Statistica Sinica **34** (2024), 2115-2139
doi:https://doi.org/10.5705/ss.202022.0092

GAUSSIAN MIXTURE MODELS WITH CONCAVE PENALIZED FUSION

Yiwei Fan*[1] and Guosheng Yin[2,3]

[1]*Beijing Institute of Technology,* [2]*The University of Hong Kong and* [3]*Imperial College London*

Abstract: Estimating finite mixture models is a fundamental and challenging problem. We propose a penalized method for a Gaussian mixture linear regression, where the error terms follow a location–scale mixture of Gaussian distributions. The objective function is a combination of the likelihood function of the observed data and a penalty on the pairwise differences of the parameters. We develop an alternating direction method of multipliers algorithm, and establish its convergence property. By clustering and merging similar observations in an automatic manner, our method provides an integrated tool for simultaneously determining the number of components and estimating the parameters in finite mixture models. Moreover, the proposed method allows the mean and precision parameters to have different structures, enabling us to obtain pooled estimators. We also establish the statistical properties of our estimators. Extensive simulations and real-data examples are presented to evaluate the numerical performance of the proposed method.

Key words and phrases: Alternating direction method of multipliers, consistency, linear regression, pairwise difference, pooled estimator

1. Introduction

Finite mixture models are often used to model heterogeneous data from complex distributions, in areas such as density estimation (Escobar and West (1995)), pattern clustering (Liu et al. (2022)), and quality control (Li et al. (2021)). As the most popular mixture model, the Gaussian mixture model (GMM) possesses many appealing features, including computational tractability, affine invariance, and flexibility of representations.

Several methods have been proposed to estimate the parameters of GMMs. Owing to missing information in the component membership, the complete-data likelihood cannot be calculated directly. The expectation–maximization (EM) algorithm is often used to estimate the parameters for a given number of components, which is usually unknown in practice. To determine the number of components, conventional methods often take a model-selection approach using the Akaike information criterion (AIC) or the Bayesian information criterion (BIC). Leroux (1992) show that the number of components estimated using

*Corresponding author.

the AIC or BIC is at least as large as the true number. Another approach is using penalized methods, which jointly learn the cluster structures and estimate the parameters. Chen and Khalili (2009) impose two penalty functions on the mixing proportions and the location parameters in GMMs, but do not consider heterogeneity among precisions. Huang, Peng and Zhang (2017) propose a new penalized likelihood method for multivariate GMMs in which they penalize the mixing probabilities, which can be applied to location–scale mixtures. Hao et al. (2018) introduce a joint graphical lasso penalty on the elements of the precision matrices to extract both homogeneity and heterogeneity components for high-dimensional Gaussian graphical mixture models. Recently, Ren et al. (2022) conducted heterogeneity analyses for Gaussian graphical models by imposing fusion penalties on the mean and on the precision matrix parameters, thus determining the number of components and estimating the parameters in an automated way.

In a linear regression, when the distribution of the error terms deviates significantly from normality, an effective strategy is to assume that the error terms follow a mixture of Gaussian distributions. As suggested by Rossi (2014), any distribution can be approximated by a Gaussian mixture, to a sufficient level of accuracy, by using an adequate number of components. The EM algorithm can be generalized naturally to the regression setting (Bartolucci and Scaccia (2005)), which also requires a specification of the number of mixture components. Together, the linear regression mixture model and the pairwise fusion penalty can accommodate subject-specific intercepts (Ma and Huang (2017)), but focus only on the skewness of the errors as a departure from normality, without considering heterogeneity among precisions. Motivated by this work, we consider a more general framework that incorporates the heterogeneity among both the means and the precisions. These location–scale Gaussian mixtures are expected to perform better in a heteroscedastic model, for example, when the errors follow a leptokurtic distribution. Our objective function is a combination of the likelihood of the observed data and a concave fusion penalty that measures the pairwise differences of the means and the precisions. An alternating direction method of multipliers (ADMM) algorithm is developed for optimization. As the weight for the penalty term increases, some pairwise differences of the estimated parameters shrink to zero, enabling us to identify the number of components, and estimate the parameters.

Our work builds on, but differs from existing works on GMMs (Huang, Peng and Zhang (2017); Hao et al. (2018); Ren et al. (2022)) by considering regression settings. Our framework is motivated by the penalization and shrinkage strategies in existing heterogeneity studies (Ma and Huang (2017)), extending them to accommodate heterogeneity among precisions. More importantly, most existing GMM methods assume that either the means and the precisions share the same structure, or that the precisions of all components are equal. An appealing

feature of our method is that it allows the means and precisions to have different structures, thus obtaining more accurate pooled estimators than those of previous studies from the perspective of computation and theory. In particular, we conduct a rigorous theoretical investigation of our pooled estimators. Lastly, we extensively investigate the numerical convergence of the ADMM algorithm for optimization with finite samples when using nonconvex penalties.

The rest of this paper is organized as follows. In Section 2, we develop a penalized method for a linear regression in order to identify the number of components in a GMM using pairwise fusion and estimate the unknown parameters. In Section 3, we develop an ADMM algorithm to facilitate the computation, and establish the statistical properties in Section 4. In Section 5, we conduct extensive simulations to evaluate the numerical performance of the proposed method. Section 6 demonstrates the feasibility of our method based on real data. Finally, Section 7 concludes the paper with a discussion.

2. Gaussian Mixture with Pairwise Fusion

For any vector $\boldsymbol{u}$, let $\|\boldsymbol{u}\|_2$ denote its L_2-norm, and $\boldsymbol{u}^{-1}$ denote a vector with components that are the reciprocals of those of $\boldsymbol{u}$. For a matrix $\boldsymbol{\Theta}$, let $\boldsymbol{\theta}_{[i\cdot]}$ and $\boldsymbol{\theta}_{[\cdot i]}$ denote its ith row and ith column, respectively. Let $y \in \mathbb{R}$ be the response variable and $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^p$ be the p-dimensional vector of covariates. We consider the linear regression model,

$$y_i = \boldsymbol{\beta}^\top \boldsymbol{x}_i + \epsilon_i, \quad i = 1, \ldots, n,$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ is the vector of unknown coefficients, ϵ_i is the random error, and n is the sample size. Let

$$\phi(z; \mu, \tau) = \left(\frac{\tau}{2\pi}\right)^{1/2} \exp\left\{-\frac{\tau(z-\mu)^2}{2}\right\},$$

which is the density function of a Gaussian distribution with mean μ and precision τ. Suppose that ϵ_i is from a Gaussian distribution $\phi(\epsilon_i; \theta_{i1}, \theta_{i2})$, where θ_{i1} and θ_{i2} are the mean and precision, respectively, for subject i. The data heterogeneity between ϵ_i and ϵ_j is represented by the difference between two vectors, namely, $\boldsymbol{\theta}_{[i\cdot]} = (\theta_{i1}, \theta_{i2})^\top$ and $\boldsymbol{\theta}_{[j\cdot]} = (\theta_{j1}, \theta_{j2})^\top$. If ϵ_i and ϵ_j are from the same component, then $\boldsymbol{\theta}_{[i\cdot]} = \boldsymbol{\theta}_{[j\cdot]}$; otherwise, $\boldsymbol{\theta}_{[i\cdot]} \neq \boldsymbol{\theta}_{[j\cdot]}$, meaning that at least one element is not equal. Suppose there are K_m distinct values in $\boldsymbol{\theta}_{[\cdot m]}$, for $m = 1, 2$, denoted by $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{K_1})^\top$ and $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{K_2})^\top$, respectively, where $\mu_i \neq \mu_j$ and $\tau_i \neq \tau_j$, for any $i \neq j$. Given K_1 and K_2, one can apply the EM algorithm to estimate $\boldsymbol{\beta}$, $\boldsymbol{\theta}_{[\cdot 1]}$, and $\boldsymbol{\theta}_{[\cdot 2]}$ by introducing latent variables (Bartolucci and Scaccia (2005)). However, in practice, it is difficult to identify the values of K_1 and K_2. By introducing pairwise fusion penalties, we propose a novel approach to automatically determine the number of components and simultaneously estimate

the parameters.

Let $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$ and $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$ denote the observed data. The log-likelihood function is

$$L(\boldsymbol{\beta}, \boldsymbol{\Theta}) = \sum_{i=1}^{n} \log \phi(\epsilon_i; \theta_{i1}, \theta_{i2}) = \sum_{i=1}^{n} \log \phi(y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i; \theta_{i1}, \theta_{i2}).$$

Our goal is to identify the values of K_1 and K_2, and to estimate $\boldsymbol{\beta}$ and $\boldsymbol{\Theta}$. We introduce a fusion penalty (Tibshirani et al. (2005)) to penalize the pairwise differences between θ_{im}, encouraging the sparsity of the pairwise differences. The objective function is

$$Q(\boldsymbol{\beta}, \boldsymbol{\Theta}) = -L(\boldsymbol{\beta}, \boldsymbol{\Theta}) + \sum_{m=1}^{2} \sum_{1 \le i < j \le n} p(|\theta_{im} - \theta_{jm}|, \lambda_m, \gamma_m), \tag{2.1}$$

where $p(\cdot, \lambda, \gamma)$ is a penalty function with tuning parameters λ and γ.

It is critical to choose an appropriate penalty function $p(\cdot, \lambda, \gamma)$. The L_1-penalty, which is similar to the least absolute shrinkage and selection operator (lasso) (Tibshirani (1996)), with $p(|\theta_{im} - \theta_{jm}|, \lambda_m, \gamma_m) = \lambda_m |\theta_{im} - \theta_{jm}|$, penalizes all paired differences $|\theta_{im} - \theta_{jm}|$. The L_1-penalty tends to overshrink large coefficients, and fails to recover the group structure (Fan and Li (2001); Zou (2006)). On the other hand, there are established theories for nonconvex penalties, such as hard penalties and the smoothly clipped absolute deviation (SCAD) penalty. The hard penalty defined in Antoniadis (1997) takes the form

$$p(u, \lambda, \gamma) = \left(\frac{-u^2}{2} + \lambda |u| \right) I(|u| < \lambda) + \left(\frac{\lambda^2}{2} \right) I(|u| \ge \lambda), \tag{2.2}$$

where $I(\cdot)$ is the indicator function. The SCAD penalty function is $p(u, \lambda, \gamma) = \lambda \int_0^u \min\{1, (\gamma - t/\lambda)_+/(\gamma - 1)\} dt$, where $t_+ = tI(t \ge 0)$ denotes the nonnegative part of $t \in \mathbb{R}$, with $\lambda \ge 0$ and $\gamma > 2$. The tuning parameter γ controls the concavity of the SCAD penalty function, that is, how fast the penalization rate goes to zero. As $\gamma \to \infty$, it reduces to the L_1-penalty. A smaller γ results in more concavity and less bias, but the estimates become unstable, because there is a greater chance of multiple local minima. These nonconvex penalties achieve sparsity at individual levels and, more importantly, lead to an approximately unbiased estimation of the coefficients and correctly shrink group differences, with high probability, under regular conditions.

