Information Sciences 513 (2020) 581–599

Contents lists available at ScienceDirect

Information Sciences

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

Robust principal component analysis: A factorization-based approach with linear complexity

Chong Peng[a], Yongyong Chen[b], Zhao Kang[c], Chenglizhao Chen[a,*], Qiang Cheng[d,e]

[a] College of Computer Science and Technology, Qingdao University, China
[b] Department of Computer and Information Science, University of Macau, China
[c] School of Computer Science and Engineering, University of Electronic Science and Technology of China, China
[d] Department of Computer Science, University of Kentucky, USA
[e] Institute of Biomedical Informatics, University of Kentucky, USA

ARTICLE INFO

Article history:
Received 14 April 2019
Revised 27 September 2019
Accepted 29 September 2019
Available online 30 September 2019

Keywords:
Robust principal component analysis
Factorization
Linear complexity

ABSTRACT

Low-rankness has been widely observed in real world data and there is often a need to recover low-rank matrices in many machine learning and data mining problems. Robust principal component analysis (RPCA) has been used for such problems by separating the data into a low-rank and a sparse part. The convex approach to RPCA has been well studied due to its elegant properties in theory and many extensions have been developed. However, the state-of-the-art algorithms for the convex approach and their extensions are usually expensive in complexity due to the need for solving singular value decomposition (SVD) of large matrices. In this paper, we propose a novel RPCA model based on matrix tri-factorization, which only needs the computation of SVDs for very small matrices. Thus, this approach reduces the complexity of RPCA to be linear and makes it fully scalable. It also overcomes the drawback of the state-of-the-art scalable approach such as AltProj, which requires the precise knowledge of the true rank of the low-rank component. As a result, our method is about 4 times faster than AltProj. Our method can be used as a light-weight, scalable tool for RPCA in the absence of the precise value of the true rank.

1. Introduction

Principal component analysis (PCA) is one of the most widely used unsupervised learning method for dimension reduction. It is often used as a pre-processing or intermediate step for data analysis in various applications, such as face recognition, classification, recommender systems design, etc. Given a data matrix $X \in \mathcal{R}^{d\times n}$, the basic idea of PCA is to find a low-dimensional subspace where the dataset lies such that the most variability of the data is retained. Mathematically, the classic PCA seeks the orthogonal basis vectors by solving the minimization problem of $\min_{PP^T=I_k} \|X - PP^TX\|_F^2$, where I_k is an identity matrix of size $k \times k$ and $\|\cdot\|_F$ is the Frobenius norm. A straightforward solution to PCA is obtained via singular value decomposition (SVD), which gives the best rank-k approximation of the data. It is well known that PCA is sensitive to outliers due to the use of the Frobenius norm. However, in many cases, modern datasets are noisy due to various reasons at data collection stage such as sensor failures, which brings challenges to the classic PCA.

* Corresponding author.
E-mail address: cclz123@163.com (C. Chen).

https://doi.org/10.1016/j.ins.2019.09.074
0020-0255/

Various approaches to robust PCA (RPCA) have been proposed to combat the above mentioned drawback, including alternating minimization [1], random sampling techniques [2,3], multivariate trimming [4], and others [5,6], among which a new type of RPCA method has emerged and drawn significant attentions [7,8]. It assumes that X can be separated into two parts: a low-rank L and a sparse S, where the separation can be obtained by solving the following problem:

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

where $\|\cdot\|_0$ is the ℓ_0 (pseudo) norm which counts the number of nonzero elements of a matrix, and $\lambda > 0$ is a balancing parameter. Because it is generally NP-hard to minimize the rank or the ℓ_0 norm, in practice, (1) is often relaxed to the following convex optimization problem [8]:

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

where $\|L\|_* = \sum_{i=1}^{\min(d,n)} \sigma_i(L)$ is the nuclear norm of L with $\sigma_i(L)$ denoting the i-th largest singular value of L, and $\|S\|_1 = \sum_{ij} |S_{ij}|$ is the ℓ_1 norm. It has been revealed that (2) can exactly separate L with the true rank r from S under some mild conditions with overwhelming probability [8]. A number of algorithms have been developed to solve (2), including singular value thresholding (SVT) [9], accelerated proximal gradient (APG) [10], and two versions of augmented Lagrange multipliers (ALM) based approaches [11]: exact ALM and inexact ALM (IALM). Among these algorithms, the ALM based are state-of-the-art ones for solving (2), which need to compute SVDs of $d \times n$ matrices per iteration. To improve efficiency, another ALM based algorithm adopts PROPACK package [12], which solves only partial instead of full SVDs. However, this is still computationally expensive when d and n are both large. Despite the elegant theory of the convex RPCA formulation of (2), it has three major drawbacks: 1) When the assumed underlying low-rank matrix has no incoherence guarantee [8], or the data get grossly corrupted, the results can be far from the true underlying ones; 2) The nuclear norm may lead to a biased estimation of the rank [13]; 3) It has high complexity in computation. To combat these drawbacks, [13] uses a nonconvex rank approximation to more accurately approximate the rank of L. However, it still needs to solve full SVDs. Methods such as [14,15] need only to solve partial SVDs, which significantly reduces the complexity compared to the computation of full SVD; for example, AltProj has a complexity of $O(r^2dn)$ [15]. However, if r is not known a *priori*, [14,15] may fail to recover L correctly.

To further reduce the complexity, enhance the scalability and alleviate the dependence on the knowledge of r, in this paper, we propose a factorization-based model for RPCA. With the factorization approach, we assume that L can be decomposed as UCV^T with $U \in \mathcal{R}^{d\times k}$, $C \in \mathcal{R}^{k\times k}$, $V \in \mathcal{R}^{n\times k}$, and $k \ll \min(d, n)$. This model relaxes the requirement on a *priori* knowledge of the rank of L, which only assumes that it is upper bounded by k. With this special structure, scalable algorithms can be developed to optimize our model efficiently. Briefly, we summarize the key contributions of this paper as follows:

- We propose a factorization-based model for RPCA, allowing the recovery of the low-rank component with or without a *priori* knowledge of its true rank r.
- Efficient and scalable ALM-type optimization algorithms are developed with theoretical convergence guarantees. Moreover, it can be formally proven that our model is equivalent to the classic, convex formulation of RPCA under certain mild conditions (as specified in Theorem 1); hence, theoretical properties of the convex approach can extend to our model.
- Empirically, extensive experiments confirm the effectiveness of our algorithms both quantitatively and qualitatively in various applications.

We organize the rest of this paper as follows. We briefly review related work in Section 2. Then we present the proposed models in Section 3. The optimization algorithms for two variants of our model are developed in Section 4. Then we theoretically analyze the convergence of the proposed algorithms in Section 5. We conduct extensive experiments to evaluate the proposed algorithms in Section 6. Finally, we conclude our paper in Section 7.

2. Related work

The approach to RPCA in (1) has received considerable attention and its convex relaxation (2) has been thoroughly studied [9]. To exploit the example-wise sparsity of the sparse component, the $\ell_{2,1}$ norm has been adopted to replace the ℓ_1 norm in (2) [16,17]:

$$\min_{Z,S} \|L\|_* + \lambda \|S\|_{2,1}, \quad s.t. \quad X = L + S, \tag{3}$$

where $\|\cdot\|_{2,1}$ is defined to be the sum of ℓ_2 norms of column vectors of a matrix, which promotes column-wise sparsity. When a matrix has large singular values, the nuclear norm may be far from an accurate approximation of the rank. To combat this issue, nonconvex rank approximations have been considered in RPCA as well as other applications, such as subspace clustering, faster numerical linear algebra, and matrix completion [13,18–20]. For example, [13] developed a nonconvex RPCA model with nonconvex rank appproximation:

$$\min_{L,S} \|L\|_\gamma + \lambda \|S\|_{2,1}, \quad s.t. \quad X = L + S, \tag{4}$$

where $\|L\|_\gamma = \sum_i \frac{(1+\gamma)\sigma_i(L)}{\gamma+\sigma_i(L)}$ is a rank approximation with $\gamma > 0$. As a common requirement, the above approaches usually need to solve SVDs. When the matrix is large, the computation of SVD, in general, is intensive. To reduce the complexity of RPCA, several approaches have been attempted. For example, nonconvex alternating minimization techniques have been used in RPCA [15]. As previously mentioned, the resulting algorithm AltProj has a cost of $O(r^2dn)$ per iteration which is comparable to PCA when r is small. It involves finding the rank-r approximation of $d \times n$ matrices, where r is the true rank of L. Another approach, fixed-rank RPCA (FrALM) [14], also assumes the availability of this prior knowledge with the following formulation:

$$\min_{L,S} \|S\|_F \quad s.t. \quad X = L + S, \quad \text{rank}(L) = r. \tag{5}$$

The above optimization problem is solved by adopting the exact ALM. Another approach such as [21] assumes the pair-wise similarity of the data to recover low-rank part, which achieves fast computation with the Sherman–Morrison–Woodbury formula. The above approaches all exploit the prior knowledge on the rank of the low-rank component to reduce the computation of full SVDs to that of partial SVDs.

3. Fast Factorization-based RPCA

In this section, we formulate the Fast Factorization-based RPCA model (FFP). Depending on whether the information on the true rank r is present or not, two variants of our model and their corresponding optimization algorithms are developed. The proposed models and algorithms are shown to be closely related to the convex RPCA model (2).

3.1. Formulation

Motivated by the convex RPCA approach, we model the data as $X = L + S$, where L is the low-rank part and can be decomposed as $L = UCV^T$ with $U \in \mathcal{R}^{d\times k}$, $C \in \mathcal{R}^{k\times k}$, and $V \in \mathcal{R}^{n\times k}$. The decomposition UCV^T effectively provides a natural upper bound for the rank of L, which is k. The upper bound k can be used to relax the stringent requirement on the knowledge of the true rank by AltProj algorithm. In this paper, we adopt the ℓ_1 norm to recover the sparse part S due to the following reasons: 1) As will be clearer in later sections, our method is closely related with the classic convex RPCA of (2). Thus, the ℓ_1 norm-based objective function ensures that the elegant theoretical results are applicable to our method; 2) The ℓ_1 norm is easy to solve and theoretical convergence can be guaranteed; 3) It is not within the scope of this paper to try out various norms for the sparse term. Thus, with ℓ_1 norm, we propose the following factorization-based objective function:

$$\min_{S,U,C,V} \|S\|_1, \quad s.t. \quad X = UCV^T + S. \tag{6}$$

It is seen that (6) is nonconvex and its solution can change with scaling factors multiplied on the factor matrices U, V, and C. To facilitate the uniqueness of the solution, we enforce two constraints $U^TU = I_k$ and $V^TV = I_k$ in the above model. Then these constraints naturally lead to an interpretation: the left factor matrix U can be regarded as basis vectors of the column-subspace in which L resides, while the right factor matrix V can be regarded as the relaxed indicator matrix that indicates which subspaces the columns of L belong to. This interpretation also implies that C can be regarded as a core matrix of the data that retains the essential information since U and V are required to be orthonormal. We incorporate this structural requirement into (6), leading to the following model:

$$\min_{S,U,C,V} \|S\|_1 \quad s.t. \quad X = UCV^T + S, \quad U^TU = I, V^TV = I. \tag{7}$$

Up to now, we have considered the case where the precise knowledge of r is known, where we naturally let $k = r$ in (7). However, in the absense of such information on r, an arbitrary value that is far from r may be picked for k. Hence, (7) may lack the desired capability of recovering L with an (unknown) rank of r when an arbitrary k is used. To resolve this problem, we propose to combine the advantages of the convex RPCA and the proposed factorization-based fixed-rank approach, which leads to the following optimization problem:

$$\begin{aligned} &\min_{S,U,C,V} \|S\|_1 + \lambda\|UCV^T\|_* \\ &s.t. \quad X = UCV^T + S, \quad U^TU = I, V^TV = I, \end{aligned} \tag{8}$$

where $\lambda > 0$ is a trade-off parameter that balances the two terms in (8). Frequently, with specific domain information, a proper upper bound $k \ll \min(d, n)$ can be chosen. $k \ll \min(d, n)$ is meaningful in that it provides a close rather than arbitrary upper bound for the true rank, thus yielding more accurate recovery. Moreover, smaller k reduces the complexity and renders more efficient computation, which is essential for real world applications. Even in the case that the precise value of r or clear domain information is not available, a proper $k \geq r$ can still be chosen. Because, in the worst case, we may let $k = \min(d, n)$, such that $k \geq r$ always hold. It should be noted that there exists a close connection between (8) and the convex RPCA, which is revealed by the following theorem.

