Biometrics 000, 000–000 DOI: 000
000 0000

Leveraging Independence in High-dimensional Mixed Linear Regression

Ning Wang[‡&], Kai Deng[†&], Qing Mai[†] and Xin Zhang[†*]

[‡]Beijing Normal University and [†]Florida State University

[&] equally contributed co-first authors

[*]*email:* henry@stat.fsu.edu

Summary: We address the challenge of estimating regression coefficients and selecting relevant predictors in the context of mixed linear regression (MLR) in high dimensions, where the number of predictors greatly exceeds the sample size. Recent advancements in this field have centered on incorporating sparsity-inducing penalties into the expectation-maximization (EM) algorithm, which seeks to maximize the conditional likelihood of the response given the predictors. However, existing procedures often treat predictors as fixed or overlook their inherent variability. In this paper, we leverage the independence between the predictor and the latent indicator variable of mixtures to facilitate efficient computation and also achieve synergistic variable selection across all mixture components. We establish the non-asymptotic convergence rate of the proposed Fast Group-penalized EM (FGEM) estimator to the true regression parameters. The effectiveness of our method is demonstrated through extensive simulations and an application to the Cancer Cell Line Encyclopedia (CCLE) dataset for the prediction of anticancer drug sensitivity.
Key words: EM algorithm; Group lasso; Finite mixture model; Latent variable model.

This paper has been submitted for consideration for publication in *Biometrics*

1. Introduction

Mixed linear regression (MLR), also known as finite mixture of linear regressions, models the heterogeneous linear relationship between a response variable and multiple predictors. It is more flexible and adaptive than usual linear regression and is widely accepted in various applications including biomedical research, environmental science, economics, and engineering (Qin et al., 2015; McLachlan et al., 2019; Li et al., 2019). As a special case of finite mixture models (McLachlan et al., 2019), the mixed linear regression model can be efficiently estimated through the expectation-maximization (EM) algorithm. Algorithms and variable selection approaches have been proposed in recent years to extend MLR to high-dimensional settings. Khalili and Chen (2007) considered penalized MLR for variable selection and established the asymptotic results with a fixed number of predictors. Städler et al. (2010) proposed an alternative parameterization for high-dimensional mixed linear regression, where more efficient computation and non-asymptotic convergence results were obtained for the lasso-type regularized estimator. Wang and Paschalidis (2019) introduced a mixed integer programming formulation and proved the convergence under the special case of a single component. Wang et al. (2015) generalize the standard EM algorithm by attaching a truncation step to introduce sparsity in regression coefficients. More recently, Zhang et al. (2020) considered both estimation and inference for high-dimensional mixed linear regression and developed a lasso penalized EM algorithm based on sample-splitting. Chaganty and Liang (2013); Anandkumar et al. (2014) developed tensor power methods as an alternative to EM algorithms. We adopted the tensor power method as a computationally efficient initialization for our FGEM method.

In recent years, the statistical convergence rate of the EM algorithm for mixed linear regression has been rigorously studied (Yi et al., 2014; Chen et al., 2014; Xu et al., 2016; Balakrishnan et al., 2017; Kwon and Caramanis, 2020; Kwon et al., 2021). These results are

established in the low-dimensional settings where the sample size n is much larger than the dimension of the predictor p and p is often assumed to be fixed. More recent theoretical studies on high-dimensional sparse MLR estimators often require strong model assumptions. For example, Wang et al. (2015) require two symmetric components with equal mixture weights to establish a convergence rate; similar assumptions are also considered in (Yi and Caramanis, 2015), where the theoretical results require sample-splitting. Sample-splitting penalized EM algorithms such as ones developed in Zhang et al. (2020); Klusowski et al. (2019) divide the full data set into T independent batches and use a fresh batch of data in every EM algorithm iteration, $t = 1, \ldots, T$. Therefore, the effective sample size is n/T, which could cause a substantial loss in the estimation efficiency of the regression parameters.

Most methods for penalized MLR impose separate regularization for different mixture components. In this paper, we consider a joint groupwise penalization across all the mixtures to achieve more interpretable and efficient variable selection. In the related pursuit, Hui et al. (2015) approximate the group lasso penalty (Yuan and Lin, 2006) with a quadratic function that is separable in terms of mixture components within each EM iteration. However, the accuracy of this penalty approximation remains unclear as the theoretical studies are challenging. To the best of our knowledge, Wang et al. (2023) is the first theoretical result for high-dimensional estimation with group-lasso penalized EM algorithm without sample-splitting or assuming known mixing proportion or design covariance matrix. Unfortunately, as we discuss in Section 2, this group-lasso EM algorithm poses a substantial computational challenge in estimation, making it infeasible for larger datasets. In this paper, we propose a fast group-penalized EM (FGEM) algorithm that modifies the M-step by utilizing the independence information between the latent variable and predictor and imposing the group lasso penalty on mixture coefficients. The proposed FGEM algorithm achieves both interpretable group variable selection and efficient computation simultaneously and is a practical alter-

native to its theoretical-motivated counterpart in Wang et al. (2023). Furthermore, we also establish theoretical results without sample splitting and under general model assumptions that mixture weights and covariance of predictors are both unknown.

The rest of this paper is organized as follows. Section 2 reviews the EM algorithms for estimating mixed linear regression and introduces our new formulation and algorithm for groupwise variable selection and fast estimation. Theoretical guarantees are presented in Section 3. Numerical studies are presented in Section 4. Finally, Section 5 concludes the paper with a brief discussion. Additional numerical results and technical proofs are provided in the Supplementary Materials.

2. Method

2.1 *Background on the MLR model assumption and the EM algorithm*

We consider the regression problem of a univariate response $Y \in \mathbb{R}$ and a multivariate predictor $\mathbf{X} \in \mathbb{R}^p$. The mixed linear regression (MLR) model assumes a mixture of K subpopulations, with probability $\omega_k > 0$, $k = 1, \ldots, K$, so that $Y = \boldsymbol{\beta}_k^T \mathbf{X} + \epsilon$. The error $\epsilon \sim N(0, \sigma^2)$ is independent of $\mathbf{X}$; $\boldsymbol{\beta}_k \in \mathbb{R}^p$ is the regression coefficient captures the linear dependence between Y and $\mathbf{X}$; the mixture probability satisfies $\sum_{k=1}^K \omega_k = 1$.

By introducing a latent categorical variable $W \in \{1, \ldots, K\}$, the MLR can be rewritten as,

$$Y \mid (\mathbf{X}, W = k) \sim N(\boldsymbol{\beta}_k^T \mathbf{X}, \sigma^2), \quad \Pr(W = k) = \omega_k, \tag{1}$$

where $\mathbf{X}$ is assumed to be independent of W with $\mathrm{E}(\mathbf{X}) = 0$ and $\mathrm{var}(\mathbf{X}) = \boldsymbol{\Sigma} > 0$. As a convention in the high-dimensional mixed linear regression literature (e.g., Yi and Caramanis, 2015; Zhang et al., 2020; Wang et al., 2023), the imposed assumption $\mathrm{E}(\mathbf{X}) = 0$ is for ease of presentation. The implication of this additional assumption is that $\mathrm{E}(Y \mid \mathbf{X} = 0, W = k) = 0$ and thus the center of each mixture does not change over k. Therefore, $\mathrm{E}(\mathbf{X}) = 0$ can be

satisfied in finite sample by centering the data. Alternatively, we can use $(1, \mathbf{X}^T)^T$ to replace the predictor in the algorithm to capture different intercepts in each mixture and relax this additional assumption. The maximum likelihood estimator for (1) has no explicit formula and the EM algorithm and its variants are routinely employed to maximize the likelihood. In what follows, we first discuss the estimation of parameters in MLR (1) using the standard EM algorithm (Dempster et al., 1977). We then propose a novel reformulation of the M-step for high-dimensional predictors.

Suppose we have n independent observations $\{(Y_i, \mathbf{X}_i)\}_{i=1}^n$ from MLR (1). Assume σ^2 is a known constant, we denote $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\beta}_k, k = 1, \ldots, K\}$ as the model parameter and let $f(y, w \mid \mathbf{x}, \boldsymbol{\theta})$ be the joint distribution function of W and Y. If the latent variables $\{W_i\}_{i=1}^n$ can be observed, the log-likelihood function for the complete data is given by

$$\ell_n(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log f(Y_i, W_i \mid \mathbf{X}_i, \boldsymbol{\theta}) = \sum_{i=1}^{n} \{\log \omega_{W_i} + \log f_{W_i}(Y_i \mid \mathbf{X}_i, \boldsymbol{\theta})\}, \tag{2}$$

where $f_k(Y_i \mid \mathbf{X}_i, \boldsymbol{\theta})$ is the probability density function of $Y_i \mid (\mathbf{X}_i, W_i = k) \sim N(\boldsymbol{\beta}_k^T \mathbf{X}_i, \sigma^2)$. The EM algorithm tries to maximize $\ell_n(\boldsymbol{\theta})$ by alternating between the expectation-step (E-step) and maximization-step (M-step) and iteratively updating the estimated parameter $\widehat{\boldsymbol{\theta}}^{(t)}$ for $t = 0, 1, \ldots$. More specifically, consider the $(t+1)$-th iteration with current estimator $\widehat{\boldsymbol{\theta}}^{(t)}$. In the E-step, we evaluate the Q-function, which is defined by taking the expectation of $\ell_n(\boldsymbol{\theta})$ with respect to the latent variable W given the observed data $(Y, \mathbf{X})$ and current estimator $\widehat{\boldsymbol{\theta}}^{(t)}$. Under model (1), the Q-function can be explicitly written as

$$Q(\boldsymbol{\theta} \mid \widehat{\boldsymbol{\theta}}^{(t)}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \widehat{\eta}_{ik}^{(t+1)} \{-\frac{1}{2\sigma^2}(Y_i - \boldsymbol{\beta}_k^T \mathbf{X}_i)^2 + \log(\frac{\omega_k}{\sqrt{2\pi}\sigma})\}, \tag{3}$$

where $\widehat{\eta}_{ik}^{(t+1)} = \Pr(W_i = k | Y_i, \mathbf{X}_i, \widehat{\boldsymbol{\theta}}^{(t)})$ is the membership weight and is calculated by

$$\widehat{\eta}_{ik}^{(t+1)} = \frac{\widehat{\omega}_k^{(t)} f_k(Y_i \mid \mathbf{X}_i, \widehat{\boldsymbol{\theta}}^{(t)})}{\sum_{k=1}^{K} \widehat{\omega}_k^{(t)} f_k(Y_i \mid \mathbf{X}_i, \widehat{\boldsymbol{\theta}}^{(t)})}. \tag{4}$$

In the M-step, we update the estimator $\widehat{\boldsymbol{\theta}}^{(t+1)} = \{\widehat{\omega}_k^{(t+1)}, \widehat{\boldsymbol{\beta}}_k^{(t+1)}, k = 1, \ldots, K\}$ by maximizing

the Q-function as $\widehat{\boldsymbol{\theta}}^{(t+1)} = \text{argmax}_{\boldsymbol{\theta}}\, Q(\boldsymbol{\theta} \mid \widehat{\boldsymbol{\theta}}^{(t)})$. This leads to the following updates:

$$\begin{aligned} \widehat{\omega}_k^{(t+1)} &= \sum_{i=1}^{n} \widehat{\eta}_{ik}^{(t+1)}/n, \\ \widehat{\boldsymbol{\beta}}_k^{(t+1)} &= (\widehat{\boldsymbol{\Sigma}}_k^{(t+1)})^{-1}\widehat{\boldsymbol{\rho}}_k^{(t+1)}, \quad k = 1, \ldots, K, \end{aligned} \tag{5}$$

where $\widehat{\boldsymbol{\Sigma}}_k^{(t+1)} = \sum_{i=1}^n \widehat{\eta}_{ik}^{(t+1)}\mathbf{X}_i\mathbf{X}_i^T / \sum_{i=1}^n \widehat{\eta}_{ik}^{(t+1)}$ and $\widehat{\boldsymbol{\rho}}_k^{(t+1)} = \sum_{i=1}^n \widehat{\eta}_{ik}^{(t+1)}\mathbf{X}_i Y_i / \sum_{i=1}^n \widehat{\eta}_{ik}^{(t+1)}$ can be viewed as intermediate estimators for $\boldsymbol{\Sigma} = \text{var}(\mathbf{X})$ and $\boldsymbol{\rho}_k = \text{E}(Y\mathbf{X} \mid W = k)$.

