

Online Reweighted Least Squares Robust PCA

Athanasios A. Rontogiannis[✉], Paris V. Giampouras[✉], and Konstantinos D. Koutroumbas

Abstract—The letter deals with the problem known as robust principal component analysis (RPCA), that is, the decomposition of a data matrix as the sum of a low-rank matrix component and a sparse matrix component. After expressing the low-rank matrix component in factorized form, we develop a novel online RPCA algorithm that is based entirely on reweighted least squares recursions and is appropriate for sequential data processing. The proposed algorithm is fast, memory optimal and, as corroborated by indicative empirical results on simulated data and a video processing application, competitive to the state-of-the-art in terms of estimation performance.

Index Terms—Matrix factorization, online processing, low-rank, sparse, iteratively reweighted least squares, alternating minimization, robust PCA.

I. INTRODUCTION

SINCE the seminal work of Candes *et al.* [1], robust principal component analysis (RPCA) has attracted considerable attention both in research and in applications. Following the principal component pursuit (PCP) [1], a plethora of new algorithms have been developed [2] and applied in diverse fields such as low-level image processing [3], [4], medical imaging [5], [6], hyperspectral imaging [7], [8], video processing and analysis [9], [10], computer vision [11], [12] to name but a few [13], [14].

It is well known that in RPCA the data signal is commonly decomposed as the sum of a low-rank term, a sparse term and a statistically modeled noise term, i.e.,

$$\mathbf{Y} = \mathbf{L} + \mathbf{S} + \mathbf{E}, \quad (1)$$

with $\mathbf{Y}, \mathbf{L}, \mathbf{S}, \mathbf{E} \in \mathbb{R}^{m \times n}$. The columns of matrix \mathbf{L} span a low-dimensional subspace and only a few of the entries of \mathbf{S} have non-negligible values. The entries of \mathbf{E} are usually assumed to be zero-mean Gaussian. The classical approach to obtain both \mathbf{L} and \mathbf{S} from \mathbf{Y} is by solving a minimization problem, whose objective function consists of a data proximity term (based on the statistics of \mathbf{E}) and two regularization terms that promote

low-rankness and sparsity to \mathbf{L} and \mathbf{S} respectively. In most cases, the data matrix \mathbf{Y} is processed as a whole, leading to batch iterative decomposition algorithms.

Nevertheless, in modern big data applications, data sequences may be long or data may arrive in a streaming fashion, as, e.g., in video processing applications. Apparently, in such cases a batch processing algorithm would be prone to fail, due to high computational complexity, latency and large memory demands. It should be noted that, although batch solvers for RPCA have been extensively investigated (see [2] and the references therein), the studies on online RPCA schemes are rather limited. In this regard, the algorithms presented in [15] and [16] (which, among others, come with complete correctness guarantees [2]) can be considered as the state-of-the-art. Other online RPCA schemes, such as those presented in [17], [18] and [19], albeit fast and memory efficient, show mediocre to poor performance [2], [16].

Capitalizing on our previous work on low-rank matrix factorization [20], [21], we present a new online RPCA algorithm that is based entirely on *reweighted least squares (LS)* time recursions. In a batch setting, iteratively reweighted least squares (IRLS) has been successfully utilized in [22] to promote sparsity and extended in [20] for low-rank matrix factorization. In the present letter, we suitably combine these two schemes in a unique RPCA solver. More specifically, after expressing the matrix \mathbf{L} as the product of two low-rank matrices, we define a new objective function, in which the matrix factors of \mathbf{L} are penalized via their reweighted Frobenius norms as in [20] and the sparse component \mathbf{S} is penalized using reweighted LS as in [22]. The respective minimization problem is then solved via alternating IRLS, leading to a batch RPCA algorithm that comprises efficient matrix-matrix operations only. Moving one step further, we reformulate the initial objective function described above by adjusting it to the online scenario that allows the processing of data on a matrix column-by-matrix column basis. The solution of this minimization problem gives rise to an online reweighted LS RPCA algorithm, in which all sought quantities are computed time-recursively. The proposed algorithm is very fast and turns out to be *memory optimal*, i.e., has the minimum possible memory requirements. Indicative empirical results verify the high computational efficiency of the new algorithm and demonstrate that it may be competitive, in terms of estimation performance, to the state-of-the-art NORST algorithm [15], on both simulated data and real video foreground-background separation tasks.

