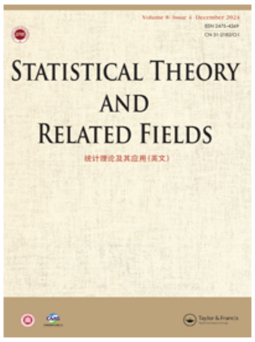

STATISTICAL THEORY AND RELATED FIELDS

Statistical Theory and Related Fields

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tstf20

L_0-regularized high-dimensional sparse multiplicative models

Hao Ming, Hu Yang & Xiaochao Xia

To cite this article: Hao Ming, Hu Yang & Xiaochao Xia (16 Feb 2025): L_0-regularized high-dimensional sparse multiplicative models, Statistical Theory and Related Fields, DOI: 10.1080/24754269.2025.2460148

To link to this article: https://doi.org/10.1080/24754269.2025.2460148

Published online: 16 Feb 2025.

STATISTICAL THEORY AND RELATED FIELDS
https://doi.org/10.1080/24754269.2025.2460148

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

L_0-regularized high-dimensional sparse multiplicative models

Hao Ming, Hu Yang and Xiaochao Xia

College of Mathematics and Statistics, Chongqing University, Chongqing, People's Republic of China

ABSTRACT

In this paper, we study high-dimensional sparse multiplicative models for positive response data and propose a variable sorted active set (VSAS) algorithm for finding the L_0 regularized least product relative error (LPRE) estimator. The VSAS algorithm is derived from the local quadratic approximation based on the Karush-Kuhn-Tucker (KKT) conditions of L_0-penalized LPRE objective function. Under the condition of restricted invertibility, we establish an explicit L_∞ upper bound for the sequence of solutions generated by the VSAS algorithm. We further obtain an optimal convergence rate for the proposed estimator with high probability in finite iterations. In addition, our estimator enjoys the oracle property with high probability if the target signal exceeds the detectable level. Finally, extensive simulations and two real-world applications are conducted to illustrate the effectiveness of the proposal.

ARTICLE HISTORY
Received 27 March 2024
Revised 26 August 2024
Accepted 18 January 2025

KEYWORDS
Multiplicative models; L_0 penalty; least product relative error; KKT conditions; oracle property

1. Introduction

The data with positive response variables, such as wages, survival time, duration, stock prices, etc., are often encountered in many real-world applications. We usually collect such types of data, $\{(y_i, \boldsymbol{X}_i^\top), i = 1, 2, \dots, n\}$, where y_i is a univariate positive response variable, and $\boldsymbol{X}_i \in \mathbb{R}^{p_n}$ is the column vector of p_n predictors for the ith observation. Here, p_n denotes the number of predictors, which is allowed to diverge as the sample size n increases. Denote $\mathbb{Y} = (y_1, y_2, \dots, y_n)^\top \in \mathbb{R}^n$ and $\mathbb{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_n)^\top \in \mathbb{R}^{n \times p_n}$. We assume that $y_i, i = 1, \dots, n$, are independent and generated from a continuous probability distribution. To describe the relationship between the positive response y_i and the predictors $\boldsymbol{X}_i$, we consider the following multiplicative model:

$$y_i = \exp(\boldsymbol{X}_i^\top \boldsymbol{\beta}^*)\varepsilon_i, \quad i = 1, 2, \dots, n, \tag{1}$$

where $\boldsymbol{\beta}^* = (\beta_1^*, \beta_2^*, \dots, \beta_{p_n}^*)^\top \in \mathbb{R}^{p_n}$ is the true parameter vector with $q(< n)$ non-zero elements and ε_i is a positive random error. Without loss of generality, we exclude the intercept in model (1), which can be achieved by scaling both $\mathbb{Y}$ and each column of $\mathbb{X}$ with a column norm being $\sqrt{n}$. Clearly, applying a logarithmic transformation to both sides of model (1) results in the following log-linear model:

$$\log(y_i) = \boldsymbol{X}_i^\top \boldsymbol{\beta}^* + \log(\varepsilon_i), \quad i = 1, 2, \dots, n. \tag{2}$$

However, if one directly applies the least squares (LS) approach to model (2), the resulting estimator of the parameter vector $\boldsymbol{\beta}^*$ might not be the most efficient either under $E(\log(\varepsilon_i)) \neq 0$ or when $\log(\varepsilon_i)$ is far from a sub-Gaussian distribution. In this paper, we focus on the situation where ε_i satisfies $E(\varepsilon_i) = E(\varepsilon_i^{-1})$ (see Section 4 for more detailed discussion), which implies that the least product relative error (LPRE) approach proposed by K. Chen et al. (2016) is the most efficient for model (1). The LPRE loss is formulated as

$$\begin{aligned} Q(\boldsymbol{\beta}) &= \frac{1}{n}\sum_{i=1}^{n}\left\{\left|\frac{y_i - \exp(\boldsymbol{X}_i^\top\boldsymbol{\beta})}{y_i}\right| \times \left|\frac{y_i - \exp(\boldsymbol{X}_i^\top\boldsymbol{\beta})}{\exp(\boldsymbol{X}_i^\top\boldsymbol{\beta})}\right|\right\} \\ &= \frac{1}{n}\sum_{i=1}^{n}\left\{y_i^{-1}\exp(\boldsymbol{X}_i^\top\boldsymbol{\beta}) + y_i\exp(-\boldsymbol{X}_i^\top\boldsymbol{\beta}) - 2\right\} \\ &=: L(\boldsymbol{\beta}) - 2. \end{aligned} \tag{3}$$

It can be seen that the LPRE loss consists of two types of relative errors, $|y_i\exp(-\boldsymbol{X}_i^\top\boldsymbol{\beta}) - 1|$ and $|y_i^{-1}\exp(\boldsymbol{X}_i^\top\boldsymbol{\beta}) - 1|$, relative to the regression function and the response value. This loss has three advantages:

CONTACT Xiaochao Xia xxc@cqu.edu.cn College of Mathematics and Statistics, Chongqing University, Chongqing401331, People's Republic of China

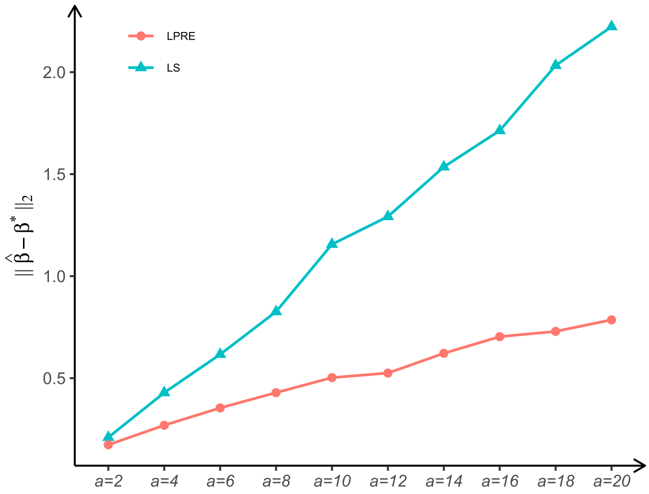

Figure 1. When $\log(\varepsilon_i) \sim \text{Uniform}(-a, a)$, the estimates obtained by LS and LPRE methods for solving the log-linear model (2) with 100 replications, where $\mathbb{X}$ is from Case (*i*) of Section 4.3, and $n = 200, p = q = 5, \rho = 0.5, \boldsymbol{\beta}^* = (1, 2, 3, 4, 5)^\top$.

(i) The LPRE criterion, which minimizes the above LPRE loss, is scale-free since it does not require each individual of the response to have a unified measurement unit. This may be crucial in some applications, such as modelling the stock price data due to non-comparable stock prices among different listed companies.

(ii) The LPRE loss serves as strictly convex and infinitely differentiable. Thus, the optimal solution satisfies the first-order condition and can be obtained easily through Newton-Raphson iteration.

(iii) The LPRE loss serves as a minimization function for modelling the error distribution that satisfies $E(\varepsilon_i) = E(\varepsilon_i^{-1})$, and it can also be used to estimate ordinary linear models by exponentiating the response variable. Figure 1 shows that the estimates obtained using the LPRE loss are significantly superior to those obtained using the LS loss under the log-uniform error distribution.

In recent years, the LPRE loss has been extensively investigated in the literature. For example, K. Chen et al. (2016) proved that under certain conditions, the LPRE estimator, which is the minimizer of the LPRE loss, performs more efficiently than the least absolute relative error (LARE)-based estimator proposed by K. Chen et al. (2010), the least squares (LS)-based estimator as well as the least absolute deviation (LAD)-based estimators, both of which apply the logarithm transform to the response. Z. Wang et al. (2015) developed a nonparametric LPRE approach to detect and estimate the change point in multiplicative regression models. Hao et al. (2016) investigated the variable selection problem based on a regularized LPRE loss for multiplicative models in fixed dimension and divergent dimension, respectively, and designed an alternating direction method of multipliers (ADMM) algorithm for computing the solution path effectively. Liu and Xia (2018) studied single-index multiplicative models and proposed a local kernel weighted LPRE method. Zhang et al. (2018) and Zhang et al. (2019) considered the estimation and hypothesis testing in partial linear multiplicative models and single-index multiplicative models, respectively. Hu (2019) studied the LPRE-based estimation for the varying-coefficient multiplicative regression model with kernel smoothing techniques. More recently, Zhang et al. (2022) proposed a kernel density-based estimation for multiplicative linear regression models. Y. Chen et al. (2022) developed a new method to fit single-index varying-coefficient multiplicative models on the basis of the LPRE and local kernel smoothing techniques. Ming et al. (2022) studied the identification and estimation of nonparametric functions in multiplicative additive models through employing the LPRE and smoothly clipped absolute deviation (SCAD) penalty of Fan and Li (2001).

However, all of the existing literature related to multiplicative models focuses merely on low-dimensional or moderate-dimensional setting, that is, $p_n < n$. In contrast, it is very likely to encounter the situations where the dimension p_n far outstrips the sample size, i.e. $p_n \gg n$, in many high-dimensional applications, such as modelling gene expression data. This motivates us to study the high-dimensional sparse multiplicative models. To this end, one may consider the following regularized LPRE loss

$$L_\lambda(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} \left\{ y_i^{-1} \exp(\boldsymbol{X}_i^\top \boldsymbol{\beta}) + y_i \exp(-\boldsymbol{X}_i^\top \boldsymbol{\beta}) \right\} + \sum_{j=1}^{p_n} p_\lambda(|\beta_j|), \tag{4}$$

where $p_\lambda(\cdot)$ is a penalty function that relies on the tuning parameter λ. Regarding the penalty function, there are several popular choices such as the least absolute shrinkage and selection operator (LASSO, Tibshirani (1996)), the SCAD penalty, and the minimum concave penalty (MCP, Zhang (2010)). For linear models, Fan et al. (2014), Shi et al. (2020) and Huang, Jiao, Lu, et al. (2022) proposed some efficient algorithms for solving LASSO, SCAD and MCP penalized least squares. Furthermore, since the classical Newton-Raphson algorithm cannot be applied directly to LASSO, SCAD and MCP, Cao et al. (2023) proposed a cubic Hermite interpolation penalty (CHIP). Nevertheless, the LASSO method is generally biased. The SCAD, MCP and CHIP methods enjoy nice properties, but still require a minimum signal strength to achieve support recovery. Moreover, the parameter estimators corresponding to these penalty functions strictly depend on whether $\mathbb{X}$ is normalized. Therefore, the L_0 regularization is more preferable to some researchers. For instance, Huang et al. (2018), Huang, Jiao, Kang, et al. (2021), Zhu et al. (2020), Do et al. (2020), Zhou et al. (2021), P. Li et al. (2022), Y. Zhang et al. (2023) and Ming and Yang (2024a) examined the L_0-regularized linear model. Wen et al. (2020), Huang, Jiao, Kang, et al. (2022) and Ming and Yang (2024b) examined the L_0-regularized logistic regression model. X. Li et al. (2022) investigated the index tracking problem using L_0 regularization. Zheng et al. (2022) investigated L_0 regularized learning for high-dimensional additive hazards regression. Wen, Li, et al. (2023) and Wen, Wang, et al. (2023) investigated the L_0-regularized multinomial logistic regression model and trend filtering model, respectively.

Although (Y. Chen et al., 2024) investigated non-convex penalized ADMM algorithms for high-dimensional multiplicative models (4) based on local linear approximation, the requirement to compute $p_n \times p_n$-dimensional inverse matrices results in a significantly high computational cost. In this paper, we consider the more efficient, sparser minimization problem of

$$L_\lambda(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\left\{y_i^{-1}\exp(\boldsymbol{X}_i^\top\boldsymbol{\beta}) + y_i\exp(-\boldsymbol{X}_i^\top\boldsymbol{\beta})\right\} + \lambda\|\boldsymbol{\beta}\|_0. \tag{5}$$

It is well known that solving (5) is an NP-hard problem (X. Chen et al., 2014; Natarajan, 1995). It is extremely difficult, even infeasible to find an exact solution to the minimization of (5). Inspired by Wen et al. (2020), we propose an approximate algorithm named the variable sorted active set (VSAS) to minimize (5). If one takes the logarithm on both sides of model (1), i.e. model (2), the support detection and root finding (SDAR, Huang et al. (2018)) algorithm is applicable, which can be viewed as a special case of the VSAS algorithm. Note that our VSAS algorithm can also be applicable to the L_0 regularization problem with convex, second-order differentiable loss functions, such as logistic regression model and Cox's proportional hazards model considered by Wen et al. (2020). Compared to the generalized SDAR (GSDAR, Huang, Jiao, Kang, et al. (2022)) algorithm that only uses the first derivative of loss function as the direction of descent, our VSAS uses the first two derivatives of loss function to make the convergence of the algorithm faster. More specifically, our VSAS algorithm is an iterated variable sorted active set algorithm, which is rooted in the local quadratic approximation based on the KKT conditions. We call the minimizer, $\hat{\boldsymbol{\beta}}$, of (5) achieved by the VSAS algorithm the L_0-LPRE estimator.

