Supplementary Material

Here we outline the FMR and FMRlasso models, how they are estimated, and implementation details used in the simulation study. Given these building blocks, the steps in the estimation algorithm for the MRF-FMRlasso model are summarized.

1. FMR Model

A finite mixture of regression models (FMR) is implemented in the `R` package `mixtools` for maximum likelihood estimation of a mixture regression model via a standard EM algorithm (Benaglia et al. 2009), named regmixEM. We refer to this model as the finite mixture of regression (FMR) model. It does not subject the component coefficient parameters, $\boldsymbol{\beta}_k$, to any form of penalty. It assumes spatial independence within each component and across the component assignments.

The FMR model assumes independence among component assignments and also conditional independence among observations within each component, and we use this as our baseline model for comparison. Let $Y_1, \ldots, Y_n$ be random response variables, each with a corresponding vector of known predictors, $\mathbf{x}_1, \ldots, \mathbf{x}_n$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})'$ for $i = 1, \ldots, n$. Let $\mathbf{Y} = (Y_1, \ldots, Y_n)'$, and let $\mathbf{X}$ be the $n \times p$ matrix of observed predictor values. Suppose that each variable, Y_i, is generated under one of K components. Conditional on membership in the kth component, $k = 1, \ldots, K$, the relationship between Y_i and $\mathbf{x}_i$ is the typical linear regression model $Y_i = \mathbf{x}_i'\boldsymbol{\beta}_k + \epsilon_i$, with $\epsilon_i \sim \mathcal{N}(0, \sigma_k^2)$, where $\boldsymbol{\beta}_k$ and σ_k^2 are the p-dimensional vector of regression coefficients and the error variance for component k, respectively. Accounting for the mixture structure, the conditional density of $Y_i|\mathbf{x}_i$ is

$$f(y_i|\mathbf{x}_i, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \cdot \phi(y_i|\mathbf{x}_i'\boldsymbol{\beta}_k, \sigma_k^2),$$

where π_k is the proportion of observations from the kth component, and $\phi(\cdot|\mathbf{x}_i'\boldsymbol{\beta}_k, \sigma_k^2)$ is the normal density with mean $\mathbf{x}_i'\boldsymbol{\beta}_k$ and variance σ_k^2. The parameter vector for this model is $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_k; \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k; \sigma_1^2, \ldots, \sigma_k^2)'$. Assuming independence among observations, the joint density of $\mathbf{Y}$ is $\prod_{i=1}^{n} f(y_i|\mathbf{X}, \boldsymbol{\theta})$, and thus the observed log-likelihood is

$$\ell(\boldsymbol{\theta}|\mathbf{y}, \mathbf{X}) = \sum_{i=1}^{n} \log\left(f(y_i|\mathbf{X}, \boldsymbol{\theta})\right) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \pi_k \phi(y_i|\mathbf{x}_i'\boldsymbol{\beta}_k, \sigma_k^2)\right). \tag{1}$$

Maximum likelihood estimation consists in finding $\hat{\boldsymbol{\theta}} = \text{argmin}_{\boldsymbol{\theta}\in\Theta} \{-\ell(\boldsymbol{\theta}|\mathbf{y}, \mathbf{X})\}$, where Θ is the set of all possible parameter values. Calculating $\hat{\boldsymbol{\theta}}$ for a finite regression mixture model is known to be a difficult problem (Dempster et al. 1977; McLachlan and Peel 2004), and it is helpful to consider the observation y_i as a partial observation of the *complete data*, which includes the unobservable random vector $\mathbf{Z}_i = (Z_{i1}, \ldots, Z_{iK})'$, where $Z_{ik} \in \{0, 1\}$ is a Bernoulli random variable indicating whether observation i comes from component k or not. Each $\mathbf{Z}_i$ therefore follows a onetrial multinomial distribution with K groups. Since each observation comes from exactly one component, this implies $\sum_{k=1}^{K} Z_{ik} = 1$, so $P(Z_{ik} = 1) = \pi_k$, and $(Y_i|Z_{ik} = 1, \mathbf{X}) \sim \mathcal{N}(\mathbf{x}_i'\boldsymbol{\beta}_k, \sigma_k^2)$, $k = 1, \ldots, K$.

The complete-data distribution for one observation is then

$$P(y_i, \mathbf{z}_i|\mathbf{X}, \boldsymbol{\theta}) = f(y_i|\mathbf{z}_i, \mathbf{X}, \boldsymbol{\theta})P(\mathbf{z}_i|\mathbf{X}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \mathbb{I}_{\{z_{ik}=1\}}\pi_k\phi(y_i|\mathbf{x}_i'\boldsymbol{\beta}_k, \sigma_k^2),$$

where $\mathbb{I}_{\{\cdot\}}$ is the indicator function. Thus, $P(y_i, \mathbf{z}_i|\mathbf{X}, \boldsymbol{\theta}) = \pi_k\phi(y_i|\mathbf{x}_i'\boldsymbol{\beta}_k, \sigma_k^2)$ when $z_{ik} = 1$. With the assumptions of independence among the component assignments, $\mathbf{Z}_1, \ldots, \mathbf{Z}_n$, and conditional independence among $Y_1, \ldots, Y_n$ given $\mathbf{Z}$ and the predictors $\mathbf{X}$, the complete joint density is $\prod_{i=1}^{n} P(y_i, \mathbf{z}_i|\mathbf{X}, \boldsymbol{\theta})$, and thus the complete log-likelihood is

