Adaptive Weighted Robust Principal Component Analysis[1]

Zhengqin Xu
School of Machinery and Automation, Institute of Robustics and Intelligent Systems, Wuhan University of Science and Technology
Wuhan , China
xuzhengqin@wust.edu.cn

Yang Lu
School of Machinery and Automation, Institute of Robustics and Intelligent Systems, Wuhan University of Science and Technology
Wuhan , China
luyanglymj@foxmail.com

Jiaxing Wu
School of Information Science and Engineering, Institute of Robustics and Intelligent Systems, Wuhan University of Science and Technology
Wuhan , China
wust_jiaxin.wu@hotmail.com

Rui He
School of Information Science and Engineering, Institute of Robustics and Intelligent Systems, Wuhan University of Science and Technology
Wuhan , China
herui@wust.edu.cn

Shiqian Wu
School of Information Science and Engineering, Institute of Robustics and Intelligent Systems, Wuhan University of Science and Technology
Wuhan , China
shiqian.wu@wust.edu.cn

Shoulie Xie
Signal Processing, RF & Optical Dept. Institute for Infocomm Research
A*STAR, Singapore
slxie@i2r.a-star.edu.sg

***Abstract*— Robust principal component analysis (RPCA) via the nuclear norm minimization (NNM) is a powerful tool for image processing problems. However, most of NNM methods only consider the number of non-zero singular values of the observation matrix, and ignore the different proportions of data information in different singular values, which are related to the exact rank of clean data and should be treated differently. In this paper, we propose an adaptive weighted RPCA to simultaneously preserve low-rank structure and restore the corrupted parts. In our method, the sum of weighted singular values is included in the objective function of minimization. We first estimate the rank of the clean data contained in the observation data by Gerschgorin disks method. Then the weights are adaptively updated by considering some singular values based on the estimated rank, thus both the number and size of the singular values are considered to recover the low-rank matrix with correct information. Experimental results show that the proposed adaptive weighted RPCA algorithm can achieve better performance under various conditions compared to the existing algorithms.**

Keywords—Robust Principal component analysis; nuclear norm minimization; Gerschgorin disks estimation

I. INTRODUCTION

RPCA [1, 2] is a technique to decompose the observation matrix $\boldsymbol{M}$ into low-rank matrix $\boldsymbol{L}$ and sparse one $\boldsymbol{S}$, namely $\boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}$, where both components can be of arbitrary magnitude. More precisely, RPCA can be formulated as the following optimization problem:

$$\arg\min_{\boldsymbol{L},\boldsymbol{S}} rank(\boldsymbol{L}) + \lambda\|\boldsymbol{S}\|_0 \quad s.t. \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}, \qquad (1)$$

where $rank(\cdot)$ denotes the rank function, $||\cdot||_0$ denotes the ℓ_0-norm and λ is the relative weight between the two terms. The above minimization problem is NP-hard in general due to the non-convexity and discontinuous property of the rank function and ℓ_0-norm [3 - 6]. To solve the problem, Candès et al. [2] proved that the original RPCA problem (1) can be effectively solved by convex relaxation as follows:

$$\arg\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \quad s.t. \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}. \qquad (2)$$

where $\|\boldsymbol{L}\|_* = \sum_i \sigma_i(\boldsymbol{L})$ denotes the nuclear norm of matrix $\boldsymbol{L}$ and $\sigma_i(\boldsymbol{L})$ is the i-th singular value of matrix $\boldsymbol{L}$. Problem (2) can be solved by various convex optimization methods [2, 3, 8, 9]. Among these methods, the low rank matrix can be estimated by solving an F-norm based least square sub-problem, i.e., the nuclear norm minimization problem (NNM) [7], which can be easily obtained by the singular value soft-thresholding operator [8, 9]: $\hat{\boldsymbol{L}} = \mathrm{Prox}_\tau(\boldsymbol{L}) = \arg\min_{\boldsymbol{X}} \|\boldsymbol{M} - \boldsymbol{L}\|_F + \tau\|\boldsymbol{L}\|_* = \boldsymbol{U}\boldsymbol{\Sigma}_\tau[\boldsymbol{\Sigma}]\boldsymbol{V}^T$, where $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the singular value decomposition (SVD) of $\boldsymbol{M}$ with $\boldsymbol{\Sigma} = \mathrm{diag}\left(\sigma_{i_{1\le i\le \min(m,n)}}\right)$, $\boldsymbol{\Sigma}_\tau[\boldsymbol{\Sigma}]_{ii} = \mathrm{sgn}(\sigma_i)\cdot\max(|\sigma_i| - \tau, 0)$ denotes the soft-thresholding on $\boldsymbol{\Sigma}$ with parameter τ, which controls the degree of sparsity.

