Journal of Computational and Applied Mathematics 424 (2023) 115007

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

An efficient semi-proximal ADMM algorithm for low-rank and sparse regularized matrix minimization problems with real-world applications[☆]

Wentao Qu [a], Xianchao Xiu [b], Haifei Zhang [c], Jun Fan [d,*]

[a] School of Mathematics and Statistics, Beijing Jiaotong University, 100044, Beijing, China
[b] School of Mechatronic Engineering and Automation, Shanghai University, 200444, Shanghai, China
[c] Tsinghua University High School-Daxing, 102629, Beijing, China
[d] Institute of Mathematics, Hebei University of Technology, 300401, Tianjin, China

ARTICLE INFO

Article history:
Received 1 May 2022
Received in revised form 3 December 2022

MSC:
49M05
90C25
65K05

Keywords:
Low-rank and sparse decomposition
Matrix minimization
Signal processing
Theoretical analysis

ABSTRACT

With the development of technology and the arrival of the era of big data, a large number of complex data structures have been generated, which makes matrix minimization become important and necessary. However, in the process of analyzing those data, there exist some latent structures such as low-rank and sparse decomposition. To address this issue, in this paper, we propose and study a novel low-rank and sparse regularized matrix minimization. Then an efficient semi-proximal alternating direction method of multipliers (sPADMM) is developed from the dual perspective. In theoretical analysis, it has proved that the sequence generated by the sPADMM converges to a global minimizer. Furthermore, the non-asymptotic statistical property is derived which suggests that the gap between the corresponding estimators and the true values is bounded with high probability and their consistency is guaranteed under rather weak regularity conditions. We also prove that the sequence generated by the proposed algorithm converges to the true value with high probability. Finally, extensive numerical experiments are conducted to verify the superiority of the proposed method in wide applications such as signal processing, image reconstruction, and video denoising.

1. Introduction

During the past few years, more and more complex data, possibly in the form of matrices rather than vectors, have emerged in various research fields including information technology, medical imaging, and industrial engineering [1]. A famous example is the investigation of an electroencephalography (EEG for short) dataset, which consists of 122 subjects divided into two groups, the alcoholic group and the normal control group. Voltage values are measured at 256 time points from 64 electrode channels placed on the subject's scalp, and thus each sample is a matrix of size 256 × 64. In order to characterize the scientific question arising from those data, Zhou and Li [2] introduced the matrix minimization (MM) model and proposed a regularized method which enforces the measurement matrix to be low-rank with nuclear norm penalty. Furthermore, the authors developed an efficient algorithm with an explicit and non-asymptotic convergence rate.

[☆] This work was supported in part by the National Natural Science Foundation of China under Grant 12071022, Grant 12001019, and Grant 12271022; and in part by the Natural Science Foundation of Hebei Province, China under Grant A2019202135.

[*] Corresponding author.
E-mail addresses: wtqu96@bjtu.edu.cn (W. Qu), xcxiu@shu.edu.cn (X. Xiu), 17121580@bjtu.edu.cn (H. Zhang), junfan@hebut.edu.cn (J. Fan).

https://doi.org/10.1016/j.cam.2022.115007
0377-0427/© 2022 Elsevier B.V. All rights reserved.

In [3–8], MM has received extensive attention in academia and also has been successfully applied to pattern recognition and machine learning.

However, there exist low-rank and sparse priors for many learning problems. For instance, in [9–11], the unknown data matrix of multi-task learning has the low-rank plus sparse decomposition. In [12–14], the measurement data of surveillance videos can be (approximately) decomposed into a low-rank background matrix representing spatiotemporal correlations and a sparse foreground matrix representing dynamic interests. The third example is industrial process monitoring in which clean process information is low-rank and minor faults appear to be sparse [15–17]. In fact, the above three examples could be summarized as low-rank and sparse matrix decomposition-related problems. Up to now, many researchers have made a lot of contributions to the theory, algorithm, and application, for example [18–23]. In theory, Chandrasekaran et al. [18] profoundly considered the concept of rank-sparse incoherence to measure the recognition capability of low-rank and sparse decomposition recovery. Candès et al. [19] proposed a robust principal component analysis to reconstruct low-rank matrices from highly corrupted measurements. The important idea behind is that one can adopt ℓ_1-norm to represent sparsity and nuclear norm to represent low-rank [20]. In addition, there comes another two critical questions to be settled: whether the proposed method is sufficient to estimate the true values of parameters and how to calculate the estimated values. To answer the first question correctly, the statistical performance of the corresponding estimator needs to be evaluated. In statistics, the performance of any estimator is usually examined in terms of its consistency. An estimator of a given parameter is said to be consistent if it converges with probability to the true value as the sample size goes to infinity. Meinshausen et al. [21] showed that the ℓ_1 regularized least squares estimator can be consistent under some conditions. Fan et al. [22] established the consistency of the ℓ_q $(0 < q < 1)$ regularized least squares estimator. Kong et al. [23] investigated the estimation consistency for the low-rank linear regression model under some mild conditions. For the second question, although a great number of fast solvers exist, the alternating direction method of multipliers (ADMM) is one of the most widely applied first-order optimization methods; see, e.g., [24–27]. In particular, Li et al. [27] proposed a semi-proximal ADMM (sPADMM) algorithm for solving sparse problems. Moreover, the generated sequence could be demonstrated to converge to an optimal solution under mild conditions. Very recently, Han [28] gave a survey on the latest developments about the ADMM. Note that when it comes to a convex problem with more than three blocks of variables, the ADMM may be divergent; see [25] for a counterexample. Although low-rank and spare structures have gained great success in many fields [29–31], it seems that the integration with MM has not been sufficiently investigated in the literature.

Inspired by the above observations, in this paper, we propose a novel low-rank and sparse decomposition regularized matrix minimization (LSDMM) model that can take full advantage of latent structures. On the optimization side, we develop an efficient algorithm based on semi-proximal ADMM (sPADMM) from the dual perspective and discuss the convergence property, which guarantees global convergence. On the statistical side, we establish the non-asymptotic statistical property for estimators, which guarantees high recovery accuracy. Furthermore, we drive that the error bound between the sequence generated by sPADMM and the true value converges in probability to zero. The contributions of this paper can be summarized in the following five aspects:

(1) A novel matrix regression model is proposed by low-rank and sparse decomposition for high-dimensional complex data analysis. To the best of our knowledge, this is the first time to integrate the low-rank and sparse decomposition into the matrix regression.
(2) An efficient optimization algorithm based on the ADMM is designed to solve the dual problem. Moreover, the global convergence is established.
(3) The statistical property of the parameter estimation for the proposed method is derived, which reveals that the true value can be approximated by the corresponding estimator with high probability, and their consistency condition is easily satisfied.
(4) Combining the global convergence and the non-asymptotic statistical property, the sequence generated by sPADMM is proved to converge to the true value with high probability.
(5) Extensive numerical experiments are conducted on the simulation data and real-world data, which demonstrate that the proposed method can achieve better performance than the existing state-of-the-art ones.

The rest of this paper is organized as follows. Section 2 introduces some basic preliminaries. The optimization algorithm and theoretical analysis are discussed in Section 3 and Section 4, respectively. Section 5 conducts extensive numerical experiments to show the effectiveness of the proposed method. Section 6 concludes this paper.

2. Preliminaries

2.1. Notation

Let $\mathbb{R}^n$ and $\mathbb{R}^{m\times q}$ be the set of n-dimensional vectors and $m\times q$ matrices, respectively. For a vector $x \in \mathbb{R}^n$, let x_i represent its ith entry. The ℓ_1 norm, ℓ_2 norm, and ℓ_∞ norm of the vector x are denoted by $\|x\|_1$, $\|x\|_2$, and $\|x\|_\infty$, respectively. For a matrix $X \in \mathbb{R}^{m\times q}$, X_{ij} represents its ij-th entry. The ℓ_1 norm, ℓ_2 norm, and nuclear norm of the matrix X are defined by $\|X\|_1 = \|vec(X)\|_1$, $\|X\|_2 = \max_i \sigma_i(X)$, and $\|X\|_* = \sum_{i=1}^{r} \sigma_i(X)$ respectively, where $\sigma_i(X)$ is ith singular value of X with $r = rank(X)$. For a matrix X, the ℓ_∞ norm is defined as the maximum absolute value of all element of X. For two vectors x and y (resp., two matrices X and Y) of the same size, their inner product is denoted by $\langle x, y\rangle = \sum_{i=1}^{n} x_i y_i$ (resp., $\langle X, Y\rangle = \sum_{i=1}^{m}\sum_{j=1}^{q} X_{ij}Y_{ij}$). In addition, $\mathrm{Diag}(x)$ or $\mathrm{Diag}(x_1, \ldots, x_m)$ denotes a square matrix with the vector x on its main diagonal and zero elsewhere. Finally, the largest eigenvalue of the matrix X is denoted by $\lambda_{\max}(X)$.

For nonsmooth function ℓ_1 norm, i.e., $\|x\|_1$, the sub-differential is defined as

$$\partial\|x\|_1 = \{\nu \mid \nu_i = \mathrm{sign}(x_i), i = 1, 2, \ldots, n\},$$

where sign denotes the following operator

$$\mathrm{sign}(x_i) = \begin{cases} 1 & \text{if } x_i > 0, \\ [-1, 1] & \text{if } x_i = 0, \\ -\ 1 & \text{if } x_i < 0. \end{cases}$$

The closed balls centered at point $x' \in \mathbb{R}^n$ with radius $r \geq 0$ based on ℓ_2 norm and ℓ_∞ norm are given by

$$\begin{aligned} \mathcal{B}_{\|\cdot\|_2(x';r)} &= \{x \in \mathbb{R}^n \mid \|x - x'\|_2 \leq r\}, \\ \mathcal{B}_{\|\cdot\|_\infty(x';r)} &= \{x \in \mathbb{R}^n \mid \|x - x'\|_\infty \leq r\}. \end{aligned}$$

For a nonempty convex set $\Omega \subset \mathbb{R}^n$, the indicator function is denoted by

$$\delta(x; \Omega) = \begin{cases} 0 & \text{if } x \in \Omega, \\ \infty & \text{otherwise.} \end{cases}$$

In addition, the distance function associated with set Ω is defined as

$$d(x; \Omega) = \inf\ \{\|x - \omega\|^2 \mid \omega \in \Omega\},$$

and the Euclidean projection from x to Ω is given by

$$\Pi(x; \Omega) = \{\omega \in \Omega \mid \|x - \omega\|^2 = d(x; \Omega)\}.$$

It is worth noting that the above definitions can be generalized to matrix cases if necessary.

2.2. Proximal mapping

To characterize the proposed algorithm, the proximal mappings are introduced. Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be a proper closed convex function such that for a given $\mu > 0$, the Moreau envelope function $\Phi_f(\cdot)$ of f is denoted by

$$\Phi_{f/\mu}(y) = \min_x\{f(x) + \frac{\mu}{2}\|x - y\|^2\}, \tag{2.1}$$

and the corresponding solution is called proximal mapping, which is given by

$$\mathrm{Prox}_{f/\mu}(y) = \operatorname*{argmin}_x\{f(x) + \frac{\mu}{2}\|x - y\|^2\}.$$

When $\mu = 1$, the optimization problem of (2.1) becomes

$$\min_x\{f(x) + \frac{1}{2}\|x - y\|^2\},$$

which has a unique solution as

$$\mathrm{Prox}_f(y) = \operatorname*{argmin}_x\{f(x) + \frac{1}{2}\|x - y\|^2\}.$$

In the following, some proximal mappings of special functions will be presented. More details can be found in [32,33] and references therein.

Proposition 2.1. *Let $f(x) = \delta(x; \Omega)$ with Ω being the nonempty closed convex set, then the corresponding proximal mapping of the indicator function $\delta(x; \Omega)$ just be the projection operator on the set Ω, that is,*

$$\begin{aligned} Prox_{\delta(x;\Omega)}(y) &= \operatorname*{argmin}_x\{\delta(x; \Omega) + \frac{\sigma}{2}\|x - y\|^2\} \\ &= \Pi(y; \Omega). \end{aligned}$$

Particularly, when $\Omega = \mathcal{B}_{\|\cdot\|_\infty(x';r)}$, the corresponding proximal mapping of $\delta(x; \Omega)$ is given by

$$\mathrm{Prox}_{\delta(x;\Omega)}(y) = x - \mathrm{sign}(x - x') \cdot \max\left\{|x - x'| - r, 0\right\}.$$

Proposition 2.2. *Let $f(x) = \lambda\|x\|_1$, then the corresponding proximal mapping of ℓ_1 norm is*

$$\begin{aligned} Prox_f(y) &= shrink(x, \lambda) \\ &= sign(x) \cdot \max\{|x| - \lambda, 0\}, \end{aligned}$$

see [34] for more details.

Proposition 2.3. *Let $f(X) = \|X\|_*$, then the corresponding proximal mapping of nuclear norm is*

$$\begin{aligned} Prox_f(Y) &= \underset{X}{\operatorname{argmin}}\{\|X\|_* + \frac{\mu}{2}\|X - Y\|_F^2\} \\ &= U_Y Diag(\hat{\xi}) V_Y^T \end{aligned}$$

with $\hat{\xi} = shrink(\xi, 1/\mu) = sign(\xi) \cdot \max\{|\xi| - 1/\mu, 0\}$, where U_Y, V_Y, Σ_Y are the singular value decomposition (SVD) of Y, i.e., $Y = U_Y \Sigma_Y V_Y^T$, and ξ is a vector containing the diagonal element of Σ_Y.

