{(n)}}^T \beta_k$.

Under this setting, we can write down the likelihood function as

$$\mathbf{L}(\theta, \sigma^{\mathbf{2}}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \frac{e^{\gamma_k^T X^{(n)}}}{\sum_{l=1}^{K} e^{\gamma_l^T X^{(n)}}} \cdot \frac{1}{\sqrt{2\pi}\sigma} \exp\left( - \frac{(Y^{(n)} - \beta_k^T X^{(n)})^2}{2\sigma^2} \right)$$

and the log-likelihood as

$$\ell(\theta, \sigma^{\mathbf{2}}) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \frac{e^{\gamma_k^T X^{(n)}}}{\sum_{l=1}^{K} e^{\gamma_l^T X^{(n)}}} \cdot \frac{1}{\sqrt{2\pi}\sigma} \exp\left( - \frac{(Y^{(n)} - \beta_k^T X^{(n)})^2}{2\sigma^2} \right).$$

The values of $\theta$ and $\sigma^2$ which maximize the log-likelihood function cannot be found analytically, so an EM algorithm is typically used to fit the ME model [17]. For $n = 1, \ldots, N$, let $Z_1^{(n)}, \ldots, Z_K^{(n)} \in \{0, 1\}$ such that $\sum_{k=1}^{K} Z_k^{(n)} = 1$ so that only one of the $Z_1^{(n)}, \ldots, Z_K^{(n)}$ is equal to 1 while the rest are equal to 0. These indicator variables are labels that indicate which expert in the model generated the data point.

After dropping constants, we may write the full log-likelihood as

$$\ell_{\mathbf{f}}(\theta, \sigma^{\mathbf{2}}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_k^{(n)} \ln \left[ \frac{e^{\gamma_k^T X^{(n)}}}{\sum_{l=1}^{K} e^{\gamma_l^T X^{(n)}}} \cdot \frac{1}{\sigma} \exp\left( - \frac{(Y^{(n)} - \beta_k^T X^{(n)})^2}{2\sigma^2} \right) \right]. \quad (1)$$

Next, following [6], as the E-step, we take the conditional expectation given $Y$ and $X$ of the full log-likelihood (1), which gives

$$\mathbf{Q}(\theta, \sigma^2) = \sum_{n=1}^{N}\sum_{k=1}^{K} h_k^{(n)} \ln\left[\frac{e^{\gamma_k^T X^{(n)}}}{\sum_{l=1}^{K} e^{\gamma_l^T X^{(n)}}} \cdot \frac{1}{\sigma} \exp\left( - \frac{(Y^{(n)} - \beta_k^T X^{(n)})^2}{2\sigma^2}\right)\right], \quad (2)$$

where

$$h_k^{(n)} = \frac{\exp(\gamma_k^T X^{(n)}) \exp\big( - (Y^{(n)} - \beta_k^T X^{(n)})^2/2\sigma^2\big)}{\sum_{l=1}^{K} \exp(\gamma_l^T X^{(n)}) \exp\big( - (Y^{(n)} - \beta_l^T X^{(n)})^2/2\sigma^2\big)}.$$

To perform the M-step we maximize $\mathbf{Q}(\theta, \sigma^2)$ with respect to $(\theta, \sigma^2)$. We observe that (2) can be decomposed as

$$\mathbf{Q}(\theta, \sigma^2) = \sum_{n=1}^{N}\sum_{k=1}^{K} h_k^{(n)} \ln\left[\frac{e^{\gamma_k^T X^{(n)}}}{\sum_{l=1}^{K} e^{\gamma_l^T X^{(n)}}}\right] - N\ln(\sigma) - \sum_{n=1}^{N}\sum_{k=1}^{K} \frac{h_k^{(n)}(Y^{(n)} - \beta_k^T X^{(n)})^2}{2\sigma^2}, \quad (3)$$

where the first term involves only gating network parameters and the remaining two terms involve only expert parameters. We call the first term the gating network part and the remaining two terms the experts part. In the unpenalized version of ME, we can maximize each part separately using an iteratively reweighted least-squares algorithm.