3. The ADMM Algorithm

Because the penalty function is not separable in θ_{im} for $m = 1, 2$, it is difficult to directly minimize the objective function in (2.1). To overcome this challenge, we introduce new variables, defined as $\Delta_{ijm} = \theta_{im} - \theta_{jm}$, for $1 \le i < j \le n$ and

$m = 1, 2$. Let $\boldsymbol{\Delta} = \{(\Delta_{ij1}, \Delta_{ij2})^\top, i < j\}$. The optimization problem is

$$\begin{aligned} &\min Q(\boldsymbol{\beta}, \boldsymbol{\Theta}, \boldsymbol{\Delta}) = -L(\boldsymbol{\beta}, \boldsymbol{\Theta}) + \sum_{m=1}^{2} \sum_{1 \le i < j \le n} p(|\Delta_{ijm}|, \lambda_m, \gamma_m), \\ &\text{subject to} \quad \theta_{im} - \theta_{jm} - \Delta_{ijm} = 0, \quad 1 \le i < j \le n; m = 1, 2. \end{aligned} \tag{3.1}$$

Using the augmented Lagrangian method, we can estimate the parameters by minimizing

$$\begin{aligned} H(\boldsymbol{\beta}, \boldsymbol{\Theta}, \boldsymbol{\Delta}, \boldsymbol{\nu}) = Q(\boldsymbol{\beta}, \boldsymbol{\Theta}, \boldsymbol{\Delta}) + \sum_{m=1}^{2} \sum_{i<j} \nu_{ijm}(\theta_{im} - \theta_{jm} - \Delta_{ijm}) \\ + \frac{\rho}{2} \sum_{m=1}^{2} \sum_{i<j} (\theta_{im} - \theta_{jm} - \Delta_{ijm})^2, \end{aligned}$$

where $\boldsymbol{\nu} = \{(\nu_{ij1}, \nu_{ij2})^\top, i < j\}$ are Lagrangian multipliers, and $\rho > 0$ is the penalty parameter. To implement the ADMM algorithm (Boyd et al. (2011)), we first derive the updating equations for $\boldsymbol{\beta}$ and $\boldsymbol{\theta}_{[\cdot 1]}$. We aim to minimize

$$\begin{aligned} H(\boldsymbol{\beta}, \boldsymbol{\Theta}, \boldsymbol{\Delta}, \boldsymbol{\nu}) &= -L(\boldsymbol{\beta}, \boldsymbol{\Theta}) + \frac{\rho}{2} \sum_{i<j} \left\{ (\boldsymbol{e}_i - \boldsymbol{e}_j)^\top \boldsymbol{\theta}_{[\cdot 1]} - \Delta_{ij1} + \rho^{-1} \nu_{ij1} \right\}^2 + C, \\ &= -L(\boldsymbol{\beta}, \boldsymbol{\Theta}) + \frac{\rho}{2} \left\| \boldsymbol{E}\boldsymbol{\theta}_{[\cdot 1]} - \boldsymbol{\Delta}_{[\cdot 1]} + \rho^{-1} \boldsymbol{\nu}_{[\cdot 1]} \right\|_2^2 + C, \end{aligned}$$

where $\boldsymbol{e}_i$ is a vector of length n of zeros except for the ith element being one, $\boldsymbol{E} = \{(\boldsymbol{e}_i - \boldsymbol{e}_j), i < j\}^\top$, $\boldsymbol{\Delta}_{[\cdot 1]} = \{\Delta_{ij1}, i < j\}^\top$, $\boldsymbol{\nu}_{[\cdot 1]} = \{\nu_{ij1}, i < j\}^\top$, and C is a generic symbol for a constant. As shown in the Supplementary Material, given the current estimates $\boldsymbol{\theta}_{[\cdot 2]}^{(t)}$, $\boldsymbol{\Delta}_{[\cdot 1]}^{(t)}$, and $\boldsymbol{\nu}_{[\cdot 1]}^{(t)}$, the updating equations for $\boldsymbol{\theta}_{[\cdot 1]}$ and $\boldsymbol{\beta}$ at the $(t+1)$th iteration are

$$\boldsymbol{\theta}_{[\cdot 1]}^{(t+1)} = \left(\rho \boldsymbol{E}^\top \boldsymbol{E} + \boldsymbol{A}^{(t)} \right)^{-1} \left\{ \boldsymbol{A}^{(t)} \boldsymbol{y} + \rho \boldsymbol{E}^\top \left(\boldsymbol{\Delta}_{[\cdot 1]}^{(t)} - \rho^{-1} \boldsymbol{\nu}_{[\cdot 1]}^{(t)} \right) \right\}, \tag{3.2}$$

$$\boldsymbol{\beta}^{(t+1)} = (\boldsymbol{X}^\top \boldsymbol{W}^{(t)} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}^{(t)} (\boldsymbol{y} - \boldsymbol{\theta}_{[\cdot 1]}^{(t+1)}), \tag{3.3}$$

respectively, where $\boldsymbol{A}^{(t)} = \boldsymbol{W}^{(t)} \{ \boldsymbol{I}_n - \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{W}^{(t)} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}^{(t)} \}$, with $\boldsymbol{W}^{(t)}$ a diagonal matrix of $\boldsymbol{\theta}_{[\cdot 2]}^{(t)}$ and $\boldsymbol{I}_n$ an $n \times n$ identity matrix.

Because there is no closed form when we update θ_{i2} simultaneously, we use a cyclic coordinate descent scheme at the $(t+1)$th iteration. Specifically, we cycle through θ_{i2}, for $i = 1, \ldots, n$, so that at the ith step, we update $\theta_{i2}^{(t+1)}$, while holding all other $\{\theta_{j2}^{(t+1)}, j \neq i\}$ fixed as

$$\theta_{i2}^{(t+1)} = \{2\rho(n-1)\}^{-1} \left\{ -b_i^{(t+1)} + \sqrt{(b_i^{(t+1)})^2 + 2\rho(n-1)} \right\}, \tag{3.4}$$

with

$$b_i^{(t+1)} = \frac{\{y_i - (\boldsymbol{\beta}^{(t+1)})^\top \boldsymbol{x}_i - \theta_{i1}^{(t+1)}\}^2}{2} + \sum_{j>i} \nu_{ij2}^{(t)} - \sum_{j<i} \nu_{ji2}^{(t)} - \rho \left\{ \sum_{j>i} (\theta_{j2}^{(t+1)} + \Delta_{ij2}^{(t)}) + \sum_{j<i} (\theta_{j2}^{(t+1)} - \Delta_{ji2}^{(t)}) \right\},$$

for which the derivation is given in the Supplementary Material. The updating of $\boldsymbol{\theta}_{[\cdot 2]} = (\theta_{12}, \ldots, \theta_{n2})^\top$ at the $(t+1)$th iteration proceeds by applying (3.4) repeatedly in a cyclical manner, until the relative distance of the parameters between two cycles is smaller than a tolerance (e.g., 10^{-3}).

To update $\boldsymbol{\Delta}_{[\cdot m]}$, for $m = 1, 2$, we minimize

$$\sum_{i<j} p(|\Delta_{ijm}|, \lambda_m, \gamma_m) + \sum_{i<j} \nu_{ijm}(\theta_{im} - \theta_{jm} - \Delta_{ijm}) + \frac{\rho}{2} \sum_{i<j} (\theta_{im} - \theta_{jm} - \Delta_{ijm})^2$$
$$= 2^{-1}\rho \sum_{i<j} \left\{ \left(\theta_{im} - \theta_{jm} + \frac{\nu_{ijm}}{\rho} - \Delta_{ijm} \right)^2 \right\} + \sum_{i<j} p(|\Delta_{ijm}|, \lambda_m, \gamma_m) + C, \quad (3.5)$$

where C is a constant. For simplicity, let $r_{ijm} = \theta_{im} - \theta_{jm} + \nu_{ijm}/\rho$ and $\boldsymbol{r}_{[\cdot m]} = (r_{ijm})_{i<j}$. Minimizing (3.5) with respect to $\boldsymbol{\Delta}_{[\cdot m]}$ is equivalent to solving the penalized linear regression problem

$$\min \left\{ 2^{-1} \|\boldsymbol{r}_{[\cdot m]} - \boldsymbol{\Delta}_{[\cdot m]}\|^2 + \rho^{-1} \sum_{i<j} p(|\Delta_{ijm}|, \lambda_m, \gamma_m) \right\}. \quad (3.6)$$

Because the design matrix in (3.6) is orthogonal, even for nonconvex penalties such as the hard penalty and SCAD, it still often results in a unique solution, as suggested by She (2009). As shown in the Supplementary Material, the updating equation for Δ_{ijm} under the hard penalty (2.2) is

$$\Delta_{ijm}^{(t+1)} = \begin{cases} \dfrac{\mathcal{S}(r_{ijm}^{(t+1)}, \rho^{-1}\lambda_m)}{1 - \rho^{-1}}, & \text{if } |r_{ijm}^{(t+1)}| < \lambda_m, \\ r_{ijm}^{(t+1)}, & \text{if } |r_{ijm}^{(t+1)}| \geq \lambda_m, \end{cases} \quad (3.7)$$

where $\mathcal{S}(u, c) = \text{sign}(u)(|u| - c)_+$ is the soft-thresholding function. In addition, the updating equation of Δ_{ijm} under the SCAD penalty with $\gamma_m > (1 + \rho^{-1})$ is

$$\Delta_{ijm}^{(t+1)} = \begin{cases} \mathcal{S}(r_{ijm}^{(t+1)}, \rho^{-1}\lambda_m), & \text{if } |r_{ijm}^{(t+1)}| \leq \lambda_m(1 + \rho^{-1}), \\ \dfrac{\mathcal{S}(r_{ijm}^{(t+1)}, \gamma_m\rho^{-1}\lambda_m/(\gamma_m - 1))}{1 - \rho^{-1}/(\gamma_m - 1)}, & \text{if } (1 + \rho^{-1})\lambda_m < |r_{ijm}^{(t+1)}| \leq \gamma_m\lambda_m, \\ r_{ijm}^{(t+1)}, & \text{if } |r_{ijm}^{(t+1)}| > \gamma_m\lambda_m. \end{cases} \quad (3.8)$$

Finally, for $1 \le i < j \le n$ and $m = 1, 2$, ν_{ijm} is updated using

$$\nu_{ijm}^{(t+1)} = \nu_{ijm}^{(t)} + \rho \left(\theta_{im}^{(t+1)} - \theta_{jm}^{(t+1)} - \Delta_{ijm}^{(t+1)} \right). \tag{3.9}$$

Define the primal and dual residuals as $\boldsymbol{R}_p(\boldsymbol{\Theta}, \boldsymbol{\Delta}) = \boldsymbol{E}\boldsymbol{\Theta} - \boldsymbol{\Delta}$ and $\boldsymbol{R}_d^{(t)} = \rho \boldsymbol{E}^\top (\boldsymbol{\Delta}^{(t+1)} - \boldsymbol{\Delta}^{(t)})$, respectively. Boyd et al. (2011) suggest that a reasonable termination criterion for the ADMM algorithm is $\|\boldsymbol{R}_p(\boldsymbol{\Theta}^{(t)}, \boldsymbol{\Delta}^{(t)})\|_F \le \kappa^{\text{pri}}$ and $\|\boldsymbol{R}_d^{(t)}\|_F \le \kappa^{\text{dual}}$, with

$$\begin{aligned} \kappa^{\text{pri}} &= \sqrt{\frac{n(n-1)}{2}} \kappa^{\text{abs}} + \kappa^{\text{rel}} \max(\|\boldsymbol{E}\boldsymbol{\Theta}^{(t)}\|_F, \|\boldsymbol{\Delta}^{(t)}\|_F), \\ \kappa^{\text{dual}} &= \sqrt{n} \kappa^{\text{abs}} + \kappa^{\text{rel}} \|\boldsymbol{E}^\top \boldsymbol{\nu}^{(t)}\|_F, \end{aligned} \tag{3.10}$$

where $\|\cdot\|_F$ is the Frobenius norm, κ^{abs} is an absolute tolerance, κ^{rel} is a relative tolerance, and both of the latter are small positive numbers. The detailed procedure for estimating $\boldsymbol{\beta}$, $\boldsymbol{\theta}_{[\cdot 1]}$, and $\boldsymbol{\theta}_{[\cdot 2]}$ is summarized in Algorithm S1 in the Supplementary Material.