It can be seen that the singular values that exceed τ are inevitably reduced on the same scale. To improve the accuracy of NNM-based low rank estimation, Hu et al. [10] proposed a truncated nuclear norm regularization (TNNR) method where a binary decision is needed to regularize a special singular value. Gu et al. [11, 12] proposed a weighted nuclear norm minimization (WNNM) strategy and studied its minimization. The weighted nuclear norm of matrix $\boldsymbol{L}$ is defined as: $\|\boldsymbol{L}\|_{\boldsymbol{w},*} = \sum_i \omega_i\sigma_i(\boldsymbol{L})$, where $\omega_i > 0$ is a non-negative weight assigned to $\sigma_i(\boldsymbol{L})$. The weight setting scheme is determined by an empirical parameter C. In some practical applications, the rank of $\boldsymbol{L}$ is known, Oh et al. [13, 14] proposed a partial sum of singular values (PSSV) minimization based on the known rank :

$$\arg\min_{\boldsymbol{L},\boldsymbol{S}} \sum_{i=N+1}^{\min(m,n)} \sigma_i(\boldsymbol{L}) + \lambda\|\boldsymbol{S}\|_1 \ s.t. \ \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}$$

where N is the target rank of low-rank matrix, for instance, $N = 1$ for background subtraction, $N = 3$ for photometric stereo. Experimental results show that PSSV method can achieve more accurate solutions compared to the other methods under that the rank is known. However, when the rank of $\boldsymbol{L}$ cannot be known in advance, the work loses its ability to recover the low-rank structure correctly. Hence, in

[1] This work was supported in part by the National Natural Science Foundation of China (51805386 and 61775172).

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on February 07,2023 at 12:19:07 UTC from IEEE Xplore. Restrictions apply.

order to recover the low-rank matrix without the prior knowledge, we should design a new scheme to adaptively shrink singular values of the input matrix.

In this paper, we propose a simple and effective approach to recover low-rank matrix and restore corrupted parts by employing adaptive weighting scheme. First the rank of the low-rank matrix $\boldsymbol{L}$ is estimated by using the Gerschgorin disks estimation method [15]. To further improve the performance of the low-rank matrix recovery, we propose an adaptive weighting strategy based on the estimated rank. The weights are adaptively updated by using some singular values, which contain low-rank structure and the proportion of data information in corresponding component directions. Therefore, we can not only get a robust approximation to the rank, but also recover the low-rank matrix with uncorrupted values faultlessly.

II. Adaptive Weighted Robust PCA

A. Estimation of Rank of Low-Rank Matrix

The rank of low-rank matrix is a very important parameter for the weighted nuclear norm minimization problem such as PSSV. To solve the problem, here the idea of Gerschgorin disks estimation [15] is employed to estimate the rank of low-rank matrix.

For N observation samples with M elements, each observation sample is transformed into a row through the stretching process, and the N samples form a new observation matrix $\boldsymbol{O} \in \boldsymbol{R}^{N\times M}$. If each row of the observation matrix is regarded as a one-dimensional signal $o(t)$ with the snapshot number is M, and N samples are regarded as N signal sensors. In this way, the multiple sample data processing is transformed to array signal processing, and the mutual conversion of data in the time dimension and the space dimension is realized. For a low-rank matrix $\tilde{\boldsymbol{L}}\ (rank(\tilde{\boldsymbol{L}}) = r)$, it can be regarded as an array signal $\tilde{\boldsymbol{L}}(t) = [l_1(t), l_2(t), \cdots, l_N(t)]^T$ that includes r independent signals of M snapshots received by the N signal sensors. The r independent signals can be defined as $\widetilde{\boldsymbol{F}}(t) = [f_1(t), f_2(t), \cdots, f_r(t)]^T$. SVD of the low-rank matrix $\tilde{\boldsymbol{L}}$ is

$$\tilde{\boldsymbol{L}} = \boldsymbol{U}_{\tilde{\boldsymbol{L}}}\boldsymbol{\Sigma}_{\tilde{\boldsymbol{L}}}\boldsymbol{V}_{\tilde{\boldsymbol{L}}}^{\mathrm{T}} = \sum_{i=1}^{r} u_{\tilde{L}i}\sigma_{\tilde{L}i}v_{\tilde{L}i}^{\mathrm{T}} \tag{3}$$

where $\boldsymbol{U}_{\tilde{\boldsymbol{L}}} = (u_{\tilde{L}1}, u_{\tilde{L}2}, \cdots, u_{\tilde{L}r})$ is the left singular value vector matrix of $\tilde{\boldsymbol{L}}$, $\boldsymbol{\Sigma}_{\tilde{\boldsymbol{L}}} = diag(\sigma_{\tilde{L}1}, \sigma_{\tilde{L}2}, \cdots, \sigma_{\tilde{L}r})$ is the singular value matrix that is a diagonal matrix, $\boldsymbol{V}_{\tilde{\boldsymbol{L}}} = (v_{\tilde{L}1}, v_{\tilde{L}2}, \cdots, v_{\tilde{L}r})$ is the right singular value vector matrix of $\tilde{\boldsymbol{L}}$, $u_{\tilde{L}i} \in \boldsymbol{R}^{N\times 1}$, and $v_{\tilde{L}i} \in \boldsymbol{R}^{M\times 1}$.