2.2 *Regularized M-step and Groupwise Variable Selection*

In the high-dimensional setting when $p \gg n$, the standard EM algorithm becomes infeasible as the covariance in (5) is not invertible and the estimators $\widehat{\boldsymbol{\beta}}_k^{(t+1)}$ are not well-defined. To resolve this issue, we assume that only a subset of predictors are relevant to the MLR modeling. We say the j-th predictor is not important if and only if it contributes nothing to regression in any mixtures, that is,

$$\beta_{1j} = \beta_{2j} = \cdots = \beta_{Kj} = 0, \tag{6}$$

where β_{kj} is the j-th element in $\boldsymbol{\beta}_k$ and thus is the regression coefficient of the j-th predictor in k-th mixture. Let $\mathcal{S} \subseteq \{1, \ldots, p\}$ be the index set of important variables, $\mathcal{S}^c$ be the complement of $\mathcal{S}$, and $(\boldsymbol{\beta}_k)_{\mathcal{S}^c}$ be the vector in $\mathbb{R}^p$ which has the same coordinates as $\boldsymbol{\beta}_k$ on $\mathcal{S}^c$ and zero coordinates on the S. Then, (6) implies $(\boldsymbol{\beta}_k)_{\mathcal{S}^c} = 0$ for all $k = 1, \ldots, K$.

Based on this set $\mathcal{S}$ of important variables, the groupwise variable selection, which aims at selecting important variables simultaneously across all mixtures, is very natural and interpretable under the MLR model (1). Moreover, this definition of the important variable based on (6) is also well-integrated in the EM algorithm. To see this, we can rewrite the membership weight $\widehat{\eta}_{ik}$ updates in (4) as

$$\widehat{\eta}_{ik}^{(t+1)} = \widehat{\omega}_k^{(t)} / \Big(\widehat{\omega}_k^{(t)} + \sum_{k' \neq k} \widehat{\omega}_{k'}^{(t)} \exp\Big\{(\widehat{\boldsymbol{\beta}}_{k'}^{(t)} - \widehat{\boldsymbol{\beta}}_k^{(t)})^T\mathbf{X}_i(Y_i - \frac{(\widehat{\boldsymbol{\beta}}_{k'}^{(t)} + \widehat{\boldsymbol{\beta}}_k^{(t)})^T\mathbf{X}_i}{2})/\sigma^2\Big\}\Big). \tag{7}$$

From (7), it is evident that the j-th predictor is irrelevant to the E-step if and only if

$\widehat{\boldsymbol{\beta}}_{k'j}^{(t)} - \widehat{\boldsymbol{\beta}}_{kj}^{(t)} = 0$ for all k and k'. The joint sparsity by (6) directly ensures only predictors belonging to $\mathcal{S}$ are useful for estimating $\widehat{\eta}_{ik}^{(t+1)}$ and hence updating the Q-function (3).

Motivated by this, Wang et al. (2023) generalized the standard EM algorithm by using a group lasso penalty (Yuan and Lin, 2006) in the M-step and established excellent statistical properties. We refer to this method as GEM. Specifically, the group lasso penalty for all $\boldsymbol{\beta}$'s is denoted as $\mathcal{P}_\lambda(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} \sqrt{\sum_{k=1}^{K} \beta_{kj}^2}$, where $\lambda > 0$ is the tuning parameter. The regularized M-step is then defined as the maximization of $Q(\boldsymbol{\theta} \mid \widehat{\boldsymbol{\theta}}^{(t)}) - \mathcal{P}_\lambda(\boldsymbol{\beta})$. The resulting maximization over $\boldsymbol{\beta}$ is equivalent to the following minimization problem,

$$(\widehat{\boldsymbol{\beta}}_1^{(t+1)}, \ldots, \widehat{\boldsymbol{\beta}}_K^{(t+1)}) = \underset{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{k=1}^{K} L_k(\boldsymbol{\beta}_k) + \mathcal{P}_\lambda(\boldsymbol{\beta}), \tag{8}$$

where the convex objective function $L_k(\boldsymbol{\beta}_k)$ is

$$L_k(\boldsymbol{\beta}_k) = \boldsymbol{\beta}_k^T \widehat{\boldsymbol{\Sigma}}_k^{(t+1)} \boldsymbol{\beta}_k - 2(\widehat{\boldsymbol{\rho}}_k^{(t+1)})^T \boldsymbol{\beta}_k. \tag{9}$$

The optimization (8) is convex and can be solved by several available algorithms including the block coordinate descent with composite gradient methods (Nesterov, 2013) and the group-wise majorization descent algorithm (Yang and Zou, 2015). However, all these algorithms iteratively update p blocks of the coefficient matrix $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K) \in \mathbb{R}^{p \times K}$, and within each block, an additional iterative algorithm is required to compute the solution. In other words, each EM iteration involves two nested inner loops, making the computation of the overall algorithm computationally very expensive.

On the other hand, if we replace $\mathcal{P}_\lambda(\boldsymbol{\beta})$ with K separate penalties on $\boldsymbol{\beta}_k$, $k = 1, \ldots, K$, then the estimation problem is greatly simplified. For example, using K separate lasso penalties (Yi and Caramanis, 2015; Zhang et al., 2020), the regularized M-step is separated in K much simpler estimation problems,

$$\widehat{\boldsymbol{\beta}}_k^{(t+1)} = \underset{\boldsymbol{\beta}_k \in \mathbb{R}^p}{\operatorname{argmin}} L_k(\boldsymbol{\beta}_k) + \lambda_k \|\boldsymbol{\beta}_k\|_1, \tag{10}$$

where λ_k is the tuning parameter for the k-th mixture. We call this lasso penalized EM algorithm LPEM. Computationally, (10) can be tens or hundreds of times faster than (8).

For the GEM, in each M-step of the EM algorithm, we solve a group lasso penalized least square problem, for which we need to construct a Majorization Minimization (MM) algorithm (Hunter and Lange, 2004, e.g.) and in each MM updates, we run a blockwise coordinate descent algorithms. Thus, GEM has two inner loops, but FGEM and LPEM only have one inner loop. However, the variable selection from (10) can only be interpreted in terms of each mixture and would fail to exploit the synergy in variable selection across mixtures.

Based on these considerations, we next propose a new formulation for group lasso penalized M-step with a single inner loop as in LPEM. The new algorithm, which enjoys strong theoretical guarantees, is shown to be computationally as efficient as solving K separate lasso problems in (10).

2.3 *Fast group-penalized EM (FGEM) algorithm*

Recall that $\widehat{\boldsymbol{\Sigma}}_k^{(t+1)} = \sum_{i=1}^n \widehat{\eta}_{ik}^{(t+1)} \mathbf{X}_i \mathbf{X}_i^T / \sum_{i=1}^n \widehat{\eta}_{ik}^{(t+1)}$ and $\widehat{\boldsymbol{\rho}}_k^{(t+1)} = \sum_{i=1}^n \widehat{\eta}_{ik}^{(t+1)} \mathbf{X}_i Y_i / \sum_{i=1}^n \widehat{\eta}_{ik}^{(t+1)}$ in (9) are estimators for $\boldsymbol{\Sigma} = \mathrm{var}(\mathbf{X})$ and $\boldsymbol{\rho}_k = \mathrm{E}(Y\mathbf{X} \mid W = k)$. Because of the independence between the latent variable W and the predictor $\mathbf{X}$, it can be shown that $\boldsymbol{\rho}_k = \boldsymbol{\Sigma}\boldsymbol{\beta}_k$ and $\boldsymbol{\beta}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\rho}_k$. Based on this observation, we propose to re-formulate the objective function (9) in the M-step by replacing the iterative estimator $\widehat{\boldsymbol{\Sigma}}_k^{(t+1)}$ with the sample covariance $\widehat{\boldsymbol{\Sigma}} = \sum_{i=1}^n \mathbf{X}_i \mathbf{X}_i^T / n$. Then in the regularized M-step, we replace $L_k(\boldsymbol{\beta}_k)$ with the following modified convex objective function,

$$\widetilde{L}_k(\boldsymbol{\beta}_k) = \boldsymbol{\beta}_k^T \widehat{\boldsymbol{\Sigma}} \boldsymbol{\beta}_k - 2(\widehat{\boldsymbol{\rho}}_k^{(t+1)})^T \boldsymbol{\beta}_k. \tag{11}$$

With a slight re-use of notation, we define our estimates as follows,

$$(\widehat{\boldsymbol{\beta}}_1^{(t+1)}, \ldots, \widehat{\boldsymbol{\beta}}_K^{(t+1)}) = \underset{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{k=1}^K \widetilde{L}_k(\boldsymbol{\beta}_k) + \mathcal{P}_\lambda(\boldsymbol{\beta}), \tag{12}$$

where $\mathcal{P}_\lambda(\boldsymbol{\beta}) = \lambda \sum_{j=1}^p \sqrt{\sum_{k=1}^K \beta_{kj}^2}$. Although the optimization in (12) is only a minor modification from (8), it is frequently solved in high-dimensional classification and clustering

(Mai et al., 2019; Cai et al., 2019; Mai et al., 2021). It can be solved efficiently by the blockwise coordinate descent algorithm, see Algorithm S.1 in the Supplementary Materials for details.

The formulation of (11) exploits the model assumption that W and $\mathbf{X}$ are independent. It replaces the likelihood-based mixture-dependent estimator $\widehat{\boldsymbol{\Sigma}}_k$ with the moment-based sample estimator $\widehat{\boldsymbol{\Sigma}}$ for $\mathrm{E}(\mathbf{X}\mathbf{X}^T \mid W = k) \equiv \mathrm{E}(\mathbf{X}\mathbf{X}^T)$. This enables a more stable EM algorithm in high-dimensional scenarios because the iterative updates of covariance estimates for each mixture bring extraneous variability. By this modification, the quadratic forms of $\boldsymbol{\beta}_k$ in (11) become identical for all $k = 1, \ldots, K$. This modification breaks the monotonicity property in the EM algorithm (i.e., sequentially increasing objective function values over each iterations). Nevertheless, the direct output of the proposed algorithm is shown (in Section 3) to converge to the ground truth. To gain additional insights of this nice theoretical property, we can verity that our modification maintains the unbiasedness of the algorithmic output: Given the true parameters $\boldsymbol{\theta}^*$, the solutions for the expectation of (9) and (11) are exactly $\boldsymbol{\beta}_k^*$. This property is the foundation for both the EM algorithm and the proposed algorithm to work. The proposed approach is thus an intuitive and computationally efficient modification to the M step that preserves the unbiasedness property of the classical EM algorithm.

[Figure 1 about here.]

The implementation details of our FGEM are summarized in Algorithm 1. The computation of FGEM in (12) is much more efficient than that of GEM in (8) because each of the p blocks can be updated with a closed-form solution, which reduces the double-looped iterations within each M step into a single loop. To illustrate the computational time costs of the proposed FGEM (12), the GEM from (8), and the LPEM from (10), we consider Model **M2 (S1)** from the simulation section. The top two panels of Figure 1 summarize the relative computational cost of each method. Clearly, GEM is considerably more expensive than FGEM, especially when the dimension p is large. On average, FGEM is about 100

Algorithm 1 FGEM algorithm for MLR.