$$\begin{aligned}\ell(\boldsymbol{\theta}|\mathbf{y}, \mathbf{z}, \mathbf{X}) &= \sum_{i=1}^{n} \log\left(P(y_i, \mathbf{z}_i|\mathbf{X}, \boldsymbol{\theta})\right) \\ &= \sum_{i=1}^{n} \log\left(\sum_{k=1}^{K} \mathbb{I}_{\{z_{ik}=1\}}\pi_k\phi(y_i|\mathbf{x}_i'\boldsymbol{\beta}_k, \sigma_k^2)\right).\end{aligned}$$

To estimate the parameters in this model, the EM algorithm is used. Starting from an arbitrary value of the parameters, $\boldsymbol{\theta}^{(0)}$, the principle of the EM algorithm is to iteratively build a sequence of estimates, $\hat{\boldsymbol{\theta}}^{(1)}, \ldots, \hat{\boldsymbol{\theta}}^{(m)}$, over which the *observed* negative log-likelihood, $-\ell(\boldsymbol{\theta}|\mathbf{y}, \mathbf{X})$, monotonically decreases. This is achieved by choosing at iteration $m + 1$ the value $\boldsymbol{\theta}^{(m+1)}$ that minimizes $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) \equiv \mathbb{E}[-\ell(\boldsymbol{\theta}|\mathbf{y}, \mathbf{z}, \mathbf{X})|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}^{(m)}]$, the conditional expectation of the negative *complete* log-likelihood given the observed data and the current parameter value, $\boldsymbol{\theta}^{(m)}$, at iteration m. With some basic manipulation, this can be written as follows:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) = -\sum_{i=1}^{n}\sum_{k=1}^{K} \log(\pi_k)\mathbb{E}\left[\mathbb{I}_{\{z_{ik}=1\}}\middle|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}^{(m)}\right] - \sum_{i=1}^{n}\sum_{k=1}^{K} \log\left(\phi(y_i|\mathbf{x}_i'\boldsymbol{\beta}_k, \sigma_k^2)\right)\mathbb{E}\left[\mathbb{I}_{\{z_{ik}=1\}}\middle|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}^{(m)}\right]. \quad (2)$$

The EM algorithm is broken into two steps:

1. E-Step (Expectation): compute parts of $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ that do not depend on $\boldsymbol{\theta} = (\pi_1, ..., \pi_k; \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k; \sigma_1^2, \ldots, \sigma_k^2)$.
2. M-Step (Minimization): set $\boldsymbol{\theta}^{(m+1)} = \text{argmin}_{\boldsymbol{\theta}\in\Theta}\{Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})\}$.

From $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$, we see that the E-Step consists of computing

$$\mathbb{E}\left[\mathbb{I}_{\{z_{ik}=1\}}\middle|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}^{(m)}\right] = P(Z_{ik} = 1|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}^{(m)}) = P(\mathbf{Z}_i = \mathbf{e}_k|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}^{(m)})$$

for $k = 1, \ldots, K$ and $i = 1, \ldots, n$, where $\mathbf{e}_k$ is a $K \times 1$ vector with kth entry equal to 1, and all other entries equal to 0.

Since the ith component assignment depends only on the ith observation, $(y_i, \mathbf{x}_i)$, we have

$$\begin{aligned} P(\mathbf{Z}_i = \mathbf{e}_k|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}^{(m)}) &= P(\mathbf{Z}_i = \mathbf{e}_k|y_i, \mathbf{x}_i, \boldsymbol{\theta}^{(m)}) \\ &= \frac{f(y_i|\mathbf{Z}_i = \mathbf{e}_k, \mathbf{x}_i, \boldsymbol{\theta}^{(m)})P(\mathbf{Z}_i = \mathbf{e}_k|\mathbf{x}_i, \boldsymbol{\theta}^{(m)})}{f(y_i|\mathbf{x}_i, \boldsymbol{\theta}^{(m)})} \qquad (by\ Bayes'\ Rule) \\ &= \frac{\pi_k^{(m)}\phi\left(y_i|\mathbf{x}_i'\boldsymbol{\beta}_k^{(m)}, \sigma_k^{2\ (m)}\right)}{\sum_{j=1}^K \pi_j^{(m)}\phi\left(y_i|\mathbf{x}_i'\boldsymbol{\beta}_j^{(m)}, \sigma_j^{2\ (m)}\right)}. \end{aligned} \tag{3}$$

Defining $\gamma_{ik}^{(m)}$ to be the expression in Equation (3), the E-step reduces to computing values $\gamma_{ik}^{(m)}$, which is the posterior probability (conditional on the observed data $\mathbf{y}$, $\mathbf{X}$, and $\boldsymbol{\theta}^{(m)}$) of the ith observation's membership in component k.