Proposition 2.4. *Let $\Omega = \mathcal{B}_{\|\cdot\|_2(0,\lambda)}$, then the corresponding proximal mapping of the indicator function is*

$$Prox_f(Y) = U_Y Diag(\hat{\xi}) V_Y^T$$

with $\hat{\xi} = \Pi(\xi; \mathcal{B}_{\|\cdot\|_\infty}(0, \lambda))$, where U_Y, V_Y, Σ_Y are the SVD of Y with ξ being the diagonal elements of Σ_Y.

3. Proposed algorithm

3.1. Model analysis

Recall the matrix regression in [2] as

$$y_i = \langle X_i, B\rangle + \langle z_i, \gamma\rangle + \varepsilon_i, \quad i = 1, \ldots, n,$$

where $y_i \in \mathbb{R}$ is the response, $X_i \in \mathbb{R}^{m\times q}$ is the design matrix and $z_i \in \mathbb{R}^p$ is the covariate vector, $B \in \mathbb{R}^{m\times q}$ and $\gamma \in \mathbb{R}^p$ are the unknown regression coefficients, $\varepsilon_i \in \mathbb{R}$ are the independent and identically distributed random variables with mean zero. For the convenience, denote $X = (vec(X_1), vec(X_2), \ldots, vec(X_n))^T \in \mathbb{R}^{n\times mq}$, $y = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^n$, and $Z = (z_1, z_2, \ldots, z_n)^T \in \mathbb{R}^{n\times p}$.

Considering that the true regression coefficient is usually or approximately low-rank plus sparse, we propose the following low-rank and sparse decomposition regularized MM (LSDMM) optimization problem

$$\begin{aligned} \min_{B,E,F,\gamma} \ & \frac{1}{2}\sum_{i=1}^{n}(y_i - \langle X_i, B\rangle - \langle z_i, \gamma\rangle)^2 + \lambda_1\|E\|_* + \lambda_2\|F\|_1 + \lambda_3\|\gamma\|_1 \\ \text{s.t.} \ & B = E + F, \end{aligned} \tag{3.1}$$

where λ_1, λ_2, and $\lambda_3 > 0$ are the tuning parameters. It should be pointed out that, when $E = 0$, $F = 0$, model (3.1) reduces to Lasso [35], and when $F = 0$, $\gamma = 0$, model (3.1) reduces to LMM [2]. Moreover, model (3.1) includes the minimization proposed by [36] as a special case, that is, when $F = 0$, and that model is referred as LSMM.

Let $\zeta_i = y_i - \langle X_i, B\rangle - \langle z_i, \gamma\rangle$ and $\zeta = (\zeta_1, \zeta_2, \ldots, \zeta_n)^T \in \mathbb{R}^n$, model (3.1) can be written as the form of

$$\begin{aligned} \min_{B,E,F,\gamma,\zeta} \ & \frac{1}{2}\sum_{i=1}^{n}\zeta_i^2 + \lambda_1\|E\|_* + \lambda_2\|F\|_1 + \lambda_3\|\gamma\|_1 \\ \text{s.t.} \ & \zeta_i = y_i - \langle X_i, B\rangle - \langle z_i, \gamma\rangle, \quad i = 1, \ldots, n, \\ & B = E + F. \end{aligned} \tag{3.2}$$

The Lagrangian function of (3.2) is

$$\begin{aligned} \mathcal{L}(B, E, F, \gamma, \zeta; b, V) =& \frac{1}{2}\sum_{i=1}^{n}\zeta_i^2 + \lambda_1\|E\|_* + \lambda_2\|F\|_1 + \lambda_3\|\gamma\|_1 \\ &+ \sum_{i=1}^{n} b_i(\zeta_i - y_i + \langle X_i, B\rangle + \langle z_i, \gamma\rangle) \\ &+ \langle V, B - E - F\rangle, \end{aligned}$$

where $b = (b_1, b_2, \dots, b_n)^T \in \mathbb{R}^n$ and $V \in \mathbb{R}^{m\times q}$ are the Lagrangian multipliers. Therefore, the dual optimization problem of (3.2) is achieved as follows

$$\begin{aligned} \min_{b,G,V,\omega}\ & \frac{1}{2}\sum_{i=1}^{n} b_i^2 + \sum_{i=1}^{n} b_i y_i + \delta(G; \mathcal{B}_{\|\cdot\|_2(0;\lambda_1)}) \\ & \quad + \delta(V; \mathcal{B}_{\|\cdot\|_\infty(0;\lambda_2)}) + \delta(\omega; \mathcal{B}_{\|\cdot\|_\infty(0;\lambda_3)}) \\ \text{s.t.}\ & \sum_{i=1}^{n} b_i X_i + V = 0, \\ & -\ G \qquad + V = 0, \\ & \sum_{i=1}^{n} b_i z_i \ \ + \omega = 0. \end{aligned} \tag{3.3}$$

It is easy to verify that the objective functions of the original problem (3.2) and dual problem (3.3) are proper closed convex functions, so the strong duality theorem holds. From the strong duality theorem, solving the original problem (3.2) is equivalent to solving the dual problem (3.3). In high-dimensional data settings ($n \ll \min\{p, mq\}$), it is necessary to calculate $X^TX \in \mathbb{R}^{mq\times mq}$ and $Z^TZ \in \mathbb{R}^{p\times p}$ for the problem (3.2). However, it only requires to compute $XX^T \in \mathbb{R}^{n\times n}$ and $ZZ^T \in \mathbb{R}^{n\times n}$ for the problem (3.3). Consequently, the computational load for solving the dual problem (3.3) can be greatly reduced. In the next subsection, how to efficiently solve problem (3.3) will be discussed.

Algorithm 1: sPADMM for solving (3.3)

```
Input: Measurements $X, Z$, parameters $\sigma > 0$, $\tau \in (0, \frac{1+\sqrt{5}}{2})$
Initialize: $(b^0, G^0, V^0, \omega^0; D_1^0, D_2^0, d_3^0)$
while not converged do
    Compute $b^{k+1}$ by optimizing
        $$\min_{b} \ \mathcal{L}_\sigma(b, G^k, V^k, \omega^k; D_1^k, D_2^k, d_3^k) + \frac{\sigma}{2}\|b - b^k\|_{S_1}^2;$$
    Compute $G^{k+1}$ by optimizing
        $$\min_{G} \ \mathcal{L}_\sigma(b^{k+1}, G, V^k, \omega^k; D_1^k, D_2^k, d_3^k) + \frac{\sigma}{2}\|G - G^k\|_{S_2}^2;$$
    Compute $V^{k+1}$ by optimizing
        $$\min_{V} \ \mathcal{L}_\sigma(b^{k+1}, G^{k+1}, V, \omega^k; D_1^k, D_2^k, d_3^k) + \frac{\sigma}{2}\|V - V^k\|_{T_1}^2;$$
    Compute $\omega^{k+1}$ by optimizing
        $$\min_{\omega} \ \mathcal{L}_\sigma(b^{k+1}, G^{k+1}, V^{k+1}, \omega; D_1^k, D_2^k, d_3^k) + \frac{\sigma}{2}\|\omega - \omega^k\|_{T_2}^2;$$
    Update multipliers $(D_1^{k+1}, D_2^{k+1}, d_3^{k+1})$ by
        $$\begin{cases} D_1^{k+1} = D_1^k + \tau\sigma(\sum_{i=1}^{n} b_i^{k+1} X_i + V^{k+1}), \\ D_2^{k+1} = D_2^k + \tau\sigma(-G^{k+1} + V^{k+1}), \\ d_3^{k+1} = d_3^k + \tau\sigma(\sum_{i=1}^{n} b_i^{k+1} z_i + \omega^{k+1}); \end{cases}$$
end
```

3.2. Semi-proximal ADMM algorithm

According to the semi-proximal ADMM (sPADMM) [27], problem (3.3) could be approximately solved by alternatively optimizing one variable when the other variables are determined. The augmented Lagrangian function of (3.3) is

$$\begin{aligned}\mathcal{L}_\sigma(b, G, V, \omega; D_1^k, D_2^k, d_3^k) = & \frac{1}{2}\sum_{i=1}^n b_i^2 + \sum_{i=1}^n b_i y_i + \delta(G; \mathcal{B}_{\|\cdot\|_2(0;\lambda_1)}) \\ & + \delta(V; \mathcal{B}_{\|\cdot\|_\infty(0;\lambda_2)}) + \delta(\omega; \mathcal{B}_{\|\cdot\|_\infty(0;\lambda_3)}) \\ & + \langle D_1, \sum_{i=1}^n b_i X_i + V\rangle + \frac{\sigma}{2}\|\sum_{i=1}^n b_i X_i + V\|_F^2 \\ & + \langle D_2, -G + V\rangle + \frac{\sigma}{2}\| - G + V\|_F^2 \\ & + \langle d_3, \sum_{i=1}^n b_i z_i + \omega\rangle + \frac{\sigma}{2}\|\sum_{i=1}^n b_i z_i + \omega\|^2,\end{aligned}$$

where $\sigma > 0$ is the penalty parameter, $D_1, D_2 \in \mathbb{R}^{m\times q}$ and $d_3 \in \mathbb{R}^p$ are the corresponding Lagrangian multipliers. The iterative framework for solving dual problem (3.3) is summarized in Algorithm 1, where the two positive definite operators S_i and T_i $(i = 1, 2)$ are introduced to ensure that these subproblems are easier to be computed.

- b-subproblem: Once G, V, ω are fixed, the optimization of variable b can be simplified as the form of

$$\begin{aligned}\min_b \ & \frac{1}{2}\|b\|^2 + \langle b, y\rangle + \langle Xvec(D_1^k), b\rangle \\ & + \frac{\sigma}{2}\|X^T b\|^2 + \sigma\langle Xvec(V^k), b\rangle + \langle Zd_3^k, b\rangle \\ & + \frac{\sigma}{2}\|Z^T b\|^2 + \sigma\langle Z\omega^k, b\rangle + \frac{\sigma}{2}\|b - b^k\|_{S_1}^2,\end{aligned} \tag{3.4}$$

where $S_1 = (t + 1)I - (XX^T + ZZ^T)$ with $I \in \mathbb{R}^{n\times n}$ being an identify matrix and t being $\lambda_{\max}(XX^T + ZZ^T)$. Therefore, the operator S_1 is positive definite. Denote $\bar{b} = -y - \sigma Xvec(V^k) - Xvec(D_1^k) - \sigma Z\omega^k - \sigma Zd_3^k + \sigma S_1 b^k$, hence it derives a solution of (3.4) as

$$b^{k+1} = \frac{1}{1 + \sigma t}\bar{b}.$$

- G-subproblem: By simple algebraic calculations, the optimization of variable G can be reformulated as

$$\begin{aligned}\min_G \ & \delta(G; \mathcal{B}_{\|\cdot\|_2(0;\lambda_1)}) + \langle D_2^k, -G + V^k\rangle \\ & + \frac{\sigma}{2}\| - G + V^k\|_F^2 + \frac{\sigma}{2}\|G - G^k\|_{S_2}^2.\end{aligned} \tag{3.5}$$

Here, assume that $S_2 = I \in \mathbb{R}^{m\times m}$ is an identity matrix. Hence, the closed-form solution of (3.5) is

$$G^{k+1} = \Pi(\frac{1}{2}(V^k + \frac{D_2^k}{\sigma} + G^k); \mathcal{B}_{\|\cdot\|_2(0;\lambda_1)}) = U\mathrm{Diag}(\hat{\xi})D^T$$

with $\hat{\xi} = \Pi(\xi, \mathcal{B}_{\|\cdot\|_\infty(0;\lambda_1)})$; see Proposition 2.4.

- V-subproblem: Once b, G are fixed, the optimization of variable V can be simplified to

$$\begin{aligned}\min_V \ & \delta(V; \mathcal{B}_{\|\cdot\|_\infty(0;\lambda_2)}) + \langle D_1^k, \sum_{i=1}^n b_i^{k+1} X_i + V^k\rangle \\ & + \langle D_2^k, -G^k + V^k\rangle + \frac{\sigma}{2}\|\sum_{i=1}^n b_i^{k+1} X_i + V^k\|_F^2 \\ & + \frac{\sigma}{2}\| - G^k + V^k\|_F^2 + \frac{\sigma}{2}\|V - V^k\|_{T_1}^2.\end{aligned} \tag{3.6}$$

It is assumed that $T_1 = I \in \mathbb{R}^{m\times m}$ is an identity matrix. Denote $\bar{V} = -\frac{1}{3}(\sum_{i=1}^n b_i^{k+1} X_i - G^{k+1} + \frac{D_1^k}{\sigma} + \frac{D_2^k}{\sigma} - V^k)$, hence the solution of (3.6) is

$$V^{k+1} = \Pi(\bar{V}; \mathcal{B}_{\|\cdot\|_\infty(0;\lambda_2)}) = \mathrm{shrink}(\bar{V}, \lambda_2).$$

- ω-subproblem: Similarly, the optimization of variable ω can be transformed to

$$\begin{aligned}\min_\omega \ & \delta(\omega; \mathcal{B}_{\|\cdot\|_\infty(0;\lambda_3)}) + \langle d_3^k, \sum_{i=1}^n b_i^{k+1} z_i + \omega\rangle \\ & + \frac{\sigma}{2}\|\sum_{i=1}^n b_i^{k+1} z_i + \omega\|_2^2 + \frac{\sigma}{2}\|\omega - \omega^k\|_{T_2}^2,\end{aligned} \tag{3.7}$$

where $T_2 = I \in \mathbb{R}^{p\times p}$ is an identity matrix. Denote $\bar{\omega} = \frac{1}{2}(-\sum_{i=1}^{n} b_i^{k+1} z_i - \frac{d_3^k}{\sigma} + \omega^k)$, hence the closed-form solution of (3.7) is

$$\omega^{k+1} = \Pi(\bar{\omega}; \mathcal{B}_{\|\cdot\|_\infty(0;\lambda_3)}) = \mathrm{shrink}(\bar{\omega}, \lambda_3).$$

According to the above discussion, all the resulting subproblems of Algorithm 1 with respect to variables b, G, V, ω can be efficiently solved. It is worth noting that many efforts have been made to improve the computational efficiency for low-rank optimization problems induced by nuclear norm in recent years (see [37–40]), while SVD is still an inevitable step. Although the proposed method does not necessarily reduce the computational load of SVD, it does greatly reduce the total computational load.

