Signal Processing 204 (2023) 108816

Contents lists available at ScienceDirect

Signal Processing

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

Robust PCA via non-convex half-quadratic regularization

Zhi-Yong Wang, Xiao Peng Li*, Hing Cheung So[1], Zhaofeng Liu

Department of Electrical Engineering, City University of Hong Kong, Hong Kong, China

ARTICLE INFO

Article history:
Received 28 June 2022
Revised 14 October 2022
Accepted 16 October 2022
Available online 21 October 2022

Keywords:
Robust PCA
Matrix factorization
Low-rank
Sparse
Non-convex regularization
Proximal block coordinate descent

ABSTRACT

In this paper, we propose a new non-convex regularization term named half-quadratic function to achieve robustness and sparseness for robust principal component analysis, and derive its proximity operator, indicating that the resultant optimization problem can be solved in computationally attractive manner. In addition, the low-rank matrix component is expressed as the factorization form and proximal block coordinate descent is leveraged to seek its solution, whose convergence is rigorously analyzed. We prove that any limit point of the iterations is a critical point of the objective function. Furthermore, the parameter that controls the robustness and sparseness in our algorithm, is automatically adjusted according to the statistical residual error. Experimental results based on synthetic and real-world data demonstrate that the devised algorithm can effectively extract the low-rank and sparse components. MATLAB code is available at https://github.com/bestzywang.

1. Introduction

Principal component analysis (PCA) [1], as a workhorse intended for dimensionality reduction, aims to find the optimal low-dimensional linear subspace of high-dimensional data in the Euclidean space. It has a variety of applications such as computer vision [2], bioinformatics [3,4], as well as signal and image processing [5], since most information/energy of many real-world data lies in a low-dimensional subspace. However, its performance will be degraded in the presence of gross errors, and outliers are ubiquitous in many situations due to sudden intense interference in the transmission, sensor failure and calibration error [6,7].

To tackle this problem, robust PCA (RPCA) [8–10] has been proposed to decompose the observed matrix data contaminated by outliers into a sum of low-rank and sparse matrices. Intuitively, the RPCA can be formulated as rank and ℓ_0-norm minimization problem, but it is NP-hard as the rank function is discrete. In [9,10], the authors replace the rank function and ℓ_0-norm with the nuclear norm and ℓ_1-norm, respectively, to obtain a convex optimization problem since many ready-made convex optimization methods such as the inexact augmented Lagrange multiplier (IALM) [11] and alternating direction method (ALM) [12], can be directly applied. Although one can acquire low-rank and sparse matrices exactly under mild conditions via this convex model, it does not take the dense noise with small magnitudes into account. This point is difficult to guarantee because the data in practical scenarios may also be corrupted by Gaussian noise or other noise that affects every matrix entries. Hence, Zhou et al. [13] decompose the data matrix into a sum of low-rank, sparse and dense noise matrices, leading to accurate recovery in the presence of point-wise noise. In addition, efficient algorithms, including alternating splitting augmented Lagrangian method (ASALM) [15], and partially smooth proximal gradient (PSPG) [16], are developed. Recently, Gu et al. [14] replace the nuclear norm with weighted nuclear norm, which adaptively assigns a distinct weight for each singular value, to attain low-rank matrix recovery, and experiments have verified its effectiveness. However, the bottleneck of the above approaches is that they require a full singular value decomposition (SVD) calculation at each iteration, and thus their computational complexity significantly increases with the data dimensions, implying that large matrices cannot be tackled efficiently.

As a remedy, RPCA methods based on factorization, which decompose the low-rank matrix as the product of two rank-r matrices, say, $\boldsymbol{U}$ and $\boldsymbol{V}$, including go decomposition (GoDec) [17], low-rank matrix fitting (LMaFit) [18] and greedy bilateral smoothing (GreBsmo) [19], have been suggested. Besides, Bayesian schemes such as Bayesian RPCA (BRPCA) [20], variational Bayesian RPCA (VBRPCA) [21] and sparse Bayesian learning for RPCA (SBLRPCA) [22], are proposed. However, they leverage the ℓ_1-norm as regularization penalty to achieve sparseness, and it has been pointed out that the ℓ_1-norm over-penalizes the data with large

* Corresponding author.
E-mail addresses: z.y.wang@my.cityu.edu.hk (Z.-Y. Wang), x.p.li@my.cityu.edu.hk (X.P. Li), hcso@ee.cityu.edu.hk (H.C. So), zhaofeliu3-c@my.cityu.edu.hk (Z. Liu).
[1] EURASIP Member.

https://doi.org/10.1016/j.sigpro.2022.108816
0165-1684/

magnitudes, resulting in a biased solution [23]. In contrast, non-convex regularization penalties, such as smoothly clipped absolute deviation (SCAD) [23], ℓ_p-norm ($0 < p < 1$) and Laplace function [24], can ameliorate this problem. This results in many algorithms, including robust recovery of corrupted low-rank matrix by implicit regularizer (IR) [25], robust low-rank matrix decomposition based on maximum correntropy (GoDec+) [26], non-convex ℓ_p-norm based robust PCA (LPRPCA) [27] and non-convex regularized robust PCA (NCRPCA) [28]. However, the SCAD requires two user-defined parameters, while the ℓ_p-norm (except for $p = \frac{1}{2}$ and $p = \frac{2}{3}$) and Laplace function have no closed-solutions for their corresponding proximity operators.

In this paper, we propose a non-convex regularization penalty term called half-quadratic function (HQF) to achieve robustness and develop an efficient algorithm, called robust PCA via HQF regularization (RPCA-HQF). Although the regularization term is non-convex, we obtain its proximity operator, which facilitates solving the resultant optimization problem in a computationally efficient manner. Besides, the parameter that controls robustness is self-adaptive in accordance to current residual error, and experiments demonstrate that our method exhibits better restoration results and is more robust than the competing approaches.

Compared with non-convex regularization [25,26], which only obtains the low-rank components, our method can seek both the low-rank and sparse matrices, and does not require other tunable parameters except for the termination conditions. Accordingly, our main contributions are highlighted as follows:

1. A new non-convex regularization is utilized to attain robustness and sparseness, and its proximity operator is derived, which makes the corresponding optimization problem solvable in an efficient manner.
2. We theoretically analyze that any accumulation point of $\{\boldsymbol{U}^k, \boldsymbol{V}^k\}$ generated by the proposed algorithm is a critical point.
3. There are no tunable parameters other than the termination conditions in the proposed algorithm and experiments demonstrate that our method exhibits better restoration results, compared with the competing approaches.

The remainder of this paper is organized as follows. In Section 2, we introduce notations and related works about RPCA. The RPCA-HQF algorithm is developed in Section 3. Then experimental results based on synthetic data, real-world videos and face images are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. Preliminaries

In this section, notations are provided and related works are reviewed.