Theorem 1. *Given a proper λ that exactly recovers L and S for convex RPCA of (2) with the true rank $r \leq k \leq \min(d, n)$. It is always possible to obtain a global minimizer for the nonconvex problem (8).*

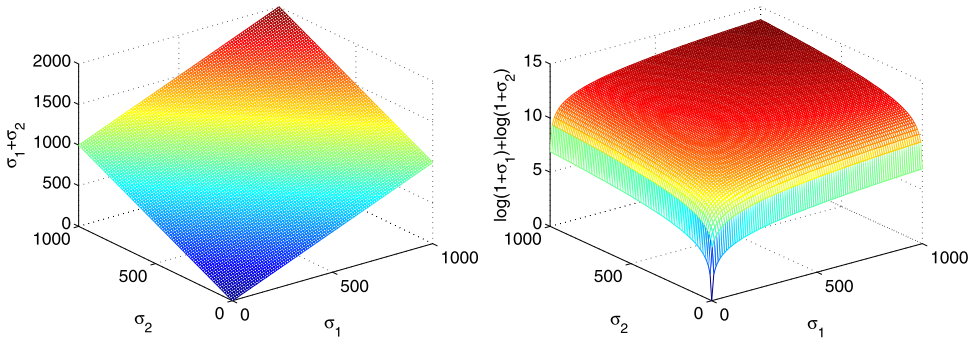


Fig. 1. Example of approximations with two singular values: (a) the nuclear norm approximation; (b) the first-order log-determinant rank approximation.

Proof. For ease of notation, we define the objectives in (2) and (8) to be $f(L, S)$ and $g(U, C, V, S)$, respectively. It is seen that $f(L, S)$ is convex and thus there exists a global minimizer which we denote as $\{L^*, S^*\}$. Hence, for any $\{L, S\}$, we have $f(L, S) \geq f(L^*, S^*)$.

For any solution of (8), namely $\{U', C', V', S'\}$, we can define $L' = U'C'V'^T$. It is straightforward to see that $\{L', S'\}$ is a feasible point of (2) since the constraint $L' + S' = U'C'V'^T + S' = X$ is satisfied. Hence, for any solution $\{U', C', V', S'\}$ of (8), the corresponding objective value $g(U', C', V', S')$ is lower-bounded and we have $g(U', C', V', S') = f(L', S') \geq f(L^*, S^*)$.

For $\{L^*, S^*\}$, it is straightforward to decompose L^* in the following way. Let $L^* = \mathcal{U}\Sigma_{L^*}\mathcal{V}^T$ be the thin SVD of L^* with $\mathcal{U} \in \mathcal{R}^{d\times n}$, $\Sigma_{L^*} \in \mathcal{R}^{n\times n}$, $\mathcal{V} \in \mathcal{R}^{n\times n}$, and $\mathcal{U}^T\mathcal{U} = \mathcal{V}^T\mathcal{V} = I_n$. Since the rank of L^* is $r \leq k \leq \min(d, n)$, we can write L^* as

$$L^* = \begin{bmatrix}\mathcal{U}_1 & \mathcal{U}_2 & \mathcal{U}_3\end{bmatrix}\begin{bmatrix}\Sigma_{L_1^*} & & \\ & \Sigma_{L_2^*} & \\ & & \Sigma_{L_3^*}\end{bmatrix}\begin{bmatrix}\mathcal{V}_1^T \\ \mathcal{V}_2^T \\ \mathcal{V}_3^T\end{bmatrix} = \begin{bmatrix}\mathcal{U}_1 & \mathcal{U}_2\end{bmatrix}\begin{bmatrix}\Sigma_{L_1^*} & \\ & \Sigma_{L_2^*}\end{bmatrix}\begin{bmatrix}\mathcal{V}_1^T \\ \mathcal{V}_2^T\end{bmatrix} = \mathcal{U}_{12}\Sigma_{L_{12}^*}\mathcal{V}_{12}^T,$$

where $\mathcal{U}_{12} = \begin{bmatrix}\mathcal{U}_1 & \mathcal{U}_2\end{bmatrix}$, $\Sigma_{L_{12}^*} = \begin{bmatrix}\Sigma_{L_1^*} & \\ & \Sigma_{L_2^*}\end{bmatrix}$, $\mathcal{V}_{12} = \begin{bmatrix}\mathcal{V}_1 & \mathcal{V}_2\end{bmatrix}$, $\Sigma_{L_1^*} \in \mathcal{R}^{r\times r}$, and $\Sigma_{L_2^*} \in \mathcal{R}^{(k-r)\times(k-r)}$, $\Sigma_{L_3^*} \in \mathcal{R}^{(n-k)\times(n-k)}$ are zero matrices. Here, the second equality holds due to the fact that $\Sigma_{L_3^*}$ is a zero matrix.

Define $\mathcal{U}_0$, $\mathcal{V}_0$ to be unitary matrices such that $\mathcal{U}_0^T\mathcal{U}_0 = \mathcal{U}_0\mathcal{U}_0^T = \mathcal{V}_0^T\mathcal{V}_0 = \mathcal{V}_0\mathcal{V}_0^T = I_k$. Then it is seen that

$$L^* = \mathcal{U}_{12}\Sigma_{L_{12}^*}\mathcal{V}_{12}^T = \mathcal{U}_{12}\mathcal{U}_0\mathcal{U}_0^T\Sigma_{L_{12}^*}\mathcal{V}_0\mathcal{V}_0^T\mathcal{V}_{12}^T = (\mathcal{U}_{12}\mathcal{U}_0)(\mathcal{U}_0^T\Sigma_{L_{12}^*}\mathcal{V}_0)(\mathcal{V}_{12}\mathcal{V}_0)^T.$$

Define $U^* = \mathcal{U}_{12}\mathcal{U}_0$, $V^* = \mathcal{V}_{12}\mathcal{V}_0$, and $C^* = \mathcal{U}_0^T\Sigma_{L_{12}^*}\mathcal{V}_0$, then it is straightforward to see that $g(U^*, C^*, V^*, S^*) = f(L^*, S^*)$, whereas the constraints $L^* + S^* = U^*C^*(V^*)^T + S^* = X$, $(U^*)^TU^* = \mathcal{U}_0^T\mathcal{U}_{12}^T\mathcal{U}_{12}\mathcal{U}_0 = I_k$, and $(V^*)^TV^* = \mathcal{V}_0^T\mathcal{V}_{12}^T\mathcal{V}_{12}\mathcal{V}_0 = I_k$ are satisfied. Hence $\{U^*, C^*, V^*, S^*\}$ is a global minimizer of (8). □

Theorem 1 reveals the close connection between the convex RPCA and (8), where it provides a way to solve (8). However, the provided solution is expensive to obtain and more efficient optimization strategy is desirable. In this paper, we will focus on designing a scalable algorithm for RPCA with a provable convergence certificate.

It is noted that (8) incorporates the nuclear norm to approximate the rank function, which has been widely adopted in low-rank learning problems [22–29]. However, recent studies show that the nuclear norm is not accurate in approximating the true rank of a matrix, while more accurate nonconvex rank approximations may help improve learning performance [20]. Here, we adopt the typical log-determinant rank approximation, $\|Y\|_{ld}$, which is defined as:

$$\|Y\|_{ld} = \log\det(I + (Y^TY)^{\frac{1}{2}}) = \sum_{i=1}^{\min(d,n)}\log(1+\sigma_i(Y)).$$

To visually understand how the new rank approximation approximates the true rank function, we show a simple yet convincing example in Fig. 1. It is seen that the nuclear norm is far larger than the true rank if there are large singular values while the log-determinant rank approximation significantly improves the behavior. It is easy to check that the following properties hold, which will be used in later sections:

- Since $\sigma_i(Y) \geq 0$, $\log(1+\sigma_i(Y)) \geq 0$. Thus $\|Y\|_{ld} \geq 0$ always holds.
- For large $\sigma_i(Y)$, we have $\log(1+\sigma_i(Y)) \ll \sigma_i(Y)$. This reveals that $\text{rank}(Y) < \|Y\|_{ld} \ll \|Y\|_*$, implying closer approximation to the true rank.
- $\|Y\|_{ld}$ is nonconvex, continuous, and differentiable.

Such properties allow the nonconvex rank approximation to better approximate the true rank function. Moreover, the problem is easy to solve and it will be seen that the convergence can be readily guaranteed by using the log-determinant

rank approximation. Thus, to exploit the desirable properties and without loss of generality, we adopt the log-determinant rank approximation in the RPCA and obtain the following model:

$$\begin{aligned} &\min_{S,U,C,V} \|S\|_1 + \lambda \|UCV^T\|_{ld} \\ &s.t. \quad X = UCV^T + S, \quad U^TU = I, V^TV = I. \end{aligned} \tag{9}$$

It is seen that model (9) can be reduced to the following model:

$$\begin{aligned} &\min_{S,C,U,V} \|S\|_1 + \lambda \|C\|_{ld} \\ &s.t. \quad X = UCV^T + S, \quad U^TU = I, \quad V^TV = I. \end{aligned} \tag{10}$$

The equivalence is formally given in the following theorem.

Theorem 2. *Models (9) and (10) are equivalent.*

Proof. To prove the theorem, we only need to show that $\|UCV^T\|_{ld} = \|C\|_{ld}$.

Let $C = \mathcal{U}\Sigma_C\mathcal{V}^T$ be the SVD of C. Then it is seen that $(U\mathcal{U})^T(U\mathcal{U}) = I_k$ and $(V\mathcal{V})^T(V\mathcal{V}) = I_k$. Hence, $(U\mathcal{U})\Sigma_C(V\mathcal{V})^T$ is the SVD of UCV^T. Thus UCV^T and C have identical singular values. By the definition of the log-determinant rank approximation function, it is straightforward that $\|UCV^T\|_{ld} = \|C\|_{ld}$. □

Up to now, we have proposed two models, including (7) and (10) for the cases where the knowledge of the precise true rank r is present or not, respectively. Accordingly, we name models (7) and (10) Fixed Rank FFP (F-FFP) and Unfixed Rank FFP (U-FFP). For the optimization, theoretical analysis, and experimental evaluation, we will present them in the following sections.

4. Optimization

In this section, we aim at developing the ALM-type algorithms for optimization of models (7) and (10), which will be separately discussed in the rest of this section.

4.1. Optimization of (7)

The augmented Lagrange function of (7) is

$$\begin{aligned} &\min_{S,U,C,V} \|S\|_1 + \tfrac{\rho}{2}\|X - UCV^T - S + \tfrac{1}{\rho}\Theta\|_F^2 \\ &s.t. \quad U^TU = I, V^TV = I. \end{aligned} \tag{11}$$

We will develop an alternating minimization strategy to iteratively optimize each variable while keeping all the others fixed. The detailed optimization strategy is presented as follows.