### *2.2. The group fused lasso penalty term*

First we define two norms that will be used later. Given an arbitrary vector $b = (b_1^T, ..., b_m^T)^T$ where each block $b_i, i = 1, 2, ...m$, has length $2P$, let

$$||b||_{2,1} = \sum_{j=1}^{m} ||b_j||_2 \quad \text{and} \quad ||b||_{2,\infty} = \max_{1\le j\le m} ||b_j||_2.$$

Even though ME is an extremely powerful and flexible model, it can potentially underfit or overfit if there are too few or too many experts in the model. We aim to alleviate this drawback by first initializing the model with a large number of experts and then adding to (3) the penalty term

$$\Omega_\lambda(\theta) = \lambda \sum_{1\le i<j\le K} \left|\left| \begin{pmatrix}\beta_i \\ \gamma_i\end{pmatrix} - \begin{pmatrix}\beta_j \\ \gamma_j\end{pmatrix} \right|\right|_2, \quad (4)$$

where $\lambda$ is a non-negative tuning parameter that controls the regularization level. By initializing the model with a large enough number of experts (the number of experts is a subjective choice; based on our experiments, 6 to 10 experts seemed reasonable for the datasets we considered in Section 5), we can ensure that, for some value of $\lambda$, the penalty will admit a model complex enough to fit the data. Next, we increase $\lambda$ incrementally from 0. As $\lambda$ gets larger, this penalty term will shrink the coefficients of different experts together as well as the coefficients governing the weights assigned to them by the gating network. When $\lambda$ is large enough, pairs of experts and the

corresponding pairs of functions in the gating network assigning weights to them will become identical, preventing overfitting and providing a natural way to choose the appropriate number of experts at the same time. Essentially we are fusing groups of coefficients together.

We find it useful to rewrite the penalty term in the following way. Define the matrix $D = D'C_p$, where $D'$ is the $2P(2K-1) \times 2P(K-1)$ matrix given by

$$D' = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 1 & 0 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 0 & -1 \\ 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 \end{bmatrix} \otimes I_{2P}$$

and $C_p$ is the matrix such that

$$C_p\theta = (\beta_1^T, \gamma_1^T, \beta_2^T, \gamma_2^T, ..., \beta_K^T, \underline{0}^T)^T.$$

Then we may rewrite the penalty in (4) as

$$\Omega_\lambda(\theta) = ||D\theta||_{2,1},$$

which strongly resembles the penalty term used in the generalized lasso [18]. In the next section we will express the log-likelihood in a way that facilitates computation of the penalized maximization step.

## 3. Reformulating the likelihood

Since the M step in (3) can be divided into 2 parts, the experts part and the gating network part, in this section we will deal with each part separately.

### *3.1. The experts part*

We now consider the last term in (3) since it is the only term in (3) that involves $\beta$. We will delay the treatment of $\sigma$ until later since it is relatively easy to find the update for $\sigma$ and it does not appear in the penalty term. We may express the last term in (3) (without the negative sign) as

$$\mathbf{Q_E}(\beta, \boldsymbol{\sigma^2}) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} \sum_{k=1}^{K} h_k^{(n)} (Y^{(n)} - \beta_k^T X^{(n)})^2 = \sum_{k=1}^{K} ||W_k^{\frac{1}{2}} (Y - X\beta_k)||^2,$$

where $W_k = (2\sigma^2)^{-1}\text{diag}(h_k^{(1)}, \ldots, h_k^{(N)})$ for $k = 1, \ldots, K$. Letting

$$Y^* = \begin{pmatrix} W_1^{\frac{1}{2}} Y \\ \vdots \\ W_K^{\frac{1}{2}} Y \end{pmatrix} \text{ and } X^* = \begin{pmatrix} W_1^{\frac{1}{2}} X & & \\ & \ddots & \\ & & W_K^{\frac{1}{2}} X \end{pmatrix},$$

we may write

$$\mathbf{Q_E}(\beta, \sigma^{\mathbf{2}}) = \|Y^* - X^*\beta\|^2.$$

So in the case of no regularization, the EM algorithm update for the expert parameters is

$$\beta^{\text{new}} = \arg\min_{\beta} ||Y^* - X^*\beta||^2. \tag{5}$$

### *3.2. The gating network part*

The first term in (3), the gating network part, is slightly more complicated, since we cannot express the update as the solution to a least-squares problem. We have

$$\mathbf{Q_g}(\gamma) = \sum_{n=1}^{N} \sum_{k=1}^{K} h_k^{(n)} \ln \left[ \frac{e^{\gamma_k^T X^{(n)}}}{1 + \sum_{l=1}^{K-1} e^{\gamma_l^T X^{(n)}}} \right].$$