After the E-step, the values in Equation (3) are substituted into $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$, and minimizing yields the following updates:

$$\pi_k^{(m+1)} = \frac{1}{n}\sum_{i=1}^n \gamma_{ik}^{(m)} \tag{4}$$

$$\boldsymbol{\beta}_k^{(m+1)} = \left(\mathbf{X}'\mathbf{W}_k^{(m)}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}_k^{(m)}\mathbf{y} \tag{5}$$

$$\sigma_k^{2\ (m+1)} = \frac{\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}_k^{(m+1)}\right)'\mathbf{W}_k^{(m)}\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}_k^{(m+1)}\right)}{\mathrm{tr}(\mathbf{W}_k^{(m)})}, \tag{6}$$

where $\mathbf{W}_k^{(m)} = \mathrm{diag}(\gamma_{1k}^{(m)}, \ldots, \gamma_{nk}^{(m)})$. Equation (5) is a weighted least squares (WLS) estimate of $\boldsymbol{\beta}_k$, and Equation (6) resembles the WLS variance estimate.

The E-step and M-step are then iterated through until the absolute change in the value of the parameters is less than a specified threshold, here 10^{-8}, and the algorithm is initiated from a random parameter value. In particular, $\boldsymbol{\pi}^{(0)} = (\pi_1^{(0)}, \ldots, \pi_K^{(0)})$ is drawn from a uniform Dirichlet; $\boldsymbol{\beta}_1^{(0)}, \ldots, \boldsymbol{\beta}_K^{(0)}$ each have standard normal entries; and the inverses of $\sigma_1^{2\ (0)}, \ldots, \sigma_K^{2\ (0)}$ are each generated from a standard exponential. After convergence, observations are then assigned to the component under which they have the greatest probability of being generated (i.e., $z_{ik} = 1$ if $k = \mathrm{argmax}\{\hat{\gamma}_{i1} \ldots \hat{\gamma}_{iK}\}$).

2. FMRlasso Estimation Algorithm

FMRlasso, developed by Städler et al. (2010), is presented, and it introduces a penalty on the component parameters, $\boldsymbol{\beta}_k$. It assumes spatial independence within each component and across the component assignments. Städler et al. (2010) propose the FMRlasso algorithm for fitting a model that addresses the issue of variable selection via a lasso penalization. In a non-mixture Gaussian linear model, $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_n)$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix, the lasso estimator is obtained as

$$\hat{\boldsymbol{\beta}} = \mathrm{argmin}_{\boldsymbol{\beta}}\ \{-\ell(\boldsymbol{\beta}|\mathbf{Y}, \mathbf{X}) + \lambda||\boldsymbol{\beta}||_1\} = \mathrm{argmin}_{\boldsymbol{\beta}}\ \{||\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_1\}.$$

If an estimate of σ^2 is also needed, then the following would be required:

$$\{\hat{\boldsymbol{\beta}}, \hat{\sigma}^2\} = \text{argmin}_{\{\boldsymbol{\beta},\sigma^2\}} \left\{ n\log(\sigma) + \frac{1}{2\sigma^2}||\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_1 \right\}. \tag{7}$$

In this case, only $\boldsymbol{\beta}$ is penalized, but the variance parameter estimate $\hat{\sigma}^2$, is indirectly influenced by the shrinkage parameter λ. For instance, a stricter penalty, λ, will result in a more sparse estimate of $\boldsymbol{\beta}$, which will explain less of the variability in $\mathbf{Y}$, thus increasing the variance estimate, $\hat{\sigma}^2$. There are two issues with the estimator in Equation (7): 1) the function to optimize is non-convex, which diminishes the computational advantages of lasso for high-dimensional problems, and 2) the estimator is not equivariant under scaling of the response.

To address these issues, Städler et al. (2010) propose using the penalty $\lambda\frac{||\beta||_1}{\sigma}$, leading to the estimator

$$\{\hat{\boldsymbol{\beta}}, \hat{\sigma}^2\} = \text{argmin}_{\{\boldsymbol{\beta},\sigma^2\}} \left\{ n\log(\sigma) + \frac{1}{2\sigma^2}||\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda\frac{||\boldsymbol{\beta}||_1}{\sigma} \right\}.$$

This estimator is equivariant under scaling, and it penalizes both the ℓ_1-norm of the coefficients and small variances σ^2 simultaneously. As a consequence, the model is penalized for choosing many components with very few observations in each component. Additionally, convexity of the optimization problem can be achieved with the re-parameterization $\rho = \frac{1}{\sigma}$ and $\varphi_j = \frac{\beta_j}{\sigma}$ for $j = 1, \ldots, p$. This yields the following estimator, which is equivariant under scaling and whose computation involves convex optimization,

$$\{\hat{\boldsymbol{\varphi}}, \hat{\rho}\} = \text{argmin}_{\{\boldsymbol{\varphi},\rho\}} \left\{ -n\log(\rho) + \frac{1}{2}||\rho\mathbf{Y} - \mathbf{X}\boldsymbol{\varphi}||^2 + \lambda||\boldsymbol{\varphi}||_1 \right\}. \tag{8}$$