The contribution of this paper is summarized as follows,

(i) We propose an L_0-regularized LPRE estimator for high-dimensional multiplicative models. To find the estimator, we derive a fast VSAS algorithm to handle high-dimensional data with positive response. The algorithm is rooted from the local quadratic approximation based on the KKT conditions, and it is applicable to the L_0 regularization problem as long as the loss function is convex and twice differentiable.
(ii) We establish an L_∞ upper bound for the sequence of solutions generated by VSAS algorithm, and achieve an optimal convergence rate in finite iterations.
(iii) We propose an adaptive VSAS algorithm to select the optimal tuning parameters and illustrate the superiority of the L_0-LPRE over existing methods by extensive simulations and two real-world applications.

The paper is structured as follows. Section 2 presents the detailed methodology of the VASA algorithm. Section 3 introduces some regularity conditions and theoretical properties of the L_0-LPRE estimator. Sections 4 and 5 present simulation studies and applications. Finally, Section 6 provides a summary, while the proofs of theoretical results are included in the Appendix.

2. Methodology

In this section, we will first introduce the procedure of VSAS algorithm (Algorithms 1 and 2) for finding L_0 regularized LPRE estimator. For simplicity of notation, we use $[a]$ to denote the set of $\{1, 2, \ldots, a\}$ for any positive integer a, $\mathrm{supp}(\boldsymbol{\beta}) = \{j \in [p_n] | \beta_j \neq 0\}$ to denote the support of $\boldsymbol{\beta}$, and $|A|$ to denote the size of set A. Let $\boldsymbol{\beta}_A = (\beta_j, j \in A) \in \mathbb{R}^{|A|}$ and $\boldsymbol{\beta}|_A \in \mathbb{R}^{p_n}$ with its jth element being $(\boldsymbol{\beta}|_A)_j = \beta_j \mathbb{I}\{j \in A\}$, where $\mathbb{I}\{\cdot\}$ means an indicator function.

Let $\hat{A} = \mathrm{supp}(\hat{\boldsymbol{\beta}})$ and $\hat{I} = \hat{A}^c$. According to Lemma 3.1, we can obtain two sets $\hat{A}$ and :

$$\begin{aligned} \hat{A} &= \left\{ j \in [p_n] | \sqrt{\hat{g}_j} |\hat{\beta}_j + \hat{d}_j| \geq \sqrt{2\lambda} \right\}, \\ \hat{I} &= \left\{ j \in [p_n] | \sqrt{\hat{g}_j} |\hat{\beta}_j + \hat{d}_j| < \sqrt{2\lambda} \right\}, \end{aligned} \tag{6}$$

where the definitions of $\hat{g}_j$, $\hat{\beta}_j$, $\hat{d}_j$ and λ are given in Lemma 3.1. As a consequence, we have a set of equations:

$$\begin{cases} \hat{\boldsymbol{\beta}}_{\hat{I}} = \mathbf{0}, \\ \hat{\boldsymbol{d}}_{\hat{A}} = \mathbf{0}, \\ \hat{\boldsymbol{\beta}}_{\hat{A}} \in \underset{\boldsymbol{\beta}_{\hat{A}}}{\mathrm{argmin}} L(\boldsymbol{\beta}_{\hat{A}}), \\ \hat{d}_j = -\left(\dfrac{\partial^2 L(\hat{\boldsymbol{\beta}}_{-j}, \beta_j)}{\partial^2 \beta_j} |_{\hat{\beta}_j} \right)^{-1} \dfrac{\partial L(\hat{\boldsymbol{\beta}}_{-j}, \beta_j)}{\partial \beta_j} |_{\hat{\beta}_j}, \quad j \in \hat{I}, \\ \hat{g}_j = \dfrac{\partial^2 L(\hat{\boldsymbol{\beta}}_{-j}, \beta_j)}{\partial^2 \beta_j} |_{\hat{\beta}_j}, \quad j \in [p_n], \end{cases} \tag{7}$$

where $\hat{\boldsymbol{\beta}}_{-j} = \hat{\boldsymbol{\beta}}_{[p_n] \backslash j}$ and

$$L(\boldsymbol{\beta}_{\hat{A}}) = \frac{1}{n} \sum_{i=1}^{n} \left\{ y_i^{-1} \exp(\boldsymbol{X}_{i\hat{A}}^{\top} \boldsymbol{\beta}_{\hat{A}}) + y_i \exp(-\boldsymbol{X}_{i\hat{A}}^{\top} \boldsymbol{\beta}_{\hat{A}}) \right\}.$$

More specifically, assume that $(\boldsymbol{\beta}^{(k)}, \boldsymbol{d}^{(k)}, \boldsymbol{g}^{(k)})$ are the outputs in the kth iteration. We can update approximation pair $(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{d}^{(k+1)}, \boldsymbol{g}^{(k+1)})$ by (7). Following Huang, Jiao, Kang, et al. (2021) and Ming and Yang (2024b), we introduce a step size $\tau \in (0, 1]$ to balance primal variable and dual variable. This condition is the weakest requirement for the design matrix and is necessary and sufficient for model identifiability. If our goal is to achieve a T-sparse solution, we can set

$$\sqrt{2\lambda^{(k)}} \triangleq \left\| \sqrt{\boldsymbol{g}^{(k)}} \cdot |\boldsymbol{\beta}^{(k)} + \tau \boldsymbol{d}^{(k)}| \right\|_{(T)}, \tag{8}$$

where $\|\boldsymbol{x}\|_{(T)}$ represents the Tth largest elements of $\boldsymbol{x}$ based on absolute value, and $\boldsymbol{a} \cdot \boldsymbol{b}$ denotes the componentwise product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. The detailed algorithm is given in Algorithms 1 and 2.

Algorithm 1 Newton iterative algorithm for LPRE estimator

```
Input: Data $(\mathbb{Y}, \mathbb{X})$, an initial estimator $\boldsymbol{\beta}^{(0)}$, an active set $A$, tolerance $\epsilon$, and maximum number of iterations $K$;
for $k = 0, 1, \ldots, K$ do
    Calculate $\nabla L(\boldsymbol{\beta}_A^{(k)}) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{X}_{iA} \{y_i^{-1} \exp(\boldsymbol{X}_{iA}^{\top} \boldsymbol{\beta}_A^{(k)}) - y_i \exp(-\boldsymbol{X}_{iA}^{\top} \boldsymbol{\beta}_A^{(k)})\}$ and $\nabla L^2(\boldsymbol{\beta}_A^{(k)}) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{X}_{iA}$
    $\boldsymbol{X}_{iA}^{\top} \{y_i^{-1} \exp(\boldsymbol{X}_{iA}^{\top} \boldsymbol{\beta}_A^{(k)}) + y_i \exp(-\boldsymbol{X}_{iA}^{\top} \boldsymbol{\beta}_A^{(k)})\}$;
    Update $\boldsymbol{\beta}_A^{(k+1)}$ by $\boldsymbol{\beta}_A^{(k+1)} = \boldsymbol{\beta}_A^{(k)} - (\nabla^2 L(\boldsymbol{\beta}_A^{(k)}))^{-1} \nabla L(\boldsymbol{\beta}_A^{(k)})$;
    if $\|\boldsymbol{\beta}_A^{(k+1)} - \boldsymbol{\beta}_A^{(k)}\|_2 \leq \epsilon$ then
        Stop and denote the last update by $\boldsymbol{\beta}_A^{(k+1)}$;
    end if
end for
Output: $\hat{\boldsymbol{\beta}}_A = \boldsymbol{\beta}_A^{(k+1)}$ as the estimator of $\boldsymbol{\beta}_A^*$.
```

Remark 2.1: In Algorithms 1 and 2, we set $\boldsymbol{\beta}^{(0)} = \mathbf{0}$ and terminate the iteration if the estimation error is less than a given threshold ϵ such as $\epsilon = 10^{-4}$, or if the active set remains unchanged. It can be observed that the output $\hat{\boldsymbol{\beta}}$ in Algorithm 2 becomes the oracle estimator when $\hat{A} = A^*$.

Remark 2.2: The algorithms of VSAS, SDAR, GSDAR and ESDAR (i.e. the enhanced support detection and root finding approach in Huang, Jiao, Kang, et al. (2021)) have some similarities and differences. For instance, SDAR is a special case of VSAS, which corresponds to the L_0 regularized least squares estimator after taking the logarithm on both sides of model (1). ESDAR is utilized to maintain balance between the primal and dual variables using

Algorithm 2 Variable sorted active set for L_0-LPRE estimator

```
Input: $\boldsymbol{\beta}^{(0)}, \boldsymbol{d}^{(0)}, \boldsymbol{g}^{(0)}$, a step size $\tau$, an integer $T$, and maximum number of iterations $K$;
for $k = 0, 1, \ldots, K$ do
    Set $A^{(k)} = \{j \in [p_n] | \sqrt{g_j^{(k)}} |\beta_j^{(k)} + \tau d_j^{(k)}| \ge \|\sqrt{\boldsymbol{g}^{(k)}} \cdot (\boldsymbol{\beta}^{(k)} + \tau \boldsymbol{d}^{(k)})\|_{(T)}\}$ and $I^{(k)} = (A^{(k)})^c$;
    $\boldsymbol{\beta}_{I^{(k)}}^{(k+1)} = \mathbf{0}$;
    $\boldsymbol{d}_{A^{(k)}}^{(k+1)} = \mathbf{0}$;
    Update $\boldsymbol{\beta}_{A^{(k)}}^{(k+1)}$ by Algorithm 1 with $A = A^{(k)}$;
    $g_j^{(k+1)} = \frac{\partial^2 L(\boldsymbol{\beta})}{\partial^2 \beta_j}|_{\boldsymbol{\beta}^{(k+1)}}, j \in [p_n]$;
    $d_j^{(k+1)} = -(g_j^{(k+1)})^{-1} \frac{\partial L(\boldsymbol{\beta})}{\partial \beta_j}|_{\boldsymbol{\beta}^{(k+1)}}, j \in I^{(k)}$;
    if $A^{(k+1)} = A^{(k)}$ then
        Stop and denote the last updates by $(\boldsymbol{\beta}_{\hat{A}}, \boldsymbol{\beta}_{\hat{I}})$;
    end if
end for
Output: $\hat{\boldsymbol{\beta}} = (\boldsymbol{\beta}_{\hat{A}}^\top, \boldsymbol{\beta}_{\hat{I}}^\top)^\top$ as the estimator of $\boldsymbol{\beta}^*$.
```

a constant step size, while the VSAS algorithm provides the capability of a variable step size for each predictor, denoted as $\sqrt{g_j^{(k)}}$ and $\tau\sqrt{g_j^{(k)}}$. Unlike the GSDAR algorithm which relies solely on the first derivative of loss for descent direction, our VSAS algorithm is based on a local quadratic approximation and incorporates both the first and second derivatives of loss to ensure a faster convergence rate.

3. Theoretical properties

In this section, we derive an explicit L_∞ upper bound for $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*$ and demonstrate that $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*$ can achieve the optimal convergence rate with high probability in finite iterations. When the target signal exceeds the detectable threshold, we show that $\hat{\boldsymbol{\beta}}$ can serve as the oracle estimator with high probability. The establishment of theoretical results requires the following conditions.

(C1) There are two constants $0 < L < U < \infty$ such that, for all $\boldsymbol{\alpha}_1 \neq \boldsymbol{\alpha}_2$ with $\|\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2\|_0 \le 2T$,

$$L \le \frac{(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2)^\top \nabla^2 L(\tilde{\boldsymbol{\alpha}})(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2)}{\|\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2\|_1 \|\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2\|_\infty} \le U,$$

where $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha}_1 + \nu(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_1)$ for any $\nu \in (0, 1)$.

(C2) Let $\tilde{\varepsilon}_i = \varepsilon_i^{-1} - \varepsilon_i$. Suppose that $\tilde{\varepsilon}_i, i = 1, 2, \ldots, n$ are independent and identically distributed with mean zero and sub-Gaussian tails, and $n \succeq \log(p_n)$.

(C3) $\|\boldsymbol{\beta}_{A^*}^*\|_{\min} \ge \frac{3c_1}{L}\sqrt{\frac{\log(p_n)}{n}}$ for a positive constant c_1.

Remark 3.1: Condition (C1) is an extended constrained strongly convex condition, which is essential for bounding the estimation error in high-dimensional models. Similar conditions are imposed in Huang, Jiao, Kang, et al. (2021), Huang, Jiao, Kang, et al. (2022) and Ming and Yang (2024b). Condition (C2) is a reasonable assumption regarding the model error, which is used to bound the estimation error with high probability. Condition (C3) is necessary to ensure that the target signal is sufficiently strong to be detectable.

Lemma 3.1: *If $\hat{\boldsymbol{\beta}}$ denotes a minimizer of* (5), *then*

$$\begin{cases} \hat{g}_j = \dfrac{\partial^2 L(\hat{\boldsymbol{\beta}}_{-j}, \beta_j)}{\partial^2 \beta_j}|_{\hat{\beta}_j}, & j \in [p_n], \\ \hat{d}_j = -\hat{g}_j^{-1} \dfrac{\partial L(\hat{\boldsymbol{\beta}}_{-j}, \beta_j)}{\partial \beta_j}|_{\hat{\beta}_j}, & j \in [p_n], \\ \hat{\beta}_j = H_\lambda\left(\sqrt{\hat{g}_j}(\hat{\beta}_j + \hat{d}_j)\right), & j \in [p_n], \end{cases} \tag{9}$$

where $H_\lambda(\cdot)$ is the hard thresholding operator given by

$$H_\lambda\left(\sqrt{\hat{g}_j}(\hat{\beta}_j + \hat{d}_j)\right) = \begin{cases} 0, & \left|\sqrt{\hat{g}_j}(\hat{\beta}_j + \hat{d}_j)\right| < \sqrt{2\lambda}, \\ \hat{\beta}_j, & \left|\sqrt{\hat{g}_j}(\hat{\beta}_j + \hat{d}_j)\right| \geq \sqrt{2\lambda}. \end{cases}$$

Conversely, if $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{d}}$ and $\hat{\boldsymbol{g}}$ satisfy (9)*, we claim that $\hat{\boldsymbol{\beta}}$ is a local minimizer of* (5)*.*

Remark 3.2: Lemma 3.1 gives the KKT conditions for the L_0-LPRE estimator, which are similar to those in Huang et al. (2018), C. Cheng et al. (2022) and Ming and Yang (2024b). Its proof can be found in the Appendix.