According to the definition in (3), we can relate $\sigma_{\tilde{L}i}v_{\tilde{L}i}^{\mathrm{T}}$ to correspond to the ith independent signal and $u_{\tilde{L}i}$ to the array manifold of an array of N signal sensors. $v_{\tilde{L}i}$ is independent of each other. Let $\boldsymbol{A}_{\tilde{\boldsymbol{L}}} = \boldsymbol{U}_{\tilde{\boldsymbol{L}}}$, and $\widetilde{\boldsymbol{F}}_{\tilde{\boldsymbol{L}}} = \boldsymbol{\Sigma}_{\tilde{\boldsymbol{L}}}\boldsymbol{V}_{\tilde{\boldsymbol{L}}}^{\mathrm{T}}$, then the observation matrix can be defined as

$$\boldsymbol{M} = \boldsymbol{A}_{\tilde{\boldsymbol{L}}}\widetilde{\boldsymbol{F}}_{\tilde{\boldsymbol{L}}} + \tilde{\boldsymbol{S}}, \tag{4}$$

where noise $\tilde{\boldsymbol{S}}$ is spare matrix. According to [2], it can be seen that the sparse matrix and the low-rank matrix are independent of each other in most cases, and the sparse part in each sample is also independent of each other. Thus, (4) can be equivalent to the array output signal in the array signal processing problem [16] as fellow:

$$\mathbf{y}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{v}(t). \tag{5}$$

In this way, the rank estimation problem of the low rank matrix is transformed into the problem of estimating the number of sources in the array signal processing. The information corresponding to each rank in the low-rank matrix can be equivalent to the information of the signal sent by each source in the array signal processing. The covariance matrix of the observation matrix $\boldsymbol{M}$ is defined as

$$\boldsymbol{R}_{\boldsymbol{M}} = \boldsymbol{M}\boldsymbol{M}^{T}. \tag{6}$$

Eigenvalue decomposition of $\boldsymbol{R}_{\boldsymbol{M}}$

$$\boldsymbol{R}_{\boldsymbol{M}} = \boldsymbol{U}_{\boldsymbol{R}_{\boldsymbol{M}}}\boldsymbol{\Sigma}_{\boldsymbol{R}_{\boldsymbol{M}}}\boldsymbol{U}_{\boldsymbol{R}_{\boldsymbol{B}}}^{H}, \tag{7}$$

where $\boldsymbol{U}_{\boldsymbol{R}_{\boldsymbol{M}}} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_N]$ is the eigenvector matrix, $\boldsymbol{\Sigma}_{\boldsymbol{R}_{\boldsymbol{M}}} = diag(\sigma_1, \sigma_2, \ldots, \sigma_N)$ is the eigenvalue matrix. When there is no noise, the eigenvalues of $\boldsymbol{R}_{\boldsymbol{M}}$ are

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_N = 0. \tag{8}$$

However, in the practical environment, due to the interference of the sparse matrix, the eigenvalues of the covariance matrix $\boldsymbol{R}_{\boldsymbol{M}}$ are

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq \sigma_{r+1} \geq \cdots \geq \sigma_N. \tag{9}$$

In order to more accurately estimate the rank of the low rank matrix, we use Gerschgorin's disk theorem. first the covariance matrix $\boldsymbol{R}_{\boldsymbol{M}}$ is partitioned as

$$\boldsymbol{R}_{\boldsymbol{M}} = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1N} \\ R_{21} & R_{22} & \cdots & R_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ R_{N1} & R_{N2} & \cdots & R_{NN} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{M1}} & \boldsymbol{R} \\ \boldsymbol{R}^{\boldsymbol{H}} & R_{NN} \end{bmatrix}, \tag{10}$$

where $\boldsymbol{R}_{\boldsymbol{M1}}$ is an $(N-1)\times(N-1)$ matrix obtained by deleting the last column and row of $\boldsymbol{R}_{\boldsymbol{M}}$. We define each row of $\boldsymbol{A}_{\tilde{\boldsymbol{L}}}$ in (4) as a vector:

$$\boldsymbol{A}_{\tilde{\boldsymbol{L}}} = [b_1, b_2, \cdots, b_N]^T. \tag{11}$$

It is worth noting that $\boldsymbol{R}$ in (10) can be expressed by

$$\begin{aligned} \boldsymbol{R} &= \left[R_{1N}, R_{2N}, \cdots, R_{(N-1)N}\right]^T \\ &= [\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_{N-1}]^T\boldsymbol{R}_{\boldsymbol{F}}b_N^* = \boldsymbol{A}_1\boldsymbol{R}_{\boldsymbol{F}}b_N^*, \end{aligned} \tag{12}$$

where $\boldsymbol{R}_{\boldsymbol{F}}$=$\widetilde{\boldsymbol{F}}_{\tilde{\boldsymbol{L}}}\widetilde{\boldsymbol{F}}_{\tilde{\boldsymbol{L}}}^{\ T}$ and $\boldsymbol{A}_1 = [b_1, b_2, \cdots, b_{N-1}]^T$. We perform eigenvalue decomposition on the covariance matrix as

$$\boldsymbol{R}_{\boldsymbol{M1}} = \boldsymbol{U}_{\boldsymbol{M1}}\boldsymbol{\Sigma}_{\boldsymbol{1}}\boldsymbol{U}_{\boldsymbol{M1}}^{H}, \tag{13}$$

where $\boldsymbol{U}_{\boldsymbol{M1}}$ is an $(N-1)\times(N-1)$ unitary matrix composed of the eigenvectors of $\boldsymbol{R}_{\boldsymbol{M1}}$ as

$$\boldsymbol{U}_{\boldsymbol{M1}} = [\boldsymbol{q}_{\boldsymbol{1}}', \boldsymbol{q}_{\boldsymbol{2}}', \cdots, \boldsymbol{q}_{N-1}'], \tag{14}$$

and $\boldsymbol{\Sigma}_{\boldsymbol{1}} = diag\{\sigma_1', \sigma_2', \cdots, \sigma_{N-1}'\}$ is a diagonal matrix of eigenvalues of $\boldsymbol{R}_{\boldsymbol{M1}}$. As in (9), the eigenvalues can be expressed as