1. **Input**: Initialization of $\widehat{\omega}_k^{(0)}$, $\widehat{\boldsymbol{\beta}}_k^{(0)}$, maximum number of iterations T and sample covariance $\widehat{\boldsymbol{\Sigma}} = \sum_{i=1}^n \mathbf{X}_i\mathbf{X}_i^T/n$.
2. For $t = 0, 1, 2, \ldots, T$, repeat the following steps
   (a) E-step: For all k and i, compute $\eta_{ik}^{(t+1)}$ through (7):
   $$\widehat{\eta}_{ik}^{(t+1)} = \widehat{\omega}_k^{(t)} / \Big(\widehat{\omega}_k^{(t)} + \sum_{k' \neq k} \widehat{\omega}_{k'}^{(t)} \exp\Big\{ (\widehat{\boldsymbol{\beta}}_{k'}^{(t)} - \widehat{\boldsymbol{\beta}}_k^{(t)})^T \mathbf{X}_i (Y_i - \frac{(\widehat{\boldsymbol{\beta}}_{k'}^{(t)} + \widehat{\boldsymbol{\beta}}_k^{(t)})^T \mathbf{X}_i}{2}) / \sigma^2 \Big\} \Big).$$
   (b) M-step:
      i. For $k = 1, \ldots, K$, update $\widehat{\boldsymbol{\rho}}_k^{(t+1)}$ and $\widehat{\omega}_k^{(t+1)}$ by
      $$\widehat{\boldsymbol{\rho}}_k^{(t+1)} = \sum_{i=1}^n \widehat{\eta}_{ik}^{(t+1)} \mathbf{X}_i Y_i / \sum_{i=1}^n \widehat{\eta}_{ik}^{(t+1)}, \quad \widehat{\omega}_k^{(t+1)} = \sum_{i=1}^n \widehat{\eta}_{ik}^{(t+1)} / n.$$
      ii. Update $\widehat{\boldsymbol{\beta}}^{(t+1)} = (\widehat{\boldsymbol{\beta}}_1^{(t+1)}, \ldots, \widehat{\boldsymbol{\beta}}_K^{(t+1)})$ by minimizing the following objective function:
      $$\sum_{k=1}^K \Big\{ \boldsymbol{\beta}_k^T \widehat{\boldsymbol{\Sigma}} \boldsymbol{\beta}_k - 2 (\widehat{\boldsymbol{\rho}}_k^{(t+1)})^T \boldsymbol{\beta}_k \Big\} + \lambda \sum_{j=1}^p \sqrt{\sum_{k=1}^K \beta_{kj}^2}.$$
3. **Output**: $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}^{(t+1)}$.

times faster than FGEM. Compared to LPEM, which does not consider groupwise variable selection, FGEM is as efficient computationally as LPEM for this $K = 2$ example. Next, the bottom two panels of Figure 1 use the real data example in Section 4.3 to further demonstrate that FGEM has a edge over LPEM when K increases. Specifically, we report the computational time in seconds of FGEM and LPEM for each of the two responses AZD6244 and PD0325901 when $p = 500$. Both algorithms were implemented by `R` and the comparison was conducted using a high-performance computing cluster with 16GB of CPU memory. We can see that the computation of FGEM scales well with the number of mixtures K and is particularly suitable for group variable selection when K is not small.

2.4 *Practical considerations and implementation details*

Initialization We adopt the tensor power method Anandkumar et al. (2014) to obtain the initialization for the proposed method in practice. Under our model (1), tensor power method uses the centered third-order moments $\mathbf{M}_3 \in \mathbb{R}^{p\times p\times p}$ between response Y and predictor $\mathbf{X}$ to obtain consistent estimation for $\boldsymbol{\beta}_k$, $k = 1, \ldots, K$. Let $\mathbf{e}_i \in \mathbb{R}^p$ be the unit vector whose i-th entry is 1 and all other entries are 0. Then $\mathbf{M}_3 = \mathrm{E}(Y \cdot \mathbf{X} \circ \mathbf{X} \circ \mathbf{X}) - \sum_i \mathrm{E}(Y \cdot \mathbf{e}_i \circ \mathbf{X} \circ \mathbf{e}_i) - \sum_i \mathrm{E}(Y \cdot \mathbf{e}_i \circ \mathbf{e}_i \circ \mathbf{X}) - \sum_i \mathrm{E}(Y \cdot \mathbf{X} \circ \mathbf{e}_i \circ \mathbf{e}_i)$ has the following tensor rank decomposition form $\mathbf{M}_3 = \sum_{k=1}^{K} 6\omega_k \cdot \boldsymbol{\beta}_k \circ \boldsymbol{\beta}_k \circ \boldsymbol{\beta}_k$. This form allows us to obtain the moment-based estimator for all $\boldsymbol{\beta}_k$'s through tensor power decomposition algorithm (Anandkumar et al., 2014). Note that the tensor power method is only proven to work under low-dimensional settings and requires a large sample size to warrant accurate performance (Sedghi et al., 2016; Chaganty and Liang, 2013; Zeng et al., 2023). In our experience, the tensor power method works better than random initialization but could still fail to provide good initialization for some settings. To initialize the high-dimensional EM algorithms, we first fit a simple lasso regression as a pre-screening step and apply the tensor power method on the selected variables to obtain sparse initial values for $\boldsymbol{\beta}_k$'s. In addition, we initialize η_{ik} from (7) with equal probability ω_k's across mixtures and known σ. From Section 4, we notice that FGEM is relatively insensitive to initialization compared to other high-dimensional EM algorithms and the above mentioned initialization method provides satisfactory results for FGEM under most settings. The initialization problem for the mixture model in high-dimensional data is a challenging issue, and the tensor power method may serve as a feasible solution in mostly low-dimensional settings.

Convergence. Our implementation of FGEM has a stopping criterion based on the log-likelihood at the end of the t-th iteration $\ell^{(t)}$, $t = 0, 1, \ldots$. If $\ell^{(t)}$ decrease in the next two iterations such that $\ell^{(t)} > \ell^{(t+1)}$ and $\ell^{(t)} > \ell^{(t+2)}$, then we back track the FGEM estimates

$\widehat{\boldsymbol{\theta}}^{(t)}$ as the output. In our experience, this stopping criterion works well in high dimensions, where the proposed FGEM algorithm usually improves the likelihood dramatically in the first few iterations and has accurate estimates at convergence.

Tuning parameter. As with many other penalized approaches, selecting the regularization parameter λ is a critical step for practice. For theoretical studies, we vary $\lambda^{(t)}$ with iterations as

$$\lambda_n^{(t+1)} = \kappa\lambda_n^{(t)} + C_\lambda\sqrt{\frac{\log(p)\log(n)^2}{n}}, \tag{13}$$

where $\kappa \in (0,1)$ and $C_\lambda > 0$ are some constants. In practice we fix $\lambda^{(t)} = \lambda$ for all t and adopt a BIC-type criterion to tune λ. Specifically, we seek the value of λ that minimizes the BIC defined as

$$\mathrm{BIC}(\lambda) = -2\sum_{i=1}^{n}\log(\sum_{k=1}^{K}\widehat{\omega}_k^\lambda f_k(Y_i \mid \mathbf{X}_i, \widehat{\boldsymbol{\theta}}^\lambda)) + \log(n)\cdot|\widehat{\mathcal{D}}^\lambda|, \tag{14}$$

where $\widehat{\boldsymbol{\theta}}^\lambda$ is the output of FGEM with the tuning parameter fixed at λ, and $\widehat{\mathcal{D}}^\lambda = \{(k,j) : \widehat{\boldsymbol{\beta}}_{k,j}^\lambda \neq 0\}$ is the set of nonzero elements in $\widehat{\boldsymbol{\beta}}_1^\lambda, \ldots, \widehat{\boldsymbol{\beta}}_K^\lambda$. Alternatively, one can use cross-validation to select the best λ that minimizes the cross-validated negative log-likelihood or prediction squared error (Städler et al., 2010).

Choice of σ^2. In the FGEM algorithm (Algorithm 1) and our numerical and theoretical studies, we assume $\sigma^2 = \mathrm{var}(\epsilon)$ is known and fixed. This is because the main goal of this paper is in the sparse estimation of $\boldsymbol{\beta}_k$. Accurate estimation and theoretical investigation on mis-specification of σ^2 is usually treated as a separate research problem in MLR and is beyond the scope of this paper. One can iteratively estimate the variance during the M-step by $\widehat{\sigma}^2 = \sum_{i=1}^n\sum_{k=1}^K \widehat{\eta}_{ik}^{(t+1)}(Y_i - \widehat{\boldsymbol{\beta}}_k^{(t+1)T}\mathbf{X}_i)^2/n$. This works reasonably well as we illustrate in the in the Supplementary Materials Section S.1.1.

Selection of K. To select groups K, we consider a modified Bayesian information criterion:

$$\mathrm{BIC}(\lambda, K) = -2\sum_{i=1}^{n}\log(\sum_{k=1}^{K}\widetilde{\omega}_k^\lambda f_k(Y_i \mid \mathbf{X}_i, \widetilde{\boldsymbol{\theta}}^\lambda)) + \log(n)K\cdot|\widehat{\mathcal{D}}^\lambda|, \tag{15}$$

where $\widetilde{\boldsymbol{\theta}}^{\lambda}$ is the output of the low-dimensional EM algorithm based on the selected variables from FGEM (which is labeled as FGEM* in the simulation studies), and $\widehat{\mathcal{D}}^{\lambda} = \{(k,j) : \widehat{\boldsymbol{\beta}}^{\lambda}_{k,j} \neq 0\}$ is the set of nonzero elements in $\widehat{\boldsymbol{\beta}}^{\lambda}_1, \ldots, \widehat{\boldsymbol{\beta}}^{\lambda}_K$. Unlike the Bayesian information criterion in (15), where the output of FGEM is used for tuning λ, we use FGEM* instead for selecting K. Such a modification is necessary because the group lasso penalty brings bias for FGEM estimates. From both theoretical and empirical considerations, the bias is very harmful to the selection of K. Although FGEM* overlooked the variable selection variability, it is still a favorable plug-in estimates for the BIC-type selection procedure. The simulation studies in Supplementary Section S.1.5 confirm the encouraging performance of this modified BIC in selecting K. Additionally, it is often desirable in practice to have multiple methods implemented for selecting K. For example, the modified S4 method in Liu et al. (2023) may be served as an alternative method.

3. Theory

In this section, we consider the statistical convergence analysis for the FGEM algorithm. The theoretical analysis for FGEM is a non-trivial extension from the PEM theory in Wang et al. (2023): New lemmas and modifications in the technical proof are developed in view of the existing work in Wang et al. (2023), because of our computationally motivated changes in the objective function from penalized likelihood (8) to a convex quadratic form (12).

We begin with some notations. For numbers a and b, $a \vee b$ means $\max\{a, b\}$. For an integer n, we let $[n]$ denote the set $\{1, \cdots, n\}$. For a vector $\mathbf{x} = (x_1, \cdots, x_p)^T$, $\|\mathbf{x}\|_0$ is the number of non-zero elements in $\mathbf{x}$, $\|\mathbf{x}\|_1 = \sum_{i=1}^p |\mathbf{x}_i|$, and $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^p x_i^2}$. For a symmetric matrix $\mathbf{A}$, we denote $\lambda_{min}(\mathbf{A})$ and $\lambda_{max}(\mathbf{A})$ as the smallest and largest eigenvalues of $\mathbf{A}$, respectively. For a subset $\mathcal{A} \subseteq \{1, \cdots, p\}$, $\mathcal{A}^c$ denotes its complement. For two sequences of positive numbers a_n and b_n, $a_n = O(b_n)$ means $a_n \leqslant c b_n$ for some constant $c > 0$ for all n, $a_n = o(b_n)$ means that $a_n / b_n \to 0$ as $n \to \infty$, and $b_n \gg a_n$ means that $a_n = o(b_n)$. Let $\mathcal{S}^{p-1}$ be the unit sphere.

For a positive integer $s \leqslant p/2$, let $\Gamma_{2p}(2s) = \{\boldsymbol{\mu} \in \mathbb{R}^{2p} : \|\boldsymbol{\mu}_{S^c}\|_1 \leqslant 5\sqrt{2s}\|\boldsymbol{\mu}_S\|_2 + 2\sqrt{2s}\|\boldsymbol{\mu}\|_2$ for some $S \subset [2p]$ with $|S| = 4s\}$ and $\Gamma(s) = \Gamma_{2p}(2s)_{1:p} = \{\mu_{1:p} : \boldsymbol{\mu} \in \Gamma_{2p}(2s)\}$.

