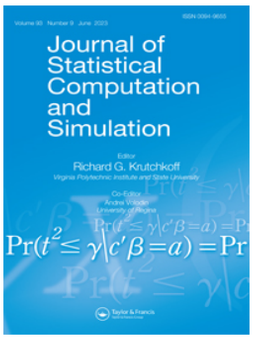

Taylor & Francis
Taylor & Francis Group

Journal of Statistical Computation and Simulation

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/gscs20

LASSO–penalized clusterwise linear regression modelling: a two–step approach

Roberto Di Mari, Roberto Rocci & Stefano Antonio Gattone

To cite this article: Roberto Di Mari, Roberto Rocci & Stefano Antonio Gattone (2023): LASSO–penalized clusterwise linear regression modelling: a two–step approach, Journal of Statistical Computation and Simulation, DOI: 10.1080/00949655.2023.2220058

To link to this article: https://doi.org/10.1080/00949655.2023.2220058

JOURNAL OF STATISTICAL COMPUTATION AND SIMULATION
https://doi.org/10.1080/00949655.2023.2220058

LASSO–penalized clusterwise linear regression modelling: a two–step approach

Roberto Di Mari [a], Roberto Rocci[b] and Stefano Antonio Gattone[c]

[a]Department of Economics and Business, University of Catania, Catania, Italy; [b]Department of Statistical Sciences, University of Rome La Sapienza, Rome, Italy; [c]DISFIPEQ, University G. d'Annunzio, Chieti-Pescara, Italy

ABSTRACT

In clusterwise regression analysis, the goal is to predict a response variable based on a set of explanatory variables, each with cluster-specific effects. In many real–life problems, the number of candidate predictors is typically large, with perhaps only a few of them meaningfully contributing to the prediction. A well–known method to perform variable selection is the LASSO, with calibration done by minimizing the Bayesian Information Criterion (BIC). However, existing LASSO-penalized estimators are problematic for several reasons. First, only certain types of penalties are considered. Second, the computations may sometimes involve approximate schemes. Third, variable selection is usually time consuming, due to a complex calibration of the penalty term, possibly requiring several multiple evaluations of an estimator for each plausible value of the tuning parameter(s). We introduce a two–step approach to fill these gaps. In step 1, we fit LASSO clusterwise linear regressions with some pre–specified level of penalization (*Fit* step). In step 2 (*Selection* step), we perform covariate selection locally, i.e. on the weighted data, with weights corresponding to the posterior probabilities from the previous step. This is done by using a generalization of the Least Angle Regression (LARS) algorithm, which permits covariate selection with a single evaluation of the estimator. In addition, both *Fit* and *Selection* steps leverage on an Expectation Maximization (EM) algorithm, fully in closed forms, designed with a very general version of the LASSO penalty. The advantages of our proposal, in terms of computation time reduction, and accuracy of model estimation and selection, are shown by means of a simulation study, and illustrated with a real data application.

ARTICLE HISTORY
Received 7 September 2022
Accepted 24 May 2023

KEYWORDS
Clusterwise linear regression; penalized likelihood; regularized ML; covariate selection

1. Introduction

Estimating the relationship between a response variable and various explanatory variables is a long-standing problem in statistics and applied sciences. In most applications,

CONTACT Roberto Di Mari roberto.dimari@unict.it Corso Italia 55, 95129 Catania (Italy)

Supplemental data for this article can be accessed here. https://doi.org/10.1080/00949655.2023.2220058

applied researchers must face two main challenges: (i) population heterogeneity, and (ii) a potentially large number of candidate covariates.

The assumption that a single population model adequately describes the relationship between the response variable, and the covariates is often too restrictive. Finite mixtures of linear regressions (FMLR; see, e.g. [1–6]) provide a solid modelling framework, which allows relaxing this assumption. They can be used to classify sample observations into different sub-populations (latent groups), and estimate the cluster-specific contribution of each covariate to the response. For model fitting, the EM algorithm [7] is frequently used to obtain maximum likelihood estimates of the parameters of interest.

Still, the number of candidate covariates can be overly large, with possibly only few of them having a predictive ability over the response. Being able to identify these predictors becomes crucial for FMLR and, more in general, for any regression analysis.

The LASSO, originally proposed by [8], has become a popular tool for large–dimension regression analysis with *sparsity* – i.e. in the presence of a vector of regression coefficients with some zero elements. Several algorithms have been proposed to solve the LASSO problem. Among the most notable examples, the LARS algorithm [9]; greedy approaches, as the coordinate descent method [10]; robust distance–based techniques [11].

Over the last twenty years, LASSO-type penalties have been introduced in the context of FMLR models for covariate selection and, more in general, sparse clusterwise regression analysis. Khalili and Chen [12] were the first to extend the penalized likelihood approach to FMLR models. For (penalized) maximum likelihood estimation, they developed an EM algorithm, in which the penalty function is locally approximated by a quadratic function. Another ℓ_1-penalization was proposed by [13], who integrated a coordinate descent algorithm in a block-wise EM algorithm to update the regression parameters. Mortier et al. [14] elaborated an adaptive LASSO penalty for generalized linear models. Finally, [15] developed a minorization-maximization algorithm to fit FMLR models with LASSO penalty. Variable selection for FMLR has been performed in the Bayesian framework as well. Among the many contributions, recent examples are [16,17].

The above (frequentist) works on penalized FMLR share, to a different extent, three main difficulties. First, the optimization problem entailed in model estimation. Namely, how the penalty is specified, and which parameters are involved therein. Second, how the iterative algorithm is designed, since the penalized log-likelihood cannot be maximized analytically. Third, how covariate/variable/feature selection is handled. That is, the tuning of the regularization parameter(s) controlling the amount of sparsity of the final solution. These hurdles complicate the implementation of penalized FMLR models, limiting their use in practice.

In the present work, we jointly address the above three issues. First, we consider a new LASSO penalty term, which is more general, and extends earlier proposals. Among its nice features, the penalty is specified such that the penalized likelihood is bounded, and each mixture component has a cluster-specific penalty, and penalizing constant. Second, we develop an efficient EM algorithm for computing the penalized ML estimates of the FMLR model parameters. Differently from the existing ones, it is entirely based on closed form updates, and uses no approximate schemes. Third, we generalize the use of the LARS algorithm [9], to efficiently obtain all cluster-specific LASSO solutions, given an estimate of the posterior probabilities. This provides what we define as a *LARS-adaptive grid* – the

main constituent of a novel covariate selection procedure, built from a single evaluation of the penalized estimator. The overall best model is then chosen based on BIC.

The structure of the paper is laid out as follows. We first describe the modelling framework (Section 2), along with some background literature. Sections 3 and 4 constitutes the core of the methodological contribution, and are devoted to the EM algorithm (Section 3), and the covariate selection procedure (Section 4). The effectiveness of the proposal is illustrated by means of a simulation study (Section 5), and an empirical application (Section 6). Section 7 concludes with some final remarks, and potential directions for future research.

2. The technical framework: penalized FMLR models and estimation issues

Let $\mathrm{y}_1, \ldots, \mathrm{y}_n$ be a sample of independent observations, each observed alongside with a vector of J explanatory variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Let us assume $\mathrm{Y}_i \mid \mathbf{x}_i$ is distributed as a finite mixture of linear regression models, that is

$$f(\mathrm{y}_i \mid \mathbf{x}_i; \boldsymbol{\psi}) = \sum_{g=1}^{G} \pi_g \phi(\mathrm{y}_i \mid \mathbf{x}_i, \sigma_g^2, \boldsymbol{\beta}_g) = \sum_{g=1}^{G} \pi_g \frac{1}{\sqrt{2\pi\sigma_g^2}} \exp\left[-\frac{(\mathrm{y}_i - \mathbf{x}_i'\boldsymbol{\beta}_g)^2}{2\sigma_g^2}\right], \quad (1)$$

where G is the number of clusters, and π_g, $\boldsymbol{\beta}_g$, and σ_g^2 are the mixing proportion, the vector of $J+1$ regression coefficients that includes an intercept, and the variance term for the gth cluster. The set of all model parameters is given by $\boldsymbol{\psi} = \{(\pi_2, \ldots, \pi_G; \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_G; \sigma_1^2, \ldots, \sigma_G^2) \in \mathbb{R}^{G-1+(J+1)G+G} : \pi_1 + \ldots + \pi_G = 1, \pi_g > 0, \sigma_g^2 > 0, \text{for } g = 1, \ldots, G\}$.

Under some mild regularity conditions, the parameters in $\boldsymbol{\psi}$ are identified [18]. Based on the model of Equation (1), we compute the posterior membership probabilities for each observation as follows

$$P(g \mid \mathrm{y}_i, \mathbf{x}_i) = \frac{\pi_g \phi(\mathrm{y}_i \mid \mathbf{x}_i; \sigma_g^2, \boldsymbol{\beta}_g)}{\sum_{h=1}^{G} \pi_h \phi(\mathrm{y}_i \mid \mathbf{x}_i; \sigma_h^2, \boldsymbol{\beta}_h)}, \quad (2)$$

which can be used to cluster observations according to, for instance, fuzzy or crisp classification rules.