Theorem 3.2: *Let $q \leq T$ and $\boldsymbol{\beta}^{(0)} = \mathbf{0}$ in Algorithm* 2*. If condition* (C1) *holds with $0 < U < \frac{1}{\tau\sqrt{T}}$ and $\tau \in (0, 1]$, then*

$$\|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^*\|_\infty \leq \sqrt{(q+T)\left(1 + \frac{U}{L}\right)}(\sqrt{\xi})^k\|\boldsymbol{\beta}^*\|_\infty + \frac{2}{L}\|\nabla L(\boldsymbol{\beta}^*)\|_\infty, \tag{10}$$

where $\xi = 1 - \frac{2\tau L(1-\tau\sqrt{T}U)}{\sqrt{T}\kappa(1+q)} \in (0, 1)$ and $1 \leq \kappa < \infty$.

Theorem 3.3: *Under the conditions of Theorem* 3.2 *and condition* (C2)*, we have*

$$\|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^*\|_\infty \leq \sqrt{(q+T)\left(1 + \frac{U}{L}\right)}(\sqrt{\xi})^k\|\boldsymbol{\beta}^*\|_\infty + \frac{2c_1}{L}\sqrt{\frac{\log(p_n)}{n}}, \tag{11}$$

with probability at least $1 - c_2\exp(-c_3\log(p_n))$, where (c_1, c_2, c_3) are universal constants. If the condition $k \geq O\left(\log_{\frac{1}{\xi}} \frac{n}{\log(p_n)}\right)$ further holds, then we have, with high probability,

$$\|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^*\|_\infty \leq O\left(\sqrt{\frac{\log(p_n)}{n}}\right). \tag{12}$$

Remark 3.3: Theorems 3.2 and 3.3 provide the L_∞ upper bound for $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*$ and its optimal order, respectively. Theorem 3.4 below demonstrates that $\hat{\boldsymbol{\beta}}$ becomes the oracle estimator with high probability when the target signal exceeds the detectable threshold. Furthermore, it is important to emphasize that these theoretical results are directly linked to the solution sequence produced by the VSAS algorithm, but unrelated to the theoretical global solution of (5).

Theorem 3.4: *Suppose that conditions* (C1)–(C3) *hold with $0 < U < \frac{1}{\tau\sqrt{T}}$, $\tau \in (0, 1]$ and $1 \leq \kappa < \infty$. If $q \leq T$, $n \succeq \log(p_n)$ and $\boldsymbol{\beta}^{(0)} = \mathbf{0}$ in Algorithm* 2*, then $A^* \subseteq A^{(k)}$ with probability at least $1 - c_2\exp(-c_3\log(p_n))$, provided that $k > \log_{\frac{1}{\xi}}\left(9(T+q)\left(1 + \frac{U}{L}\right)r^2\right)$, where $r = \frac{\|\boldsymbol{\beta}^*\|_\infty}{\|\boldsymbol{\beta}^*_{A^*}\|_{\min}}$ is the range ratio of $\boldsymbol{\beta}^*_{A^*}$.*

4. Simulation study

In this section, we perform a simulation study to demonstrate the practical effectiveness of our L_0-LPRE estimator. Additionally, we propose an adaptive VSAS (AVSAS) algorithm for solving L_0-LPRE estimator. See Algorithm 3 for details. For the maximum integer L in Algorithm 3, we set $L = \lfloor n/\log(n) \rfloor$ as done by Fan and Lv (2008) and Huang et al. (2018). The parameter t in Algorithm 3 represents the increment of the solution path and can take values of 1, 2, or 4. Following L. Wang et al. (2013), we consider the high-dimensional Bayesian information criterion (HBIC) below to select the optimal tuning parameter T,

$$\text{HBIC}(\hat{T}) = \log(L(\hat{\boldsymbol{\beta}})) + \frac{C_n\log(p_n)}{n}|\hat{A}|, \tag{13}$$

where $C_n = 2\log(\log(n))$ is used throughout. All the experiments are implemented in R software, and the R code is accessed at https://github.com/hming177/VSAS.git.

Algorithm 3 Adaptive variable sorted active set for L_0-LPRE estimator

```
Input: Data $(\mathbb{Y}, \mathbb{X})$, $\boldsymbol{\beta}^{(0)}, \boldsymbol{d}^{(0)}, \boldsymbol{g}^{(0)}$, a step size $\tau$, an integer $t$, an integer $L$;
for $l = 1, 2 \ldots,$ do
    Run Algorithm 2 with tuning parameter $T = tl$ and initial value $(\boldsymbol{\beta}^{(l-1)}, \boldsymbol{d}^{(l-1)}, \boldsymbol{g}^{(l-1)})$. Denote the output by $\boldsymbol{\beta}^{l}$;
    if $T > L$ then
        Break;
    end if
    Calculate the HBIC value for $\boldsymbol{\beta}^{l}$, denoted by HBIC$(T_l)$;
end for
Choose the value of $T$ that corresponds to the smallest HBIC as the optimal tuning parameter $\hat{T}$;
Output: $\hat{\boldsymbol{\beta}}(\hat{T})$, the estimator of $\boldsymbol{\beta}^*$.
```

4.1. *Methods for comparisons*

We have designed six experiment examples (Examples 4.1–4.6) in Section 4.3 to investigate the finite sample performance of our method, L_0-LPRE, with several competitive methods. The details are provided below.

In Example 4.1, we consider a diverging-dimensional multiplicative model to compare our L_0-LPRE with the adaptive LASSO-based LPRE estimator (ALASSO-LPRE, Hao et al. (2016)) for which an ADMM algorithm is used and the involved tuning parameter λ is selected by the HBIC criterion. Typically, we search the optimal value of λ over 100 equally spaced grids in $[\lambda_{\min}, \lambda_{\max}]$ in log scale, where $\lambda_{\max} = \|\mathbb{X}^\top \log(\mathbb{Y})/n\|_\infty$ and $\lambda_{\min} = 10^{-3}\lambda_{\max}$.

In Example 4.2, we consider a high-dimensional multiplicative model to compare our proposed method with five methods. (i) LASSO-LS is referred to a two-step method for which we apply a logarithmic transformation to the response in the first step, and a standard LASSO estimator is computed using the *glmnet* package in the second stage. (ii) SCAD-LS is a method similar to LASSO-LS except that the SCAD penalty function is used. (iii) MCP-LS is a method similar to SCAD-LS but uses the MCP penalty function. Note that the SCAD and MCP estimators are computed using the *ncvreg* package. (iv) CHIP-LS is a method similar to LASSO-LS but with the CHIP penalty. (v) L_0-LS is a method similar to LASSO-LS except that the L_0 regularization is considered in the second step and solved by the ESDAR algorithm. Furthermore, we set $\gamma = 3$ for MCP-LS, $\gamma = 3.7$ for SCAD-LS and CHIP-LS in the simulations and applications. The tuning parameter λ involved in LASSO-LS, SCAD-LS and MCP-LS is selected, which resembles that in Example 4.1. The parameter T involved in L_0-LS is determined in the same way as in L_0-LPRE.

In Examples 4.3–4.5, we follow Huang, Jiao, Kang, et al. (2021) and Huang et al. (2018) to examine the influence of sample size n, sparsity level q, and dimension p_n on the performance of L_0-LPRE and L_0-LS, respectively.

In Example 4.6, we allow the sparsity level q and dimension p_n to vary with n. In this example, we fix $T = q$ and change the sample size n to examine how the number of iterations in the algorithm affects the performance of L_0-LPRE and L_0-LS.

4.2. *Evaluation of methods*

To evaluate the performance of the various methods, we use the following criteria with 100 independent runs:

(i) the average estimation error (AEE), AEE $= \frac{1}{100}\sum \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2$,

(ii) the average prediction error (APE), APE $= \frac{1}{100}\sum \frac{\|\mathbb{Y}_{\text{te}}^{-1}\cdot\mathbb{X}_{\text{te}}\hat{\boldsymbol{\beta}} - \mathbf{1}\|_1}{n_{\text{te}}}$,

(iii) the average positive discovery rate (APDR), APDR $= \frac{1}{100}\sum \frac{|\hat{A}\cap A^*|}{|A^*|}$,

(iv) the average true model rate (ATMR), ATMR $= \frac{1}{100}\sum \mathbb{I}\{\hat{A} = A^*, \hat{I} = I^*\}$,

where $(\mathbb{Y}_{\text{te}}, \mathbb{X}_{\text{te}})$ are test data with size $n_{te} = 100$. Obviously, it is expected that one particular method performs good if its AEE and APE are close to zero and its APDR and ATMR are close to one. Moreover, we define the evaluation criteria for other methods relative to the L_0-LPRE method, including the average relative estimation error (AREE), the average relative prediction error (ARPE), and the average relative running time (ARRT), which are quantified respectively by

(v) AREE $= \frac{1}{100}\sum\{\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 / \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2\}$,

(vi) ARPE $= \frac{1}{100}\sum\{\|\mathbb{Y}_{\mathrm{te}}^{-1}\cdot\mathbb{X}_{\mathrm{te}}\tilde{\boldsymbol{\beta}} - \mathbf{1}\|_1/\|\mathbb{Y}_{\mathrm{te}}^{-1}\cdot\mathbb{X}_{\mathrm{te}}\hat{\boldsymbol{\beta}} - \mathbf{1}\|_1\}$,

(vii) ARRT $= \frac{1}{100}\sum\{\mathrm{RT}^{\tilde{\boldsymbol{\beta}}}/\mathrm{RT}^{\hat{\boldsymbol{\beta}}}\}$,

where $\tilde{\boldsymbol{\beta}}$ denotes the optimal estimator of the other methods, and $\mathrm{RT}^{\tilde{\boldsymbol{\beta}}}$ and $\mathrm{RT}^{\hat{\boldsymbol{\beta}}}$ denote the running time for obtaining the optimal estimator of the other methods and using the AVSAS algorithm, respectively. Obviously, when the values of AREE, ARPE and ARRT are greater than 1, the L_0-LPRE method performs better than other methods. The averaged number of iterations (ANI) is reported in Example 4.6.

4.3. Simulation examples

Following Huang, Jiao, Jin, et al. (2021), we consider two cases to generate the design matrix $\mathbb{X}$ as follows:

(i) $\boldsymbol{X}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, and $\boldsymbol{\Sigma}_{jk} = 0.5^{|j-k|}, j, k \in [p_n]$;

(ii) $\tilde{\boldsymbol{X}}_i \sim N(\mathbf{0}, \mathbf{I}_{p_n}), i \in [n]$, then consider $\boldsymbol{X}_1 = \tilde{\boldsymbol{X}}_1$, $\boldsymbol{X}_j = \tilde{\boldsymbol{X}}_j + 0.2(\tilde{\boldsymbol{X}}_{j-1} + \tilde{\boldsymbol{X}}_{j+1}), j = 2, \ldots, p_n - 1$, and $\boldsymbol{X}_{p_n} = \tilde{\boldsymbol{X}}_{p_n}$.

For the true parameter vector $\boldsymbol{\beta}^*$, we set $\beta_j^* \sim \mathrm{Uniform}(r_*, r^*)$ for $j \in A^*$ and $\boldsymbol{\beta}_{I^*}^* = \mathbf{0}$, where A^* is a subset randomly chosen from $[p_n]$ with $|A^*| = q < n$. Here, we fix $r_* = \sqrt{2\log(p_n)/n}, r^* = Rr_*$ and $\tau = 1$. For ε_i with $i \in [n]$, we consider the following four error distributions,

(d1) $\varepsilon_i \sim f_1$, where $f_1(x) = c_1\exp(-x - 1/x - \log x + 2)$ and c_1 is a normalization constant,
(d2) $\varepsilon_i \sim LN(0, 1)$, the standard log normal distribution,
(d3) $\log(\varepsilon_i) \sim \mathrm{Uniform}(-2, 2)$,
(d4) $\varepsilon_i \sim \mathrm{Uniform}(0.5, a)$, where $a \approx 1.608$ such that $E(\varepsilon_i) = E(\varepsilon_i^{-1})$,
(d5) $\varepsilon_i \sim \Gamma(2, 1)$, the gamma distribution with shape parameter 2 and scale parameter 1,

where (d1) and (d2) correspond to the effective distributions for LPRE and LS methods, respectively, and both (d3) and (d4) satisfy $E(\varepsilon_i) = E(\varepsilon_i^{-1})$. (d5) does not satisfy $E(\log(\varepsilon_i)) = 0$ and $E(\varepsilon_i) = E(\varepsilon_i^{-1})$, i.e. the model has systematic errors.

Example 4.1: Following Hao et al. (2016), we consider a diverging number of predictors with $p_n = \lfloor 4n^{1/4} - 4\rfloor$, where $\mathbb{X}$ is generated from Case (i). Here, we fix $q = 6$, $R = 10$, and let n vary from 100 to 700 with an increment being 100. To compare the algorithm runtime, we take the same candidate tuning parameters for both L_0-LPRE and ALASSO-LPRE, meaning that λ takes on the value of the p_n equally spaced grid in $[\lambda_{\min}, \lambda_{\max}]$, and T varies from 1 to p_n. The Figure 2 presents the simulation results.

Example 4.2: In this example, we consider a high-dimensional multiplicative model with $n = 200, q = 10, p_n = 2000, R = 5$, where $\mathbb{X}$ is generated from either Case (i) or Case (ii). The simulation results for LASSO-LS, MCP-LS, SCAD-LS, CHIP-LS, L_0-LS and L_0-LPRE are shown in Table 1.