Notations: Matrices are denoted by bold capital letters, e.g., \mathbf{X} . We adopt Matlab notation for matrix entries, matrix rows and matrix columns, i.e., $\mathbf{X}(i, j)$ is the ij th element of \mathbf{X} and $\mathbf{X}(i, :)$, $\mathbf{X}(:, j)$ are the i th row and j th column of \mathbf{X} respectively. The symbols \odot and \oslash are used for entry-wise multiplication and division respectively, transposition is denoted by \cdot^T , $\|\cdot\|_2$, $\|\cdot\|_F$ denote the ℓ_2 and the Frobenius norms, $\sigma_i(\mathbf{X})$ is the i th singular value of matrix \mathbf{X} and $\mathbf{1}$ is the all ones vector or matrix depending on the context. Finally, \mathbb{R}_+ is the set of non-negative real numbers.

Manuscript received May 15, 2020; revised July 13, 2020; accepted July 18, 2020. Date of publication July 27, 2020; date of current version August 12, 2020. The work of P. V. Giampouras was supported by the European Union under the Horizon 2020 Marie-Sklodowska-Curie Global Fellowship Program: HyPPOCRATES-H2020-MSCA-IF-2018, under Grant 844290. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yongwei Huang. (Corresponding author: Paris Giampouras.)

Athanasios A. Rontogiannis and Konstantinos D. Koutroumbas are with the Institute for Astronomy, Astrophysics, Space Applications and Remote Sensing of the National Observatory of Athens, 15236 Penteli, Greece (e-mail: tronto@noa.gr; koutroum@space.noa.gr).

Paris V. Giampouras is with the Institute for Astronomy, Astrophysics, Space Applications and Remote Sensing of the National Observatory of Athens, 15236 Penteli, Greece, and also with the Mathematical Institute for Data Science, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: parisg@jhu.edu).

Digital Object Identifier 10.1109/LSP.2020.3011896

II. PROBLEM FORMULATION AND BATCH IRLS RPCA

Using a matrix factorization (MF) representation for the low-rank component \mathbf{L} , i.e., $\mathbf{L} = \mathbf{U}\mathbf{V}^T$, (1) is rewritten as

$$\mathbf{Y} = \mathbf{U}\mathbf{V}^T + \mathbf{S} + \mathbf{E}, \quad (2)$$

where $\mathbf{U} \in \mathbb{R}^{m \times d}$, $\mathbf{V} \in \mathbb{R}^{n \times d}$ and $d \ll m, n$. Note that due to the last inequality, MF implicitly imposes low rank to \mathbf{L} , although d may still be much greater than its true rank. In (2), the columns of \mathbf{U} span a low dimensional subspace where the columns of \mathbf{L} lie and \mathbf{V} contains the representation coefficients in this subspace. The estimation of $\mathbf{U}, \mathbf{V}, \mathbf{S}$ given \mathbf{Y} is an ill-posed problem that can only be tackled by imposing certain constraints to the sought matrix variables. In that regard, we propose the following minimization problem, which falls entirely into the IRLS framework

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{S}} \frac{1}{2} \|\mathbf{Y} - \mathbf{U}\mathbf{V}^T - \mathbf{S}\|_{\text{F}}^2 + \lambda (\|\mathbf{U}\mathbf{D}^{\frac{1}{2}}\|_{\text{F}}^2 + \|\mathbf{V}\mathbf{D}^{\frac{1}{2}}\|_{\text{F}}^2) + \mu \|\mathbf{S} \odot \mathbf{W}^{\frac{1}{2}}\|_{\text{F}}^2, \quad (3)$$

where λ, μ are regularization parameters. The weighting matrix \mathbf{D} is a diagonal matrix defined as [20]

$$\mathbf{D} = \text{diag} \left((\|\mathbf{U}(:, 1)\|_2^2 + \|\mathbf{V}(:, 1)\|_2^2 + \epsilon)^{-\frac{1}{2}}, \dots, (\|\mathbf{U}(:, d)\|_2^2 + \|\mathbf{V}(:, d)\|_2^2 + \epsilon)^{-\frac{1}{2}} \right), \quad (4)$$

while the ij th weight element of \mathbf{W} is expressed as

$$\mathbf{W}(i, j) = (\mathbf{S}(i, j)^2 + \epsilon)^{-\frac{1}{2}}, \quad (5)$$

where ϵ is a very small positive constant that provides protection against singularities. The first term in the objective function of (3) is a model fitting term, the second term promotes the low-rankness of $\mathbf{L} = \mathbf{U}\mathbf{V}^T$ and the third term induces sparsity on \mathbf{S} . While the latter is very well known from the sparse vector recovery literature [22], the low-rank imposition properties of the second term have not been established yet in a rigorous way. In this vein, an enlightening result is presented and proved next [23].