Maximizing the above term is quite similar to maximizing the likelihood for multinomial logistic regression and Cox regression, so we consider a Newton-Raphson algorithm. The first and second derivatives of $Q_g$ are

$$\begin{aligned} \frac{\partial Q_g}{\partial \gamma_{ij}} &= \sum_{n=1}^{N} x_j^{(n)} \left[ h_i^{(n)} - p_i(x^{(n)}; \gamma) \right] \\ \frac{\partial^2 Q_g}{\partial \gamma_{ij} \partial \gamma_{mp}} &= -\sum_{n=1}^{N} x_j^{(n)} x_p^{(n)} p_i(x^{(n)}; \gamma) \left[ I(i = m) + p_m(x^{(n)}; \gamma) \right], \end{aligned}$$

for $1 \leq i, j, m, p \leq K - 1$, where $p_i(x^{(n)}; \gamma) = e^{\gamma_k^T X^{(n)}} / [1 + \sum_{l=1}^{K-1} e^{\gamma_l^T X^{(n)}}]$. We can express the first and second derivatives in matrix form as

$$\frac{\partial Q_g}{\partial \gamma} = \tilde{X}^T (h - p) \quad \text{and} \quad \frac{\partial^2 Q_g}{\partial \gamma \partial \gamma^T} = -\tilde{X}^T W \tilde{X},$$

where $\tilde{X} = I_{K-1} \otimes X$ and $h = (h_1^T, \ldots, h_{K-1}^T)$ and $p = (p_1^T, \ldots, p_{K-1}^T)$, with

$$\begin{aligned} p_k &= \left( p_k(x^{(1)}; \gamma), \ldots p_k(x^{(N)}; \gamma) \right)^T \\ h_k &= \left( h_k^{(1)}, \ldots, h_k^{(N)} \right)^T, \end{aligned}$$

for $k = 1 \ldots, K-1$, and $W = \big(W_{ij}\big)_{1 \leq i,j \leq K-1}$, where $W_{ij}$ is an $N \times N$ diagonal matrix with diagonal elements given by

$$\begin{cases} p_i * (\mathbf{1}_N - p_i) & \text{if } i = j \\ -p_i * p_j & \text{if } i \neq j, \end{cases}$$

where we denote by $\mathbf{1}_N$ a vector of length $N$ with all entries equal to 1 and by $*$ the element-wise multiplication between two vectors.

It is well-known that we may run into numerical issues when fitting multinomial logistic regression models using Newton-Raphson [19]. It is indeed the case in ME since the matrix $W$ almost always becomes computationally singular after some iterations. This is a problem because we need to invert $W$ to compute the update. Böhning [20] suggests to fix the Hessian $(-\tilde{X}^T W \tilde{X})$, or choose a fixed matrix, say $\overline{W}$, with which to replace $W$ across all Newton-Raphson iterations. More specifically, we need to choose a fixed matrix $\overline{W}$ so that $(-\tilde{X}^T W \tilde{X}) - (-\tilde{X}^T \overline{W} \tilde{X})$ is positive semi-definite. The choice of $\overline{W}$ prescribed by Böhning has entries given by

$$\overline{W}_{ij} = \begin{cases} \left(\frac{1}{2} - \frac{1}{2K}\right) I_N & \text{if} \quad i = j \\ \left(-\frac{1}{2K}\right) I_N & \text{if} \quad i \neq j \end{cases} \qquad \text{for } 1 \leq i, j, \leq K-1.$$

With this choice of $W$, each Newton-Raphson update is (when there is no regularization)

$$\gamma^{new} = \gamma^{old} - \left(\frac{\partial^2 Q_g}{\partial \gamma \partial \gamma^T}\right)^{-1} \frac{\partial Q_g}{\partial \gamma} = \gamma^{old} + \big(\tilde{X}^T \overline{W} \tilde{X}\big)^{-1} \tilde{X}^T (h - p) = \big(\tilde{X}^T \overline{W} \tilde{X}\big)^{-1} \tilde{X}^T \overline{W} t,$$

where $t = \tilde{X}\gamma^{old} + \overline{W}^{-1}(h - p)$. Using the Cholesky decomposition $\overline{W} = LL^T$ of $\overline{W}$, the update $\gamma^{new}$ can be expressed as the least-squares solution

$$\gamma^{new} = \underset{\gamma}{\operatorname{argmin}} ||\tilde{t} - \tilde{X}_2 \gamma||^2, \tag{6}$$

where $\tilde{t} = L^T t$ and $\tilde{X}_2 = L^T \tilde{X}$.