4.1.1. Optimization w.r.t. S

The subproblem of S is

$$\min_S \|S\|_1 + \frac{\rho}{2}\|X - UCV^T + \frac{1}{\rho}\Theta - S\|_F^2, \tag{12}$$

which can be efficiently solved with the shrinkage-thresholding operator provided in [30,31]. Denote $D = X - UCV^T + \frac{1}{\rho}\Theta$, then the optimal S is given element-wisely as

$$[S]_{ij} = (|D_{ij} - 1/\rho|)\mathrm{sgn}(D_{ij}), \tag{13}$$

where $\mathrm{sgn}(\cdot)$ returns the sign of the input.

4.1.2. Optimization w.r.t. V

Denote $M = X - S + \frac{1}{\rho}\Theta$, then V is to be optimized with the following problem:

$$\min_{V^TV=I} \|M^TU - V\|_F^2 \tag{14}$$

which is the classical Orthogonal Procrustes problem [32] and can be solved using the following lemma:

Lemma 1. *For the optimization problem*

$$\min_{V^TV=I} \|V - N\|_F^2, \tag{15}$$

the optimal V is defined as

$$V = PQ^T, \tag{16}$$

where P and Q are the left and right singular vectors of the thin SVD of N.

Now suppose that $N = M^TUC = P\Sigma_N Q^T$ is the thin SVD of N, then the optimal solution to (14) is

$$V = PQ^T. \tag{17}$$

4.1.3. Optimization w.r.t. U

The sub-problem of optimizing U is

$$\min_{U^TU=I} \|M - UV^T\|_F^2. \tag{18}$$

Similar to the minimization of V in (17), U is obtained with a closed-form solution

$$U = P'Q'^T, \tag{19}$$

where P' and Q' are left and right singular vectors of MVC^T, respectively. For the ease of notation, we define $\mathcal{P}(\cdot)$ and $\mathcal{Q}(\cdot)$ to be left and right singular vectors of the input matrix. Hence, V and U are updated by

$$V = \mathcal{P}(M^TUC)\mathcal{Q}^T(M^TUC) \tag{20}$$

$$U = \mathcal{P}(MVC^T)\mathcal{Q}^T(MVC^T). \tag{21}$$

4.1.4. Optimization of C

The sup-problem of optimizing C is

$$\min_C \frac{\rho}{2}\|X - UCV^T - S + \frac{1}{\rho}\Theta\|_F^2. \tag{22}$$

According to the first-order optimality condition, C is updated by

$$C = U^TMV. \tag{23}$$

4.1.5. Updating Θ *and* ρ

The updating of Θ and ρ are given as follows with an ALM-type procedure:

$$\begin{aligned} \Theta &= \Theta + \rho(X - UCV^T - S), \\ \rho &= \rho\kappa, \end{aligned} \tag{24}$$

where $\kappa > 1$ ensures that ρ is increasing.

4.2. Optimization of (10)

The augmented Lagrange function of (10) is

$$\begin{aligned} &\min_{S,U,V} \|S\|_1 + \lambda\|C\|_{ld} + \tfrac{\rho}{2}\|X - UCV^T - S + \tfrac{1}{\rho}\Theta\|_F^2 \\ &s.t. \quad U^TU = I, V^TV = I. \end{aligned} \tag{25}$$

It is noted that, other than C, the other sub-problems are identical to (11) and thus same updating rules in (16), (13), (19), (24) apply. In the rest of this subsection, we focus on the optimization of (10) with respect to C.

The sub-problem for optimizing C is as follows:

$$\tfrac{\rho}{2}\|M - UCV^T\|_F^2 + \lambda\|C\|_{ld} = \quad \tfrac{\rho}{2}\|U^TMV - C\|_F^2 + \lambda\|C\|_{ld}. \tag{26}$$

Similar to the theoretical derivations in [20], we can solve (26) by the following operator:

$$C = \mathcal{D}_{\frac{\lambda}{\rho}}(U^TMV), \tag{27}$$

where for a matrix D, $\mathcal{D}_\tau(D) = \mathcal{P}(D)\text{diag}\{\sigma_i^*\}(\mathcal{Q}(D))^T$, with

$$\sigma_i^* = \begin{cases} \xi, & \text{if } f_i(\xi) \le f_i(0) \text{ and } (1+\sigma_i(D))^2 > 4\tau, \\ 0, & \text{otherwise,} \end{cases} \tag{28}$$

where $f_i(x) = \frac{1}{2}(x - \sigma_i(D))^2 + \tau\log(1+x)$, and $\xi = \frac{\sigma_i(D)-1}{2} + \sqrt{\frac{(1+\sigma_i(D))^2}{2} - \tau}$.

Up to now, we have developed optimization procedures for (7), (10). For clearer representation, we summarize them in Algorithm 1.

Algorithm 1 F-FFP for Solving (7) (and, U-FFP for Solving (10)) .

Input: X, k, λ, ρ, κ, $t_{\max}$
Initialize: S_0, U_0, V_0, Θ_0, ρ_0, and $t = 1$.
repeat
  Update V_t by (20);
  Update U_t by (21);
  For F-FFP, update C_t by (23);
  For U-FFP, update C_t by (27);
  Update S_t by (13);
  Update Θ and ρ by (24);
  $t = t + 1$.
until $t \geq t_{\max}$ or convergence
Output: S_t, U_t, V_t, C_t

4.3. Complexity analysis

For F-FFP, the overall complexity for updating S, Θ, and ρ is $O(dnk)$ per iteration. The complexity for updating V and U is $O(dnk + nk^2)$ and $O(dnk)$ per iteration, respectively. For U-FFP, the complexity for updating U is $O(dk^2 + k^3)$ while the others are the same as F-FFP. Therefore, the overall complexities of F-FFP and U-FFP are $O(dnk + nk^2)$ and $O(dnk + nk^2 + dk^2 + k^3)$, respectively. When $k \ll \min(d, n)$, the complexity of F-FFP and U-FFP is reduced to $O(dnk)$. Hence, the proposed algorithms have linear complexity in both dimension and sample size, which is promising for real world applications.

5. Convergence analysis

In this section, we will provide theoretical analysis to show that U-FFP converges to a stationary point. The proof holds for F-FFP, where it can be regarded as a special case of U-FFP with $\lambda = 0$.

Theorem 3. *The sequences* $\{S_t\}$, $\{U_t\}$, $\{C_t\}$, $\{V_t\}$, *and* $\{\Theta_t\}$ *are bounded as long as* $\sum \frac{\rho_{t+1}}{\rho_t^2} < \infty$ *and* $\sum \frac{1}{\rho_t} < \infty$.

Proof. In the proof, we will provide the boundedness of each sequence one by one.

To minimize S at iteration $t+1$, S_{t+1} needs to satisfy the first-order optimality condition, that is,

$$\begin{aligned}&\nabla_S \mathcal{L}(S, U_{t+1}, C_{t+1}, V_{t+1}, \Theta_t, \rho_t)|_{S_{t+1}} \\ &= \nabla_S \|S\|_1|_{S_{t+1}} + \rho_t(S_{t+1} + U_{t+1}C_{t+1}V_{t+1}^T - X - \Theta_t/\rho_t) \\ &= 0.\end{aligned} \tag{29}$$

Note that the updating rule for Θ is

$$\Theta_{t+1} = \Theta_t + \rho_t(X - S_{t+1} - U_{t+1}C_{t+1}V_{t+1}^T), \tag{30}$$

hence, $\nabla_S \|S\|_1|_{S_{t+1}} - \Theta_{t+1} = 0$. Because $\|S\|_1$ is not smooth at some points, we define $\nabla_S \|S\|_1|_{S_{t+1}}$ to be its sub-gradient at point S_{t+1} in the following:

$$\left[\nabla_S \|S\|_1|_{S_{t+1}}\right]_{ij} \in \begin{cases}[-1, 1], & \text{if } [S_{t+1}]_{ij} = 0 \\ \operatorname{sgn}\left([S_{t+1}]_{ij}\right), & \text{otherwise.}\end{cases} \tag{31}$$

It is seen from the above definition that $\|\nabla_S \|S\|_1|_{S_{t+1}}\|_F^2 \leq dn$. The boundedness of $\nabla_S \|S\|_1|_{S_{t+1}}$ implies that Θ_{t+1} is also bounded. Then we may derive the following chain of equations:

$$\begin{aligned}&\mathcal{L}(S_t, U_t, C_t, V_t, \Theta_t, \rho_t) \\ &= \mathcal{L}(S_t, U_t, C_t, V_t, \Theta_{t-1}, \rho_{t-1}) + \tfrac{\rho_t}{2}\|X - U_tC_tV_t^T - S_t + \Theta_t/\rho_t\|_F^2 - \tfrac{\rho_{t-1}}{2}\|X - U_tC_tV_t^T - S_t + \Theta_{t-1}/\rho_{t-1}\|_F^2 \\ &= \mathcal{L}(S_t, U_t, C_t, V_t, \Theta_{t-1}, \rho_{t-1}) + \tfrac{\rho_t - \rho_{t-1}}{2}\|X - U_tC_tV_t^T - S_t\|_F^2 + \operatorname{Tr}((\Theta_t - \Theta_{t-1})^T(X - U_tC_tV_t^T - S_t)) \\ &\quad + \tfrac{1}{2\rho_t}\|\Theta_t\|_F^2 - \tfrac{1}{2\rho_{t-1}}\|\Theta_{t-1}\|_F^2 \\ &= \mathcal{L}(S_t, U_t, C_t, V_t, \Theta_{t-1}, \rho_{t-1}) + \tfrac{\rho_t + \rho_{t-1}}{2\rho_{t-1}^2}\|\Theta_t - \Theta_{t-1}\|_F^2 + \tfrac{1}{2\rho_t}\|\Theta_t\|_F^2 - \tfrac{1}{2\rho_{t-1}}\|\Theta_{t-1}\|_F^2.\end{aligned} \tag{32}$$

Here, the last equation is obtained by the updating of Θ_t. By the alternatively optimizating the objective w.r.t V, C, U, and S, we have

$$\begin{aligned}\mathcal{L}(S_{t+1},U_{t+1},C_{t+1},V_{t+1},\Theta_t,\rho_t) &\le \mathcal{L}(S_t,U_t,C_t,V_t,\Theta_{t-1},\rho_{t-1})\\ &\quad+\tfrac{\rho_t+\rho_{t-1}}{2\rho_{t-1}^2}\|\Theta_t-\Theta_{t-1}\|_F^2+\tfrac{1}{2\rho_t}\|\Theta_t\|_F^2-\tfrac{1}{2\rho_{t-1}}\|\Theta_{t-1}\|_F^2\\ &\le \mathcal{L}(S_t,U_t,C_t,V_t,\Theta_{t-1},\rho_{t-1})+\tfrac{\rho_t+\rho_{t-1}}{2\rho_{t-1}^2}\|\Theta_t-\Theta_{t-1}\|_F^2+\tfrac{1}{2\rho_t}\|\Theta_t\|_F^2.\end{aligned} \tag{33}$$