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on February 07,2023 at 12:19:07 UTC from IEEE Xplore. Restrictions apply.

$$\sigma_1' \geq \sigma_2' \geq \cdots \geq \sigma_r' \geq \sigma_{r+1}' \geq \cdots \geq \sigma_{N-1}' \quad (15)$$

The eigenvalues in (9) and (15) satisfy the interlacing property

$$\sigma_1 \geq \sigma_1' \geq \sigma_2 \geq \sigma_2' \geq \cdots \geq \sigma_r \geq \sigma_r' \geq \sigma_{r+1} \geq \sigma_{r+1}' \geq \cdots \geq \sigma_{N-1} \geq \sigma_{N-1}' \geq \sigma_N \quad (16)$$

Hence, we define an $(N \times N)$ unitary transformed matrix $\boldsymbol{U}$ $(\boldsymbol{U}\boldsymbol{U}^H = \boldsymbol{I})$ as

$$\boldsymbol{U} = \begin{pmatrix} \boldsymbol{U_{M1}} & \boldsymbol{0} \\ \boldsymbol{0}^T & \boldsymbol{1} \end{pmatrix}. \quad (17)$$

The transformed covariance matrix is

$$\boldsymbol{R_T} = \boldsymbol{U}^H \boldsymbol{R_M} \boldsymbol{U} = \begin{pmatrix} \boldsymbol{U_{M1}^H R_{M1} U_{M1}} & \boldsymbol{U_{M1}^H R} \\ \boldsymbol{R^H U_{M1}} & R_{NN} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Sigma_1} & \boldsymbol{U_{M1}^H R} \\ \boldsymbol{R^H U_{M1}} & R_{NN} \end{pmatrix} = \begin{pmatrix} \sigma_1' & 0 & 0 & \cdots & 0 & \rho_1 \\ 0 & \sigma_2' & 0 & \cdots & 0 & \rho_2 \\ 0 & 0 & \sigma_3' & \cdots & 0 & \rho_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{N-1}' & \rho_{N-1} \\ \rho_1^* & \rho_2^* & \rho_3^* & \cdots & \rho_{N-1}^* & R_{NN} \end{pmatrix} \quad (18)$$

where

$$\rho_i = \boldsymbol{q_i'}^H \boldsymbol{R} = \boldsymbol{q_i'}^H \boldsymbol{A_1 R_F} b_N^*, \; i = 1,2,\cdots,N-1. \quad (19)$$

The eigenvalues of $\boldsymbol{R_T}$ can be estimated by Gerschgorin's disk theorem [15]. Obviously, the radii of the first $(N-1)$ Gaussian disks can be expressed as

$$r_i = |\rho_i| = \left|\boldsymbol{q_i'}^H \boldsymbol{A_1 R_F} b_N^*\right| = \left|\boldsymbol{q_i'}^H \boldsymbol{R}\right|, \; i = 1,2,\cdots,N-1. \quad (20)$$

By using Cauchy-Schwartz inequality, we can obtain that

$$r_i = |\rho_i| = \left|\boldsymbol{q_i'}^H \boldsymbol{A_1 R_F} b_N^*\right| \leq \left|\boldsymbol{q_i'}^H \boldsymbol{A_1}\right| \cdot |\boldsymbol{R_F} b_N^*| = \gamma \left|\boldsymbol{q_i'}^H \boldsymbol{A_1}\right|, \quad (21)$$

where $\gamma = |\boldsymbol{R_F} b_N^*|$ is independent of i. Then the radius r_i of the ith Gerschgorin's disk actually depends on the size of $\boldsymbol{q_i'}^H \boldsymbol{A_1}$.

If $\boldsymbol{q_i'}$ is the eigenvector of noise, the radius of the ith Gerschgorin's disk of noise will be significantly reduced. If $\boldsymbol{q_i'}$ is the eigenvector of the low-rank part, the radius of the ith Gerschgorin's disk of the low-rank part will not be zero. Therefore, we can estimate the rank by the heuristic decision rule as

$$\boldsymbol{GDE}(k) = r_k - \frac{D(M)}{N-1} \sum_{i=1}^{N-1} r_i, \quad (22)$$

where $k = 1,2,\cdots,N-2$, and the adjustment factor $D(M)$ (between 0 to 1) is a constant related to M. In this paper, we define $D(M) = 2.3 \,/\, log(M)$. We can calculate $\boldsymbol{GDE}(k)$ from $k = 1$. When $\boldsymbol{GDE}(k)$ is negative for the first time, the rank of the low-rank matrix is $r = k - 1$. Therefore the estimated rank can be used to recover the low-rank matrix accurately in our proposed adaptive RPCA in the next subsections.