## 4. Penalized ME

In this section, we introduce the penalty term into the reformulated likelihood.

### *4.1. Updating experts and the gating network together*

Combining (5) and (6), we can further simplify the update of all coefficients of the experts and the gating network (again assuming there is no regularization) by writing

$$||Y^* - X^* \beta||^2 + ||\tilde{z} - \tilde{X}_2 \gamma||^2 = \begin{pmatrix} Y^* - X^* \beta \\ \tilde{z} - \tilde{X}_2 \gamma \end{pmatrix}^T \begin{pmatrix} Y^* - X^* \beta \\ \tilde{z} - \tilde{X}_2 \gamma \end{pmatrix} = ||Y^{**} - X^{**} \theta||^2,$$

where

$$Y^{**} = \begin{pmatrix} Y^* \\ \tilde{z} \end{pmatrix} \quad \text{and} \quad X^{**} = \begin{pmatrix} X^* & . \\ . & \tilde{X}_2 \end{pmatrix}.$$

Then we may write

$$\theta^{new} = \arg\min_{\theta} ||Y^{**} - X^{**}\theta||^2. \tag{7}$$

Note that (5) is an update for each EM iteration while (6) is an update for each Newton Raphson iteration. We combine them together into (7) because in the penalty term, we cannot separate the coefficients of the gating network from experts' ones. Therefore we need to optimize both parts jointly. We discuss this in the next subsection.

### *4.2. Adding the penalty term*

With the penalty term (4) added to the model, at each Newton-Raphson iteration in each Maximization step of the EM algorithm, we solve the optimization problem

$$\underset{\theta}{\text{minimize}} \; ||Y^{**} - X^{**}\theta||^2 + \lambda||D\theta||_{2,1}. \tag{8}$$

It is apparent that when $N > P$, $X^{**}$ has full-column rank based on the way we construct $X^{**}$. With this result, we will employ the same strategy as the one in [18] because our penalty is like the group-generalization of the generalized lasso penalty. The minimization problem (8) is equivalent to the problem

$$\underset{\theta}{\text{minimize}} \; ||Y^{**} - X^{**}\theta||^2 + \lambda||z||_{2,1} \quad \text{subject to } z = D\theta.$$

The Lagrangian form is

$$\mathcal{L}(\theta, z, u) = ||Y^{**} - X^{**}\theta||^2 + \lambda||z||_{2,1} + u^T(D\theta - z), \tag{9}$$

and the dual problem for this is

$$\underset{u}{\text{minimize}} \; ||\widetilde{Y} - \widetilde{D}^T u||^2 \text{ subject to } ||u||_{2,\infty} \leq \lambda, \tag{10}$$

where $\widetilde{Y} = X^{**}X^{**+}Y^{**}$ and $\widetilde{D} = DX^{**+}$. Here the pseudoinverse of a matrix $A$ is calculated as $A^+ = (A^TA)^{-1}A^T$. Since we are in the low-dimensional setting where $N > P$, $X^{**+}$ exists.

The dual problem in (10) is convex and thus can be solved using any convex solver. Nonetheless, we develop our own algorithm to solve it using blockwise coordinate descent. We minimize over each block of length $2P$ of $u$, with all other elements of $u$ fixed, until convergence. The update for $u_j$ is given by

$$u_j = T_\lambda\left[\left((\widetilde{D}^T_{.j})^T\widetilde{D}^T_{.j}\right)^{-1}(\widetilde{D}^T_{.j})^T(\widetilde{y} - \widetilde{D}^T_{-.j}u_{-j})\right] \quad \text{for } j = 1, 2, \ldots, \frac{K(K-1)}{2}, \tag{11}$$

where $T_s$ is the truncating function

$$T_s(t) = \begin{cases} s * t/||t||, & \text{if } ||t|| > s \\ t, & \text{if } ||t|| \leq s. \end{cases}$$