We now analyze the computational complexity of Algorithm S1. Because $\boldsymbol{E}^\top \boldsymbol{E}$ is computed in advance with complexity $O(n^4)$, we do not need to compute it again in the loops. Updating $\boldsymbol{\theta}_{[\cdot 1]}^{(t+1)}$ has complexity $O(p^3 + n^3)$. When updating $\boldsymbol{\beta}^{(t+1)}$, note that $(\boldsymbol{X}^\top \boldsymbol{W}^{(t)} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}^{(t)}$ has been computed when updating $\boldsymbol{\theta}_{[\cdot 1]}^{(t+1)}$, and thus need not be computed again. As a result, the complexity of updating $\boldsymbol{\beta}^{(t+1)}$ is $O(pn)$. The computational complexity of updating $\boldsymbol{\theta}_{[\cdot 2]}^{(t+1)}$ is $O((n+p)N_{\text{C}})$, where N_{C} is the iterative number of our coordinate descent method. Finally, updating $\boldsymbol{\Delta}^{(t+1)}$ and $\boldsymbol{\nu}^{(t+1)}$ has complexity $O(n^2)$. Therefore, for each loop of the ADMM method, the overall computational complexity is $O(p^3 + n^3 + (n+p)N_{\text{C}})$. Next, we compare the computational complexity of our method with that of Ma and Huang (2017), who do not consider heterogeneity among precisions. First, many operations for updating the mean vector in Ma and Huang (2017) can be computed in advance, rather than in each loop. For example, in (3.2), we need to compute $(\boldsymbol{X}^\top \boldsymbol{W}^{(t)} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W}^{(t)}$ in each loop, whereas in the updating equation for means in Ma and Huang (2017), $(\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top$ can be computed in advance. Second, there is no need to update precisions in Ma and Huang (2017). Third, the sizes of $\boldsymbol{\Delta}^{(t+1)}$ and $\boldsymbol{\nu}^{(t+1)}$ in Ma and Huang (2017) are half of those in our method. As a result, in Ma and Huang (2017), the computational complexity for operations in advance is $O(p^3 + n^4)$, and that for operations in each loop is $O(pn + n^3)$. When $p \le n$, for operations in advance, the computational complexity of both methods is $O(n^4)$; for each loop of the ADMM algorithm, the computational complexity is $O(n^3 + nN_{\text{C}})$ in our method, and $O(n^3)$ in the method of Ma and Huang (2017). Therefore, when $p \le n$, the increase in computational complexity of our method is due mainly to updating precisions.

The convergence of the ADMM in nonconvex optimization has been studied extensively, for example, by Wang, Yin and Zeng (2019). However, to the best of our knowledge, existing conclusions in the literature cannot be applied directly to establish the convergence of Algorithm S1. Nevertheless, we can still prove it following similar steps to those in Wang, Yin and Zeng (2019), with some modifications. We first present two lemmas, and then show the convergence property of our ADMM algorithm.

Lemma 1. *Assume the penalty function $p(\cdot, \lambda, \gamma)$ is weakly convex with modulus C_p and its subdifferential is bounded, that is, $|\partial p(\cdot, \lambda, \gamma)| \leq C_s$, for some constant C_s. If $\rho > C_p$, it holds that the augmented Lagrangian $H(\boldsymbol{\beta}^{(t)}, \boldsymbol{\Theta}^{(t)}, \boldsymbol{\Delta}^{(t)}, \boldsymbol{\nu}^{(t)})$ is lower bounded.*

Lemma 2. *Under the assumption in Lemma* 1*, it holds that*

$$\begin{aligned} &H(\boldsymbol{\beta}^{(t)}, \boldsymbol{\Theta}^{(t)}, \boldsymbol{\Delta}^{(t)}, \boldsymbol{\nu}^{(t)}) - H(\boldsymbol{\beta}^{(t-1)}, \boldsymbol{\Theta}^{(t-1)}, \boldsymbol{\Delta}^{(t-1)}, \boldsymbol{\nu}^{(t-1)}) \\ &\leq 4\rho^{-1} n(n-1) C_s^2 - \frac{\rho}{2} \|\boldsymbol{E}\boldsymbol{\Theta}^{(t)} - \boldsymbol{E}\boldsymbol{\Theta}^{(t-1)}\|_F^2 - \frac{\rho - C_p}{2} \|\boldsymbol{\Delta}^{(t)} - \boldsymbol{\Delta}^{(t-1)}\|_F^2. \end{aligned}$$

Theorem 1. *Under the assumption in Lemma* 1*, if $\rho > C_p$, the following hold:*

*(1) the primal residual $\boldsymbol{R}_p(\boldsymbol{\Theta}^{(t)}, \boldsymbol{\Delta}^{(t)})$ and the dual residual $\boldsymbol{R}_d^{(t)}$ of Algorithm S*1 *satisfy $\lim_{t\to\infty} \|\boldsymbol{R}_p(\boldsymbol{\Theta}^{(t)}, \boldsymbol{\Delta}^{(t)})\|_F = 0$ and $\lim_{t\to\infty} \|\boldsymbol{R}_d^{(t)}\|_F = 0$, respectively.*

(2) the sequence $\{\boldsymbol{\beta}^{(t)}, \boldsymbol{\Theta}^{(t)}, \boldsymbol{\Delta}^{(t)}, \boldsymbol{\nu}^{(t)}\}$ has at least a limit point $\{\boldsymbol{\beta}^*, \boldsymbol{\Theta}^*, \boldsymbol{\Delta}^*, \boldsymbol{\nu}^*\}$, and any limit point is a stationary point.

Both the hard and SCAD penalties are weakly convex, and their subdifferentials are bounded by constants. Lemma 1 shows that, for sufficiently large ρ, the augmented Lagrangian is lower bounded, and Lemma 2 shows that its change between successive iterations is upper bounded. Then, Theorem 1 presents that the ADMM algorithm achieves primal feasibility and dual feasibility. Moreover, it converges to an optimal solution, which may be a local minimum. The proof is given in the Supplementary Material.

Given the tuning parameters, the pairwise fused penalty may result in $\Delta_{ijm} = 0$ for some i and j. As discussed in Section 2, we assume that ϵ_i and ϵ_j are from the same component if $\Delta_{ij1} = \Delta_{ij2} = 0$. Therefore, we can recover the group structure of the errors using the shrinkage procedure. Denote the estimates as $\widehat{\boldsymbol{\beta}}$, $\widehat{\boldsymbol{\theta}}_{[\cdot 1]}$, and $\widehat{\boldsymbol{\theta}}_{[\cdot 2]}$. As a result, we have $\widehat{K}_1$ estimated distinct values for the mean, which divide the data into groups $\widehat{\mathcal{G}}_1^{(1)}, \ldots, \widehat{\mathcal{G}}_{\widehat{K}_1}^{(1)}$. The estimated mean for the kth group is $\hat{\mu}_k = |\widehat{\mathcal{G}}_k^{(1)}|^{-1} \sum_{i \in \widehat{\mathcal{G}}_k^{(1)}} \hat{\theta}_{i1}$, where $|\cdot|$ is the cardinality of a set. Similarly, there are $\widehat{K}_2$ estimated distinct values for the precision, which divide the data into groups $\widehat{\mathcal{G}}_1^{(2)}, \ldots, \widehat{\mathcal{G}}_{\widehat{K}_2}^{(2)}$. The estimated precision for the k'th group is $\hat{\tau}_{k'} = |\widehat{\mathcal{G}}_{k'}^{(2)}|^{-1} \sum_{i \in \widehat{\mathcal{G}}_{k'}^{(2)}} \hat{\theta}_{i2}$.

4. Asymptotic Properties

4.1. Heterogeneous model

We first study the theoretical properties of the proposed estimator under a heterogeneous model, where at least two components exist in the mixture, that is, $\max(K_1, K_2) \geq 2$. We discuss the homogeneous setting in the next section. We show that under some regularity conditions, there exists a local minimizer of the objective function converging to the true parameter. Specifically, we first prove that the oracle estimator converges to the true parameter, and then show that the oracle estimator is a local minimizer of the objective function, with probability approaching one. Let $\boldsymbol{\beta}^0$, $\boldsymbol{\theta}^0_{[\cdot 1]}$, and $\boldsymbol{\theta}^0_{[\cdot 2]}$ denote the true parameters. For $m = 1, 2$, suppose there are K_m distinct values in $\boldsymbol{\theta}^0_{[\cdot m]}$, which divide $\{\epsilon_i\}_{i=1}^n$ into K_m groups, $\mathcal{G}^{(m)}_1, \ldots, \mathcal{G}^{(m)}_{K_m}$. Let $\mathcal{I}_{\mathcal{G}^{(m)}}$ be the subspace of $\mathbb{R}^n$, defined as $\mathcal{I}_{\mathcal{G}^{(m)}} = \{\boldsymbol{\theta}_{[\cdot m]} \in \mathbb{R}^n : \theta_{im} = \theta_{jm} \text{ for any } i, j \in \mathcal{G}^{(m)}_k, 1 \leq k \leq K_m\}$. Let $\boldsymbol{Z}^{(m)} = (z^{(m)}_{ik})$ be the $n \times K_m$ matrix with $z^{(m)}_{ik} = 1$ for $i \in \mathcal{G}^{(m)}_k$, and $z^{(m)}_{ik} = 0$ otherwise. In addition, let $\boldsymbol{\mu}^0 = (\mu^0_1, \ldots, \mu^0_{K_1})^\top$ and $\boldsymbol{\tau}^0 = (\tau^0_1, \ldots, \tau^0_{K_2})^\top$, where μ^0_k is the mean for group $\mathcal{G}^{(1)}_k$ and $\tau^0_{k'}$ is the precision for group $\mathcal{G}^{(2)}_{k'}$. When the underlying group structures $\mathcal{G}^{(m)}_1, \ldots, \mathcal{G}^{(m)}_{K_m}$, for $m = 1, 2$, are known, the oracle estimators for $\boldsymbol{\beta}$, $\boldsymbol{\theta}_{[\cdot 1]}$, and $\boldsymbol{\theta}_{[\cdot 2]}$ are defined as the maximizers of the log-likelihood function

$$((\widehat{\boldsymbol{\beta}}^{\text{or}})^\top, (\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 1]})^\top, (\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 2]})^\top) = \underset{\boldsymbol{\theta}_{[\cdot 1]} \in \mathcal{I}_{\mathcal{G}^{(1)}}, \boldsymbol{\theta}_{[\cdot 2]} \in \mathcal{I}_{\mathcal{G}^{(2)}}}{\operatorname{argmax}} \sum_{i=1}^n \log \phi(y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i; \theta_{i1}, \theta_{i2}). \quad (4.1)$$

Moreover, define the oracle estimators for $\boldsymbol{\beta}$, $\boldsymbol{\mu}$, and $\boldsymbol{\tau}$ as

$$\begin{aligned}&((\widehat{\boldsymbol{\beta}}^{\text{or}})^\top, (\widehat{\boldsymbol{\mu}}^{\text{or}})^\top, (\widehat{\boldsymbol{\tau}}^{\text{or}})^\top)\\ &= \operatorname{argmax} n^{-1} \sum_{i=1}^n \sum_{k'=1}^{K_2} z^{(2)}_{ik'} \left\{ \log \tau_{k'} - \tau_{k'} (y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i - \boldsymbol{\mu}^\top \boldsymbol{z}^{(1)}_{[i\cdot]})^2 \right\}.\end{aligned}$$

For notational simplicity, we define for any vector $\boldsymbol{u} = (u_1, \ldots, u_s)^\top \in \mathbb{R}^s$, $\|\boldsymbol{u}\|_\infty = \max_{1 \leq l \leq s} |u_l|$. For any $a_n, b_n \in \mathbb{R}^+$, we denote $a_n \gg b_n$, if $a_n^{-1} b_n = o(1)$. Let $p'(|t|, \lambda, \gamma)$ be the derivative of $p(|t|, \lambda, \gamma)$ with respect to $|t|$, that is, $p'(|t|, \lambda, \gamma) = \partial p(|t|, \lambda, \gamma)/\partial |t|$. Let $|\mathcal{G}^{(m)}_{\min}| = \min(|\mathcal{G}^{(m)}_1|, \ldots, |\mathcal{G}^{(m)}_{K_m}|)$, for $m = 1, 2$. To establish the asymptotic properties for the estimators, the following regular conditions are required:

(C1) There exist constants $0 < M, c_1 < +\infty$ such that $\|\boldsymbol{x}\|_\infty \leq M$ for any $\boldsymbol{x} \in \mathcal{X}$, and the smallest eigenvalues of $(\boldsymbol{X}, \boldsymbol{Z}^{(1)})^\top (\boldsymbol{X}, \boldsymbol{Z}^{(1)})$ are bounded by $c_1 |\mathcal{G}^{(1)}_{\min}|$.