Proposition 1: If $\|\mathbf{L}\|_{\mathcal{S}_p} = (\sum_{i=1}^d (\sigma_i(\mathbf{L}))^p)^{1/p}$ is the Schatten- p norm of a low-rank matrix $\mathbf{L} = \mathbf{U}\mathbf{V}^T$ with $0 < p < 1$, the following holds

$$\begin{aligned} \|\mathbf{L}\|_{\mathcal{S}_p}^p &= \min_{\mathbf{U}, \mathbf{V}, \mathbf{L}=\mathbf{U}\mathbf{V}^T} \sum_{i=1}^d (\|\mathbf{U}(:, i)\|_2 \|\mathbf{V}(:, i)\|_2)^p \\ &= \min_{\mathbf{U}, \mathbf{V}, \mathbf{L}=\mathbf{U}\mathbf{V}^T} \frac{1}{2^p} \sum_{i=1}^d (\|\mathbf{U}(:, i)\|_2^2 + \|\mathbf{V}(:, i)\|_2^2)^p. \end{aligned} \quad (6)$$

Proof: The function $f(x) = x^p$ with $0 < p < 1$ is concave, non-decreasing on $[0, \infty)$ and $f(0) = 0$. Then by defining

$$R(\mathbf{L}) \triangleq \sum_{i=1}^d f(\sigma_i(\mathbf{L})) = \sum_{i=1}^d (\sigma_i(\mathbf{L}))^p = \|\mathbf{L}\|_{\mathcal{S}_p}^p,$$

from Theorem 1 and Corollary 1 of [24] we get directly (6). ■

Notice that (6) generalizes the expressions for the variational form of the nuclear norm [25], which are obtained by setting $p = 1$ in (6) (recall that the nuclear norm is the Schatten-1 norm). While the nuclear norm is equivalent to the ℓ_1 norm on the

Algorithm 1: Batch IRLS RPCA Algorithm.

Input: $\mathbf{Y}, \lambda > 0, \mu > 0$

Initialize: $k = 0, \mathbf{U}_0, \mathbf{V}_0, \mathbf{S}_0, \mathbf{D}_0$

repeat

 Compute \mathbf{W}_{k+1} from (5) based on \mathbf{S}_k

$\mathbf{S}_{k+1} \leftarrow (\mathbf{Y} - \mathbf{U}_k \mathbf{V}_k^T) \odot (\mathbf{I} + \mu \mathbf{W}_{k+1})$

$\mathbf{V}_{k+1} \leftarrow (\mathbf{Y} - \mathbf{S}_{k+1})^T \mathbf{U}_k (\mathbf{U}_k^T \mathbf{U}_k + \lambda \mathbf{D}_k)^{-1}$

$\mathbf{U}_{k+1} \leftarrow (\mathbf{Y} - \mathbf{S}_{k+1}) \mathbf{V}_{k+1} (\mathbf{V}_{k+1}^T \mathbf{V}_{k+1} + \lambda \mathbf{D}_k)^{-1}$

 Compute \mathbf{D}_{k+1} from (4) based on $\mathbf{U}_{k+1}, \mathbf{V}_{k+1}$

$k = k + 1$

until convergence

Output: $\hat{\mathbf{U}} = \mathbf{U}_k, \hat{\mathbf{V}} = \mathbf{V}_k, \hat{\mathbf{S}} = \mathbf{S}_k$

singular values of \mathbf{L} , the Schatten- p norm is equivalent to the ℓ_p norm with $0 < p < 1$ and thus is more effective in imposing low rank. If, now, ϵ is neglected in (4) we easily get

$$\|\mathbf{U}\mathbf{D}^{\frac{1}{2}}\|_{\text{F}}^2 + \|\mathbf{V}\mathbf{D}^{\frac{1}{2}}\|_{\text{F}}^2 = \sum_{i=1}^d (\|\mathbf{U}(:, i)\|_2^2 + \|\mathbf{V}(:, i)\|_2^2)^{\frac{1}{2}} \quad (7)$$

From (6) and (7) we see that the second term in (3) is an upper bound of the Schatten-1/2 norm and thus is expected to impose low rank to \mathbf{L} , as desired.¹ An alternative interpretation is given in [20] by recognizing (7) as the $\ell_{1,2}$ of the matrix $\begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix}$. Due to the column sparsity promoting property of the $\ell_{1,2}$ norm [27], low rank may be enforced on \mathbf{L} via the joint elimination of corresponding columns of \mathbf{U} and \mathbf{V} [20].