The log-likelihood function can be specified as

$$\ell(\boldsymbol{\psi}) = \sum_{i=1}^{n} \log \left\{ \sum_{g=1}^{G} \pi_g \frac{1}{\sqrt{2\pi {\sigma_g}^2}} \exp\left[-\frac{(\mathrm{y}_i - \mathbf{x}_i'\boldsymbol{\beta}_g)^2}{2\sigma_g^2}\right] \right\}, \quad (3)$$

which we maximize to estimate $\boldsymbol{\psi}$ by means of the popular EM algorithm [7]. The latter uses the following complete-data log-likelihood

$$\mathrm{CD}\ell(\boldsymbol{\psi}) = \sum_{g=1}^{G} \sum_{i=1}^{n} z_{ig} \log \pi_g + \sum_{g=1}^{G} \sum_{i=1}^{n} z_{ig} \log \left\{ \phi\left(\mathrm{y}_i | \mathbf{x}_i, \boldsymbol{\beta}_g, \sigma_g^2\right) \right\}, \quad (4)$$

where $z_{ig} = 1$ if observation i comes from component g, and $z_{ig} = 0$ otherwise.

The standard design of the EM algorithm for ML estimation of FMLR models consists of the following four steps. In step 1 (E-step), the posterior probabilities are updated

according to the expression given by (2). The M-step starts with step 2, where each mixing proportion is updated as the mean of the corresponding posterior class membership probabilities. In step 3, the regression coefficients are updated by regressing the response on the explanatory variables, and the corresponding posteriors are used as weights. Step 4 completes the M-step, updating each component variance as the weighted mean of the squared errors, with weights corresponding to the group posteriors.

From now onward, let us suppose that $\boldsymbol{\beta}_g$ is sparse ($g = 1, \ldots, G$), i.e. some of its elements are exactly equal to zero. In fact, note that maximum likelihood estimates of the elements of $\boldsymbol{\beta}_g$ will be close (but never exactly equal) to zero. To estimate all model parameters, shrinking to zero the regression coefficients of unnecessary predictors, a penalty can be added to the log-likelihood. This yields the following penalized log–likelihood

$$p\ell(\boldsymbol{\psi}) = \ell(\boldsymbol{\psi}) - p(\boldsymbol{\psi}). \tag{5}$$

Different LASSO penalties have been used in the literature to address the sparsity of regression coefficients in FMLR models. A *Single weighted lambda penalty* was used in [13]

$$p(\boldsymbol{\psi}) = \lambda \sum_{g=1}^{G} \pi_g^{\gamma} \frac{\sum_{j=1}^{J} |\beta_{jg}|}{\sigma_g^{\delta}}, \tag{6}$$

with cluster weight equal to $\frac{\pi_g^{\gamma}}{\sigma_g^{\delta}}$, and $\gamma, \delta \in \{0, 1/2, 1\}$. A *component-wise lambda penalty* was adopted by [12,15]

$$p(\boldsymbol{\psi}) = \sum_{g=1}^{G} \lambda_g \pi_g \sum_{j=1}^{J} |\beta_{jg}|, \tag{7}$$

with cluster weight equal to π_g.

Arguably, the expressions in (6) and (7) are the two most representative instances of LASSO penalties for clusterwise linear regression models. Both require calibrating one or more tuning constants, a crucial task in LASSO regression analysis. Yet, the inclusion of the mixing proportions, and/or of the component scales in the penalty term further complicates model fitting. Namely, the update for the mixing proportion – and potentially for the cluster–specific variances – of the standard EM algorithm is no longer valid. In the existing literature, several alternative algorithms have been proposed, each with specific features related to the penalty specification, and tuning strategies.

2.1. Literature background

Khalili and Chen [12] developed an EM algorithm for which, in order to ensure differentiability, the update of the regression coefficients is derived by replacing the penalty (7) with a local quadratic approximation. Yet, the standard expression is used to update the mixing proportions, although the latter quantities are involved in the penalty. The amount of shrinkage is selected by minimizing a deviance-based generalized cross validation criterion in each component.

Städler et al. [13] incorporate the soft-thresholding operator, used to update the regression coefficients, into a generalized EM algorithm that proceeds as follows. During 10

iterations, only the non-zero coefficients – i.e. those related to the variables in the so–called *active set* – are updated. Every 11th iteration, the remaining coordinates are visited, to update the active set. For the mixing proportions, they improve the objective function by using a line search on a grid, in the direction of the standard update. The tuning of the lambda is based on cross-validation and BIC, with search within the interval $[0, \lambda_{\max}]$.

In [14], the penalized log-likelihood is maximized by combining routines available from the R packages *flexmix* [19] (for mixture model estimation), and *glmnet* [20] (for estimating penalized linear models). Still, the standard update for the mixing proportions is adopted. Their strategy to select the lambdas is based on cross-validation within the EM algorithm. In order to ensure convergence, they recommend using the same sub-samples across all EM iterations.

Lloyd-Jones et al. [15], to make (5) differentiable, use an ϵ-approximate version of (7), which requires the tuning of a positive constant ϵ. Subsequently, they maximize a minorizer of (5). Note that such a formulation, in itself, is not able to achieve exact sparsity of the regression coefficients. They circumvent this limitation by refitting the model twice. In the refitting model, the coefficients that are smaller in absolute value than some $M_\epsilon > 0$, which must be calibrated as well, are set to zero. To select the lambdas, they optimize the BIC over all parameters in $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_G]$ via a Nealder-Mead method – with upper and lower bounds for the lambdas that must be specified by the user.

All in all, the estimation, and computation issues that we presented above are still somewhat unresolved. In the present work, we address these issues by proposing a modified EM algorithm, for which we develop a more general and flexible penalty term. The latter can be easily shown to embed existing LASSO penalties for FMLR as special cases. In addition, to correctly update the mixture proportions, we derive a fully closed–form Newton–Raphson step, replacing the classic, though incorrect, expression. Subsequently, covariate selection is carried out in an innovative and computationally efficient way, exploiting the features of the LARS algorithm. These novelties are implemented in a two–step technique, which is described in the next Section.

3. A two–step LARS algorithm for penalized FMLR modelling

Our general LASSO penalty is specified as follows

$$p(\boldsymbol{\psi}) = \sum_{g=1}^{G} \lambda_g \pi_g^{\gamma} \frac{\sum_{j=1}^{J} |\beta_{jg}|}{2\sigma_g^{2\delta}}, \tag{8}$$

with $\gamma, \delta \in \{0, 1\}$. It is straightforward to note that [12]'s penalty (7) can be recovered by setting $\gamma = 1$, $\delta = 0$, and pre-multiplying it by 2. In addition, as a distinctive and innovative feature with respect to [12,15], (8) penalizes the ℓ_1-norm of the regression coefficients, along with small values of the variance of each component. Note that also [13]'s penalty (6) can be obtained as a special case, by constraining the tuning parameters to be the same across clusters, pre-multiplying (8) by 2 and setting $\delta \in \{0, 1/4, 1/2\}$.

The penalized log-likelihood (5) can be formulated, combining the log-likelihood (3), and the penalty (8), as follows

$$p\ell(\boldsymbol{\psi}) = \sum_{i=1}^{n} \log \left\{ \sum_{g=1}^{G} \pi_g \frac{1}{\sqrt{2\pi {\sigma_g}^2}} \exp\left[-\frac{(\mathrm{y}_i - \mathbf{x}_i'\boldsymbol{\beta}_g)^2}{2\sigma_g^2} \right] \right\} - \sum_{g=1}^{G} \lambda_g \pi_g^{\gamma} \frac{\sum_{j=1}^{J} |\beta_{jg}|}{2\sigma_g^{2\delta}}. \quad (9)$$

We remark that, by involving the component scales in the penalty, when $\delta = 1$, we keep σ_g^2 *tied* too, as overly small variance values would increase the relative load of the penalty on the objective function. Consequently, this should ensure that (9) is bounded below. In [13], the Authors have shown that the penalized log–likelihood is bounded when (6) is used with $\gamma = 0$ and conjectured that this property should hold even when $\gamma \neq 0$. Here we formulate the same conjecture but we cannot exclude pathological cases of unboundedness.

The ML estimate $\widehat{\boldsymbol{\psi}}$ is obtained by maximizing the penalized log-likelihood (9). We address such optimization problem employing an EM-type algorithm, for which the complete data penalized log-likelihood can be expressed as

$$\begin{aligned} \mathrm{CD}p\ell(\boldsymbol{\psi}) = & \sum_{g=1}^{G} \sum_{i=1}^{n} z_{ig} \log \pi_g + \sum_{g=1}^{G} \sum_{i=1}^{n} z_{ig} \log \left\{ \phi\left(\mathrm{y}_i | \mathbf{x}_i, \boldsymbol{\beta}_g, \sigma_g^2\right) \right\} \\ & - \sum_{g=1}^{G} \lambda_g \pi_g^{\gamma} \frac{\sum_{j=1}^{J} |\beta_{jg}|}{2\sigma_g^{2\delta}}. \end{aligned} \quad (10)$$

In the expectation step (E-step), the posterior class membership probabilities are updated as follows

$$\hat{z}_{ig} = \frac{\pi_g \phi(\mathrm{y}_i \mid \mathbf{x}_i; \sigma_g^2, \boldsymbol{\beta}_g)}{\sum_{h=1}^{G} \pi_h \phi(\mathrm{y}_i \mid \mathbf{x}_i; \sigma_h^2, \boldsymbol{\beta}_h)}, \quad (11)$$

for $i = 1, \ldots, n$ and $g = 1, \ldots, G$. The $\hat{z}_{ig}$'s are carried over to the maximization step (M-step), to compute the updates for the remaining quantities of interest. The E- and M-steps of the algorithm are alternated until some convergence criterion is met. The full M-step is described in details in the next subsections.

3.1. M-step: update of the mixing proportions

The classic update of $\pi_g = \frac{1}{n} \sum_{i=1}^{n} \hat{z}_{ig}$, if $\gamma > 0$, is incorrect. We propose replacing it by one Newton-Raphson step applied to (10).