Iterating the inequality t times in a way similar to (33), we arrive at:

$$\begin{aligned}&\mathcal{L}(S_{t+1},U_{t+1},C_{t+1},V_{t+1},\Theta_t,\rho_t)\\ &\le \mathcal{L}(S_1,U_1,C_1,V_1,\Theta_0,\rho_0)+\textstyle\sum_{i=1}^t\frac{\rho_t+\rho_{t-1}}{2\rho_{t-1}^2}\|\Theta_t-\Theta_{t-1}\|_F^2+\sum_{i=1}^t\frac{1}{2\rho_t}\|\Theta_t\|_F^2\\ &\le \mathcal{L}(S_1,U_1,C_1,V_1,\Theta_0,\rho_0)+\Gamma_1\textstyle\sum_{i=1}^t\frac{\rho_t+\rho_{t-1}}{2\rho_{t-1}^2}+\Gamma_2\sum_{i=1}^t\frac{1}{2\rho_t},\end{aligned} \tag{34}$$

where Γ_1 is an upper bound of $\{\|\Theta_i-\Theta_{i-1}\|_F^2\}$, and Γ_2 is an upper bound of $\{\|\Theta_i\|_F^2\}$. Because $\{\Theta_i\}$ is bounded, $\|\Theta_i\|_F^2$ and $\|\Theta_i-\Theta_{i-1}\|_F^2\le 2\|\Theta_i\|_F^2+2\|\Theta_{i-1}\|_F^2$ are bounded. Thus, the existences of Γ_1 and Γ_2 are guaranteed, e.g., $\Gamma_2=\max\{\|\Theta_i\|_F^2\}$ and $\Gamma_1=4\Gamma_2$. Therefore, under the conditions that $\sum\frac{\rho_{t+1}}{\rho_t^2}<\infty$ and $\sum\frac{1}{\rho_t}<\infty$, it is clear that

$$\mathcal{L}(S_{t+1},U_{t+1},C_{t+1},V_{t+1},\Theta_t,\rho_t)=\|S_{t+1}\|_1+\|C_{t+1}\|_{Id}+\frac{\rho_t}{2}\|X-S_{t+1}-U_{t+1}C_{t+1}V_{t+1}^T+\Theta_t/\rho_t\|_F^2 \tag{35}$$

is bounded. Then each term on the right hand side of the above equation is bounded, which implies that S_{t+1} and C_{t+1} are bounded. According to the updating rules of V and U in (20), (21), $U^TU=V^TV=I$ always hold, where I is an identity matrix. Therefore, U and V are also bounded. Hereby, we have proved that $\{S_t\}$, $\{U_t\}$, $\{C_t\}$, $\{V_t\}$, and $\{\Theta_t\}$ are bounded. □

Theorem 4. *Let $\{S_t, U_t, C_t, V_t, \Theta_t\}$ be the sequence generated by Algorithm 1. Under the assumptions that $\sum\frac{\rho_{t+1}}{\rho_t^2}<\infty$, $\sum\frac{1}{\rho_t}<\infty$, and $\rho_t(C_t-C_{t+1})\to 0$, sequence of $\{S_t, U_t, C_t, V_t, \Theta_t\}$ has at least one accumulation point. For any accumulation point $\{S^*, U^*, C^*, V^*, \Theta^*\}$, $\{S^*, U^*, C^*, V^*\}$ is a stationary point of optimization problem (10).*

Proof. Under the conditions that $\sum\frac{\rho_{t+1}}{\rho_t^2}<\infty$ and $\sum\frac{1}{\rho_t}<\infty$, by Theorem 3, we know that $\{S_t, U_t, C_t, V_t, \Theta_t\}$ is bounded. By the Bolzano–Weierstrass theorem, the sequence must have at least one accumulation point, namely, $\{S^*, U^*, C^*, V^*, \Theta^*\}$. Without loss of generality, we assume that $\{S_t, U_t, C_t, V_t, \Theta_t\}$ itself converges to $\{S^*, U^*, C^*, V^*, \Theta^*\}$. Next, we prove that $\{S^*, U^*, C^*, V^*\}$ is a stationary point of the problem (10). As

$$\Theta_{t+1}=\Theta_t+\rho_t(X-S_{t+1}-U_{t+1}C_{t+1}V_{t+1}^T),$$

we have

$$X-S_{t+1}-U_{t+1}C_{t+1}V_{t+1}^T=\frac{1}{\rho_t}(\Theta_{t+1}-\Theta_t).$$

Because $\rho_t\to\infty$ and Θ_t is bounded,

$$X-S_{t+1}-U_{t+1}C_{t+1}V_{t+1}^T\to 0,$$

i.e.,

$$X-S^*-U^*C^*(V^*)^T=0.$$

Besides, based on the updating of V and U, $V_t^TV_t=U_t^TU_t=I$ always hold. Therefore,

$$(V^*)^TV^*=(U^*)^TU^*=I.$$

Hence, the primal feasibility conditions are satisfied by S^*, U^*, C^*, and V^*. Next, we will show that the stationary conditions also hold. By the first-order optimality condition of S_t, we have

$$\nabla_S\|S\|_1|_{S_t}+\rho_{t-1}(S_t+U_tC_tV_t^T-X-\Theta_{t-1}/\rho_{t-1})=\nabla_S\|S\|_1|_{S_t}-\Theta_t=0.$$

Let $t\to\infty$, we get

$$\nabla_S\|S\|_1|_{S^*}-\Theta^*=0.$$

By the first-order optimality condition of C, we have

$$\begin{aligned}&\nabla_C\|C\|_{Id}|_{C_{t+1}}+\rho_tU_{t+1}^T(S_t+U_{t+1}C_{t+1}V_{t+1}^T-X-\Theta_t/\rho_t)V_{t+1}\\ &=\nabla_C\|C\|_{Id}|_{C_{t+1}}+\rho_tU_{t+1}^T(S_t-S_{t+1})V_{t+1}+\rho_tU_{t+1}^T(S_{t+1}+U_{t+1}C_tV_{t+1}^T-X-\Theta_t/\rho_t)V_{t+1}\\ &=\nabla_C\|C\|_{Id}|_{C_{t+1}}+\rho_tU_{t+1}^T(S_t-S_{t+1})V_{t+1}-U_{t+1}^T(\Theta_t+\rho_t(X-S_{t+1}-U_{t+1}C_{t+1}V_{t+1}^T))V_{t+1}\\ &=\nabla_C\|C\|_{Id}|_{C_{t+1}}+\rho_tU_{t+1}^T(S_t-S_{t+1})V_{t+1}-U_{t+1}^T\Theta_{t+1}V_{t+1}=0.\end{aligned}$$

Table 1
Benchmarking datasets.

Data set	Data size	Used size	# of backgrounds
Escalator Airport	$130 \times 160 \times 3{,}417$	$130 \times 160 \times 3{,}417$	1
Hall Airport	$144 \times 176 \times 3{,}584$	$144 \times 176 \times 2{,}389$	1
Bootstrap	$120 \times 160 \times 2{,}055$	$120 \times 160 \times 2{,}055$	1
Campus	$128 \times 160 \times 1{,}439$	$128 \times 160 \times 1{,}439$	1
Fountain	$128 \times 160 \times 0{,}523$	$128 \times 160 \times 0{,}523$	1
Water Surface	$128 \times 160 \times 0{,}633$	$128 \times 160 \times 0{,}633$	1
Shopping Mall	$256 \times 320 \times 1{,}286$	$128 \times 160 \times 1{,}286$	1
Curtain	$128 \times 160 \times 2{,}964$	$128 \times 160 \times 2{,}964$	1
Office	$128 \times 160 \times 2{,}964$	$128 \times 160 \times 2{,}964$	1
PETS2006	$576 \times 720 \times 2{,}964$	$128 \times 160 \times 2{,}964$	1
Pedestrian	$240 \times 360 \times 1{,}099$	$120 \times 180 \times 1{,}099$	1
Highway	$240 \times 320 \times 1{,}700$	$120 \times 160 \times 1{,}700$	1
Lobby	$128 \times 160 \times 1{,}546$	$128 \times 160 \times 1{,}546$	2
Camera Parameter	$240 \times 320 \times 5{,}001$	$120 \times 160 \times 2{,}501$	2
Light Switch-1	$120 \times 160 \times 2{,}800$	$120 \times 160 \times 2{,}800$	2
Light Switch-2	$120 \times 160 \times 2{,}715$	$120 \times 160 \times 2{,}715$	2
Time Of Day	$128 \times 160 \times 5{,}890$	$128 \times 160 \times 1{,}964$	2

For data size, the first two dimensions represent the size of each frame while the third represents the number of frames.

Under the assumption that $\rho_t(S_t - S_{t+1}) \to 0$, we have

$$\rho_t U_{t+1}^T (S_t - S_{t+1})) V_{t+1} \to 0.$$

Let $t \to \infty$, then we have

$$\nabla_C \|C\|_{ld}|_{C^*} - (U^*)^T \Theta^* V^* = 0.$$

Now we can see that $\{S^*, U^*, C^*, V^*, \Theta^*\}$ satisfies the KKT conditions of $\mathcal{L}$; therefore $\{S^*, U^*, C^*, V^*\}$ is a stationary point of the problem (10). □

6. Experiments

In this section, we will conduct extensive experiments to empirically evaluate the proposed method. In particular, three crucial applications are considered, including foreground-background separation in video sequences, shadow removal from face images, and anomaly detection from pen digits. Due to elegant theory as well as the guaranteed performance of IALM[1] [8] and linear efficiency of AltProj,[2] these methods are used as a benchmark to illustrate the effectiveness and efficiency of our algorithm. To improve the efficiency of IALM and AltProj, we make use of the PROPACK package [12] to solve SVDs. We conduct all experiments on a dual-core Intel Xeon E3-1240 V2 3.40 GHz Linux Server with 8 GB memory. For purpose of reproduction, we provide our code at https://www.researchgate.net/publication/316656069_codes_icdm2016.

6.1. Foreground-background separation

A video sequence can be decomposed into a background (the low-rank part) and a foreground (the sparse part). The problem of foreground-background separation is to detect moving objects or interesting activities in a scene and remove background(s) from a video sequence. To testify the proposed method on this application we use 17 benchmark data sets, among which 12 contain a single background while 5 have 2 backgrounds.[3] We summarize some key characteristics of these data sets in Table 1. It should be noted that the number of backgrounds summarized in Table 1 reveals the underlying ground true rank of L. Due to a computational acceleration consideration, we perform down-sampling or down-resolution on some of these data sets, for which the rates can be found in Table 1. For each data set, we construct a data matrix by vectorizing and collecting all frame images. In the following, we will consider two cases based on whether the knowledge of precise r value is available or not.

6.1.1. Case 1 (r is known)

When r is available, we set $k = r$ for F-FFP and AltProj, where the value of r for each data set can be found in Table 1. For the parameter settings, we fix $\rho = 0.0001$ and $\kappa = 1.5$ for IALM for fast convergence and fairly good visual quality, which

[1] http://perception.csl.illinois.edu/matrix-rank/sample_code.html#RPCA.
[2] http://www.personal.psu.edu/nsa10/codes.html.
[3] The datasets used in this subsection can be found at http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html
http://limu.ait.kyushu-u.ac.jp/dataset/en/
http://wordpress-jodoin.dmi.usherb.ca/dataset2012/
http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm.

Table 2
Results over different datasets with r known.