Our theoretical results are established for $K = 2$. This is a common assumption in most of the existing literature for high-dimensional penalized EM algorithms (Yi and Caramanis, 2015; Cai et al., 2019; Zhang et al., 2020; Wang et al., 2023, e.g.). We further assume independent random predictors $\mathbf{X}_i \sim N(0, \boldsymbol{\Sigma})$, which is also used in (Zhang et al., 2020) and (Wang et al., 2023) and is less restrictive than the assumption of $\mathbf{X}_i \sim N(0, \mathbf{I}_p)$ in Yi and Caramanis (2015) and Balakrishnan et al. (2017). Let $\boldsymbol{\theta}^*$ be the true value of $\boldsymbol{\theta}$, and $\widehat{\boldsymbol{\theta}}^{(t)}$ be the estimate of $\boldsymbol{\theta}$ at the t-th step in Algorithm 1. The true parameter space we consider is

$$\boldsymbol{\Theta}^* = \{\boldsymbol{\theta}^* : \omega_1^* \in (c_w, 1 - c_w), \|\boldsymbol{\beta}_k^*\|_0 \leqslant s, \|\boldsymbol{\beta}_k^*\|_2 \leqslant M_b, \text{ for } k = 1, 2\},$$

where M_b is a generic constant. Similar to existing theoretical works (Yi and Caramanis, 2015; Balakrishnan et al., 2017; Zhang et al., 2020; Wang et al., 2023, e.g.), we treat σ^2 as a known parameter to facilitate the theoretical studies. Without loss of generality, we further assume $\sigma^2 = 1$. Let $\Delta = \sqrt{(\boldsymbol{\beta}_2^* - \boldsymbol{\beta}_1^*)^T \boldsymbol{\Sigma} (\boldsymbol{\beta}_2^* - \boldsymbol{\beta}_1^*)}$, which is a measure of the signal-to-noise ratio of the mixed linear regression model. Heuristically, a large Δ means separable between the two mixtures and makes the estimate easier. Because FGEM handles a non-convex optimization problem, to guarantee global convergence, we require the initialization is not far away from the true value. Hence, we define the contraction basin $\mathcal{B}_{con}(\boldsymbol{\theta}^*)$ as follows.

$$\mathcal{B}_{con}(\boldsymbol{\theta}^*) = \{\boldsymbol{\theta}^* : \omega_k \in (c_0, 1 - c_0), \|\boldsymbol{\beta}_k - \boldsymbol{\beta}_k^*\|_2 \leqslant C_b \Delta, \boldsymbol{\beta}_k - \boldsymbol{\beta}_k^* \in \Gamma(s), \text{ for } k = 1, 2\}.$$

We further introduce some technical lemmas as follows.

(1) The eigenvalues of $\boldsymbol{\Sigma}$ satisfy that $M_1 \leqslant \lambda_{min}(\boldsymbol{\Sigma}) \leqslant \lambda_{max}(\boldsymbol{\Sigma}) \leqslant M_2$, for all $k = 1, \cdots, K$.

(2) $n \gg s \log(p) \log(n)^2$.

(3) The signal-to-noise ratio $\Delta > C_1(c_0)$ for some constant $C_1(c_0)$ only depends on c_0, and $C_b < C_2(c_0, M_2)$ for some constant $C_2(c_0, M_2)$ only depends on c_0, M_2.

(4) The initialization $\widehat{\boldsymbol{\theta}}_0^{(0)} = (\widehat{\omega}_1^{(0)}, \widehat{\boldsymbol{\beta}}_1^{(0)}, \widehat{\boldsymbol{\beta}}_2^{(0)}) \in \mathcal{B}_{con}(\boldsymbol{\theta}^*)$.

The same technical conditions are also used in Wang et al. (2023). We briefly explain why these conditions are required for the theorem. Condition (C1) assumes that the covariance matrix for the predictor is well-behaved and rules out the cases, where the elements of $\mathbf{X}$ are highly correlated. Condition (C2) is a requirement for the sample size, which guarantees the restrictive eigenvalue condition. Condition (C3) requires that the signal-to-noise ratio is large enough and the initialization is not far away from the true value of the parameters. Condition (C4) ensures that the initialization is in the contraction basin, which guarantees the subsequent estimators in FGEM are all contained in the contraction basin.

Based on the above conditions, we establish the following theorem.

THEOREM 1: *Suppose we have n independent observations from Model* (1) *with the true parameter $\boldsymbol{\theta}^* \in \boldsymbol{\Theta}^*$. Under conditions (C1)-(C4), there exists constant $0 < \kappa < 1/2$, such that $\widehat{\boldsymbol{\beta}}_k^{(t+1)}$ obtained by Algorithm 1 satisfies, with probability $1 - o(1)$,*

$$\|\widehat{\boldsymbol{\beta}}_k^{(t+1)} - \boldsymbol{\beta}_k^*\|_2 = O\Big(\kappa^t(|\widehat{\omega_1}^{(0)} - \omega_1^*| \vee \|\widehat{\boldsymbol{\beta}}_1^{(0)} - \boldsymbol{\beta}_1^*\|_2 \vee \|\widehat{\boldsymbol{\beta}}_2^{(0)} - \boldsymbol{\beta}_2^*\|_2) + \sqrt{\frac{s\log(n)^2\log(p)}{n}}\Big).$$

Consequently, for $t \geqslant (-\log(\kappa))^{-1}\log(n(|\widehat{\omega_1}^{(0)} - \omega_1^*| \vee \|\widehat{\boldsymbol{\beta}}_1^{(0)} - \boldsymbol{\beta}_1^*\|_2 \vee \|\widehat{\boldsymbol{\beta}}_2^{(0)} - \boldsymbol{\beta}_2^*\|_2))$,

$$\|\widehat{\boldsymbol{\beta}}_k^{(t+1)} - \boldsymbol{\beta}_k^*\|_2 = O\Big(\sqrt{\frac{s\log(n)^2\log(p)}{n}}\Big).$$

Some technical lemmas and detailed proofs are included in the Supplementary Materials. Intuitively, the convergence rate consists of two parts: the first part $\kappa^t(|\widehat{\omega_1}^{(0)} - \omega_1^*| \vee \|\widehat{\boldsymbol{\beta}}_1^{(0)} - \boldsymbol{\beta}_1^*\|_2 \vee \|\widehat{\boldsymbol{\beta}}_2^{(0)} - \boldsymbol{\beta}_2^*\|_2)$ can be viewed as computational error, which decreases with iteration step t, and the second part $\sqrt{s\log(n)^2\log(p)/n}$ is the statistical error. After some iterations, the computational error will be dominated by the statistical error. The proposed FGEM method shares similar statistical convergence properties as the state of the art high-dimensional MLR algorithms, such as Zhang et al. (2020) and Wang et al. (2023). We will demonstrate the encouraging performance of FGEM in the numerical studies. The nice theoretical results and fast computation make FGEM more plausible in real-world applications.

4. Numerical studies

4.1 *Simulations under MLR model* (1)

In all the simulations considered in this subsection, we set the number of important predictors $s = 10$, and vary dimension $p \in \{500, 1000, 2000\}$; the sample size is set to be $n = 1000$ to ensure algorithmic convergence for most if not all methods; 100 replicated data sets are independently generated for each simulation settings. We consider five settings of different ω_k's and $\boldsymbol{\beta}_k$'s denoted as models **M1**–**M5**:

M1, $\omega_1 = \omega_2 = 0.5$, $\boldsymbol{\beta}_{[1,1:s]} = 3$, and $\boldsymbol{\beta}_{[2,1:s]} = -1$.

M2, $\omega_1 = \omega_2 = 0.5$, $\boldsymbol{\beta}_{[1,1:s]} = 3 + N(0,1)$, and $\boldsymbol{\beta}_{[2,1:s]} = -1 + N(0,1)$.

M3, $\omega_1 = \omega_2 = 0.5$, $\boldsymbol{\beta}_{[1,j]} = 3 + (j-1)/(s-1)$, and $\boldsymbol{\beta}_{[2,1:s]} = -1 + N(0,1)$.

M4, $\omega_1 = \omega_2 = \omega_3 = 1/3$, $\boldsymbol{\beta}_{[1,j]} = 1 + (j-1)/(s-1)$, $\boldsymbol{\beta}_{[1,1:s]} = -1$, and $\boldsymbol{\beta}_{[3,1:s]} = 4$.

M5, $\omega_1 = \omega_2 = 0.5$, $\boldsymbol{\beta}_{[1,1:(s/2)]} = 2$, and $\boldsymbol{\beta}_{[2,(s/2+1):s]} = -2$.

In all the models, $\sigma^2 = 1$. Models **M1**–**M3** have $K = 2$ mixtures with different regression coefficient structures. Model **M4** sets $K = 3$. Model **M5** is against the group sparsity assumption, where the nonzero entries in $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are in complement sets. Three different covariance structures of $\boldsymbol{\Sigma}$, **S1**–**S3**, are considered for each of the five models. Details are as follows. We use $CS(\gamma)$ and $AR(\gamma)$, $-1 < \gamma < 1$, to denote correlation matrices with compound symmetric (i.e. γ on all off-diagonals) and autoregressive (i.e. $\gamma^{|i-j|}$ for the (i,j)-th element) structures, respectively. The visualization of these correlation structures are included in Supplementary Materials Figure S.2.

S1, autoregressive covariance $AR(0.3)$: Entries of $\boldsymbol{\Sigma}$ take the form $\boldsymbol{\Sigma}_{ij} = 0.3^{|i-j|}$.

S2, block-wise sparse precision matrix: Let $\boldsymbol{\Omega} = (\mathbf{B} + \delta \mathbf{I}_p)/(1+\delta)$, where $\mathbf{B}_{ij} = \mathbf{B}_{ji} = 0.5 \times \text{Ber}(1, 0.1)$ for $1 \leqslant i \leqslant s, i < j \leqslant p$; $\mathbf{B}_{ij} = \mathbf{B}_{ji} = 0.5$ for $s+1 \leqslant i < j \leqslant p$; diagonal entries $\mathbf{B}_{ii} = 1$. Here $\delta = \max\{-\phi_{\min}(\mathbf{B}), 0\} + 0.05$ and $\phi_{\min}(\mathbf{B})$ represents the smallest eigenvalue of $\mathbf{B}$. Finally, the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\Omega}^{-1}$.

S3, block-wise compound symmetric: Let $\tilde{\boldsymbol{\Sigma}} \in \mathbb{R}^{s \times s}$ be $CS(0.3)$, then $\boldsymbol{\Sigma} = \mathbf{I}_{p/s} \otimes \tilde{\boldsymbol{\Sigma}}$.

For comparison purposes, we include the following methods besides the proposed FGEM: the low-dimensional EM algorithm fitted on the s important variables only (Oracle); the estimation obtained by the initialization method (Initial); the Gaussian Locally Linear Mapping EM algorithm (GLLiM) by Deleforge et al. (2015), which is implemented using the `R` package `xLLiM`; the lasso penalized EM algorithm (LPEM) in Zhang et al. (2020); the un-penalized EM algorithm based on the selected variables by FGEM (FGEM*); the GEM method (Wang et al., 2023).

[Table 1 about here.]

[Table 2 about here.]

For all combinations of the five models **M1**–**M5** and the three covariance settings **S1**–**S3**, we summarize parameter estimation error in Table 1 and variable selection accuracy in Table 2. All reported numbers (means and standard errors) are based on 100 independent replications. Specifically, estimation error is defined as $\Delta_{\boldsymbol{\beta}} = \min_{\Pi\in\mathcal{P}} \sqrt{\sum_{k=1}^{K} \|\widehat{\boldsymbol{\beta}}_k - \boldsymbol{\beta}_{\Pi(k)}\|_F^2}$, where $\mathcal{P} = \{\Pi : [1,\ldots,K] \to [1,\ldots,K]\}$ is a set of index permutation functions to account for latent mixture identification issue. The usual true positive rate (TPR) and false positive rate (FPR) are used for variable selection of the s important predictors out of a total of p.