2.1. Notations

Throughout this paper, the (i, j) entry of a matrix $\boldsymbol{A}$ is represented by A_{ij} and $\boldsymbol{0}$ means a matrix with all entries being zeros. In addition, the Frobenius norm of $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is denoted by $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2}$, and unless stated otherwise, the matrix norm refers to the Frobenius norm. Moreover, $\|\boldsymbol{S}\|_0$ signifies the number of non-zero entries of $\boldsymbol{S}$. Finally, $(\cdot)^T$ and $|\cdot|$ are the transpose and absolute operators, respectively.

2.2. Related works

Mathematically, RPCA via decomposition into low-rank and sparse matrices can be directly formulated as [8–10]:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \ \mathrm{rank}(\boldsymbol{L}) + \lambda\|\boldsymbol{S}\|_0, \ \text{s.t. } \boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S} \tag{1}$$

where $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ is the observed low-rank matrix corrupted by outliers, $\boldsymbol{L} \in \mathbb{R}^{m\times n}$ is the low-rank matrix, $\boldsymbol{S} \in \mathbb{R}^{m\times n}$ is the sparse matrix, and $\|\boldsymbol{S}\|_0$ denotes the cardinality of $\boldsymbol{S}$ (for cardinality, please refer to [29,30]). However, (1) is an intractable problem as the rank function is discrete. To make it computationally feasible, the nuclear norm and ℓ_1-norm are employed to replace the rank function and ℓ_0-norm, respectively, resulting in the convex relaxation of (1) [9,10]:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \ \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1, \ \text{s.t. } \boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S} \tag{2}$$

where $\|\boldsymbol{L}\|_*$ is the nuclear norm, which is the sum of singular values of $\boldsymbol{L}$. Although (2) can recover $\boldsymbol{L}$ and $\boldsymbol{S}$ exactly with high probability under mild conditions, it requires that the low-rank and sparse components are strictly low-rank and exactly sparse, respectively. Nevertheless, many real-world data are approximately low-rank and may be contaminated by Gaussian noise as well as outliers, leading to the relaxed model of (2), known as stable principal component pursuit (SPCP) [13]:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \ \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1, \ \text{s.t. } \|\boldsymbol{X} - \boldsymbol{L} - \boldsymbol{S}\|_F \le \delta \tag{3}$$

where $\delta > 0$ is a constant related to the dense noise matrix. Besides, (3) needs to perform SVD at each iteration, and to avoid this decomposition, RPCA based on factorization is suggested [19], corresponding to:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{S}} \ \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V} - \boldsymbol{S}\|_F^2 + \lambda\|\boldsymbol{S}\|_1 \tag{4}$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r\times n}$. Since the ℓ_1-norm regularization brings about the bias problem, non-convex regularization is introduced [26], resulting in:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{S}} \ \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V} - \boldsymbol{S}\|_F^2 + \lambda\|\boldsymbol{S}\|_\varphi \tag{5}$$

where $\|\boldsymbol{S}\|_\varphi = \sum_{i=1}^m \sum_{j=1}^n \varphi(\boldsymbol{S}_{ij})$, and $\varphi(\cdot)$ is an implicit non-convex function.

3. Proposed algorithm

First, we put forth a new non-convex function, whose expression is:

$$g_e(t) = \begin{cases} -\dfrac{(|t| - e)^2}{2} + \dfrac{e^2}{2}, & |t| < e \\ \dfrac{e^2}{2}, & |t| \ge e \end{cases} \tag{6}$$

Since when $0 \le t \le e$, $g_e(t)$ is quadratic and when $t \ge e$, $g_e(t)$ is a constant, we name $g_e(t)$ as HQF.

Motivated by (4) and (5), our problem formulation is:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{S}} \ \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V} - \boldsymbol{S}\|_F^2 + \|\boldsymbol{S}\|_{g_e} \tag{7}$$

where $\|\boldsymbol{S}\|_{g_e}$ is separable, namely, $\|\boldsymbol{S}\|_{g_e} = \sum_{i=1}^m \sum_{j=1}^n g_e(\boldsymbol{S}_{ij})$. Different from (4) that employs λ to control sparseness and attain robustness, (7) adopts e to achieve sparseness and robustness, and we will design a strategy to choose the value of e. We first show how to solve $\boldsymbol{S}$. Given $\boldsymbol{U}^k$ and $\boldsymbol{V}^k$, (7) is equal to:

$$\boldsymbol{S}^{k+1} = \arg\min_{\boldsymbol{S}} \ \frac{1}{2}\left\|\boldsymbol{R}^k - \boldsymbol{S}\right\|_F^2 + \|\boldsymbol{S}\|_{g_e} \tag{8}$$

where $\boldsymbol{R}^k = \boldsymbol{X} - \boldsymbol{U}^k\boldsymbol{V}^k$. We define a proximity operator prior to solving (8), defined as:

$$P(r) := \arg\min_t \ \left\{\frac{1}{2}(r - t)^2 + g_e(t)\right\} \tag{9}$$

For the HQF, its proximity operator is derived in the following theorem.

Theorem 1. *For a constant $r \in \mathbb{R}$, the proximity operator of HQF is:*

$$P(r) = \begin{cases} 0, & |r| < e \\ r, & |r| \geq e \end{cases} \tag{10}$$

Proof: Substituting (6) into (9), we have:

$$\begin{aligned} P(r) :&= \arg\min_t \frac{(r-t)^2}{2} + g_e(t) \\ &= \begin{cases} \arg\min_t \frac{2(e-r)t + r^2}{2}, & 0 \leq t < e \\ \arg\min_t \frac{-2(e+r)t + r^2}{2}, & -e < t < 0 \\ \arg\min_t \frac{(r-t)^2}{2} + \frac{e^2}{2}, & |t| \geq e \end{cases} \\ &= \begin{cases} 0, & |r| < e \\ r, & |r| \geq e \end{cases} \end{aligned}$$

The proof is complete. ■

Therefore, the solution to (8) is:

$$\boldsymbol{S}^{k+1} = P(\boldsymbol{R}^{k+1}) \tag{11}$$

It is worth mentioning that e controls the robustness of (7) because any entry $\boldsymbol{R}_{ij} \geq e$ will be regarded as an outlier. In fact, the choice of e depends on the inner noise level. Since the conventional standard deviation is no longer a reliable spread measure in the presence of outliers, the robust normalized median absolute deviation (MADN) [35] is exploited, that is,

$$\sigma^k = 1.4815 \times \mathrm{Med}\left(\left|\mathrm{vec}\left(\boldsymbol{R}^k\right) - \mathrm{Med}\left(\mathrm{vec}\left(\boldsymbol{R}^k\right)\right)\right|\right) \tag{12}$$

with $\mathrm{Med}(\cdot)$ being the sample median operator. Thus, we have:

$$e^k = \min\left\{\zeta\sigma^k, e^{k-1}\right\} \tag{13}$$

where $\zeta > 0$ is a constant, and we set $\zeta = 3$ according to rule of thumb [36].