In our problem, both sparsity and low-rankness may be imposed via IRLS with the aid of the weighting matrices \mathbf{D} and \mathbf{W} . In this regard, by minimizing the respective objective function in (3) alternately with respect to \mathbf{U}, \mathbf{V} and \mathbf{S} (and assuming that \mathbf{D} and \mathbf{W} depend on the estimates of \mathbf{U}, \mathbf{V} and \mathbf{S} of the previous iteration k) we end up with the batch IRLS RPCA algorithm tabulated as Algorithm 1. In analogy to our previous work [20], Algorithm 1 can be also derived following a block successive upper bound minimization (BSUM) procedure [28] with blocks the matrices \mathbf{U}, \mathbf{V} and \mathbf{S} . The upper bound functions minimized for each block satisfy the conditions of BSUM and minimization leads in all three cases to closed-form solutions. Thus, according to Theorem 1 of [28], the resulting algorithm converges to stationary points of the objective function of (3).

III. ONLINE REWEIGHTED LS RPCA ALGORITHM

As mentioned previously, in many applications (e.g., online video processing) data may become available sequentially on a frame-by-frame or block-by-block basis. In our setting, this means that we might need to process the columns of the data matrix \mathbf{Y} sequentially in an online fashion. In such an online scenario, given $\mathbf{Y}(:, n)$ at time n , we must be able to estimate $\mathbf{S}(:, n)$, update the fixed size $(m \times d)$ subspace matrix estimate \mathbf{U}_{n-1} to \mathbf{U}_n and compute $\mathbf{V}(n, :)$, the new row of \mathbf{V} . In addition, a “forgetting” mechanism should be included to account for possible time-variations of the data subspace. Based on the

¹In [26] matrix factorization based variational forms of the Schatten- p quasi-norm have been used for batch RPCA. In contrast to the new ones, which are defined over the columns of \mathbf{U}, \mathbf{V} (cf. (6)), the forms proposed in [26] are defined over the full matrix factors \mathbf{U}, \mathbf{V} and thus their application to the online processing scenario is not straightforward.

above, we may define the following minimization problem for each time iteration n ,

$$\min_{\mathbf{U}_n, \mathbf{V}(n,:), \mathbf{S}(:,n)} \frac{1}{2} \|(\mathbf{Y}_n - \mathbf{U}_n \mathbf{V}_n^T - \mathbf{S}_n) \mathbf{\Xi}_n^{\frac{1}{2}}\|_F^2 + \lambda (\|\mathbf{U}_n \mathbf{D}_n^{\frac{1}{2}}\|_F^2 + \|\mathbf{V}_n \mathbf{D}_n^{\frac{1}{2}}\|_F^2) + \mu \|\mathbf{S}_n \odot \mathbf{W}_n^{\frac{1}{2}}\|_F^2, \quad (8)$$

where $\mathbf{Y}_n \triangleq \mathbf{Y}(:, 1:n)$, $\mathbf{S}_n = [\mathbf{S}_{n-1}, \mathbf{S}(:, n)]$, $\mathbf{V}_n^T = [\mathbf{V}_{n-1}^T, \mathbf{V}(n, :)^T]$ and \mathbf{S}_{n-1} , \mathbf{V}_{n-1} contain the estimates of the columns and rows of \mathbf{S}, \mathbf{V} respectively, up to time $n-1$. \mathbf{D}_n is computed from (4) based on the columns of \mathbf{U}_n and \mathbf{V}_n . The weight matrix $\mathbf{\Xi}_n = \text{diag}(\xi^{n-1}, \xi^{n-2}, \dots, 1)$, $0 < \xi < 1$ weighs more heavily recent estimates than older ones. By minimizing (8) with respect to $\mathbf{S}(:, n)$ we get

$$\mathbf{S}(:, n) = (\mathbf{Y}(:, n) - \mathbf{U}_n \mathbf{V}(n, :)^T) \odot (\mathbf{1} + \mu \mathbf{W}(:, n)), \quad (9)$$

where $\mathbf{W}(i, n) = (\mathbf{S}(i, n)^2 + \epsilon)^{-\frac{1}{2}}$, $i = 1, 2, \dots, m$. Minimization of (8) w.r.t $\mathbf{V}(n, :)$ gives

$$\mathbf{V}(n, :) = (\mathbf{Y}(:, n) - \mathbf{S}(:, n)) \mathbf{U}_n^T (\mathbf{U}_n^T \mathbf{U}_n + \lambda \mathbf{D}_n)^{-1} \quad (10)$$

Finally, \mathbf{U}_n , the subspace data matrix estimate at time n , is obtained from (8) as follows,

$$\mathbf{U}_n = (\mathbf{Y}_n - \mathbf{S}_n) \mathbf{\Xi}_n \mathbf{V}_n (\mathbf{V}_n^T \mathbf{\Xi}_n \mathbf{V}_n + \lambda \mathbf{D}_n)^{-1}. \quad (11)$$