Not surprisingly, FGEM achieves substantially smaller estimation error (see Table 1) than GLLiM in all model settings, because GLLiM does not perform variable selection. Because LPEM selects variables separately in each mixture, FGEM outperforms LPEM in most settings **M1**–**M4** where group sparsity exists. However, in model **M5**, where different mixtures have completely different important predictors, LPEM is more favorable than FGEM. Even so, FGEM still achieves comparable variable selection performance under **M5**, and also outperforms LPEM under the more challenging covariance structure **S2** when $p = 1000$ and 2000. For most cases of $p = 500$ and some cases of $p = 1000$, GEM outperforms FGEM in terms of estimation accuracy. This result is not surprising because GEM aims to

solve the "original" penalized likelihood function with the group sparsity structure and is optimal in that sense. However, when the dimension becomes higher, $p = 2000$, FGEM would outperform GEM for most settings. It is encouraging to notice that the estimation error of FGEM remains relatively consistent when dimension p increases, while the estimation errors of LPEM and GEM increase quickly with dimension p. One possible explanation is that LPEM and GEM are more sensitive to initialization. In contrast, the proposed FGEM equipped with modified and simplified M-step is less sensitive to initial values compared to traditional or regularized EM algorithms. The insensitivity to initialization can also be verified by examining the performance of our initialization method in the tables. Clearly, the initialization method on its own has the worst performance among all competing methods but will be improved dramatically by FGEM.

In terms of variable selection accuracy, from Table 2, we can see that FGEM is able to identify the important variables with high TPR and low FPR, while LPEM and GEM are less effective with often higher FPR. In particular, under more complex covariance structures like **S2** where unimportant variables can have big variance and important variables can be highly correlated with unimportant variables, LPEM and GEM fail to capture the important variables and thus obtained high FPR. Additionally, GEM takes an enormous amount of time to converge for our high-dimensional settings (see Figure 1 for the computation time comparison), becoming computationally less practical. LPEM has a significantly bigger estimation error than FGEM. On the other hand, FGEM is more robust with covariance structure and is a fast and stable variable selection approach for high-dimensional heterogeneous data in real applications.

Furthermore, due to the computational advantage, the proposed FGEM can be used as a variable selection step for low-dimensional methods such as the standard EM algorithm. This is illustrated by the outstanding performance of the FGEM* method, which is post-selection

refitting from FGEM. In many cases, FGEM* can perform as well as the oracle method. Compared to FGEM, the dramatic improvement by refitting in FGEM* is because of the reduction in estimative bias in penalized M-steps.

Overall, FGEM's estimation performance is similar to GEM's and superior to LPEM's in most of our simulation settings. However, in terms of computational time, FGEM is significantly more efficient than GEM and comparable to LPEM. In Table 3, we reported the computational time of FGEM, LPEM, and GEM. We can see that FGEM is much faster than GEM, especially when $p = 1000$ or $p = 2000$, where the computational time of GEM is even beyond reasonable.

[Table 3 about here.]

Additional simulation results are relegated to the Supplementary Materials. We consider four additional covariance structures **S4**–**S7** for the same five models **M1**–**M5**. The findings are consistent with the results in Tables 1 and 2. We conducted simulations for unbalanced mixtures and for varying sample size n and dimension p in the Supplementary Materials. In particular, for $K = 2$ and $\omega = \omega_1 = 1 - \omega_2 \neq 0.5$, the FGEM is observed to be more robust than the competitors when ω is closer to 0 or 1. The simulation results in Supplementary Section S.1.5 also comfirm the good performance of BIC interms of selection of λ and K.

4.2 *Simulations under alternative mixture models*

When the assumption $\mathrm{E}(\mathbf{X}\mathbf{X}^T \mid W = k) = \mathrm{E}(\mathbf{X}\mathbf{X}^T)$ is violated, the proposed algorithm may return a biased violation. However, the bias is small when the assumption is not seriously violated in practice. As a demonstration, we included the following two simulation studies, for which the independence between $\mathbf{X}$ and W is violated.

A1: the model settings are the same as M1 except that $P(W_i = 1) = \frac{1}{1+\exp(\boldsymbol{\alpha}^T \mathbf{X}_i)}$, where the first 10 elements of $\boldsymbol{\alpha}$ are generated from $0.1 \times N(0, 1)$ independently, and the other elements are 0.

A2: the model settings are the same as M1 except that $\mathbf{X}_i \mid W_i = 1 \sim N(\widetilde{\boldsymbol{\mu}}, \boldsymbol{\Sigma})$ and $\mathbf{X}_i \mid W_i = 2 \sim N(-\widetilde{\boldsymbol{\mu}}, \boldsymbol{\Sigma})$, where the first 10 elements of $\widetilde{\boldsymbol{\mu}}$ are generated from $0.2 \times N(0, 1)$ independently, and the other elements are 0.

In the new simulation examples **A1** and **A2**, the assumption that $\mathbf{X}$ is independent of W is violated. Specifically, for **A1**, the model assumptions for all of LPEM, GEM, and FGEM are violated, and for **A2**, those for FGEM are violated but not violated for LPEM and GEM. From Table 4, it deserves to be noticed that FGEM performs better and is more stable than LPEM and GEM for both **A1** and **A2**. The reason is that, compared with LPEM and GEM, FGEM is less sensitive to the initialization. Note that the model settings for the tensor power method (initialization method) are also violated for those two simulation models. As a result, the quality of the initialization can be poor. To verify the reason, we reported the results of LPEM and GEM using the output of FGEM as initialization (denoted as LPEM† and GEM†, respectively) in Table 4. We found that LPEM and GEM gain substantial improvement. Because the mixed linear regression model is a non-convex optimization problem, the proposed FGEM, which is less sensitive to the initialization, is preferred in practice. Another result that deserves to be noticed in Table 4 is, except for the Oracle method, FGEM* is the best or among the best. To make a fair comparison, we also included two methods LPEM* and GEM*, which represent the EM algorithm refit on the selected variables from LPEM and GEM, respectively. We can see that FGEM* performs better than LPEM* and GEM* as a result of better variable selection ability and less sensitivity to the initialization. To sum up, FGEM is a fast and robust method, which can be considered as a nice initialization method for GEM and LPEM, and FGEM* further improves FGEM by removing the bias caused by the group lasso penalty and is usually a superior method.

[Table 4 about here.]

4.3 *Real data analysis*

In this section, we demonstrate the proposed FGEM on the cancer cell line encyclopedia (CCLE) dataset. Four additional real datasets are further included in the Supplementary Materials. The Cancer Cell Line Encyclopedia (CCLE) dataset describes 8-point dose-response curves for 24 chemical compounds across over 400 cell lines. Each cell line consists of expression data of 18,926 genes. Following Li et al. (2019), we use the area under the dose-response curve, also known as the activity area, to measure the sensitivity of a drug for each cell line. We consider two chemical compounds as response variables: AZD6244 and PD0325901. For $n = 490$ cell lines treated with the two chemical compounds, we screen the genes by keeping $p = 500$ genes with the biggest absolute correlations with the responses. We also repeat the analysis for $p = 2000$ to demonstrate the applicability of the method to higher dimensions. The complexity of cancer data implies significant heterogeneity and motivates us to fit mixed linear regression models.

We compare the proposed FGEM with LPEM, the EM algorithm refitted on FGEM selected predictors (FGEM*), the initialization method (Initial), and the LASSO regression (i.e., always using $K = 1$). We randomly split the data 100 times and kept 1/5 of the samples as testing set to evaluate prediction performance. For each response, we consider the number of mixtures $K \in \{2, 4, 6, 8, 10\}$ to capture the heterogeneity in the data. The mean squared prediction errors evaluated on the testing data are summarized in Figure 2 for both $p = 500$ and $p = 2000$. For both responses, the prediction error of LASSO is significantly improved by fitting MLR with $K \geqslant 2$, suggesting heterogeneity in the data. For prediction, FGEM performed better than LPEM when $K > 2$ and is about the same for $K = 2$. The post-selection refitting of FGEM* has a similar prediction performance as FGEM for PD0325901. However, for response AZD6244, when K is large, FGEM* tends to perform slightly worse

than FGEM without refitting. This is probably due to the advantage of shrinkage in high-dimensional prediction problems, where the bias in estimation can be helpful.

The prediction error of FGEM tends to first decrease and then remains the same when the number of mixtures K increases. This suggests that over-specification of K is not a serious issue for FGEM's prediction performance. Of course, over-specification of K makes the model more expensive to fit and has a higher variability in parameter estimation. In the Supplementary Materials, we use Figure S.4 to plot BIC versus K for each response and observe that $K = 4$ is selected (in both cases, BIC first decreases and then increases monotonically after $K = 4$).

[Figure 2 about here.]

5. Discussion

We propose a fast group-penalized EM (FGEM) algorithm for high-dimensional mixed linear regression. By exploiting the independence assumption between the latent mixture indicator and the predictor, FGEM is computationally efficient and robust to initialization. Theoretically, we obtain a non-asymptotic convergence rate for a two-mixture setting ($K = 2$). In real data analysis, the prediction performance of the FGEM algorithm is insensitive to the over-specification of K and converges quickly even when $K = 20$. The proposed modification to M-step in EM algorithm is also promising for fast computation of joint penalized EM algorithm in mixed generalized linear models (e.g., Stroup, 2012) and mixed autoregressive models (e.g., Hannan and Rissanen, 1982).

In numerical studies, we consider moderately violating the assumption that $\mathbf{X}$ is independent of W. FGEM still performs well for those alternative simulation examples. Nevertheless, there are potential ways to further improve the performance of FGEM. For example, if we assume $\text{logit}\omega_k = \mathbf{X}^T\boldsymbol{\alpha}_k$, we may extend the idea of FGEM to construct a fast group lasso penalized EM algorithm for such a mixture of expert model. Another possible future

direction is in allowing the variance σ_k^2 to be different across mixtures $k = 1, \ldots, K$. Wang et al. (2023) showed that the influence of σ_k^2 is minor when the signal-to-noise ratio is large, it is still interesting to consider the case when the signal-to-noise ratio is moderately large.

Acknowledgement

The authors would like to thank the Co-Editor, the Associate Editor and two reviewers for helpful comments. Research in this article is partly supported by grants CCF-1908969 (Mai), DMS-2053697 (Zhang), and DMS-2113590 (Zhang) from NSF.

Supporting information

Supplementary materials include additional numerical results, proofs, and the R code.

References

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. (2014). Tensor decompositions for learning latent variable models. *Journal of machine learning research* **15,** 2773–2832.

Balakrishnan, S., Wainwright, M. J., and Yu, B. (2017). Statistical guarantees for the em algorithm: From population to sample-based analysis. *The Annals of Statistics* **45,** 77–120.

Cai, T. T., Ma, J., and Zhang, L. (2019). Chime: Clustering of high-dimensional gaussian mixtures with em algorithm and its optimality. *The Annals of Statistics* **47,** 1234–1267.

Chaganty, A. T. and Liang, P. (2013). Spectral experts for estimating mixtures of linear regressions. In *International Conference on Machine Learning*, pages 1040–1048. PMLR.

Chen, Y., Yi, X., and Caramanis, C. (2014). A convex formulation for mixed regression with two components: Minimax optimal rates. In *Conference on Learning Theory*, pages 560–604.

Deleforge, A., Forbes, F., and Horaud, R. (2015). High-dimensional regression with gaussian mixtures and partially-latent response variables. *Statistics and Computing* **25,** 893–911.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* **39,** 1–22.

Hannan, E. J. and Rissanen, J. (1982). Recursive estimation of mixed autoregressive-moving average order. *Biometrika* **69,** 81–94.

Hui, F. K., Warton, D. I., and Foster, S. D. (2015). Multi-species distribution modeling using penalized mixture of regressions. *The Annals of Applied Statistics* **9,** 866–882.

Hunter, D. R. and Lange, K. (2004). A tutorial on mm algorithms. *The American Statistician* **58,** 30–37.

Khalili, A. and Chen, J. (2007). Variable selection in finite mixture of regression models. *Journal of the american Statistical association* **102,** 1025–1038.

Klusowski, J. M., Yang, D., and Brinda, W. (2019). Estimating the coefficients of a mixture of two linear regressions by expectation maximization. *IEEE Transactions on Information Theory* **65,** 3515–3524.

Kwon, J. and Caramanis, C. (2020). Em converges for a mixture of many linear regressions. In *International Conference on Artificial Intelligence and Statistics*, pages 1727–1736. PMLR.