Next, given $\boldsymbol{S}^{k+1}$, the proximal block coordinate descent (BCD) [31,32] is leveraged to find the solutions to $\boldsymbol{U}$ and $\boldsymbol{V}$ via alternately updating $\boldsymbol{U}$ and $\boldsymbol{V}$ as:

$$\boldsymbol{U}^{k+1} = \arg\min_{\boldsymbol{U}} \ \tfrac{1}{2}\left\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^k - \boldsymbol{S}^{k+1}\right\|_F^2 + \tfrac{\lambda}{2}\left\|\boldsymbol{U} - \boldsymbol{U}^k\right\|_F^2 \tag{14}$$

$$\boldsymbol{V}^{k+1} = \arg\min_{\boldsymbol{V}} \ \tfrac{1}{2}\left\|\boldsymbol{X} - \boldsymbol{U}^{k+1}\boldsymbol{V} - \boldsymbol{S}^{k+1}\right\|_F^2 + \tfrac{\lambda}{2}\left\|\boldsymbol{V} - \boldsymbol{V}^k\right\|_F^2 \tag{15}$$

where $\lambda > 0$ is the proximal parameter [28].

Given $\boldsymbol{X}$, $\boldsymbol{V}^k$ and $\boldsymbol{S}^{k+1}$, (14) is convex, whose closed-form solution is:

$$\boldsymbol{U}^{k+1} = \left(\boldsymbol{D}^{k+1}\left(\boldsymbol{V}^k\right)^T - \lambda\boldsymbol{U}^k\right)\left(\boldsymbol{V}^k\left(\boldsymbol{V}^k\right)^T - \lambda\boldsymbol{I}\right)^{-1} \tag{16}$$

where $\boldsymbol{D}^{k+1} = \boldsymbol{X} - \boldsymbol{S}^{k+1}$ and its computational complexity is $\mathcal{O}(mnr^2)$.

Similarly, given $\boldsymbol{X}$, $\boldsymbol{U}^{k+1}$ and $\boldsymbol{S}^{k+1}$, the optimal solution to (15) is:

$$\boldsymbol{V}^{k+1} = \left(\left(\boldsymbol{U}^{k+1}\right)^T\boldsymbol{U}^{k+1} - \lambda\boldsymbol{I}\right)^{-1}\left(\left(\boldsymbol{U}^{k+1}\right)^T\boldsymbol{D}^{k+1} - \lambda\boldsymbol{V}^k\right) \tag{17}$$

whose computational complexity is also $\mathcal{O}(mnr^2)$.

Besides, since (7) is non-convex, PowerFactorization [33] is adopted to initialize $\boldsymbol{U}$ and $\boldsymbol{V}$ in (7), that is:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \ \tfrac{1}{2}\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}\|_F^2 \tag{18}$$

which is a special case of (7) when $\boldsymbol{S} = \boldsymbol{0}$, and its solutions are:

$$\boldsymbol{U}^{n+1} = \boldsymbol{X}\left((\boldsymbol{V}^n)^\dagger\right)^T \tag{19}$$

$$\boldsymbol{V}^{n+1} = \boldsymbol{X}\left(\left(\boldsymbol{U}^{n+1}\right)^\dagger\right)^T \tag{20}$$

Algorithm 1 RPCA-HQF.

Input: $\boldsymbol{X}$, I_m, I_n, e^0, ζ and η
Initialize: Randomize $\boldsymbol{V}^0 \in \mathbb{R}^{r\times n}$.
for $n = 1, 2, \cdots, I_n$ **do**
  Find $\boldsymbol{U}^n$ according to (19) // Fix $\boldsymbol{V}^{n-1}$, optimize $\boldsymbol{U}$
  Find $\boldsymbol{V}^n$ according to (20) // Fix $\boldsymbol{U}^n$, optimize $\boldsymbol{V}$
end for
Set $\boldsymbol{U}^0 = \boldsymbol{U}^n$ and $\boldsymbol{V}^0 = \boldsymbol{V}^n$
for $k = 1, 2, \cdots, I_m$ **do**
  Calculate e^k according to (13) // Fix $\boldsymbol{U}^{k-1}$ and $\boldsymbol{V}^{k-1}$, update e^k
  Find $\boldsymbol{S}^k$ according to (11) // Fix $\boldsymbol{U}^{k-1}$, $\boldsymbol{V}^{k-1}$ and e^k, optimize $\boldsymbol{S}$
  Find $\boldsymbol{U}^k$ according to (16) // Fix $\boldsymbol{V}^{k-1}$ and $\boldsymbol{S}^k$, optimize $\boldsymbol{U}$
  Find $\boldsymbol{V}^k$ according to (17) // Fix $\boldsymbol{U}^k$ and $\boldsymbol{S}^k$, optimize $\boldsymbol{V}$
  Stop, if a termination condition is satisfied.
end for
Output: $\boldsymbol{L} = \boldsymbol{U}^k\boldsymbol{V}^k$ and $\boldsymbol{S}^k$.

The detailed optimization procedure is summarized in Algorithm 1. Since (19) and (20) are used to provide initialization for (7) and alternating minimization has a fast convergence rate [34], we set $I_n = 3$ in this study. Besides, Algorithm 1 will be terminated when $\eta = \left(\|\boldsymbol{R}_k\|_F - \|\boldsymbol{R}_{k+1}\|_F\right)/\sqrt{m\times n}$ is less than a preset threshold value η and/or the iteration number reaches the maximum allowable number of outer iterations I_m. Moreover, we define $\mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{S},e) = \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V} - \boldsymbol{S}\|_F^2 + \|\boldsymbol{S}\|_{g_e}$, and the theoretical analysis of Algorithm 1 is provided in Theorem 2. We first provide the definition of a critical point.

Definition 1. Given a smooth function $f(x)$, x^* is a critical point of $f(x)$ if $0 = \partial f(x^*)$ [37,38].

