INVITED PAPER

Efficient Optimization Algorithms for Robust Principal Component Analysis and Its Variants

This paper reviews specialized efficient optimization algorithms that have been developed to solve convex relaxations of various optimization programs that can be defined to solve robust PCA and related problems.

By Shiqian Ma and Necdet Serhat Aybat

ABSTRACT | Robust principal component analysis (RPCA) has drawn significant attention in the last decade due to its success in numerous application domains, ranging from bioinformatics, statistics, and machine learning to image and video processing in computer vision. RPCA and its variants such as sparse PCA and stable PCA can be formulated as optimization problems with exploitable special structures. Many specialized efficient optimization methods have been proposed to solve robust PCA and related problems. In this paper, we review existing optimization methods for solving convex and nonconvex relaxations/variants of RPCA, discuss their advantages and disadvantages, and elaborate on their convergence behaviors. We also provide some insights for possible future research directions including new algorithmic frameworks that might be suitable for implementing on multiprocessor setting to handle large-scale problems.

KEYWORDS | Convergence rate; convex optimization; iteration complexity; nonconvex optimization; principal component analysis (PCA); robust PCA (RPCA); ϵ-stationary solution.

Manuscript received January 31, 2018; revised May 14, 2018; accepted June 8, 2018. Date of publication July 18, 2018; date of current version August 2, 2018. The work of S. Ma was supported in part by a startup package from the Department of Mathematics, University of California Davis, Davis, CA, USA. The work of N. S. Aybat was supported in part by the National Science Foundation (NSF) under Grants CMMI-1400217 and CMMI-1635106 and by the Army Research Office (ARO) under Grant W911NF-17-1-0298. *(Corresponding author: Shiqian Ma.)*

S. Ma is with the Department of Mathematics, University of California, Davis, Davis, CA 95616 USA (e-mail: sqma@ucdavis.edu).

N. S. Aybat is with the Department of Industrial and Manufacturing Engineering, Pennsylvania State University, University Park, PA 16802 USA.

Digital Object Identifier 10.1109/JPROC.2018.2846606

I. INTRODUCTION

Principal component analysis (PCA) is a fundamental tool in statistics and data science. It obtains a low-dimensional expression for high-dimensional data in an ℓ_2 sense. However, it is known that the classical PCA is sensitive to gross errors. Robust PCA (RPCA) has been proposed to remove the effect of sparse gross errors. For a given data matrix $M \in \mathbb{R}^{m\times n}$, RPCA seeks to decompose it into two parts $M := L^o + S^o$ where L^o is a low-rank matrix and S^o is a sparse matrix. That is, RPCA assumes that M is a superposition of L^o and S^o. As a result, the gross errors will be captured by the sparse matrix S^o so that the low-rank matrix L^o can still approximate M well. RPCA does not only provide a low-dimensional approximation which is robust to outliers, but it also finds vast applications in a variety of real applications such as computer vision [1], image alignment [2], subspace recovery [3], clustering [4], and so on. Mathematically, the conditions on the low-rank and sparse components (L^o, S^o) have been investigated in [1] and [5]–[7], so that the inverse problem of recovering unknown (L^o, S^o) given M is well defined. One particular formulation of RPCA can be stated as follows:

$$\min_{L,S\in\mathbb{R}^{m\times n}} \ \operatorname{rank}(L) + \rho\|S\|_0 \quad \text{s.t. } L + S = M \qquad (1)$$

where $\|S\|_0$ is called the ℓ_0-norm[1] of S and counts the number of nonzero entries of S, and $\rho > 0$ is a trade-

[1]Technically, it is not a norm because it is not homogeneous; but, we still call it a norm following the convention.