Here $u_j$ and $u_{-j}$ denote the $j$th block of $u$ and the vector $u$ after removing the $j$th block, respectively. Similarly, $\widetilde{D}^T_{\cdot j}$ and $\widetilde{D}^T_{-\cdot j}$ represent the $j$th column-block of the matrix $\widetilde{D}^T$ and the whole matrix $\widetilde{D}^T$ after removing the $j$th column-block, respectively. We obtain (11) by first differentiating the least-squares term in (10) with respect to $u_j$, setting the first derivative to 0, and solving for $u_j$. Since we have a box constraint on $u$, we apply the truncating function to this value of $u_j$.

After $u$ has converged, we recover the primal solution via

$$\theta^{new} = X^{**+}(\widetilde{y} - \widetilde{D}^T u^{new}).$$

This primal-dual relationship is derived by taking the gradient of (9) with respect to $\theta$ and setting this equal to 0.

### *4.3. The full algorithm*

We can now put everything together to get the algorithm to fit the penalized ME. Full details are given in Algorithm 1.

**Algorithm 1** The algorithm to fit the penalized mixture of experts model.

Choose a value of lambda $\lambda$
Choose the maximum number of experts $K$
Initialize $\theta, \sigma$ and calculate $h, Y^*, X^*, p, W, z, L, \tilde{z}, \tilde{X}_2, Y^{**}, X^{**}, \widetilde{Y}, \widetilde{D}$
**while** $\theta$ and $\sigma$ are not converged (EM loop) **do**
    **while** $\gamma$ is not converged (Newton-Raphson loop) **do**
        Solve the dual problem using block coordinate descent:

$$\underset{u}{\text{minimize}} \; ||\widetilde{Y} - \widetilde{D}^T u||^2 \text{ subject to } ||u||_{2,\infty} \leq \lambda$$

        With the new $u$, recalculate $\theta, p, W, t, L, \tilde{t}, \tilde{X}_2, Y^{**}, X^{**}, \widetilde{Y}, \widetilde{D}$
    **end while**
    Update $\sigma$

$$\sigma^{new} = \sqrt{\sum_{n=1}^{N}\sum_{k=1}^{K} h_k^{(n)} \left(Y^{(n)} - (\beta_k^{new})^T X^{(n)}\right)^2 \Big/ N} \tag{12}$$

    With the new $\theta$ and $\sigma$, recalculate $h, Y^*, X^*, t, \tilde{t}, Y^{**}, X^{**}, \widetilde{Y}, \widetilde{D}$
**end while**

We obtain the expression in (12) by differentiating (3) with respect to $\sigma$, setting it to 0, and solving for $\sigma$.

### *4.4. Efficient way to initialize the parameters*

The log-likelihood of ME is not convex, meaning that different starting values may affect the final solution. Based on our experiments, choosing small initial values (such as 0.1) for all elements of $\theta$ seems to converge to a good solution in most cases but the computation time may be long.

One way to improve the choice of initial values is to initialize $\beta$ by $k$-means clustering and linear regression with a ridge penalty. Specifically, first we apply $k$-means clustering to the covariate $X$, with $k$ equal to the number of experts we want to fit in ME. Then we fit a linear regression for data points in each cluster with a small ridge penalty (with ridge-penalty tuning parameter equal to, say 0.0001). As the result, we obtain $k$ sets of coefficients from $k$ ridge-penalized linear regressions and we can use those coefficients as initial values for experts in ME. Xing and Hu in [21] show that this strategy speeds up the convergence significantly for the unpenalized ME. The reason we choose to use ridge regression is that the number of observations in each cluster may be less than the number of covariates, making it impossible to fit least-squares linear regression. In the rare case when there is one or more clusters which only contain one class of a categorical covariate (for example: say we have a gender covariate which has 2 classes (male and female). It can happen that after doing clustering, observations in one particular cluster are all males), we fit a ridge linear regression to the whole dataset again with a small penalization parameter. We then use the coefficients of that categorical covariate obtained from the ridge regression fitted on the whole dataset as the initial value for that particular covariate in those clusters.