For the mixture regression model, the parameter vector is $\boldsymbol{\theta} = (\pi_1, ..., \pi_k; \boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_k; \rho_1, \ldots, \rho_k)'$, and the objective is to minimize the negative observed log-likelihood, $-\ell(\boldsymbol{\theta}|\mathbf{y}, \mathbf{X})$ of Equation (1), subject to penalization as

$$\hat{\boldsymbol{\theta}} = \text{argmin}_{\boldsymbol{\theta}\in\Theta} \left\{ -\sum_{i}^{n} \log\left(\sum_{k=1}^{K} \pi_k \frac{\rho_k}{\sqrt{2\pi}} \exp\left\{ -\frac{1}{2}(\rho_k y_i - \mathbf{x}_i'\boldsymbol{\varphi}_k)^2 \right\} \right) + \lambda \sum_{k=1}^{K} \pi_k^{\omega} ||\boldsymbol{\varphi}_k||_1 \right\}, \tag{9}$$

where the term π_k^{ω} is included to control to what extent the penalty depends on the expected proportion of observations within each component. We set $\omega = 1$ throughout, and this implies that a component with greater probability is subject to a stronger penalty, which serves to prevent identifying one overly complex component.

Städler et al. (2010) test a grid of candidate values for λ and choose the one that minimizes $\text{BIC} = -2\ell(\hat{\boldsymbol{\theta}}|\mathbf{y}, \mathbf{X}) + \log(n)d_e$, where $d_e = K + (K - 1) + \sum_{j=1}^{p} \sum_{k=1}^{K} \mathbb{I}_{\{\varphi_{jk}\neq 0\}}$ is the effective number of parameters (Pan and Shen 2007). Städler et al. (2010) also suggest selecting λ via cross-validation. In a simulation study with K assumed to be known, they found that BIC and 10-fold cross-validation performed similarly.

To optimize the criterion in Equation (9), Städler et al. (2010) propose the FMRlasso algorithm, which is a generalized EM (GEM) algorithm. Starting from an arbitrary value of the parameter, $\boldsymbol{\theta}^{(0)}$, they seek to iteratively build a sequence of estimates, $\hat{\boldsymbol{\theta}}^{(1)}, \ldots, \hat{\boldsymbol{\theta}}^{(m)}$, over which the observed penalized negative log-likelihood, $-\ell_{\text{pen},\lambda}(\boldsymbol{\theta})$, monotonically decreases. This is achieved by choosing at iteration $m+1$ a value $\boldsymbol{\theta}^{(m+1)}$ that improves

$$Q_{\text{pen}}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) = Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) + \lambda \sum_{k=1}^{K} \pi_k^{\omega} ||\boldsymbol{\varphi}_k||, \tag{10}$$

where $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ is the expected negative complete data log-likelihood in Equation (2).

The GEM algorithm is broken into two steps:

1. E-Step (Expectation): compute parts of $Q_{\text{pen}}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ that do not depend on $\boldsymbol{\theta} = (\pi_1, ..., \pi_k; \boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_k; \rho_1^2, \ldots, \rho_k^2)$.

2. Generalized M-Step: Improve $Q_{\text{pen}}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ w.r.t. $\boldsymbol{\theta} \in \Theta$,

This is a "generalized" EM algorithm because, rather than optimizing $Q_{\text{pen}}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ w.r.t. $\boldsymbol{\theta} \in \Theta$ at each iteration, here we simply seek to improve it.

E-step:

The addition of the penalty term in Equation (10) does not affect the E-step, which is the same as the FMR E-step in Equation (3), so with reparameterization we compute

$$\gamma_{ik}^{(m)} \equiv P(\mathbf{Z}_i = \mathbf{e}_k|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}^{(m)}) = \frac{\pi_k^{(m)} \rho_k^{(m)} e^{-\frac{1}{2}(\rho_k^{(m)} y_i - \mathbf{x}_i' \boldsymbol{\varphi}_k^{(m)})^2}}{\sum_{j=1}^{K} \pi_j^{(m)} \rho_j^{(m)} e^{-\frac{1}{2}(\rho_j^{(m)} y_i - \mathbf{x}_i' \boldsymbol{\varphi}_j^{(m)})^2}},$$

for $k = 1, \ldots, K$; $i = 1, \ldots, n$.