Example 4.3: In this example, our aim is to examine the influence of sample size n on the performance of L_0-LS and L_0-LPRE. Here, we generate $\mathbb{X}$ according to Case (i), while keeping q fixed at 10, p_n at 1000, and R at 5. The sample size n ranges from 200 to 800, incremented by 100. The simulation results are depicted in Figure 3.

Example 4.4: To examine the impact of the sparsity level q on the performance of L_0-LS and L_0-LPRE, we generate $\mathbb{X}$ based on Case (ii) and set $n = 400, p_n = 5000$, and $R = 3$. The sparsity level q is incremented by 8, starting from 4 and reaching up to 52. The simulation results are illustrated in Figure 4.

Example 4.5: In this example, we investigate the impact of the dimension p_n on the performance of L_0-LS and L_0-LPRE. Likewise, $\mathbb{X}$ is generated from Case (ii). We set $n = 400$, $q = 10$ and $R = 5$. The dimension p_n ranges from 2000 to 10000, incremented by 2000. The simulation results are displayed in Figure 5.

Example 4.6: To illustrate the number of iterations in the algorithm, we generate $\mathbb{X}$ using Case (ii), and take $q = \lfloor n^{1/2}\rfloor, p_n = \lfloor\exp(n^{0.35})\rfloor, R = 3$. The sample size n varies from 100 to 700 in increments of 100 and $T = q$. The simulation results are presented in Figure 6.

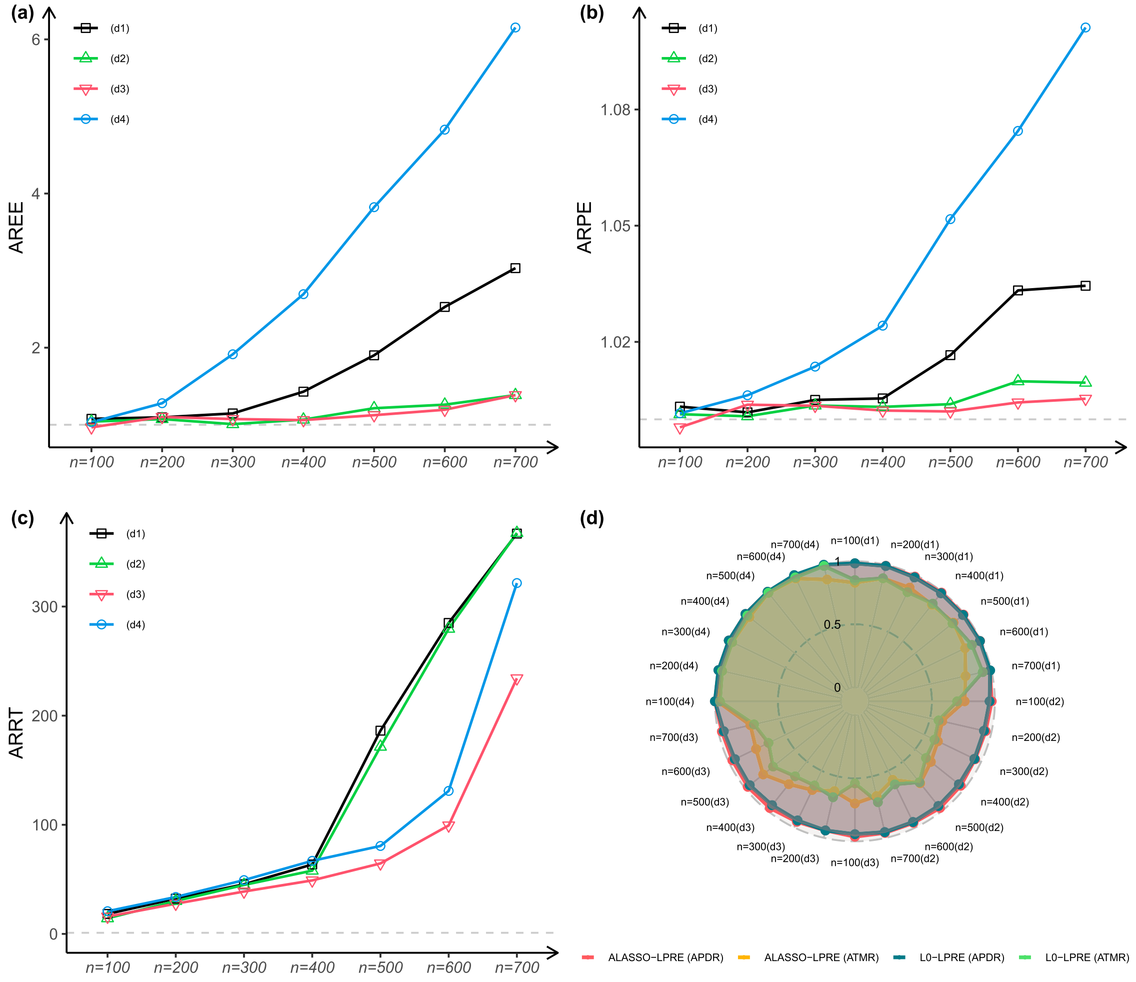

Figure 2. Simulation results for ALASSO-LPRE and L_0-LPRE in Example 4.1, where the average relative evaluation criterion in Figures (a), (b), and (c) is ALASSO-LPRE relative to L_0-LPRE.

Table 1. The simulation results for Example 4.2, and the standard deviations are given in parentheses.

		Case (*i*)				Case (*ii*)			
Error	Method	AEE	APE	APDR	ATMR	AEE	APE	APDR	ATMR
$\varepsilon \sim$(d1)	LASSO-LS	0.791(0.171)	1.223(0.386)	0.978	0.21	0.756(0.165)	1.226(0.369)	0.967	0.27
	MCP-LS	0.186(0.056)	0.627(0.073)	0.998	0.97	0.179(0.065)	0.644(0.080)	0.995	0.94
	SCAD-LS	0.273(0.085)	0.662(0.082)	0.998	0.89	0.248(0.092)	0.676(0.099)	0.995	0.87
	CHIP-LS	0.272(0.093)	0.664(0.088)	0.998	0.84	0.256(0.099)	0.681(0.099)	0.995	0.86
	L_0-LS	0.148(0.043)	0.616(0.070)	0.998	0.98	0.152(0.060)	0.637(0.075)	0.994	0.92
	L_0-LPRE	**0.145(0.035)**	**0.616(0.071)**	**0.999**	**0.99**	**0.146(0.053)**	**0.634(0.076)**	**0.996**	**0.96**
$\varepsilon \sim$(d2)	LASSO-LS	1.575(0.669)	32.133(179.238)	0.604	0.05	1.468(0.619)	17.014(56.553)	0.678	0.07
	MCP-LS	0.491(0.238)	1.469(0.892)	0.896	0.41	0.455(0.230)	1.411(0.429)	0.921	0.49
	SCAD-LS	0.702(0.343)	1.950(1.627)	0.875	0.26	0.638(0.275)	1.73(0.944)	0.918	0.34
	CHIP-LS	0.809(0.479)	3.509(8.057)	0.845	0.19	0.727(0.354)	1.958(1.207)	0.892	0.26
	L_0-LS	0.391(0.189)	1.291(0.349)	0.894	0.43	0.373(0.176)	1.324(0.333)	0.901	0.38
	L_0-LPRE	**0.354(0.145)**	**1.233(0.278)**	**0.934**	**0.46**	**0.337(0.146)**	**1.288(0.316)**	**0.936**	**0.49**
$\varepsilon \sim$(d3)	LASSO-LS	1.870(0.713)	42.424(96.276)	0.484	0.02	1.862(0.713)	27.464(41.424)	0.473	0.07
	MCP-LS	0.642(0.305)	2.046(1.185)	0.849	0.29	0.621(0.276)	1.973(0.965)	0.877	0.41
	SCAD-LS	0.909(0.426)	3.567(5.76)	0.808	0.15	0.893(0.393)	3.132(3.463)	0.846	0.19
	CHIP-LS	1.024(0.504)	6.628(24.101)	0.780	0.11	0.992(0.477)	4.704(9.085)	0.826	0.18
	L_0-LS	0.541(0.214)	1.767(0.402)	0.835	0.22	0.517(0.243)	1.713(0.454)	0.854	0.29
	L_0-LPRE	**0.384(0.186)**	**1.574(0.282)**	**0.906**	**0.44**	**0.312(0.144)**	**1.487(0.24)**	**0.950**	**0.63**
$\varepsilon \sim$(d4)	LASSO-LS	0.390(0.071)	0.458(0.073)	0.999	0.17	0.367(0.061)	0.448(0.062)	**1**	0.28
	MCP-LS	0.073(0.018)	0.302(0.026)	**1**	**1**	0.070(0.015)	0.295(0.022)	**1**	**1**
	SCAD-LS	0.078(0.022)	0.303(0.027)	**1**	**1**	0.072(0.016)	0.295(0.022)	**1**	**1**
	CHIP-LS	0.076(0.021)	0.303(0.026)	**1**	**1**	0.071(0.016)	0.295(0.022)	**1**	**1**
	L_0-LS	0.072(0.018)	0.302(0.026)	**1**	**1**	0.070(0.015)	0.295(0.022)	**1**	**1**
	L_0-LPRE	**0.071(0.018)**	**0.302(0.026)**	**1**	0.99	**0.069(0.015)**	**0.295(0.022)**	**1**	**1**
$\varepsilon \sim$(d5)	LASSO-LS	1.241(0.424)	4.932(21.309)	0.782	0.06	1.285(0.517)	5.944(13.479)	0.759	0.08
	MCP-LS	0.354(0.186)	0.810(0.258)	0.948	**0.69**	0.318(0.161)	0.825(0.246)	0.961	**0.75**
	SCAD-LS	0.495(0.217)	0.888(0.291)	0.947	0.56	0.434(0.173)	0.882(0.270)	0.964	0.58
	CHIP-LS	0.666(0.326)	3.101(20.838)	0.910	0.35	0.592(0.281)	1.484(3.570)	0.927	0.40
	L_0-LS	0.334(0.173)	0.802(0.250)	0.923	0.54	0.283(0.137)	0.805(0.228)	0.949	0.61
	L_0-LPRE	**0.291(0.155)**	**0.781(0.241)**	**0.951**	**0.69**	**0.249(0.118)**	**0.791(0.232)**	**0.970**	0.72

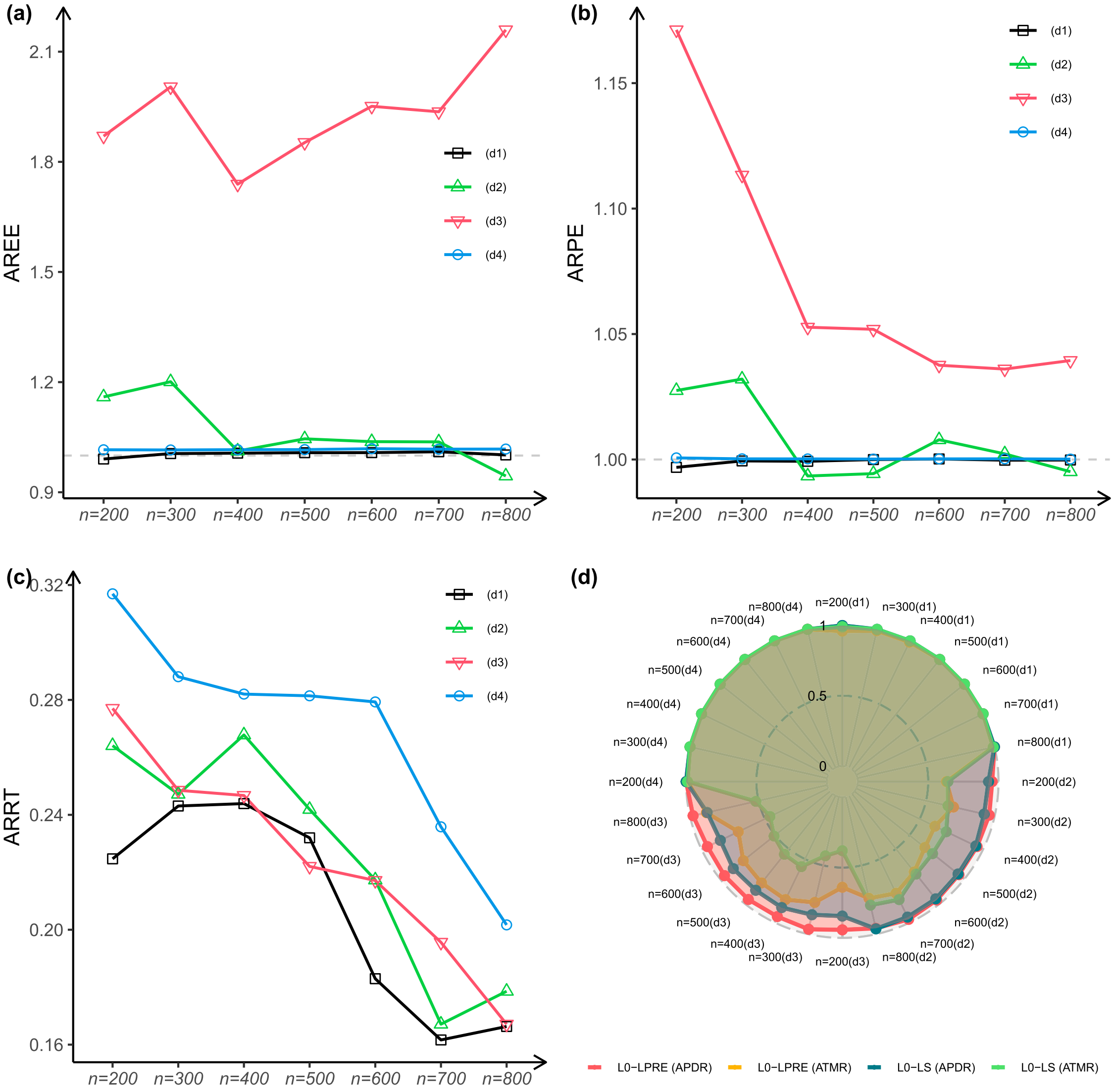

Figure 3. Simulation results for L_0-LS and L_0-LPRE in Example 4.3, where the average relative evaluation criterion in Figures (a), (b), and (c) is L_0-LS relative to L_0-LPRE.