4. Algorithmic efficiency

4.1. Convergence analysis

Although there exist four variables in problem (3.3), one can define the following notations

$$\begin{aligned} &P = \begin{pmatrix} b & 0 & 0 \\ 0 & G & 0 \\ 0 & 0 & b \end{pmatrix}, \ Q = \begin{pmatrix} vec(V) & 0 & 0 \\ 0 & V & 0 \\ 0 & 0 & \omega \end{pmatrix}, \\ &f(P) = \frac{1}{2}\sum_{i=1}^{n} b_i^2 + \sum_{i=1}^{n} b_i y_i + \delta(G; \mathcal{B}_{\|\cdot\|_2(0;\lambda_1)}), \\ &\quad g(Q) = \delta(V; \mathcal{B}_{\|\cdot\|_\infty(0;\lambda_2)}) + \delta(\omega; \mathcal{B}_{\|\cdot\|_\infty(0;\lambda_3)}), \end{aligned} \tag{4.1}$$

and

$$\begin{aligned} &\mathcal{F}P = \begin{pmatrix} X^T b & 0 & 0 \\ 0 & -G & 0 \\ 0 & 0 & Z^T b \end{pmatrix} = \begin{pmatrix} X^T & 0 & 0 \\ 0 & -I & 0 \\ 0 & 0 & Z^T \end{pmatrix} \begin{pmatrix} b & 0 & 0 \\ 0 & G & 0 \\ 0 & 0 & b \end{pmatrix}, \\ &\mathcal{G}Q = \begin{pmatrix} vec(V) & 0 & 0 \\ 0 & V & 0 \\ 0 & 0 & \omega \end{pmatrix} = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} vec(V) & 0 & 0 \\ 0 & V & 0 \\ 0 & 0 & \omega \end{pmatrix}. \end{aligned} \tag{4.2}$$

According to (4.1) and (4.2), f and g are closed proper convex functions, $\mathcal{F}P$ and $\mathcal{G}Q$ are linear operators. Consequently, problem (3.3) can be reformulated as the standard convex optimization problem with two blocks

$$\begin{aligned} &\min_{P,Q} f(P) + g(Q) \\ &\ \text{s.t. } \mathcal{F}P + \mathcal{G}Q = 0. \end{aligned} \tag{4.3}$$

It is easy to verify that there exists $(P_0, Q_0) \in \mathrm{ri}(\mathrm{dom}\, f \times \mathrm{dom}\, g) \bigcap \Omega$, for example $(P_0, Q_0) = (0, 0)$, where Ω is the constraint set in (4.3). That is, (4.3) satisfies the Slater constraint conditions. Thus, it follows from [41, Corollary 28.2.2] and [41, Corollary 28.3.1] that $(\overline{P}, \overline{Q})$ is an optimal solution to problem (4.3) if and only if there exists a Lagrangian multiplier $\overline{W}$ such that

$$\mathcal{F}^*\overline{W} \in \partial f(\overline{Q}), \quad \mathcal{G}^*\overline{W} \in \partial g(\overline{Q}), \quad \mathcal{F}\overline{P} + \mathcal{G}\overline{Q} = 0,$$

where the ∂f and ∂g are the subdifferential of f and g, respectively. Moreover, any $\overline{W}$ satisfying is an optimal solution to the dual problem of (4.3). Due to the convexity of the function f and g, it follows from [32, Theorem 12.17] that both ∂f and ∂g are maximal monotone. Consequently, there exist two self-adjoint and positive semidefinite operators $\sum_f$ and $\sum_g$ such that for all $P, \hat{P} \in \mathrm{dom}\,(f)$, $K \in \partial f(P)$ and $\hat{K} \in \partial f(\hat{P})$, and for all $Q, \hat{Q} \in \mathrm{dom}\,(f)$, $L \in \partial g(Q)$ and $\hat{L} \in \partial g(\hat{Q})$,

$$\langle K - \hat{K}, P - \hat{P}\rangle \geq \|P - \hat{P}\|^2_{\sum_f}, \quad \langle L - \hat{L}, Q - \hat{Q}\rangle \geq \|Q - \hat{Q}\|^2_{\sum_g}.$$

Furthermore, the selection of positive definite operators S_i, T_i $(i = 1, 2)$ in Section 3.2 indicates that the operators $S = \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix}$ and $T = \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix}$ are both positive definite. Hence, both $\mathcal{F}^*\mathcal{F} + \sigma^{-1}\sum_f + S$ and $\mathcal{G}^*\mathcal{G} + \sigma^{-1}\sum_g + T$ are positive definite. Therefore, the global convergence results for the 2-block sPADMM can be established as follows.

Theorem 4.1. *Let $\{(P^k, Q^k; W^k)\}$ be the sequence generated by the semi-proximal ADMM Algorithm. If $\tau \in (0, \frac{1+\sqrt{5}}{2})$, then the sequence $\{(P^k, Q^k)\}$ converges to an optimal solution of problem (4.3) and $\{W^k\}$ converges to an optimal solution of the dual problem of (4.3).*

Remark 4.2. Since both $\mathcal{F}^*\mathcal{F} + \sigma^{-1}\sum_f + S$ and $\mathcal{G}^*\mathcal{G} + \sigma^{-1}\sum_g + T$ are positive definite, the global convergence of the sPADMM algorithm for solving problem (4.3) is easily satisfied. Moreover, the conclusions of Theorem 4.1 can follow directly from the results given in [42, Theorem B.1].

We now drive the convergence result of Algorithm 1. It follows from [41, Corollary 28.2.2] and [41, Corollary 28.3.1] that $(\overline{b}, \overline{G}, \overline{V}, \overline{\varpi})$ is an optimal solution to problem (3.3) if and only if there exists a Lagrangian multiplier $(\overline{D}_1, \overline{D}_2, \overline{d}_3)$ such that $(\overline{b}, \overline{G}, \overline{V}, \overline{\varpi}, \overline{D}_1, \overline{D}_2, \overline{d}_3)$ satisfies the following KKT system

$$\begin{aligned}
&0 = \overline{b}_i + y_i + X_i^T\overline{D}_1 + z_i^T d_3, 0 = -\overline{D}_2 + \varrho, 0 = \rho + \overline{D}_1 + \overline{D}_2, 0 = \varpi + \overline{d}3,\\
&\varrho \in \partial\delta(\overline{G}; \mathcal{B}_{\|\cdot\|_2(0;\lambda_1)}), \quad \rho \in \partial\delta(\overline{V}; \mathcal{B}_{\|\cdot\|_\infty(0;\lambda_2)}), \quad \varpi \in \partial\delta(\overline{\varpi}; \mathcal{B}_{\|\cdot\|_\infty(0;\lambda_3)}),\\
&\sum_{i=1}^{n} \overline{b}_i X_i + \overline{V} = 0, \quad -\overline{G} + \overline{V} = 0, \quad \sum_{i=1}^{n} \overline{b}_i z_i \ + \overline{\varpi} = 0.
\end{aligned} \tag{4.4}$$

Denote the solution set to the KKT system (4.4) by Ω. By the connection between (3.3) and (4.3), the global convergence of sPADMM algorithm for solving optimization problem (3.3) can be directly obtained from Theorem 4.1.

Theorem 4.3. *Let $\{(b^k, G^k, V^k, \omega^k, D_1^k, D_2^k, d_3^k)\}$ be the sequence generated by the proposed Algorithm 1. If $\tau \in (0, \frac{1+\sqrt{5}}{2})$, then $\{(b^k, G^k, V^k, \omega^k, D_1^k, D_2^k, d_3^k)\}$ converges to a point $(\overline{b}, \overline{G}, \overline{V}, \overline{\varpi}, \overline{D}_1, \overline{D}_2, \overline{d}_3)$ in the solution set Ω.*

4.2. Parameter estimation

Although the global convergence of the proposed algorithm has been established in the previous subsection, there comes another two critical issues to be settled. The first is whether the proposed method is sufficient to estimate the true parameters B^* and γ^*. The other is how to characterize the gap between the solution generated by Algorithm 1 and the optimal solution. For the first question, one can study the consistency of the corresponding estimators, which is usually used to evaluate the performance of an estimator in statistics. The consistent estimator tends to produce more and more accurate estimates of the given parameter as the sample size increases. Indeed, we study the non-asymptotic error bound between the estimators and the true values, which can provide more information than the consistency.

Let $(B^*, E^*, F^*, \gamma^*)$ be the true values in (3.1). To study the performance of finite sample for the proposed method, the following conditions are needed.

Condition 1. There exist two positive constants $\underline{C}_X$ and $\overline{C}_X$ such that

$$\underline{C}_X \|B\|_F^2 \le \frac{1}{n}\sum_{i=1}^{n}\langle X_i, B\rangle^2 \le \overline{C}_X \|B\|_F^2.$$

Condition 2. There exist two positive constants $\underline{C}_Z$ and $\overline{C}_Z$ that bound all eigenvalues of $n^{-1}Z^TZ$ from below and above, respectively.

Condition 3. Let the random error ε_i satisfy $\mathbb{E}(\varepsilon_i) = 0$ and obey sub-Gaussian, i.e., there exist two constants k and σ_0 such that $k^2[\mathbb{E}(e^{|\varepsilon_i|^2/k^2}) - 1] \le \sigma_0^2$.

Condition 4. Denote $r^* = \mathrm{rank}(E^*)$, $s_F^* = \|F^*\|_0$, and $s_\gamma^* = \|\gamma^*\|_0$, where $\|\cdot\|_0$ denotes the number of nonzero entries. Then

$$\begin{aligned}
&\lambda_1 \le n/(r^*\|E^*\|_2), \ \lambda_2 \le n/(s_F^*\|F^*\|_\infty),\\
&\lambda_3 \le n/(s_\gamma^*\|\gamma^*\|_\infty).
\end{aligned}$$

Here, Condition 1 can be regarded as a matrix version of restricted isometry condition [43], and Conditions 2-3 are often adopted for variable selection in high-dimensional linear regression problems [21]. We now establish the nonasymptotic property of the estimators $\hat{B}$ and $\hat{\gamma}$.

Theorem 4.4. *Assume that Conditions 1-4 are satisfied and*

$$\|\frac{1}{n}\sum_{i=1}^{n} z_i \otimes X_i\|_F \le \frac{1}{2}\min\{\underline{C}_X, \underline{C}_Z\}, \tag{4.5}$$

where the symbol $\otimes$ indicates Kronecker product. Then,

$$\begin{aligned}
&\|\hat{B} - B^*\|_F^2 + \|\hat{\gamma} - \gamma^*\|_2^2\\
&\le 8\sqrt{(K^2 + \sigma_0^2)(c+4)}\sqrt{\frac{(\overline{C}_X mq + \overline{C}_Z p)\log(1+2cn)}{n}}\\
&\quad + \frac{\lambda_1 r^*\|E^*\|_2 + \lambda_2 s_F^*\|F^*\|_\infty + \lambda_3 s_\gamma^*\|\gamma^*\|_\infty}{n}
\end{aligned} \tag{4.6}$$

with probability at least $1 - e^{-n\sigma_0^2/K^2} - 3e^{-mq\log(1+2cn)} - 3e^{-p\log(1+2cn)}$ where $c = \sqrt{\frac{8\left(3+2\sigma_0^2/K^2\right)}{\min\{\underline{C}_X, \underline{C}_Z\}}}$.

Proof. In our proof, we will divide the proof into two steps: Step 1. We prove that $\|\hat{B}-B^*\|_F^2+\|\hat{\gamma}-\gamma^*\|_2^2$ has a upper bound; Step 2. We prove that $\|\hat{B}-B^*\|_F^2+\|\hat{\gamma}-\gamma^*\|_2^2$ converges in probability to the upper bound in (4.6).