Generalized M-step:

After the E-step we have

$$Q_{\text{pen}}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)}) = -\sum_{i=1}^{n}\sum_{k=1}^{K} \log(\pi_k)\gamma_{ik}^{(m)} - \sum_{i=1}^{n}\sum_{k=1}^{K} \log\left(\phi(y_i|\mathbf{x}_i'\boldsymbol{\beta}_k, \sigma_k^2)\right)\gamma_{ik}^{(m)} + \lambda \sum_{k=1}^{K} \pi_k \frac{||\boldsymbol{\beta}_k||_1}{\sigma_k}.$$

Then, we obtain $Q_{\text{pen}}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$

$$\begin{aligned}
&= -\sum_{i=1}^{n}\sum_{k=1}^{K} \gamma_{ik}^{(m)} \log(\pi_k) - \sum_{i=1}^{n}\sum_{k=1}^{K} \gamma_{ik}^{(m)} \log\left(\frac{1}{\sigma_k\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{y_i - \mathbf{x}_i'\boldsymbol{\beta}_k}{\sigma_k}\right)^2\right\}\right) + \lambda \sum_{k=1}^{K} \pi_k \frac{||\boldsymbol{\beta}_k||_1}{\sigma_k} \\
&= -\sum_{i=1}^{n}\sum_{k=1}^{K} \gamma_{ik}^{(m)} \log(\pi_k) - \sum_{i=1}^{n}\sum_{k=1}^{K} \gamma_{ik}^{(m)} \log\left(\frac{\rho_k}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(\rho_k y_i - \mathbf{x}_i'\boldsymbol{\varphi}_k)^2\right\}\right) + \lambda \sum_{k=1}^{K} \pi_k ||\boldsymbol{\varphi}_k||_1
\end{aligned}$$

(a) *Improvement of* $Q_{pen}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ *w.r.t.* $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$:

Fix $\boldsymbol{\varphi}$ at the present value $\boldsymbol{\varphi}^{(m)}$ and improve

$$-\sum_{i=1}^{n}\sum_{k=1}^{K}\gamma_{ik}^{(m)}\log(\pi_k) + \lambda\sum_{k=1}^{K}\pi_k\|\boldsymbol{\varphi}_k^{(m)}\|_1, \tag{11}$$

w.r.t. the condition that $\pi_k > 0$ for $k = 1, \ldots, K$ and $\sum_{k=1}^{K}\pi_k = 1$. To do so, Städler et al. (2010) propose the following update. First, let

$$\bar{\boldsymbol{\pi}}^{(m+1)} = \frac{\sum_{i=1}^{n}\boldsymbol{\gamma}_i^{(m)}}{n},$$

where $\boldsymbol{\gamma}_i^{(m)} = (\gamma_{i1}^{(m)}, \ldots, \gamma_{iK}^{(m)})'$. The value $\bar{\boldsymbol{\pi}}^{(m+1)}$ is simply the update used in FMR, Equation (5). To account for the added penalty term in Equation (11), they suggest

$$\boldsymbol{\pi}^{(m+1)} = (1 - t^{(m)})\boldsymbol{\pi}^{(m)} + t^{(m)}\bar{\boldsymbol{\pi}}^{(m+1)}, \tag{12}$$

where $t^{(m)} \in (0, 1]$. In practice, $t^{(m)}$ is chosen to be the largest value in the grid $\{\delta^j; j = 0, 1, 2, \ldots\}$ $(0 < \delta < 1)$ such that the value of Equation (11) is not increased. They find that $\delta = 0.1$ worked well in their example, and we follow this approach in our implementation. Thus, the update is a weighted combination of $\bar{\boldsymbol{\pi}}^{(m)}$ and $\boldsymbol{\pi}^{(m)}$. If the update $\boldsymbol{\pi}^{(m+1)} = \bar{\boldsymbol{\pi}}^{(m+1)}$ does not lead to an increase in (11), they will use this value. If it does lead to an increase, then they will give more and more weight to the previous value, $\boldsymbol{\pi}^{(m)}$, in Equation (12) until the computed value does not increase the value of Equation (11).

(b) *Coordinate descent improvement of* $Q_{pen}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ *w.r.t.* $\boldsymbol{\rho}$ *and* $\boldsymbol{\varphi}$:

To obtain estimates of $\boldsymbol{\varphi}_k$ and ρ_k for $k = 1, \dots, K$, we first show how the expected complete penalized log likelihood decouples into K distinct optimization problems. With some re-arranging, we have $Q_{\text{pen}}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$