(C2) There exist constants $0 < \tau_{\min} \leq \tau_{\max} < +\infty$ such that $\tau_{\min} \leq \tau^0_{k'} \leq \tau_{\max}$, for $k' = 1, \ldots, K_2$.

(C3) The mixing probability $\pi_{kk'}$ that a subject belongs to $\mathcal{G}^{(1)}_k \cap \mathcal{G}^{(2)}_{k'}$ satisfies $\min_{k,k'} \pi_{kk'} = O(\max_{k,k'} \pi_{kk'})$.

(C4) The penalty function $p(|t|, \lambda, \gamma)$ is symmetric with respect to t, and nondecreasing and concave in terms of $|t|$. There exists some constant $0 < a < +\infty$ such that $p(|t|, \lambda, \gamma)$ is a constant for all t with $|t| \geq a\lambda$, and $p(0, \lambda, \gamma) = 0$. The derivative $p'(|t|, \lambda, \gamma)$ exists and is continuous, except for a finite number of t, and $\lambda^{-1} p'(|t|, \lambda, \gamma) = 1$ as $|t| \to 0$.

By definition, the smallest eigenvalue of $(\boldsymbol{Z}^{(1)})^\top \boldsymbol{Z}^{(1)}$ is $|\mathcal{G}_{\min}^{(1)}|$, and it is reasonable to assume that the smallest eigenvalue of $\boldsymbol{X}^\top \boldsymbol{X}$ is bounded by Cn, for some constant $0 < C < +\infty$. Therefore, Condition (C1) assumes the smallest eigenvalue of $(\boldsymbol{X}, \boldsymbol{Z}^{(1)})^\top (\boldsymbol{X}, \boldsymbol{Z}^{(1)})$ is bounded by $c_1 |\mathcal{G}_{\min}^{(1)}|$, similarly to Ma and Huang (2017). Condition (C2) assumes that the true value of the precision is bounded, which is a common assumption in GMMs (Hao et al. (2018); Ren et al. (2022)). Condition (C3) requires that the groups in the mixture model are not too imbalanced, similarly to Ren et al. (2022). Condition (C4) is widely adopted in high-dimensional settings (Ma and Huang (2017)), and is satisfied by the hard and SCAD penalties.

Theorem 2. *Under Conditions* (C1) *to* (C3)*, assuming* $\max\{K_1, K_2\}\sqrt{p + K_1} K_2 = o(\sqrt{n(\log n)^{-1}})$*, it holds that*

$$\begin{aligned} &\|((\widehat{\boldsymbol{\beta}}^{\mathrm{or}})^\top, (\widehat{\boldsymbol{\theta}}_{[\cdot 1]}^{\mathrm{or}})^\top, (\widehat{\boldsymbol{\theta}}_{[\cdot 2]}^{\mathrm{or}})^\top) - ((\boldsymbol{\beta}^0)^\top, (\boldsymbol{\theta}_{[\cdot 1]}^0)^\top, (\boldsymbol{\theta}_{[\cdot 2]}^0)^\top)\|_\infty \\ &= O_p\left(\max(K_1, K_2) \sqrt{\frac{(p + K_1) K_2^2 \log n}{n}} + \max(K_1, K_2) \sqrt{\frac{K_2 \log n}{n}} \right). \end{aligned}$$

Theorem 2 states that the oracle estimators of $\boldsymbol{\beta}$, $\boldsymbol{\theta}_{[\cdot 1]}$, and $\boldsymbol{\theta}_{[\cdot 2]}$ converge to the true parameters; the proof is given in the Supplementary Material. It allows p, K_1, and K_2 to diverge with n, and requires $\max(K_1, K_2)\sqrt{p + K_1} K_2 = o(\sqrt{n(\log n)^{-1}})$. The result in Ma and Huang (2017) can be viewed as a special case of Theorem 2 by assuming $K_2 = 1$ (the negative log-likelihood reduces to the mean squared error) or that the heterogeneity precisions are already known. In these cases, we need only estimate the coefficients and the means. The required condition is then $K_1\sqrt{p + K_1} = o(\sqrt{n(\log n)^{-1}})$, and hence $K_1 = o(n^{1/3}(\log n)^{-1/3})$, which is the same as in Ma and Huang (2017). The bound in Theorem 2 is then $K_1\sqrt{(p + K_1) n^{-1} \log n}$, which is also the same as in Ma and Huang (2017, Remark 4). Moreover, Hao et al. (2018) consider high-dimensional Gaussian graphical mixture models, which assume that the mean and precision vectors have the same group structure and do not incorporate covariates. If we set $K_1 = K_2$ and $p = 0$, the bound in Theorem 2 is $\sqrt{K_1^5 n^{-1} \log n} + \sqrt{K_1^3 n^{-1} \log n}$, which is the same as in Hao et al. (2018) when applied to a one-dimensional GMM. In particular, when p, K_1, and K_2 are fixed, the error bound is $\sqrt{n^{-1} \log n}$.

As suggested by Hao et al. (2018), the first term of the bound in Theorem 2 represents the mean error, and the second term is the precision error. The structure of the means affects the estimation of the precisions, and vice versa.

Specifically, given K_1, the mean error is affected by the value of K_2, and given K_2, the value of K_1 also affects the precision error. Theorem 2 reveals the advantage of separately investigating the structures of the means and the precisions. We consider two special cases, $(K_1 = K, K_2 = 1)$ and $(K_1 = 1, K_2 = K)$. The error bound is $\sqrt{K^2(p+K)n^{-1}\log n} + \sqrt{K^2 n^{-1}\log n}$ in the first case, and $\sqrt{K^4(p+1)n^{-1}\log n} + \sqrt{K^3 n^{-1}\log n}$ for the latter. If we assume the mean and precision share the same group structure, as in the literature, that is, $K_1 = K_2 = K$, then the error bound is $\sqrt{K^4(p+K)n^{-1}\log n} + \sqrt{K^3 n^{-1}\log n}$ for both cases. As expected, identifying the structure of the parameters separately leads to estimates with smaller estimation errors. In addition, the estimation problem with the same mean but heterogeneous precisions $(K_1 = 1, K_2 = K)$ is more difficult than that with heterogeneous means and the same precision $(K_1 = K, K_2 = 1)$.

Remark 1. Let $\widetilde{\boldsymbol{X}} = (\boldsymbol{X}, \boldsymbol{Z}^{(1)})$. By the first-order optimality condition, we have $((\widehat{\boldsymbol{\beta}}^{\text{or}})^\top, (\widehat{\boldsymbol{\mu}}^{\text{or}})^\top)^\top = (\widetilde{\boldsymbol{X}}^\top \text{diag}(\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 2]})\widetilde{\boldsymbol{X}})^{-1}(\widetilde{\boldsymbol{X}}^\top \text{diag}(\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 2]})\boldsymbol{y})$, which can be viewed as a weighted least squares estimator for a heteroskedastic linear regression. Because $\boldsymbol{\theta}^0_{[\cdot 2]} = \boldsymbol{Z}^{(2)}\boldsymbol{\tau}^0$ is a smooth function and $\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 2]}$ is a consistent estimator of $\boldsymbol{\theta}^0_{[\cdot 2]}$, by Carroll (1982), we have that

$$((\widehat{\boldsymbol{\beta}}^{\text{or}})^\top, (\widehat{\boldsymbol{\mu}}^{\text{or}})^\top)^\top - ((\boldsymbol{\beta}^0)^\top, (\boldsymbol{\mu}^0)^\top)^\top \xrightarrow{d} N\Big(\mathbf{0}, \Big(\widetilde{\boldsymbol{X}}^\top \text{diag}(\boldsymbol{\theta}^0_{[\cdot 2]})\widetilde{\boldsymbol{X}}\Big)^{-1}\Big),$$

where $\xrightarrow{d}$ represents convergence in distribution. Therefore, as claimed in a large amount of literature (e.g., Shao (1989)), $((\widehat{\boldsymbol{\beta}}^{\text{or}})^\top, (\widehat{\boldsymbol{\mu}}^{\text{or}})^\top)^\top$ is more efficient than the ordinary least squares estimator in Ma and Huang (2017) when $K_2 \geq 2$.

Assuming $\max(K_1, K_2) \geq 2$, let $b_n = \min(\min_{i \neq j} |\theta^0_{i1} - \theta^0_{j1}|, \min_{i \neq j} |\theta^0_{i2} - \theta^0_{j2}|)$ be the minimal difference of the means or precisions between two groups. For simplicity, let $\psi_n = \max(K_1, K_2)\sqrt{n^{-1}\log n}\{\sqrt{(p+K_1)K_2^2} + \sqrt{K_2}\}$.

Theorem 3. *Under Conditions* (C1) *to* (C4)*, and assuming that the conditions in Theorem* 2 *hold,* $\max(K_1, K_2) \geq 2$*,* $b_n \geq a\max(\lambda_1, \lambda_2)$*, and* $\min(\lambda_1, \lambda_2) \gg \psi_n$*, with a defined in Condition* (C4)*, there exists a local minimizer* $(\widehat{\boldsymbol{\beta}}^{(\lambda,\gamma)}, \widehat{\boldsymbol{\theta}}^{(\lambda,\gamma)}_{[\cdot 1]}, \widehat{\boldsymbol{\theta}}^{(\lambda,\gamma)}_{[\cdot 2]})$ *of the objective function* $Q(\boldsymbol{\beta}, \boldsymbol{\Theta})$*, such that*

$$P\left\{((\widehat{\boldsymbol{\beta}}^{(\lambda,\gamma)})^\top, (\widehat{\boldsymbol{\theta}}^{(\lambda,\gamma)}_{[\cdot 1]})^\top, (\widehat{\boldsymbol{\theta}}^{(\lambda,\gamma)}_{[\cdot 2]})^\top) = ((\widehat{\boldsymbol{\beta}}^{\text{or}})^\top, (\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 1]})^\top, (\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 2]})^\top)\right\} \to 1$$

as $n \to \infty$.

Theorem 3 shows that the oracle estimator $((\widehat{\boldsymbol{\beta}}^{\text{or}})^\top, (\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 1]})^\top, (\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 2]})^\top)$ is a local minimizer of the objective function $Q(\boldsymbol{\beta}, \boldsymbol{\Theta})$ with probability approaching one as $n \to \infty$; the proof is given in the Supplementary Material. Combining Theorems 2 and 3, we conclude that there exists a local minimizer of the objective function converging to the true parameter.