Data	Method	Rank(L)	$\|S\|_0/(dn)$	$\frac{\|X-L-S\|_F}{\|X\|_F}$	# of Iter.	# of SVDs	Time
Shopping	AltProj	1	0.9853	3.91e-5	45	46	45.35
Mall	IALM	328	0.8158	9.37e-4	11	12	123.99
	F-FFP	1	0.9122	7.72e-4	23	23	11.65
PETS2006	AltProj	1	0.8590	5.20e-4	35	36	44.64
	IALM	293	0.8649	5.63e-4	12	13	144.26
	F-FFP	1	0.8675	5.71e-4	24	24	14.09
Hall Airport	AltProj	1	0.9806	5.16e-5	46	47	113.98
	IALM	556	0.8500	8.15e-4	11	12	408.03
	F-FFP	1	0.9168	5.89e-4	23	23	26.19
Campus	AltProj	1	0.9790	9.50e-5	41	42	54.1
	IALM	488	0.8136	9.30e-4	11	12	242.59
	F-FFP	1	0.9378	6.28e-4	23	23	12.85
Pedestrian	AltProj	1	0.5869	9.32e-4	41	42	37.90
	IALM	35	0.8910	5.69e-4	11	12	36.18
	F-FFP	1	0.6719	6.03e-4	23	23	10.53
Water	AltProj	1	0.8890	3.97e-4	47	48	27.27
Surface	IALM	224	0.7861	5.32e-4	12	13	51.00
	F-FFP	1	0.8355	9.91e-4	23	23	5.68
Curtain	AltProj	1	0.8280	7.46e-4	40	41	102.41
	IALM	834	0.7398	6.84e-4	12	13	747.36
	F-FFP	1	0.8680	6.31e-4	24	24	27.51
Fountain	AltProj	1	0.9113	2.91e-4	50	51	23.90
	IALM	102	0.8272	4.91e-4	12	13	25.62
	F-FFP	1	0.8854	5.10e-4	24	24	5.00
Office	AltProj	1	0.8018	9.40e-4	51	52	84.43
	IALM	374	0.7582	9.46e-4	11	12	230.53
	F-FFP	1	0.8761	5.42e-4	24	24	19.92
Highway	AltProj	1	0.9331	2.96e-4	37	38	49.65
	IALM	539	0.8175	6.02e-4	12	13	269.10
	F-FFP	1	0.8854	5.75e-4	24	24	14.83
Bootstrap	AltProj	1	0.9747	1.17e-4	44	45	107.15
	IALM	1146	0.8095	6.27e-4	12	13	1182.92
	F-FFP	1	0.9288	7.72e-4	23	23	25.38
Escalator	AltProj	1	0.9152	2.29e-4	40	41	110.75
Airport	IALM	957	0.7744	7.76e-4	11	12	1,040.91
	F-FFP	1	0.8878	5.68e-4	23	23	30.78

For IALM and AltProj, (partial) SVDs are for $d \times n$ matrices. For F-FFP, SVDs are for $n \times k$ matrices, which are computationally far less expensive than those required by IALM and AltProj.

remains the same for F-FFP for fair comparison. For IALM, we use the theoretically optimal value of the balance parameter, which is provided in the original paper [8]. For AltProj, the default parameters are used, where the precise value of r is used as input. All algorithms are terminated when the condition of $\frac{\|X-L-S\|_F}{\|X\|_F} \leq 10^{-3}$ is satisfied or a maximum number of 200 iterations is reached. Unless specified, the parameter settings remain the same throughout this paper.

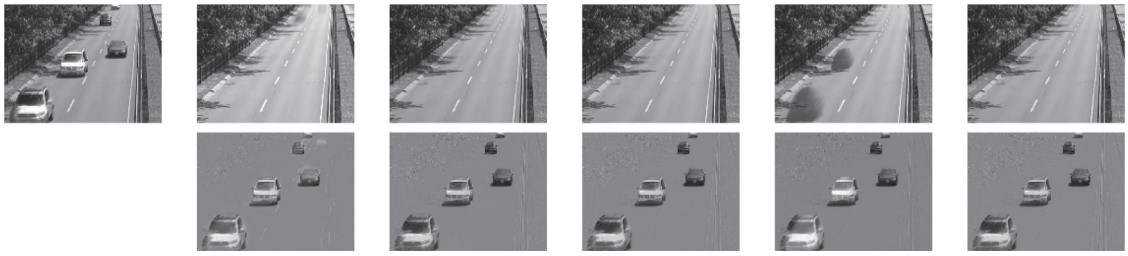
We perform all algorithms on data sets in Table 1 and report the numerical results in Tables 2 and 3. It is observed that, in general, IALM obtains more sparse S, but fails to recover L with a desired low rank. Meanwhile, both F-FFP and AltProj recover L properly with the true rank. Moreover, a general observation by comparing AltProj and F-FFP is that the later generates more sparse S than the former. In terms of the fitting error, it is observed that all methods have competitive performance. However, it should be noted that the F-FFP needs the shortest time and thus the fitting error can be further reduced if comparable time to other methods or more iterations are allowed. It is notable that F-FFP needs the least amount of time on all these data sets. Roughly, F-FFP is 4 times faster than AltProj and more than 10 times faster than IALM. To better illustrate the effectiveness of the proposed method, we show some visual results in the first four columns in Figs. 2–4. It is observed that the backgrounds recovered by IALM still have some residues from the moving foregrounds; for example, some residues of cars in the top area of the highway, people standing at the top area of the escalator, and people sitting on the chair are clearly perceived in these figures. These observations explain why L produced by IALM has high ranks. As for AltProj and F-FFP, their results are visually comparable, where they can well separate clean backgrounds from the moving foregrounds.

*6.1.2. Case 2 (*r *is unknown)*

When r is unavailable, we may specify a proper k as an upper bound of r based on domain knowledge. In this test, we set $k = 5$ throughout this subsection. In this paper, we provide an empirical approach to estimate a proper k, which can be found in remark of this subsection. Since we do not need to specify the value of k for IALM, the performance of IALM remains the same as in Section 6.1.1. For U-FFP, λ is chosen from a set of values {100, 200, 300, 400, 500}. We show the numerical as well as some visual results in Table 4 and the last two columns of Figs. 2–4, respectively. It is observed

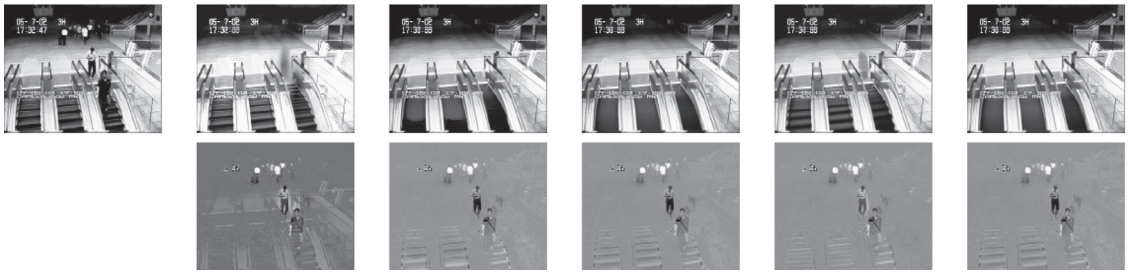
Table 3
Results over different datasets with r known.

Data	Method	Rank(L)	$\|S\|_0/(dn)$	$\frac{\|X-L-S\|_F}{\|X\|_F}$	# of Iter.	# of SVDs	Time
Lobby	AltProj	2	0.9243	1.88e-4	39	41	47.32
	IALM	223	0.8346	6.19e-4	12	13	152.54
	F-FFP	2	0.8524	6.53e-4	24	24	15.20
Light Switch-1	AltProj	2	0.9600	1.12e-4	55	57	121.27
	IALM	499	0.6737	9.99e-4	11	12	359.40
	F-FFP	2	0.8829	8.10e-4	23	23	23.81
Light Switch-2	AltProj	2	0.9050	2.24e-4	47	49	87.35
	IALM	591	0.7921	7.93e-4	12	13	613.98
	F-FFP	2	0.8324	7.71e-4	24	24	24.12
Camera Parameter	AltProj	2	0.8806	5.34e-4	47	49	84.99
	IALM	607	0.7750	6.86e-4	12	13	433.47
	F-FFP	2	0.8687	6.26e-4	24	24	22.25
Time Of Day	AltProj	2	0.8646	4.72e-4	44	46	61.63
	IALM	351	0.6990	6.12e-4	13	14	265.87
	F-FFP	2	0.8441	6.82e-4	25	25	18.49



(a) Original (b) IALM (c) AltProj (d) F-FFP (e) AltProj (k=5) (f) U-FFP

Fig. 2. Foreground-background separation in the Highway video. The top left is the original frame and the rest are extracted background (top) and foreground (bottom).



(a) Original (b) IALM (c) AltProj (d) F-FFP (d) AltProj (k=5) (f) U-FFP

Fig. 3. Foreground-background separation in the Escalator Airport video. The top left is the original frame and the rest are extracted background (top) and foreground (bottom).

that U-FFP recovers L with the true rank, whereas AltProj fails. Besides, the time cost of U-FFP increases slightly by less than 1 second on most of the datasets while AltProj needs about another 10–20 s. Visually, it is seen that AltProj fails to separate the backgrounds from the foregrounds. In the backgrounds, we can still observe the residues of the moving objects. For U-FFP, we observe quite comparable visual results to F-FFP. These observations illustrate the enhanced efficiency of the proposed method as well as alleviated dependence on the knowledge of r.

Remark. In this paper, we will provide a simple yet effective way to estimate a proper k in the following. It is natural that low-rank matrices have a small number of dominant singular values. Based on this observation, it is natural to set k as $\bar{k}$ if the top $\bar{k}$ largest singular values are significantly larger than the rest ones. To illustrate this, we show some examples in Fig. 5. It is evident that singular values of Airport and Lobby data sets significantly decrease from the second and third ones, respectively. This suggests possible upper bounds for Airport and Lobby data sets to be 1 and 2, respectively. It should be noted that the singular values can be computed in a greedy way, i.e., we calculate the largest one among the unknown ones until $\bar{k}$ appears. This approach only requires $O(dn\bar{k}) = O(dnk)$ complexity, implying that it is scalable to estimate a proper r in real world applications.

Fig. 4. Foreground-background separation in the Light Switch-2 video. The top and bottom two panels correspond to two frames, respectively. For each frame, the top left is the original image while the rest are the extracted background (top) and foreground (bottom), respectively.

Table 4
Results over datasets with r unknown.

Data	Method	Rank(L)	$\|S\|_0/(dn)$	$\frac{\|X-L-S\|_F}{\|X\|_F}$	# of Iter.	# of SVDs	Time
Shopping Mall	AltProj	5	0.9611	9.82e-5	41	46	63.34
	U-FFP	1	0.9120	7.80e-4	23	23+23	12.14
PETS2006	AltProj	5	0.8543	6.15e-4	39	43	63.33
	U-FFP	1	0.8681	5.70e-4	24	24+24	14.79
Hall Airport	AltProj	5	0.8960	5.12e-5	45	50	130.59
	U-FFP	1	0.9168	5.91e-4	23	23+23	27.24
Campus	AltProj	5	0.9482	3.18e-5	46	51	92.90
	U-FFP	1	0.9376	6.29e-4	23	23+23	14.02
Pedestrian	AltProj	5	0.6202	6.37e-4	44	49	58.10
	U-FFP	1	0.6740	6.05e-4	23	23+23	10.92
Water Surface	AltProj	5	0.9090	2.38e-4	46	50	33.78
	U-FFP	1	0.9368	9.89e-4	23	23+23	6.15
Curtain	AltProj	5	0.8079	8.82e-4	36	39	101.79
	U-FFP	1	0.8684	6.30e-4	24	24+24	29.11
Fountain	AltProj	5	0.7435	7.55e-4	48	52	32.24
	U-FFP	1	0.8873	5.29e-4	24	24+24	5.26
Office	AltProj	5	0.7159	8.61e-4	47	52	98.54
	U-FFP	1	0.8764	5.41e-4	24	24+24	21.15
Highway	AltProj	5	0.9007	3.66e-4	43	48	75.60
	U-FFP	1	0.8862	5.75e-4	24	24+24	15.39
Bootstrap	AltProj	5	0.9875	3.02e-4	47	52	169.06
	U-FFP	1	0.9298	7.67e-4	23	23+23	26.19
Escalator Airport	AltProj	5	0.8474	8.43e-4	43	48	162.49
	U-FFP	1	0.8876	5.70e-4	23	23+23	31.95
Lobby	AltProj	5	0.9176	1.71e-4	40	44	61.50
	U-FFP	2	0.9197	6.24e-4	25	25+25	15.85
Light Switch-1	AltProj	5	0.8474	4.29e-4	43	47	105.24
	U-FFP	2	0.9172	6.37e-4	24	24+24	25.01
Light Switch-2	AltProj	5	0.8507	4.29e-4	37	41	80.37
	U-FFP	2	0.8324	7.75e-4	24	24+24	24.70
Camera Parameter	AltProj	5	0.7311	8.34e-4	50	55	147.28
	U-FFP	2	0.8521	7.09e-4	24	24+24	22.67
Time Of Day	AltProj	5	0.8651	4.61e-4	46	51	73.35
	U-FFP	2	0.8880	8.05e-4	25	25+25	18.35

For AltProj, (partial) SVDs are performed on $d \times n$ matrices. For U-FFP, SVDs are for both $d \times k$ and $n \times k$ matrices, which are computationally far less expensive than those required by AltProj.