### *4.5. High-dimensional situation*

So far we have considered only the $N > P$ case. However, the case of $P > N$, or the high-dimensional setting, is becoming increasingly common. When $P > N$, as [18] points out, there is a small complication for the dual problem of (8) since $X^{**}$ is no longer full-column rank. To handle this situation, a quick fix is to modify the penalty term in (4) by the addition of a small ridge penalty so that the penalty becomes

$$\Omega_\lambda^h(\theta) = \lambda||D\theta||_{2,1} + \epsilon||\theta||_2^2, \tag{13}$$

where $\epsilon$ is a small positive constant we choose. With this modified term, the minimization problem in (8) becomes

$$\underset{\theta}{\text{minimize}}\ ||Y^{**} - X^{**}\theta||^2 + \lambda||D\theta||_{2,1} + \epsilon||\theta||_2^2.$$

This is equivalent to minimizing

$$||Y^{***} - X^{***}\theta||^2 + \lambda||D\theta||_{2,1},$$

where $Y^{***} = (Y^{**}, 0)^T$ and $X^{***} = [(X^{**})^T, \sqrt{\epsilon}I]^T$. Since $X^{***}$ has full-column rank, we can proceed with the same strategy discussed in Section 4.2. Besides, we can choose $\epsilon$ to be extremely small so that the difference in solutions between using $\Omega_\lambda^h(\theta)$ and $\Omega_\lambda(\theta)$ is likely to be negligible.

## 5. Numerical results

### *5.1. Illustration of penalty on a simulated data set*

In this section we give a brief illustration of how penalized ME works in a simple example. The data have a single covariate and 3 experts are enough to capture the relationship between the predictor and the response. Nevertheless, we will initialize the model with 6 experts. Then we will incrementally increase the value of $\lambda$ (from 0 to 2.5) to make the coefficients of experts and gating network closer together. As a result, we can see that the regression curve representing the conditional mean of the response given the covariate becomes smoother and smoother. Eventually, it becomes a straight line when $\lambda$ is big enough to make all experts the same. The fitted models are depicted in Figure 1.

**Table 1.** Models to compare with penalized ME, their tuning parameters and R packages.

| **Models** | **Tuning parameters** | **R package** |
|---|---|---|
| Elastic net | $\lambda$ and $\alpha$ | glmnet |
| Decision tree | Complexity | rpart |
| Random forest | Number of variables sampled as candidates at each split | randomForest |
| Gradient boosting | Number of trees | gbm |
| Gaussian process | Kernel | kernlab |

**Table 2.** Test MSE for different models in the original unit.

| | **Boston** | **Galaxy** | **Air** | **Diabetes** | **Prostate** |
|---|---|---|---|---|---|
| **Elastic net** | 18.05 | 810.76 | 391.74 | 3169.72 | 18.45 |
| **Decision tree** | 17.10 | 339.30 | 603.97 | 3592.72 | 43.46 |
| **Random forest** | 8.12 | 233.36 | 240.90 | 3343.69 | 19.91 |
| **Gradient boosting** | 11.21 | 328.38 | 247.33 | 3678.71 | 25.01 |
| **Gaussian process** | 10.45 | 225.37 | 322.18 | 3438.42 | 28.55 |
| **ME** | 10.11 | 326.57 | 304.06 | 3266.83 | 34.87 |
| **Penalized ME** | 10.11 | 324.18 | 303.83 | 3186.38 | 32.39 |

### *5.2. Real-world applications*

In this section we apply our model to the following 5 real-world regression datasets.

- Median housing price in Boston (dimension: 506x13) [22]
- Radial Velocity of Galaxy NGC7531 (dimension: 323x3) [23]
- Air quality (dimension: 111x5) [24]
- Diabetes progression (dimension: 442x10) [25]
- Prostate (dimension: 97x8) [26]

For the second dataset, we remove all incomplete observations. We split each dataset into: training set (70%), validation set (15%) and test set (15%), and we compare the performance of the penalized ME to six other commonly used machine learning models that are listed in Table 1. These methods are tuned using the validation set, except for the elastic net which uses 10-fold cross validation on the combined training and validation set. For the penalized ME, we tune the value of $\lambda$ over the range 0 to 2.5. We then train the model on the combined training and validation set using the chosen tuning parameters and make predictions on the test set.