By defining $\mathbf{T}_n = (\mathbf{Y}_n - \mathbf{S}_n) \mathbf{\Xi}_n \mathbf{V}_n$ and $\mathbf{P}_n = \mathbf{V}_n^T \mathbf{\Xi}_n \mathbf{V}_n$ we easily derive the following efficient recursive equations

$$\mathbf{T}_n = \xi \mathbf{T}_{n-1} + (\mathbf{Y}(:, n) - \mathbf{S}(:, n)) \mathbf{V}(n, :), \quad (12)$$

$$\mathbf{P}_n = \xi \mathbf{P}_{n-1} + \mathbf{V}(n, :)^T \mathbf{V}(n, :). \quad (13)$$

By arranging the above equations properly and adjusting in a reasonable way the time indexes of the various quantities, we obtain the online RPCA algorithm tabulated as Algorithm 2. The main features of the proposed online reweighted LS RPCA algorithm are summarized in the following remarks.

- 1) To estimate the sparse vector $\mathbf{S}(:, n)$ the first two steps of Algorithm 2 are repeated K times for each time iteration n . As verified in our experiments a value of $K = 2$ or 3 may be sufficient in most cases.
- 2) To compute \mathbf{D}_n we do not need to store the whole matrix \mathbf{V}_n (whose size increases with time). It can be shown that the squared Euclidean norms of the columns of \mathbf{V}_n required in (4) can be recursively calculated with $O(d)$ cost ($O(1)$ per each column).
- 3) The proposed algorithm is *memory optimal* having the minimum possible memory requirements, i.e., $O(md)$, [2].
- 4) The computational complexity of Algorithm 2 is $O(md^2)$ per iteration (basically operations required to compute $\mathbf{V}(n, :)$ and \mathbf{U}_n). According to the information provided in [2], it can be considered as one of the fastest online RPCA algorithms.
- 5) As verified in our experiments, the new online algorithm converges fast starting with a random subspace initialization, i.e., ‘good’ initial subspace estimates are not necessary for the algorithm to work properly, as e.g., in [15] and [16].
- 6) The algorithm uses two regularization parameters λ and μ , which can be selected via cross-validation. We have observed though that (possibly) due to its IRLS nature, the proposed scheme seems to have low sensitivity to λ, μ .

Algorithm 2: Online Reweighted LS RPCA Algorithm.

Input: $\lambda > 0, \mu > 0, \xi$

Initialize: $\mathbf{U}_0, \mathbf{V}(0, :), \mathbf{S}^{(0)}(:, 0), \mathbf{D}_0$

for $n = 1 \dots$

Input $\mathbf{Y}(:, n)$

for $k = 1$ to K

Compute $\mathbf{W}^{(k)}(:, n)$ based on $\mathbf{S}^{(k-1)}(:, n-1)$

$\mathbf{S}^{(k)}(:, n) \leftarrow [\mathbf{Y}(:, n) - \mathbf{U}_{n-1} \mathbf{V}(n-1, :)^T]$

$\odot [\mathbf{1} + \mu \mathbf{W}^{(k)}(:, n)]$

end for

$\mathbf{V}(n, :) \leftarrow [\mathbf{Y}(:, n) - \mathbf{S}^{(K)}(:, n)]^T \mathbf{U}_{n-1}$
 $(\mathbf{U}_{n-1}^T \mathbf{U}_{n-1} + \lambda \mathbf{D}_{n-1})^{-1}$

$\mathbf{T}_n \leftarrow \xi \mathbf{T}_{n-1} + [\mathbf{Y}(:, n) - \mathbf{S}^{(K)}(:, n)] \mathbf{V}(n, :)$

$\mathbf{P}_n \leftarrow \xi \mathbf{P}_{n-1} + \mathbf{V}(n, :)^T \mathbf{V}(n, :)$

$\mathbf{U}_n \leftarrow \mathbf{T}_n (\mathbf{P}_n + \lambda \mathbf{D}_{n-1})^{-1}$

Compute \mathbf{D}_n from (4) based on $\mathbf{U}_n, \mathbf{V}_n$

$\mathbf{S}^{(0)}(:, n) \leftarrow \mathbf{S}^{(K)}(:, n)$

end for

The effectiveness and high computational efficiency of the proposed algorithm are corroborated in the next section, where some indicative empirical results are presented.

IV. NUMERICAL SIMULATIONS

The proposed algorithm is compared with the state-of-the-art NORST algorithm, reported in [15]. As is shown in [2], NORST outperforms other state-of-the-art algorithms, while it also enjoys theoretical guarantees regarding its performance. NORST’s Matlab code has been provided by the authors of [15], whom we would like to thank.