Kwon, J., Ho, N., and Caramanis, C. (2021). On the minimax optimality of the em algorithm for learning two-component mixed linear regression. In *International Conference on Artificial Intelligence and Statistics*, pages 1405–1413. PMLR.

Li, Q., Shi, R., and Liang, F. (2019). Drug sensitivity prediction with high-dimensional mixture regression. *PloS one* **14,** e0212108.

Liu, D., Zhao, C., He, Y., Liu, L., Guo, Y., and Zhang, X. (2023). Simultaneous cluster

structure learning and estimation of heterogeneous graphs for matrix-variate fmri data. *Biometrics* **79,** 2246–2259.

Mai, Q., Yang, Y., and Zou, H. (2019). Multiclass sparse discriminant analysis. *Statistica Sinica* **29,** 97–111.

Mai, Q., Zhang, X., Pan, Y., and Deng, K. (2021). A doubly enhanced em algorithm for model-based tensor clustering. *Journal of the American Statistical Association* pages 1–15.

McLachlan, G. J., Lee, S. X., and Rathnayake, S. I. (2019). Finite mixture models. *Annual review of statistics and its application* **6,** 355–378.

Nesterov, Y. (2013). Gradient methods for minimizing composite functions. *Mathematical Programming* **140,** 125–161.

Qin, L.-T., Wu, J., Mo, L.-Y., Zeng, H.-H., and Liang, Y.-P. (2015). Linear regression model for predicting interactive mixture toxicity of pesticide and ionic liquid. *Environmental Science and Pollution Research* **22,** 12759–12768.

Sedghi, H., Janzamin, M., and Anandkumar, A. (2016). Provable tensor methods for learning mixtures of generalized linear models. In *Artificial Intelligence and Statistics*, pages 1223–1231. PMLR.

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

Stroup, W. W. (2012). *Generalized linear mixed models: modern concepts, methods and applications*. CRC press.

Wang, N., Zhang, X., and Mai, Q. (2023). Statistical analysis for a penalized em algorithm in high-dimensional mixture linear regression model. *arXiv preprint arXiv:2307.11405* .

Wang, T. and Paschalidis, I. C. (2019). Convergence of parameter estimates for regularized mixed linear regression models. In *2019 IEEE 58th Conference on Decision and Control*

(CDC), pages 3664–3669. IEEE.

Wang, Z., Gu, Q., Ning, Y., and Liu, H. (2015). High dimensional em algorithm: Statistical optimization and asymptotic normality. *Advances in neural information processing systems* pages 2512–2520.

Xu, J., Hsu, D. J., and Maleki, A. (2016). Global analysis of expectation maximization for mixtures of two gaussians. *Advances in Neural Information Processing Systems* **29,** 2676–2684.

Yang, Y. and Zou, H. (2015). A fast unified algorithm for solving group-lasso penalize learning problems. *Statistics and Computing* **25,** 1129–1141.

Yi, X. and Caramanis, C. (2015). Regularized em algorithms: A unified framework and statistical guarantees. *Advances in Neural Information Processing Systems* **28,** 1567–1575.

Yi, X., Caramanis, C., and Sanghavi, S. (2014). Alternating minimization for mixed linear regression. In *International Conference on Machine Learning*, pages 613–621.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68,** 49–67.

Zeng, Z., Gu, Y., and Xu, G. (2023). A tensor-em method for large-scale latent class analysis with binary responses. *Psychometrika* **88,** 580–612.

Zhang, L., Ma, R., Cai, T. T., and Li, H. (2020). Estimation, confidence intervals, and large-scale hypotheses testing for high-dimensional mixed linear regression. *arXiv preprint arXiv:2011.03598* .

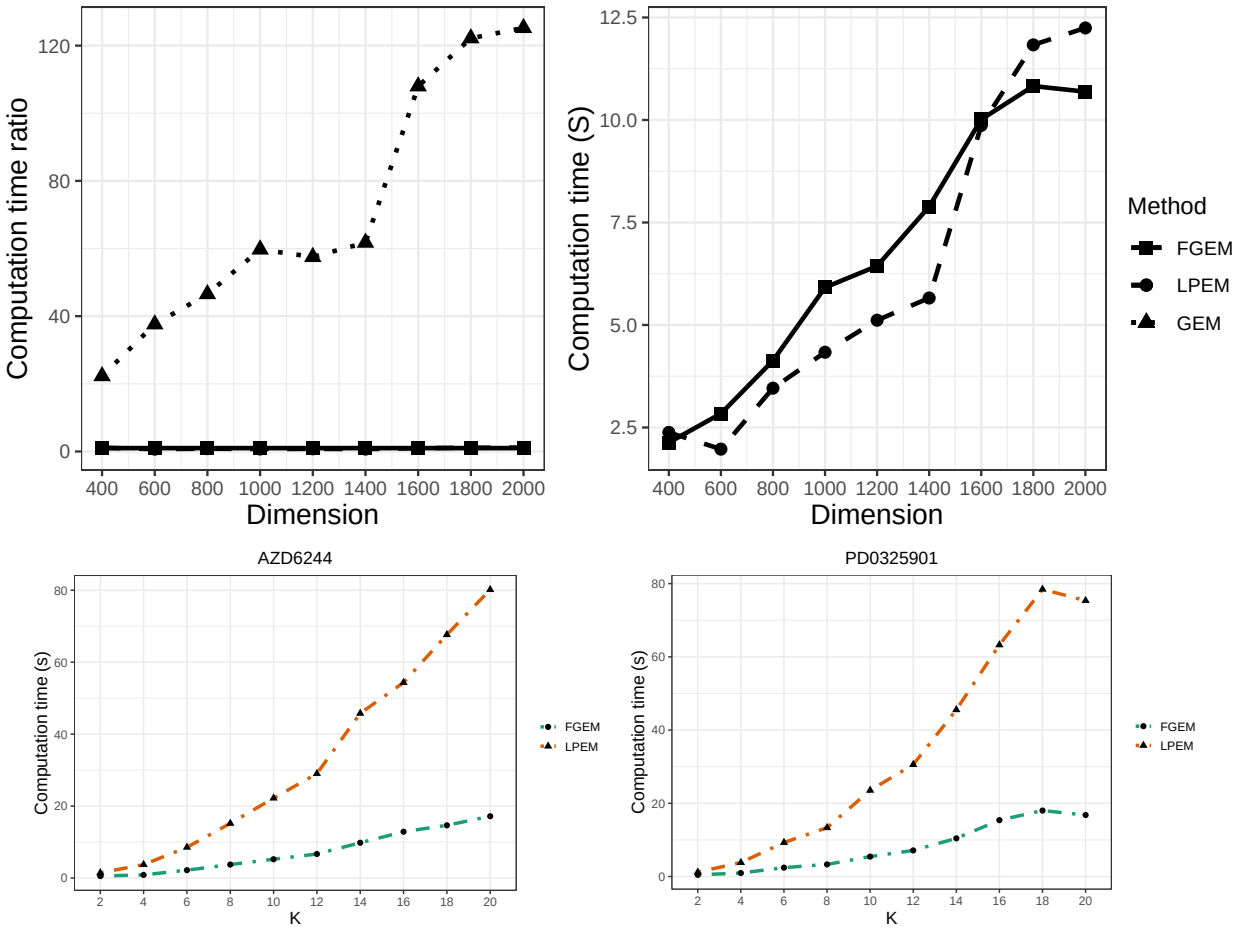

Figure 1. Computation time comparison. Results in the top two panels are averaged over 100 independent data replications under Model **M2** and covariance structure **S1** from Section 4.1. Top left: computation time ratio in reference to FGEM (comparing GEM to FGEM). Top right: computation time in seconds (comparing LPEM to FGEM). Bottom two panels report the computation time (in seconds) of FGEM and LPEM for each of the two response variables in the CCLE data set.

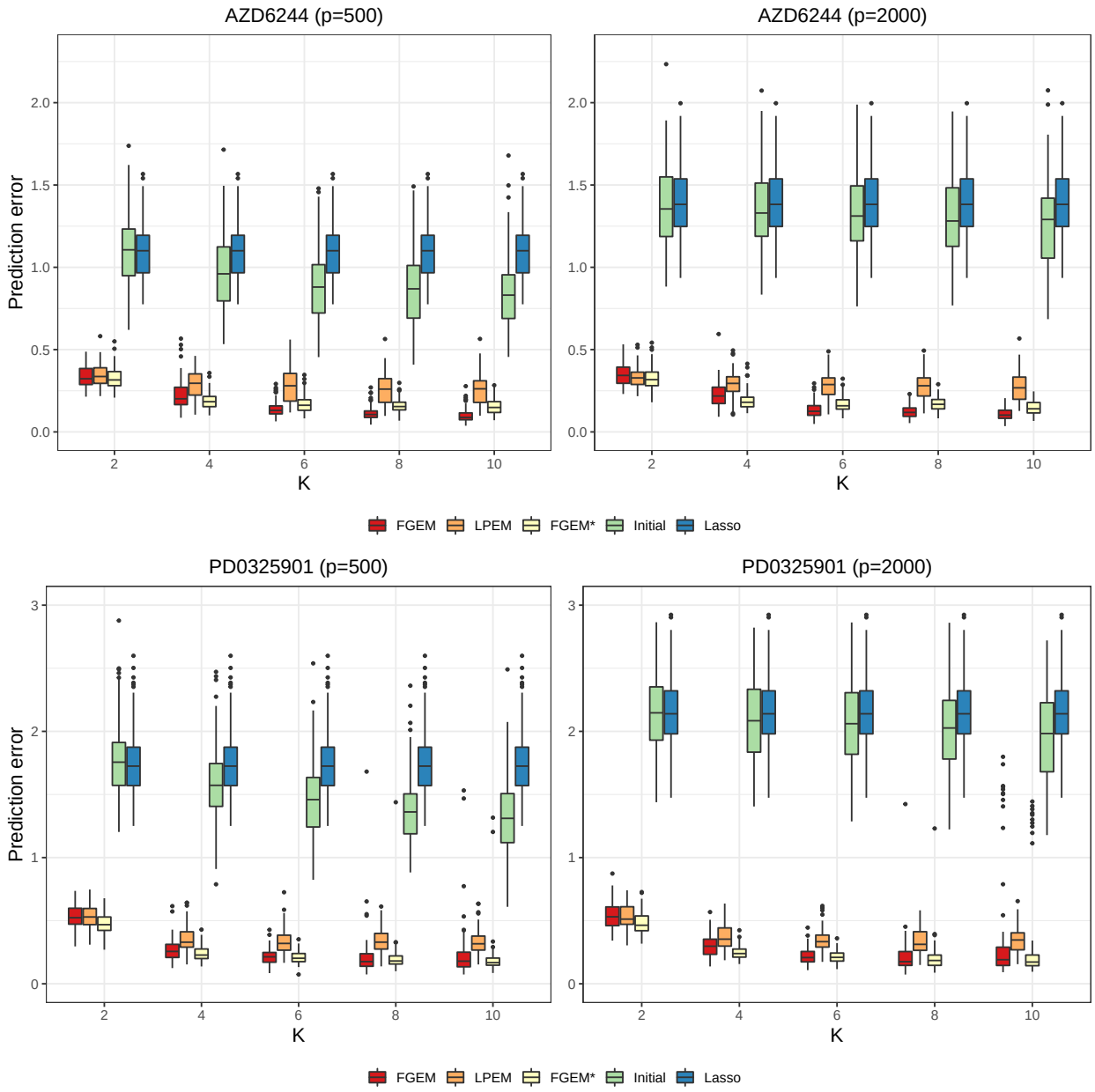

Figure 2. Mean prediction errors for two responses in the CCLE data set.