4.4. Summary of simulation results

From Figures 2–6 and Table 1, the following observations can be made.

- Compared to ALASSO-LPRE estimator, we can observe in Figure 2 that L_0-LPRE estimator not only significantly outperforms ALASSO-LPRE in estimation and prediction (except for $n = 100$ and error distribution (d3)), but also takes lower computing time. In terms of APDR and ATMR, we can see that L_0-LPRE and ALASSO-LPRE are very comparable even though L_0-LPRE performs better under the (d1) and (d4) distributions with sample size $n = 700$ and ALASSO-LPRE performs better under the (d3) distribution.
- Our L_0-LPRE method clearly outperforms LASSO-LS, MCP-LS, SCAD-LS, CHIP-LS and L_0-LS in high-dimensional settings across five different error distributions and two cases of generating covariates, and is followed by L_0-LS. This is illustrated in Table 1.
- Figures 3–6 show that in terms of AREE, ARPE, APDR and ATMR, L_0-LPRE works much better than L_0-LS under all error distributions except (d2). In the setting (d2), when the number of truly active covariates is unknown, L_0-LPRE performs slightly worse than L_0-LS in a few cases, which is reasonable since (d2) is the efficient distribution of the LS-based method. In terms of computing time, the cost of L_0-LPRE is six times that of L_0-LS. The discrepancy decreases as the dimension p_n increases. Furthermore, we find that the L_0-penalized methods (L_0-LPRE and L_0-LS) converge within a few steps when the true number of active covariates is known, and that L_0-LPRE converges faster than L_0-LS in the setting of (d3).

In summary, the above evidence has demonstrated the effectiveness of our L_0-LPRE approach.

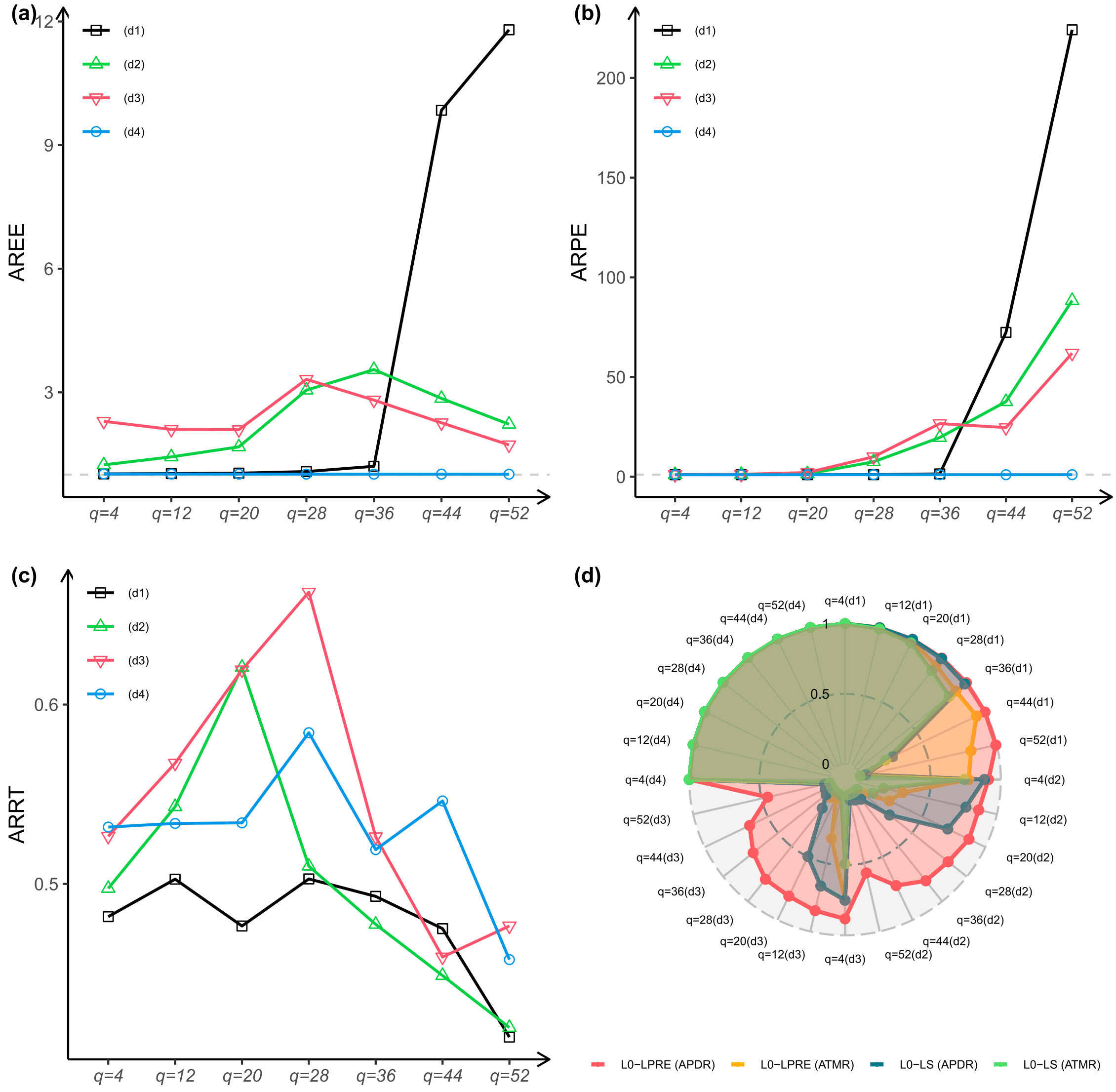

Figure 4. Simulation results for L_0-LS and L_0-LPRE in Example 4.4, where the average relative evaluation criterion in Figures (a), (b), and (c) is L_0-LS relative to L_0-LPRE.

5. Real data applications

In this section, we employ our L_0-LPRE to two real-world datasets, in which LASSO-LS, MCP-LS, SCAD-LS, CHIP-LS and L_0-LS methods are also compared.

5.1. Riboflavin data

In this subsection, we apply our method to the Riboflavin data available in the R package *hdi*. This dataset has been analysed in Buhlmann et al. (2014) and Zhao et al. (2022), in which a linear model is used to describe the relationship between the log riboflavin production rate and 4088 log gene expression levels ($\mathbb{X}$). Figure 7 shows that the log riboflavin production rate does not follow a normal distribution according to the normality test. Thus, we use a multiplicative model and the LPRE loss to predict the riboflavin production rate ($\mathbb{Y}$) using $\mathbb{X}$. For this data, the dimension is $p_n = 4088$ and the sample size is $n = 71$.

Like Zhao et al. (2022), we use the mean squared log prediction residuals (i.e. MSLPR $= \frac{1}{n_{\text{te}}}\sum_{i=1}^{n_{\text{te}}}[\log(y_i) - \log(\hat{y}_i)]^2$) to evaluate the performance of various methods. Here, we randomly draw a subset from the whole data with a size n_{tr} among $\{40, 50, 60\}$ as the training set and the remaining n_{te} data points as the test set. We repeat this procedure 1000 times. The results are shown in Tables 2 and 3. From Table 2, we can observe that L_0-LPRE performs best, followed by L_0-LS, regardless of the sample size of training set. The performance of all methods can get improved as the sample size of training set increases. On the other hand, in terms of MMS, L_0-LS and L_0-LPRE behave similarly, while the latter gives a smaller MSLPR. Furthermore, we report the frequencies of the top 6 genes selected by L_0-LS and L_0-LPRE in Table 3. Clearly, the first six genes selected by L_0-LS and L_0-LPRE are almost

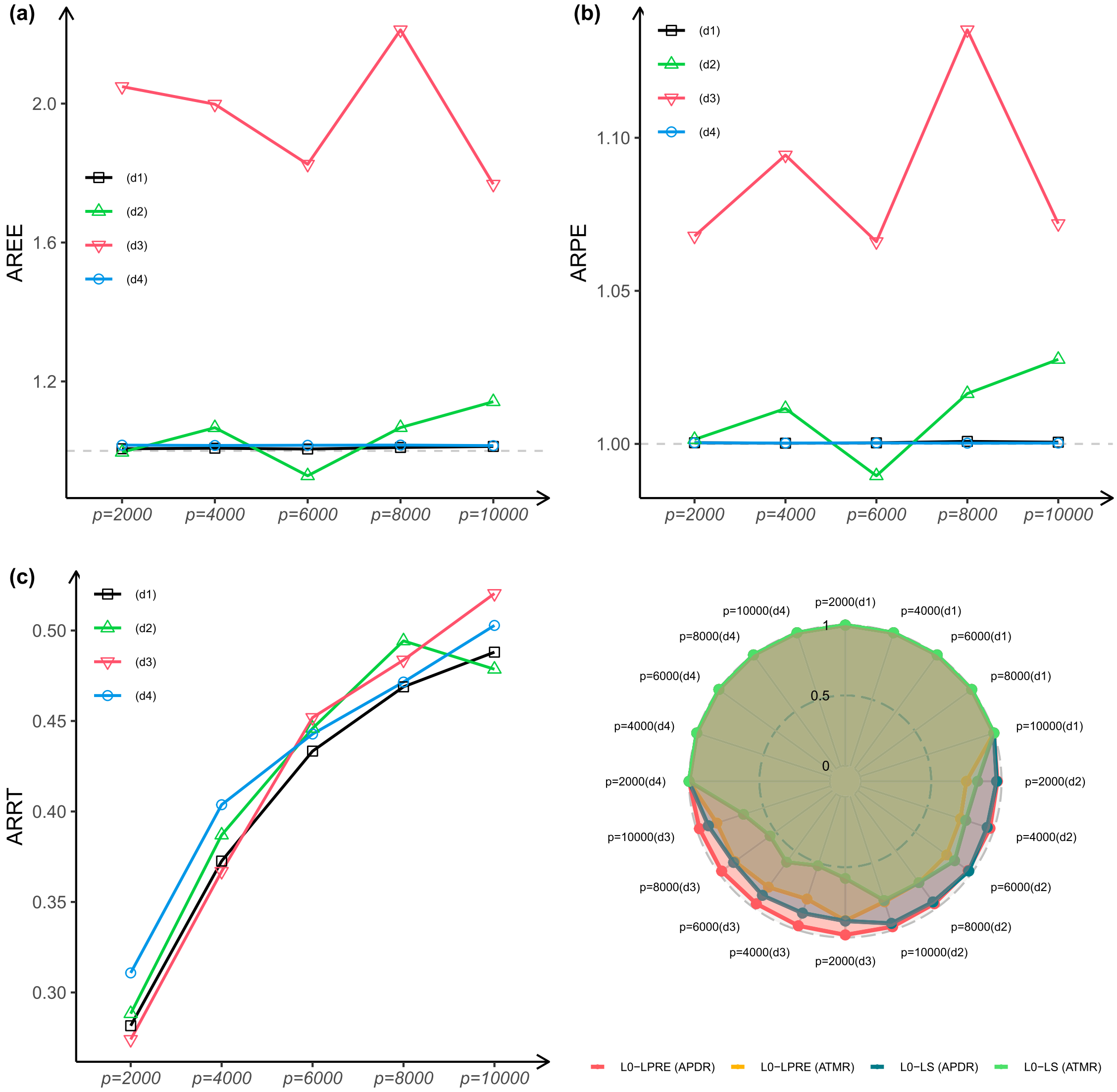

Figure 5. Simulation results for L_0-LS and L_0-LPRE in Example 4.5, where the average relative evaluation criterion in Figures (a), (b), and (c) is L_0-LS relative to L_0-LPRE.

identical for different sample sizes of training set. Meanwhile, L_0-LPRE yields a higher discovery rate for the gene XHLA_at confirmed by Buhlmann et al. (2014). This indicates that XHLA_at could be a potential gene to have an important effect on the riboflavin production rate. We also find that both the L_0-LS and L_0-LPRE methods select the genes XHLA_at, YOAB_at, YXLD_at and YCKE_at when the sample size of training set is 40, 50 and 60. Besides, L_0-LPRE tends to select YXLE_at when the sample size of training set is 50 and 60. In addition, these findings are consistent with the results of the study (Javanmard & Montanari, 2014), in which the two genes, YXLD_at and YXLE_at, were identified. To sum up, we can draw the conclusion that the five genes, XHLA_at, YOAB_at, YXLD_at, YCKE_at and YXLE_at, may be the most relevant ones for the riboflavin production rate. Afterwards, we further obtain the following multiplicative model

$$\hat{y} = \exp(-1.5481 + 0.3771 \times \text{XHLA\_at} - 1.0210 \times \text{YOAB\_at}$$
$$+ 0.0043 \times \text{YXLD\_at} + 0.2584 \times \text{YCKE\_at} - 0.3725 \times \text{YXLE\_at}),$$

which indicates that the genes, XHLA_at, YXLD_at and YCKE_at, have a positive effect, while YOAB_at and YXLE_at have a negative effect on the log response.