$$
\begin{aligned}
&= -\sum_{i=1}^{n}\sum_{k=1}^{K}\gamma_{ik}^{(m)}\log(\pi_k) - \sum_{i=1}^{n}\sum_{k=1}^{K}\gamma_{ik}^{(m)}\log\left(\frac{\rho_k}{\sqrt{2\pi}}\exp\left\{-\frac{1}{2}(\rho_k y_i - \mathbf{x}_i'\boldsymbol{\varphi}_k)^2\right\}\right) + \lambda\sum_{k=1}^{K}\pi_k\|\boldsymbol{\varphi}_k\|_1 \\
&= \sum_{k=1}^{K}\left\{-\sum_{i=1}^{n}\gamma_{ik}^{(m)}\log(\pi_k) - \sum_{i=1}^{n}\gamma_{ik}^{(m)}\log\left(\frac{\rho_k}{\sqrt{2\pi}}\exp\left\{-\frac{1}{2}(\rho_k y_i - \mathbf{x}_i'\boldsymbol{\varphi}_k)^2\right\}\right) + \lambda\pi_k\|\boldsymbol{\varphi}_k\|_1\right\} \\
&= \sum_{k=1}^{K}\left\{-\log(\pi_k)\sum_{i=1}^{n}\gamma_{ik}^{(m)} - \sum_{i=1}^{n}\left\{\gamma_{ik}^{(m)}\log\left(\frac{\rho_k}{\sqrt{2\pi}}\right) - \frac{\gamma_{ik}^{(m)}}{2}(\rho_k y_i - \mathbf{x}_i'\boldsymbol{\varphi}_k)^2\right\} + \lambda\pi_k\|\boldsymbol{\varphi}_k\|_1\right\} \\
&= \sum_{k=1}^{K}\left\{-\log(\pi_k)\sum_{i=1}^{n}\gamma_{ik}^{(m)} - \sum_{i=1}^{n}\gamma_{ik}^{(m)}\log\left(\frac{\rho_k}{\sqrt{2\pi}}\right) + \sum_{i=1}^{n}\frac{\gamma_{ik}^{(m)}}{2}(\rho_k y_i - \mathbf{x}_i'\boldsymbol{\varphi}_k)^2 + \lambda\pi_k\|\boldsymbol{\varphi}_k\|_1\right\} \\
&= \sum_{k=1}^{K}\left\{-\log(\pi_k)\sum_{i=1}^{n}\gamma_{ik}^{(m)} - \sum_{i=1}^{n}\gamma_{ik}^{(m)}\left(\log(\rho_k) - \log(\sqrt{2\pi})\right) + \sum_{i=1}^{n}\frac{\gamma_{ik}^{(m)}}{2}(\rho_k y_i - \mathbf{x}_i'\boldsymbol{\varphi}_k)^2 + \lambda\pi_k\|\boldsymbol{\varphi}_k\|_1\right\} \\
&= \sum_{k=1}^{K}\left\{-\log(\pi_k)\sum_{i=1}^{n}\gamma_{ik}^{(m)} - \log(\rho_k)\sum_{i=1}^{n}\gamma_{ik}^{(m)} + \log(\sqrt{2\pi})\sum_{i=1}^{n}\gamma_{ik}^{(m)} + \sum_{i=1}^{n}\frac{\gamma_{ik}^{(m)}}{2}(\rho_k y_i - \mathbf{x}_i'\boldsymbol{\varphi}_k)^2 + \lambda\pi_k\|\boldsymbol{\varphi}_k\|_1\right\} \\
&= \sum_{k=1}^{K}\left\{n_k\left(-\log(\pi_k) - \log(\rho_k) + \log(\sqrt{2\pi})\right) + \sum_{i=1}^{n}\frac{1}{2}(\rho_k\tilde{y}_i - \tilde{\mathbf{x}}_i'\boldsymbol{\varphi}_k)^2 + \lambda\pi_k\|\boldsymbol{\varphi}_k\|_1\right\} \\
&= \sum_{k=1}^{K}\left\{n_k\left(-\log(\pi_k) - \log(\rho_k) + \log(\sqrt{2\pi}) + \frac{1}{2n_k}\|\rho_k\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\varphi}_k\|^2 + \frac{\lambda}{n_k}\pi_k\|\boldsymbol{\varphi}_k\|_1\right)\right\},
\end{aligned}
$$

where $n_k = \sum_{i=1}^{n}\gamma_{ik}^{(m)}$, $\tilde{y}_i = \sqrt{\gamma_{ik}^{(m)}}y_i$, $\tilde{\mathbf{x}}_i = \sqrt{\gamma_{ik}^{(m)}}\mathbf{x}_i$. Thus, the M-step decouples into K distinct optimization problems of the form

$$
n_k\left(-\log(\pi_k) - \log(\rho_k) + \log(\sqrt{2\pi}) + \frac{1}{2n_k}\|\rho_k\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\varphi}_k\|^2 + \frac{\lambda}{n_k}\pi_k\|\boldsymbol{\varphi}_k\|_1\right) \quad \text{for } k = 1, \dots, K.
$$

Fixing $\boldsymbol{\pi}$ to the updated value, $\boldsymbol{\pi}^{(m+1)}$, obtained in the previous step and minimizing $Q_{\text{pen}}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(m)})$ with respect to $\boldsymbol{\varphi}$ and $\boldsymbol{\rho}$ reduces to minimizing

$$
-\log(\rho_k) + \frac{1}{2n_k}\|\rho_k\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\varphi}_k\|^2 + \frac{\lambda}{n_k}\left(\pi_k^{(m+1)}\right)\|\boldsymbol{\varphi}_k\|_1 \quad \text{for } k = 1, \dots, K. \tag{13}
$$