off parameter. It is known that (1) is NP-hard and thus numerically intractable. Later, it was shown in [1], [5], and [6] that under certain conditions, (1) is equivalent to the following convex program with high probability:

$$\min_{L,S\in\mathbb{R}^{m\times n}} \|L\|_* + \rho\|S\|_1 \quad \text{s.t. } L+S=M \tag{2}$$

where $\|L\|_*$ is called the nuclear norm of L and equal to the sum of the singular values of L, and $\|S\|_1 := \sum_{ij}|S_{ij}|$ is called the ℓ_1-norm of S. The optimization problem in (2) is called robust principal component pursuit (RPCP), and it can be reformulated as a semidefinite program (SDP) [8] and solved by an interior point method for SDPs. However, RPCA problems arising in practice are usually of very large scale, and interior point methods do not scale well for these problems. More efficient algorithms that solve (2) and its variants by exploiting the structure in these problems were studied extensively in the literature. One variant of (2) deals with an additional dense noise component. In particular, when M contains also a dense noise component N^o such that $\|N^o\|_F \le \sigma$ for some noise level $\sigma > 0$, i.e., $M = L^o + S^o + N^o$, instead of RPCP formulation in (2), the following so-called stable PCP (SPCP) problem is solved:

$$\min_{L,S\in\mathbb{R}^{m\times n}} \|L\|_* + \rho\|S\|_1 \quad \text{s.t. } \|L+S-M\|_F \le \sigma. \tag{3}$$

It is proved in [9] that, under certain conditions on M, solving (3) gives a stable estimate of L^o and S^o with high probability in the sense that $\|\hat{L}-L^o\|_F^2 + \|\hat{S}-S^o\|_F^2 \le \mathcal{O}(mn\sigma^2)$ where $(\hat{L},\hat{S})$ denotes the optimal solution to (3). Since (3) satisfies the Slater's condition, it is equivalent to the following unconstrained problem for an appropriately chosen parameter $\mu > 0$ depending on σ:

$$\min_{L,S\in\mathbb{R}^{m\times n}} \|L\|_* + \rho\|S\|_1 + \frac{\mu}{2}\|L+S-M\|_F^2. \tag{4}$$

Note that if M is only partially observed, that is, if we only have observations on M_{ij} for some indices (i,j) from a subset Ω, then (2)–(4) can be, respectively, reformulated as

$$\min_{L,S\in\mathbb{R}^{m\times n}} \|L\|_* + \rho\|S\|_1 \quad \text{s.t. } P_\Omega(L+S-M)=\mathbf{0} \tag{5}$$

$$\min_{L,S\in\mathbb{R}^{m\times n}} \|L\|_* + \rho\|S\|_1 \quad \text{s.t. } \|P_\Omega(L+S-M)\|_F \le \sigma \tag{6}$$

$$\min_{L,S\in\mathbb{R}^{m\times n}} \|L\|_* + \rho\|S\|_1 + \frac{\mu}{2}\|P_\Omega(L+S-M)\|_F^2 \tag{7}$$

where the operator $P_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is defined as $[P_\Omega(M)]_{ij} = M_{ij}$, if $(i,j)\in\Omega$, and $[P_\Omega(M)]_{ij} = 0$ otherwise. Most algorithms we discuss in this paper for solving (2)–(4) can be used to solve (5)–(7) directly or with very little modification. For brevity, we will only describe algorithms for solving (2)–(4) in this paper.

For the sake of completeness, we here briefly describe the results in [1] and [9]. Assume that given data matrix $M\in\mathbb{R}^{m\times n}$ is a superposition of the unknown component matrices L^o, S^o, and N^o, i.e., $M = L^o + S^o + N^o$, such that L^o is low-rank ($r := \operatorname{rank}(L^0) \ll \min\{m,n\}$), S^o is sparse ($s := \|S^o\|_0 \ll mn$), and $\|N^o\| \le \sigma$ for some $\sigma > 0$. Robust/stable PCA is an inverse problem with the objective of recovering L^o and S^o from the data matrix M. Clearly, if S^o is low-rank and/or L^o is sparse, the recovery is hopeless. To avoid these pathological instances, Candès *et al.* [1] and Zhou *et al.* [9] consider (L^o,S^o) pairs coming from a particular class satisfying some incoherence and randomness conditions. Suppose the singular value decomposition (SVD) of L^o is given by $L^o = \sum_{i=1}^r \sigma_i u_i v_i^\top$, where $U = [u_1,\ldots,u_r]$ and $V = [v_1,\ldots,v_r]$ are formed by its left- and right-singular vectors. The incoherence conditions assume that there exists a parameter $\delta > 0$ such that