Theorem 2. *The loss function $\mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{S},e)$ is non-increasing and lower bounded by 0, thus the sequence $\mathcal{L}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k,e^k)$ generated by Algorithm 1 is convergent. In addition, any limit point of $\{\boldsymbol{U}^k, \boldsymbol{V}^k\}$ is a critical point of problem (7).*

Proof: First, we prove that when updating $e^k \to e^{k+1}$ via (13), $\mathcal{L}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k,e^{k+1}) \leq \mathcal{L}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k,e^k)$. $\|\boldsymbol{S}\|_{g_e}$ is the only component related to e in $\mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{S},e)$ and $\|\boldsymbol{S}\|_{g_e} = \sum_{i,j} g_e(S_{i,j})$. By taking the partial derivative of $g_e(\boldsymbol{S}_{i,j})$ with respect to (w.r.t.) e, we have $g_e(\boldsymbol{S}_{i,j}) > 0$, because

$$\frac{\partial g_e(S_{i,j})}{\partial e} = \begin{cases} |S_{i,j}|, & |S_{i,j}| < e \\ e, & |S_{i,j}| \geq e \end{cases} \tag{21}$$

and it is easy to obtain:

$$\frac{\partial \mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{S},e)}{\partial e} = \frac{\partial \sum_{i,j} g_e(\boldsymbol{S}_{i,j})}{\partial e} = \sum_{i,j}\frac{\partial g_e(\boldsymbol{S}_{i,j})}{\partial e} > 0 \tag{22}$$

which means that $\mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{S},e)$ is monotonically increasing w.r.t. e. Since the updated rule in (13) is non-increasing, we attain

$$\mathcal{L}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k,e^{k+1}) \leq \mathcal{L}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k,e^k) \tag{23}$$

Second, when updating $\boldsymbol{S}^k$, we have

$$\mathcal{L}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^{k+1},e^{k+1}) \leq \mathcal{L}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k,e^{k+1}) \tag{24}$$

which is due to the proximity operator (10).

In addition, since $\boldsymbol{U}^{k+1}$ is the optimal solution to (14), we have:

$$\frac{1}{2}\left\|\boldsymbol{X} - \boldsymbol{U}^{k+1}\boldsymbol{V}^k - \boldsymbol{S}^{k+1}\right\|_F^2 + \frac{\lambda}{2}\left\|\boldsymbol{U}^{k+1} - \boldsymbol{U}^k\right\|_F^2 \leq \frac{1}{2}\left\|\boldsymbol{X} - \boldsymbol{U}^k\boldsymbol{V}^k - \boldsymbol{S}^{k+1}\right\|_F^2 \tag{25}$$

$$
\begin{aligned}
&\frac{1}{2}\left\|\boldsymbol{X}-\boldsymbol{U}^{k+1}\boldsymbol{V}^{k}-\boldsymbol{S}^{k+1}\right\|_F^2+\|\boldsymbol{S}^{k+1}\|_{g_e} \\
&\le \frac{1}{2}\left\|\boldsymbol{X}-\boldsymbol{U}^{k}\boldsymbol{V}^{k}-\boldsymbol{S}^{k+1}\right\|_F^2+\|\boldsymbol{S}^{k+1}\|_{g_e}-\frac{\lambda}{2}\left\|\boldsymbol{U}^{k+1}-\boldsymbol{U}^{k}\right\|_F^2
\end{aligned} \tag{26}
$$

$$
\mathcal{L}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k},\boldsymbol{S}^{k+1},e^{k+1}) \le \mathcal{L}(\boldsymbol{U}^{k},\boldsymbol{V}^{k},\boldsymbol{S}^{k+1},e^{k+1})-\frac{\lambda}{2}\left\|\boldsymbol{U}^{k+1}-\boldsymbol{U}^{k}\right\|_F^2 \tag{27}
$$

Finally, similar to the development of $\boldsymbol{U}^k$, when updating $\boldsymbol{V}^k$, it is easy to obtain:

$$
\mathcal{L}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^{k+1},e^{k+1}) \le \mathcal{L}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k},\boldsymbol{S}^{k+1},e^{k+1})-\frac{\lambda}{2}\left\|\boldsymbol{V}^{k+1}-\boldsymbol{V}^{k}\right\|_F^2 \tag{28}
$$

Adding (23), (24), (27) and (28), we thus have for all $k \ge 0$,

$$
\begin{aligned}
&\mathcal{L}(\boldsymbol{U}^{k},\boldsymbol{V}^{k},\boldsymbol{S}^{k},e^{k})-\mathcal{L}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^{k+1},e^{k+1}) \\
&\ge \frac{\lambda}{2}\left\|\boldsymbol{U}^{k+1}-\boldsymbol{U}^{k}\right\|_F^2+\frac{\lambda}{2}\left\|\boldsymbol{V}^{k+1}-\boldsymbol{V}^{k}\right\|_F^2
\end{aligned} \tag{29}
$$

From (29), we can conclude that the sequence $\{\mathcal{L}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k,e^k)\}_{k\in\mathbb{N}}$ is non-increasing and convergent since $\mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{S},e)$ is bounded from below.

Besides, let N be a positive integer, and we sum (29) from $k=0$ to $N-1$ to yield:

$$
\begin{aligned}
&\frac{\lambda}{2}\sum_{k=1}^{N-1}\left\|\boldsymbol{U}^{k+1}-\boldsymbol{U}^{k}\right\|_F^2+\frac{\lambda}{2}\sum_{k=1}^{N-1}\left\|\boldsymbol{V}^{k+1}-\boldsymbol{V}^{k}\right\|_F^2 \\
&\le \mathcal{L}(\boldsymbol{U}^{0},\boldsymbol{V}^{0},\boldsymbol{S}^{0},e^{0})-\mathcal{L}(\boldsymbol{U}^{N},\boldsymbol{V}^{N},\boldsymbol{S}^{N},e^{N})
\end{aligned} \tag{30}
$$

Since $\mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{S},e)$ is bounded below, when $N\to\infty$, we have:

$$
\begin{aligned}
&\lim_{N\to\infty}\left(\sum_{k=1}^{N-1}\left\|\boldsymbol{U}^{k+1}-\boldsymbol{U}^{k}\right\|_F^2+\sum_{k=1}^{N-1}\left\|\boldsymbol{V}^{k+1}-\boldsymbol{V}^{k}\right\|_F^2\right) \\
&\le \frac{2}{\lambda}\left(\mathcal{L}(\boldsymbol{U}^{0},\boldsymbol{V}^{0},\boldsymbol{S}^{0},e^{0})-\mathcal{L}(\boldsymbol{U}^{N},\boldsymbol{V}^{N},\boldsymbol{S}^{N},e^{N})\right)\le\infty
\end{aligned} \tag{31}
$$

Thus,

$$
\begin{aligned}
&\lim_{k\to\infty}\|\boldsymbol{U}^{k+1}-\boldsymbol{U}^{k}\|_F=0 \\
&\lim_{k\to\infty}\|\boldsymbol{V}^{k+1}-\boldsymbol{V}^{k}\|_F=0
\end{aligned} \tag{32}
$$

Moreover, it is easy to obtain:

$$
\begin{aligned}
&\partial_{\boldsymbol{U}}\mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{S},e)=\frac{\partial\mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{S},e)}{\partial\boldsymbol{U}}=(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}-\boldsymbol{S})\boldsymbol{V}^T \\
&\partial_{\boldsymbol{V}}\mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{S},e)=\frac{\partial\mathcal{L}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{S},e)}{\partial\boldsymbol{V}}=\boldsymbol{U}^T(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}-\boldsymbol{S})
\end{aligned} \tag{33}
$$