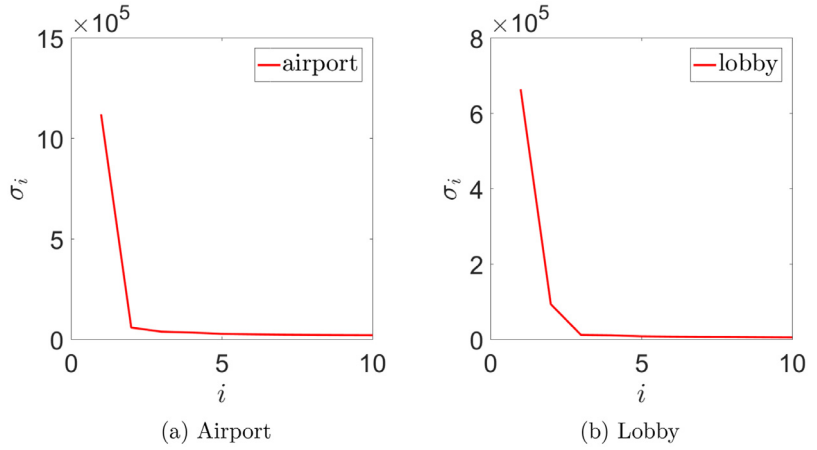


Fig. 5. Plots of the largest 10 singular values of Airport and Lobby data sets.

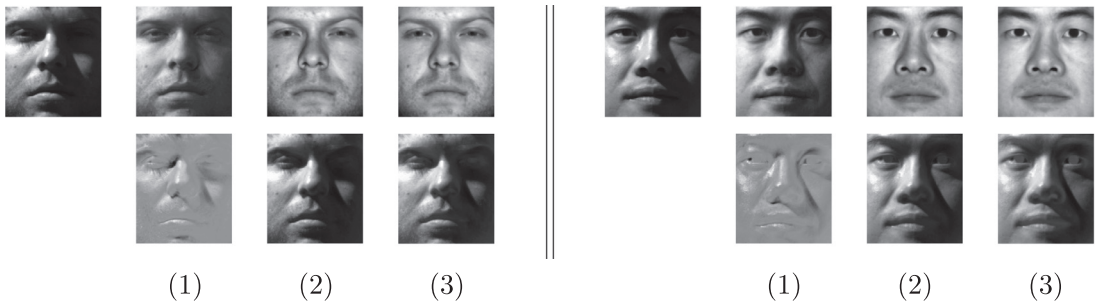


Fig. 6. Shadow removal results for subjects 1 and 2 from EYaleB data. For each of the two parts, the top left is the original image and the rest are recovered clean images (top) and shadows (bottom) by (1) IALM, (2) AltProj, and (3) F-FFP, respectively.

6.2. Shadow removal from face images

Face recognition is an important topic in pattern recognition community [33–35]; however, face images taken under various lighting conditions may introduce heavy shadows into face images, which makes it challenging to learn patterns [36]. Hence, to improve the learning capability on face image data and improve recognition accuracy, it is crucial to handle shadows, peculiarities and saturations on face images. Face images can be naturally separated into a low-rank part and a sparse part: Clean images reside in a low-rank subspace while shadows correspond to sparse components. Due to such natures of images, RPCA has been shown to successfully address such a challenging task as shadow removal. For this problem, we follow [8] and use the Extended Yale B (EYaleB) data set [37]. Without loss of generality, we choose the first two persons out of 38 individuals from this data set and treat each person as a subject. For each subject, there are 64 heavily corrupted face images taken under varying lighting conditions. Each image has a size of 192×168 pixels and we vectorize and record it as a column in a 32, 256×64 data matrix. Since each data matrix collects face images from a single person, it is reasonable to assume these images reside in the same rank-1 subspace and thus we set the underlying true rank of the data matrix to be $r = 1$.

Similar to the previous subsection, we consider two cases in this test. First, we consider the case where r is known. We follow the strategy in [8] and apply IALM, AltProj, and F-FFP to each subject. We show the quantitative and visual results in Table 5 and Fig. 6, respectively. From Fig. 6, we can observe that AltProj and F-FFP can successfully remove shadows and recover clean face images while IALM fails in this task. It is seen that although the majority of shadows are removed by

Table 5
Recovery results of face data with $k = 1$.

Data	Method	Rank(Z)	$\|S\|_0/(dn)$	$\frac{\|X-Z-S\|_F}{\|X\|_F}$	# of Iter.	# of SVDs	Time
Subject 1	AltProj	1	0.9553	8.18e-4	50	51	4.62
	IALM	32	0.7745	6.28e-4	25	26	2.43
	F-FFP	1	0.9655	8.86e-4	36	36	1.37
Subject 2	AltProj	1	0.9755	2.34e-4	49	50	5.00
	IALM	31	0.7656	6.47e-4	25	26	2.66
	F-FFP	1	0.9492	9.48e-4	36	36	1.37

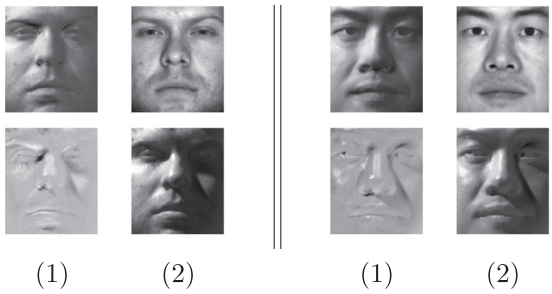


Fig. 7. Shadow removal results for subjects 1 and 2 from EYaleB data. The top panel are the recovered clean images and the bottom panel are the shadows by (1) AltProj (k=5) and (2) U-FFP, respectively.

IALM, some still remain. Quantitatively, we can see that both AltProj and F-FFP have the exact rank recovery, where the recovered low-rank part has rank 1, while IALM recovers the low-rank part with a higher rank. It is notable that though F-FFP has comparable performance to AltProj, the former is about 4 times faster than the latter. These observations have verified the effectiveness of F-FFP for face shadow removal.

Next, we consider the case where r is unknown. In a way similar to the setting in foreground-background separation, we set $k = 5$. We apply AltProj and U-FFP to each subject and report the quantitative as well as visual results in Table 6 and Fig. 7, respectively. From Table 6, it is observed that visually U-FFP is comparable to F-FFP while AltProj appears unable to remove shadows neatly. Quantitatively, as shown in Table 6, AltProj generates an L that has a higher rank while U-FFP produces an L that has the true rank. These observations imply that U-FFP is allowed to have more flexibility in choosing k. Besides, U-FFP is the fastest among all these methods, suggesting its potential in real world application.

6.3. Face clustering

To further exploit the effectiveness of the proposed method on face shadow removal, we expand the experiment to face clustering. In this test, we use the EYaleB data. We follow the strategy in [38] to apply RPCA as a pre-processing step to obtain the low-rank part of the data on which we perform standard clustering methods. Here, without loss of generality, we follow the previous subsection and set $k = 5$. It is natural to believe that better clustering performance implies more effective removal of the shadow from face images. To better investigate the clustering performance, in clustering stage, we follow the strategy in [20,38,39] and divide the 38 individuals into 4 groups, containing individuals 1–10, 11–20, 21–30, and 31–38, respectively. Then within each group, we consider all possible combinations of n persons, where n takes value within {2, 3, 5, 8, 10}. For each n value, all combinations across the groups are collected to obtain a collection of face images, which contain all possible subsets of n persons. Finally, we apply clustering methods, including K-means, spectral clustering, and hierarchical clustering, to all component subsets within each collection of face images and we record the mean and median results for each method. For K-means, we use a fast implementation [40] and repeat it 100 times for the smallest objective. For spectral clustering, we use RBF kernel [41] with radius parameter ranges in the set {0.001, 0.01, 0.1, 1, 10, 100, 1000}. For hierarchical clustering, we use the complete-linkage approach. For the evaluation metric, we use clustering accuracy, normalized mutual information, and purity, whose detailed descriptions can be found in [42]. In Tables 7–9, we report the best mean and median performance with respect to each collection. It is observed that these clustering methods achieve the best performance on data recovered by the proposed method throughout this test, which implies that the proposed method can more effectively remove the shadow from face images.

6.4. Anomaly detection

Given a number of images from one subject, they form a low-dimensional subspace. Any image that significantly differs from the majority of the images can be regarded as an outlier; besides, fewer images from another subject can be regarded as outliers. Anomaly detection is to identify such kinds of outliers. USPS dataset contains 9298 images of hand-written digits of size 16×16. Following [13,21], among these images, we select the first 190 images of '1's and the last 10 of '7's and construct a data matrix of size 256×200 by regarding each vectorized image as a column. Since there are much more '1's than '7's, we treat the '1's as dominant while the '7's as outliers. It is straightforward to point out that the true rank

Table 6
Recovery results of face data with $k = 5$.

Data	Method	Rank(Z)	$\|S\|_0/(dn)$	$\frac{\|X-Z-S\|_F}{\|X\|_F}$	# of Iter.	# of SVDs	Time
Subject 1	AltProj	5	0.9309	3.93e-4	51	55	6.08
	U-FFP	5	0.9632	9.01e-4	36	36+36	1.44
Subject 2	AltProj	5	0.8903	6.40e-4	54	58	7.92
	U-FFP	1	0.9645	5.85e-4	37	37+37	1.53

Table 7
Comparison of clustering in accuracy.

No. of Subjects		2 Subjects		3 Subjects		5 Subjects		8 Subjects		10 Subjects	
Algorithm		Average	Median	Average	Median	Average	Median	Average	Median	Average	Median
IALM	+K-means	0.5152	0.5156	0.3597	0.3594	0.2483	0.2375	0.1956	0.1953	0.1698	0.1672
AltProj		0.5191	0.5156	0.3655	0.3594	0.2572	0.2437	0.2026	0.2031	0.1859	0.1953
U-FFP		0.9989	1.0000	0.9990	1.0000	0.9975	1.0000	0.9829	1.0000	0.9995	1.0000
IALM	+Spectral	0.5151	0.5156	0.3616	0.3594	0.2303	0.2281	0.1546	0.1543	0.1323	0.1328
AltProj		0.5204	0.5156	0.3656	0.3646	0.2331	0.2344	0.1565	0.1563	0.1318	0.1313
U-FFP		0.9929	1.0000	0.9384	1.0000	0.9907	1.0000	0.9913	1.0000	0.9661	1.0000
IALM	+Hierarichical	0.5210	0.5078	0.3613	0.3438	0.2349	0.2094	0.1669	0.1367	0.1432	0.1109
AltProj		0.6015	0.5078	0.4445	0.3438	0.2945	0.2125	0.2060	0.2539	0.1740	0.2062
U-FFP		0.9335	1.0000	0.9426	1.0000	0.9520	1.0000	0.9645	1.0000	0.9661	1.0000

Table 8
Comparison of clustering in normalized mutual information.