A. Simulated Data Experiment

In this experiment we aim at testing the performance of the proposed algorithm when it comes to the recovery accuracy of both the low-rank and the sparse components. The simulated low-rank component $\mathbf{L} \in \mathbb{R}_+^{800 \times 5000}$ is constructed as $\mathbf{L} = [\mathbf{L}_1 \mathbf{L}_2]$, where $\mathbf{L}_1 = \mathbf{U}_1 \mathbf{V}_1^T \in \mathbb{R}_+^{800 \times 2000}$ and $\mathbf{L}_2 = \mathbf{U}_2 \mathbf{V}_2^T \in \mathbb{R}_+^{800 \times 3000}$. The low-rank matrix factors $\mathbf{U}_1 \in \mathbb{R}_+^{800 \times 10}$, $\mathbf{V}_1 \in \mathbb{R}_+^{2000 \times 10}$ and $\mathbf{U}_2 \in \mathbb{R}_+^{800 \times 8}$, $\mathbf{V}_2 \in \mathbb{R}_+^{3000 \times 8}$ take values from a uniform distribution defined in the interval $[0 \ 1]$. As a result, by construction, the first 2000 columns of \mathbf{L} lie in a 10-dimensional subspace and its last 3000 columns in a 8-dimensional subspace. We simulate a sparse (outliers) matrix $\mathbf{S} \in \mathbb{R}_+^{800 \times 5000}$ having 5% nonzero entries randomly distributed in the matrix, which take values from a uniform distribution defined in the interval $[0 \ 20]$. Independent identically distributed zero mean Gaussian noise with variance $\sigma^2 = 10^{-2}$ is also added to the data, which thus obey the model given in (1). The tested online algorithms process the resulting data matrix \mathbf{Y} column-by-column.

The proposed algorithm is initialized randomly and its parameters are selected as $\lambda = 200$ and $\mu = 0.5$. The forgetting factor ξ is set to 0.99. NORST algorithm is initialized as described in [15]. The algorithms are assessed in terms of the normalized mean squared error (NMSE) between the true and the estimated columns of both their low-rank and sparse components. These measures are computed as the average of the respective results of 100 independent realizations of the algorithms and are depicted in Fig. 1. We observe that the proposed online reweighted LS

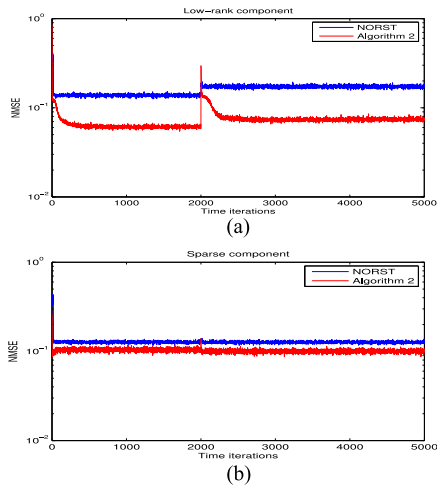


Fig. 1. NMSE of a) the low-rank matrix columns and b) the sparse matrix columns vs. time iterations.

RPCA algorithm has a lower NMSE than NORST regarding the recovery of both the low-rank and the sparse terms of the data matrix. Both algorithms are able to detect the change of the low-rank data subspace dimension at time iteration 2000, but again Algorithm 2 converges to a lower NMSE. It should be noted that Algorithm 2 starts with an overestimate ($d = 20$) of the data subspace dimension and as it evolves sequentially, reduces it first to 10 and after time iteration 2000, to 8, thus providing a correct estimation of the true ranks of L_1 and L_2 . In addition, in this specific experiment, Algorithm 2 processes on average 658 matrix columns/sec and NORST 151 columns/sec, which verifies the computational efficiency of the proposed method.

B. Real Data Experiments

In this section we validate the effectiveness of the proposed algorithm on the video foreground-background separation problem. To this end, two different popular datasets are utilized i.e., the Meeting Room dataset and the Lobby dataset, described next. In this experiment the convex batch PCP algorithm of [1] is used also for comparison purposes. It should be noted that for the NORST algorithm, we set its parameters as suggested by the authors in [15] and the regularization parameter of PCP has been fine-tuned. Moreover, the subspace matrix for the NORST algorithm has been previously initialized by adopting the same scheme as in [15]. In the case of the proposed online reweighted LS RPCA algorithm, we initialize the rank of the background to 5, the low-rank regularization parameter λ is set to 640 and the sparsity promoting parameter μ is fixed to 0.1. Additionally, $\xi = 0.99$ and $K = 3$ inner iterations are performed to estimate the sparse foreground frames. The same setting was followed for the experiments conducted on both real datasets.