According to (14) and (15), we know:

$$
\begin{aligned}
&\boldsymbol{0}=\partial_{\boldsymbol{U}}\mathcal{L}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k},\boldsymbol{S}^{k+1},e^{k+1})+\lambda\left(\boldsymbol{U}^{k+1}-\boldsymbol{U}^{k}\right) \\
&\boldsymbol{0}=\partial_{\boldsymbol{V}}\mathcal{L}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^{k+1},e^{k+1})+\lambda\left(\boldsymbol{V}^{k+1}-\boldsymbol{V}^{k}\right)
\end{aligned} \tag{34}
$$

which amounts to:

$$
\begin{aligned}
&\boldsymbol{Q}_1=\partial_{\boldsymbol{U}}\mathcal{L}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^{k+1},e^{k+1}) \\
&\boldsymbol{Q}_2=\partial_{\boldsymbol{V}}\mathcal{L}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^{k+1},e^{k+1})
\end{aligned} \tag{35}
$$

where $\boldsymbol{Q}_1=-\lambda\left(\boldsymbol{U}^{k+1}-\boldsymbol{U}^{k}\right)-\partial_{\boldsymbol{U}}\mathcal{L}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k},\boldsymbol{S}^{k+1},e^{k+1})+\partial_{\boldsymbol{U}}\mathcal{L}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^{k+1},e^{k+1})=-\lambda\left(\boldsymbol{U}^{k+1}-\boldsymbol{U}^{k}\right)+\boldsymbol{X}\left(\boldsymbol{V}^{k+1}-\boldsymbol{V}^{k}\right)^T-\boldsymbol{U}^{k+1}\left(\boldsymbol{V}^{k+1}-\boldsymbol{V}^{k}\right)\left(\boldsymbol{V}^{k+1}+\boldsymbol{V}^{k}\right)^T-\boldsymbol{S}^{k+1}\left(\boldsymbol{V}^{k+1}-\boldsymbol{V}^{k}\right)^T$ and $\boldsymbol{Q}_2=-\lambda\left(\boldsymbol{V}^{k+1}-\boldsymbol{V}^{k}\right)$. Moreover, let $\{\boldsymbol{U}^{k_j}\}$ and $\{\boldsymbol{V}^{k_j}\}$ be the bounded subsequences of $\{\boldsymbol{U}^k\}$ and $\{\boldsymbol{V}^k\}$, respectively, produced by Algorithm 1 such that $\lim_{k_j\to\infty}\boldsymbol{U}^{k_j}=\boldsymbol{U}^*$ and $\lim_{k_j\to\infty}\boldsymbol{V}^{k_j}=\boldsymbol{V}^*$, then we can conclude that $(\boldsymbol{U}^*,\boldsymbol{V}^*)$ is the critical point of (7), because $\lim_{k_j\to\infty}\boldsymbol{Q}_1=\boldsymbol{0}$ and $\lim_{k_j\to\infty}\boldsymbol{Q}_2=\boldsymbol{0}$. This completes the proof. ■

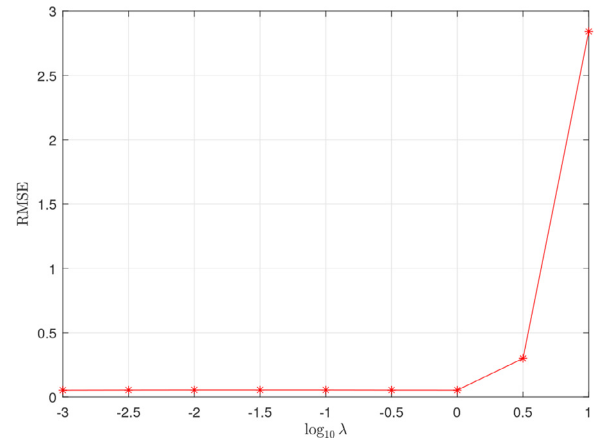

Fig. 1. RMSE versus $\log_{10}\lambda$.

4. Experimental results

In this section, our method is compared with four state-of-the-art RPCA algorithms, namely, IALM [12], GoDec+ [26], NCR-PCA [28] and WNNM-RPCA [14]. We evaluate these approaches using synthetic data as well as real datasets, and all experiments are run on a computer with 3.2 GHz CPU and 16 GB memory. Besides, for the parameters in the competing algorithms, we adopt their recommended setting. If it is not available, we determine their appropriate values via experiments. In the proposed algorithm, we set the maximum allowable outer iterations $I_m=100$ and $\eta=10^{-6}$.

4.1. Results of synthetic data

The synthetic data model in [40,41] is employed. Two random matrices $\boldsymbol{U}\in\mathbb{R}^{m\times r}$ and $\boldsymbol{V}\in\mathbb{R}^{r\times n}$, whose entries satisfy the standard Gaussian distribution, are generated to construct the synthetic matrix $\boldsymbol{X}=\boldsymbol{U}\boldsymbol{V}$. For convenience, we set $m=n$, and $r=m/50$. Impulsive noise generated by Gaussian mixture model (GMM) is added into $\boldsymbol{X}$. The probability density function of GMM is:

$$
p_v(v)=\frac{1-c}{\sqrt{2\pi}\sigma_1}\exp\left(-\frac{v^2}{2\sigma_1^2}\right)+\frac{c}{\sqrt{2\pi}\sigma_2}\exp\left(-\frac{v^2}{2\sigma_2^2}\right) \tag{36}
$$

where σ_1^2 and σ_2^2 are variances with $\sigma_1^2\ll\sigma_2^2$, and c controls the proportion of outliers. In our experiments, to model gross errors, we set $\sigma_2^2=100\sigma_1^2$ and $c=0.1$. The signal-to-noise ratio (SNR) of the impulsive noise is defined as:

$$
\mathrm{SNR}=\log_{10}\left(\frac{\|\boldsymbol{X}\|_F^2}{(1-c)\sigma_1^2+c\sigma_2^2}\right) \tag{37}
$$

To test the performance of all algorithms, the root mean square error (RMSE) is employed, given by:

$$
\mathrm{RMSE}=\frac{\|\boldsymbol{X}-\boldsymbol{M}\|_F}{\sqrt{mn}} \tag{38}
$$

where $\boldsymbol{M}=\boldsymbol{U}\boldsymbol{V}$.

We first discuss the choice of the parameter λ in the proposed algorithm, which is the weight of the proximal term in (14) and (15). Fig. 1 plots the curve of RMSE w.r.t. $\log_{10}\lambda$. As can be seen, RMSE is relatively stable when $\log_{10}\lambda\le 0$, while the error increases when $\log_{10}\lambda>0$. In this paper, we set $\lambda=0.001$.

In addition, all the algorithms are evaluated via different matrix dimensions and SNRs of GMM, and the average results by 20 independent runs are tabulated in Table 1. It is seen that

Table 1
Synthetic data results by different algorithms.