4.2. Homogeneous model

When the true model is homogeneous, that is, $K_1 = K_2 = 1$, we show that the minimizer of the penalized objective function $Q(\boldsymbol{\beta}, \boldsymbol{\Theta})$ also has the oracle property. For $m = 1, 2$, let $\mathcal{I}_m$ be the subspace of $\mathbb{R}^n$, defined as $\mathcal{I}_m = \{\boldsymbol{\theta}_{[\cdot m]} \in \mathbb{R}^n : \theta_{1m} = \cdots = \theta_{nm}\}$. The oracle estimators under the homogeneous model are defined as $((\widehat{\boldsymbol{\beta}}^{\text{or}})^\top, (\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 1]})^\top, (\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 2]})^\top) = \operatorname{argmax}_{\boldsymbol{\theta}_{[\cdot 1]} \in \mathcal{I}_1, \boldsymbol{\theta}_{[\cdot 2]} \in \mathcal{I}_2} \sum_{i=1}^n \log \phi(y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i; \theta_{i1}, \theta_{i2})$.

Theorem 4. *Under Conditions* (C1) *and* (C4)*, assuming* $p = o(n(\log n)^{-1})$*, the following hold:*

(1) $\|((\widehat{\boldsymbol{\beta}}^{\text{or}})^\top, (\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 1]})^\top, (\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 2]})^\top) - ((\boldsymbol{\beta}^0)^\top, (\boldsymbol{\theta}^0_{[\cdot 1]})^\top, (\boldsymbol{\theta}^0_{[\cdot 2]})^\top)\|_\infty = O_p\left(\sqrt{(p+1)n^{-1}\log n} + \sqrt{n^{-1}\log n}\right)$.

(2) if $\lambda \gg \sqrt{(p+1)n^{-1}\log n} + \sqrt{n^{-1}\log n}$*, there exists a local minimizer* $(\widehat{\boldsymbol{\beta}}^{(\lambda,\gamma)}, \widehat{\boldsymbol{\theta}}^{(\lambda,\gamma)}_{[\cdot 1]}, \widehat{\boldsymbol{\theta}}^{(\lambda,\gamma)}_{[\cdot 2]})$ *of* $Q(\boldsymbol{\beta}, \boldsymbol{\Theta})$ *such that*

$$P\left\{((\widehat{\boldsymbol{\beta}}^{(\lambda,\gamma)})^\top, (\widehat{\boldsymbol{\theta}}^{(\lambda,\gamma)}_{[\cdot 1]})^\top, (\widehat{\boldsymbol{\theta}}^{(\lambda,\gamma)}_{[\cdot 2]})^\top) = ((\widehat{\boldsymbol{\beta}}^{\text{or}})^\top, (\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 1]})^\top, (\widehat{\boldsymbol{\theta}}^{\text{or}}_{[\cdot 2]})^\top)\right\} \to 1$$

as $n \to \infty$.

Theorem 4 shows that under a homogeneous model, there exists a local minimizer $((\widehat{\boldsymbol{\beta}}^{(\lambda,\gamma)})^\top, (\widehat{\boldsymbol{\theta}}^{(\lambda,\gamma)}_{[\cdot 1]})^\top, (\widehat{\boldsymbol{\theta}}^{(\lambda,\gamma)}_{[\cdot 2]})^\top)$ converging to $((\boldsymbol{\beta}^0)^\top, (\boldsymbol{\theta}^0_{[\cdot 1]})^\top, (\boldsymbol{\theta}^0_{[\cdot 2]})^\top)$; the proof is given in the Supplementary Material.

5. Simulations

We conduct extensive simulations to demonstrate the numerical performance of the proposed method for GMMs using the hard and SCAD penalties (abbreviated as Hard-GMM and SCAD-GMM, respectively), and compare the results with those of several existing methods. Specifically, we consider the following methods: (i) the method proposed by Ma and Huang (2017), which conducts subgroup analyses in a linear regression with different means using a concave fusion penalty (SubAna); (ii) the EM algorithm for finite mixtures in a linear regression, with the Gaussian error terms implemented using the R package "flexmix" (Grün and Leisch (2008)), in which regression coefficients are restricted to be equal over all components (FlexMix); and (iii) the method using model selection for GMMs without covariates proposed by Huang, Peng and Zhang (2017), which penalizes mixing probabilities, and implements a modified EM algorithm for the estimation (MS-GMM). As suggested by the authors, we use the SCAD penalty for SubAna and MS-GMM.

Based on preliminary experiments, we fix $\rho = 1.2$ for the hard penalty and $\rho = 0.5$ for the SCAD penalty. To apply the proposed method, one needs to select the tuning parameters λ_1, λ_2 (for both penalties) and γ_1, γ_2 (for the SCAD

penalty). Simulation results show that the numerical performance is not sensitive to the selection of γ_1, γ_2, and we set $\gamma_1 = \gamma_2 = 3.7$, following Fan and Li (2001). Although information criteria such as the AIC and BIC have been proposed for parameter tuning in the context of clustering, the model complexity penalty in these criteria is often ad hoc. Motivated by the work of She (2010) and She and Tran (2019), we set aside a separate validation data set to calculate the validation error (negative log-likelihood) and select the tuning parameters using the one standard error rule, which lead to the simplest model and a validation error that falls within one standard error of the minimum. The size of the validation set is fixed as 10 times that of the training set in our analysis. Moreover, we adopt an alternative search strategy to tune the parameters, which has been shown to be efficient (She (2009)). Specifically, we first search along the λ_2-path with λ_1 fixed at the minimum, median, and maximum in its candidate set. Then, we select the optimal value, denoted by $\lambda_2^{(\mathrm{opt})}$. Then, we search along the λ_1-path with λ_2 fixed at $\lambda_2^{(\mathrm{opt})}$. Accordingly, we search along four one-dimensional paths in total, including three λ_1-paths and one λ_2-path. Although this strategy does not cover the full parameter space, it is more computationally efficient than a grid search, and leads to satisfactory estimates.

Owing to the critical role of the initial values in Algorithm S1, we borrow ideas from prior works (Ma et al. (2020); Hu et al. (2021); Wang, Zhu and Zhang (2023)) and consider the optimization problem with a ridge fusion penalty,

$$\min \left\{ \sum_{i=1}^{n} (y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i - \theta_{i1})^2 + \sum_{m=1}^{2} \sum_{1 \le i < j \le n} \widetilde{\lambda}_m (\theta_{im} - \theta_{jm})^2 \right\}. \tag{5.1}$$

The parameters $\widetilde{\lambda}_1$ and $\widetilde{\lambda}_2$ are selected from the set $\{10^{-1}, 10^{-2}, \ldots, 10^{-6}\}$ using the same procedure described above. The objective function in (5.1) is differentiable, and thus we apply the limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm for bound constrained optimization to solve it, which is computationally fast. Denote the solutions as $\boldsymbol{\beta}^{(\mathrm{rid})}$, $\boldsymbol{\theta}_{[\cdot 1]}^{(\mathrm{rid})}$, and $\boldsymbol{\theta}_{[\cdot 2]}^{(\mathrm{rid})}$. Then, for $m = 1, 2$, we divide the subjects into $\lfloor n^{1/2} \rfloor$ subgroups by ranking $\boldsymbol{\theta}_{[\cdot m]}^{(\mathrm{rid})}$, where $\lfloor n^{1/2} \rfloor$ represents the maximum integer that does not exceed $n^{1/2}$. Denote these subgroups as $\widetilde{\mathcal{G}}_1^{(m)}, \ldots, \widetilde{\mathcal{G}}_{\lfloor n^{1/2} \rfloor}^{(m)}$. Lastly, we set the initial estimates $\boldsymbol{\beta}^{(0)} = \boldsymbol{\beta}^{(\mathrm{rid})}$ and $\boldsymbol{\theta}_{[\cdot m]}^{(0)} = (\theta_{1m}^{(0)}, \ldots, \theta_{nm}^{(0)})^\top$, where $\theta_{im}^{(0)}$ is equal to the median of $\{\theta_{jm}^{(\mathrm{rid})} : j \in \widetilde{\mathcal{G}}_k^{(m)}\}$, with $\widetilde{\mathcal{G}}_k^{(m)}$ the subgroup to which the ith subject belongs, for $k = 1, \ldots, \lfloor n^{1/2} \rfloor$.

Previous studies (Ma et al. (2020); Hu et al. (2021); Wang, Zhu and Zhang (2023)) have verified the validity of such an initialization procedure in various scenarios. As indicated by the following numerical studies, it can also provide a good start point for our ADMM algorithm.

To evaluate the performance of our method, we consider the identification of K_1 and K_2, as well as the estimation of $\boldsymbol{\beta}$, $\boldsymbol{\theta}_{[\cdot 1]}$, and $\boldsymbol{\theta}_{[\cdot 2]}$. Note that SubAna does

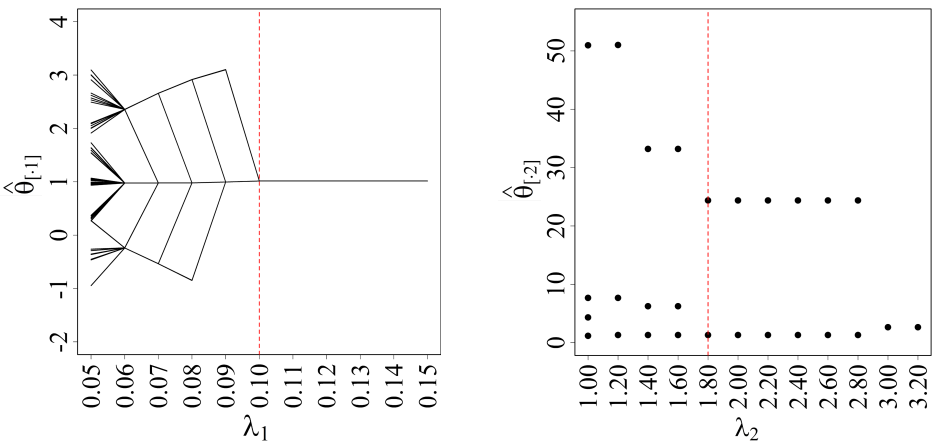
not estimate the precision parameters, thus there is no result for K_2 and $\boldsymbol{\theta}_{[\cdot 2]}$ for this method. To apply MS-GMM in our regression setting, we first obtain the ordinary least squares (OLS) estimator $\widehat{\boldsymbol{\beta}}^{\mathrm{ols}}$, and then implement MS-GMM on the pseudo errors $y_i - (\widehat{\boldsymbol{\beta}}^{\mathrm{ols}})^\top \boldsymbol{x}_i$. In addition, FlexMix and MS-GMM assume the means and precisions have the same structure, thus the estimated values of K_1 and K_2 are always the same. We investigate two scenarios of mixture models. Scenario 1 assumes a scale GMM with two components, which have the same mean, but different precisions. Scenario 2 adopts a much more complicated mixture model, with six distinct means and three distinct precisions. Set $\kappa^{\mathrm{abs}} = 0$ and $\kappa^{\mathrm{rel}} = 0.01$ in (3.10) for the termination criterion. Under each scenario, we conduct 100 replications.

Scenario 1. For $i = 1, \ldots, n$, ϵ_i is from a Gaussian distribution with density $\phi(\epsilon_i; \theta_{i1}, \theta_{i2})$, where $\theta_{i1} \equiv 1$ and θ_{i2} is generated from the distribution $P(\theta_{i2} = (0.2)^{-2}) = 1/3$ and $P(\theta_{i2} = (0.9)^{-2}) = 2/3$. Let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{i5})^\top$, where x_{ij} are independent and identically generated from the standard normal distribution. We simulate responses as $y_i = \boldsymbol{\beta}^\top \boldsymbol{x}_i + \epsilon_i$, with $\boldsymbol{\beta} = (3, 2, 0.5, -2, -3)^\top$, and set $n = 200$.