We will also fit an unpenalized ME with 6 experts to see whether adding a penalty term helps. Table 2 displays MSEs on the testing data sets for different models. As we can see, the concept of no free lunch still applies since there is no method that wins in

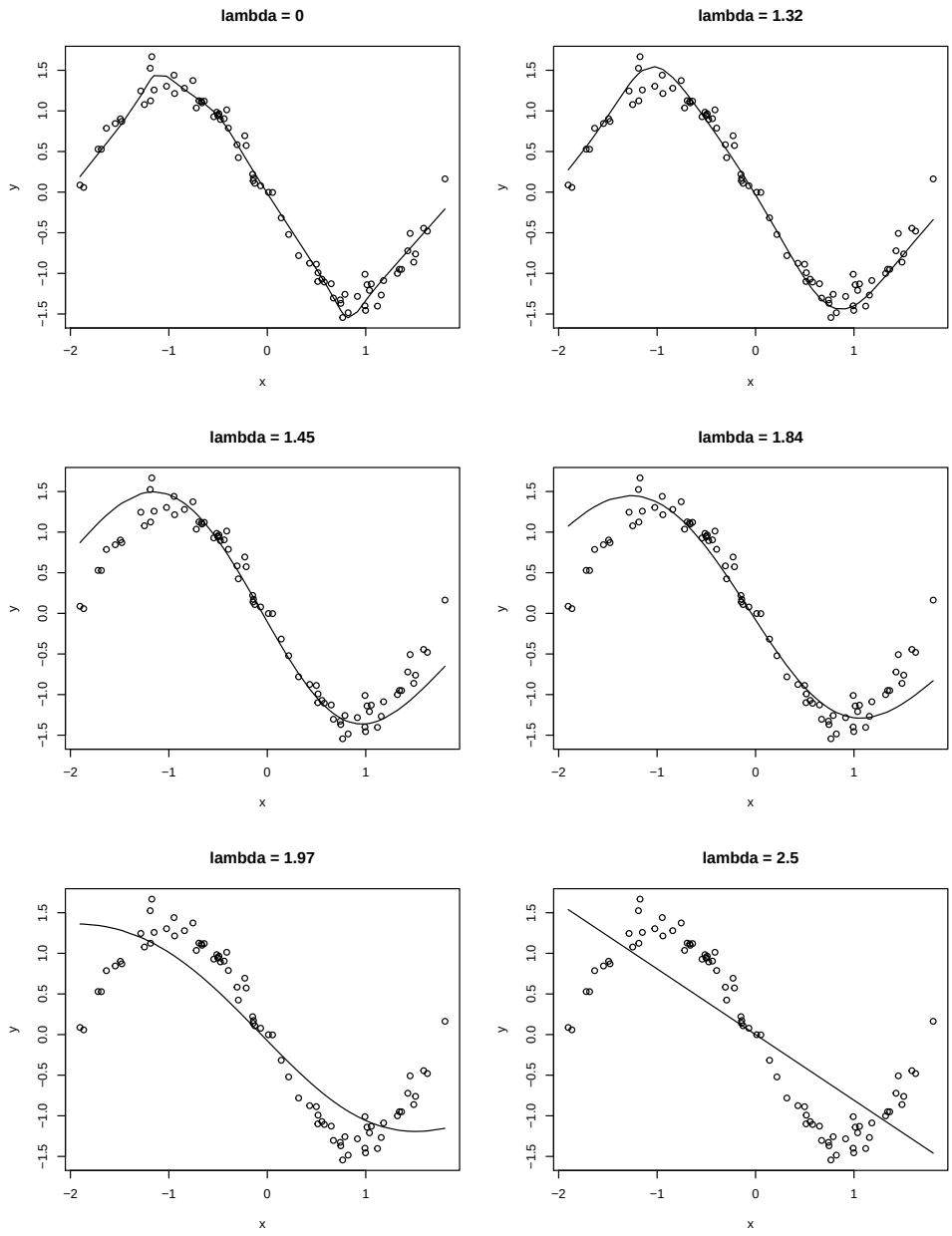


**Figure 1.** Penalized ME curves at different values of the penalty parameter $\lambda$.

all datasets. Nonetheless random forest performs particularly well in almost all cases. The ME and penalized ME also do well compared to other models. Comparing the ME to the penalized ME, we see that in all situations, adding the penalty term improved the prediction accuracy, as the test MSE of the penalized ME was in every case less than or equal to that of the unpenalized ME.