First, let us re-parametrize the π_g's using the following multinomial logistic parametrization

$$\pi_g = \frac{\exp(\alpha_g)}{1 + \sum_{h=1}^{G-1} \exp(\alpha_h)} \quad \text{for} \quad g = 1, \ldots, G,$$

where the last category G is taken as reference setting $\alpha_G = 0$. Note that we can write $\pi_g = \exp(\alpha_g)\pi_G$, for $g = 1, \ldots G$.

The penalty in (10), under the above logistic parametrization, can be written as

$$p(\boldsymbol{\alpha}) = \pi_G^{\gamma} \sum_{g=1}^{G} \lambda_g \exp(\gamma \alpha_g) \sum_{j=1}^{J} \frac{|\beta_{jg}|}{2\sigma_g^{2\delta}}, \tag{12}$$

yielding the following complete data penalized log-likelihood function of $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_{G-1})$

$$\mathrm{CD}p\ell(\boldsymbol{\alpha}) = \sum_{g=1}^{G-1} z_{+g}\alpha_g + n\log(\pi_G) - p(\boldsymbol{\alpha}), \tag{13}$$

where $z_{+g} = \sum_{i=1}^{n} z_{ig}$.

A Newton-Raphson step, used to increase (13), can be set up according to the usual formulation

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} + c(t)^{(t)} I(\boldsymbol{\alpha})^{-1} S(\boldsymbol{\alpha}), \tag{14}$$

where $S(\boldsymbol{\alpha})$ and $I(\boldsymbol{\alpha})$ are the gradient, and the Hessian of (13), respectively. The lth component of the gradient of the penalty function $p(\boldsymbol{\alpha})$ is given by

$$\frac{\partial p(\boldsymbol{\alpha})}{\partial \alpha_l} = -\gamma \pi_G \exp(\alpha_l) p(\boldsymbol{\alpha}) + \gamma \pi_G^{\gamma} \lambda_l \exp(\gamma \alpha_l) \sum_{j=1}^{J} \frac{|\beta_{jl}|}{\sigma_l^{2\delta}}. \tag{15}$$

The lth component of the gradient $S(\boldsymbol{\alpha})$ is given by

$$\frac{\partial \mathrm{CD}p\ell(\boldsymbol{\alpha})}{\partial \alpha_l} = z_{+l} - n\pi_G \exp(\alpha_l) + \gamma \pi_G \exp(\alpha_l) p(\boldsymbol{\alpha}) - \gamma \pi_G^{\gamma} \lambda_l \exp(\gamma \alpha_l) \sum_{j=1}^{J} \frac{|\beta_{jl}|}{2\sigma_l^{2\delta}}. \tag{16}$$

The (l, l)th and (l, m)th components of the Hessian matrix $I(\boldsymbol{\alpha})$ are respectively

$$\begin{aligned}
\frac{\partial \mathrm{CD}p\ell(\boldsymbol{\alpha})}{\partial \alpha_l^2} &= -n\pi_G \exp(\alpha_l)\left(\pi_G \exp(\alpha_l) - 1\right) - \gamma \pi_G^2 \exp(2\alpha_l) p(\boldsymbol{\alpha}) \\
&\quad + \gamma \pi_G \exp(\alpha_l) \left[p(\boldsymbol{\alpha}) - \gamma \pi_G \exp(\alpha_l) p(\boldsymbol{\alpha}) + \gamma \pi_G^{\gamma} \lambda_l \exp(\gamma \alpha_l) \sum_{j=1}^{J} \frac{|\beta_{jl}|}{2\sigma_l^{2\delta}} \right] \\
&\quad + \gamma^2 \pi_G^{\gamma+1} \exp(\alpha_l) \lambda_l \exp(\gamma \alpha_l) \sum_{j=1}^{J} \frac{|\beta_{jl}|}{\sigma_l^{\delta}} - \gamma^2 \pi_G^{\gamma} \lambda_l \exp(\gamma \alpha_l) \sum_{j=1}^{J} \frac{|\beta_{jl}|}{2\sigma_l^{2\delta}},
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial \mathrm{CD}p\ell(\boldsymbol{\alpha})}{\partial \alpha_l \partial \alpha_m} &= -n \exp(\alpha_l)\left(-\pi_G^2 \exp(\alpha_m)\right) \\
&\quad + \gamma \exp(\alpha_l) \left(-\pi_G^2 \exp(\alpha_m) p(\boldsymbol{\alpha}) + \pi_G \frac{\partial p}{\partial \alpha_m} \right) + \\
&\quad - \gamma^2 \pi_G^{\gamma-1} \left(-\pi_G^2 \exp(\alpha_m)\right) \lambda_l \exp(\gamma \alpha_l) \sum_{j=1}^{J} \frac{|\beta_{jl}|}{2\sigma_l^{2\delta}}.
\end{aligned}$$

The value of $c(t)$ is a constant, chosen with a line search over the values 0, 0.5, 1 and 1.5, such that $\mathrm{CD}p\ell(\boldsymbol{\alpha})^{(t+1)} \geq \mathrm{CD}p\ell(\boldsymbol{\alpha})^{(t)}$.

3.2. M-step update of regression coefficients and component variances

The M-step for the update of the regression coefficients decouples into G separate weighted LASSO optimization problems of the form

$$-\sum_{i=1}^{n} \mathrm{z}_{ig} \frac{(\mathrm{y}_i - \mathbf{x}_i'\boldsymbol{\beta}_g)^2}{2\sigma_g^2} - \lambda_g \pi_g^{\gamma} \sum_{j=1}^{J} \frac{|\beta_{jg}|}{2\sigma_g^{2\delta}}$$

equivalent to the minimization of

$$= \sum_{i=1}^{n} \frac{\mathrm{z}_{ig}}{\sigma_g^{2(1-\delta)}\pi_g^{\gamma}} (\mathrm{y}_i - \mathbf{x}_i'\boldsymbol{\beta}_g)^2 + \lambda_g \sum_{j=1}^{J} |\beta_{jg}| . \tag{17}$$

By acting a preliminary local centring and weighting of the data, (17) can be cast into a simple LASSO regression problem.

Let $\bar{\mathrm{y}}_g = \frac{\sum_{i=1}^{n} \mathrm{z}_{ig}\mathrm{y}_i}{\sum_{i=1}^{n} \mathrm{z}_{ig}}$, and $\bar{\mathrm{x}}_{jg} = \frac{\sum_{i=1}^{n} \mathrm{z}_{ig}\mathrm{x}_{ij}}{\sum_{i=1}^{n} \mathrm{z}_{ig}}$ be respectively the response, and the jth predictor weighted means, for $j = 1, \ldots, J$. We collect all predictors group means in the vector $\bar{\mathbf{x}}_g = (\bar{\mathrm{x}}_{1g}, \ldots, \bar{\mathrm{x}}_{Jg})'$.

We let $\tilde{\mathbf{y}} = \mathbf{y} - \mathbf{1}_n \bar{\mathrm{y}}_g$, and $\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{1}_n \bar{\mathbf{x}}_g'$ be respectively the locally centred response, and the locally centred explanatory variables, where $\mathbf{1}_n$ is an n-vector of ones.

We will now consider the locally weighted data $\{\mathrm{y}_i^{(g)}, \mathbf{x}_i^{(g)}\}_{i=1}^{n}$, where the ith record is equal to $\mathrm{y}_i^{(g)} = \frac{\sqrt{\mathrm{z}_{ig}}}{\sigma_g^{(1-\delta)}\sqrt{\pi_g^{\gamma}}} \tilde{\mathrm{y}}_i$, and $\mathbf{x}_i^{(g)} = \frac{\sqrt{\mathrm{z}_{ig}}}{\sigma_g^{(1-\delta)}\sqrt{\pi_g^{\gamma}}} \tilde{\mathbf{x}}_i$, with σ_g being the current update for the component scale. Note that if $\delta = \gamma = 1$, the weighting of data simplifies to $\sqrt{\mathrm{z}_{ig}/\pi_g}$.

With the locally weighted data, available algorithms such as the coordinate descent method [10], or the LARS [9] can be used to compute the current estimate of $\boldsymbol{\beta}_g$.

After updating the regression coefficients $\boldsymbol{\beta}_g$, with simple algebra it is possible to show that the M-step update for each component variance is

$$\sigma_g^2 = \frac{\sum_{i=1}^{n} z_{ig}(\mathrm{y}_i - \mathbf{x}_i'\boldsymbol{\beta}_g)^2}{\sum_{i=1}^{n} z_{ig}} + \delta\lambda_g \pi_g^{\gamma} \frac{\sum_{j=1}^{J} |\beta_{jg}|}{\sum_{i=1}^{n} z_{ig}}, \tag{18}$$

which reduces to the standard M-step update for a heteroscedastic FMLR if $\delta = 0$.

The above steps constitute a *generalized* EM algorithm [7] where, in the M step, the conditional expectation of the complete log-likelihood is increased rather than maximized. More specifically, while the objective function is maximized with respect to a subset of the parameters in each sub-step, it is increased with reference to the update of the mixing proportions. Note that, in case $\gamma = 0$, the latter quantities do not affect the penalty term, and thus even their update constitutes a maximization.

Therefore, our generalized EM belongs to the family of *ECM algorithms* [21]: these have been shown, under mild regularity conditions [22], to converge to a stationary point that is usually a local maximum. This is obtained, in practice, by iterating the four sub-steps until

the increment in the log-likelihood between two subsequent iterations is less than some pre-specified threshold.