No. of Subjects		2 Subjects		3 Subjects		5 Subjects		8 Subjects		10 Subjects	
Algorithm		Average	Median	Average	Median	Average	Median	Average	Median	Average	Median
IALM	+K-means	0.0011	0.0007	0.0037	0.0023	0.0319	0.0108	0.0734	0.0765	0.0754	0.0800
AltProj		0.0018	0.0007	0.0053	0.0031	0.0396	0.0131	0.0724	0.0725	0.0840	0.0998
U-FFP		0.9921	1.0000	0.9954	1.0000	0.9962	1.0000	0.9883	1.0000	0.9988	1.0000
IALM	+Spectral	0.0011	0.0007	0.0039	0.0029	0.0089	0.0060	0.0134	0.0107	0.0200	0.0173
AltProj		0.0020	0.0007	0.0049	0.0038	0.0074	0.0060	0.0125	0.0100	0.0158	0.0160
U-FFP		0.9803	1.0000	0.9213	1.0000	0.9908	1.0000	0.9927	1.0000	0.9799	1.0000
IALM	+Hierarichical	0.0326	0.0079	0.0422	0.0105	0.0516	0.0126	0.0605	0.0157	0.0617	0.0157
AltProj		0.1966	0.0079	0.1866	0.0105	0.1455	0.0126	0.1184	0.1885	0.1059	0.1517
U-FFP		0.8660	1.0000	0.9275	1.0000	0.9582	1.0000	0.9765	1.0000	0.9799	1.0000

Table 9
Comparison of clustering in purity.

No. of Subjects		2 Subjects		3 Subjects		5 Subjects		8 Subjects		10 Subjects	
Algorithm		Average	Median	Average	Median	Average	Median	Average	Median	Average	Median
IALM	+K-means	0.5152	0.5156	0.3611	0.3594	0.2545	0.2453	0.2107	0.2129	0.1844	0.1812
AltProj		0.5191	0.5156	0.3676	0.3646	0.2617	0.2500	0.2121	0.2109	0.2031	0.2125
U-FFP		0.9989	1.0000	0.9990	1.0000	0.9976	1.0000	0.9832	1.0000	0.9995	1.0000
IALM	+Spectral	0.5151	0.5156	0.3641	0.3594	0.2363	0.2344	0.1612	0.1621	0.1385	0.1391
AltProj		0.5204	0.5156	0.3678	0.3646	0.2382	0.2375	0.1626	0.1641	0.1365	0.1359
U-FFP		0.9929	1.0000	0.9392	1.0000	0.9910	1.0000	0.9904	1.0000	0.9667	1.0000
IALM	+Hierarichical	0.5210	0.5078	0.3629	0.3438	0.2379	0.2125	0.1711	0.1406	0.1479	0.1156
AltProj		0.6015	0.5078	0.4461	0.3438	0.2982	0.2125	0.2108	0.2578	0.1786	0.2109
U-FFP		0.9335	1.0000	0.9428	1.0000	0.9526	1.0000	0.9651	1.0000	0.9667	1.0000

of L should be 1. Some examples of these selected images are shown in Fig. 8. It is observed that besides the '7's, some '1's are quite different from the majority. Therefore, anomaly detection, in this case, is not only to detect the '7's, but also the anomaly of '1's. After applying F-FFP, the columns in S that correspond to anomalies contain relatively larger values. We use the ℓ_2 norm to measure the values in each column of S and show the values in Fig. 9. The highest bars suggests the corresponding examples to be outliers. For ease of visualization, we vanish the values that are smaller than 5 in Fig. 9. We

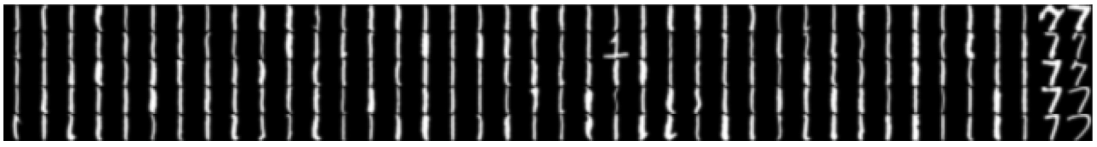


Fig. 8. Selected '1's and '7's from USPS dataset.

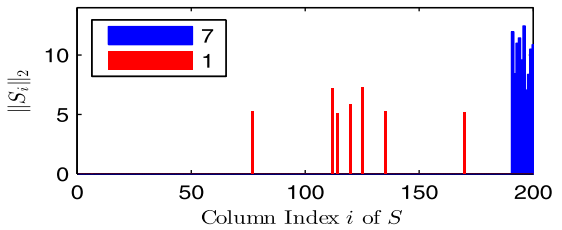
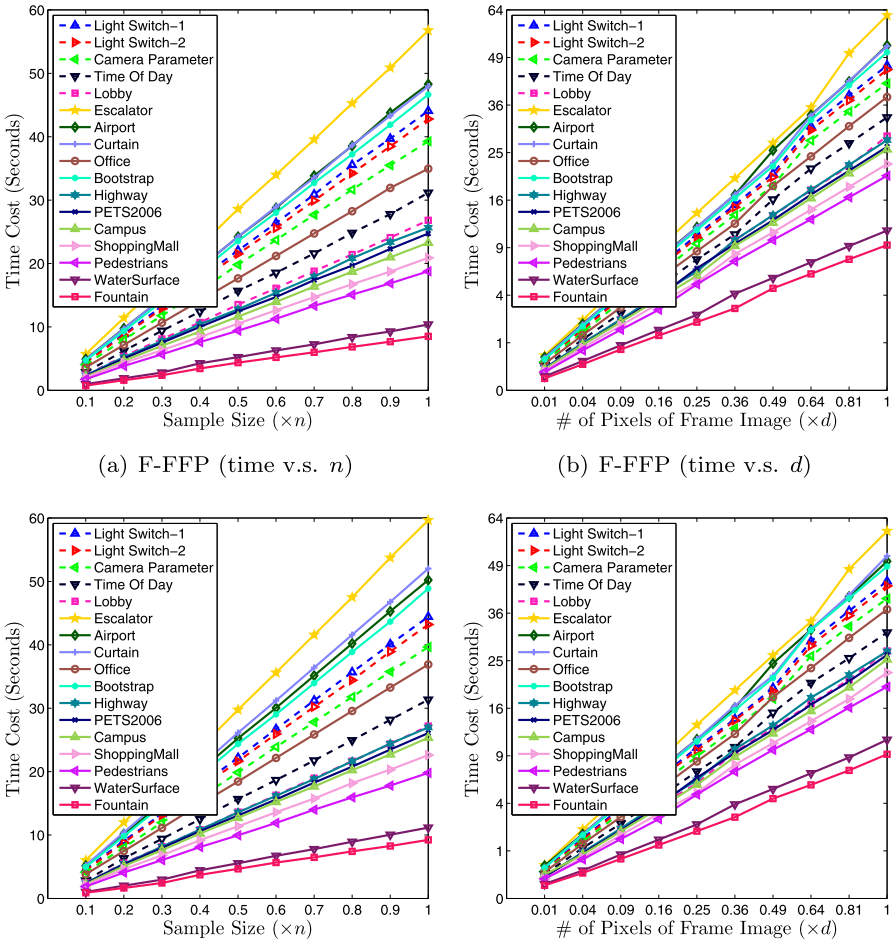


Fig. 9. ℓ_2-norms of each row of S.

Fig. 10. Written '1's and outliers identified by F-FFP.



(a) F-FFP (time v.s. n) (b) F-FFP (time v.s. d)

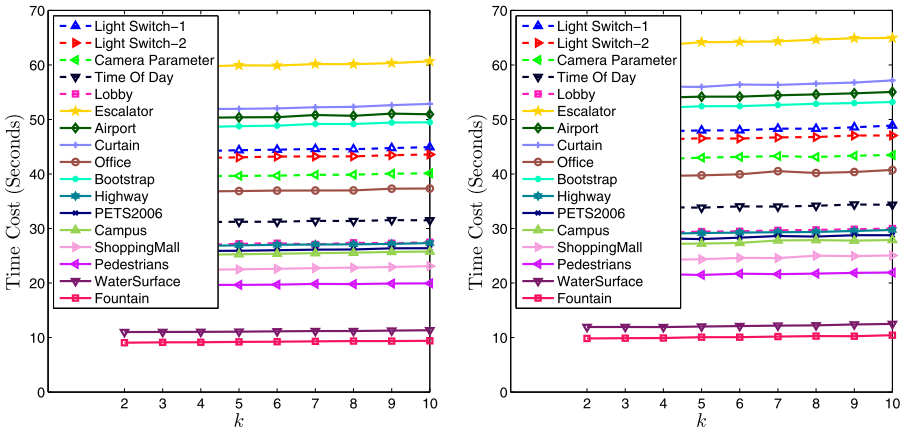
(c) U-FFP (time v.s. n) (d) U-FFP (time v.s. d)

Fig. 11. Plots of time cost of F-FFP and U-FFP as functions of data size (left column) and dimension (right column), respectively.

show the corresponding digits in Fig. 10, which include all '7's and additional '1's. It is seen that these digits starkly differ from others, which is reasonable to regard them as outliers. This implies the efficiency of the proposed method.

6.5. Scalability

To numerically illustrate the scalability of F-FFP and U-FFP, we test how the time cost increases with values of n and d, respectively. To test the effect of n, we uniformly sample a partition of the frames with the sampling rate in {0.1, 0.2,..., 1.0}. To test the relationship of time with d, we down-sample pixels of each frame with different rates varying over $\{0.1^2,$

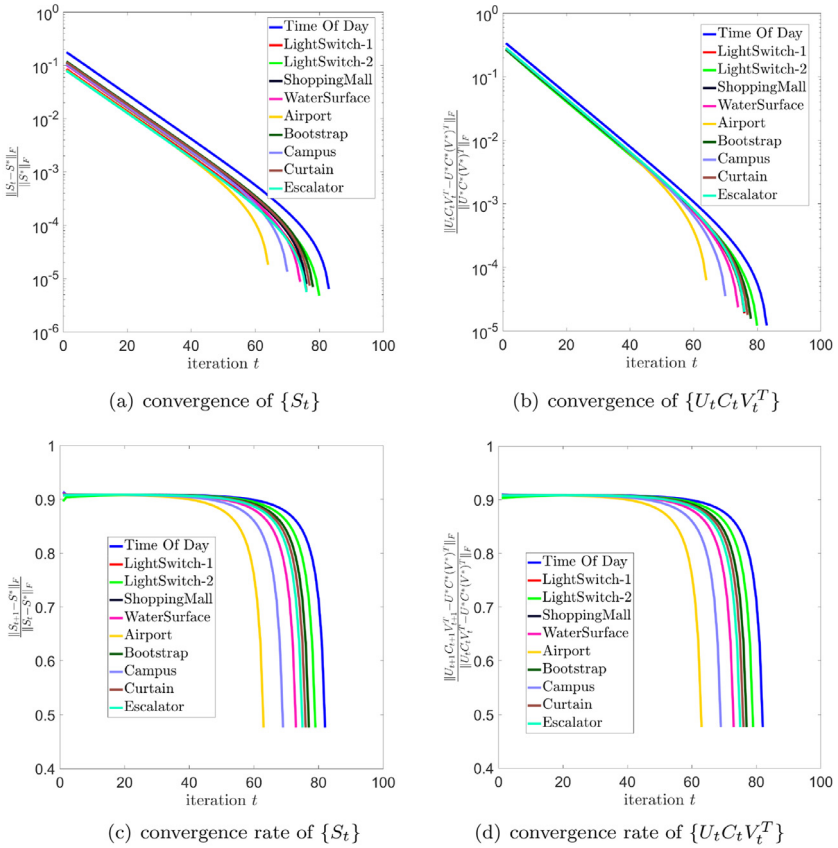


(a) F-FFP (time v.s. k) (b) U-FFP (time v.s. k)

Fig. 12. Plots of time cost of F-FFP and U-FFP versus k, respectively.

