Robust PCA via non-convex half-quadratic regularization

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

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

**A R T I C L E  I N F O**

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

**A B S T R A C T**

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.

© 2022 Elsevier B.V. All rights reserved.

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 ℓ₀-norm minimization problem, but it is NP-hard as the rank function is discrete. In [9,10], the authors replace the rank function and ℓ₀-norm with the nuclear norm and ℓ₁-norm, respectively, to obtain a convex optimization problem since many ready-made convex optimization methods such as the inexact augmented Lagrangian 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, i.e., $\mathbf{U}$ and $\mathbf{V}$, 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 PCA (VBPCA) [21] and sparse Bayesian learning for RPCA (SBLR-PCA) [22], are proposed. However, they leverage the ℓ₁-norm as regularization penalty to achieve sparseness, and it has been pointed out that the ℓ₁-norm over-penalizes the data with large
magnitudes, resulting in a biased solution [23]. In contrast, non-convex regularization penalties, such as smoothly clipped absolute deviation (SCAD) [23], $\ell_p$-norm ($0 < p < 1$) and Laplacian penalty [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 $\ell_p$-norm based robust PCA (LPRPCA) [27] and non-convex regularized robust PCA (NCRPCA) [28]. However, the SCAD requires two user-defined parameters, while the $\ell_p$-norm (except for $p = \frac{1}{2}$ and $p = 2$) and Laplacian 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 proximty 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 $\{U^k, V^k\}$ 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 $A$ is represented by $A_{ij}$ and $\mathbf{0}$ means a matrix with all entries being zeros. In addition, the Frobenius norm of $A \in \mathbb{R}^{m \times n}$ is denoted by $\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |A_{ij}|^2}$, and unless stated otherwise, the matrix norm refers to the Frobenius norm. Moreover, $\|S\|_0$ signifies the number of non-zero entries of $S$. 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]:

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

where $X \in \mathbb{R}^{m \times n}$ is the observed low-rank matrix corrupted by outliers, $L \in \mathbb{R}^{m \times n}$ is the low-rank matrix, $S \in \mathbb{R}^{m \times n}$ is the sparse matrix, and $\|S\|_0$ denotes the cardinality of $S$ (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 $\ell_1$-norm are employed to replace the rank function and $\ell_0$-norm, respectively, resulting in the convex relaxation of (1) [9,10]:

$$\min_{L,S} \|L\|_* + \lambda \|S\|_1, \text{ s.t. } X = L + S$$  \hspace{1cm} (2)

where $\|L\|_*$ is the nuclear norm, which is the sum of singular values of $L$. Although (2) can recover $L$ and $S$ 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 (S SCP) [13]:

$$\min_{U,V,S} \|X - UV - S\|_F^2 + \lambda \|S\|_1$$  \hspace{1cm} (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:

$$\min_{U,V,S} \|X - U(V^T S)\|_F^2 + \lambda \|S\|_p$$  \hspace{1cm} (4)

where $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{r \times n}$. Since the $\ell_1$-norm regularization brings about the bias problem, non-convex regularization is introduced [26], resulting in:

$$\min_{U,V,S} \|X - U(V^T S)\|_F^2 + \lambda \|S\|_p$$  \hspace{1cm} (5)

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

### 3. Proposed algorithm

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

$$g_e(t) = \begin{cases} \frac{-(|t| - e)^2}{2} + \frac{e^2}{2}, & |t| < e \\ \frac{e^2}{2}, & |t| \geq e \end{cases}$$  \hspace{1cm} (6)

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

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

$$\min_{U,V,S} \frac{1}{2} \|X - U(V^T S)\|_F^2 + \|S\|_{g_e}$$  \hspace{1cm} (7)

where $\|S\|_{g_e}$ is separable, namely, $\|S\|_{g_e} = \sum_{i=1}^{m} \sum_{j=1}^{n} g_e(S_{ij})$. Different from (4) that employs $\lambda$ 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 $S$. Given $U^k$ and $V^k$, (7) is equal to:

$$S^{k+1} = \arg \min_{S} \frac{1}{2} \|R^k - S\|_F^2 + \|S\|_{g_e}$$  \hspace{1cm} (8)

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

$$P(t) := \arg \min_{t} \left\{ \frac{1}{2} (r - t)^2 + g_e(t) \right\}$$  \hspace{1cm} (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:

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

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

$$P(r) = \arg \min_t \frac{(r-t)^2}{2} + g_e(t)$$

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

The proof is complete.