4. Covariate selection

The proposed EM algorithm works with a pre–specified value of the penalty parameter. Consequently the LASSO, by shrinking some of the regression coefficients to zero, identifies useful and useless predictors – i.e. it performs covariate selection. However, the existing selection strategies, which have been reviewed above, share one main pitfall: the algorithms used for fitting the LASSO penalized FMLR model must be run for all potential candidate penalty parameters, resulting in a computationally intensive and time consuming task. In fact, covariate selection for FMLR models can be performed in a single run. Doing so is possible by combining the characteristics of the LARS algorithm with BIC.

In linear regression analysis, one of the most attractive properties of the LARS algorithm is that of solving the LASSO problem over the whole domain of the penalty parameter. This allows obtaining the *entire regularization path* – from the empty set, until the full set of predictors is reached. The appropriate level of sparsity can then be chosen, e.g. by BIC minimization.

Our approach consists in fitting first a FMLR with some pre-specified $\lambda > 0$. Subsequently, covariate selection is performed *componentwise* minimizing BIC on the *local* LARS path. Further improvements can be achieved by re-estimating the model parameters, and consequently updating the local weights, before evaluating the BIC. One of the objectives of the simulation study will be to assess if the increase in computational cost entailed by this additional step is worthwhile – in terms of enhanced accuracy of covariate selection.

4.1. The two–step L-LARS estimator

Our two–step L-LARS estimator can be described as follows.

Step 0: *Initialization*. Set $\boldsymbol{\lambda}^{(ini)} = \{\lambda_1, \lambda_2, \ldots, \lambda_g\}$ to some pre-specified positive values.
Step 1: *Fit*. For fixed lambdas, estimate the penalized FMLR model using the modified EM algorithm of Section 3, obtaining $\hat{\boldsymbol{\psi}}_{\boldsymbol{\lambda}^{(ini)}}$.
Step 2: *Covariate selection*. Use the posterior class probabilities of the previous step to weigh the data (Subsection 3.2). Compute the full LASSO path for each component by means of the LARS algorithm. The procedure returns the $K+1$ knots of the LASSO solution $\{\|\boldsymbol{\beta}_g^{(k)}\|_1\}_{k=0}^{K} = \{\|\boldsymbol{\beta}_g^{(0)}\|_1 < \|\boldsymbol{\beta}_g^{(1)}\|_1 < \ldots < \|\boldsymbol{\beta}_g^{(K)}\|_1\}$ for $g = 1, \ldots G$, with $\|\boldsymbol{\beta}_g^{(k)}\|_k = \sum_{j=1}^{J} |\beta_{jg}^{(k)}|$, and $K = \min(J, n_g - 1)$. For each component $g = 1, \ldots, G$, select the best local model by minimizing the BIC on the LASSO solutions computed on the sequence $\{\|\boldsymbol{\beta}_g^{(k)}\|_1\}_{k=0}^{K}$, with data locally weighted using the estimated posteriors and component scales.
Step 3: *Refit (optional)*. Fit an unpenalized finite mixture, eliminating from the components all regressors corresponding to zero coefficients, as selected in Step 2.

A few important remarks follow. First, the expression herein used to compute the BIC counts the number of nonzero coefficients as an estimate for the degrees of freedom of each

Table 1. Crossed simulation conditions for sample size, regression parameter configuration and mixing proportions, for the uncorrelated predictor scenario.

Simulation condition	Mixing proportions	Model	Sample size
1	$(0.5, 0.5)'$	1	100
2	$(0.5, 0.5)'$	1	200
3	$(0.5, 0.5)'$	2	100
4	$(0.5, 0.5)'$	2	200
5	$(0.3, 0.7)'$	1	100
6	$(0.3, 0.7)'$	1	200
7	$(0.3, 0.7)'$	2	100
8	$(0.3, 0.7)'$	2	200
9	$(0.1, 0.9)'$	1	100
10	$(0.1, 0.9)'$	1	200
11	$(0.1, 0.9)'$	2	100
12	$(0.1, 0.9)'$	2	200

Notes: The same scheme applies also to the correlated predictor scenario (conditions 13–24, with $\rho_{h,j} = 0.5^{|h-j|}$ for $h, j = 1, \ldots, J$).

Table 2. Regression parameter configurations.

Parameters	Model = 1	Model = 2
$\boldsymbol{\beta}_1$	(1,0,0,3,0)	(1,0.6,0,3,0)
$\boldsymbol{\beta}_2$	(-1,2,0,0,3)	(-1,0,0,4,0.7)

cluster–specific lasso regression [23]. However, we report that alternative criteria are also available. See, e.g. [24].

Second, the above covariate selection step may be seen as a *shortcut* version of a *joint* search over all $(K + 1)^G$ crossed local LARS specifications in terms of non–zero coefficients – *LARS grid*. Evidently, the evaluation of such a number of potential models is unfeasible already for moderate K and/or G.

Third, with the local search, standard tasks like selection of the number of mixture components, are relatively easy. Indeed, one can look for the best FMLR under LASSO penalty for each choice of $G = 1, \ldots, G_{\max}$. Then, the optimal G is the one minimizing BIC.

5. Simulation study

To evaluate the performance of the proposed methodology, we consider the simulation setup of [12]. The data are generated from a 2-component clusterwise linear regression with 5 regressors and component-specific intercepts. The simulation factors, described in Tables 1 and 2, are sample size (100 and 200), mixing proportions (0.5 and 0.5; 0.7 and 0.3; 0.9 and 0.1), cluster separation (*high* and *low*), and degree of collinearity between regressors (*uncorrelated* and *correlated*).

Overall, there are 24 crossed simulation conditions. 250 data sets are generated for each simulation condition. For each of them, the target measures are: (i) the Adjusted Rand Index (ARI, [25]) as measure of cluster recovery; (ii) the rate of correctly classified regression coefficients between zeros and nonzeros (CC); (iii) the Mean Squared Error (MSE) of the regression coefficient estimates averaged over regressors and groups; (iv) the average CPU time in seconds of a simulation run (Time).

For the sake of space, in this section only results from the methods with $\gamma = 1$ and $\delta = 1$ are reported. Results with $\gamma = 0$ and $\delta = 1$, and with $\gamma = 1$ and $\delta = 0$ are provided in the Appendix.

The proposed L-LARS approach has been implemented as follows.

- *g1d1L-grid-r*: with an exhaustive search, in which the FMLR model together with the specific LASSO solutions are estimated on the group-specific LARS grid; the model minimizing the BIC is selected.
- *g1d1-2step-r*: our two-step procedure where, in the first step, the FMLR model is estimated with a fixed value of $\lambda = 5$. In the second step, covariate selection is performed over the group-specific LARS-adaptive grid; the model minimizing the BIC is selected.
- *g1d1E-grid-r*: exaustive equally spaced fixed grid search, in which the FMLR model, together with the specific LASSO solutions, are estimated on a fixed grid of $J + 1$ equally spaced values of λ. The latter is allowed to vary between 0 and $\lambda_{\max}$, where $\lambda_{\max}$ represents the maximum value of λ in the LASSO-solution path provided by the LARS estimated in the one-component model. This type of grid was used by [13], and more generally is the literature's most common strategy.

For each specification of our method, we include the optional refit step.

Arguably, comparing *g1d1L-grid-r* with *g1d1E-grid-r* allows the evaluation of the proposed LARS-adaptive grid with respect to the fixed grid of equally spaced values of λ, the latter being routinely used in the literature.

In addition, the inclusion of the exhaustive search in the comparison ensures a fairer assessment of our local covariate selection step.

To complete the evaluation, the proposed procedures have been compared with the following competing methods: the non-penalized estimator of the clusterwise linear regression model as computed by the R package `flexmix` [26] (*FlexMix*); the penalized LASSO extension of `flexmix` used by [14] (*FlexLASSO*); the penalized estimator, where BIC is minimized over all parameters in $\boldsymbol{\lambda}$ via the Nealder-Mead method (similar optimization strategy of [15] – *NealMead*). To achieve the greatest possible computational efficiency, the routines implementing our approach are built by integrating C++ code with R [27,28]. All code and scripts, which we used for both simulated and real data analyses, are available as online supplementary materials.

For each sample, the EM algorithm was randomly initialized with 5 starting points, and the solution with the smallest BIC was chosen.

In Table 3, we report the overall quartiles of all the target measures, over the aggregated simulation conditions. Not surprisingly, the unpenalized estimates provided by the *flex* method deliver the worst performance in terms of all assessing measures. The proposed L-LARS algorithm with the exhaustive search (*g1d1L-grid-r*) is the most accurate in terms of cluster recovery, variable selection, and accuracy of the estimation. However, the computationally more efficient two-step version (*g1d1-2step-r*) delivers similar performances, though with much reduced computing time.

Interestingly, the two-step L-LARS approach is, in median, over 50 times faster than its exaustive grid search counterpart – and 250 times faster than the *flexLASSO*. A remarkable computing time gain is achieved even with respect to *flexmix*.

Table 3. Quartiles of Adjusted Rand Index (ARI), rate of Correct Classification (CC), Mean Squared Error (MSE), and CPU computing time in seconds (Time), over simulated samples and simulation conditions.

	FlexMix	FlexLasso	NelMead	g1d1-Egrid-r	g1d1-2step-r	g1d1-Lgrid-r
			ARI			
Q1	0.638	0.651	0.636	0.703	0.704	0.710
Q2	0.768	0.773	0.760	0.815	0.810	0.819
Q3	0.861	0.873	0.857	0.894	0.896	0.900
			CC			
Q1	5	6	5	7	8	8
Q2	5	7	6	8	8	9
Q3	6	8	6	9	9	9
			MSE			
Q1	0.204	0.162	0.211	0.132	0.147	0.120
Q2	0.429	0.352	0.445	0.387	0.328	0.307
Q3	1.176	0.894	1.276	1.336	0.947	1.140
			Time			
Q1	0.0170	1.6985	0.0800	0.4060	0.0040	0.3889
Q2	0.0219	2.2766	0.1022	0.5803	0.0090	0.5307
Q3	0.0319	3.4987	0.1355	0.8838	0.0156	0.7635

Note: Results for $\gamma = \delta = 1$.