1) *Meeting Room Dataset*: This video dataset consists of 2964 grayscale images (frames) of spatial resolution 64×80 . Since the first 1755 video frames contain no “outliers,” we consider only the last 1209 frames. In this part of the video, after the first 400 frames, a man wearing a white shirt is getting in the room. Since the color of the curtain in the room is the same with the color of the man’s shirt, the foreground-background separation task becomes quite challenging. As it can be observed

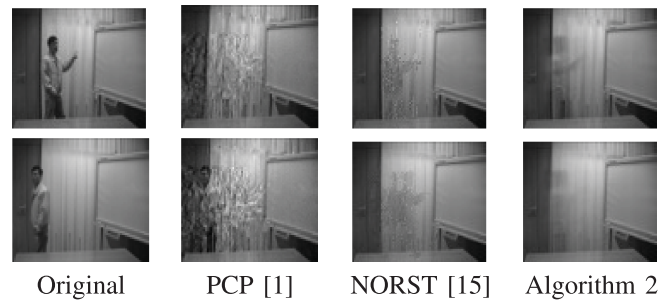


Fig. 2. Background estimation results of PCP [1], NORST [15] and Algorithm 2 on the Meeting Room dataset. Frames 760, 1160.

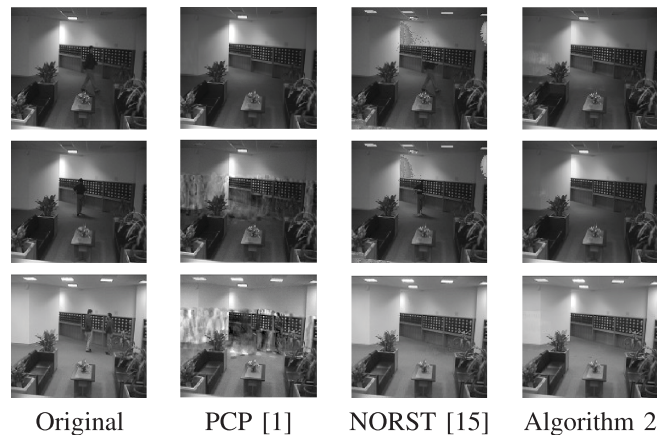


Fig. 3. Background estimation results of PCP [1], NORST [15] and Algorithm 2 on the Lobby dataset. Frames 350, 610, 1150.

in Fig. 2, the proposed reweighted LS RPCA algorithm offers comparable background estimation results to those of NORST, though it has been randomly initialized. Both algorithms, outperform PCP. PCP being a batch algorithm fails to capture the dynamic changes of the background. In addition, Algorithm 2 processes 344 frames/sec and hence is much faster than NORST, which achieves a rate of 16 frames/sec.

2) *Lobby Dataset*: This dataset consists of 1555 frames of 128×160 pixels. Background estimation results of NORST, Algorithm 2 and PCP for three frames of this dataset are shown in Fig. 3. We see that, especially in video frames 350 and 610, Algorithm 2 is able to remove foreground information more reliably than NORST and PCP. This is achieved by processing the video sequence much faster with a rate of 277 frames/sec, as compared to the 15 frames/sec rate of NORST. The above results regarding runtime complexity indicate that the proposed algorithm might be used for near real-time video processing.

V. CONCLUSION

In this letter a novel online RPCA algorithm has been presented. The main characteristic of the proposed algorithm is that both the factors of the low-rank component and the sparse component of the data matrix are effectively updated via a reweighted LS optimization mechanism. The new online estimation algorithm exhibits high computational and memory efficiency without compromising performance. A theoretical convergence and correctness analysis of the algorithm is under current investigation.