Step 1. According to the definition of $\hat{B}, \hat{E}, \hat{F}, \hat{\gamma}$, we obtain that

$$\begin{aligned}&\frac{1}{2}\sum_{i=1}^{n}(y_i-\langle X_i,\hat{B}\rangle-\langle z_i,\hat{\gamma}\rangle)^2+\lambda_1\|\hat{E}\|_*+\lambda_2\|\hat{F}\|_1+\lambda_3\|\hat{\gamma}\|_1\\&\le\frac{1}{2}\sum_{i=1}^{n}\varepsilon_i^2+\lambda_1\|E^*\|_*+\lambda_2\|F^*\|_1+\lambda_3\|\gamma^*\|_1,\end{aligned}$$

which yields that

$$\begin{aligned}&\frac{1}{2}\sum_{i=1}^{n}\Big(\varepsilon_i-\langle X_i,\hat{B}-B^*\rangle-\langle z_i,\hat{\gamma}-\gamma^*\rangle\Big)^2\\&\le\frac{1}{2}\sum_{i=1}^{n}\varepsilon_i^2+\lambda_1\|E^*\|_*+\lambda_2\|F^*\|_1+\lambda_3\|\gamma^*\|_1.\end{aligned}\tag{4.7}$$

Here $\|\cdot\|_*$ is the nuclear norm, defined as the sum of all singular values. Therefore,

$$\begin{aligned}&\frac{1}{2}\sum_{i=1}^{n}\big(\langle X_i,\hat{B}-B^*\rangle+\langle z_i,\hat{\gamma}-\gamma^*\rangle\big)^2\\&=\frac{1}{2}\sum_{i=1}^{n}\big(\langle X_i,\hat{B}-B^*\rangle+\langle z_i,\hat{\gamma}-\gamma^*\rangle-\varepsilon_i+\varepsilon_i\big)^2\\&\le\sum_{i=1}^{n}\big(\langle X_i,\hat{B}-B^*\rangle+\langle z_i,\hat{\gamma}-\gamma^*\rangle-\varepsilon_i\big)^2+\sum_{i=1}^{n}\varepsilon_i^2\\&\le 2\sum_{i=1}^{n}\varepsilon_i^2+2\lambda_1\|E^*\|_*+2\lambda_2\|F^*\|_1+2\lambda_3\|\gamma^*\|_1.\end{aligned}$$

By Cauchy's inequality and Condition 4, we have

$$\begin{aligned}&\lambda_1\|E^*\|_*+\lambda_2\|F^*\|_1+\lambda_3\|\gamma^*\|_1\\&\le\lambda_1(r^*\|E^*\|_2)+\lambda_2(|F^*|_\infty s_F^*)+\lambda_3(s_\gamma^*\|\gamma^*\|_\infty)\le 3n.\end{aligned}$$

By the above two inequalities, we obtain that

$$\frac{1}{2}\sum_{i=1}^{n}\big(\langle X_i,\hat{B}-B^*\rangle+\langle z_i,\hat{\gamma}-\gamma^*\rangle\big)^2\le 2\sum_{i=1}^{n}\varepsilon_i^2+6n.\tag{4.8}$$

By Chebyshev's inequality and Condition 3, we obtain that

$$\begin{aligned}\mathbb{P}\Big(\sum_{i=1}^{n}\varepsilon_i^2\ge 2n\sigma_0^2/K^2\Big)\le&e^{-2n\sigma_0^2/K^2}\mathbb{E}(e^{\sum_{i=1}^{n}\varepsilon_i^2/K^2})\\=&e^{-2n\sigma_0^2/K^2}\cdot e^{\sum_{i=1}^{n}\log\mathbb{E}(e^{\varepsilon_i^2/K^2})}\\\le&e^{-2n\sigma_0^2/K^2+n\log(1+\sigma_0^2/K^2)}\\\le&e^{-2n\sigma_0^2/K^2+n\sigma_0^2/K^2}\\=&e^{-n\sigma_0^2/K^2}.\end{aligned}$$

Notice that

$$\begin{aligned}\langle X_i,\hat{B}-B^*\rangle\langle z_i,\hat{\gamma}-\gamma^*\rangle=&\mathrm{trace}\Big(\big((\hat{\gamma}-\gamma^*)^T z_i\big)\otimes\big((\hat{B}-B^*)^T X_i\big)\Big)\\=&\mathrm{trace}\Big(\big((\hat{\gamma}-\gamma^*)^T\otimes(\hat{B}-B^*)^T\big)\big(z_i\otimes X_i\big)\Big)\\=&\mathrm{trace}\Big(\big((\hat{\gamma}-\gamma^*)\otimes(\hat{B}-B^*)\big)^T\big(z_i\otimes X_i\big)\Big)\end{aligned}$$

$$
\begin{aligned}
&=\langle z_i \otimes X_i, (\hat{\gamma}-\gamma^*)\otimes(\hat{B}-B^*)\rangle\\
&\le \|z_i \otimes X_i\|_F \|(\hat{\gamma}-\gamma^*)\otimes(\hat{B}-B^*)\|_F\\
&\le \|z_i \otimes X_i\|_F \|\hat{\gamma}-\gamma^*\|_2 \|\hat{B}-B^*\|_F.
\end{aligned}
$$

From the above inequality, Conditions 1-2 and (4.5), we conclude that

$$
\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{n}\big(\langle X_i, \hat{B}-B^*\rangle + \langle z_i, \hat{\gamma}-\gamma^*\rangle\big)^2\\
&= \frac{1}{2}\sum_{i=1}^{n}\langle X_i, \hat{B}-B^*\rangle^2 + \frac{1}{2}\sum_{i=1}^{n}\langle z_i, \hat{\gamma}-\gamma^*\rangle^2 + \sum_{i=1}^{n}\langle X_i, \hat{B}-B^*\rangle\langle z_i, \hat{\gamma}-\gamma^*\rangle\\
&\ge \frac{n}{2}\underline{C}_X\|\hat{B}-B^*\|_F^2 + \frac{n}{2}\underline{C}_Z\|\hat{\gamma}-\gamma^*\|^2\\
&\quad -n\|\frac{1}{n}\sum_{i=1}^{n}(z_i\otimes X_i)\|_F\cdot\|\hat{B}-B^*\|_F\cdot\|\hat{\gamma}-\gamma^*\|\\
&\ge \frac{n}{2}\min\{\underline{C}_X, \underline{C}_Z\}(\|\hat{B}-B^*\|_F^2+\|\hat{\gamma}-\gamma^*\|^2)\\
&\quad -\frac{n}{2}\min\{\underline{C}_X, \underline{C}_Z\}\cdot\frac{1}{2}(\|\hat{B}-B^*\|_F^2+\|\hat{\gamma}-\gamma^*\|^2)\\
&\ge \frac{n}{4}\min\{\underline{C}_X, \underline{C}_Z\}(\|\hat{B}-B^*\|_F^2+\|\hat{\gamma}-\gamma^*\|^2).
\end{aligned}
\tag{4.9}
$$

Combining (4.8) and (4.9), we have

$$
\|\hat{B}-B^*\|_F^2+\|\hat{\gamma}-\gamma^*\|_2^2 \le \frac{8}{\min\{\underline{C}_X, \underline{C}_Z\}}\cdot(3+\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^2)\le c \tag{4.10}
$$

with probability at least $1-e^{-n\sigma_0^2/K^2}$.

Step 2. By [21, Lemma 14.27], we can obtain that

$$
\{U\in\mathbb{R}^{m\times q}:\|U\|_F\le c\}\subseteq\bigcup_{j=1}^{m_n}\mathbf{B}(U_j, n^{-1}),
$$

where $U_j\in\mathbb{R}^{m\times q}$, $m_n=(1+2cn)^{mq}$ and $\mathbf{B}(U_j, n^{-1})=\{U\in\mathbb{R}^{m\times q}:\|U-U_j\|_F\le n^{-1}, \|U_j\|_F\le c\}$ and also by [21, Lemma 14.27], we can get that

$$
\{v\in\mathbb{R}^p:\|v\|_2\le c\}\subseteq\bigcup_{l=1}^{m_n'}\mathbf{B}'(v_l, n^{-1}),
$$

where $v_l\in\mathbb{R}^p$, $m_n'=(1+2cn)^p$ and $\mathbf{B}'(v_l, \frac{1}{n})=\{v\in\mathbb{R}^p:\|v-v_l\|_2\le n^{-1}, \|v_l\|_2\le c\}$.

Then

$$
\begin{aligned}
&\{U\in\mathbb{R}^{m\times q}, v\in\mathbb{R}^p:\|U\|_F^2+\|v\|_2^2\le c^2\}\\
&\subseteq\{U\in\mathbb{R}^{m\times q}, \|U\|_F\le c\}\bigcup\{v\in\mathbb{R}^p:\|v\|_2\le c\}\\
&\subseteq(\cup_{j=1}^{m_n}\mathbf{B}(U_j, n^{-1}))\bigcup(\cup_{l=1}^{m_n'}\mathbf{B}(v_l, n^{-1})).
\end{aligned}
$$

Therefore

$$
\begin{aligned}
&\sup_{\|U\|_F^2+\|v\|_2^2\le c^2, U\in\mathbb{R}^{m\times q}, v\in\mathbb{R}^p}|\frac{1}{n}\sum_{i=1}^{n}[\langle X_i, U\rangle+\langle z_i, v\rangle]\varepsilon_i|\\
&\le\max_{1\le j\le m_n, 1\le l\le m_n'}\sup_{U\in\mathbf{B}(U_j, \frac{1}{n}), v\in B'(v_l, \frac{1}{n})}\left|\frac{1}{n}\sum_{i=1}^{n}[\langle X_i, U_j\rangle+\langle z_i, v_l\rangle]\varepsilon_i\right.\\
&\quad\left.+\frac{1}{n}\sum_{i=1}^{n}[\langle X_i, U-U_j\rangle+\langle z_i, v-v_l\rangle]\varepsilon_i\right|\\
&\le\max_{1\le j\le m_n, 1\le l\le m_n'}\sup_{U\in\mathbf{B}(U_j, \frac{1}{n}), v\in\mathbf{B}'(v_l, \frac{1}{n})}\left|\frac{1}{n}\sum_{i=1}^{n}[\langle X_i, U_j\rangle+\langle z_i, v_l\rangle]\varepsilon_i\right|
\end{aligned}
$$

$$
\begin{aligned}
&+ \max_{1\le j\le m_n, 1\le l\le m_n'} \sup_{U\in \mathbf{B}(U_j,\frac{1}{n}), v\in \mathbf{B}'(v_l,\frac{1}{n})} \left|\frac{1}{n}\sum_{i=1}^{n}[\langle X_i, U-U_j\rangle + \langle z_i, v-v_l\rangle]\varepsilon_i\right| \\
&\le \max_{1\le j\le m_n}\left|\frac{1}{n}\sum_{i=1}^{n}\langle X_i, U_j\rangle\varepsilon_i\right| + \max_{1\le l\le m_n'}\left|\frac{1}{n}\sum_{i=1}^{n}\langle z_i, v_l\rangle\varepsilon_i\right| \\
&\quad + \max_{1\le j\le m_n}\sup_{U\in \mathbf{B}(U_j,\frac{1}{n})}\left|\frac{1}{n}\sum_{i=1}^{n}\langle X_i, U-U_j\rangle\varepsilon_i\right| + \max_{1\le l\le m'_n}\sup_{v\in \mathbf{B}'(v_l,\frac{1}{n})}\left|\frac{1}{n}\sum_{i=1}^{n}\langle z_i, v-v_l\rangle\varepsilon_i\right| \\
&\triangleq I_1 + I_2 + I_3 + I_4.
\end{aligned}
$$

We fist estimate I_1. Notice that for any $\tau_1 > 0$ and $\alpha > 0$, it follows that

$$
\begin{aligned}
\mathbb{P}(\frac{1}{n}\sum_{i=1}^{n}\langle X_i, U_j\rangle\varepsilon_i > \tau_1) &\le e^{-\alpha n\tau_1}\mathbb{E}(e^{\alpha\sum_{i=1}^{n}\langle X_i,U_j\rangle\varepsilon_i}) \\
&= e^{-\alpha n\tau_1}\prod_{i=1}^{n}\mathbb{E}(e^{\alpha\langle X_i,U_j\rangle\varepsilon_i}) \\
&\le e^{-\alpha n\tau_1}\prod_{i=1}^{n}e^{2\alpha^2\langle X_i,U_j\rangle^2(K^2+\sigma_0^2)} \\
&= e^{-\alpha n\tau_1+2\alpha^2\sum_{i=1}^{n}\langle X_i,U_j\rangle^2(K^2+\sigma_0^2)} \\
&\le e^{-\alpha n\tau_1+2\alpha^2 n\overline{C}_X\|U_j\|_F^2(K^2+\sigma_0^2)},
\end{aligned}
$$

where the first inequality dues to Chebyshev's inequality and the second inequality derives from [21, Condition 3] and [21, Lemma 14.3]. Taking $\alpha = \frac{n\tau_1}{4(K^2+\sigma_0^2)\overline{C}_X\|U_j\|_F^2}$, we have

$$
\mathbb{P}(\frac{1}{n}\sum_{i=1}^{n}\langle X_i, U_j\rangle\varepsilon_i > \tau_1) \le e^{-\frac{n\tau_1^2}{8(K^2+\sigma_0^2)\overline{C}_X\|U_j\|_F^2}}.
$$

Then,

$$
\begin{aligned}
\mathbb{P}(I_1 \ge \tau_1) &\le \mathbb{P}(\max_{1\le j\le m_n}|\frac{1}{n}\sum_{i=1}^{n}\langle X_i, U_j\rangle\varepsilon_i| > \tau_1) \\
&\le \sum_{j=1}^{m_n}\mathbb{P}(|\frac{1}{n}\sum_{i=1}^{n}\langle X_i, U_j\rangle\varepsilon_i| > \tau_1) \\
&\le \sum_{j=1}^{m_n}\mathbb{P}(\frac{1}{n}\sum_{i=1}^{n}\langle X_i, U_j\rangle\varepsilon_i > \tau_1) \\
&\qquad + \sum_{j=1}^{m_n}\mathbb{P}(\frac{1}{n}\sum_{i=1}^{n}\langle X_i, U_j\rangle(-\varepsilon_i) > \tau_1) \\
&\le 2m_n e^{-\frac{n\tau_1^2}{8(K^2+\sigma_0^2)\overline{C}_X\|U_j\|_F^2}} \\
&\le 2m_n e^{-\frac{n\tau_1^2}{8(K^2+\sigma_0^2)c^2\overline{C}_X}} \\
&= 2e^{-n(\frac{\tau_1^2}{8(K^2+\sigma_0^2)c^2\overline{C}_X}-\frac{mq\log(1+2cn)}{n})}.
\end{aligned}
$$