$m = 500$		RPCA-HQF	IALM	GoDec+	NCRPCA	WNNM-RPCA
3 dB	RMSE	$\mathbf{0.1508} \pm \mathbf{0.0027}$	0.4344 ± 0.0078	0.1744 ± 0.0029	0.7842 ± 0.0052	0.1852 ± 0.0034
	Runtime	0.0609	7.7019	**0.0549**	4.3212	10.489
6 dB	RMSE	$\mathbf{0.1074} \pm \mathbf{0.0019}$	0.3102 ± 0.0048	0.1242 ± 0.0022	0.7533 ± 0.0052	0.1324 ± 0.0023
	Runtime	0.0575	9.4670	**0.0499**	4.1120	8.6189
9 dB	RMSE	$\mathbf{0.0757} \pm \mathbf{0.0011}$	0.2190 ± 0.0034	0.0873 ± 0.0012	0.7191 ± 0.0037	0.0933 ± 0.0014
	Runtime	0.0588	10.842	**0.0449**	4.1463	7.2874
12 dB	RMSE	$\mathbf{0.0534} \pm \mathbf{0.0006}$	0.1551 ± 0.0017	0.0614 ± 0.0009	0.6826 ± 0.0020	0.0659 ± 0.0010
	Runtime	0.0579	11.301	**0.0435**	3.9881	6.5494
15 dB	RMSE	$\mathbf{0.0383} \pm \mathbf{0.0008}$	0.1108 ± 0.0021	0.0440 ± 0.0010	0.6568 ± 0.0020	0.0426 ± 0.0011
	Runtime	0.0585	10.089	**0.0410**	3.8308	5.6711
$m = 1000$		RPCA-HQF	IALM	GoDec+	NCRPCA	WNNM-RPCA
3 dB	RMSE	$\mathbf{0.2141} \pm \mathbf{0.0020}$	0.6185 ± 0.0054	0.2477 ± 0.0022	0.7465 ± 0.0142	0.4387 ± 0.0115
	Runtime	0.2305	61.639	**0.2301**	26.834	124.58
6 dB	RMSE	$\mathbf{0.1514} \pm \mathbf{0.0012}$	0.4372 ± 0.0033	0.1747 ± 0.0013	0.6240 ± 0.0058	0.1860 ± 0.0014
	Runtime	0.2261	57.477	**0.2133**	25.253	89.090
9 dB	RMSE	$\mathbf{0.1072} \pm \mathbf{0.0008}$	0.3104 ± 0.0023	0.1236 ± 0.0009	0.5768 ± 0.0044	0.1322 ± 0.0010
	Runtime	0.2296	72.376	**0.1909**	23.940	65.024
12 dB	RMSE	$\mathbf{0.0757} \pm \mathbf{0.0006}$	0.2196 ± 0.0013	0.0873 ± 0.0006	0.5448 ± 0.0026	0.0935 ± 0.0006
	Runtime	0.2328	68.210	**0.1802**	23.243	51.074
15 dB	RMSE	$\mathbf{0.0536} \pm \mathbf{0.0004}$	0.1553 ± 0.0012	0.0616 ± 0.0005	0.5110 ± 0.0017	0.0661 ± 0.0006
	Runtime	0.2294	81.534	**0.1632**	22.393	39.355

where Runtime is in seconds.

the proposed method has smaller RMSEs, implying that the former achieves more accurate low-rank recovery, compared with the competing algorithms. Moreover, RPCA-HQF requires less computational time than IALM, NCRPCA and WNNM-RPCA, and has comparable runtime to GoDec+. We now explain why RPCA-HQF attains good recovery performance in the following. First, it is known that the ℓ_1-norm is more vulnerable to big outliers compared to non-convex functions, because the influence function of the former is not redescending [7,41]. Furthermore, the ℓ_1-norm as the regularization term leads to the bias problem [28,42]. Since IALM and WNNM-RPCA employ the ℓ_1-norm to resist outliers and achieve sparseness, RPCA-HQF is superior to them, because we use non-convex regularization. For the GoDec+ and NCRPCA, although they leverage non-convex functions, i.e., Welsch function and ℓ_p-norm ($0 < p < 1$), respectively, to resist gross errors, the parameters that control robustness in their methods are set manually, compared to RPCA-HQF, whose parameter e is adjusted in a data-driven manner according to MADN.

4.2. Real-world applications

In this subsection, we apply the proposed algorithm on two typical RPCA applications [10], i.e., background /foreground separation for videos and shadow/specularity removal for face images, to validate its effectiveness.

4.2.1. Background modeling from video

We first test all the methods on the video foreground-background separation problem since RPCA can decompose the video into low-rank and sparse components, corresponding to static background and moving objects, respectively. Four videos from CDnet 2014 [39] dataset, namely, cubicle (240×352 for each frame), blizzard (240×360), skating (180×270), overpass (240×320) and park (288×352), are employed. We choose 200 successive frames for each video and stack each frame as a column to construct $\boldsymbol{X}$. Taking cubicle as an example, after frame vectorization, we have $\boldsymbol{X} \in \mathbb{R}^{84480 \times 200}$. Besides, Gaussian noise with variance being 0.001 is added into the data matrix. Since the background is almost static, we set the rank $r = 1$. The foreground-background separation results by different methods are shown in Fig. 2. It is easy to observe that the recovered backgrounds by our algorithm are more visually clear than IALM, GoDec+ and NCRPCA. For example, the backgrounds obtained by IALM, GoDec+ and NCRPCA, contain ghost, and the foregrounds obtained by our algorithm are clearer, compared with that by the competing methods, which contain noticeable Gaussian noise. Four performance metrics, namely, iteration number (Iter.), runtime in seconds, sparseness in foreground ($\|\boldsymbol{S}\|_0/mn$) and F-measure, are used. For the F-measure, given true positive (TP), false positive (FP), false negative (FN) and true negative (TN), it is defined as [43]:

$$\mathrm{F_m} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{39}$$

where $\text{precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$ and $\text{recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$. Numerical average evaluation results by 20 independent runs for all algorithms are tabulated in Table 2. Although both RPCA-HQF and WNNM-RPCA can attain good low-rank background recovery, the former needs

Table 2
Recovered results of four videos by different algorithms .