B. Weighted RPCA

In this subsection, the weighted form for RPCA problem (2) is considered, which can be formulated as the following optimization problem:

$$\arg\min_{\boldsymbol{L},\boldsymbol{S}} \; ||\boldsymbol{L}||_{\boldsymbol{W}} + \lambda \|\boldsymbol{S}\|_1 \; s.t. \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}, \quad (23)$$

The optimization problem (23) can be efficiently solved via the alternating direction minimization technique. First the augmented Lagrangian function of (23) can be written as:

$$\mathcal{L}(\boldsymbol{L},\boldsymbol{S},\boldsymbol{Y}) = \|\boldsymbol{L}\|_{\boldsymbol{W}} + \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{Y}, \boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S} \rangle + \tfrac{\mu}{2}\|\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S}\|_F^2 \quad (24)$$

where $\langle \cdot,\cdot \rangle$ represents matrix inner product, μ is a positive penalty scalar, and $\boldsymbol{Y}$ is the Lagrangian multiplier. There are three variables $\boldsymbol{L}$, $\boldsymbol{S}$, and $\boldsymbol{Y}$. Then we can iteratively update one variable at a time by fixing the others, and each step has a simple closed-form solution. Thus the optimization problem can be divided into the following three sub-problems.

$\boldsymbol{S}$ *sub-problems*: While both $\boldsymbol{L}$, and $\boldsymbol{Y}$ are fixed, (24) can be reduced to the following optimization problem:

$$\boldsymbol{S}^* = \arg\min_{\boldsymbol{S}} \; \lambda\|\boldsymbol{S}\|_1 + \langle \boldsymbol{Y}, \boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S} \rangle + \tfrac{\mu}{2}\|\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S}\|_F^2 = \arg\min_{\boldsymbol{S}} \; \tfrac{1}{2}\|\boldsymbol{S} - (\boldsymbol{M} - \boldsymbol{L} + \mu^{-1}\boldsymbol{Y})\|_F^2 + \tfrac{\lambda}{\mu}\|\boldsymbol{S}\|_1 \quad (25)$$

$\boldsymbol{L}$ *sub-problems*: Given $\boldsymbol{S}$, and $\boldsymbol{Y}$, (24) leads to the following optimization problem:

$$\boldsymbol{L}^* = \arg\min_{\boldsymbol{L}} \; \|\boldsymbol{L}\|_{\boldsymbol{W}} + \langle \boldsymbol{Y}, \boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S} \rangle + \tfrac{\mu}{2}\|\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S}\|_F^2 = \arg\min_{\boldsymbol{L}} \; \tfrac{1}{2}\|\boldsymbol{L} - (\boldsymbol{M} - \boldsymbol{S} + \mu^{-1}\boldsymbol{Y})\|_F^2 + \tfrac{1}{\mu}\|\boldsymbol{L}\|_{\boldsymbol{W}} \quad (26)$$

$\boldsymbol{Y}$ *sub-problems*: $\boldsymbol{Y}$ can be updated by

$$\boldsymbol{Y}_{k+1} = \boldsymbol{Y}_k + \mu(\boldsymbol{M} - \boldsymbol{L}_{k+1} - \boldsymbol{S}_{k+1}) \quad (27)$$

C. Adaptive Weighting Mechanism

In this subsection, an adaptive choice of weights $\boldsymbol{W} = \mathrm{diag}\left(\{\omega_i\}_{1 \leq i \leq \min(m,n)}\right)$ in (23) is introduced, which leads to a new adaptive RPCA. For the sake of high quality of image and signal recovery, each singular value that exceeds the threshold should be adaptively shrunk. For the larger singular values which quantify the principal directions of principal component, they should be reduced a little to preserve the low-rank part as much as possible. For smaller singular values, they should be kept a bit to preserve the low-rank data

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on February 07,2023 at 12:19:07 UTC from IEEE Xplore. Restrictions apply.

information conversely. It is noted that in the shrinkage form of the weighted nuclear norm, $\boldsymbol{\Sigma}_{ii} - \tau \boldsymbol{W}_{ii}$ can be rewritten

$$\boldsymbol{\Sigma}_{ii} - \tau \boldsymbol{W}_{ii} = \boldsymbol{\Sigma}_{ii} - \tau + (1 - \boldsymbol{W}_{ii})\tau \stackrel{\text{def}}{=} \boldsymbol{\Sigma}_{ii} - \tau + k_i \tau.$$

Clearly, the term $\boldsymbol{\Sigma}_{ii} - \tau$ is the soft-thresholding operation of nuclear norm without weighting, $k_i \tau = (1 - \boldsymbol{W}_{ii})\tau$ can be regarded as compensation item in the weighted nuclear norm minimization.

Therefore, it is very important to determine the coefficient k_i in adaptive RPCA model. Before each singular value is shrunk, the proportion of data information of low-rank component for each singular value will never exceed but be very close to the ratio of information of the corresponding singular value in $\boldsymbol{M}_r$, where $\boldsymbol{M}_r$ is the approximate of data matrix $\boldsymbol{M}$ using the first r largest singular values. The parameter r is the rank of the low-rank matrix. Once r is estimated by (22), we can determine the coefficient k_i as fellow: first reconstruct matrix $\boldsymbol{M}_i$ by using the ith singular value, then project $\boldsymbol{M}_i$ to $\boldsymbol{M}_r$, finally assign the projected coefficient to k_i. That is,

$$k_i = \frac{\langle \boldsymbol{M}_i, \boldsymbol{M}_r \rangle}{||\boldsymbol{M}_r||_F^2} = \frac{z_i^2}{\sum_{i=1}^{r} z_i^2}. \tag{28}$$

In the sequel, we have:

$$\boldsymbol{W}_{ii} = \begin{cases} 1 - \frac{z_i^2}{\sum_{i=1}^{r} z_i^2}, & \text{if } i \le r \\ 1, & \text{otherwise} \end{cases} \tag{29}$$

It follows from (28) that $0 < \boldsymbol{W}_{ii} \le \boldsymbol{W}_{i+1\,i+1} \le 1$.