Therefore, the solution to (8) is:

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

It is worth mentioning that $e$ controls the robustness of (7) because any entry $R_i \geq e$ 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,

$$\sigma^k = 1.4815 \times \text{Med}(|\text{vec}(R)| - \text{Med}(\text{vec}(R)))$$ (12)

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

$$e^k = \min \{\xi \sigma^k, e^{k-1}\}$$ (13)

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

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

$$U^{k+1} = \arg \min_U \left\{ \frac{1}{2} \|X - UV^k - S^{k+1}\|_F^2 + \frac{\lambda}{2} \|U - U^k\|_F^2 \right\}$$ (14)

$$V^{k+1} = \arg \min_V \left\{ \frac{1}{2} \|X - U^{k+1}V - S^{k+1}\|_F^2 + \frac{\lambda}{2} \|V - V^k\|_F^2 \right\}$$ (15)

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

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

$$U^{k+1} = (D^{k+1}V^k)^T - \lambda U^k$$ (16)

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

Similarly, $V^{k+1} = (U^{k+1})^T - \lambda V^k$(17)

whose computational complexity is also $O(mn^2)$.

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

$$\min_{U,V} \frac{1}{2} \|X - UV\|_F^2$$ (18)

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

$$U^{n+1} = X(V^n)^T$$ (19)

$$V^{n+1} = X(U^{n+1})^T$$ (20)

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 $l_0 = 3$ in this study. Besides, Algorithm 1 will be terminated when $\eta = (\|R_i\|_F - \|R_i^{k+1}\|_F) / \sqrt{m \times n}$ is less than a preset threshold value $\eta$ and/or the iteration number reaches the maximum allowable number of outer iterations $l_m$. Moreover, we define $\mathcal{L}(U, V, S, e) = \frac{1}{2} \|X - UV - S\|_F^2 + \|S\|_E$, 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 $0 = \delta f(x^*)$ [37,38].

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

Proof: First, we prove that when updating $e^k \rightarrow e^{k+1}$ via (13),

$$\mathcal{L}(U^{k+1}, V^k, S^{k+1}, e^{k+1}) \leq \mathcal{L}(U^k, V^k, S^k, e^k), \quad \|S\|_E \leq \sum_{i,j} \|S_{i,j}\|_E$$

by taking the partial derivative of $\mathcal{L}(S_{i,j})$ with respect to $m$-wise $e$, we have $g_e(S_{i,j}) > 0$, because

$$\frac{\partial g_e(S_{i,j})}{\partial e} = \frac{\partial}{\partial e} \left( \frac{|S_{i,j}|}{e} I \right) = \frac{1}{e} I \cdot I = \sum_{i,j} \frac{\partial g_e(S_{i,j})}{\partial e} > 0$$ (22)

and it is easy to obtain:

$$\frac{\partial \mathcal{L}(U, V, S, e)}{\partial e} = \frac{\partial}{\partial e} \sum_{i,j} g_e(S_{i,j}) = \sum_{i,j} \frac{\partial g_e(S_{i,j})}{\partial e} > 0$$ (23)

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

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

which is due to the proximity operator (10).