By taking $\tau_1 = 4\sqrt{\frac{(K^2+\sigma_0^2)c^2\overline{C}_X mq\log(1+2cn)}{n}}$, we then have

$$
\mathbb{P}\Big(I_1 \ge 4\sqrt{\frac{(K^2+\sigma_0^2)c^2\overline{C}_X mq\log(1+2cn)}{n}}\Big) \le 2e^{-mq\log(1+2cn)}. \tag{4.11}
$$

Similarly,

$$
\mathbb{P}\Big(I_2 \ge 4\sqrt{\frac{(K^2+\sigma_0^2)c^2\overline{C}_Z p\log(1+2cn)}{n}}\Big) \le 2e^{-p\log(1+2cn)}. \tag{4.12}
$$

To estimate I_3 and I_4, we denote $\theta_i = \langle X_i, V\rangle/\sqrt{n}$, $\tilde{\theta}_i = \langle z_i, v\rangle/\sqrt{n}$ for each $i = 1, \ldots, n$ and define $\theta = (\theta_1, \ldots, \theta_n)^T$, $\tilde{\theta} = (\tilde{\theta}_1, \ldots, \tilde{\theta}_n)^T$ and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^T$. Then

$$\|\theta\|^2 = \frac{1}{n}\sum_{i=1}^{n}\langle X_i, V\rangle^2 \le \overline{C}_X\|V\|_F^2,$$

and

$$\|\tilde{\theta}\|^2 = \frac{1}{n}\sum_{i=1}^{n}\langle z_i, v\rangle^2 \le \overline{C}_Z\|v\|^2,$$

which implies respectively that

$$\begin{aligned}\sup_{U\in\mathbf{B}(U_j,\frac{1}{n})}\left|\frac{1}{n}\sum_{i=1}^{n}\langle X_i, U-U_j\rangle\varepsilon_i\right| &= \sup_{\|V\|_F\le n^{-1}}\left|\frac{1}{n}\sum_{i=1}^{n}\langle X_i, V\rangle\varepsilon_i\right| \\ &\le \frac{1}{\sqrt{n}}\sup_{\|\theta\|\le\sqrt{\overline{C}_X}n^{-1}}|\theta^T\boldsymbol{\varepsilon}|,\end{aligned}$$

and

$$\begin{aligned}\sup_{u\in\mathbf{B}'(u_l,\frac{1}{n})}\left|\frac{1}{n}\sum_{i=1}^{n}\langle z_i, u-u_l\rangle\varepsilon_i\right| &= \sup_{\|v\|\le n^{-1}}\left|\frac{1}{n}\sum_{i=1}^{n}\langle z_i, v\rangle\varepsilon_i\right| \\ &\le \frac{1}{\sqrt{n}}\sup_{\|\theta\|\le\sqrt{\overline{C}_Z}n^{-1}}|\tilde{\theta}^T\boldsymbol{\varepsilon}|.\end{aligned}$$

Notice that

$$\sup_{\|\theta\|\le\sqrt{\overline{C}_X}n^{-1}}|\theta^T\boldsymbol{\varepsilon}| \le \sqrt{\overline{C}_X}n^{-1}\sup_{\|\theta\|\le 1}|\theta^T\boldsymbol{\varepsilon}|,$$

and

$$\sup_{\|\tilde{\theta}\|\le\sqrt{\overline{C}_Z}n^{-1}}|\tilde{\theta}^T\boldsymbol{\varepsilon}| \le \sqrt{\overline{C}_Z}n^{-1}\sup_{\|\tilde{\theta}\|\le 1}|\tilde{\theta}^T\boldsymbol{\varepsilon}|.$$

By [44, Theorem 1.19], we have

$$\mathbb{P}\Big(\sup_{\|\theta\|\le 1}|\theta^T\boldsymbol{\varepsilon}| \ge 8\sqrt{K^2+\sigma_0^2}(\sqrt{n}+\sqrt{\log m_n})\Big) \le e^{-2\log m_n},$$

and

$$\mathbb{P}\Big(\sup_{\|\tilde{\theta}\|\le 1}|\tilde{\theta}^T\boldsymbol{\varepsilon}| \ge 8\sqrt{K^2+\sigma_0^2}(\sqrt{n}+\sqrt{\log m_n'})\Big) \le e^{-2\log m_n'}.$$

Then, it follows that

$$\begin{aligned}&\mathbb{P}\Big(I_3 \ge 8\sqrt{K^2+\sigma_0^2}\big(\sqrt{n}+\sqrt{mq\log(1+2cn)}\big)\sqrt{\overline{C}_X n^{-3}}\Big) \\ &= \mathbb{P}\Big(\max_{1\le j\le m_n}\sup_{U\in\mathbf{B}(U_j,\frac{1}{n})}\left|\frac{1}{n}\sum_{i=1}^{n}\langle X_i, U-U_j\rangle\varepsilon_i\right| \\ &\qquad\qquad \ge 8\sqrt{K^2+\sigma_0^2}(\sqrt{n}+\sqrt{mq\log(1+2cn)})\sqrt{\overline{C}_X n^{-3}}\Big) \\ &\le m_n\mathbb{P}\Big(\sup_{U\in\mathbf{B}(U_j,\frac{1}{n})}\left|\frac{1}{n}\sum_{i=1}^{n}\langle X_i, U-U_j\rangle\varepsilon_i\right| \\ &\qquad\qquad \ge 8\sqrt{K^2+\sigma_0^2}(\sqrt{n}+\sqrt{mq\log(1+2cn)})\sqrt{\overline{C}_X n^{-3}}\Big) \\ &\le m_n\mathbb{P}\Big(\sqrt{\overline{C}_X}n^{-1}\frac{1}{\sqrt{n}}\sup_{\|\theta\|\le 1}|\theta^T\boldsymbol{\varepsilon}| \ge 8\sqrt{K^2+\sigma_0^2}(\sqrt{n}+\sqrt{mq\log(1+2cn)})\sqrt{\overline{C}_X n^{-3}}\Big)\end{aligned}$$

$$
\begin{aligned}
&= m_n\mathbb{P}\Big(\sup_{\|\theta\|\le 1} |\theta^T\boldsymbol{\varepsilon}| \ge 8\sqrt{K^2+\sigma_0^2}(\sqrt{n}+\sqrt{mq\log(1+2cn)})\Big)\\
&\le e^{\log m_n + 2mq\log(1+2cn)} = e^{-mq\log(1+2cn)},
\end{aligned}
$$

and

$$
\begin{aligned}
&\mathbb{P}(I_4 \ge 8\sqrt{K^2+\sigma_0^2}(\sqrt{n}+\sqrt{p\log(1+2cn)})\sqrt{\overline{C}_Z n^{-3}})\\
&= \mathbb{P}\Big(\max_{1\le l\le m_n'} \sup_{v\in\mathbf{B}'(v_l,\frac{1}{n})} \Big|\frac{1}{n}\sum_{i=1}^{n}\langle z_i, v-v_l\rangle\varepsilon_i\Big|\\
&\qquad\qquad\qquad \ge 8\sqrt{K^2+\sigma_0^2}(\sqrt{n}+\sqrt{p\log(1+2cn)})\sqrt{\overline{C}_Z n^{-3}}\Big)\\
&\le m_n'\mathbb{P}\Big(\sup_{v\in\mathbf{B}'(v_l,\frac{1}{n})} \Big|\frac{1}{n}\sum_{i=1}^{n}\langle z_i, v-v_l\rangle\varepsilon_i\Big|\\
&\qquad\qquad\qquad \ge 8\sqrt{K^2+\sigma_0^2}(\sqrt{n}+\sqrt{p\log(1+2cn)})\sqrt{\overline{C}_Z n^{-3}}\Big)\\
&\le m_n'\mathbb{P}\Big(\sqrt{\overline{C}_Z}n^{-1}\frac{1}{\sqrt{n}}\sup_{\|\tilde{\theta}\|\le 1}|\tilde{\theta}^T\boldsymbol{\varepsilon}| \ge 8\sqrt{K^2+\sigma_0^2}(\sqrt{n}+\sqrt{p\log(1+2cn)})\sqrt{\overline{C}_Z n^{-3}}\Big)\\
&\le e^{\log m_n' + 2p\log(1+2cn)} = e^{-p\log(1+2cn)}.
\end{aligned}
$$

Combining the above two inequalities with (4.7), (4.9), (4.10), (4.11), and (4.12) we have

$$
\begin{aligned}
\|\hat{B}-B^*\|_F^2 + \|\hat{\gamma}-\gamma^*\|_2^2 &\le 4\sqrt{\frac{(K^2+\sigma_0^2)c^2\overline{C}_X mq\log(1+2cn)}{n}}\\
&\quad +4\sqrt{\frac{(K^2+\sigma_0^2)c^2\overline{C}_Z p\log(1+2cn)}{n}}\\
&\quad +\frac{8\sqrt{(K^2+\sigma_0^2)\overline{C}_X}\big(\sqrt{n}+\sqrt{mq\log(1+2cn)}\big)}{n^{3/2}}\\
&\quad +\frac{8\sqrt{(K^2+\sigma_0^2)\overline{C}_Z}\big(\sqrt{n}+\sqrt{p\log(1+2cn)}\big)}{n^{3/2}}\\
&\quad +\frac{\lambda_1 r^*\|E^*\|_2 + \lambda_2|F^*|_\infty s_F^* + \lambda_3 s_\gamma^*\|\gamma^*\|_\infty}{n}\\
&\le 8c\sqrt{(K^2+\sigma_0^2)}\sqrt{\frac{(\overline{C}_X mq + \overline{C}_Z p)\log(1+2cn)}{n}}\\
&\quad +32\sqrt{(K^2+\sigma_0^2)}\frac{\sqrt{(\overline{C}_X mq + \overline{C}_Z p)\log(1+2cn)}}{n}\\
&\quad +\frac{\lambda_1 r^*\|E^*\|_2 + \lambda_2|F^*|_\infty s_F^* + \lambda_3 s_\gamma^*\|\gamma^*\|_\infty}{n}\\
&\le 8\sqrt{(K^2+\sigma_0^2)(c+4)}\sqrt{\frac{(\overline{C}_X mq + \overline{C}_Z p)\log(1+2cn)}{n}}\\
&\quad +\frac{\lambda_1 r^*\|E^*\|_2 + \lambda_2|F^*|_\infty s_F^* + \lambda_3 s_\gamma^*\|\gamma^*\|_\infty}{n}
\end{aligned}
$$

with probability at least $1 - e^{-n\sigma_0^2/K^2} - 3e^{-mq\log(1+2cn)} - 3e^{-p\log(1+2cn)}$. Then, we get the inequality (4.6). □

It is worth pointing out that the assumption (4.5) assumes that the covariates of matrix coefficients and the covariates of the vector coefficients are only weakly correlated. Similar condition is used to study the consistency of the bridge estimator for sparse regression with quadratic measurements in [22].

As mentioned before, the above theorem guarantees that the true parameters B^* and γ^* can be approximated by the estimators $\hat{B}$ and $\hat{\gamma}$ with high probability, respectively. Additionally, the estimator $\hat{B}$ may be decomposed as the sum of a low-rank matrix and a sparse matrix, and $\hat{\gamma}$ may be a sparse vector. As a direct result of Theorem 4.4, it derives the consistency of $\hat{B}$ and $\hat{\gamma}$ as follows.

Corollary 4.5. *Further, if*

$$\frac{\lambda_1 r^*\|E^*\|_2 + \lambda_2 s_F^*\|F^*\|_\infty + \lambda_3 s_\gamma^*\|\gamma^*\|_\infty}{n} \to 0 \tag{4.13}$$

and

$$\frac{(mq+p)\log n}{n} \to 0 \tag{4.14}$$

as $n \to \infty$, it follows that $\|\hat{B} - B^*\|_F + \|\hat{\gamma} - \gamma^*\|_2$ converges in probability to zero.

It should be noted that Theorem 4.3 shows that the sequence $\{(D_1^k, D_2^k, d_3^k)\}$ generated by Algorithm 1 converges to a optimal solution $(\overline{D}_1, \overline{D}_2, \overline{d}_3)$. The strong dual theorem implies that the optimal solution coincides with the original minimizer $(\hat{E}, \hat{F}, \hat{\gamma})$. Combining this with Theorem 4.4, we get the following theorem immediately.

Theorem 4.6. *Let $\{(D_1^k, D_2^k, d_3^k)\}$ be a sequence generated by Algorithm 1 and $(\overline{D}_1, \overline{D}_2, \overline{d}_3)$ be its optimal solution. It then follows that*

$$\begin{aligned}
&\|\overline{D}_1 - E^*\|_F^2 + \|\overline{D}_2 - F^*\|_F^2 + \|\overline{d}_3 - \gamma^*\|_2^2 \\
&\le 8\sqrt{(K^2+\sigma_0^2)}(c+4)\sqrt{\frac{(\overline{C}_X mq + \overline{C}_Z p)\log(1+2cn)}{n}} \\
&\quad + \frac{\lambda_1 r^*\|E^*\|_2 + \lambda_2 s_F^*\|F^*\|_\infty + \lambda_3 s_\gamma^*\|\gamma^*\|_\infty}{n}
\end{aligned}$$

with probability at least $1 - e^{-n\sigma_0^2/K^2} - 3e^{-mq\log(1+2cn)} - 3e^{-p\log(1+2cn)}$. This, together with the relations (4.13) and (4.14), *derives that*

$$\|\overline{D}_1 - E^*\|_F + \|\overline{D}_2 - F^*\|_F + \|\overline{d}_3 - \gamma^*\|_2$$

converges in probability to zero, as $n \to \infty$.