0.2^2,..., 1.0^2}. For both F-FFP and U-FFP, we temporarily ignore other terminating conditions and terminate them within 50 iterations. 10 runs are repeated for each case and the average time cost is reported in Fig. 11. It is observed that the time cost increases essentially linearly with n for both F-FFP and U-FFP. We show the plots with both x- and y-coordinates scaled to their square roots. The line plots essentially reveal the scalability of the algorithms in d.

Besides, we also test how the time cost increases as k does. For this test, we use the overall datasets and record the time cost with $k \in \{2, 3, 4,..., 10\}$. We report these results in Fig. 12. It is observed that the time cost increases only slightly when k increases from 2 to 10, implying low cost for selecting larger k.



(a) convergence of $\{S_t\}$ (b) convergence of $\{U_tC_tV_t^T\}$

(c) convergence rate of $\{S_t\}$ (d) convergence rate of $\{U_tC_tV_t^T\}$

Fig. 13. Examples of convergence on 10 data sets.

6.6. Convergence rate

In previous sections, we have theoretically analyzed the convergence of our algorithm. Besides, the complexity of our algorithm is $O(dnk)$ per iteration. However, it is highly non-trivial to provide theoretical results on the convergence rate. Thus, the overall theoretical complexity is out of the scope of this paper. In this test, we will provide some empirical results on the coverall convergence complexity. For a clearer illustration yet without loss of generality, in this test we fix the parameters $\rho = 1$ and $\kappa = 1.1$ for FFP. We set the solution of S^* and $U^*C^*(V^*)^T$ to be S_t and $U_tC_tV_t^T$, respectively, when $\frac{\|X-S_t-U_tC_tV_t^T\|_F}{\|X\|_F} \leq \epsilon$ is satisfied, where $\epsilon = 10^{-8}$ is the terminating tolerance. Results in this test can be found in Fig. 13. Here, we first plot the sequence of $\{\frac{\|S_t-S^*\|_F}{\|S^*\|_F}\}$, which clearly shows the convergence of $\{S_t\}$. We further show the plot of $\{\frac{\|S_{t+1}-S^*\|_F}{\|S_t-S^*\|_F}\}$ to show the convergence rate. It is observed that $\frac{\|S_{t+1}-S^*\|_F}{\|S_t-S^*\|_F} < 1$ and tends to be decreasing, implying a superlinear convergence rate. For U, C, and V, we show the results of UCV^T for ease of illustration, where similar observations to S are found. Thus, it is convincing that the proposed method has an overall complexity of $O(dnk\frac{1}{\epsilon})$ in real world applications, where ϵ is the terminating tolerance or recovery accuracy.

7. Conclusion

In this paper, we propose a new factorization-based RPCA model, which is equivalent to the traditional convex RPCA under some mild conditions. We develop an ALM-type optimization strategy, which provably converges to a stationary point. The proposed optimization algorithms have scalability in both data dimension and sample size, which is crucial for large-scale data analysis. Extensive experiments confirm the effectiveness and efficiency of the proposed model and algorithms both quantitatively and qualitatively.

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.

Acknowledgments

This work is supported by National Natural Science Foundation of China (NSFC) under grants 61806106, 61802215, and 61806045, Shandong Provincial Natural Science Foundation, China under grants ZR2019QF009, ZR2019BF028, and ZR2019BF011; Q.C. is partially supported by NIH UH3 NS100606-03 and a grant from the University of Kentucky.

References

[1] Q. Ke, T. Kanade, Robust l_1 norm factorization in the presence of outliers and missing data by alternative convex programming, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 1, IEEE, 2005, pp. 739–746.
[2] R. Maronna, D. Martin, V. Yohai, Robust Statistics, John Wiley & Sons, Chichester. ISBN, 2006.
[3] F. De La Torre, M.J. Black, A framework for robust subspace learning, Int. J. Comput. Vis. 54 (1–3) (2003) 117–142.
[4] R. Gnanadesikan, J.R. Kettenring, Robust estimates, residuals, and outlier detection with multiresponse data, Biometrics (1972) 81–124.
[5] L. Xu, A.L. Yuille, Robust principal component analysis by self-organizing rules based on statistical physics approach, IEEE Trans. Neural Netw. 6 (1) (1995) 131–143.
[6] C. Croux, G. Haesbroeck, Principal component analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies, Biometrika 87 (3) (2000) 603–618.
[7] J. Wright, A. Ganesh, S. Rao, Y. Peng, Y. Ma, Robust principal component analysis: exact recovery of corrupted low-rank matrices via convex optimization, in: Advances in neural information processing systems, 2009, pp. 2080–2088.
[8] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM (JACM) 58 (3) (2011) 11.
[9] J.-F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. 20 (4) (2010) 1956–1982.
[10] K.-C. Toh, S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems, Pacific J. Optim. 6 (615–640) (2010) 15.
[11] Z. Lin, M. Chen, Y. Ma, The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices (2010) arXiv:1009.5055.
[12] X. Ding, L. He, L. Carin, Bayesian robust principal component analysis, Image Process. IEEE Trans. 20 (12) (2011) 3419–3430.
[13] Z. Kang, C. Peng, Q. Cheng, Robust pca via nonconvex rank approximation, in: Data Mining (ICDM), 2015 IEEE International Conference on, IEEE, 2015, pp. 211–220.
[14] W.K. Leow, Y. Cheng, L. Zhang, T. Sim, L. Foo, Background recovery by fixed-rank robust principal component analysis, in: Computer Analysis of Images and Patterns, Springer, 2013, pp. 54–61.
[15] P. Netrapalli, U. Niranjan, S. Sanghavi, A. Anandkumar, P. Jain, Non-convex robust PCA, in: Advances in Neural Information Processing Systems, 2014, pp. 1107–1115.
[16] H. Xu, C. Caramanis, S. Sanghavi, Robust PCA via outlier pursuit, in: Advances in Neural Information Processing Systems, 2010, pp. 2496–2504.
[17] M. McCoy, J.A. Tropp, et al., Two proposals for robust PCA using semidefinite programming, Electron. J. Stat. 5 (2011) 1123–1160.
[18] F. Nie, H. Huang, C. Ding, D. Luo, H. Wang, Robust principal component analysis with non-greedy -norm maximization, in: International Joint Conference on Artificial Intelligence, 2011, pp. 1433–1438.
[19] F. Nie, H. Huang, Subspace clustering via new low-rank model with discrete group structure constraint, in: International Joint Conference on Artificial Intelligence, 2016, pp. 1874–1880.
[20] C. Peng, Z. Kang, H. Li, Q. Cheng, Subspace clustering using log-determinant rank approximation, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 925–934.
[21] C. Peng, C. Chen, Z. Kang, J. Li, Q. Cheng, Res-pca: a scalable approach to recovering low-rank matrices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7317–7325.

[22] C. Chen, S. Li, H. Qin, A. Hao, Robust salient motion detection in non-stationary videos via novel integrated strategies of spatio-temporal coherency clues and low-rank analysis, Pattern Recognit. 52 (2016) 410–432, doi:10.1016/j.patcog.2015.09.033.
[23] C. Chen, S. Li, Y. Wang, H. Qin, A. Hao, Video saliency detection via spatial-temporal fusion and low-rank coherency diffusion, IEEE Trans. Image Process. 26 (7) (2017) 3156–3170, doi:10.1109/TIP.2017.2670143.
[24] Z. Kang, H. Pan, S.C.H. Hoi, Z. Xu, Robust graph learning from noisy data, IEEE Trans. Cybern. (2019) 1–11, doi:10.1109/TCYB.2018.2887094.
[25] P. Jing, Y. Su, L. Nie, X. Bai, J. Liu, M. Wang, Low-rank multi-view embedding learning for micro-video popularity prediction, IEEE Trans. Knowl. Data Eng. 30 (8) (2018) 1519–1532, doi:10.1109/TKDE.2017.2785784.
[26] P. Jing, Y. Su, L. Nie, H. Gu, J. Liu, M. Wang, A framework of joint low-rank and sparse regression for image memorability prediction, IEEE Trans. Circuits Syst. Video Technol. 29 (5) (2019) 1296–1309, doi:10.1109/TCSVT.2018.2832095.
[27] Z. Kang, L. Wen, W. Chen, Z. Xu, Low-rank kernel learning for graph-based clustering, Knowl.-Based Syst. 163 (2019) 510–517.
[28] A. Li, D. Chen, Z. Wu, G. Sun, K. Lin, Self-supervised sparse coding scheme for image classification based on low rank representation, PloS one 13 (6) (2018) e0199141.
[29] A. Li, X. Liu, Y. Wang, D. Chen, K. Lin, G. Sun, H. Jiang, Subspace structural constraint-based discriminative feature learning via nonnegative low rank representation, PloS one 14 (5) (2019) e0215450.
[30] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imag. Sci. 2 (1) (2009) 183–202.
[31] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Commun. PureAppl. Math. 57 (11) (2004) 1413–1457.
[32] P.H. Schönemann, A generalized solution of the orthogonal procrustes problem, Psychometrika 31 (1) (1966) 1–10.
[33] A. Li, Z. Wu, H. Lu, D. Chen, G. Sun, Collaborative self-regression method with nonlinear feature based on multi-task learning for image classification, IEEE Access 6 (2018) 43513–43525, doi:10.1109/ACCESS.2018.2862159.
[34] C. Peng, J. Cheng, Q. Cheng, A supervised learning model for high-dimensional and large-scale data, ACM Trans. Intell. Syst.Technol. (TIST) 8 (2) (2017) 30.
[35] C. Peng, Q. Cheng, Discriminative regression machine: a classifier for high-dimensional data or imbalanced data (2019) arXiv:1904.07496.
[36] R. Basri, D.W. Jacobs, Lambertian reflectance and linear subspaces, Pattern Anal. Machine Intell. IEEE Trans. 25 (2) (2003) 218–233.
[37] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, Pattern Anal. Mach. Intell. IEEE Trans. 23 (6) (2001) 643–660.
[38] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications., IEEE Trans. Pattern Anal. Mach.Intell. 35 (11) (2013) 2765–2781.
[39] C. Peng, Z. Kang, Q. Cheng, Subspace clustering via variance regularized ridge regression, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 682–691, doi:10.1109/CVPR.2017.80.
[40] D. Cai, Litekmeans: the fastest matlab implementation of kmeans, Available at: http://www.zjucadcg.cn/dengcai/Data/Clustering.html (2011).
[41] L. Nie, Y.-L. Zhao, X. Wang, J. Shen, T.-S. Chua, Learning to recommend descriptive tags for questions in social forums, ACM Trans. Inf. Syst. (TOIS) 32 (1) (2014) 5.
[42] C. Peng, Z. Kang, S. Cai, Q. Cheng, Integrate and conquer: double-sided two-dimensional k-means via integrating of projection and manifold construction, ACM Trans. Intell. Syst. Technol. 9 (5) (2018) 57:1–57:25.