For clarity, the entire procedure to solve problem (23) is summarized in Algorithm 1.

Algorithm 1 Adaptive RPCA

Input: $\boldsymbol{M} \in \mathbb{R}^{m \times n}, \lambda = 1/\sqrt{\max(m,n)}$;

1: **Initialization:** $\boldsymbol{S}_0 = \boldsymbol{Y}_0 = \boldsymbol{0} \in \mathbb{R}^{m \times n}$, r is estimated by (22), $\mu_0 = 1/Z_r$, and $\boldsymbol{W}$ is initialized by (29);
2: **while** not converged **do**
3: compute $\boldsymbol{L}_{k+1} = \arg\min_{\boldsymbol{L}} \frac{1}{2}\|\boldsymbol{L}_k - (\boldsymbol{M} - \boldsymbol{S}_k + \mu^{-1}\boldsymbol{Y}_k)\|_F^2 + \frac{1}{\mu}\|\boldsymbol{L}_k\|_{\boldsymbol{W}}$;
4: compute $\boldsymbol{S}_{k+1} = \arg\min_{\boldsymbol{S}} \frac{1}{2}\|\boldsymbol{S}_k - (\boldsymbol{M} - \boldsymbol{L}_{k+1} + \mu^{-1}\boldsymbol{Y}_k)\|_F^2 + \frac{\lambda}{\mu}\|\boldsymbol{S}_k\|_1$;
5: compute $\boldsymbol{Y}_{k+1} = \mu(\boldsymbol{M} - \boldsymbol{L}_{k+1} - \boldsymbol{S}_{k+1})$;
6: update $\mu_{k+1} = 1.05 \times \mu_k$;
7: **end while**
8: **output:** $\boldsymbol{L}, \boldsymbol{S}$.

In Algorithm 1, the coefficient λ is set to be $\lambda = 1/\sqrt{\max(m,n)}$, which is the default parameter recommended in RPCA. The iteration is terminated when $||\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S}||_F \le 10^{-6}||\boldsymbol{M}||_F$.

The sequences $\{\boldsymbol{L}_k\}$ and $\{\boldsymbol{S}_k\}$ generated by Algorithm 1 satisfy:

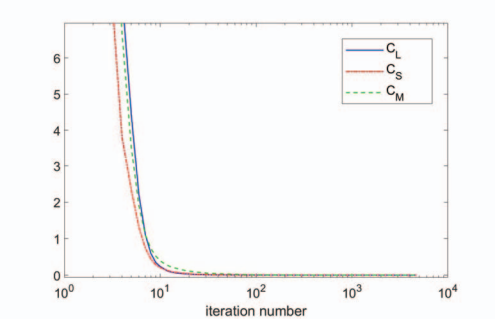


Fig. 1. The value of C_L, C_S and C_M with the number of iterations. Assuming that $\boldsymbol{M} \in \boldsymbol{R}^{m \times n}$, $m = 10000$, $n = 20$, $rank(\boldsymbol{L}) = 3$, and the corrupted rate $p = 0.05$.

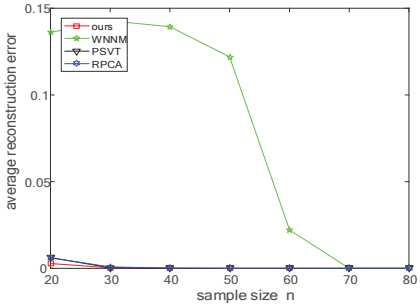


Fig. 2. Reconstruction errors of different algorithms under different sample numbers.

$$C_{\boldsymbol{L}} = \lim_{k \to \infty} \|\boldsymbol{L}_{k+1} - \boldsymbol{L}_k\|_F = 0$$

$$C_{\boldsymbol{S}} = \lim_{k \to \infty} \|\boldsymbol{S}_{k+1} - \boldsymbol{S}_k\|_F = 0$$

$$C_{\boldsymbol{M}} = \lim_{k \to \infty} \|\boldsymbol{M} - \boldsymbol{L}_{k+1} - \boldsymbol{S}_{k+1}\|_F = 0$$

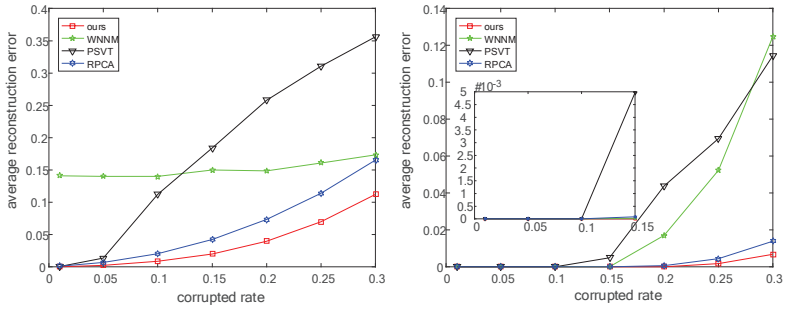
These claims can be proved by the experimental results shown in Fig. 1 and the theoretical analysis will be our future work.