5.2. Supermarket data

In this subsection, we use supermarket data to illustrate the effectiveness of our method. The dataset has been analysed in the study of Wang (2009), Z. Chen et al. (2018) and Liu et al. (2022), which contains a sample of 464 daily records from a supermarket. The response variable we are interested in is the number of customers who visited

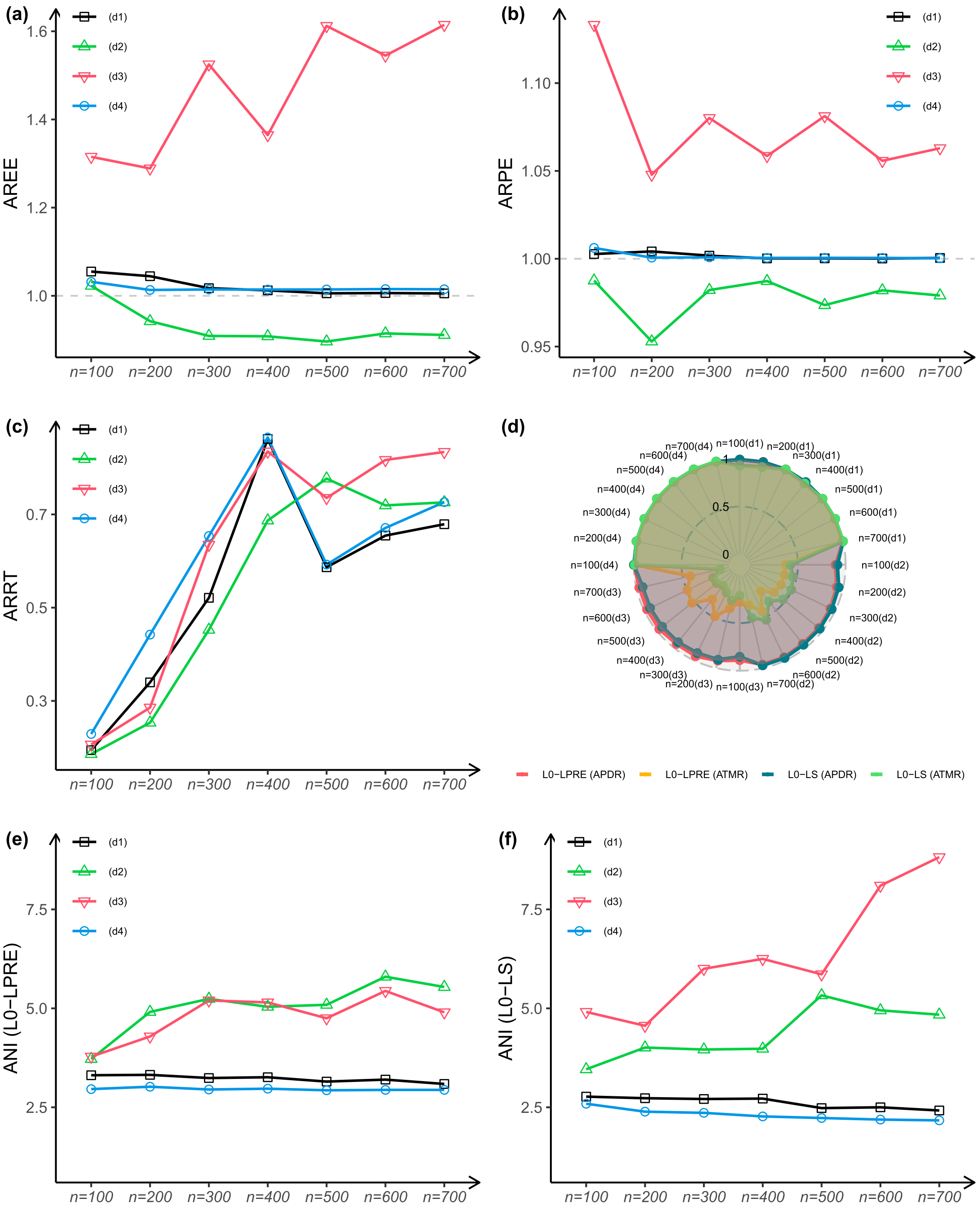

Figure 6. Simulation results for L_0-LS and L_0-LPRE in Example 4.6, where the average relative evaluation criterion in Figures (a), (b), and (c) is L_0-LS relative to L_0-LPRE.

the supermarket that day. The available covariates are the sale volume data of 6398 products. For data privacy reasons, the response variable and predictors have been standardized to have zero mean and unit variance. In order to use our model, we perform an exponential operation on the response variable in the following data analysis. This transformation enables the LPRE loss to be applied. Our objective is to select some important products that have a substantial impact on the number of customers every day through their sale volumes. Similar to the previous analysis, we randomly select 200, 300 and 400 data points from the original data as a training set and the remaining data as a test set. We repeat this procedure 100 times, and report the results in Tables 4–5.

From Table 4, we can see that the proposed L_0-LPRE achieves the minimum value of MSLPR when the sample size of training set is 200, 300 and 400, which indicates that our L_0-LPRE method has best performance among these methods. Table 5 shows that both L_0-LS and L_0-LPRE tend to select nearly the same top 12 predictors across different training sample sizes. It is important to note that for each training sample, both methods select X_3, $X_6, X_{11}, X_{39}, X_{62}, X_{139}$ and X_{2830}. In the training sample size of 300 and 400, both L_0-LS and

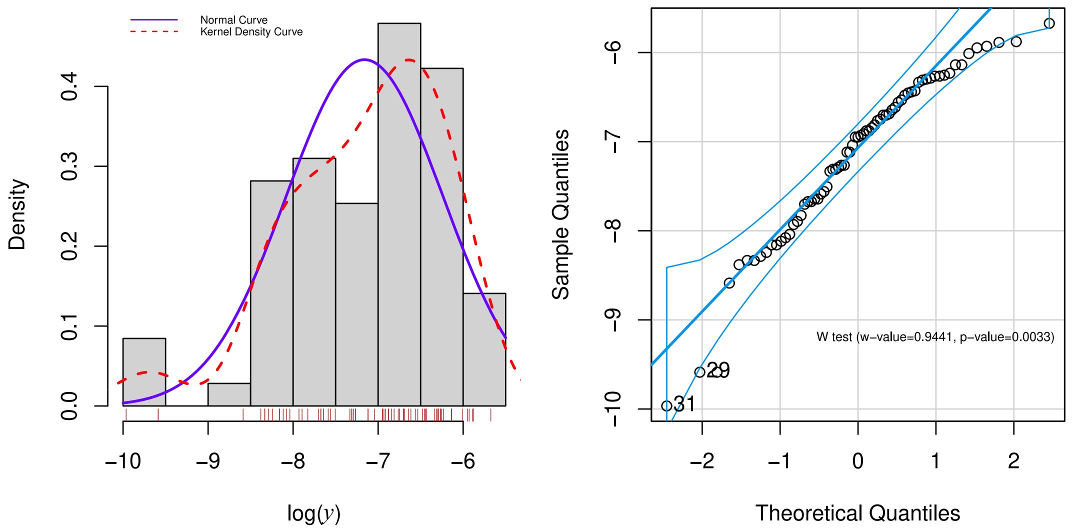

Figure 7. Density curves and Q-Q plots for $\log(y)$ in Riboflavin data.

Table 2. Results of 1000 repetitions of MSLPR and MMS for Riboflavin data when training set size is among $\{40, 50, 60\}$.

	$n_{tr} = 40$		$n_{tr} = 50$		$n_{tr} = 60$	
Method	MSLPR	MMS	MSLPR	MMS	MSLPR	MMS
LASSO-LS	0.8737(0.1896)	0(0)	0.8625(0.2421)	0(0)	0.8418(0.3749)	0(0)
MCP-LS	0.8200(0.2136)	0.513(1.1538)	0.7622(0.2936)	0.896(1.5722)	0.5798(0.4498)	2.385(2.1632)
SCAD-LS	0.8718(0.1918)	0.014(0.2277)	0.8614(0.2420)	0.008(0.1843)	0.8408(0.3756)	0.015(0.2736)
CHIP-LS	0.8728(0.1895)	1.158(0.4186)	0.8618(0.2419)	1.150(0.4021)	0.8414(0.3748)	1.086(0.2944)
L_0-LS	0.6967(0.1996)	1.292(0.5959)	0.6472(0.2502)	1.590(0.8177)	0.5575(0.3725)	2.228(1.0813)
L_0-LPRE	**0.6807(0.206)**	1.491(0.7839)	**0.6136(0.2399)**	1.797(0.9234)	**0.5255(0.3322)**	2.333(1.0767)

Note: Standard deviations are given in parentheses.

Table 3. Frequency of the first 6 genes selected with 1000 replications for L_0-LS and L_0-LPRE in Riboflavin data.

$n_{tr} = 40$				$n_{tr} = 50$				$n_{tr} = 60$			
L_0-LS		L_0-LPRE		L_0-LS		L_0-LPRE		L_0-LS		L_0-LPRE	
Genes	Frequency	Genes	Frequency	Genes	Frequency	Genes	Frequency	Genes	Frequency	Genes	Frequency
XHLA_at	266	XHLA_at	279	XHLA_at	416	XHLA_at	474	XHLA_at	612	XHLA_at	718
YOAB_at	151	YOAB_at	134	YXLD_at	209	YOAB_at	170	YXLD_at	343	YOAB_at	287
YXLD_at	126	YCKE_at	133	YOAB_at	188	YCKE_at	139	YOAB_at	305	YXLD_at	203
YDAR_at	103	YDAR_at	90	YCKE_at	113	YXLD_at	132	YXLG_at	104	YXLE_at	160
YCKE_at	93	YXLD_at	77	YDAR_at	72	YXLE_at	74	YCKE_at	102	YCKE_at	110
XTRA_at	44	XKDF_at	49	YXLG_at	43	YDAR_at	57	XHLB_at	80	YXLG_at	102

Table 4. Results of MSLPR and MMS for Supermarket data when training sample size is among 200, 300 and 400.

	$n_{tr} = 200$		$n_{tr} = 300$		$n_{tr} = 400$	
Method	MSLPR	MMS	MSLPR	MMS	MSLPR	MMS
LASSO-LS	0.758(0.173)	1.70(1.541)	0.630(0.144)	3.42(2.531)	0.448(0.151)	7.90(3.311)
MCP-LS	0.371(0.084)	2.56(2.438)	0.294(0.107)	6.08(4.970)	0.160(0.069)	14.67(5.297)
SCAD-LS	0.754(0.182)	1.79(1.665)	0.620(0.153)	3.66(3.019)	0.411(0.146)	9.20(4.058)
CHIP-LS	1.006(0.051)	1.07(0.256)	1.005(0.087)	1.01(0.100)	1.010(0.161)	1(0)
L_0-LS	0.216(0.037)	5.70(1.367)	0.166(0.033)	8.01(1.867)	0.129(0.028)	9.71(1.701)
L_0-LPRE	**0.206(0.032)**	6.22(1.418)	**0.160(0.028)**	8.39(1.476)	**0.124(0.029)**	10.39(1.912)

Note: Standard deviations are given in parentheses over 100 repetitions.

L_0-LPRE can select X_{56}, X_{410} and X_{417}. In addition, L_0-LPRE selects the covariate X_{176} more frequently than L_0-LS. This may explain why L_0-LPRE outperforms L_0-LS. Overall, we have detected 11 products named X_3, X_6, $X_{11}, X_{39}, X_{56}, X_{62}, X_{139}, X_{176}, X_{410}, X_{417}$ and X_{2830} that could significantly contribute to the number of customers each day. In order to precisely quantify the relationship between the response and these 11 predictors, we fit the

Table 5. Frequency of the first 12 genes selected by L_0-LS and L_0-LPRE with 100 replications for Supermarket data.

$n_{\text{tr}} = 200$				$n_{\text{tr}} = 300$				$n_{\text{tr}} = 400$			
L_0-LS		L_0-LPRE		L_0-LS		L_0-LPRE		L_0-LS		L_0-LPRE	
Variable	Frequency	Variable	Frequency	Variable	Frequency	Variable	Frequency	Variable	Frequency	Variable	Frequency
X_{11}	67	X_{11}	63	X_{11}	99	X_{11}	88	X_6	100	X_6	100
X_{139}	58	X_{139}	62	X_{139}	87	X_{139}	86	X_{11}	100	X_{11}	100
X_3	47	X_3	50	X_6	82	X_6	83	X_{139}	100	X_{139}	99
X_5	39	X_6	47	X_{2830}	62	X_{2830}	60	X_{2830}	87	X_{410}	82
X_6	39	X_5	43	X_3	46	X_3	49	X_{410}	81	X_{62}	70
X_{21}	25	X_{21}	25	X_{410}	37	X_{62}	45	X_{39}	71	X_{2830}	69
X_{62}	24	X_{62}	25	X_{62}	36	X_{39}	42	X_{62}	61	X_{56}	67
X_{39}	17	X_{1213}	22	X_{417}	34	X_{417}	30	X_{56}	60	X_{39}	64
X_{10}	16	X_{39}	16	X_{39}	33	X_{410}	28	X_{417}	54	X_{176}	57
X_{2830}	15	X_{2830}	16	X_{56}	22	X_{56}	24	X_3	48	X_7	48
X_{1213}	14	X_{10}	14	X_{107}	21	X_{176}	23	X_7	31	X_3	36
X_{4981}	10	X_{107}	13	X_7	15	X_5	22	X_{107}	21	X_{417}	33

following multiplicative model

$$\begin{aligned}\hat{y} = \exp(&0.0985X_3 + 0.2153X_6 + 0.2400X_{11} + 0.1182X_{39}\\ &+ 0.1078X_{56} + 0.1082X_{62} + 0.1792X_{139} + 0.1014X_{176}\\ &+ 0.1117X_{410} + 0.1131X_{417} + 0.1342X_{2830}).\end{aligned}$$

From the above, we can see that all the coefficients of the 11 variables are positive, indicating that increasing the sale volume of each of these 11 products causes the number of customers every day to increase. We also note that no intercept is included in the model due to data standardization. Moreover, to make a deep comparison, we compare the method proposed by Z. Chen et al. (2018), which is a two-stage approach applied to this data. A distance correlation-based screening (DC-SIS) is first employed to reduce the dimension and then an additive model is fitted on the reduced data with Wald's χ^2 test. Z. Chen et al. (2018) identified the seven predictors: $X_3, X_6, X_{11}, X_{39}, X_{42}, X_{62}$, and X_{139}. In contrast to Z. Chen et al. (2018), Liu et al. (2022) proposed a PC-Knockoff procedure and selected 12 significant variables: $X_3, X_6, X_{10}, X_{11}, X_{30}, X_{42}$, X_{48}, X_{71}, X_{129}, X_{139}, X_{176} and X_{400}. Figure 8 displays a Venn diagram to illustrates the overlaps between the 11 variables selected by our method and the variables by Z. Chen et al. (2018) and Liu et al. (2022), respectively. From this figure, one can see that six out of the 11 variables we have selected coincide with those identified by Z. Chen et al. (2018), while seven of the selected variables match those selected by Liu et al. (2022). In addition, it is worth noting that Liu et al. (2022) has selected X_{176} as a significant variable, which is also selected by our method. Finally, we examine the more influence of the 11 variables on the number of customers per day. To this end, we construct a multiplicative model based on the variables obtained from Z. Chen et al. (2018) and Liu et al. (2022) and then compare them with our L_0-LPRE. Following Liu et al. (2022), the entire dataset is randomly divided into a training set of size 400 and a test set of size 64 over 200 repetitions. The average $R^2 = 1 - n^{-1}\| \log(\mathbb{Y}) - \mathbb{X}\hat{\boldsymbol{\beta}}\|_2^2$ defined in Wang (2009) for both the training sample and the test sample is shown in Table 6. Based on the results given in Table 6, we can conclude that the proposed L_0-LPRE method surpasses the two methods, DC-SIS with χ^2-test and PC-Knockoff, in terms of R^2 for both the training and test samples. Furthermore, when constructing a multiplicative model using the 11 variables, we find that the LPRE-related estimators can achieve the highest R^2. This suggests that the 11 variables selected may yield more predictive performance compared to those selected by Z. Chen et al. (2018) and Liu et al. (2022).