Model		p	Initial	GLLiM	LPEM	FGEM	GEM	FGEM*	Oracle
S1	**M1**	500	25.25 (0.93)	9.93 (0.003)	2.36 (0.46)	2.25 (0.04)	1.64 (0.36)	0.40 (0.01)	0.22 (0.003)
		1000	24.45 (0.82)	9.93 (0.003)	3.43 (0.53)	2.39 (0.03)	3.07 (0.49)	0.44 (0.01)	0.23 (0.004)
		2000	24.32 (0.86)	9.94 (0.003)	5.08 (0.57)	2.51 (0.03)	4.73 (0.77)	0.47 (0.01)	0.22 (0.004)
	M2	500	26.00 (4.23)	8.80 (0.01)	1.03 (0.21)	2.16 (0.03)	0.77 (0.15)	0.31 (0.01)	0.28 (0.06)
		1000	20.03 (0.82)	8.80 (0.003)	3.21 (0.36)	2.20 (0.03)	1.36 (0.26)	0.40 (0.01)	0.23 (0.004)
		2000	36.10 (9.88)	8.81 (0.002)	3.36 (0.43)	2.30 (0.03)	3.48 (0.61)	0.45 (0.01)	0.23 (0.004)
	M3	500	29.12 (1.03)	11.77 (0.002)	5.41 (0.70)	2.71 (0.04)	5.62 (0.71)	0.43 (0.01)	0.47 (0.18)
		1000	29.27 (1.00)	11.77 (0.002)	8.05 (0.71)	2.87 (0.04)	6.70 (0.69)	0.48 (0.01)	0.23 (0.004)
		2000	29.38 (1.05)	11.78 (0.003)	8.38 (0.63)	3.04 (0.03)	8.64 (0.88)	0.50 (0.01)	0.22 (0.004)
	M4	500	40.9 6(1.46)	13.83 (0.002)	2.81 (0.49)	4.26 (0.07)	0.81 (0.10)	0.72 (0.02)	0.66 (0.15)
		1000	41.95 (1.45)	13.83 (0.001)	5.03 (0.63)	4.53 (0.08)	1.05 (0.14)	0.91 (0.06)	0.79 (0.24)
		2000	43.53 (1.94)	13.85 (0.002)	7.39 (0.62)	4.81 (0.09)	1.49 (0.28)	1.08 (0.08)	0.51 (0.11)
	M5	500	8.82 (0.25)	6.18 (0.002)	0.36 (0.01)	1.43 (0.02)	0.46 (0.01)	0.33 (0.01)	0.22 (0.004)
		1000	9.09 (0.40)	6.19 (0.002)	0.38 (0.01)	1.48 (0.02)	0.50 (0.02)	0.35 (0.01)	0.24 (0.004)
		2000	9.19 (0.30)	6.22 (0.001)	0.39 (0.01)	1.57 (0.02)	0.64 (0.02)	0.33 (0.01)	0.23 (0.004)
S2	**M1**	500	153.00 (3.69)	9.96 (0.001)	4.00 (0.73)	2.64 (0.04)	1.59 (0.27)	0.30 (0.01)	1.31(0.79)
		1000	155.02 (3.13)	9.96 (0.001)	15.87 (1.00)	3.51(0.14)	1.95 (0.31)	0.47 (0.04)	0.10 (0.002)
		2000	112.43 (2.42)	9.96 (0.001)	18.65 (0.06)	2.86 (0.12)	2.76 (0.52)	0.46 (0.02)	0.08 (0.002)
	M2	500	113.01 (1.73)	8.77 (0.01)	11.05 (0.56)	2.64 (0.10)	1.10 (0.15)	0.49 (0.10)	0.20 (0.09)
		1000	75.51 (1.60)	8.82 (0.001)	16.30 (0.31)	2.24 (0.05)	2.19 (0.05)	0.32 (0.01)	0.15 (0.05)
		2000	219.84 (5.85)	8.82 (0.002)	25.47 (0.07)	2.51 (0.09)	3.46 (0.53)	0.28 (0.01)	0.06 (0.001)
	M3	500	157.49 (4.04)	11.80 (0.003)	18.82 (1.68)	5.05 (0.12)	4.54 (0.09)	0.45 (0.01)	0.20 (0.08)
		1000	311.70 (5.39)	11.79 (0.001)	11.29 (1.26)	5.76 (0.13)	9.86 (1.18)	0.61 (0.01)	0.10 (0.001)
		2000	72.49 (2.15)	11.80 (0.001)	19.74 (0.06)	3.37 (0.13)	17.42 (0.33)	0.52 (0.06)	0.09 (0.001)
	M4	500	228.76 (4.66)	13.83 (0.001)	19.30 (0.49)	6.48 (0.18)	1.49 (0.28)	1.10 (0.27)	3.47 (1.28)
		1000	249.24 (4.15)	13.82 (0.002)	25.24 (0.12)	11.78 (0.31)	14.10 (0.53)	10.27 (0.99)	0.60 (0.18)
		2000	196.12 (4.02)	13.84 (0.001)	23.11 (0.08)	12.54 (0.30)	11.62 (0.96)	17.74 (0.97)	0.58 (0.19)
	M5	500	149.78 (2.96)	6.14 (0.001)	0.77 (0.16)	2.80 (0.05)	0.36 (0.01)	0.47 (0.01)	0.11 (0.002)
		1000	149.08 (1.98)	6.15 (0.001)	2.48 (0.40)	3.54 (0.06)	4.06 (0.63)	0.83 (0.07)	0.13 (0.03)
		2000	154.65 (1.93)	6.25 (0.007)	8.79 (0.24)	3.45 (0.07)	3.26 (0.66)	1.19 (0.14)	0.15 (0.05)
S3	**M1**	500	161.79 (86.14)	9.94 (0.002)	5.20 (0.87)	2.87 (0.09)	6.13 (0.64)	0.55 (0.08)	0.23 (0.003)
		1000	76.74 (3.13)	9.93 (0.004)	6.51 (0.85)	3.24 (0.14)	7.14 (0.60)	1.02 (0.24)	0.24 (0.004)
		2000	66.98 (2.42)	9.93 (0.003)	7.30 (0.78)	3.25 (0.11)	8.77 (0.62)	0.86 (0.02)	0.23 (0.004)
	M2	500	113.01 (1.73)	8.77 (0.01)	11.05 (0.56)	2.64 (0.10)	3.29 (0.43)	0.49 (0.10)	0.23 (0.002)
		1000	75.51 (1.60)	8.82 (0.004)	16.30 (0.31)	2.24 (0.05)	5.87 (0.44)	0.32 (0.01)	0.54 (0.20)
		2000	219.84 (5.85)	8.82 (0.003)	25.47 (0.07)	2.51 (0.09)	7.92 (0.45)	0.28 (0.01)	0.23 (0.004)
	M3	500	8.99 (0.19)	8.82 (0.001)	5.73 (1.17)	3.03 (0.14)	4.84 (0.89)	0.45 (0.08)	0.23 (0.003)
		1000	16.67 (5.38)	8.80 (0.002)	9.00 (1.06)	3.10 (0.14)	6.57 (0.90)	0.59 (0.09)	0.41 (0.17)
		2000	11.64 (2.10)	8.80 (0.003)	7.90 (0.97)	3.21 (0.12)	6.03 (1.13)	0.59 (0.09)	0.89 (0.66)
	M4	500	17.65 (3.10)	13.81 (0.002)	9.60 (1.06)	5.46 (0.11)	2.81 (0.45)	0.98 (0.11)	0.92 (0.25)
		1000	38.67 (20.74)	13.83 (0.002)	12.21 (0.95)	6.01 (0.16)	3.60 (0.48)	1.69 (0.31)	1.49 (0.35)
		2000	13.69 (0.04)	13.83 (0.001)	12.07 (0.79)	6.59 (0.15)	4.81 (0.75)	2.17 (0.31)	1.24 (0.31)
	M5	500	17.47 (3.10)	6.19 (0.007)	0.28 (0.01)	1.94 (0.04)	0.97 (0.16)	0.41 (0.05)	0.24 (0.01)
		1000	18.73 (3.38)	6.26 (0.002)	0.50 (0.20)	2.14 (0.04)	1.47 (0.23)	0.42 (0.04)	0.25 (0.04)
		2000	12.58 (1.77)	6.28 (0.002)	0.72 (0.43)	2.18 (0.05)	1.52 (0.31)	0.41 (0.04)	0.23 (0.01)

Table 1

Averages and standard errors of estimation error $\Delta_{\boldsymbol{\beta}}$ based on 100 replicates. When $p = 2000$, the results for GEM are based on 50 replicates.