Equation (13) is convex in the parameters $(\rho_k, \varphi_{k1}, \dots, \varphi_{kp})$. Instead of fully optimizing Equation (13), Städler et al. (2010) only minimize with respect to each parameter one at a time starting with ρ_k, holding the other parameters at their most recent values. Making use of the Karush-Kuhn-Tucker (KKT) conditions (i.e., necessary conditions for a solution in nonlinear programming to be optimal), closed-form updates can be computed for each component $k = 1, \dots, K$ as

$$\rho_k^{(m+1)} = \frac{\tilde{\mathbf{y}}'\tilde{\mathbf{X}}\boldsymbol{\varphi}_k^{(m)} + \sqrt{\left(\tilde{\mathbf{y}}'\tilde{\mathbf{X}}\boldsymbol{\varphi}_k^{(m)}\right)^2 + 4||\tilde{\mathbf{y}}||^2 n_k}}{2||\tilde{\mathbf{y}}||^2} \tag{14}$$

$$\varphi_{kj}^{(m+1)} = \begin{cases} 0 & \text{if } |S_j| \leq \lambda(\pi_k^{(m+1)}) \\ (\lambda(\pi_k^{(m+1)}) - S_j)/||\tilde{\mathbf{X}}_j||^2 & \text{if } S_j > \lambda(\pi_k^{(m+1)}) \\ -(\lambda(\pi_k^{(m+1)}) + S_j)/||\tilde{\mathbf{X}}_j||^2 & \text{if } S_j < -\lambda(\pi_k^{(m+1)}), \end{cases} \tag{15}$$

where S_j is defined as

$$S_j = -\rho_k^{(m+1)}\tilde{\mathbf{X}}_j'\tilde{\mathbf{y}} + \sum_{s<j} \varphi_{ks}^{(m+1)}\tilde{\mathbf{X}}_j'\tilde{\mathbf{X}}_s + \sum_{s>j} \varphi_{ks}^{(m)}\tilde{\mathbf{X}}_j'\tilde{\mathbf{X}}_s, \qquad j = 1, \ldots, p,$$

where p is the number of predictors, and $\tilde{\mathbf{X}}_j$ denotes the jth column of $\tilde{\mathbf{X}}$. The piecewise expression in Equation (15) arises from differentiating absolute value functions, i.e., the $|\varphi_{kj}|$ terms in the penalty, and as a consequence of the KKT conditions. In the definition of S_j, the sum is split over $s < j$ and $s > j$ because we include the most recent values for each parameter as we progress from $j = 1$ to $j = p$.

The E-step and M-step are alternated between until convergence. Following Städler et al. (2010), we stop the algorithm if the relative penalized log-likelihood improvement and the relative change of the parameter vector are small enough, namely

$$\begin{aligned} &\frac{|\ell_{\text{pen},\lambda}(\boldsymbol{\theta})^{(m+1)} - \ell_{\text{pen},\lambda}(\boldsymbol{\theta})^{(m)}|}{1 + |\ell_{\text{pen},\lambda}(\boldsymbol{\theta})^{(m+1)}|} \leq \tau, \\ &\max_j \left\{ \frac{|\theta_j^{(m+1)} - \theta_j^{(m)}|}{1 + |\theta_j^{(m+1)}|} \right\} \leq \sqrt{\tau}, \qquad \tau = 10^{-6}. \end{aligned} \tag{16}$$

We also follow the approach of Städler et al. (2010) for initializing the algorithm. In particular, for each observation i, $i = 1, \ldots, n$, a component $k \in \{1, \ldots, K\}$ is randomly assigned. That component is then given weight $\gamma_{ik}^{(0)} = 0.9$, and weights $\gamma_{ij}^{(0)} = 0.1$ are assigned to all other components. Finally, the weights $\gamma_{ij}^{(0)}$, $j = 1, \ldots, K$, are normalized to sum to one. This can be viewed as an initialization of the E-step. In the M-step that follows, we update all parameters from the initial values $\boldsymbol{\varphi}_k^{(0)} = \mathbf{0}, \rho_k^{(0)} = 2, \pi_k^{(0)} = \frac{1}{K}, k = 1, \ldots, K$.

2.1. Active Set Selection

Städler et al. (2010) propose a simple approach to speed up the algorithm described above. When updating the parameter φ_{kj} in the M-step (b), for every 10 EM-iterations they only update the current non-zero parameters (the "active set"). All parameters are updated every 11th EM-iteration, which allows for periodic updates of which parameters are included in the active set. Städler et al. (2010) claim that in very high-dimensional and sparse settings, this leads to a remarkable decrease in computational times.

3. MRF-FMRlasso Estimation Algorithm

The MRF-FMRlasso algorithm is summarized by the following steps:

1. **E-step**:

   (a) Compute values $\check{\mathbf{z}}_i^{(m)}$ for $i = 1, \dots, n$:

   - *Simulated field approximation*: $\check{\mathbf{z}}_i^{(m)}$ is simulated from $P(\mathbf{z}|\mathbf{y}, \mathbf{X}, \boldsymbol{\Phi}^{(m)})$ using one iteration of Gibbs sampling from the most likely component assignments based on the most recent approximate posterior probabilities, $\gamma_{ik}^{*(m-\frac{1}{2})}$.