III. Experimental Results

In this section, the performance of the proposed adaptive RPCA method is compared with classical RPCA [2] and two state-of-the-art methods (PSVT [14], WNNM-RPCA [12]) on numerical experiments. We illustrate that our method can recover matrices with various rank from errors with various density. All the codes are written in Matlab and all the experiments are conducted on a PC running Windows 10 with 16G RAM and 3.4GHz CPU.

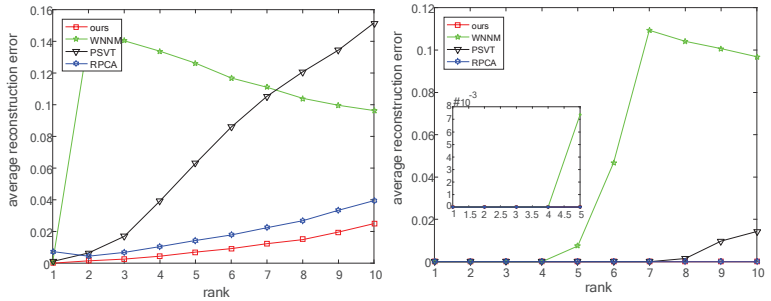
Reconstruction errors are compared among RPCA, PSVT, WNNM-RPCA and our adaptive RPCA methods under different problem settings such as different sample sizes (n), different matrix ranks (r_0), and different corruption ratios (p). A matrix $\boldsymbol{D} \in \mathbb{R}^{m \times n}$ with rank-r_0 is generated by sampling two matrices, $\boldsymbol{P} \in \mathbb{R}^{m \times r_0}$ and $\boldsymbol{Q} \in \mathbb{R}^{r_0 \times n}$ with entries belonging to uniform distribution $U[0, 1]$, namely, $\boldsymbol{D} = \boldsymbol{PQ}$. The corrupted indices Ω are uniformly at random sampled over each column. And let p be the proportion of corrupted entries over each column. And we generate synthetic data as follows:

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on February 07,2023 at 12:19:07 UTC from IEEE Xplore. Restrictions apply.



n=20 n=80

Fig. 3. Reconstruction error versus the corrupted rates. These results show that our proposed algorithms not only can achieve much lower reconstruction error, but also have a stronger adaptability to different situations.



n=20 n=80

Fig. 4. Reconstruction error versus the matrix rank (r_0). These results show that our proposed adaptive RPCA algorithm can achieve much lower reconstruction error.

$$\boldsymbol{D}_{ij} = \boldsymbol{G}_{ij}, \quad (i,j) \in \Omega,$$

where $\boldsymbol{G}$ has entries i.i.d. $\mathcal{N}(0.5,1)$. The reconstruction error is defined as $||\boldsymbol{D}_{sol} - \boldsymbol{D}||_F / ||\boldsymbol{D}||_F$, where $\boldsymbol{D}_{sol}$ denotes the solution in a certain algorithm. We run each of the tests over 30 trials and give the average errors of overall trials.

The matrix size is set as $m = 10000$, the rank is r_0=3 and the corruption ratio is chosen to be $p = 0.05$. We examine the effect of different sample sizes on reconstruction errors. The average errors are shown in Fig. 2. Compared with the other three algorithms, WNNM-RPCA obviously needs a relatively larger number of samples.

To analyze the effect of different corruption ratios on reconstruction errors, we set m=10000 and r_0=3. The average errors are shown in Fig. 3 for sample size n =20 and 80, respectively. We set m=10000 and corrupted rate $p = 0.05$, the effects of different ranks on reconstruction errors are shown in Fig. 4.

From Fig. 3 and Fig. 4, we can observe that under the ideal conditions with larger sample size, smaller corruption rate and lower rank, all algorithms have perfect performance where their reconstruction errors are smaller than 10^{-3}. However, once one of the conditions changes a little, the performance of two state-of-the-art methods PSVT and WNNM-RPCA drops sharply. For example, the algorithm WNNM-RPCA cannot work at all when the number of samples is insufficient.

The CPU time of different algorithms under 20 samples and different corrupted rates is shown in Table 1. The reconstruction errors of different algorithms under different sample numbers and different corrupted rates are shown in Table 2.

It can be seen from Table1 and Table 2 that PSVT is faster and more accurate than other methods in the case of small samples. PSVT utilizes a priori target rank information, i.e., the exact rank of low-rank matrix is known. Hence it is reasonable that PSSV outperforms other methods under known rank and limited observations. When the rank of the meaningful low-rank matrix is unknown, the time consumed by our proposed algorithm and WNNM algorithm is almost same. But as shown in Table 2, the reconstruction error of our proposed method is slightly better than PSVT and significantly better than WNNM. Through these numerical experiments, the proposed method eliminates the rank priori constraints of the low-rank matrix while ensuring the accuracy of the low-rank matrix recovery.

IV. Conclusions

This paper presents an adaptive weighted RPCA algorithm to preserve low-rank structure and restore the corrupted parts simultaneously. The weights of nuclear norm terms are adaptively determined by some singular values of the input data matrix, which is able to recover the low-rank matrix with correct information. These singular values are selected according to the estimated rank of low-rank matrix. The experimental results have revealed the efficiency of our proposed method. It can be expected that the adaptive weighted RPCA will have more successful applications under different scenes such as non-uniform illumination images, differently exposed images and different illumination images with deficient samples.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on February 07,2023 at 12:19:07 UTC from IEEE Xplore. Restrictions apply.