We set the maximum number of iterations in Algorithm S1 to 200. For the hard penalty, the candidate sets for λ_1 and λ_2 are $\{0.5, 0.6, \ldots, 1.5\}$ and $\{5, 5.2, \ldots, 7.2\}$, respectively; for the SCAD penalty, they are $\{0.05, 0.06, \ldots, 0.15\}$ and $\{1, 1.2, \ldots, 3.2\}$, respectively. Figure 1 shows the solution paths of $\widehat{\boldsymbol{\theta}}_{[\cdot 1]}$ and $\widehat{\boldsymbol{\theta}}_{[\cdot 2]}$ by SCAD-GMM for one simulated data set. The values of $\widehat{\boldsymbol{\theta}}_{[\cdot 1]}$ and $\widehat{\boldsymbol{\theta}}_{[\cdot 2]}$ show a similar pattern from divergence to convergence along the path. When λ_1 is small, the estimated means tend to be different, which should be close to the residuals $y_i - \widehat{\boldsymbol{\beta}}^\top \boldsymbol{x}_i$. As λ_1 increases, the estimated means converge to one point around the true value, one. The trend for the estimated precisions is similar. When λ_2 is small, there are more than two distinct values for the estimated precisions. They converge to the true values $(0.2)^{-2}$ and $(0.9)^{-2}$ as λ_2 increases, and finally converge to one point if λ_2 continues to increase.

Table 1 reports the average value and standard deviation (given as a subscript) of the bias and the square root of the mean squared error (RMSE) for the estimated values of $\boldsymbol{\beta}$ over 100 replications. For a vector $\boldsymbol{u} = (u_1, \ldots, u_s)^\top$ and its estimator $\widehat{\boldsymbol{u}} = (\widehat{u}_1, \ldots, \widehat{u}_s)^\top$, the bias of $\widehat{u}_j$ is defined as $|\widehat{u}_j - u_j|$, for $j = 1, \ldots, s$, and the RMSE of $\widehat{\boldsymbol{u}}$ is $\|\widehat{\boldsymbol{u}} - \boldsymbol{u}\|_2/\sqrt{s}$. We consider the four methods Hard-GMM, SCAD-GMM, SubAna, and FlexMix. The oracle and OLS estimators are also presented as references. Table 1 shows that Hard-GMM and SCAD-GMM perform similarly and deliver the results closest to those of the oracle estimators. The other two competitors, SubAna and FlexMix, are inferior to our method in terms of estimating $\boldsymbol{\beta}$.

Table 2 shows the median of $\widehat{K}_m$, the proportion of $\widehat{K}_m$ equal to the true value, and the RMSEs of $\widehat{\boldsymbol{\theta}}_{[\cdot 1]}$ and $\widehat{\boldsymbol{\theta}}_{[\cdot 2]}^{-1/2}$ (i.e., standard deviation), as well as



(a) $\widehat{\boldsymbol{\theta}}_{[\cdot 1]}$ against λ_1, with λ_2 fixed at 1.8 (b) $\widehat{\boldsymbol{\theta}}_{[\cdot 2]}$ against λ_2, with λ_1 fixed at 0.1

Figure 1. Solution paths for estimated values against tuning parameters by SCAD-GMM for one simulated data set under Scenario 1, where the dashed lines correspond to the optimal tuning parameters.

the computation time to train the model once on the whole training set with the specified tuning parameters. The results show that Hard-GMM and SCAD-GMM always correctly identify the numbers of components, and perform best in terms of estimating the parameters. SubAna also correctly identifies the number of components for the means, and ranks second in terms of estimating $\widehat{\boldsymbol{\theta}}_{[\cdot 1]}$. In comparison, the proposed method delivers more accurate and robust estimators than those of SubAna, because we consider heterogeneity among precisions. In addition, our method shows great advantages over the EM-based algorithms, FlexMix and MS-GMM, in terms of both determining the numbers of components and estimating the parameters. We next focus on the computation time, where MS-GMM runs fastest, followed by FlexMix, SubAna, Hard-GMM, and SCAD-GMM. In general, the EM-based algorithms run much faster than the ADMM-based algorithms. Compared to SubAna, our method spends twice as much time in estimating precisions. We further compare the performance of these methods in terms of clustering; detailed results and discussions are provided in the Supplementary Material, where Table S1 shows that our method performs best.

We now check the convergence of the ADMM algorithm, and present the results of Hard-GMM for illustration purposes. In Figure 2, we show the average curves over 20 runs of the primal relative residual $\|\boldsymbol{R}_p(\boldsymbol{\Theta}^{(t)}, \boldsymbol{\Delta}^{(t)})\|_F \times (\max\{\|\boldsymbol{E}\boldsymbol{\Theta}^{(t)}\|_F, \|\boldsymbol{\Delta}^{(t)}\|_F\})^{-1}$ and the dual relative residual $\|\boldsymbol{R}_d^{(t)}\|_F(\|\boldsymbol{E}^\top \boldsymbol{\nu}^{(t)}\|_F)^{-1}$ against the number of iterations. The results show that the ADMM algorithm converges steadily in this scenario, and the termination criterion is satisfied within 50 iterations, on average. The primal relative residual gets close to zero

Table 1. The average value and standard deviation of the bias and the square root of the mean squared error (RMSE) of $\widehat{\boldsymbol{\beta}}$ over 100 replications.

	Bias($\widehat{\beta}_1$)	Bias($\widehat{\beta}_2$)	Bias($\widehat{\beta}_3$)	Bias($\widehat{\beta}_4$)	Bias($\widehat{\beta}_5$)	RMSE($\widehat{\boldsymbol{\beta}}$)
			Scenario 1			
Oracle	$0.021_{0.017}$	$0.021_{0.015}$	$0.020_{0.016}$	$0.020_{0.016}$	$0.020_{0.014}$	$0.024_{0.008}$
Hard-GMM	$0.034_{0.027}$	$0.031_{0.022}$	$0.030_{0.026}$	$0.029_{0.022}$	$0.034_{0.028}$	$0.038_{0.014}$
SCAD-GMM	$0.034_{0.027}$	$0.031_{0.021}$	$0.030_{0.026}$	$0.030_{0.023}$	$0.033_{0.027}$	$0.038_{0.014}$
SubAna	$0.045_{0.034}$	$0.044_{0.031}$	$0.045_{0.030}$	$0.043_{0.034}$	$0.043_{0.036}$	$0.052_{0.018}$
FlexMix	$0.042_{0.032}$	$0.039_{0.031}$	$0.044_{0.033}$	$0.047_{0.034}$	$0.043_{0.036}$	$0.051_{0.018}$
OLS	$0.045_{0.034}$	$0.044_{0.031}$	$0.045_{0.030}$	$0.043_{0.034}$	$0.043_{0.036}$	$0.052_{0.018}$
			Scenario 2			
Oracle	$0.015_{0.011}$	$0.014_{0.010}$	$0.014_{0.010}$	$0.016_{0.011}$	$0.015_{0.010}$	$0.017_{0.005}$
Hard-GMM	$0.110_{0.257}$	$0.073_{0.226}$	$0.116_{0.207}$	$0.085_{0.208}$	$0.092_{0.217}$	$0.111_{0.224}$
SCAD-GMM	$0.112_{0.271}$	$0.077_{0.242}$	$0.115_{0.201}$	$0.080_{0.200}$	$0.091_{0.212}$	$0.110_{0.225}$
SubAna	$0.264_{0.382}$	$0.209_{0.282}$	$0.238_{0.356}$	$0.241_{0.340}$	$0.229_{0.359}$	$0.280_{0.311}$
FlexMix	$0.325_{0.412}$	$0.310_{0.404}$	$0.336_{0.513}$	$0.399_{0.539}$	$0.354_{0.387}$	$0.420_{0.388}$
OLS	$0.674_{0.482}$	$0.570_{0.421}$	$0.628_{0.481}$	$0.604_{0.472}$	$0.646_{0.472}$	$0.743_{0.234}$

Oracle: the oracle estimators defined in (4.1); Hard-GMM: the proposed method under the hard penalty; SCAD-GMM: the proposed method under the SCAD penalty; SubAna: subgroup analysis proposed by Ma and Huang (2017); FlexMix: the EM algorithm for finite mixtures of linear regression developed by Grün and Leisch (2008); OLS: the ordinary least squares estimators.

after about 10 iterations, whereas the dual relative residual decreases relatively slowly. We also show the relative residuals after 200 iterations in Figure S1 in the Supplementary Material, which verify that the dual relative residual continues to decrease, albeit slowly, as the number of iterations increases. Figure 2 also shows the average curves of the objective value $Q(\boldsymbol{\beta}^{(t)}, \boldsymbol{\Theta}^{(t)})$, which converges fast with iterations. Furthermore, the RMSEs of $\boldsymbol{\beta}^{(t)}$, $\boldsymbol{\theta}^{(t)}_{[\cdot 1]}$, and $(\boldsymbol{\theta}^{(t)}_{[\cdot 2]})^{-1/2}$ in Figure 2 show that the ADMM algorithm converges to a stationary point after a number of iterations. We study the convergence of the RMSEs further for parameters under different sample sizes; see Figure 3, which shows that the obtained solutions converge faster, and eventually converge to smaller RMSEs with larger sample sizes.

We finally investigate the sensitivity of SCAD-GMM to γ_1 and γ_2. We set $\gamma_1 = \gamma_2 = \gamma$ to $3.1, 3.3, \ldots, 4.9$. Figure 4 shows the RMSEs of the parameters against the value of γ over 100 repetitions, indicating that the estimation of the parameters is not sensitive to the value of γ. Figure S2 in the Supplementary Material shows the primal and dual relative residuals for $\gamma = 3.1, 3.7, 4.9$. The ADMM algorithm converges for all three values of γ, and the convergence rate is slower for $\gamma = 4.9$.

Table 2. The median (Med) of $\widehat{K}_1$ and $\widehat{K_2}$, the proportion (Prop) of $\widehat{K}_1$ and $\widehat{K_2}$ equal to the true values, the average value and standard deviation of the square root of the mean squared error (RMSE) of $\widehat{\boldsymbol{\theta}}_{[\cdot 1]}$ and $\widehat{\boldsymbol{\theta}}_{[\cdot 2]}^{-1/2}$, and the average computation time in seconds over 100 repetitions.

	$\widehat{K}_1$		$\widehat{K}_2$		RMSE		Time
	Med	Prop	Med	Prop	$\widehat{\boldsymbol{\theta}}_{[\cdot 1]}$	$(\widehat{\boldsymbol{\theta}}_{[\cdot 2]})^{-1/2}$	
				Scenario 1			
Oracle	–	–	–	–	$0.019_{0.014}$	$0.044_{0.076}$	–
Hard-GMM	1	1	2	1	$0.030_{0.023}$	$0.212_{0.082}$	7.61
SCAD-GMM	1	1	2	1	$0.030_{0.023}$	$0.219_{0.080}$	8.22
SubAna	1	1	–	–	$0.039_{0.032}$	–	3.83
FlexMix	2	0.76	2	0.76	$0.064_{0.059}$	$0.277_{0.081}$	1.10
MS-GMM	2	0.74	2	0.74	$0.162_{0.206}$	$0.259_{0.126}$	0.08
				Scenario 2			
Oracle	–	–	–	–	$0.075_{0.027}$	$0.030_{0.015}$	–
Hard-GMM	6	0.69	3	0.95	$0.621_{0.779}$	$0.386_{0.184}$	38.98
SCAD-GMM	6	0.69	3	0.93	$0.656_{0.788}$	$0.383_{0.171}$	40.78
SubAna	6	0.68	–	–	$1.147_{1.150}$	–	24.26
FlexMix	6	0.54	6	0.54	$2.448_{2.103}$	$2.371_{2.056}$	1.55
MS-GMM	6	0.57	6	0.57	$1.998_{2.063}$	$2.233_{2.023}$	0.11

Oracle: the oracle estimators defined in (4.1); Hard-GMM: the proposed method under the hard penalty; SCAD-GMM: the proposed method under the SCAD penalty; SubAna: subgroup analysis proposed by Ma and Huang (2017); FlexMix: the EM algorithm for finite mixtures of linear regression developed by Grün and Leisch (2008); MS-GMM: model selection for GMMs proposed by Huang, Peng and Zhang (2017), applied to $y_i - (\widehat{\boldsymbol{\beta}}^{\text{ols}})^\top \boldsymbol{x}_i$.