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

$$\frac{1}{2} \|X - UV^k - S^{k+1}\|_F^2 + \frac{1}{2} \|U^{k+1} - U^k\|_F^2 \leq \frac{1}{2} \|X - UV^k - S^{k+1}\|_F^2$$ (25)
\[
\frac{1}{2} \| X - U^{k+1} V^k - S^{k+1} \|_F^2 + \| S^{k+1} \|_F^2 \\
\leq \frac{1}{2} \| X - U^k V^k - S^{k+1} \|_F^2 + \| S^{k+1} \|_F^2 + \frac{\lambda}{2} \| U^{k+1} - U^k \|_F^2 
\]  
\begin{equation}
(26)
\end{equation}

\[
\mathcal{L}(U^{k+1}, V^k, S^{k+1}, e^{k+1}) \leq \mathcal{L}(U^k, V^k, S^{k+1}, e^{k+1}) - \frac{\lambda}{2} \| U^{k+1} - U^k \|_F^2 
\]  
\begin{equation}
(27)
\end{equation}

Finally, similar to the development of \( U^k \), when updating \( V^k \), it is easy to obtain:

\[
\mathcal{L}(U^{k+1}, V^k, S^{k+1}, e^{k+1}) \leq \mathcal{L}(U^{k+1}, V^k, S^{k+1}, e^{k+1}) - \frac{\lambda}{2} \| V^{k+1} - V^k \|_F^2 
\]  
\begin{equation}
(28)
\end{equation}

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

\[
\mathcal{L}(U^k, V^k, S^k, e^k) - \mathcal{L}(U^{k+1}, V^k, S^{k+1}, e^{k+1}) \\
\geq \frac{\lambda}{2} \| U^{k+1} - U^k \|_F^2 + \frac{\lambda}{2} \| V^{k+1} - V^k \|_F^2 
\]  
\begin{equation}
(29)
\end{equation}

From (29), we can conclude that the sequence \( \{ \mathcal{L}(U^k, V^k, S^k, e^k) \}_{k=1}^\infty \) is non-increasing and convergent since \( \mathcal{L}(U, V, S, e) \) is bounded from below.

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

\[
\frac{\lambda}{2} \sum_{k=1}^{N-1} \| U^{k+1} - U^k \|_F^2 + \frac{\lambda}{2} \sum_{k=1}^{N-1} \| V^{k+1} - V^k \|_F^2 \\
\leq \mathcal{L}(U^0, V^0, S^0, e^0) - \mathcal{L}(U^N, V^N, S^N, e^N) \leq \infty 
\]  
\begin{equation}
(30)
\end{equation}

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

\[
\lim_{N \to \infty} \left( \sum_{k=1}^{N-1} \| U^{k+1} - U^k \|_F^2 + \sum_{k=1}^{N-1} \| V^{k+1} - V^k \|_F^2 \right) \\
\leq \frac{\lambda}{2} \mathcal{L}(U^0, V^0, S^0, e^0) - \mathcal{L}(U^N, V^N, S^N, e^N) \leq \infty 
\]  
\begin{equation}
(31)
\end{equation}

Thus,

\[
\lim_{k \to \infty} \| U^{k+1} - U^k \|_F = 0 \\
\lim_{k \to \infty} \| V^{k+1} - V^k \|_F = 0 
\]  
\begin{equation}
(32)
\end{equation}

Moreover, it is easy to obtain:

\[
\partial_U \mathcal{L}(U, V, S, e) = \frac{\partial \mathcal{L}(U, V, S, e)}{\partial U} = (X - UV - S) V^T 
\]  
\begin{equation}
(33)
\end{equation}

\[
\partial_V \mathcal{L}(U, V, S, e) = \frac{\partial \mathcal{L}(U, V, S, e)}{\partial V} = U^T (X - UV - S) 
\]  
\begin{equation}
(34)
\end{equation}

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

\[
0 = \partial_U \mathcal{L}(U^{k+1}, V^k, S^{k+1}, e^{k+1} + \lambda (U^{k+1} - U^k)) \\
0 = \partial_V \mathcal{L}(U^{k+1}, V^k, S^{k+1}, e^{k+1}) + \lambda (V^{k+1} - V^k) 
\]  
\begin{equation}
(35)
\end{equation}

which amounts to:

\[
Q_1 = \partial_U \mathcal{L}(U^{k+1}, V^k, S^{k+1}, e^{k+1}) \\
Q_2 = \partial_V \mathcal{L}(U^{k+1}, V^k, S^{k+1}, e^{k+1}) 
\]  
\begin{equation}
(36)
\end{equation}