5. Numerical experiments

This section conducts extensive numerical experiments on simulated and real-world datasets to illustrate the superiority of the proposed LSDMM in problem (3.1) over LMM [2] and LSMM [36], including signal processing, image reconstruction, and video denoising. All experiments are performed using MATLAB (R2018b) on a laptop computer with an Intel Core i5 CPU, 1.6 GHz, and 8 GB of memory.

In detail, LMM is computed by the MATLAB toolbox TensorReg from [2] with parameters set as default. LSMM and LSDMM are implemented based on the ADMM framework. According to [27], the regularization parameters λ_1, λ_2, and λ_3 in (3.1) are chosen as

$$\lambda_1 = \alpha_1\|X^T y\|_\infty,\ \ \lambda_2 = \alpha_2\lambda_1,\ \ \lambda_3 = \alpha_3\|Z^T y\|_\infty,$$

where $0 < \alpha_1, \alpha_3 < 1$ and $\alpha_2 > 0$. For the sake of fairness, all tested methods are run 100 times and averaged. In addition, the same stopping criteria are applied, i.e., relative residuals 10^{-4} and a maximum number of iterations of 50000. Note that relative residuals measure the differences between the primal and dual infeasible conditions.

5.1. Signal processing

5.1.1. Simulated data

It first constructs a matrix covariate X and a vector covariate z, both of which include pseudo-random values obtained from a standard normal distribution. After that, it generates the matrix coefficient $B = E + F$, where matrix E is low-rank and F is sparse. Specially, $E = E_1 E_2^T$ with $E_1 \in \mathbb{R}^{m\times r}$, $E_2 \in \mathbb{R}^{q\times r}$, of which E_1 and E_2 satisfy standard normal distributions and r controls the rank of E. This results in a low-rank matrix E. Next, it randomly generates a sparse matrix F which obeys the standard normal distribution. Let s_1 $(0 < s_1 < 1)$ be the sparsity level, that is, the proportion of nonzero elements in the matrix F is s_1. Hence, the matrix B is a superposition of low-rank and sparse components, i.e., $E + F$. In addition, it randomly generates a sparse vector γ with sparsity level s_2, whose elements are random numbers selected from a binomial distribution with nonzero elements in γ all taken 1. Finally, the observation data y is obtained by $y = \langle X, B\rangle + \langle z, \gamma\rangle + \varepsilon$ with $\varepsilon \sim n(0, \sigma^2 I)$.

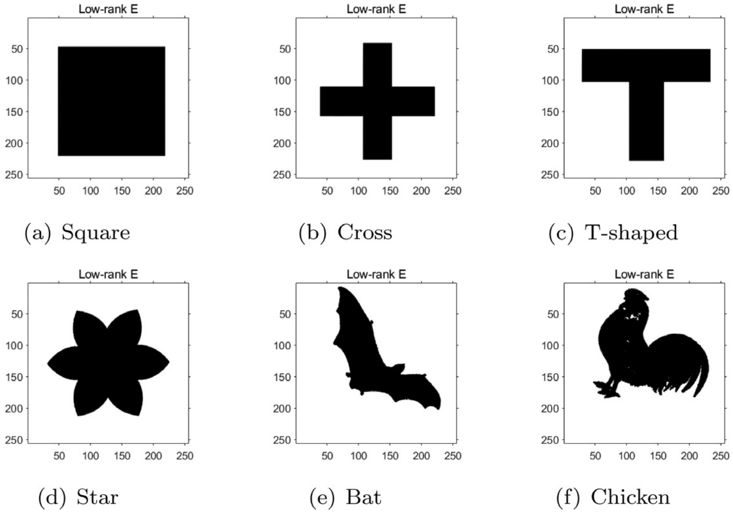
In order to measure the recovery performance of different methods, the root mean squared error (RMSE) of each estimator is compared, i.e., RMSE(B), RMSE(γ), and RMSE(y). Moreover, the best results for all cases are highlighted in bold. In numerical experiments, the sample numbers n, the dimension of vector γ, the rank of E, the sparsity level s_1, s_2, and the dimension of matrix m, q are changed as follows.

Table 1
Estimation results of LMM, LSMM, and LSDMM.

Parameter	RMSE(B)			RMSE(γ)			RMSE(y)		
	LMM	LSMM	LSDMM	LMM	LSMM	LSDMM	LMM	LSMM	LSDMM
Different n									
$n = 200$	0.0914 (0.0000)	0.0863 (0.0000)	**0.0852** **(0.0000)**	0.0461 (0.0024)	0.0064 (0.0001)	**0.0059** **0.0002**	0.0914 (0.0000)	0.0797 (0.0000)	**0.0786** **0.0001**
$n = 500$	0.0899 (0.0000)	0.0841 (0.00000)	**0.0830** **(0.0000)**	0.0056 (0.0000)	0.0045 (0.0001)	**0.0039** **(0.0001)**	0.0897 (0.0000)	0.0793 (0.0001)	**0.0782** **(0.0001)**
$n = 800$	0.0872 (0.0004)	0.0820 (0.0000)	**0.0807** **(0.0000)**	0.0035 (0.0000)	0.0031 (0.0001)	**0.0027** **(0.0001)**	0.0860 (0.0004)	0.0870 (0.0001)	**0.0808** **(0.0001)**
$n = 1200$	0.0835 (0.0000)	0.0784 (0.0000)	**0.0772** **(0.0000)**	0.0026 (0.0000)	0.0025 (0.0000)	**0.0023** **(0.0000)**	0.0831 (0.0000)	0.0787 (0.0001)	**0.0775** **(0.0001)**
$n = 1500$	0.0820 (0.0000)	0.0740 (0.0000)	**0.0731** **(0.0000)**	0.0021 (0.0000)	0.0018 (0.0001)	**0.0017** **(0.0001)**	0.0806 (0.0000)	0.0764 (0.0001)	**00752** **(0.0001)**
Different p									
$p = 100$	0.0913 (0.0000)	0.0842 (0.0000)	**0.0842** **(0.0000)**	0.0069 (0.0001)	0.0044 (0.0001)	**0.0042** **(0.0001)**	0.088 (0.0000)	0.0844 (0.0002)	**0.0843** **(0.0001)**
$p = 150$	0.0913 (0.0000)	0.0844 (0.0000)	**0.0840** **(0.0000)**	0.0083 (0.0001)	0.0048 (0.0001)	**0.0046** **(0.0001)**	0.0890 (0.0000)	0.0821 (0.0001)	**0.0800** **(0.0001)**
$p = 200$	0.0913 (0.0000)	0.0855 (0.0000)	**0.0848** **(0.0001)**	0.0101 (0.0001)	0.0053 (0.0001)	**0.0049** **(0.0002)**	0.0901 (0.0000)	0.0830 (0.0001)	**0.0821** **(0.0001)**
$p = 300$	0.0914 (0.0000)	0.0859 (0.0000)	**0.0854** **(0.0000)**	0.2387 (0.0085)	0.0065 (0.0001)	**0.0056** **(0.0002)**	0.0965 (0.0001)	0.0954 (0.0001)	**0.0942** **(0.0001)**
$p = 400$	0.0925 (0.0000)	0.0873 (0.0001)	**0.0864** **(0.0002)**	0.3584 (0.0001)	0.0074 (0.0002)	**0.0061** **(0.0001)**	0.1024 (0.0000)	0.0962 (0.0001)	**0.0947** **(0.0001)**
Different r									
$r = 5$	0.0913 (0.0000)	0.0845 (0.0000)	**0.0839** **(0.0001)**	0.0101 (0.0001)	0.0053 (0.0001)	**0.0047** **(0.0002)**	0.0901 (0.0000)	0.0830 (0.0001)	**0.0826** **(0.0002)**
$r = 10$	0.1224 (0.0000)	0.1253 (0.0000)	**0.1152** **(0.0002)**	0.0111 (0.0002)	0.0084 (0.0002)	**0.0073** **(0.0003)**	0.1143 (0.0000)	0.1238 (0.0001)	**0.1137** **(0.0001)**
$r = 15$	0.1293 (0.0000)	0.1368 (0.0000)	**0.1266** **(0.0000)**	0.0131 (0.0001)	0.0094 (0.0001)	**0.0083** **(0.0002)**	0.1264 (0.0000)	0.1337 (0.0002)	**0.1234** **(0.0001)**
$r = 20$	0.1458 (0.0000)	0.1435 (0.0001)	**0.1333** **(0.0001)**	0.0152 (0.0001)	0.0103 (0.0004)	**0.0088** **(0.0003)**	0.1482 (0.0001)	0.1491 (0.0000)	**0.1341** **(0.0001)**
$r = 25$	0.1638 (0.0000)	0.1846 (0.0000)	**0.1445** **(0.0000)**	0.0155 (0.0001)	0.0148 (0.0000)	**0.0131** **(0.0001)**	0.1664 (0.0000)	0.1772 (0.0001)	**0.1425** **(0.0001)**
Different s_1									
$s_1 = 0.05$	0.1462 (0.0000)	0.1427 (0.0000)	**0.1316** **(0.0000)**	0.0142 (0.0001)	0.0111 (0.0001)	**0.0093** **(0.0002)**	0.1478 (0.0000)	0.1401 (0.0001)	**0.1300** **(0.0000)**
$s_1 = 0.1$	0.1959 (0.0000)	0.1988 (0.0000)	**0.1889** **(0.0001)**	0.0211 (0.0001)	0.0147 (0.0001)	**0.0132** **(0.0005)**	0.2007 (0.0000)	0.1816 (0.0001)	**0.1800** **(0.0001)**
$s_1 = 0.2$	0.2633 (0.0000)	0.2637 (0.0000)	**0.2584** **(0.0000)**	0.0231 (0.0001)	0.0223 (0.0001)	**0.0208** **(0.0003)**	0.2618 (0.0000)	0.2625 (0.0001)	**0.2513** **(0.0001)**
$s_1 = 0.5$	0.3709 (0.0000)	0.3677 (0.0000)	**0.3523** **(0.0000)**	0.0337 (0.0001)	0.0318 (0.0002)	**0.0291** **(0.0003)**	0.3806 (0.0000)	0.3394 (0.0001)	**0.3209** **(0.0001)**
Different s_2									
$s_2 = 0.05$	0.0913 (0.0000)	0.0908 (0.0000)	**0.0848** **(0.0000)**	0.0082 (0.0001)	0.0089 (0.0001)	**0.0074** **(0.0001)**	0.0953 (0.0000)	0.0901 (0.0000)	**0.0801** **(0.0001)**
$s_2 = 0.1$	0.0925 (0.0000)	0.0953 (0.0000)	**0.0848** **(0.0000)**	0.0096 (0.0001)	0.0091 (0.0001)	**0.0086** **(0.0001)**	0.0953 (0.0000)	0.0955 (0.0001)	**0.0844** **(0.0001)**
$s_2 = 0.2$	0.0933 (0.0000)	0.0997 (0.0000)	**0.0856** **(0.0000)**	0.0088 (0.0001)	0.0113 (0.0002)	**0.0105** **(0.0002)**	0.0984 (0.0000)	0.0869 (0.0001)	**0.0876** **(0.0001)**
$s_2 = 0.5$	0.0951 (0.0000)	0.0864 (0.0000)	**0.0864** **(0.0000)**	0.0096 (0.0001)	0.0168 (0.0001)	**0.0157** **(0.0001)**	0.0953 (0.0000)	0.0981 (0.0001)	**0.0869** **(0.0001)**
Different m, q									
$m = 100, q = 50$	0.0913 (0.0000)	0.0855 (0.0000)	**0.0843** **(0.0000)**	0.0101 (0.0001)	0.0053 (0.0000)	**0.0047** **(0.0002)**	0.0901 (0.0000)	0.0845 (0.0002)	**0.0830** **(0.0001)**
$m = 200, q = 100$	0.0944 (0.0000)	0.0884 (0.0000)	**0.0873** **(0.0000)**	0.0091 (0.0001)	0.0064 (0.0000)	**0.0057** **(0.0002)**	0.0946 (0.0000)	0.0835 (0.0001)	**0.0829** **(0.0002)**
$m = 300, q = 200$	0.0887 (0.0000)	0.0897 (0.0000)	**0.0882** **(0.0000)**	0.0095 (0.0001)	0.0062 (0.0001)	**0.0057** **(0.0001)**	0.0885 (0.0000)	0.0893 (0.0000)	**0.0871** **(0.0000)**
$m = 500, q = 300$	0.0887 (0.0000)	0.0931 (0.0000)	**0.0919** **(0.0002)**	0.0082 (0.0001)	0.0067 (0.0000)	**0.0059** **(0.0001)**	0.0921 (0.0000)	0.0911 (0.0001)	**0.0902** **(0.0002)**
$m = 500, q = 500$	0.0914 (0.0000)	0.1005 (0.0000)	**0.0947** **(0.0000)**	0.0091 (0.0001)	0.0057 (0.0002)	**0.0048** **(0.0001)**	0.0869 (0.0000)	0.1023 (0.0000)	**0.0955** **((0.0000))**

Table 2
RMSE(y) for trip time prediction.