6. Conclusion

In this paper, we propose a new algorithm for solving L_0-regularized high-dimensional sparse multiplicative models with LPRE loss, and derive an estimation error bound and an optimal convergence rate by the VSAS algorithm. Furthermore, we demonstrate that the L_0-LPRE estimator can reach the oracle estimator with high probability if the target signal exceeds the detectable level. Extensive numerical results show that the proposed L_0-LPRE method can outperform many existing competitors such as ALASSO-LPRE, LASSO-LS, MCP-LS, SCAD-LS, CHIP-LS and L_0-LS.

Disclosure statement

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

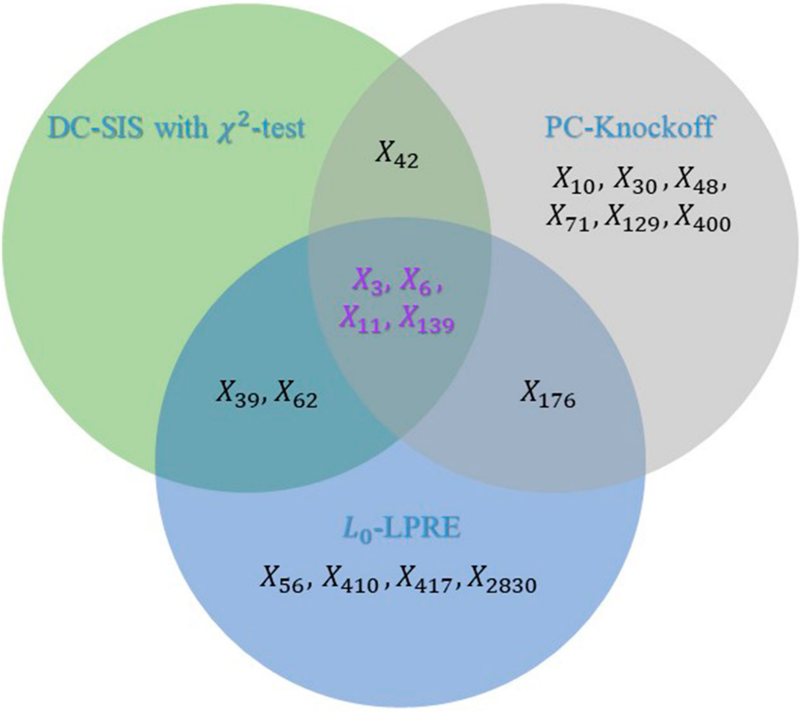
DC-SIS with χ^2-test
PC-Knockoff
L_0-LPRE
X_{42}
$X_{10}, X_{30}, X_{48}, X_{71}, X_{129}, X_{400}$
$X_3, X_6, X_{11}, X_{139}$
X_{39}, X_{62}
X_{176}
$X_{56}, X_{410}, X_{417}, X_{2830}$

Figure 8. Venn diagram on the variables recruited using three different methods: DC-SIS with χ^2-test in Z. Chen et al. (2018), PC-Knockoff procedure in Liu et al. (2022) and the proposed L_0-LPRE.

Table 6. The mean and standard deviation of the R^2 for the training and test set over 200 replications for the supermarket data.

	Training R^2		Test R^2	
Method	Mean	SD	Mean	SD
DC-SIS with χ^2-test	0.8540	0.0037	0.8451	0.0240
PC-Knockoff	0.8686	0.0041	0.8561	0.0275
L_0-LPRE	0.9024	0.0153	0.8702	0.0261
LPRE with 11 explored variables	**0.9053**	**0.0027**	**0.8983**	**0.0177**

Funding

Xia's work was supported by the Fundamental Research Funds for the Central Universities [grant number 2021CDJQY-047] and National Natural Science Foundation of China [grant number 11801202].

ORCID

Xiaochao Xia http://orcid.org/0000-0002-9414-355X

References

Bühlmann, P., Kalisch, M., & Meier, L. (2014). High-dimensional statistics with a view towards applications in biology. *Annual Review of Statistics and Its Applications*, *1*(1), 255–278.

Cao, Y., Kang, L., Li, X., Liu, Y., Luo, Y., & Yang, Q. (2023). Newton-Raphson meets sparsity: Sparse learning via a novel penalty and a fast solver. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(9), 11057–12067.

Chen, Z., Fan, J., & Li, R. (2018). Error variance estimation in ultrahigh dimensional additive models. *Journal of the American Statistical Association*, *113*(521), 315–324.

Chen, X., Ge, D., Wang, Z., & Ye, Y. (2014). Complexity of unconstrained L_2-L_p minimization. *Mathematical Programming*, *143*(1-2), 371–383.

Chen, K., Guo, S., Lin, Y., & Ying, Z. (2010). Least absolute relative error estimation. *Journal of the American Statal Association*, *105*(491), 1104–1112.

Chen, K., Lin, Y., Wang, Z., & Ying, Z. (2016). Least product relative error estimation. *Journal of Multivariate Analysis*, *144*, 91–98.

Chen, Y., Liu, H., & Ma, J. (2022). Local least product relative error estimation for single-index varying-coefficient multiplicative model with positive responses. *Journal of Computational and Applied Mathematics*, *415*, 114478.

Chen, Y., Ming, H., & Yang, H. (2024). Efficient variable selection for high-dimensional multiplicative models: A novel LPRE-based approach. *Statistical Papers*, *65*(6), 3713–3737.

Cheng, C., Feng, X., Huang, J., Jiao, Y., & Zhang, S. (2022). L_0-regularized high-dimensional accelerated failure time model. *Computational Statistics and Data Analysis*, *170*, 107430.

Do, H., Cheon, M., & Kim, S. (2020). Graph structured sparse subset selection. *Information Sciences*, *518*, 71–94.

Fan, Q., Jiao, Y., & Lu, X. (2014). A primal dual active set algorithm with continuation for compressed sensing. *IEEE Transactions on Signal Processing*, *62*(23), 6276–6285.

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, *96*(456), 1348–1360.

Fan, J., & Lv, J. (2008). Sure independence screening for ultra-high dimensional feature space. *Journal of the Royal Statistical Society Series B*, *70*(5), 849–911.

Hao, M., Lin, Y., & Zhao, X. (2016). A relative error-based approach for variable selection. *Computational Statistics and Data Analysis*, *103*, 250–262.

Hu, D. (2019). Local least product relative error estimation for varying coefficient multiplicative regression model. *Acta Mathematicae Applicatae Sinica, English Series*, *35*(2), 274–286.

Huang, J., Jiao, Y., Jin, B., Liu, J., Liu, Y., & Yang, C. (2021). A unified primal dual active set algorithm for nonconvex sparse recovery. *Statistical Science*, *36*(2), 215–238.

Huang, J., Jiao, Y., Kang, L., & Liu, Y. (2021). Fitting sparse linear models under the sufficient and necessary condition for model identification. *Statistics and Probability Letters*, *168*, 108925.

Huang, J., Jiao, Y., Kang, L., Liu, J., Liu, Y., & Lu, X. (2022). GSDAR: A fast Newton algorithm for l_0 regularized generalized linear models with statistical guarantee. *Computational Statistics*, *37*(1), 507–533.

Huang, J., Jiao, Y., Liu, Y., & Lu, X. (2018). A constructive approach to L_0 penalized regression. *Journal of Machine Learning Research*, *19*(10), 1–37.

Huang, J., Jiao, Y., Lu, X., Shi, Y., Yang, Q., & Yang, Y. (2022). PSNA: A pathwise semismooth Newton algorithm for sparse recovery with optimal local convergence and oracle properties. *Signal Processing*, *194*, 108432.

Javanmard, A., & Montanari, A. (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *Journal of Machine Learning Research*, *15*(82), 2869–2909.

Li, P., Jiao, Y., Lu, X., & Kang, L. (2022). A data-driven line search rule for support recovery in high-dimensional data analysis. *Computational Statistics and Data Analysis*, *174*, 107524.

Li, X., Shi, Z., & Leung, C. (2022). Sparse index tracking with K-sparsity or ϵ-deviation constraint via ℓ_0-norm minimization. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(12), 10930–10943.

Liu, W., Ke, Y., Liu, J., & Li, R. (2022). Model-free feature screening and FDR control with knockoff features. *Journal of the American Statistical Association*, *117*(537), 428–443.

Liu, H., & Xia, X. (2018). Estimation and empirical likelihood for single-index multiplicative models. *Journal of Statistical Planning and Inference*, *193*, 70–88.

Ming, H., Liu, H., & Yang, H. (2022). Least product relative error estimation for identification in multiplicative additive models. *Journal of Computational and Applied Mathematics*, *404*, 113886.

Ming, H., & Yang, H. (2024a). A fast robust best subset regression. *Knowledge-Based Systems*, *284*, 111309.

Ming, H., & Yang, H. (2024b). L_0 regularized regularized logistic regression for large-scale data. *Pattern Recognition*, *146*, 110024.

Natarajan, B. (1995). Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, *24*(2), 227–234.

Shi, Y., Huang, J., Jiao, Y., & Yang, Q. (2020). A semismooth Newton algorithm for high-dimensional nonconvex sparse learning. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(8), 2993–3006.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, *58*(1), 267–288.

Vershynin, R. (2018). *High-dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press.

Wang, H. (2009). Forward regression for ultra-high dimensional variable screening. *Journal of the American Statistical Association*, *104*(488), 1512–1524.

Wang, L., Kim, Y., & Li, R. (2013). Calibrating non-convex penalized regression in ultra-high dimension. *The Annals of Statistics*, *41*(5), 2505–2536.

Wang, Z., Liu, W., & Lin, Y. (2015). A change-point problem in relative error-based regression. *Test*, *24*(4), 835–856.

Wen, C., Li, Z., Dong, R., Ni, Y., & Pan, W. (2023). Simultaneous dimension reduction and variable selection for multinomial logistic regression. *INFORMS Journal on Computing*, *35*(5), 1044–1060.

Wen, C., Wang, X., & Zhang, A. (2023). 0 trend filtering. *INFORMS Journal on Computing*, *35*(6), 1491–1510.

Wen, C., Zhang, A., Quan, S., & Wang, X. (2020). BeSS: An R package for best subset selection in linear, logistic and cox proportional hazards models. *Journal of Statistical Software*, *94*(4), 1–24.

Zhang, C. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, *38*(2), 894–942.

Zhang, J., Feng, Z., & Peng, H. (2018). Estimation and hypothesis test for partial linear multiplicative models. *Computational Statistics and Data Analysis*, *128*, 87–103.

Zhang, J., Lin, B., & Yang, Y. (2022). Maximum nonparametric kernel likelihood estimation for multiplicative linear regression model. *Statistical Papers*, *63*(3), 885–918.

Zhang, J., Zhu, J., & Feng, Z. (2019). Estimation and hypothesis test for single-index multiplicative models. *Test*, *28*(1), 242–268.

Zhang, Y., Zhu, J., Zhu, J., & Wang, X. (2023). A splicing approach to best subset of groups selection. *INFORMS Journal on Computing*, *35*(1), 104–119.

Zhao, P., Yang, Y., & He, Q. (2022). High-dimensional linear regression via implicit regularization. *Biometrika*, *109*(4), 1033–1046.

Zheng, Z., Zhang, J., & Li, Y. (2022). L_0-regularized learning for high-dimensional additive hazards regression. *INFORMS Journal on Computing*, *34*(5), 2762–2775.

Zhou, S., Pan, L., & Xiu, N. (2021). Newton method for L_0-regularized optimization. *Numerical Algorithms*, *88*(4), 1541–1570.

Zhu, J., Wen, C., Zhu, J., & Wang, X. (2020). A polynomial algorithm for best-subset selection problem. *Proceedings of the National Academy of Sciences of the United States of America*, *117*(52), 33117–33123.