Scenario 2. We simulate data from a more complicated mixture model. For $i = 1, \ldots, n$, ϵ_i is from a Gaussian distribution with density $\phi(\epsilon_i; \theta_{i1}, \theta_{i2})$, where θ_{i1} is generated from $\{-20, -12, -4, 4, 12, 20\}$ with equal probabilities and

$$\theta_{i2} = \begin{cases} (0.2)^{-2}, & \text{if } \theta_{i1} = -20 \text{ or } -12, \\ (0.4)^{-2}, & \text{if } \theta_{i1} = -4, \\ (0.7)^{-2}, & \text{otherwise.} \end{cases}$$

Let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{i5})^\top$, where x_{ij} are independent and identically generated from the standard normal distribution. We simulate responses as $y_i = \boldsymbol{\beta}^\top \boldsymbol{x}_i + \epsilon_i$, with $\boldsymbol{\beta} = (3, 2, 0.5, -2, -3)^\top$, and set $n = 300$.

We set the maximum number of iterations in Algorithm S1 to 500 for this complicated scenario. The estimated results are shown in Tables 1 and 2. In this scenario, the proposed method demonstrates significant advantages in terms of structure identification and parameter estimation. Although SubAna performs similarly to our method in terms of identifying the number of components for the

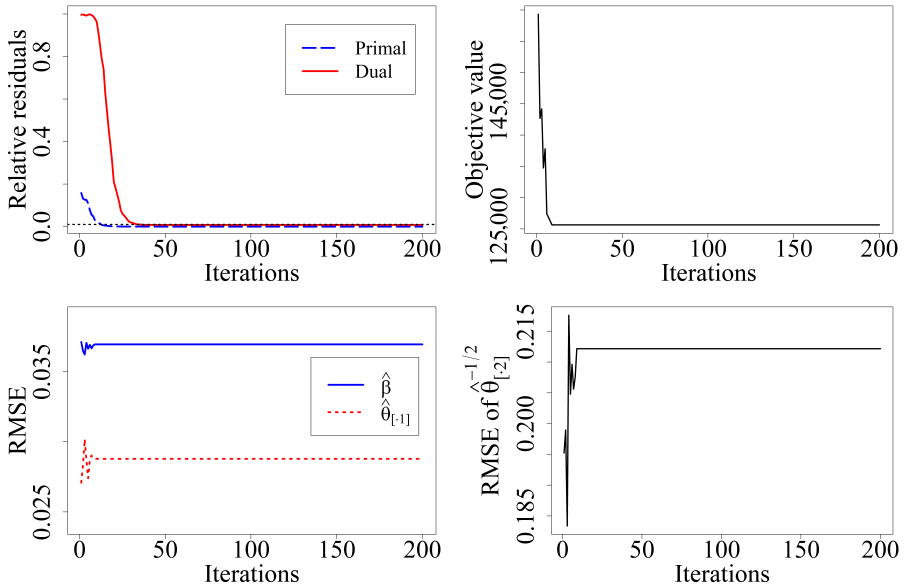


Figure 2. Average curves for the primal and dual relative residuals, the objective value, and the RMSEs of the estimated parameters against the number of iterations by Hard-GMM over 20 repetitions under Scenario 1.

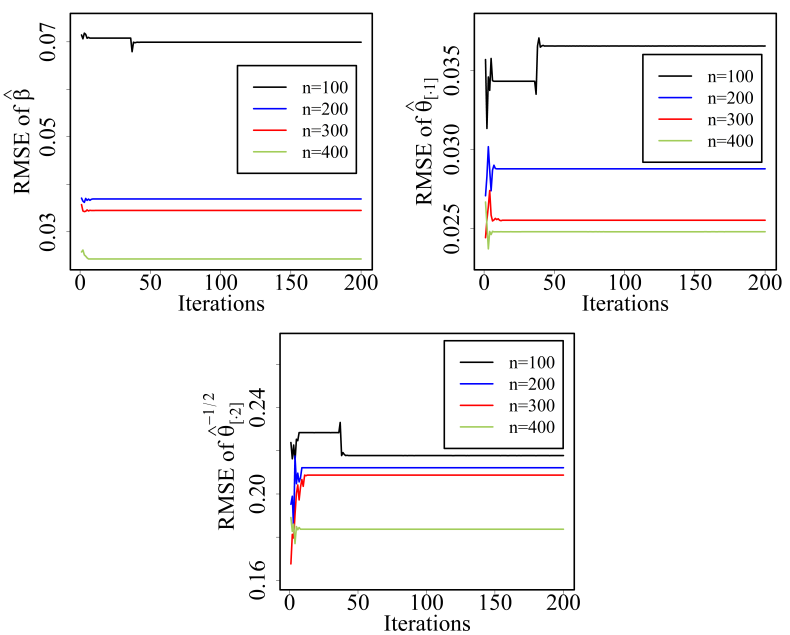


Figure 3. Average curves for the RMSEs of the estimated parameters against the number of iterations with different sample sizes by Hard-GMM over 20 repetitions under Scenario 1.

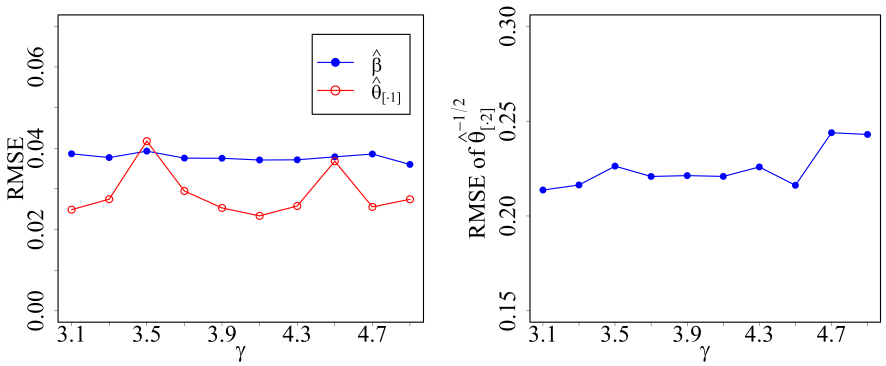


Figure 4. Average curves for the RMSEs of the estimated parameters against the value of γ by SCAD-GMM over 100 repetitions under Scenario 1.

means, it does not consider heterogeneity among precisions. On the other hand, our method achieves high accuracy in terms of identifying the structure of the precisions. As a result, the proposed method delivers more accurate and robust estimations of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}_{[\cdot 1]}$ than those of SubAna. In addition, MS-GMM performs poorly in this scenario, because it is applied to the pseudo residuals $y_i - (\widehat{\boldsymbol{\beta}}^{\text{ols}})^\top \boldsymbol{x}_i$, where the OLS estimator is biased because of heterogeneity. Although FlexMix delivers reasonable results in terms of estimating $\boldsymbol{\beta}$, it also performs poorly in terms of estimating $\boldsymbol{\theta}_{[\cdot 1]}$ and $\boldsymbol{\theta}_{[\cdot 2]}$. One possible reason is that the EM algorithm is sensitive to the initial points in this complicated scenario. Therefore, we adopt the suggested strategy of Grün and Leisch (2008) to first make several runs of the stochastic EM algorithm with different random initializations, and then start the EM using the best solution obtained. Nevertheless, it still performs unsatisfactorily. For computation, the ADMM-based methods, Hard-GMM, SCAD-GMM, and SubAna, run much slower than MS-GMM and FlexMix for this larger data set, because the latter two are less affected by the sample size. The improvement in terms of estimation accuracy of our method is achieved at the cost of computation. We also present clustering results in Table S1 in the Supplementary Material, which show the superiority of our method. To check the convergence of the ADMM algorithm, Figure 5 shows the average results for SCAD-GMM over 20 runs. As shown, although the optimization problem becomes difficult in this complicated scenario, the relative residuals still satisfy the termination criterion, and the obtained solutions converge to stationary points within 500 iterations, on average.

6. Real-Data Example

For illustration, we apply the proposed method to Cleveland Heart Disease data (`https://archive.ics.uci.edu/ml/datasets/Heart+Disease`) from the UCI repository. The selection of the tuning parameters λ_1 and λ_2 is the same as

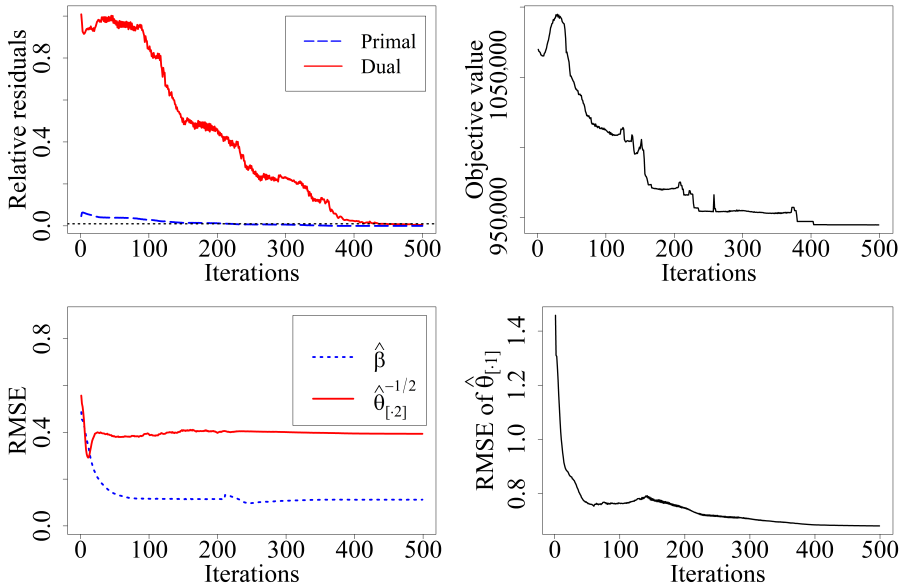


Figure 5. Average curves for the primal and dual relative residuals, the objective value, and the RMSEs of the estimated parameters against the number of iterations by SCAD-GMM over 20 repetitions under Scenario 2.

in Section 5, except that the validation error is calculated using five-fold cross-validation.

The data contain 303 individuals and 14 variables, where the first 13 variables are clinical measurements, and the last one indicates whether an individual suffers from heart disease. After deleting observations with missing values, there remain 297 observations. The variable "thalach", which represents the maximum heart rate achieved, is related to cardiac mortality (Lauer et al. (1999)). Our analysis aims to identify group structures when predicting "thalach." We are interested in six covariates: age, sex, resting blood pressure, serum cholesterol, fasting blood sugar, and a resting electrocardiographic (ECG) result, which is a categorical variable with three levels (0=normal, 1=having ST–T wave abnormality, 2=showing probable or definite left ventricular hypertrophy by Estes' criteria), and thus is converted to two dummy variables. We use six additional variables to check heart problems, namely chest pain type, exercise induced angina, ST depression induced by exercise relative to rest, slope of the peak exercise ST segment, number of major vessels colored by fluoroscopy, and heart status. Similar to the procedure in Ma and Huang (2017), we first regress "thalach" on these six additional variables using a linear model, and then use the fitted value of "thalach" as the pseudo response variable, denoted by y.