Rate	LMM	LSMM	LSDMM
3-fold	0.3471	0.1817	**0.1370**
5-fold	1.3733	0.1774	**0.1249**
8-fold	1.4998	0.1906	**0.1408**



(a) Square (b) Cross (c) T-shaped

(d) Star (e) Bat (f) Chicken

Fig. 1. The tested images: (a) Square, (b) Cross, (c) T-shaped, (d) Star, (e) Bat, (f) Chicken.

(a) Fix $m = 100, q = 50, p = 200, r = 5, s_1 = s_2 = 0.01$, and vary n.
(b) Fix $m = 100, q = 50, n = 300, r = 5, s_1 = s_2 = 0.01$, and vary p.
(c) Fix $m = 100, q = 50, n = 300, p = 200, s_1 = s_2 = 0.01$, and vary r.
(d) Fix $m = 100, q = 50, n = 300, p = 200, r = 5, s_2 = 0.01$, and vary s_1.
(e) Fix $m = 100, q = 50, n = 300, p = 200, r = 5, s_1 = 0.01$, and vary s_2.
(f) Fix $n = 300, p = 200, r = 5, s_1 = s_2 = 0.01$, and vary m, q.

From Table 1, it can be easily concluded that the proposed LSDMM always has the best estimation and prediction performance compared to LMM and LSMM, which convinces us to believe that LSDMM is promising for complex data analysis. The reason is that the priors, i.e., low-rank and sparse matrix decomposition, are embedded and utilized.

5.1.2. Trip time prediction

Estimating the travel time of taxis is very important for electronic dispatch systems. It collects trajectories of all 442 taxis in Porto over one year [45]. First, select a partial set with 5000 trajectories, each containing multiple features. The latitude and longitude coordinates are recorded every 15 s when the taxi is running. It involves 5 general variables, i.e., trip ID, call type, original call, taxi ID, and day type. Specifically speaking, the trip ID contains unique identifiers for 15 trips. The call type recognizes the way to request the service, which contains three types, including this trip dispatched from the center, directly requested from a taxi driver at a specific stop, and requested on a random street. The original call contains a unique identifier for each phone number requested. Taxi ID contains a unique identifier of the taxi driver who performs each trip. The day type labels the type of day the trip starts, i.e., a holiday or any other special day, the day before a holiday, etc.

In this case study, X is a matrix of size 922 × 2, z is a vector of dimension 421 with an index of 5 features, and the response y is the driving time for the taxi to complete the trip. The target is to extract features to predict the travel time of a complete taxi journey. Moreover, the data is divided into training samples and testing samples using 3-fold, 5-fold, and 8-fold cross-validation. The comparison results of RMSE(y) are presented in Table 2. It also demonstrates that the proposed LSDMM enjoys smaller RMSE(y) than those of LMM and LSMM in all three cases.

5.2. Image reconstruction

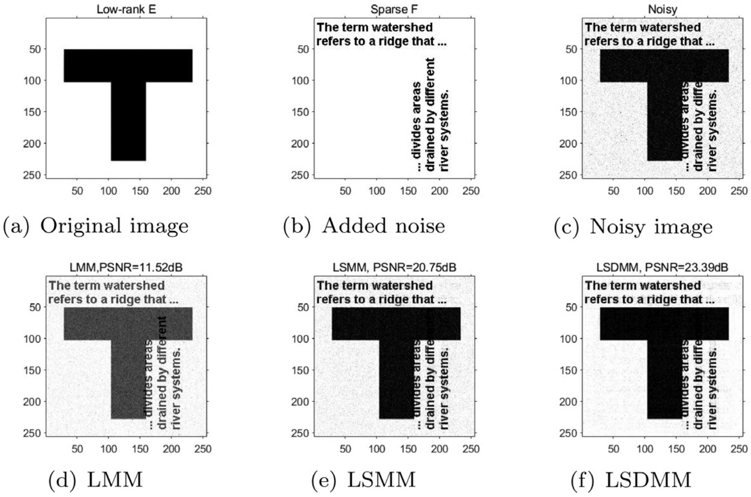
In this subsection, we test image reconstruction on different shapes. Let B be a 256 × 256 matrix with the superposition of a low-rank matrix E and a sparse matrix F, where E and F are binary. In particular, E is chosen as the images such as square, cross, T-shaped, star, bat, and chicken (see Fig. 1) and F is a picture with some lines of words, which is sparse (see

Table 3
RMSE(B) and RMSE(γ) for tested images.

Shapes	RMSE(B)			RMSE(γ)		
	LMM	LSMM	LSDMM	LMM	LSMM	LSDMM
Square	0.2655	0.0917	**0.0678**	0.2469	0.0099	**0.0086**
Cross	0.2020	0.0922	**0.0675**	0.1608	0.0112	**0.0091**
T-shape	0.2079	0.0920	**0.0678**	0.1692	0.0113	**0.0093**
Star	0.2224	0.1015	**0.0799**	0.1857	0.0125	**0.0098**
Bat	0.1924	0.1021	**0.0783**	0.1331	0.0096	**0.0046**
Chicken	0.2176	0.1031	**0.0818**	0.1723	0.0114	**0.0095**

Table 4
PSNR values for tested images.

Shapes	LMM	LSMM	LSDMM
Square	11.52	20.61	**23.42**
Cross	13.90	20.61	**23.38**
T-shape	11.52	20.75	**23.39**
Star	13.06	19.88	**21.96**
Bat	14.31	19.46	**22.12**
Chicken	13.06	19.72	**21.79**



(a) Original image (b) Added noise (c) Noisy image

(d) LMM (e) LSMM (f) LSDMM

Fig. 2. The impact of noise in reconstructed the image T-shaped: (a) Original image, (b) Added noise, (c) Noisy images, (d) Recovered by LMM, (e) Recovered by LSMM, (f) Recovered by LSDMM.

(b) in Fig. 2). Let γ be a random sparse vector of size 500. In order to make the problem easier, X and z are chosen as an identity matrix and an identity vector, respectively. Therefore, the generated noisy images satisfy $y = \langle X, B\rangle + \langle z, \gamma\rangle + \varepsilon$, where ε follows a standard normal distribution and the noise level is taken as 0.01.

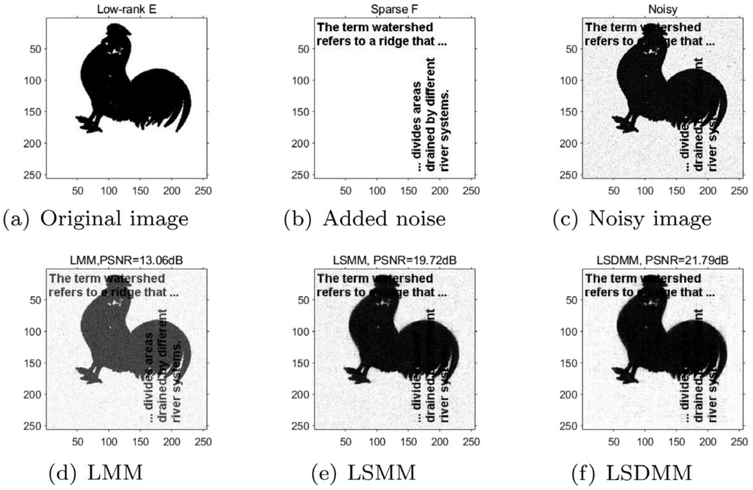
In order to measure the performance of different methods, peak-signal-to-noise-ratio (PSNR) is adopted to evaluate the quality of image reconstruction, which is defined as PSNR(B)= $20\log_{10}\frac{\sqrt{mq}}{\|B^{*}-B\|_F}$, where B^{*} is the original image and B is the corresponding reconstructed image. Naturally, the higher the PSNR value, the better the reconstruction accuracy.

Table 3 lists the comparison results of tested images with respect to RMSE(B) and RMSE(γ). Compared with LMM and LSMM, the proposed LSDMM obtains the lowest RMSE(B) and RMSE(γ) values under the same noisy conditions. Moreover, Table 4 provides the corresponding PSNR values. It can be found from Table 4, the proposed LSDMM achieves the highest PSNR values, which means that LSDMM has the best reconstruction capability.

In particular, the visual performance for images T-shaped and chicken are displayed in Fig. 2 and Fig. 3, respectively. Clearly, the reconstruction performance of the proposed LSDMM is the best, followed by LSMM and finally LMM, which is consistent with the above RMSE and PSNR values.

5.3. Video denoising

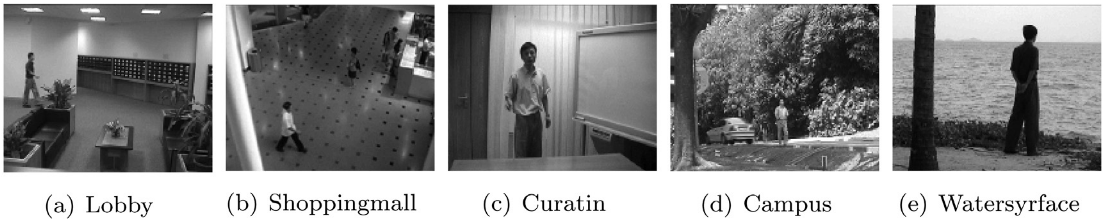
In this subsection, we conduct numerical studies on practical surveillance videos to demonstrate the superiority of the proposed LSDMM for video denoising. The CDnet dataset [46] has been considered as a difficult tracking benchmark,



(a) Original image (b) Added noise (c) Noisy image

(d) LMM (e) LSMM (f) LSDMM

Fig. 3. The impact of noise in reconstructed the image Chicken: (a) Original image, (b) Added noise, (c) Noisy image, (d) Recovered by LMM, (e) Recovered by LSMM, (f) Recovered by LSDMM.



(a) Lobby (b) Shoppingmall (c) Curatin (d) Campus (e) Watersyrface

Fig. 4. The tested videos: (a) Lobby, (b) Shoppingmall, (c) Curtain, (d) Campus, (e) Watersurface.

Table 5
RMSE(B) and RMSE(γ) for tested videos.

Videos	RMSE(B)			RMSE(γ)		
	LMM	LSMM	LSDMM	LMM	LSMM	LSDMM
Lobby	0.1607	0.159	**0.0067**	0.1666	0.0036	**0.0017**
Mall	0.1610	0.0168	**0.0088**	0.1600	0.0042	**0.0035**
Curtain	0.1546	0.0151	**0.0085**	0.1536	0.0041	**0.0029**
Campus	0.1660	0.0196	**0.0128**	0.1635	0.0042	**0.0031**
Watersurface	0.1678	0.0161	**0.0111**	0.1658	0.0042	**0.0026**

which consists of 31 surveillance videos and each has 8000 frames. To validate the denoising performance under different scenarios, two categories of five videos are chosen. The first category is with static background, i.e., Lobby, Shoppingmall; see (a)–(b) in Fig. 4. The second category is with dynamic background, i.e., Curtain, Campus, and Watersurface with the moving of curtain, leaves, and water surface, respectively; see (c)–(e) in Fig. 4.

Let X and z be an identity matrix and an identity vector, respectively, B be the above videos with 40 frames, ε follow a standard normal distribution with noise level 0.01, and γ is a random sparse vector with size 500. Meanwhile, PSNR is employed to measure the quality of video denoising.

Table 5 provides the comparison results of tested videos in terms of RMSE(B) and RMSE(γ). Compared with LMM and LSMM, the proposed LSDMM has the lowest RMSE(B) and RMSE(γ) values. Moreover, Fig. 5 intercepts a frame of videos for display. From the first row to the last row are tested, noise added, the denoised image by LMM, the denoised image by LSMM, and the denoised image by LSDMM, respectively. It is not difficult to conclude that the proposed LSDMM obtains the higher values of PSNR, that is to say, LSDMM derives the best video denoising performance, which agrees with the above RMSE results in Table 5.