   (b) Compute approximate posterior probabilities, $\gamma_{ik}^{*(m)}, i = 1, \dots, n;\ k = 1, \dots, K$:

   - *Simulated field approximation*:

$$\gamma_{ik}^{*(m)} = \frac{\phi\left(y_i|\mathbf{x}_i'\boldsymbol{\beta}_k^{(m)}, \sigma_k^{2\ (m)}\right) P(\mathbf{Z}_i = \mathbf{e}_k|\check{\mathbf{z}}_{N_i}^{(m)}, \mathbf{X}, \psi^{(m)})}{\sum_{l=1}^{K}\phi\left(y_i|\mathbf{x}_i'\boldsymbol{\beta}_l^{(m)}, \sigma_l^{2\ (m)}\right) P(\mathbf{Z}_i = \mathbf{e}_l|\check{\mathbf{z}}_{N_i}^{(m)}, \mathbf{X}, \psi^{(m)})}, \tag{17}$$

   $i = 1, \dots, n;\ k = 1, \dots, K.$

2. **Generalized M-step**:

   (a) Fix $\boldsymbol{\theta}$ to the current value $\boldsymbol{\theta}^{(m)}$, and improve $Q^*_{\text{pen}}(\boldsymbol{\Phi}, \boldsymbol{\Phi}^{(m)})$ w.r.t. ψ via numerical optimization to obtain $\psi^{(m+1)}$.

   i. Update approximate distributions by computing $\check{\mathbf{z}}_i^{(m+\frac{1}{2})}$ as in step 1(a) with parameter vector $(\psi^{m+1}, \boldsymbol{\theta}^{(m)})$.

   ii. Update approximate posterior probabilities by computing $\gamma_{ik}^{*(m+\frac{1}{2})}$ as in step 1(b) using values $\check{\mathbf{z}}_i^{(m+\frac{1}{2})}$ in place of $\check{\mathbf{z}}_i^{(m)}$.

   iii. Compute approximate component probabilities, $\tilde{\pi}_k^{*(m+\frac{1}{2})} = \frac{1}{n}\sum_i^n \gamma_{ik}^{*(m+\frac{1}{2})}, k = 1, \dots, K.$

   (b) Update $\boldsymbol{\rho}$ and $\boldsymbol{\varphi}$ with:

$$\rho_k^{(m+1)} = \frac{\tilde{\mathbf{y}}'\tilde{\mathbf{X}}\boldsymbol{\varphi}_k^{(m)} + \sqrt{\left(\tilde{\mathbf{y}}'\tilde{\mathbf{X}}\boldsymbol{\varphi}_k^{(m)}\right)^2 + 4||\tilde{\mathbf{y}}||^2 n_k}}{2||\tilde{\mathbf{y}}||^2} \qquad k = 1, \dots, K,$$

   and

$$\varphi_{kj}^{(m+1)} = \begin{cases} 0 & \text{if } |S_j| \le \lambda(\grave{\pi}_k^{*(m+\frac{1}{2})}) \\ (\lambda(\grave{\pi}_k^{*(m+\frac{1}{2})}) - S_j)/||\tilde{\mathbf{X}}_j||^2 & \text{if } S_j > \lambda(\grave{\pi}_k^{*(m+\frac{1}{2})}) \\ -(\lambda(\grave{\pi}_k^{*(m+\frac{1}{2})}) + S_j)/||\tilde{\mathbf{X}}_j||^2 & \text{if } S_j < -\lambda(\grave{\pi}_k^{*(m+\frac{1}{2})}) \end{cases}$$

where $n_k = \sum_i^n \gamma_{ik}^{*(m+\frac{1}{2})}$, $\tilde{y}_i = \sqrt{\gamma_{ik}^{*(m+\frac{1}{2})}} y_i$, $\tilde{\mathbf{x}}_i = \sqrt{\gamma_{ik}^{*(m+\frac{1}{2})}} \mathbf{x}_i$, where $\mathbf{x}_i$ is the ith row of $\mathbf{X}$, and S_j is defined as

$$S_j = -\rho_k^{(m+1)} \tilde{\mathbf{X}}_j' \tilde{\mathbf{y}} + \sum_{s<j} \varphi_{ks}^{(m+1)} \tilde{\mathbf{X}}_j' \tilde{\mathbf{X}}_s + \sum_{s>j} \varphi_{ks}^{(m)} \tilde{\mathbf{X}}_j' \tilde{\mathbf{X}}_s, \qquad j = 1, \ldots, p,$$

where p is the number of predictors, and $\tilde{\mathbf{X}}_j$ denotes the jth column of $\tilde{\mathbf{X}}$.

3. Set $m = m + 1$, and return to Step 1.

These steps are iterated through until convergence or a maximum number of iterations is reached. We initialize the algorithm from Step 2(b) and evaluate the convergence criteria after Step 1(b).

References

Benaglia, T., Chauveau, D., Hunter, D., & Young, D. (2009) “mixtools: An R package for analyzing finite mixture models,” *Journal of Statistical Software*, 32: 1–29.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977) “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Series B (Methodological)*, 1–38.

McLachlan, G. J. & Peel, D. (2004) *Finite Mixture Models*. John Wiley & Sons.

Pan W. & Shen X. (2007) “Penalized model-based clustering with application to variable selection,” *The Journal of Machine Learning Research*, 8: 1145–1164.

Städler, N., Bühlmann, P., & Van De Geer, S. (2010), “ℓ1-penalization for mixture regression models,” *TEST*, 19: 209–256.