Table 4. Median Adjusted Rand Index (ARI), rate of Correct Classification (CC), Mean Squared Error (MSE), and CPU computing time in seconds (Time), for the uncorrelated regression conditions, Model = 1, and $\gamma = \delta = 1$.

Sample size	$n = 100$			$n = 200$		
Mixing proportions	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)
Method			ARI			
Flexmix	0.737	0.869	0.77	0.809	0.915	0.913
FlexLasso	0.772	0.871	0.796	0.827	0.916	0.921
NelMead	0.737	0.864	0.757	0.809	0.913	0.906
g1d1-2step-r	0.772	0.876	0.867	0.827	0.918	0.933
g1d1-Lgrid-r	0.79	0.879	0.894	0.846	0.919	0.944
g1d1-Egrid-r	0.808	0.88	0.894	0.846	0.931	0.929
			CC			
Flexmix	5	5	5	5	5	5
FlexLasso	8	7	7	8	8	7
NelMead	5	5	5	5	5	5
g1d1-2step-r	9	9	7	9	9	8
g1d1-Lgrid-r	9	9	8	10	9	9
g1d1-Egrid-r	10	10	9	10	10	9
			MSE			
Flexmix	0.28	0.368	2.76	0.126	0.16	0.556
FlexLasso	0.246	0.3	1.74	0.111	0.138	0.436
NelMead	0.276	0.365	3.34	0.126	0.159	0.596
g1d1-2step-r	0.204	0.266	2.03	0.102	0.11	0.467
g1d1-Lgrid-r	0.17	0.185	1.24	0.0733	0.0797	0.35
g1d1-Egrid-r	0.149	0.169	1.29	0.064	0.0678	0.795

To provide further insights on the different simulation factors, Tables 4–7 display the results for uncorrelated and correlated regressors, for varying degrees of cluster separation. We observe that the proposed L-LARS approaches perform uniformly better than the competing methods.

In terms of cluster recovery, more pronounced gains are observed under small samples, high cluster separation, and very unbalanced cluster sizes. With respect to variable selection abilities and estimation accuracy, we note that the proposed LARS-adaptive grid and

Table 5. Median Adjusted Rand Index (ARI), rate of Correct Classification (CC), Mean Squared Error (MSE), and CPU computing time in seconds (Time), for the uncorrelated regression conditions, Model = 2, and $\gamma = \delta = 1$.

Sample size	$n = 100$			$n = 200$		
Mixing proportions	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)
Method			ARI			
Flexmix	0.573	0.697	0.51	0.687	0.839	0.779
FlexLasso	0.573	0.722	0.552	0.687	0.84	0.789
NelMead	0.573	0.69	0.491	0.687	0.835	0.778
g1d1-2step-r	0.605	0.768	0.757	0.687	0.856	0.864
g1d1-Lgrid-r	0.604	0.754	0.772	0.704	0.854	0.851
g1d1-Egrid-r	0.573	0.742	0.764	0.687	0.839	0.859
			CC			
Flexmix	6	6	6	6	6	6
FlexLasso	8	7	7	8	8	7
NelMead	6	6	6	6	6	6
g1d1-2step-r	9	8	8	9	9	8
g1d1-Lgrid-r	8	8	8	9	9	8
g1d1-Egrid-r	7	8	7	8	8	8
			MSE			
Flexmix	0.357	0.581	3.42	0.151	0.186	1.16
FlexLasso	0.353	0.517	2.21	0.143	0.175	0.862
NelMead	0.353	0.584	3.67	0.15	0.185	1.17
g1d1-2step-r	0.304	0.496	2.71	0.132	0.16	0.896
g1d1-Lgrid-r	0.319	0.516	2.23	0.115	0.134	1.43
g1d1-Egrid-r	0.4	0.641	3.01	0.139	0.216	1.41

Table 6. Median Adjusted Rand Index (ARI), rate of Correct Classification (CC), Mean Squared Error (MSE), and CPU computing time in seconds (Time), for the correlated regression conditions, Model = 1, and $\gamma = \delta = 1$.

Sample size	$n = 100$			$n = 200$		
Mixing proportions	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)
Method			ARI			
Flexmix	0.668	0.81	0.642	0.756	0.86	0.851
FlexLasso	0.669	0.81	0.694	0.755	0.875	0.873
NelMead	0.669	0.803	0.605	0.756	0.86	0.851
g1d1-2step-r	0.703	0.835	0.827	0.773	0.878	0.885
g1d1-Lgrid-r	0.703	0.833	0.843	0.773	0.88	0.901
g1d1-Egrid-r	0.669	0.827	0.843	0.773	0.878	0.885
			CC			
Flexmix	5	5	5	5	5	5
FlexLasso	7	7	6	8	8	7
NelMead	5	5	5	5	5	5
g1d1-2step-r	8	8	7	9	9	8
g1d1-Lgrid-r	8	8	8	9	9	8
g1d1-Egrid-r	8	8	7	8	8	8
			MSE			
Flexmix	0.47	0.672	4.52	0.189	0.26	1.27
FlexLasso	0.394	0.485	2.35	0.17	0.183	0.849
NelMead	0.471	0.662	5.03	0.183	0.261	1.47
g1d1-2step-r	0.301	0.419	2.78	0.13	0.148	1.02
g1d1-Lgrid-r	0.294	0.353	2.24	0.116	0.12	1.06
g1d1-Egrid-r	0.308	0.392	2.5	0.122	0.133	1.12

the equally spaced grid perform similarly only with uncorrelated regressors, and low cluster separation (Table 4). In all remaining scenarios (Tables 5–7), the LARS-adaptive grid

Table 7. Median Adjusted Rand Index (ARI), rate of Correct Classification (CC), Mean Squared Error (MSE), and CPU computing time in seconds (Time), for the correlated regression conditions, Model = 2, and $\gamma = \delta = 1$.

Sample size	$n = 100$			$n = 200$		
Mixing proportions	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)
Method			ARI			
Flexmix	0.604	0.712	0.559	0.704	0.837	0.804
FlexLasso	0.604	0.727	0.62	0.704	0.839	0.826
NelMead	0.604	0.694	0.545	0.704	0.837	0.791
g1d1-2step-r	0.636	0.79	0.804	0.721	0.853	0.894
g1d1-Lgrid-r	0.636	0.781	0.804	0.721	0.857	0.866
g1d1-Egrid-r	0.604	0.761	0.809	0.704	0.855	0.869
			CC			
Flexmix	6	6	6	6	6	6
FlexLasso	8	8	7	8	8	7
NelMead	6	6	6	6	6	6
g1d1-2step-r	9	9	8	9	9	8
g1d1-Lgrid-r	9	8	8	9	9	8
g1d1-Egrid-r	8	7	7	8	8	8
			MSE			
Flexmix	0.493	0.855	4.44	0.221	0.327	1.33
FlexLasso	0.404	0.703	2.07	0.199	0.296	0.964
NelMead	0.493	0.845	4.15	0.217	0.326	1.55
g1d1-2step-r	0.332	0.578	2.8	0.17	0.257	1.1
g1d1-Lgrid-r	0.323	0.623	2.45	0.15	0.259	1.4
g1d1-Egrid-r	0.44	0.786	3.6	0.183	0.268	1.44

substantially improves over the equally spaced grid. The gain is particularly relevant with correlated regressors and high cluster separation.

Our flagship two-step L-LARS (*g1d1-2step-r*) delivers the best results when the cluster separation is high, and it is very close to the L-LARS when cluster separation is low. All in all, this approach offers a computationally efficient and reliable alternative to the time-consuming exaustive grid search.

6. A penalized clusterwise regression for baseball batter salaries

In order to investigate if and how baseball batter performances affect individual salaries, we analysed a real data set available from the website of the Journal of Statistics Education.[1] The same analysis was carried out by [12,15], allowing us to have a solid baseline to compare to. These data contained salaries for 337 Major League Baseball (MLB) batters from the year 1992, participating in at least one game in both the 1991 and 1992 seasons. Together with salaries, 16 measures of batting performance from 1991 were also observed.

Together with the above 16 performance measures, we considered 16 interaction terms as [12,15], for a total of 32 predictors.[2] Each covariate was standardized to have mean 0 and variance 1.

In line with the simulation study, the analysis will be done based on the two-step L-LARS estimator (with refit), with $\gamma = \delta = 1$ only. We sequentially fitted a penalized heteroscedastic clusterwise regression model with 2 to 5 components. For each G, the two-step L-LARS estimator finds the best covariate configuration, and the best overall model – in terms of both zero/nonzero covariate effects and number of mixture components – is selected as the one with the smallest BIC.

Table 8. Regression output from KC2007 [12], LMN2018 [15], and L-LARS (our stepwise approach).