REFERENCES

- [1] E. J. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, May 2011, Art. no. 11.
- [2] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 32–55, Jul. 2018.
- [3] H. Wang, Y. Cen, Z. He, R. Zhao, Y. Cen, and F. Zhang, "Robust generalized low-rank decomposition of multimatrices for image recovery," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 969–983, May 2017.
- [4] H. Wang, Y. Li, Y. Cen, Z. He, and F. Zhang, "Multi-matrices low-rank decomposition with structural smoothness for image denoising," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 349–361, Feb. 2020.
- [5] R. Otazo, E. Candes, and D. Sodickson, "Low-rank and sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic component," *Mach. Reson. Med.*, vol. 73, no. 3, pp. 1125–1136, Mar. 2015.
- [6] R. Li, J. Pan, Y. Si, B. Yan, Y. Hu, and H. Qin, "Specular reflections removal for endoscopic image sequences with adaptive-RPCA decomposition," *IEEE Trans. Med. Imag.*, vol. 39, no. 2, pp. 328–340, Feb. 2020.
- [7] W. He, H. Zhang, H. Shen, and L. Zhang, "Hyperspectral image denoising using local low-rank matrix recovery and global spatial-spectral total variation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 3, pp. 713–729, Mar. 2018.
- [8] F. Kucuk, B. Toreyin, and F. Çelebi, "Sparse and low-rank matrix decomposition-based method for hyperspectral anomaly detection," *J. Appl. Remote Sens.*, vol. 13, no. 1, Feb. 2019, Art. no. 014519.
- [9] T. Bouwmans, A. Sobral, S. Javed, S. Jung, and E. Zahzah, "Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset," *Comput. Sci. Rev.*, vol. 23, pp. 1–71, Feb. 2017.
- [10] S. Javed, A. Mahmood, S. Al-Maadeed, T. Bouwmans, and S. Jung, "Moving object detection in complex scene using spatio-temporal structured-sparse RPCA," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 1007–1022, Feb. 2019.
- [11] J. Kennedy, L. Balzano, S. Wright, and C. Taylor, "Online algorithms for factorization-based structure from motion," *Comput. Vision Image Understanding*, vol. 150, pp. 139–152, 2016.
- [12] G. Arvanitis, A. Lalos, K. Moustakas, and N. Fakotakis, "Outliers removal of highly dense and unorganized point clouds acquired by laser scanners in urban environments," in *Proc. Int. Conf. Cyberworlds*, 2018, pp. 415–418.
- [13] T. Bouwmans, N. Aybat, and E. Zahzah, *Handbook on Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. Boca Raton, FL, USA: CRC Press, May 2016.
- [14] T. Bouwmans, S. Javed, H. Zhang, Z. Lin, and R. Otazo, "On the applications of robust PCA in image and video processing," *Proc. IEEE*, vol. 106, no. 8, pp. 1427–1457, Aug. 2018.
- [15] P. Narayanamurthy and N. Vaswani, "Nearly optimal robust subspace tracking," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 3701–3709.
- [16] P. Narayanamurthy and N. Vaswani, "Provable dynamic robust PCA or robust subspace tracking," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1547–1577, Mar. 2019.
- [17] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2012, pp. 1568–1575.
- [18] J. Feng, H. Xu, and S. Yan, "Online robust PCA via stochastic optimization," in *Proc. Advances Neural Inf. Process. Syst.*, Dec. 2013, pp. 404–412.
- [19] S. Chouvardas, Y. Kopsinis, and S. Theodoridis, "Robust subspace tracking with missing entries: The set-theoretic approach," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5060–5070, Oct. 2015.
- [20] P. V. Giampouras, A. A. Rontogiannis, and K. D. Koutroumbas, "Alternating iteratively reweighted least squares minimization for low-rank matrix factorization," *IEEE Trans. Signal Process.*, vol. 67, no. 2, pp. 490–503, Jan. 2019.
- [21] P. V. Giampouras, A. A. Rontogiannis, and K. D. Koutroumbas, "Robust PCA via alternating iteratively re-weighted low-rank matrix factorization," in *Proc. IEEE Int. Conf. Image Process.*, 2018, pp. 3383–3387.
- [22] I. Daubechies *et al.*, "Iteratively reweighted least squares minimization for sparse recovery," *Commun. Pure Appl. Math.*, vol. 63, pp. 1–38, 2010.
- [23] P. V. Giampouras, A. A. Rontogiannis, and K. D. Koutroumbas, "An IRLS approach for low-rank matrix factorization," in *Proc. SPARS*, Jul. 2019.
- [24] M. V. Örnhaug, C. Olsson, and A. Heyden, "Bilinear parameterization for differentiable rank-regularization," 2019, *arXiv: 1811.11088*.
- [25] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Proc. Advances Neural Inf. Process. Syst.*, Dec. 2005, pp. 1329–1336.
- [26] F. Shang, J. Cheng, Y. Liu, Z.-Q. Luo, and Z. Lin, "Bilinear factor matrix norm minimization for robust PCA: Algorithms and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 9, pp. 2066–2080, Sep. 2018.
- [27] Y. Hu *et al.*, "Group sparse optimization via $\ell_{p,q}$ regularization," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 960–1011, 2017.
- [28] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data," *IEEE Signal Process. Mag.*, vol. 33, no. 1, pp. 57–77, Jan. 2016.