We regress y on the original set of seven covariates using the ordinary least squares method. Figure 6 shows the KDE of $y_i - (\widehat{\boldsymbol{\beta}}^{\text{ols}})^\top \boldsymbol{x_i}$ with the bandwidth

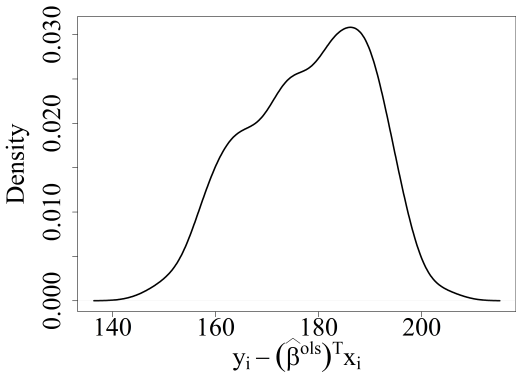


Figure 6. The kernel density estimate of $y_i - (\widehat{\boldsymbol{\beta}}^{\text{ols}})^\top \boldsymbol{x_i}$'s in the Cleveland Heart Disease data.

chosen using the method of Sheather and Jones (1991), which exhibits multiple modes in the distribution, and thus indicates the existence of heterogeneity. We apply Hard-GMM, SCAD-GMM, SubAna, and FlexMix to these data. The estimated values of K_1, K_2, $\boldsymbol{\mu}$, and $(\boldsymbol{\tau})^{-1/2}$, where the latter two are the distinct values of the means and the standard deviations, respectively, are presented in Table 3. We also show the sizes of the subgroups of means and precisions, denoted by $|\widehat{\mathcal{G}}^{(1)}|$ and $|\widehat{\mathcal{G}}^{(2)}|$, respectively. Our methods Hard-GMM and SCAD-GMM identify two subgroups for both the means and the precisions. Table 4 shows the estimates of $\boldsymbol{\beta}$ by various methods. We also report the standard errors and p-values of the significance tests, obtained by refitting a weighted linear model, incorporating the indicator vector $\boldsymbol{z}^{(1)}_{[i\cdot]}$ as covariates, and using the estimated precisions as weights. The result demonstrates that by recovering the group structure of the data, we can identify variables that do have effects on the response. For example, ECG (hypertrophy) is insignificant under the OLS method, but becomes significant under the heterogeneous methods. Moreover, the adjusted R-square of the OLS method is 0.103, indicating poor model fitting. After considering heterogeneity, the adjusted R-square is 0.782, 0.778, 0.745, and 0.746 for Hard-GMM, SCAD-GMM, SubAna, and FlexMix, respectively. By taking into account the group structure, the model fitting can be greatly improved, and the proposed method performs best.

7. Discussion

We propose a penalized approach enabling Gaussian mixture linear models to handle heterogeneity. The concave hard and SCAD penalties are adopted to shrink the pairwise differences of the means and precisions, respectively. By increasing the value of the tuning parameter for the penalty term, our method automatically clusters and merges similar instances. The theoretical properties show that under mild conditions, there exists a local minimizer of

Table 3. Estimated values of K_1, K_2, $\boldsymbol{\mu}$, and $(\boldsymbol{\tau})^{-1/2}$, and the sizes of the subgroups in means and precisions, denoted by $|\widehat{\mathcal{G}}^{(1)}|$ and $|\widehat{\mathcal{G}}^{(2)}|$, respectively, for the Cleveland Heart Disease data.

	$\widehat{K}_1$	$\widehat{K}_2$	$\widehat{\boldsymbol{\mu}}$	$\lvert\widehat{\mathcal{G}}^{(1)}\rvert$	$(\widehat{\boldsymbol{\tau}})^{-1/2}$	$\lvert\widehat{\mathcal{G}}^{(2)}\rvert$
Hard-GMM	2	2	(193.21, 167.69)	(183, 114)	(7.65, 4.22)	(223, 74)
SCAD-GMM	2	2	(198.56, 174.60)	(183, 114)	(8.80, 4.34)	(150, 147)
SubAna	2	–	(193.26, 177.97)	(183, 114)	–	–
FlexMix	2	2	(183.36, 164.84)	(151, 146)	(8.75, 5.92)	(151, 146)

Table 4. Estimated values (Est) of the coefficients with the standard errors (S.E.) and p-values (p) for the Cleveland Heart Disease data.

Model		Age	Sex	Blood Press.	Cholesterol	Sugar	ECG (wave)	ECG (hypertrophy)
OLS	Est	-0.333	-4.617	-0.026	-0.008	-0.094	-14.076	-2.700
	S.E.	0.083	1.531	0.042	0.014	2.023	6.140	1.441
	p	<0.001	0.003	0.531	0.553	0.963	0.023	0.062
Hard-GMM	Est	-0.283	-3.201	-0.022	-0.002	1.924	-11.831	-3.676
	S.E.	0.042	0.762	0.021	0.007	1.008	3.055	0.717
	p	<0.001	<0.001	0.300	0.753	0.057	<0.001	<0.001
SCAD-GMM	Est	-0.280	-3.232	-0.024	-0.001	1.916	-11.597	-3.824
	S.E.	0.040	0.746	0.021	0.007	0.981	2.974	0.699
	p	<0.001	<0.001	0.245	0.860	0.052	<0.001	<0.001
SubAna	Est	-0.286	-3.095	-0.027	-0.001	1.760	-11.564	-3.599
	S.E.	0.044	0.819	0.022	0.008	1.081	3.277	0.770
	p	<0.001	<0.001	0.228	0.876	0.105	<0.001	<0.001
FlexMix	Est	-0.250	-1.294	0.030	0.002	0.989	-7.301	-2.944
	S.E.	0.064	1.217	0.033	0.011	1.381	3.952	1.037
	p	<0.001	0.287	0.351	0.883	0.474	0.065	0.005

the objective function that converges to the true parameters. Our method can separately identify the structures of different types of parameters and calculate pooled estimators, which are more efficient. Simulation results corroborate the advantages of the proposed method in terms of estimation accuracy.

Our method has several limitations. Although the initialization approach in Section 5 performs well in numerical studies, it lacks theoretical support. As indicated by the analysis in Section 3, the computational complexity of the proposed method increases significantly with the sample size. In Section 4, we establish theoretical properties under the condition that $p \ll n/\log n$. In the high-dimensional setting, an additional penalty term needs to be imposed on the regression parameter $\boldsymbol{\beta}$ to enforce sparsity, that is, $\sum_{j=1}^{p} p(|\beta_j|, \lambda, \gamma)$. The proposed ADMM algorithm is still applicable, with minor modifications, where the updating equation (3.3) of $\boldsymbol{\beta}$ should be re-derived based on a penalized likelihood. However, extra effort is needed to develop theoretical properties of the estimators in the high-dimensional setting. Existing results (Yang, Yan and Huang (2019)) may provide ideas for solving this technical problem. Recently,

She, Shen and Zhang (2022) proposed a novel clustered reduced-rank learning (CRL) framework that imposes two joint matrix regularizations to automatically group the features in supervised multivariate learning. They prove that the CRL rate always beats the rate using the pairwise-difference penalization, and claim that the CRL method is computationally more efficient. Owing to its superiority, it is of interest, though challenging, to extend the CRL framework to GMMs.

Supplementary Material

The online Supplementary Material contains the ADMM algorithm, detailed derivations from Section 3, proofs of the theorems in Section 4, and additional simulation and application results.

Acknowledgments

We thank the associate editor and the two referees for their careful reviews and insightful suggestions. Fan's research is supported by National Natural Science Foundation of China (No. 12301329) and Beijing Institute of Technology Research Fund Program for Young Scholars. Yin's research is supported by the Research Grants Council of Hong Kong (17308321).

References

Antoniadis, A. (1997). Wavelets in statistics: A review. *Journal of the Italian Statistical Society* **6**, 97–130.

Bartolucci, F. and Scaccia, L. (2005). The use of mixtures for dealing with non-normal regression errors. *Computational Statistics & Data Analysis* **48**, 821–834.

Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* **3**, 1–122.

Carroll, R. J. (1982). Adapting for heteroscedasticity in linear models. *The Annals of Statistics* **10**, 1224–1233.

Chen, J. and Khalili, A. (2009). Order selection in finite mixture models with a nonsmooth penalty. *Journal of the American Statistical Association* **104**, 187–196.

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association* **90**, 577–588.

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

Grün, B. and Leisch, F. (2008). FlexMix version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software* **28**, 1–35.

Hao, B., Sun, W. W., Liu, Y. and Cheng, G. (2018). Simultaneous clustering and estimation of heterogeneous graphical models. *Journal of Machine Learning Research* **18**, 1–58.

Hu, X., Huang, J., Liu, L., Sun, D. and Zhao, X. (2021). Subgroup analysis in the heterogeneous Cox model. *Statistics in Medicine* **40**, 739–757.

Huang, T., Peng, H. and Zhang, K. (2017). Model selection for Gaussian mixture models. *Statistica Sinica* **27**, 147–169.

Lauer, M. S., Francis, G. S., Okin, P. M., Pashkow, F. J., Snader, C. E. and Marwick, T. H. (1999). Impaired chronotropic response to exercise stress testing as a predictor of mortality. *JAMA* **281**, 524–529.

Leroux, B. G. (1992). Consistent estimation of a mixing distribution. *The Annals of Statistics* **20**, 1350–1360.

Li, L., Damarla, S. K., Wang, Y. and Huang, B. (2021). A Gaussian mixture model based virtual sample generation approach for small datasets in industrial processes. *Information Sciences* **581**, 262–277.

Liu, G., Long, W., Yang, B. and Cai, Z. (2022). Semiparametric estimation and model selection for conditional mixture copula models. *Scandinavian Journal of Statistics* **49**, 287–330.

Ma, S. and Huang, J. (2017). A concave pairwise fusion approach to subgroup analysis. *Journal of the American Statistical Association* **112**, 410–423.

Ma, S., Huang, J., Zhang, Z. and Liu, M. (2020). Exploration of heterogeneous treatment effects via concave fusion. *The International Journal of Biostatistics* **16**, 20180026.

Ren, M., Zhang, S., Zhang, Q. and Ma, S. (2022). Gaussian graphical model-based heterogeneity analysis via penalized fusion. *Biometrics* **78**, 524–535.

Rossi, P. (2014). *Bayesian Non- and Semi-parametric Methods and Applications*. Princeton University Press.

Shao, J. (1989). Asymptotic distribution of the weighted least squares estimator. *Annals of the Institute of Statistical Mathematics* **41**, 365–382.

She, Y. (2009). Thresholding-based iterative selection procedures for model selection and shrinkage. *Electronic Journal of Statistics* **3**, 384–415.

She, Y. (2010). Sparse regression with exact clustering. *Electronic Journal of Statistics* **4**, 1055–1096.

She, Y., Shen, J. and Zhang, C. (2022). Supervised multivariate learning with simultaneous feature auto-grouping and dimension reduction. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **84**, 912–932.

She, Y. and Tran, H. (2019). On cross-validation for sparse reduced rank regression. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **81**, 145–161.

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **53**, 683–690.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **58**, 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005). Sparsity and smoothness via the fused Lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **67**, 91–108.

Wang, X., Zhu, Z. and Zhang, H. H. (2023). Spatial heterogeneity automatic detection and estimation. *Computational Statistics & Data Analysis* **180**, 107667.

Wang, Y., Yin, W. and Zeng, J. (2019). Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing* **78**, 29–63.

Yang, X., Yan, X. and Huang, J. (2019). High-dimensional integrative analysis with homogeneity and sparsity recovery. *Journal of Multivariate Analysis* **174**, 104529.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association* **101**, 1418–1429.

Yiwei Fan
School of Mathematics and Statistics, Beijing Institute of Technology, Beijing 100081, China.
E-mail: fanyiwei@live.cn
Guosheng Yin
Department of Statistics & Actuarial Science, The University of Hong Kong, Pokfulam Road, Hong Kong.
E-mail: gyin@hku.hk

(Received March 2022; accepted March 2023)