TABLE □: CPU TIME (SECONDS) OF DIFFERENT ALGORITHMS UNDER DIFFERENT SAMPLE NUMBERS AND DIFFERENT CORRUPTED RATES

rank	3					5					8				
corrupted ratio	0.05	0.10	0.15	0.20	0.25	0.05	0.10	0.15	0.20	0.25	0.05	0.10	0.15	0.20	0.25
RPCA	7.39	14.28	15.12	14.26	14.4	14.9	15.2	14.1	14.2	14.1	14.9	15.1	14.9	8.7	6.93
WNNM	2.76	2.413	2.453	2.43	2.43	2.91	2.66	2.68	2.52	2.51	2.71	2.61	2.60	2.60	2.54
PSVT	0.41	0.470	0.47	0.49	0.51	0.51	0.51	0.53	0.51	0.49	0.58	0.52	0.51	0.49	0.48
ours	2.85	2.62	2.89	2.52	2.65	3.21	2.9	2.97	2.77	2.93	3.27	3.08	3.11	2.91	2.99

TABLE □: THE RECONSTRUCTION ERRORS OF DIFFERENT ALGORITHMS UNDER DIFFERENT SAMPLE NUMBERS AND DIFFERENT CORRUPTED RATES

rank	3					5					8				
corrupted ratio	0.05	0.10	0.15	0.20	0.25	0.05	0.10	0.15	0.20	0.25	0.05	0.10	0.15	0.20	0.25
RPCA	0.004	0.018	0.028	0.054	0.078	0.013	0.026	0.055	0.079	0.119	0.027	0.049	0.074	0.103	0.144
WNNM	0.157	0.135	0.151	0.156	0.138	0.135	0.115	0.129	0.136	0.149	0.115	0.1	0.11	0.116	0.121
PSVT	0.001	0.009	0.013	0.029	0.198	0.008	0.125	0.149	0.227	0.268	0.096	0.188	0.24	0.283	0.36
ours	0.002	0.007	0.012	0.024	0.031	0.006	0.013	0.029	0.038	0.061	0.015	0.029	0.043	0.061	0.078

REFERENCES

[1] I. Jolliffe, "Principal Component Analysis," Springer, 2005.

[2] E. J. Candès, X. Li, Y. Ma, and J. Wright. "Robust principal component analysis?," Journal of the ACM, vol. 58, no. 3, 2011, pp. 1-37, 2011.

[3] V. Namrata, B. Thierry, J. Sajid, and N. Praneeth. "Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery," IEEE Signal Processing Magazine, vol. 35, no. 4, pp. 32-55, 2018.

[4] B. Thierry, J. Sajid, Z. Hongyang, L. Zhouchen, and O. Ricardo. "On the applications of robust PCA in image and video processing," Proceedings of the IEEE, vol. 106, no. 8, pp. 1427-1457, 2018.

[5] B. Recht, M. Fazel, and P.A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," SIAM review, vol. 52, no. 3, pp. 471-501, 2010.

[6] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization," Mathematical Programming, vol. 128, pp. 321-353, 2011.

[7] C. Li. "A new approximation of the matrix rank function and its application to matrix rank minimization," Journal of Optimization Theory & Applications, vol. 163, no. 2, pp. 569-594, 2014.

[8] K. C. Toh and S. Yun. "An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems," Pacific Journal of Optimization, vol. 6, no. 3, pp. 615-640, 2010.

[9] J.F. Cai, E.J. Candès and Z. Shen. "A singular value thresholding algorithm for matrix completeon," SIAM Journal on Optimization, vol. 20, no. 4, pp.1956-1982, 2010.

[10] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He. "Fast and accurate matrix completion via truncated nuclear norm regularization," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 9, pp.2117–2130, 2013.

[11] S. Gu, L. Zhang, W. Zuo and X. Feng. "Weighted nuclear nolrm minimization with application to image denoising," IEEE Conference on Computer Vision and Pattern Recognition, pp. 2862-2869, 2014.

[12] S. Gu, Q. Xie, D. Meng, W. Zuo and X. Feng. "Weighted nuclear norm minimization and its applications to low level vision," International Journal of Computer Vision, vol. 121, no. 2, pp. 183-208, 2017.

[13] T. H. Oh, Y. W. Tai, J. C. Bazin and H. Kim. "Partial sum minimization of singular values in robust PCA: algorithm and applications," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 4, pp. 744-758, 2016.

[14] T. H. Oh, H. K, Y.W. Tai, J. C. Bazin and I. C. Kweon. "Partial sum minimization of singular values in RPCA for low-level vision," The IEEE International Conference on Computer Vision, pp. 145-152, 2013.

[15] C. Shao, W. Li, Y. Guo, J. Zhang, and J. Shen. "Source number estimation with Gerschgorin radii by reduced-rank covariance matrix," International Conference on Information & Communications Technologies, 2014.

[16] P. Gupta and M. Agrawal. "Design and analysis of the sparse array for DOA estimation of noncircular signals,: IEEE Transactions on Signal Processing, vol. 67, no. 2, pp. 460-473, 2019.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on February 07,2023 at 12:19:07 UTC from IEEE Xplore. Restrictions apply.