Appendix

A.1 Proof of Lemma 3.1

Proof: As in Huang et al. (2018) and Ming and Yang (2024b), we assume that $\hat{\boldsymbol{\beta}}$ is the global minimum of $L_\lambda(\boldsymbol{\beta})$. Then for β_j, we have

$$\begin{aligned} L_\lambda(\hat{\boldsymbol{\beta}}_{-j}, \beta_j) &\simeq L(\hat{\boldsymbol{\beta}}_{-j}, \hat{\beta}_j) + \frac{\partial L(\hat{\boldsymbol{\beta}}_{-j}, \beta_j)}{\partial \beta_j}|_{\hat{\beta}_j}(\beta_j - \hat{\beta}_j) \\ &\quad + \frac{1}{2}\hat{g}_j(\beta_j - \hat{\beta}_j)^2 + \lambda|\beta_j|_0 \\ &\simeq \frac{1}{2}\hat{g}_j\left(\beta_j - (\hat{\beta}_j + \hat{d}_j)\right)^2 + \lambda|\beta_j|_0, \end{aligned} \tag{A1}$$

where $\hat{g}_j = \frac{\partial^2 L(\hat{\boldsymbol{\beta}}_{-j},\beta_j)}{\partial^2 \beta_j}|_{\hat{\beta}_j}$ and $\hat{d}_j = -\hat{g}_j^{-1}\frac{\partial L(\hat{\boldsymbol{\beta}}_{-j},\beta_j)}{\partial \beta_j}|_{\hat{\beta}_j}$. Hence, we have

$$\hat{A} = \left\{ j \in [p_n] | \sqrt{\hat{g}_j}|\hat{\beta}_j + \hat{d}_j| \geq \sqrt{2\lambda} \right\},$$

$$\hat{I} = \left\{ j \in [p_n] | \sqrt{\hat{g}_j}|\hat{\beta}_j + \hat{d}_j| < \sqrt{2\lambda} \right\}.$$

With the hard threshold operator $H_\lambda(\cdot)$, we have

$$\begin{cases} \hat{\boldsymbol{\beta}}_{\hat{I}} = \mathbf{0}, \\ \hat{\boldsymbol{d}}_{\hat{A}} = \mathbf{0}, \\ \hat{\boldsymbol{\beta}}_{\hat{A}} \in \underset{\boldsymbol{\beta}_{\hat{A}}}{\operatorname{argmin}} L(\boldsymbol{\beta}_{\hat{A}}), \\ \hat{d}_j = -\left(\frac{\partial^2 L(\hat{\boldsymbol{\beta}}_{-j}, \beta_j)}{\partial^2 \beta_j}|_{\hat{\beta}_j}\right)^{-1} \frac{\partial L(\hat{\boldsymbol{\beta}}_{-j}, \beta_j)}{\partial \beta_j}|_{\hat{\beta}_j}, \quad j \in \hat{I}, \\ \hat{g}_j = \frac{\partial^2 L(\hat{\boldsymbol{\beta}}_{-j}, \beta_j)}{\partial^2 \beta_j}|_{\hat{\beta}_j}, \quad j \in [p_n]. \end{cases}$$

Assume $\boldsymbol{h} \in \mathbb{R}^{p_n}$ is small enough with $\max_{1 \le j \le p_n} \sqrt{\hat{g}_j}|h_j| < \sqrt{2\lambda}$. Then we will show $L_\lambda(\hat{\boldsymbol{\beta}} + \boldsymbol{h}) > L_\lambda(\hat{\boldsymbol{\beta}})$ in two cases, respectively.

Case 1: $\boldsymbol{h}_{\hat{I}} \neq \mathbf{0}$. Since $\sqrt{\hat{g}_j}|\hat{\beta}_j| \geq \sqrt{2\lambda}$ for $j \in \hat{A}$ and $\max_{1 \le j \le p_n} \sqrt{\hat{g}_j}|h_j| < \sqrt{2\lambda}$, we have

$$\lambda\|\hat{\boldsymbol{\beta}} + \boldsymbol{h}\|_0 - \lambda\|\hat{\boldsymbol{\beta}}\|_0 = \lambda\|\hat{\boldsymbol{\beta}}_{\hat{A}} + \boldsymbol{h}_{\hat{A}}\|_0 + \lambda\|\boldsymbol{h}_{\hat{I}}\|_0 - \lambda\|\hat{\boldsymbol{\beta}}_{\hat{A}}\|_0 = \lambda\|\boldsymbol{h}_{\hat{I}}\|_0 \geq \lambda.$$

Then, we can obtain

$$\begin{aligned} L_\lambda(\hat{\boldsymbol{\beta}} + \boldsymbol{h}) - L_\lambda(\hat{\boldsymbol{\beta}}) &= \frac{1}{n}\sum_{i=1}^{n}\left\{ y_i^{-1}\exp(\boldsymbol{X}_i^\top(\hat{\boldsymbol{\beta}} + \boldsymbol{h})) + y_i\exp(-\boldsymbol{X}_i^\top(\hat{\boldsymbol{\beta}} + \boldsymbol{h}))\right\} \\ &\quad - \frac{1}{n}\sum_{i=1}^{n}\left\{ y_i^{-1}\exp(\boldsymbol{X}_i^\top\hat{\boldsymbol{\beta}}) + y_i\exp(-\boldsymbol{X}_i^\top\hat{\boldsymbol{\beta}})\right\} + \lambda\|\boldsymbol{h}_{\hat{I}}\|_0 \\ &\geq m(\boldsymbol{h}) + \lambda, \end{aligned}$$

where

$$\begin{aligned} m(h) = \frac{1}{n}\sum_{i=1}^{n}\Big\{ & y_i^{-1}(\exp(\boldsymbol{X}_i^\top(\hat{\boldsymbol{\beta}} + \boldsymbol{h})) - \exp(\boldsymbol{X}_i^\top\hat{\boldsymbol{\beta}})) \\ & + y_i(\exp(-\boldsymbol{X}_i^\top(\hat{\boldsymbol{\beta}} + \boldsymbol{h})) - \exp(-\boldsymbol{X}_i^\top\hat{\boldsymbol{\beta}}))\Big\} \end{aligned}$$

is a continuous function in $\boldsymbol{h}$. Thus, we have $m(\boldsymbol{h}) + \lambda > 0$ because $\boldsymbol{h}$ is small enough.

Case 2: $\boldsymbol{h}_{\hat{I}} = \mathbf{0}$. Similar to Case 1, we have

$$\lambda\|\hat{\boldsymbol{\beta}} + \boldsymbol{h}\|_0 - \lambda\|\hat{\boldsymbol{\beta}}\|_0 = \lambda\|\hat{\boldsymbol{\beta}}_{\hat{A}} + \boldsymbol{h}_{\hat{A}}\|_0 - \lambda\|\hat{\boldsymbol{\beta}}_{\hat{A}}\|_0 = 0.$$

Therefore, we can obtain

$$\begin{aligned}
L_\lambda(\hat{\boldsymbol\beta}+\boldsymbol h)-L_\lambda(\hat{\boldsymbol\beta}) &= \frac{1}{n}\sum_{i=1}^{n}\left\{y_i^{-1}\exp(\boldsymbol X_i^\top(\hat{\boldsymbol\beta}+\boldsymbol h))+y_i\exp(-\boldsymbol X_i^\top(\hat{\boldsymbol\beta}+\boldsymbol h))\right\}\\
&\quad -\frac{1}{n}\sum_{i=1}^{n}\left\{y_i^{-1}\exp(\boldsymbol X_i^\top\hat{\boldsymbol\beta})+y_i\exp(-\boldsymbol X_i^\top\hat{\boldsymbol\beta})\right\}\\
&= \frac{1}{n}\sum_{i=1}^{n}\left\{y_i^{-1}\exp(\boldsymbol X_{i\hat A}^\top(\hat{\boldsymbol\beta}_{\hat A}+\boldsymbol h_{\hat A})+y_i\exp(-\boldsymbol X_{i\hat A}^\top(\hat{\boldsymbol\beta}_{\hat A}+\boldsymbol h_{\hat A})\right\}\\
&\quad -\frac{1}{n}\sum_{i=1}^{n}\left\{y_i^{-1}\exp(\boldsymbol X_{i\hat A}^\top\hat{\boldsymbol\beta}_{\hat A})+y_i\exp(-\boldsymbol X_{i\hat A}^\top\hat{\boldsymbol\beta}_{\hat A})\right\}\\
&= L(\hat{\boldsymbol\beta}_{\hat A}+\boldsymbol h_{\hat A})-L(\hat{\boldsymbol\beta}_{\hat A})\geq 0,
\end{aligned}$$

as $\hat{\boldsymbol\beta}_{\hat A}\in\underset{\boldsymbol\beta_{\hat A}}{\operatorname{argmin}}L(\boldsymbol\beta_{\hat A})$. Therefore, $\hat{\boldsymbol\beta}$ is a local minimizer of $L_\lambda(\boldsymbol\beta)$. ■

To obtain Theorems 3.2–3.4, we initially present several lemmas. The proofs for these lemmas can be found in Ming and Yang (2024b).

Lemma A.1: *Assuming condition* (C1) *holds and* $\|\boldsymbol\beta^*\|_0\leq T$. *Then we can derive the following:*

$$\kappa\|\nabla_{B^{(k)}}L(\boldsymbol\beta^{(k)})\|_1\|\nabla_{B^{(k)}}L(\boldsymbol\beta^{(k)})\|_\infty\geq 2L\zeta[L(\boldsymbol\beta^{(k)})-L(\boldsymbol\beta^*)],$$

where $B^{(k)}=A^{(k)}\backslash A^{(k-1)}$, $\zeta=\frac{|B^{(k)}|}{|B^{(k)}|+|A^*\backslash A^{(k-1)}|}$ *and* $\kappa\geq 1$ *is a constant.*

Lemma A.2: *Under the assumption that condition* (C1) *holds for* $0<U<\frac{1}{\tau\sqrt{T}}$ *and* $\|\boldsymbol\beta^*\|_0\leq T$ *in Algorithm* 2, *with* $1\leq\kappa<\infty$ *being a universal constant, it can be concluded that before Algorithm* 2 *terminates, we have*

$$L(\boldsymbol\beta^{(k+1)})-L(\boldsymbol\beta^*)\leq\xi[L(\boldsymbol\beta^{(k)})-L(\boldsymbol\beta^*)],$$

where $\xi=1-\frac{2\tau L(1-\tau\sqrt{T}U)}{\sqrt{T}\kappa(1+q)}\in(0,1)$.

Lemma A.3: *For all* $k\geq 0$, *if Lemma* A.2 *holds, then we have*

$$\|\boldsymbol\beta^{(k)}-\boldsymbol\beta^*\|_\infty\leq\sqrt{(T+q)\left(1+\frac{U}{L}\right)}(\sqrt{\xi})^k\|\boldsymbol\beta^{(0)}-\boldsymbol\beta^*\|_\infty+\frac{2}{L}\|\nabla L(\boldsymbol\beta^*)\|_\infty.$$

Lemma A.4 (General Hoeffding's inequality (see Theorem 2.6.3, Vershynin (2018)**)):** *If* $Y_1,Y_2,\ldots,Y_n$ *are independent sub-Gaussian random variables with mean zero, then for every* $t\geq 0$, *we can conclude*

$$P\left(\left|\sum_{i=1}^{n}a_iY_i\right|\geq t\right)\leq 2\exp\left(-\frac{ct^2}{K^2\|\boldsymbol a\|_2^2}\right),$$

where $\boldsymbol a=(a_1,a_2,\ldots,a_n)^\top\in\mathbb{R}^n$ *and* $K=\max_i\|Y_i\|_{\psi_2}$.

A.2 Proof of Theorem 3.2

Proof: By Lemmas A.1–A.3, and $\boldsymbol\beta^{(0)}=\boldsymbol 0$, we have

$$\|\boldsymbol\beta^{(k)}-\boldsymbol\beta^*\|_\infty\leq\sqrt{(T+q)\left(1+\frac{U}{L}\right)}(\sqrt{\xi})^k\|\boldsymbol\beta^*\|_\infty+\frac{2}{L}\|\nabla L(\boldsymbol\beta^*)\|_\infty.$$

■

A.3 Proof of Theorem 3.3

Proof: Let $Y_j=\frac{1}{n}\sum_{i=1}^{n}X_{ij}\tilde\varepsilon_i, j\in[p_n]$, and then by condition (C2) and Lemma A.4, for $t\geq 0$, we have

$$\begin{aligned}
P\left(\|\nabla L(\boldsymbol\beta^*)\|_\infty\geq t\right) &\leq\sum_{j=1}^{p_n}P\left(|Y_j|\geq t\right)\\
&=\sum_{j=1}^{p_n}P\left(\left|\sum_{i=1}^{n}X_{ij}\tilde\varepsilon_i\right|\geq nt\right)\\
&\leq 2p_n\exp\left(-cnt^2/K^2\right).
\end{aligned}$$

Hence, by taking $t = c_1\sqrt{\log(p_n)/n}$ and $c_2 = 2$, we can obtain

$$\log(p_n) - \frac{cc_1^2}{K^2}\log(p_n) = -c_3\log(p_n),$$

where $c_3 = cc_1^2/K^2 - 1 > 0$ for a large c_1. It implies that

$$P\left(\|\nabla L(\boldsymbol{\beta}^*)\|_\infty \geq c_1\sqrt{\frac{\log(p_n)}{n}}\right) \leq c_2\exp(-c_3\log(p_n)).$$

Furthermore, from Theorem 3.2, we have

$$\|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^*\|_\infty \leq \sqrt{(T+q)\left(1+\frac{U}{L}\right)}(\sqrt{\xi})^k\|\boldsymbol{\beta}^*\|_\infty + \frac{2c_1}{L}\sqrt{\frac{\log(p_n)}{n}},$$

with probability at least $1 - c_2\exp(-c_3\log(p_n))$. Moreover, if $k \geq O\left(\log_{\frac{1}{\xi}}\frac{n}{\log(p_n)}\right)$, then we have

$$\|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^*\|_\infty \leq O\left(\sqrt{\frac{\log(p_n)}{n}}\right). \qquad \blacksquare$$

A.4 Proof of Theorem 3.4

Proof: Based on Theorem 3.3 and condition (C3), with some algebraic manipulation, we can demonstrate that if $k > \log_{\frac{1}{\xi}}(9(T+q)\left(1+\frac{U}{L}\right)r^2)$,

$$\|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^*\|_\infty \leq \sqrt{(T+q)\left(1+\frac{U}{L}\right)}(\sqrt{\xi})^k\|\boldsymbol{\beta}^*\|_\infty + \frac{2}{3}\|\boldsymbol{\beta}^*_{A*}\|_{\min} < \|\boldsymbol{\beta}^*_{A*}\|_{\min}. \qquad \blacksquare$$