where

\[
Q_1 = -\lambda (U^{k+1} - U^k) - \partial_U \mathcal{L}(U^{k+1}, V^k, S^{k+1}, e^{k+1}) + \\
\partial_U \mathcal{L}(U^{k+1}, V^k, S^{k+1}, e^{k+1}) = \lambda (U^{k+1} - U^k) + X (V^{k+1} - V^k)^T - \\
U^{k+1} (V^{k+1} - V^k)^T (V^{k+1} - V^k)^T - S^{k+1} (V^{k+1} - V^k)^T 
\]  
\begin{equation}
(37)
\end{equation}

and \( Q_2 = -\lambda (V^{k+1} - V^k) \). Moreover, let \( \{ U^k \} \) and \( \{ V^k \} \) be the bounded subsequences of \( \{ U^k \} \) and \( \{ V^k \} \), respectively, produced by Algorithm 1 such that \( \lim_{k \to \infty} U^k = U^0 \) and \( \lim_{k \to \infty} V^k = V^0 \), then we can conclude that \( \{ U^k, V^k \} \) is the critical point of (7), because \( \lim_{k \to \infty} Q_1 = 0 \) and \( \lim_{k \to \infty} Q_2 = 0 \). This completes the proof.  

4. Experimental results

In this section, our method is compared with four state-of-the-art RPCA algorithms, namely, IALM [12], GoDec+ [26], NCRPCA [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 \( l_m = 100 \) and \( \eta = 10^{-6} \).

4.1. Results of synthetic data

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

\[
p_n(v) = \frac{1 - c}{\sqrt{2\pi} \sigma_1} \exp \left( -\frac{v^2}{2\sigma_1^2} \right) + \frac{c}{\sqrt{2\pi} \sigma_2} \exp \left( -\frac{v^2}{2\sigma_2^2} \right) 
\]  
\begin{equation}
(38)
\end{equation}

where \( \sigma_1^2 \) and \( \sigma_2^2 \) 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 impulse noise is defined as:

\[
\text{SNR} = \log_{10} \left( \frac{\| X \|^2}{(1-c) \sigma_1^2 + c \sigma_2^2} \right) 
\]  
\begin{equation}
(39)
\end{equation}

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

\[
\text{RMSE} = \frac{\| X - M \|_F}{\sqrt{mn}} 
\]  
\begin{equation}
(40)
\end{equation}

where \( M = UV \).

We first discuss the choice of the parameter \( \lambda \) 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 \( \log_{10} \lambda \leq 0 \), 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
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 descending [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 \(c\) 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 \(\times\) 352 for each frame), blizzard (240 \(\times\) 360), skating (180 \(\times\) 270), overlap (240 \(\times\) 320) and park (288 \(\times\) 352), are employed. We choose 200 successive frames for each video and stack each frame as a column to construct \(X\). Taking cubicle as an example, after frame vectorization, we have \(X \in \mathbb{R}^{94480 \times 200}\). 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 \(\text{(Iter.)}\), runtime in seconds, sparseness in foreground \((\|S\|_0/mn)\) 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]:

\[
F_m = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]  

where precision = \(\frac{TP}{TP + FP}\) and recall = \(\frac{TP}{TP + FN}\). 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>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>m = 500</td>
</tr>
<tr>
<td>3 dB</td>
</tr>
<tr>
<td>6 dB</td>
</tr>
<tr>
<td>9 dB</td>
</tr>
<tr>
<td>12 dB</td>
</tr>
<tr>
<td>15 dB</td>
</tr>
</tbody>
</table>

where Runtime is in seconds.

<table>
<thead>
<tr>
<th>Table 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPCA-HQF</td>
</tr>
<tr>
<td>cubicle</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>blizzard</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>skating</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>overlap</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>park</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

where \(|S|_0\) denotes the total number of non-zero elements in \(S\), and \(m\) and \(n\) are matrix column and row lengths.
shorter computational time, and the reason why our method has clearer foreground is that RPCA-HQF has smaller $\|S\|_0/mn$.

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 $192 \times 168$ 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. 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.
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.
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 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 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. Xiang 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