## 6. Conclusion

Mixture of experts is a powerful and flexible machine learning method. In this paper, we have proposed adding a fusion penalty term to the likelihood function with the goal of penalizing the difference between the parameters of different experts. By doing this we can avoid overfitting and choose the best set of parameters at the same time. This has been illustrated in Section 5.2 as penalized ME outperformed the unpenalized version in all data sets considered and also performed competitively when compared to other popular machine learning methods.

## References


[1] Loh WY. Fifty years of classification and regression trees. International Statistical Review.2014;82(3):329–348.

[2] Breiman L. Classification and regression trees. Routledge; 2017.

[3] Bennett KP. Global tree optimization: A non-greedy decision tree algorithm. Computing Science and Statistics. 1994;:156–156.

[4] Norouzi M, Collins M, Johnson MA, et al. Efficient non-greedy optimization of decision trees. In: Advances in Neural Information Processing Systems; 2015. p. 1729–1737. [5] Grubinger T, Zeileis A, Pfeiffer KP. evtree: Evolutionary learning of globally optimal classification and regression trees in r. Working Papers in Economics and Statistics; 2011.

[6] Jordan MI, Jacobs RA. Hierarchical mixtures of experts and the em algorithm. Neural computation. 1994;6(2):181–214.

[7] Jacobs RA, Jordan MI, Nowlan SJ, et al. Adaptive mixtures of local experts. Neural computation. 1991;3(1):79–87.

[8] Friedman JH, et al. Multivariate adaptive regression splines. The annals of statistics.1991;19(1):1–67.

[9] Saito K, Nakano R. A constructive learning algorithm for an hme. In: Proceedings of International Conference on Neural Networks (ICNN'96); Vol. 2; IEEE; 1996. p. 1268–1273.

[10] Fritsch J, Finke M, Waibel A. Adaptively growing hierarchical mixtures of experts. In:Advances in Neural Information Processing Systems; 1997. p. 459–465.

[11] Waterhouse S, Robinson A. Pruning and growing hierachical mixtures of experts. 1995;.

[12] Jacobs RA, Peng F, Tanner MA. A bayesian approach to model selection in hierarchical mixtures-of-experts architectures. Neural Networks. 1997;10(2):231–241.

[13] Rasmussen CE, Ghahramani Z. Infinite mixtures of gaussian process experts. In: Advances in neural information processing systems; 2002. p. 881–888.

[14] Ueda N, Ghahramani Z. Optimal model inference for bayesian mixture of experts. In:Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop (Cat. No. 00TH8501); Vol. 1; IEEE; 2000. p. 145–154.

[15] Kanaujia A, Metaxas D. Learning ambiguities using bayesian mixture of experts. In:2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06);IEEE;

2006. p. 436–440.
[16] Bleakley K, Vert JP. The group fused lasso for multiple change-point detection. arXiv preprint arXiv:11064199. 2011;.
[17] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological). 1977;39(1):1–22.
[18] Tibshirani RJ. The solution path of the generalized lasso. Stanford University; 2011.
[19] Allison PD. Convergence failures in logistic regression. In: SAS Global Forum; Vol. 360;2008. p. 1–11.
[20] Böhning D. Multinomial logistic regression algorithm. Annals of the institute of Statistical Mathematics. 1992;44(1):197–200.
[21] Xing HJ, Hu BG. An adaptive fuzzy c-means clustering-based mixtures of experts model for unlabeled data classification. neurocomputing. 2008;71(4-6):1008–1021.
[22] Harrison Jr D, Rubinfeld DL. Hedonic housing prices and the demand for clean air.Journal of environmental economics and management. 1978;5(1):81–102.
[23] Buta R. The structure and dynamics of ringed galaxies. iii-surface photometry and kine-matics of the ringed nonbarred spiral ngc 7531. The Astrophysical Journal Supplement Series. 1987;64:1–37.
[24] Chambers JM. Graphical methods for data analysis: 0. Chapman and Hall/CRC; 2017.
[25] Efron B, Hastie T, Johnstone I, et al. Least angle regression. The Annals of statistics.2004;32(2):407–499.
[26] Stamey TA, Kabalin JN, McNeal JE, et al. Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. ii. radical prostatectomy treated patients.The Journal of urology. 1989;141(5):1076–1083.