Model		Method	$p = 500$ TPR	$p = 500$ FPR	$p = 1000$ TPR	$p = 1000$ FPR	$p = 2000$ TPR	$p = 2000$ FPR
S1	**M1**	Initial	0.89 (0.01)	0.00 (0.00)	0.87 (0.01)	0.00 (0.00)	0.85 (0.02)	0.00 (0.00)
		LPEM	1.00 (0.001)	0.11 (0.03)	1.00 (0.00)	0.13 (0.02)	1.00 (0.00)	0.13 (0.02)
		GEM	1.00 (0.00)	0.06 (0.02)	1.00 (0.00)	0.09 (0.02)	1.00 (0.002)	0.10 (0.02)
		FGEM	1.00 (0.00)	0.04 (0.002)	1.00 (0.00)	0.03 (0.001)	1.00 (0.00)	0.02 (0.001)
	M2	Initial	0.53 (0.01)	0.00 (0.00)	0.51 (0.01)	0.00 (0.00)	0.49 (0.01)	0.00 (0.00)
		LPEM	1.00 (0.002)	0.04 (0.02)	0.98 (0.005)	0.08 (0.02)	0.95 (0.01)	0.09 (0.01)
		GEM	1.00 (0.001)	0.02 (0.00)	0.99 (0.004)	0.06 (0.01)	0.95 (0.01)	0.08 (0.02)
		FGEM	1.00 (0.00)	0.02 (0.001)	1.00 (0.00)	0.02 (0.001)	1.00 (0.00)	0.01 (0.001*)
	M3	Initial	0.84 (0.01)	0.00 (0.001*)	0.84 (0.01)	0.00 (0.001*)	0.81 (0.01)	0.00 (0.001*)
		LPEM	1.00 (0.002)	0.24 (0.04)	0.99 (0.004)	0.30 (0.03)	0.98 (0.004)	0.21 (0.02)
		GEM	1.00 (0.001)	0.25 (0.04)	0.99 (0.003)	0.22 (0.03)	0.98 (0.007)	0.19 (0.02)
		FGEM	1.00 (0.00)	0.05 (0.002)	1.00 (0.00)	0.03 (0.001)	1.00 (0.00)	0.02 (0.001)
	M4	Initial	0.99 (0.003)	0.00 (0.001*)	0.99 (0.003)	0.00 (0.001*)	0.98 (0.004)	0.00 (0.001*)
		LPEM	1.00 (0.00)	0.09 (0.02)	1.00 (0.00)	0.14 (0.02)	1.00 (0.00)	0.15 (0.02)
		GEM	1.00 (0.00)	0.09 (0.06)	1.00 (0.00)	0.11 (0.05)	1.00 (0.00)	0.10 (0.05)
		FGEM	1.00 (0.00)	0.05 (0.003)	1.00 (0.00)	0.04 (0.002)	1.00 (0.00)	0.02 (0.002)
	M5	Initial	0.50 (0.001*)	0.00 (0.001*)	0.50 (0.001*)	0.00 (0.001*)	0.50 (0.001)	0.00 (0.00)
		LPEM	1.00 (0.00)	0.00 (0.001*)	1.00 (0.00)	0.00 (0.001*)	1.00 (0.00)	0.00 (0.001*)
		GEM	1.00 (0.00)	0.04 (0.003)	1.00 (0.00)	0.00 (0.001*)	1.00 (0.00)	0.00 (0.00)
		FGEM	1.00 (0.00)	0.02 (0.001)	1.00 (0.00)	0.01 (0.001)	1.00 (0.00)	0.00 (0.001*)
S2	**M1**	Initial	0.37 (0.01)	0.00 (0.001*)	0.52 (0.01)	0.00 (0.00)	0.48 (0.02)	0.00 (0.001*)
		LPEM	0.97 (0.01)	0.20 (0.03)	0.71 (0.02)	0.47 (0.03)	0.58 (0.01)	0.41 (0.001)
		GEM	1.00 (0.00)	0.05 (0.02)	1.00 (0.00)	0.04 (0.01)	1.00 (0.00)	0.05 (0.01)
		FGEM	1.00 (0.00)	0.04 (0.002)	1.00 (0.001)	0.04 (0.004)	1.00 (0.00)	0.02 (0.002)
	M2	Initial	0.49 (0.01)	0.01 (0.001)	0.48 (0.004)	0.00 (0.00)	0.48 (0.01)	0.00 (0.00)
		LPEM	0.85 (0.01)	0.63 (0.04)	0.58 (0.01)	0.61 (0.01)	0.27 (0.01)	0.44 (0.001*)
		GEM	1.00 (0.003)	0.02 (0.01)	0.97 (0.008)	0.06 (0.01)	0.92 (0.02)	0.07 (0.01)
		FGEM	1.00 (0.002)	0.06 (0.004)	1.00 (0.00)	0.02 (0.002)	1.00 (0.00)	0.01 (0.001)
	M3	Initial	0.45 (0.01)	0.00 (0.00)	0.39 (0.01)	0.00 (0.00)	0.47 (0.01)	0.00 (0.00)
		LPEM	0.87 (0.01)	0.52 (0.04)	0.82 (0.02)	0.28 (0.03)	0.65 (0.01)	0.41 (0.001)
		GEM	1.00 (0.002)	0.11 (0.03)	0.99 (0.003)	0.27 (0.03)	0.99 (0.003)	0.40 (0.08)
		FGEM	1.00 (0.00)	0.10 (0.004)	1.00 (0.00)	0.08 (0.002)	1.00 (0.001)	0.03 (0.002)
	M4	Initial	0.69 (0.02)	0.01 (0.001)	0.78 (0.01)	0.00 (0.001*)	0.79 (0.01)	0.00 (0.001*)
		LPEM	0.93 (0.01)	0.78 (0.02)	0.76 (0.01)	0.64 (0.001)	0.77 (0.01)	0.40 (0.001)
		GEM	1.00 (0.00)	0.03 (0.01)	1.00 (0.00)	0.37 (0.02)	1.00 (0.00)	0.19 (0.02)
		FGEM	1.00 (0.00)	0.11 (0.01)	1.00 (0.00)	0.15 (0.01)	1.00 (0.00)	0.11 (0.004)
	M5	Initial	0.50 (0.001*)	0.02 (0.001)	0.46 (0.01)	0.01 (0.001*)	0.43 (0.005)	0.01 (0.001*)
		LPEM	1.00 (0.00)	0.04 (0.01)	1.00 (0.00)	0.12 (0.02)	0.97 (0.01)	0.34 (0.01)
		GEM	1.00 (0.00)	0.01 (0.001*)	1.00 (0.002)	0.15 (0.03)	1.00 (0.002)	0.10 (0.02)
		FGEM	1.00 (0.00)	0.09 (0.004)	1.00 (0.001)	0.10 (0.002)	1.00 (0.00)	0.07 (0.002)
S3	**M1**	Initial	0.84 (0.01)	0.00 (0.00)	0.83 (0.01)	0.00 (0.00)	0.80 (0.01)	0.00 (0.00)
		LPEM	0.99 (0.003)	0.21 (0.04)	0.98 (0.005)	0.21 (0.03)	0.96 (0.01)	0.17 (0.02)
		GEM	1.00 (0.001)	0.34 (0.04)	0.99 (0.003)	0.31 (0.03)	0.99 (0.004)	0.25 (0.02)
		FGEM	0.99 (0.003)	0.03 (0.003)	0.98 (0.004)	0.02 (0.002)	0.98 (0.004)	0.01 (0.001)
	M2	Initial	0.54 (0.02)	0.00 (0.00)	0.54 (0.02)	0.00 (0.00)	0.52 (0.02)	0.00 (0.00)
		LPEM	0.99 (0.004)	0.27 (0.06)	0.92 (0.01)	0.35 (0.04)	0.89 (0.02)	0.20 (0.03)
		GEM	0.98 (0.004)	0.21 (0.03)	0.93 (0.01)	0.28 (0.02)	0.82 (0.02)	0.21 (0.02)
		FGEM	1.00 (0.002)	0.03 (0.004)	1.00 (0.003)	0.03 (0.003)	1.00 (0.003)	0.02 (0.002)
	M3	Initial	0.84 (0.01)	0.00 (0.00)	0.84 (0.01)	0.00 (0.00)	0.83 (0.01)	0.00 (0.00)
		LPEM	0.99 (0.002)	0.23 (0.04)	0.98 (0.004)	0.24 (0.03)	0.97 (0.004)	0.18 (0.02)
		GEM	1.00 (0.003)	0.16 (0.03)	0.99 (0.004)	0.18 (0.03)	0.99 (0.005)	0.11 (0.02)
		FGEM	0.99 (0.002)	0.04 (0.003)	0.99 (0.003)	0.02 (0.002)	0.99 (0.004)	0.02 (0.002)
	M4	Initial	0.97 (0.005)	0.00 (0.00)	0.97 (0.01)	0.00 (0.00)	0.96 (0.01)	0.00 (0.00)
		LPEM	1.00 (0.002)	0.34 (0.04)	0.99 (0.003)	0.35 (0.03)	0.99 (0.003)	0.24 (0.02)
		GEM	1.00 (0.00)	0.12 (0.02)	1.00 (0.00)	0.11 (0.02)	1.00 (0.00)	0.09 (0.01)
		FGEM	0.99 (0.002)	0.05 (0.003)	0.99 (0.003)	0.04 (0.003)	0.99 (0.003)	0.03 (0.002)
	M5	Initial	0.43 (0.01)	0.00 (0.001)	0.38 (0.02)	0.00 (0.001*)	0.38 (0.02)	0.00 (0.001*)
		LPEM	1.00 (0.00)	0.00 (0.001)	1.00 (0.00)	0.01 (0.01)	0.99 (0.01)	0.02 (0.01)
		GEM	1.00 (0.002)	0.02 (0.01)	0.99 (0.01)	0.03 (0.01)	0.98 (0.01)	0.02 (0.01)
		FGEM	1.00 (0.003)	0.03 (0.002)	1.00 (0.002)	0.01 (0.001)	1.00 (0.00)	0.01 (0.002)

Table 2

Averages and standard errors of true positive rate (TPR) and false positive rate (FPR) based on 100 replicates.. In the standard errors, the numbers smaller than 5×10−4 are rounded as 0.001∗*. When* $p = 2000$*, the results for GEM are based on 50 replicates.*

		S1			**S2**			**S3**		
Model	p	LPEM	FGEM	GEM	LPEM	FGEM	GEM	LPEM	FGEM	GEM
	500	1.5 (0.1)	1.5 (0.1)	80.1 (2.1)	4.7 (0.3)	1.5 (0.1)	75.9 (2.7)	1.7 (0.2)	1.0 (0.1)	109.9 (3.3)
M1	1000	2.8 (0.2)	2.1 (0.1)	580.3 (18.9)	26.0 (1.2)	6.7 (0.2)	544.2 (18.0)	7.7 (0.7)	5.7 (0.3)	754.5 (15.3)
	2000	13.7 (0.8)	15.3 (0.4)	3025.9 (137.2)	52.1 (0.5)	17.8 (0.4)	2692.4 (128.0)	17.0 (1.4)	14.5 (0.3)	3860.1 (70.6)
	500	2.3 (0.1)	4.6 (0.3)	147.0 (5.8)	6.5 (0.3)	1.7 (0.1)	162.1 (5.1)	1.8 (0.2)	1.9 (0.2)	207.3 (5.6)
M2	1000	2.2 (0.1)	4.2 (0.3)	622.1 (22.5)	11.9 (0.2)	3.3 (0.1)	600.9 (17.9)	4.7 (0.4)	3.5 (0.3)	758.1 (14.4)
	2000	11.1 (0.7)	21.6 (1.0)	2840.2 (149.5)	68.9 (0.6)	16.5 (0.3)	2546.3 (101.4)	17.7 (1.7)	17.5 (0.9)	3132.9 (61.4)
	500	4.0 (0.2)	3.1 (0.2)	75.6 (4.2)	17.3 (0.6)	2.6 (0.1)	82.5 (4.3)	3.6 (0.3)	2.2 (0.1)	57.4 (4.2)
M3	1000	10.2 (0.6)	6.7 (0.3)	464.3 (13.5)	23.8 (1.6)	6.9 (0.1)	589.1 (19.5)	9.4 (0.9)	6.0 (0.3)	497.4 (28.7)
	2000	18.9 (1.0)	16.8 (0.5)	156.5 (50.4)	45.6 (0.2)	17.6 (0.3)	1737.7 (17.8)	18.1 (1.5)	16.5 (0.5)	1025.2 (91.7)
	500	3.3 (0.1)	2.3 (0.1)	138.5 (6.2)	22.4 (0.4)	5.9 (0.2)	89.8 (3.4)	9.4 (0.5)	5.0 (0.2)	98.8 (3.3)
M4	1000	12.4 (0.4)	9.5 (0.2)	737.5 (37.3)	21.0 (0.2)	5.8 (0.2)	708.4 (16.4)	20.9 (0.8)	9.5 (0.2)	705.9 (27.1)
	2000	27.5 (0.8)	24.3 (0.4)	2089.8 (51.6)	75.5 (0.7)	35.0 (0.9)	2422.4 (45.2)	37.0 (1.2)	20.7 (0.3)	2125.7 (67.9)
	500	1.5 (0.1)	3.7 (0.3)	48.9 (1.8)	2.0 (0.1)	1.6 (0.1)	67.7 (3.7)	0.6 (0.1)	1.0 (0.1)	61.1 (3.6)
M5	1000	1.3 (0.1)	3.6 (0.2)	200.8 (7.5)	5.6 (0.4)	5.8 (0.2)	151.8 (8.9)	1.4 (0.2)	3.1 (0.3)	198.8 (9.3)
	2000	3.3 (0.1)	11.9 (0.6)	1135.5 (55.4)	50.2 (1.2)	34.8 (1.5)	1297.6 (55.4)	2.5 (0.4)	7.2 (0.6)	987.4 (65.9)

Table 3
Averages and standard errors of computational time (s).

Model	p	Initial	GLLiM	LPEM	GEM	FGEM	FGEM*	LPEM*	GEM*	LPEM†	GEM†	Oracle
A1	500	26.42 (1.07)	9.92 (0.002)	3.63 (0.57)	3.33 (0.54)	2.17 (0.03)	0.39 (0.01)	1.05 (0.16)	0.96 (0.16)	0.44 (0.01)	0.36 (0.01)	0.22 (0.005)
	1000	23.79 (0.67)	9.92 (0.00)	4.06 (0.57)	4.17 (0.56)	2.36 (0.03)	0.45 (0.01)	1.87 (0.27)	1.72 (0.25)	0.45 (0.004)	0.41 (0.006)	0.23 (0.004)
	2000	24.32 (0.73)	9.94 (0.003)	5.40 (0.57)	6.49 (0.82)	2.49 (0.03)	0.45 (0.01)	2.52 (0.28)	3.07 (0.40)	0.47 (0.01)	0.46 (0.004)	0.23 (0.004)
A2	500	26.47 (0.74)	9.91 (0.001)	3.37 (0.57)	2.55 (0.49)	2.27 (0.03)	0.38 (0.01)	1.04 (0.16)	0.66 (0.10)	0.43 (0.005)	0.35 (0.005)	0.22 (0.004)
	1000	26.19 (0.74)	9.92 (0.003)	4.13 (0.58)	3.63 (0.55)	2.41 (0.03)	0.44 (0.01)	1.79 (0.25)	1.50 (0.24)	0.44 (0.005)	0.40 (0.01)	0.33 (0.11)
	2000	26.21 (0.73)	9.93 (0.003)	5.30 (0.58)	5.33 (0.76)	2.51 (0.03)	0.46 (0.01)	2.36 (0.27)	2.96 (0.45)	0.43 (0.004)	0.42 (0.01)	0.22 (0.004)

Table 4

Averages and standard errors of estimation error $\Delta_{\boldsymbol{\beta}}$ based on 100 replicates. The results for GEM with $p = 2000$ are based on 10 replicates. Similar to FGEM*, LPEM* and GEM* represent the EM algorithm refit on the selected variables from LPEM and GEM, respectively. LPEM† and GEM† represent LPEM and GEM using the output of FGEM as initialization, respectively.