		RPCA-HQF	IALM	GoDec+	NCRPCA	WNNM-RPCA
cubicle	Iter.	**6**	128	7	63	171
	Runtime	3.6238	99.847	**1.2171**	65.708	66.130
	$\frac{\|\boldsymbol{S}\|_0}{mn}$	**0.0499**	0.9541	1	1	0.9709
	$\mathrm{F_m}$	0.7295	0.5516	0.4220	0.5897	**0.7299**
blizzard	Iter.	**6**	117	20	62	172
	Runtime	3.8491	92.069	**2.1405**	60.075	66.433
	$\frac{\|\boldsymbol{S}\|_0}{mn}$	**0.0290**	0.9602	1	1	0.9952
	$\mathrm{F_m}$	**0.7204**	0.7101	0.6398	0.6630	0.7163
skating	Iter.	**6**	120	9	62	171
	Runtime	2.0748	53.269	**0.8001**	32.868	39.243
	$\frac{\|\boldsymbol{S}\|_0}{mn}$	**0.0601**	0.9446	1	1	0.9946
	$\mathrm{F_m}$	**0.6322**	0.5793	0.5155	0.5796	0.6309
overpass	Iter.	**6**	136	14	100	171
	Runtime	3.2581	95.762	**1.8741**	88.3182	78.802
	$\frac{\|\boldsymbol{S}\|_0}{mn}$	**0.0889**	0.9158	1	1	0.9954
	$\mathrm{F_m}$	**0.5685**	0.4146	0.4395	0.4847	0.5678
park	Iter.	**6**	107	20	100	169
	Runtime	4.5556	99.547	**3.5304**	109.06	112.53
	$\frac{\|\boldsymbol{S}\|_0}{mn}$	**0.0219**	0.9660	1	1	0.9958
	$\mathrm{F_m}$	**0.4933**	0.4856	0.4821	0.4747	0.4923

where $\|\boldsymbol{S}\|_0$ denotes the total number of non-zero elements in $\boldsymbol{S}$, and m and n are matrix column and row lengths.

Fig. 2. Background and foreground separation results of different algorithms on the real videos. Images from (a) to (b) are original images, recovered results by RPCA-HQF, IALM, GoDec+, NCRPCA and WNNM-RPCA, respectively.

shorter computational time, and the reason why our method has clearer foreground is that RPCA-HQF has smaller $\|\boldsymbol{S}\|_0/mn$.

4.2.2. Shadow and specularity removal from face images

Another popular application of RPCA is to remove shadows and specularities from face images. Although different lighting conditions may create challenges for face recognition, if we have enough face images of the same person, RPCA methods can extract the face features and remove the shadows and specularities since the latter are sparse and have large magnitudes. The extended Yale B dataset [26], which includes 38 human subjects, is utilized to compare different algorithms. There are about 64 images under 9 poses and 64 illumination conditions with dimensions being 192×168 for each subject. To evaluate the robustness [26] of all the methods, 10 dB Gaussian noise, 10 dB impulsive noise generated by GMM model and random occlusions are added to the images, and the restoration results are shown in Fig. 3. We observe that RPCA-HQF extracts the face features well (the second column corresponds to the recovered images obtained by RPCA-HQF) and attains clearer recovery results than other algorithms. Table 3 tabulates the average iteration number and runtime via 20 independent runs for different approaches, and the proposed method requires the least runtime, compared to the competing algorithms.

Fig. 3. Shadow and specularity removal from faces. Images from column (b) to (f) are recovered images by RPCA-HQF, IALM, GoDec+, NCRPCA and WNNM-RPCA, respectively. In column (a), face-1 to face-4 from *Subject* 15 are original images, corrupted images by Gaussian noise, GMM noise and occlusions, respectively. Similarly, face-5 to face-8 from *Subject* 25 are original images, corrupted images by Gaussian noise, GMM noise and occlusions, respectively.

Table 3
Runtime comparison for different algorithms.

Method	RPCA-HQF		IALM		GoDec+		NCRPCA		WNNM-RPCA	
	Iter.	Runtime	Iter.	Runtime	Iter.	Runtime	Iter.	Runtime	Iter.	Runtime
face-1	**6**	**0.3393**	223	16.087	22	0.4009	59	5.1247	178	10.674
face-2	**6**	**0.4254**	183	12.651	26	0.4668	100	8.7044	178	10.847
face-3	**6**	**0.3960**	201	14.529	26	0.4764	100	9.4420	178	10.531
face-4	**6**	**0.3329**	232	17.040	22	0.4023	58	4.9986	176	10.606
face-5	**6**	**0.3691**	230	17.219	34	0.6422	58	5.2003	178	11.582
face-6	**6**	**0.4560**	189	13.462	36	0.6656	100	8.9083	178	11.416
face-7	**6**	**0.4275**	209	15.129	35	0.6545	100	8.8825	178	11.204
face-8	**6**	**0.3645**	242	17.602	27	0.5102	58	4.7273	176	11.479

5. Conclusion

In this paper, to avoid SVD calculation, we devise an efficient RPCA algorithm based on factorization and HQF regularization. Although HQF is non-convex, we solve the resultant optimization problem efficiently since its proximity operator is derived. In addition, proximal BCD is utilized to find the solutions to the low-rank components, and the complexity as well as convergence of our method are analyzed. Our experimental results demonstrate that compared with IALM, GoDec+, NCRPCA and WNNM-RPCA, RPCA-HQF achieves more accurate low-rank and sparse restoration performance. Furthermore, the loss function and the corresponding theoretical analysis can be extended to one-dimensional sparse recovery and tensor recovery in the presence of outliers.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Zhi-Yong Wang: Conceptualization, Methodology, Software, Writing – original draft, Data curation. **Xiao Peng Li:** Methodology, Validation, Conceptualization, Writing – review & editing. **Hing Cheung So:** Supervision, Validation, Writing – review & editing. **Zhaofeng Liu:** Validation, Writing – review & editing.

Acknowledgment

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 11207922].