	KC2007		LNM2018		two-step L-LARS	
	G_1	G_2	G_1	G_2	G_1	G_2
Intercept	6.41	7.00	6.46	6.59	6.47	6.55
π	0.72	0.28	0.24	0.76	0.41	0.58
AVG	–	−0.32	–	−0.05	–	–
OBP	–	0.29	–	–	0.02	–
R	–	−0.70	–	0.02	0.18	–
H	0.20	0.96	0.19	0.18	0.07	0.27
2B	–	–	0.28	–	–	–
3B	–	–	–	–	−0.08	–
HR	−0.19	–	–	–	–	–
RBI	0.26	–	0.25	0.20	0.10	−0.02
BB	–	–	–	0.09	0.01	–
SO	–	–	–	−0.12	–	–
SB	–	–	–	0.06	0.07	–
ERS	–	–	–	–	–	–
FAE	0.79	0.70	–	0.94	1.01	0.01
FA	0.72	–	–	–	–	–
AE	0.15	0.50	–	0.69	0.30	0.27
ARB	–	−0.36	–	–	–	–
AVG × FAE	−0.21	–	–	–	0.43	0.24
AVG × FA	0.63	–	–	–	0.39	–
AVG × AE	0.34	–	–	−0.11	0.22	–
AVG × ARB	–	–	–	–	–	–
R × FAE	–	–	–	–	−0.26	0.13
R × FA	0.14	−0.38	–	–	−1.58	0.03
R × AE	–	–	–	0.04	0.06	–
R × ARB	−0.18	0.74	–	–	−0.11	–
HR × FAE	–	–	–	0.10	0.02	–
HR × FA	–	–	–	0.02	0.76	0.04
HR × AE	–	0.34	–	–	−0.17	–
HR × ARB	–	–	–	0.04	0.09	–
RBI × FAE	0.29	−0.46	–	–	–	0.31
RBI × FA	−0.14	–	–	–	–	–
RBI × AE	–	–	–	0.05	0.34	0.21
RBI × ARB	–	–	–	–	–	–

Notes: $G = 2$. Covariate acronyms are AVG – Batting Average, OBP – On Base Percentage, R – Runs, H – Hits, 2B – Doubles, 3B – Triples, HR – Home Runs, RBI – Runs Batted In, BB – Walks, SO – Strikeouts, SB – Stolen Bases, ERS – Errors, FAE – Free Agency Eligibility, FA – Free Agent in 1991/92, AE – Arbitration Eligibility, ARB – Arbitration in 1991/92.

For each G, the algorithm was initialized randomly from 20 different starts, and the best solution according to BIC was retained. The final BIC output for $G = 2, 3, 4, 5$ was respectively 579.41, 548.66, 558.97, and 592.11 (the full 20×4 BIC values are available in the appendix). However, we retain both $G = 2$ and $G = 3$ solutions as done in the cited literature. BIC values are not reported in [12], but in [15] the $G = 3$ model had BIC equal to 573.6 (their lowest), and the $G = 2$ was their second ranked (BIC = 580.1). Note that both works imposed equal variances to avoid degenerate solutions. In fact, we can allow for variance heteroscedasticity under our penalized specification with $\delta \neq 0$, which rules out (non-pathological) unboundedness.

The regression output, for $G = 2$, is presented in Table 8. Our approach finds two clusters of similar sizes. In cluster G_2, the one with the largest number of batters and the highest average (log) salary, we observe relatively fewer non-zero coefficients than in G_1: among

Table 9. Regression output from LMN2018 [15], and L-LARS (our stepwise approach).

	LNM2018			two-step L-LARS		
	G_1	G_2	G_3	G_1	G_2	G_3
Intercept	5.17	6.58	6.60	6.76	6.51	6.69
π	0.02	0.32	0.66	0.34	0.55	0.12
AVG	–	–	–	−0.07	–	0.12
OBP	–	–	–	−0.03	–	0.16
R	–	–	–	–	–	0.17
H	–	0.28	0.16	0.32	0.46	−0.44
2B	–	–	–	0.08	−0.24	0.68
3B	–	–	–	−0.16	−0.08	0.38
HR	–	–	–	–	–	0.36
RBI	–	–	0.11	−0.30	0.16	0.03
BB	–	0.02	0.06	–	−0.07	–
SO	–	–	–	0.20	–	−1.04
SB	–	–	0.05	0.26	0.06	0.04
ERS	–	–	–	−0.24	–	0.06
FAE	–	0.01	1.03	0.13	1.06	0.02
FA	–	–	–	–	−0.71	0.70
AE	0.76	–	0.72	0.38	−0.22	0.53
ARB	–	–	–	−0.17	–	−0.56
AVG × FAE	–	–	–	–	0.42	−0.20
AVG × FA	–	–	–	–	–	–
AVG × AE	–	–	−0.13	–	0.62	–
AVG × ARB	–	–	–	–	–	–
R × FAE	–	0.18	–	–	−0.13	−0.35
R × FA	–	–	–	−0.27	0.51	−0.83
R × AE	–	–	0.07	–	–	−0.18
R × ARB	–	–	–	–	–	0.20
HR × FAE	–	–	0.07	–	0.24	−0.09
HR × FA	–	–	0.04	–	−0.10	0.22
HR × AE	–	–	–	0.11	–	–
HR × ARB	–	–	–	0.13	–	−0.06
RBI × FAE	–	0.31	–	0.57	−0.38	1.21
RBI × FA	–	–	–	0.31	0.16	–
RBI × AE	–	0.37	0.08	–	0.32	–
RBI × ARB	–	–	–	0.05	–	0.25

Notes: $G = 3$. Covariate acronyms are AVG – Batting Average, OBP – On Base Percentage, R – Runs, H – Hits, 2B – Doubles, 3B – Triples, HR – Home Runs, RBI – Runs Batted In, BB – Walks, SO – Strikeouts, SB – Stolen Bases, ERS – Errors, FAE – Free Agency Eligibility, FA – Free Agent in 1991/92, AE – Arbitration Eligibility, ARB – Arbitration in 1991/92.

others, H (number of Hits) has a positive effect on salary, whereas the Runs Batted In and Free Agency Eligibility interaction is the strongest in magnitude. Overall this is in line with [12], where there is one group with relatively larger size with fewer nonzero coefficients.

In Table 9, regression estimates for $G = 3$ are displayed, for [15] and our stepwise approach. From a clustering perspective, we observe that class sizes of our stepwise estimator are somewhat different and less *extreme* compared to those of [15]. Similarly to [15], the two largest groups are identified as high (G_1) and low (G_2) paid group. Our smallest group (G_3), with relative size of 0.12 (compared to 0.02 for [15]), is distinguished by Batting Average, Hits, On Base Percentage, Triples, Strikeouts, Free Agent and an interacted effect of Runs Batted In and Free Agency Eligibility. Regarding G_1 (high paid) and G_2 (low paid) batters, these are distinguished mainly based on Runs Batted In, Stolen bases, and Free Agency Eligibility. Also interactions of Runs Batted In with Free Agency (zero and

moderate positive effect respectively in G_1 and G_2) and Free Agency Eligibility (strong positive effect in G_1 vs a milder negative effect in G_2) seem to play a role in characterizing batters of high and low paid groups.

7. Conclusions

The current study has been focussed on penalized ML estimation of FMLR under a LASSO penalty. Our contribution within this research line is threefold. First, we have generalized existing penalty specifications by allowing for cluster-specific weighted penalties and tuning parameters, to enhance flexibility. Second, we have generalized the use of the LARS algorithm to select the optimal – in the sense of minimum BIC – mixture LASSO solution. Third, we have derived an EM algorithm to compute the penalized ML estimator that is entirely based on closed form updates.

We have illustrated two alternative implementations of our procedure. One is based on an exhaustive grid search over all combinations of local LASSO solutions. The exhaustive search quickly becomes unfeasible even for a moderate number of covariates. The second implementation performs the covariate selection step locally for each component. From a theoretical point of view, the exhaustive search should be preferred. Nonetheless, in most practical cases where the issue of covariate selection is relevant, this route is far from being computationally feasible.

In the simulation study, we observed that our stepwise estimator based on the local search performs well – and better than its competitors. Notably, it is at least 200 times faster than the approaches it was compared to in all simulation conditions. Results from the empirical application have showed that it also delivers meaningful regression fit, and clustering solution.

From an applied researcher perspective, the local search has the advantage of making standard tasks, e.g. selection of the number of mixture components in penalized FMLR modelling, relatively easy. Indeed, one can look for the best covariate configuration for each choice of $G = 1, \ldots, G_{\max}$ computing our two-step L-LARS. Then, the optimal G is that of the solution with the smallest BIC.

We acknowledge that our penalty can be even further generalized by allowing for a varying within-cluster degree of penalization of the different coefficients – *adaptive within-cluster LASSO penalization*, see [29] for the linear regression case. Another interesting extension would be to consider *group LASSO* penalties [30] for factor selection of grouped variables, both within and between clusters. Further directions for future research include organization of the C++/R code in an R package for research dissemination, reformulation of the penalty to achieve equivariance of the penalized estimator, inference on nonzero regression parameters (SEs/confidence intervals), automated selection of the number of components (as in, e.g. [31]), to mention a few.