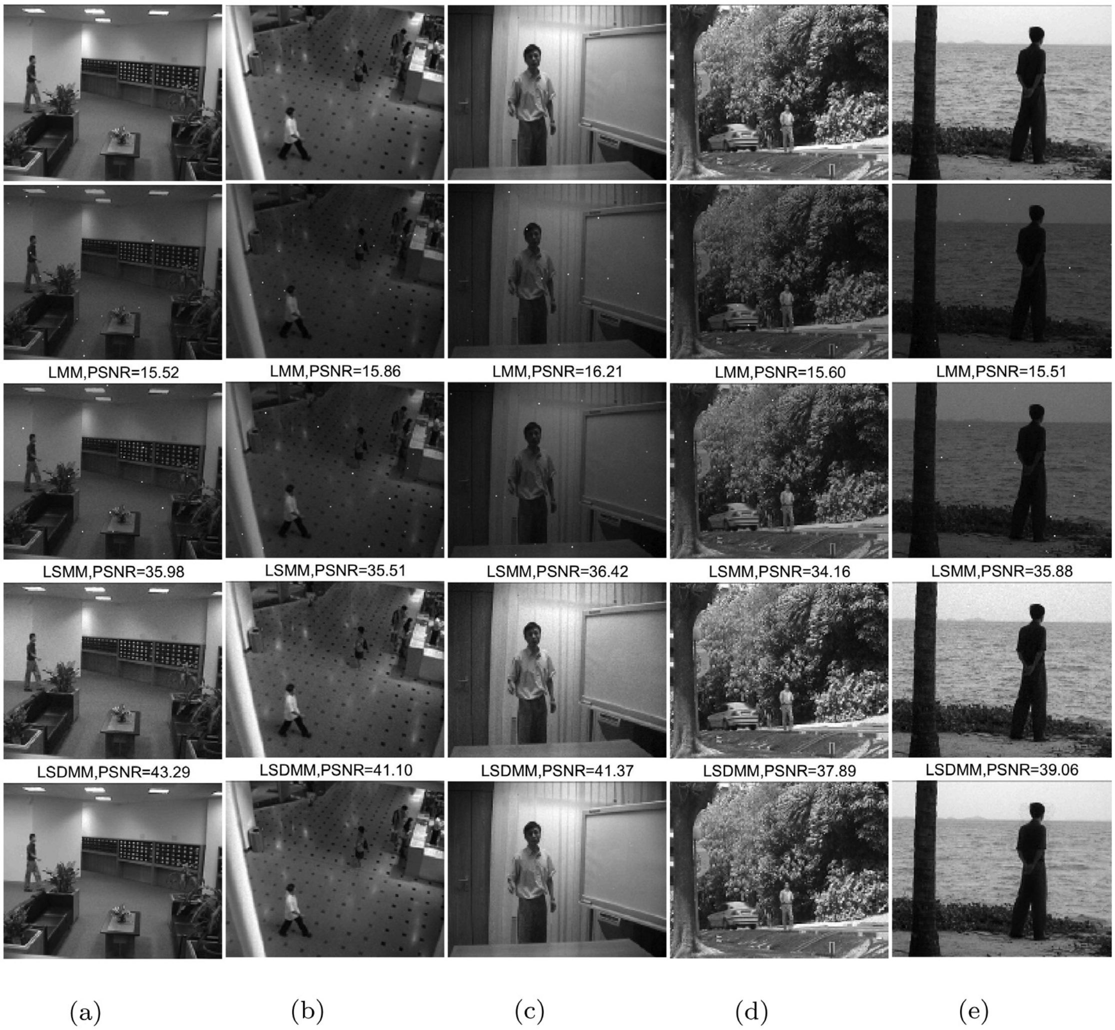


Fig. 5. The impact of denoising performance: (a) Lobby, (b) Shopping-mall, (c) Curtain, (d) Campus, (e) Watersurface.

Because of the inherent characteristics that the matrix B in the model (3.1) is a superposition of the low-rank matrix E and the sparse matrix F, the proposed LSDMM can not only be used for video denoising as described previously, but also can extract the dynamic foreground of the aforementioned surveillance videos. At the end of this section, we would like to display the extraction performance of the proposed LSDMM. In Fig. 6, the first row is the original video frames, the second row is the corresponding background part, and the third row is the corresponding foreground part. The results suggest that the proposed method is also promising for foreground/background extraction.

6. Conclusion

Matrix minimization (MM) has become an important tool for analyzing large-scale measurement data. In this paper, we have proposed a new MM approach by embedding low-rank and sparse prior structures. To solve the proposed model, an efficient optimization algorithm based on the semi-proximal ADMM has been developed from the perspective of duality. Furthermore, the convergence analysis and non-asymptotic statistical property have been discussed in detail. The performance of the proposed method has been validated on signal processing, image reconstruction, and video denoising.

Although the proposed method has achieved satisfactory performance, several interesting and important directions are worth exploring. First, the proposed method can be extended to the tensor case and applied to face recognition and magnetic resonance imaging (MRI). Secondly, it is possible to construct deep neural networks based on a general MM framework to preserve global and local prior information. Finally, how to practically and effectively choose tuning parameters is an open question, which deserves further investigation.

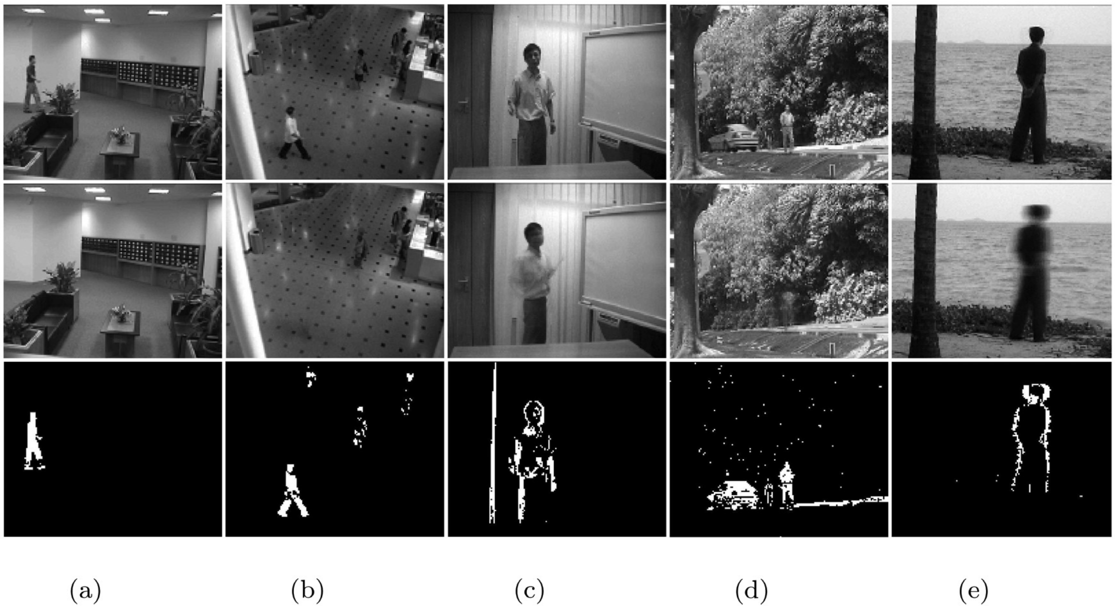


Fig. 6. The impact of extraction performance by LSDMM: (a) Lobby, (b) Shopping-mall, (c) Curtain, (d) Campus, (e) Watersurface.

Abbreviations

ADMM	:	Alternating direction method of multipliers
EEG	:	Electroencephalography
LMM	:	Low-rank
LSDMM	:	Low-rank and sparse decomposition regularized
LSMM	:	Low-rank and sparse
MRI	:	Magnetic resonance imaging
MM	:	Matrix minimization
PSNR	:	Peak-signal-to-noise-ratio
RMSE	:	Root mean square error
sPADMM	:	Semi-proximal ADMM
SVD	:	Singular value decomposition

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank the principal editor and the anonymous referee for their comments and suggestions, which have greatly improved the paper.

References

[1] M. Jordan, T. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.
[2] H. Zhou, L. Li, Regularized matrix regression, J. R. Stat. Soc. Ser. B Stat. Methodol. 76 (2) (2014) 463–483.
[3] J. Xu, E. Chi, K. Lange, Generalized linear model regression under distance-to-set penalties, Adv. Neural Inf. Process. Syst. 30 (2017).
[4] L. Li, X. Zhang, Parsimonious tensor response regression, J. Amer. Statist. Assoc. 112 (519) (2017) 1131–1146.
[5] Q. Zheng, F. Zhu, J. Qin, B. Chen, P. Heng, Sparse support matrix machine, Pattern Recognit. 76 (2018) 715–726.
[6] H. Zhao, Q. Zheng, K. Ma, H. Li, Y. Zheng, Deep representation-based domain adaptation for nonstationary EEG classification, IEEE Trans. Neural Netw. Learn. Syst. 32 (2) (2021) 535–545.
[7] L. Wang, Z. Zhang, D. Dunson, Symmetric bilinear regression for signal subgraph estimation, IEEE Trans. Signal Process. 67 (7) (2019) 1929–1940.
[8] Q. Zheng, Y. Wang, P. Heng, Multitask feature learning meets robust tensor decomposition for EEG classification, IEEE Trans. Cybern. 51 (4) (2021) 2242–2252.
[9] R. Caruana, Multitask learning, Mach. Learn. 28 (1) (1997) 41–75.

[10] A. Rohde, A. Tsybakov, Estimation of high-dimensional low-rank matrices, Ann. Statist. 39 (2) (2011) 887–930.

[11] J. Wang, G. Xu, C. Li, Z. Wang, F. Yan, Surface defects detection using non-convex total variation regularized RPCA with kernelization, IEEE Trans. Instrum. Meas. 70 (2021) 1–13.

[12] R. Otazo, E. Candes, D.K. Sodickson, Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components, Magn. Reson. Med. 73 (3) (2015) 1125–1136.

[13] T. Bouwmans, A. Sobral, S. Javed, S. Jung, E. Zahzah, Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset, Comput. Sci. Rev. 23 (2017) 1–71.

[14] B. Yousefi, C. Castanedo, X. Maldague, Measuring heterogeneous thermal patterns in infrared-based diagnostic systems using sparse low-rank matrix approximation: Comparative study, IEEE Trans. Instrum. Meas. 70 (2021) 1–9.

[15] X. Xiu, Y. Yang, L. Kong, W. Liu, Laplacian regularized robust principal component analysis for process monitoring, J. Process Control 92 (2020) 212–219.

[16] W. Yu, C. Zhao, Low-rank characteristic and temporal correlation analytics for incipient industrial fault detection with missing data, IEEE Trans. Ind. Inform. 17 (9) (2021) 6337–6346.

[17] Y. Fu, C. Luo, Z. Bi, Low-rank joint embedding and its application for robust process monitoring, IEEE Trans. Instrum. Meas. 70 (2021) 1–13.

[18] V. Chandrasekaran, S. Sanghavi, P. Parrilo, A. Willsky, Rank-sparsity incoherence for matrix decomposition, SIAM J. Optim. 21 (2) (2011) 572–596.

[19] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM 58 (3) (2011) 1–37.

[20] M. Tao, X. Yuan, Recovering low-rank and sparse components of matrices from incomplete and noisy observations, SIAM J. Optim. 21 (1) (2011) 57–81.

[21] P. Bühlmann, S. Van De Geer, Statistics for High-Dimensional Data: Methods, Theory and Applications, Springer Science and Business Media, 2011.

[22] J. Fan, L. Kong, L. Wang, N. Xiu, Variable selection in sparse regression with quadratic measurements, Statist. Sinica 28 (3) (2018) 1157–1178.

[23] D. Kong, B. An, J. Zhang, H. Zhu, L2rm: Low-rank linear regression models for high-dimensional matrix responses, J. Amer. Statist. Assoc. (2019).

[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2011) 1–122.

[25] C. Chen, B. He, Y. Ye, X. Yuan, The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent, Math. Program. 155 (1) (2016) 57–79.

[26] L. Yang, T. Pong, X. Chen, Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction, SIAM J. Imaging Sci. 10 (1) (2017) 74–110.

[27] X. Li, D. Sun, K. Toh, A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems, SIAM J. Optim. 28 (1) (2018) 433–458.

[28] D. Han, A survey on some recent developments of alternating direction method of multipliers, J. Oper. Res. Soc. China (2022) 1–52.

[29] N. Vaswani, T. Bouwmans, S. Javed, P. Narayanamurthy, Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery, IEEE Signal Process. Mag. 35 (4) (2018) 32–55.

[30] Z. Hu, F. Nie, R. Wang, X. Li, Low rank regularization: A review, Neural Netw. 136 (2021) 218–232.

[31] Y. Tian, Y. Zhang, A comprehensive survey on regularization strategies in machine learning, Inf. Fusion. 80 (2022) 146–166.

[32] R.T. Rockafellar, R. Wets, Variational Analysis, Springer, New York, 1998.

[33] N. Parikh, S. Boyd, Proximal algorithms, Found. Trends Optim. 1 (3) (2014) 127–239.

[34] D. Donoho, De-noising by soft-thresholding, IEEE Trans. Inform. Theory 41 (3) (1995) 613–627.

[35] R. Tibshirani, Regression shrinkage and selection via the Lasso, J. R. Stat. Soc. Ser. B Stat. Methodol. 58 (1) (1996) 267–288.

[36] P. Shang, L. Kong, Regularization parameter selection for the low rank matrix recovery, J. Optim. Theory Appl. 189 (3) (2021) 772–792.

[37] J.-F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. 20 (4) (2010) 1956–1982.

[38] T.-H. Oh, Y. Matsushita, Y.-W. Tai, I.S. Kweon, Fast randomized singular value thresholding for low-rank optimization, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2) (2017) 376–391.

[39] S. Ma, D. Goldfarb, L. Chen, Fixed point and bregman iterative methods for matrix rank minimization, Math. Program. 128 (1) (2011) 321–353.

[40] W. Hu, Y. Lu, J. Ren, A fixed-point proximity algorithm for recovering low-rank components from incomplete observation data with application to motion capture data refinement, J. Comput. Appl. Math. 410 (2022) 114224.

[41] R.T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, 1970.

[42] M. Fazel, T. Pong, D. Sun, P. Tseng, Hankel matrix rank minimization with applications to system identification and realization, SIAM J. Matrix Anal. Appl. 34 (3) (2013) 946–977.

[43] B. Recht, M. Fazel, P. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, SIAM Rev. 52 (3) (2010) 471–501.

[44] P. Rigollet, J.-C. Hütter, High dimensional statistics, in: Lect. Notes Course 18S997, vol. 813, (814) 2015, p. 46.

[45] C. Chen, S. Jiao, S. Zhang, W. Liu, L. Feng, Y. Wang, TripImputor: Real-time imputing taxi trip purpose leveraging multi-sourced Urban data, IEEE Trans. Intell. Transp. Syst. 19 (10) (2018) 3292–3304.

[46] Y. Wang, P. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, P. Ishwar, CDnet 2014: An expanded change detection benchmark dataset, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit, 2014, pp. 387–394.