References

[1] I.T. Jolliffe, Principal component analysis, (Springer Series in Statistics), vol. 29, New York, NY, USA: Springer-Verlag, 2002.
[2] N. Vaswani, Y. Chi, T. Bouwmans, Rethinking PCA for modern data sets: theory, algorithms, and applications, Proc. IEEE 106 (8) (2018) 1274–1276.
[3] Y.Y. Zhao, C.N. Jiao, M.L. Wang, J.X. Liu, J. Wang, C.H. Zheng, HTRPCA: Hypergraph regularized tensor robust principal component analysis for sample clustering in tumor omics data, Interdiscip. Sci. Comput. Life Sci. (2021) 22–33.
[4] J.X. Liu, Y. Xu, C.H. Zheng, H. Kong, Z.H. Lai, RPCA Based tumor classification using gene expression data, IEEE/ACM Trans. Comput. Biology Bioinf. 12 (4) (2015) 964–970.
[5] T. Bouwmans, S. Javed, H. Zhang, Z. Lin, R. Otazo, On the applications of robust PCA in image and video processing, Proc. IEEE 106 (8) (2018) 1427–1457.
[6] A.M. Zoubir, V. Koivunen, Y. Chakhchoukh, M. Muma, Robust estimation in signal processing: atutorial-style treatment of fundamental concepts, IEEE Signal Process. Mag. 29 (4) (2012) 61–80.
[7] A.M. Zoubir, V. Koivunen, E. Ollila, M. Muma, Robust statistics for signal processing, Cambridge, MA, USA: Cambridge Univ. Press, 2018.
[8] J. Wright, A. Ganesh, S. Rao, Y. Peng, Y. Ma, Robust Principal Component Analysis: Exact Recovery of Corrupted Low-rank Matrices via Convex Optimization, in: Proc. Adv. Neural Inf. Process. Syst., Vancouver, Canada, 2009, pp. 2080–2088.
[9] J. Wright, Y. Ma, Dense error correction via ℓ_1-minimization, IEEE Trans. Inf. Theory 56 (7) (2010) 3540–3560.
[10] E.J. Candés, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM 58 (3) (2011) 11.
[11] Z. Lin, R. Liu, Z. Su, Linearized Alternating Direction Method with Adaptive Penalty for Low-rank Representation, in: Proc. Adv. Neural Inf. Process. Syst., Granada, Spain, 2011, pp. 612–620.
[12] X. Yuan, J.F. Yang, Sparse and low-rank matrix decomposition via alternating direction methods, Pac. J. Optim. 9 (1) (2013) 167–180.
[13] Z. Zhou, X. Li, J. Wright, E.J. Candés, Y. Ma, Stable Principal Component Pursuit, in: Proc. IEEE Int. Symp. Inf. Theory, 2010, pp. 1518–1522.
[14] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, L. Zhang, Weighted nuclear norm minimization and its applications to low level vision, Int. J. Comput. Vis. 121 (2) (2017) 183–208.
[15] 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.
[16] N.S. Aybat, D. Goldfarb, S. Ma, Efficient algorithms for robust and stable principal component pursuit problems, Comput. Optim. Appl. 58 (1) (2014) 1–29.
[17] T. Zhou, D. Tao, Godec: Randomized Low-rank & Sparse Matrix Decomposition in Noisy Case, in: Proc. 28th ICML, Bellevue, WA, USA, 2011, pp. 33–40.
[18] Y. Shen, Z. Wen, Y. Zhang, Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization, Optim. Methods Softw. 29 (2) (2014) 239–263.
[19] T. Zhou, D. Tao, Greedy Bilateral Sketch, Completion & Smoothing, in: Proc. JMLR Workshop Conf., Scottsdale, AZ, USA, volume 31, 2013, pp. 650–658.
[20] X. Ding, L. He, L. Carin, Bayesian robust principal component analysis, IEEE Trans. Image Process. 20 (12) (2011) 3419–3430.
[21] S.D. Babacan, M. Luessi, R. Molina, A.K. Katsaggelos, Sparse bayesian methods for low-rank matrix estimation, IEEE Trans. Signal Process. 60 (8) (2012) 3964–3977.
[22] J. Liu, B.D. Rao, Sparse bayesian learning for robust PCA: algorithms and analyses, IEEE Trans. Signal Process. 67 (22) (2019) 5837–5849.
[23] J. Fan, R. Li, Variable selection via nonconcave penalized likelihood and its oracle properties, J. Am. Statist. Assoc. 96 (456) (2001) 1348–1360.
[24] J. Trzasko, A. Manduca, Highly undersampled magnetic resonance image reconstruction via homotopic ℓ_0-minimization, IEEE Trans. Med. Imag. 28 (1) (2009) 106–121.
[25] R. He, T. Tan, L. Wang, Robust recovery of corrupted low-rank matrix by implicit regularizers, IEEE Trans. Pattern Anal. Mach. Intell. 36 (4) (2014) 770–783.
[26] K. Guo, L. Liu, X. Xu, D. Xu, D. Tao, Godec+: fast and robust low-rank matrix decomposition based on maximum correntropy, IEEE Trans. Neural Netw. Learn. Syst. 29 (6) (2018) 2323–2336.
[27] K.G. Quach, C.N. Duong, K. Luu, T.D. Bui, Non-convex online robust PCA: enhance sparsity via ℓ_p-norm minimization, Comput. Vis. Image Understand. 158 (2017) 126–140.
[28] F. Wen, R. Ying, P. Liu, R.C. Qiu, Robust PCA using generalized nonconvex regularization, IEEE Trans. Circuits Syst. Video Technol. 30 (6) (2020) 1497–1510.
[29] A. Mousavi, M. Rezaee, R. Ayanzadeh, A survey on compressive sensing: classical results and recent advancements, J. Math. Model. 8 (3) (2020) 309–344.
[30] M.J. Abdi, Cardinality optimization problems, Sch. Math., Univ. Birmingham, Birmingham, U.K., 2013 Ph.d. dissertation.
[31] Y. Xu, W. Yin, A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion, SIAM J. Imag. Sci. 6 (3) (2013) 1758–1789.
[32] Z.-Q. Luo, P. Tseng, On the convergence of the coordinate descent method for convex differentiable minimization, J. Oper. Theory Appl. 72 (1) (1992) 7–35.
[33] J.P. Haldar, D. Hernando, Rank-constrained solutions to linear matrix equations using powerfactorization, IEEE Signal Process. Lett. 16 (7) (2009) 584–587.
[34] M. Hardt, On the provable convergence of alternating minimization for matrix completion, [Online]. Available: http://arxivweb3.library.cornell.edu/abs/1312.0925v1.
[35] R.A. Maronna, D.R. Martin, V.J. Yohai, Robust statistics: Theory and methods, England: Wiley, 2006.
[36] Z.-Y. Wang, H.C. So, Z. Liu, Fast and robust rank-one matrix completion via maximum correntropy criterion and half-quadratic optimization, Signal Process. 198 (2022) 108580.

[37] Q. Yao, J.T. Kwok, T. Wang, T.Y. Liu, Large-scale low-rank matrix learning with nonconvex regularizers, IEEE Trans. Pattern Anal. Mach. Intell. 41 (11) (2019) 2628–2643.
[38] H. Attouch, J. Bolte, P. Redont, A. Soubeyran, Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the kurdyka-ojasiewicz inequality, Math. Oper. Res. 35 (2) (2010) 438–457.
[39] Q. Liu, X. Li, Efficient low-rank matrix factorization based on $\ell_{1,\varepsilon}$-norm for online background subtraction, IEEE Trans. Circuits Syst. Video Technol., doi: 10.1109/TCSVT.2021.3129503, Early access.
[40] W.-J. Zeng, H.C. So, Outlier-robust matrix completion via ℓ_p-minimization, IEEE Trans. Signal Process. 66 (5) (2018) 1125–1140.
[41] M. Muma, W.-J. Zeng, A.M. Zoubir, Robust M-estimation Based Matrix Completion, in: 2019 IEEE Int. Conf. on Acoust., Speech and Signal Process., 2019, pp. 5476–5480. Brighton, UK
[42] M. Nikolova, Thresholding implied by truncated quadratic regularization, IEEE Trans. Signal Process. 48 (11) (2000) 3437–3450.
[43] T. Bouwmans, A. Sobral, S. Javed, S.K. Jung, E.H. 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.