Notes

1. www.amstat.org/publications/jse.
2. Covariate acronyms are AVG – Batting Average, OBP – On Base Percentage, R – Runs, H – Hits, 2B – Doubles, 3B – Triples, HR – Home Runs, RBI – Runs Batted In, BB – Walks, SO – Strikeouts, SB – Stolen Bases, ERS – Errors, FAE – Free Agency Eligibility, FA – Free Agent in 1991/92, AE – Arbitration Eligibility, ARB – Arbitration in 1991/92.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Roberto Di Mari http://orcid.org/0000-0001-5498-009X

References

[1] De Veaux RD. Mixtures of linear regressions. Comput Stat Data Anal. 1989;8(3):227–245.
[2] Di Mari R, Rocci R, Gattone SA. Clusterwise linear regression modeling with soft scale constraints. Int J Approx Reason. 2017;91:160–178.
[3] Faria S, Soromenho G. Fitting mixtures of linear regressions. J Stat Comput Simul. 2010;80(2):201–225.
[4] Hurn M, Justel A, Robert CP. Estimating mixtures of regressions. J Comput Graph Stat. 2003;12(1):55–79.
[5] Quandt RE, Ramsey JB. Estimating mixtures of normal distributions and switching regressions. J Am Stat Assoc. 1978;73(364):730–738.
[6] Viele K, Tong B. Modeling with mixtures of linear regressions. Stat Comput. 2002;12(4):315–330.
[7] Dempster A, Laird N, Rubin D. Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Series B (Methodol). 1977;39(1):1–22.
[8] Tibshirani R. Regression shrinkage and selection via the LASSO. J R Stat Soc Series B. 1996;58(1):267–288.
[9] Efron B, Hastie T, Johnstone I, et al. Least angle regression. Ann Stat. 2004;32(2):407–499.
[10] Friedman J, Hastie T, Holger H, et al. Pathwise coordinate optimization. Ann Appl Stat. 2007;1(2):302–332.
[11] Tang Q, Karunamuni RJ. Robust variable selection for finite mixture regression models. Ann Inst Stat Math. 2018;70(3):489–521.
[12] Khalili A, Chen J. Variable selection in finite mixture of regression models. J Am Stat Assoc. 2007;102(479):1025–1038.
[13] Städler N, Bühlmann P, Van De Geer S. ℓ_1-penalization for mixture regression models. Test. 2010;19(2):209–256.
[14] Mortier F, Ouédraogo DY, Claeys F, et al. Mixture of inhomogeneous matrix models for species-rich ecosystems. Environmetrics. 2015;26:39–51.
[15] Lloyd-Jones LR, Nguyen HD, McLachlan GJ. A globally convergent algorithm for LASSO-penalized mixture of linear regression models. Comput Stat Data Anal. 2018;119:19–38.
[16] Lee K-J, Chen R-B. Bayesian variable selection in a finite mixture of linear mixed-effects models. J Stat Comput Simul. 2019;89(13):2434–2453.
[17] Lee K-J, Feldkircher M, Chen Y-C. Variable selection in finite mixture of regression models with an unknown number of components. Comput Stat Data Anal. 2021;158:107–180.
[18] Hennig C. Identifiablity of models for clusterwise linear regression. J Classif. 2000;17(2).273–296.
[19] Grün B, Leisch F. FlexMix version 2: finite mixtures with concomitant variables and varying and constant parameters. J Stat Softw. 2008;28(4):1–35.
[20] Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. J Stat Softw. 2010;33(1):1–22.
[21] Meng X-L, Rubin DB. Maximum likelihood estimation via the ECM algorithm: a general framework. Biometrika. 1993;80(2):267–278.
[22] Wu CJ. On the convergence properties of the EM algorithm. Ann Stat. 1983;11(1):95–103.
[23] Zou H, Hastie T, Tibshirani R. On the "degrees of freedom" of the LASSO. Ann Stat. 2007;35(5):2173–2192.

[24] Bhattacharya S, McNicholas PD. A LASSO-penalized BIC for mixture model selection. Adv Data Anal Classif. 2014;8(1):45–61.
[25] Hubert L, Arabie P. Comparing partitions. J Classif. 1985;2(1):193–218.
[26] Leisch F. Flexmix: a general framework for finite mixture models and latent glass regression in R. J Stat Softw. 2004;11(8):1–18.
[27] Eddelbuettel D, François R. Rcpp: seamless R and C++ integration. J Stat Softw. 2011;40:1–18.
[28] Eddelbuettel D, Sanderson C. RcppArmadillo: accelerating R with high-performance C++ linear algebra. Comput Stat Data Anal. 2014;71:1054–1063.
[29] Zou H. The adaptive LASSO and its oracle properties. J Am Stat Assoc. 2006;101(476):1418–1429.
[30] Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. J R Stat Soc Series B (Stat Methodol. 2006;68(1):49–67.
[31] Chamroukhi F. Unsupervised learning of regression mixture models with unknown number of components. J Stat Comput Simul. 2016;86(12):2308–2334.

Appendices

Appendix 1. Additional tables for the application

Table A1. Baseball batter salary data: BIC values for $G = 2, 3, 4, 5$ components for each of the 20 starts.

	$G = 2$	$G = 3$	$G = 4$	$G = 5$
1	627.231	658.279	644.076	641.566
2	579.775	659.497	667.839	659.355
3	627.231	605.355	747.972	**592.108**
4	627.231	651.779	681.327	658.726
5	**579.775**	660.013	647.418	660.056
6	579.775	680.701	632.510	605.964
7	639.477	640.699	674.731	650.056
8	639.477	**548.660**	636.576	654.938
9	579.775	676.228	673.734	646.113
10	579.775	663.984	622.591	627.468
11	629.916	647.004	634.286	662.622
12	639.477	689.675	668.610	647.664
13	579.410	668.519	**558.971**	672.884
14	579.775	643.116	643.268	662.715
15	639.477	628.215	634.833	713.923
16	579.410	631.183	644.573	668.754
17	627.231	658.279	623.901	693.071
18	579.775	635.950	636.862	629.146
19	627.231	638.088	684.501	652.103
20	629.916	671.758	629.123	690.193

Note: Minimum BIC values for each G in bald.

Appendix 2. Additional tables for the simulation

Tables A2–A3 include results with $\gamma = 0$ and $\delta = 1$. Tables A4–A5 include results with $\gamma = 1$ and $\delta = 0$.

Table A2. Results over all simulation conditions.

	Flexmix	FlexLasso	NelMead	g0d1-Egrid-r	g0d1-2step-r	g0d1-Lgrid-r
			ARI			
Q1	0.638	0.651	0.636	0.669	0.704	0.703
Q2	0.768	0.773	0.760	0.796	0.816	0.814
Q3	0.861	0.873	0.857	0.879	0.897	0.895
			CC			
Q1	5	6	5	6	8	7
Q2	5	7	6	6	9	8
Q3	6	8	6	8	9	9
			MSE			
Q1	0.204	0.162	0.211	0.185	0.131	0.122
Q2	0.429	0.352	0.445	0.493	0.300	0.319
Q3	1.176	0.894	1.276	1.811	1.065	1.167
			Time			
Q1	0.017	1.699	0.080	0.325	0.000	0.266
Q2	0.022	2.276	0.102	0.481	0.004	0.372
Q3	0.032	3.499	0.135	0.699	0.016	0.559

Table A3. Median ARI, CC and MSE: correlated regressors, mod = 2.

Sample size	$n = 100$			$n = 200$		
Mixing proportions	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)
Method			ARI			
Flexmix	0.604	0.712	0.559	0.704	0.837	0.804
FlexLasso	0.604	0.727	0.62	0.704	0.839	0.826
NelMead	0.604	0.694	0.545	0.704	0.837	0.791
g0d1-2step-r	0.636	0.79	0.787	0.721	0.858	0.881
g0d1-Lgrid-r	0.636	0.787	0.787	0.721	0.859	0.859
g0d1-Egrid-r	0.604	0.764	0.695	0.704	0.844	0.807
			CC			
Flexmix	6	6	6	6	6	6
FlexLasso	8	8	7	8	8	7
NelMead	6	6	6	6	6	6
g0d1-2step-r	9	9	7	9	9	8
g0d1-Lgrid-r	8	8	7	9	8	7
g0d1-Egrid-r	6	6	7	6	6	7
			MSE			
Flexmix	0.493	0.855	4.44	0.221	0.327	1.33
FlexLasso	0.404	0.703	2.07	0.199	0.296	0.964
NelMead	0.493	0.845	4.15	0.217	0.326	1.55
g0d1-2step-r	0.306	0.522	5.27	0.162	0.23	1.25
g0d1-Lgrid-r	0.345	0.653	10.4	0.156	0.259	1.45
g0d1-Egrid-r	0.476	0.88	10.4	0.222	0.301	10.4

Table A4. Results over all simulation conditions.

	Flexmix	FlexLasso	NelMead	g1d0-Egrid-r	g1d0-2step-r	g1d0-Lgrid-r
			ARI			
Q1	0.638	0.651	0.636	0.669	0.703	0.702
Q2	0.768	0.773	0.760	0.791	0.809	0.808
Q3	0.861	0.873	0.857	0.877	0.894	0.892
			CC			
Q1	5	6	5	5	8	7
Q2	5	7	6	6	8	8
Q3	6	8	6	6	9	9
			MSE			
Q1	0.204	0.162	0.211	0.210	0.149	0.163
Q2	0.429	0.352	0.445	0.443	0.330	0.366
Q3	1.176	0.894	1.276	1.676	0.947	1.114
			Time			
Q1	0.017	1.699	0.080	1.280	0.000	0.590
Q2	0.022	2.276	0.102	2.038	0.006	0.946
Q3	0.032	3.499	0.135	3.352	0.016	1.403

Table A5. Median ARI, CC and MSE: correlated regressors, mod = 2.

Sample size	$n = 100$			$n = 200$		
Mixing proportions	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)
Method			ARI			
Flexmix	0.604	0.712	0.559	0.704	0.837	0.804
FlexLasso	0.604	0.727	0.62	0.704	0.839	0.826
NelMead	0.604	0.694	0.545	0.704	0.837	0.791
g1d0-2step-r	0.636	0.769	0.818	0.721	0.855	0.892
g1d0-Lgrid-r	0.636	0.768	0.772	0.721	0.85	0.879
g1d0-Egrid-r	0.605	0.768	0.724	0.704	0.843	0.819
			CC			
Flexmix	6	6	6	6	6	6
FlexLasso	8	8	7	8	8	7
NelMead	6	6	6	6	6	6
g1d0-2step-r	9	9	8	9	9	8
g1d0-Lgrid-r	8	8	7	8	8	8
g1d0-Egrid-r	6	6	7	6	6	6
			MSE			
Flexmix	0.493	0.855	4.44	0.221	0.327	1.33
FlexLasso	0.404	0.703	2.07	0.199	0.296	0.964
NelMead	0.493	0.845	4.15	0.217	0.326	1.55
g1d0-2step-r	0.35	0.549	2.38	0.17	0.257	1
g1d0-Lgrid-r	0.408	0.64	10.4	0.179	0.258	1.18
g1d0-Egrid-r	0.451	0.801	10.4	0.221	0.301	10.4

Table A6. Median ARI, CC and MSE: uncorrelated regressors, mod = 1.

Sample size	$n = 100$			$n = 200$		
Mixing proportions	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)
Method			ARI			
Flexmix	0.737	0.869	0.77	0.809	0.915	0.913
FlexLasso	0.772	0.871	0.796	0.827	0.916	0.921
NelMead	0.737	0.864	0.757	0.809	0.913	0.906
g0d1-2step-r	0.772	0.878	0.894	0.827	0.918	0.936
g0d1-Lgrid-r	0.772	0.878	0.894	0.846	0.919	0.93
g0d1-Egrid-r	0.772	0.871	0.875	0.846	0.904	0.913
			CC			
Flexmix	5	5	5	5	5	5
FlexLasso	8	7	7	8	8	7
NelMead	5	5	5	5	5	5
g0d1-2step-r	9	9	9	9	9	9
g0d1-Lgrid-r	9	9	8	10	9	9
g0d1-Egrid-r	10	8	8	10	8	5
			MSE			
Flexmix	0.28	0.368	2.76	0.126	0.16	0.556
FlexLasso	0.246	0.3	1.74	0.111	0.138	0.436
NelMead	0.276	0.365	3.34	0.126	0.159	0.596
g0d1-2step-r	0.194	0.224	1.19	0.0961	0.0969	0.345
g0d1-Lgrid-r	0.168	0.187	1.31	0.0731	0.0792	0.439
g0d1-Egrid-r	0.168	0.403	10	0.0644	0.13	1.08

Table A7. Median ARI, CC and MSE: uncorrelated regressors, mod = 2.

Sample size	$n = 100$			$n = 200$		
Mixing proportions	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)
Method			ARI			
Flexmix	0.573	0.697	0.51	0.687	0.839	0.779
FlexLasso	0.573	0.722	0.552	0.687	0.84	0.789
NelMead	0.573	0.69	0.491	0.687	0.835	0.778
g0d1-2step-r	0.604	0.759	0.728	0.704	0.854	0.857
g0d1-Lgrid-r	0.604	0.757	0.77	0.687	0.851	0.844
g0d1-Egrid-r	0.574	0.727	0.655	0.687	0.847	0.765
			CC			
Flexmix	6	6	6	6	6	6
FlexLasso	8	7	7	8	8	7
NelMead	6	6	6	6	6	6
g0d1-2step-r	9	9	7	9	9	8
g0d1-Lgrid-r	8	8	6	9	8	7
g0d1-Egrid-r	6	6	7	6	6	7
			MSE			
Flexmix	0.357	0.581	3.42	0.151	0.186	1.16
FlexLasso	0.353	0.517	2.21	0.143	0.175	0.862
NelMead	0.353	0.584	3.67	0.15	0.185	1.17
g0d1-2step-r	0.301	0.512	10.4	0.13	0.146	1.38
g0d1-Lgrid-r	0.33	0.533	3.75	0.117	0.134	1.44
g0d1-Egrid-r	0.366	0.645	10.4	0.151	0.178	10.4

Table A8. Median ARI, CC and MSE: correlated regressors, mod = 1.

Sample size	$n = 100$			$n = 200$		
Mixing proportions	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)
Method			ARI			
Flexmix	0.668	0.81	0.642	0.756	0.86	0.851
FlexLasso	0.669	0.81	0.694	0.755	0.875	0.873
NelMead	0.669	0.803	0.605	0.756	0.86	0.851
g0d1-2step-r	0.703	0.833	0.843	0.773	0.879	0.894
g0d1-Lgrid-r	0.703	0.834	0.843	0.773	0.88	0.885
g0d1-Egrid-r	0.669	0.817	0.796	0.773	0.859	0.866
			CC			
Flexmix	5	5	5	5	5	5
FlexLasso	7	7	6	8	8	7
NelMead	5	5	5	5	5	5
g0d1-2step-r	9	8	8	9	9	8
g0d1-Lgrid-r	8	8	7	9	8	8
g0d1-Egrid-r	7	7	7	8	5	5
			MSE			
Flexmix	0.47	0.672	4.52	0.189	0.26	1.27
FlexLasso	0.394	0.485	2.35	0.17	0.183	0.849
NelMead	0.471	0.662	5.03	0.183	0.261	1.47
g0d1-2step-r	0.276	0.345	2.96	0.123	0.14	1.04
g0d1-Lgrid-r	0.298	0.391	2.61	0.107	0.13	1.11
g0d1-Egrid-r	0.438	0.821	10	0.15	0.228	1.74

Table A9. Median ARI, CC and MSE: uncorrelated regressors, mod = 1.

Sample size	$n = 100$			$n = 200$		
Mixing proportions	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)
Method			ARI			
Flexmix	0.737	0.869	0.77	0.809	0.915	0.913
FlexLasso	0.772	0.871	0.796	0.827	0.916	0.921
NelMead	0.737	0.864	0.757	0.809	0.913	0.906
g1d0-2step-r	0.772	0.876	0.875	0.827	0.916	0.933
g1d0-Lgrid-r	0.772	0.874	0.857	0.827	0.916	0.929
g1d0-Egrid-r	0.737	0.871	0.857	0.809	0.916	0.913
			CC			
Flexmix	5	5	5	5	5	5
FlexLasso	8	7	7	8	8	7
NelMead	5	5	5	5	5	5
g1d0-2step-r	9	9	8	9	9	8
g1d0-Lgrid-r	8	7	7	9	8	7
g1d0-Egrid-r	5	5	5	5	5	5
			MSE			
Flexmix	0.28	0.368	2.76	0.126	0.16	0.556
FlexLasso	0.246	0.3	1.74	0.111	0.138	0.436
NelMead	0.276	0.365	3.34	0.126	0.159	0.596
g1d0-2step-r	0.216	0.275	1.69	0.103	0.114	0.442
g1d0-Lgrid-r	0.239	0.321	2.19	0.103	0.128	0.503
g1d0-Egrid-r	0.275	0.366	10	0.126	0.159	0.679

Table A10. Median ARI, CC and MSE: uncorrelated regressors, mod = 2.

Sample size	$n = 100$			$n = 200$		
Mixing proportions	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)
Method			ARI			
Flexmix	0.573	0.697	0.51	0.687	0.839	0.779
FlexLasso	0.573	0.722	0.552	0.687	0.84	0.789
NelMead	0.573	0.69	0.491	0.687	0.835	0.778
g1d0-2step-r	0.605	0.759	0.74	0.696	0.855	0.866
g1d0-Lgrid-r	0.586	0.745	0.728	0.687	0.852	0.847
g1d0-Egrid-r	0.603	0.735	0.695	0.687	0.849	0.772
			CC			
Flexmix	6	6	6	6	6	6
FlexLasso	8	7	7	8	8	7
NelMead	6	6	6	6	6	6
g1d0-2step-r	9	8	8	9	9	8
g1d0-Lgrid-r	8	8	7	8	8	8
g1d0-Egrid-r	6	6	6	6	6	7
			MSE			
Flexmix	0.357	0.581	3.42	0.151	0.186	1.16
FlexLasso	0.353	0.517	2.21	0.143	0.175	0.862
NelMead	0.353	0.584	3.67	0.15	0.185	1.17
g1d0-2step-r	0.316	0.505	2.42	0.135	0.157	0.923
g1d0-Lgrid-r	0.356	0.496	10.4	0.135	0.166	1.24
g1d0-Egrid-r	0.349	0.537	10.4	0.151	0.18	10.4

Table A11. Median ARI, CC and MSE: correlated regressors, mod = 1.

Sample size	$n = 100$			$n = 200$		
Mixing proportions	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)	(0.5,0.5)	(0.3,0.7)	(0.1,0.9)
Method			ARI			
Flexmix	0.668	0.81	0.642	0.756	0.86	0.851
FlexLasso	0.669	0.81	0.694	0.755	0.875	0.873
NelMead	0.669	0.803	0.605	0.756	0.86	0.851
g1d0-2step-r	0.703	0.834	0.796	0.773	0.876	0.885
g1d0-Lgrid-r	0.669	0.835	0.804	0.773	0.878	0.885
g1d0-Egrid-r	0.669	0.815	0.796	0.756	0.862	0.866
			CC			
Flexmix	5	5	5	5	5	5
FlexLasso	7	7	6	8	8	7
NelMead	5	5	5	5	5	5
g1d0-2step-r	8	8	7	9	9	8
g1d0-Lgrid-r	8	7	7	8	8	7
g1d0-Egrid-r	5	5	6	5	5	5
			MSE			
Flexmix	0.47	0.672	4.52	0.189	0.26	1.27
FlexLasso	0.394	0.485	2.35	0.17	0.183	0.849
NelMead	0.471	0.662	5.03	0.183	0.261	1.47
g1d0-2step-r	0.339	0.449	2.91	0.131	0.154	1.01
g1d0-Lgrid-r	0.356	0.473	4.33	0.132	0.164	1.08
g1d0-Egrid-r	0.457	0.654	10	0.188	0.258	1.75