$$\max_i \|U^\top e_i\|^2 \le \delta r/m, \quad \max_i \|V^\top e_i\|^2 \le \delta r/n \tag{8a}$$

$$\|UV^\top\|_\infty \le \sqrt{\frac{\delta r}{mn}} \tag{8b}$$

where $\|Z\|_\infty := \max_{ij}|Z_{ij}|$ and e_i denotes the ith unit vector. For S^o, it is assumed that the set of indices for the nonzero entries is random and follows a uniform distribution among all the subsets of cardinality s. Roughly speaking, these conditions assure that the low-rank matrix L^o is not sparse, and the sparse matrix S^o is not low-rank. Under these assumptions it is shown in [1] that when $N^o = \mathbf{0}$, i.e., $\sigma = 0$, solving the convex program (2) with $\rho = 1/\sqrt{\max\{m,n\}}$ recovers the optimal solution of (1) (L^*,S^*), with high probability and $(L^*,S^*) = (L^o,S^o)$ provided that L^o is sufficiently low-rank and S^o is sufficiently sparse comparing to the matrix size; see, also, [6]. These results extend to the case where M is partially observed; indeed, Candès *et al.* [1] show that solving (5) recovers L^o under similar conditions. Moreover, in [9], Zhou *et al.* showed that under certain conditions, solving the convex problem (3) again with $\rho = 1/\sqrt{\max\{m,n\}}$ generates a low-rank and sparse decomposition (L^*,S^*) such that $\|L^*-L^o\|_F^2 + \|S^*-S^o\|_F^2 \le Cmn\sigma^2$ for some constant $C > 0$ (independent of m, n, and σ) with high probability; note when $N=\mathbf{0}$, the recovery is exact with high probability.

Recently, there have been works that further study statistical guarantees of different RPCA models. Zhang *et al.* [10] provide a refined analysis of RPCA which allows the support of the sparse error matrix to be generated with nonuniform sampling, i.e., entries of the low-rank matrix are corrupted with different probabilities, hence, one can model the scenario where some entries are more prone to corruption than the others. A nonconvex model of RPCA is studied in [11] and a gradient descent (GD) method with proper initialization is shown to be able to reduce the computational complexity comparing with existing methods. Zhang and Yang [12] consider a nonconvex optimization formulation with manifold constraint for RPCA. Two algorithms for

Table 1 Convergence Rates of Different Algorithms

Algorithm	Problem	ϵ-optimality measure	Convergence rate
Algorithms for Convex Models			
PGM [16]	(4)	objective value error	$\mathcal{O}(1/k)$
APGM [16]	(4)	objective value error	$\mathcal{O}(1/k^2)$
IALM [17]	(2)	—	convergence, no rate given
ADMM [18]	(2)	—	convergence, no rate given
ALM [19]	(17)	objective value error	$\mathcal{O}(1/k)$
FALM [19]	(17)	objective value error	$\mathcal{O}(1/k^2)$
ASALM [20]	(19)	—	convergence unclear, no rate given
VASALM [20]	(19)	—	convergence, no rate given
PSPG [21]	(3)	objective value error	$\mathcal{O}(1/k)$
ADMIP [22]	(3)	objective value error	$\mathcal{O}(1/k)$
Quasi-Newton method (fastRPCA) [23]	(26)	—	convergence, no rate given
3-block ADMM [24]	(28)	—	convergence, no rate given
Frank-Wolfe [25]	(30)	objective value error	$\mathcal{O}(1/k)$
Algorithms for Nonconvex Models			
GoDec [26]	(33)	—	local convergence, no rate given
GreBsmo [27]	(36)	—	convergence unclear, no rate given
Alternating Minimization (R2PCP) [28]	(35)	—	local convergence, no rate given
Gradient Descent (GD) [11]	$\approx$ (36)	—	linear convergence with proper initialization and incoherence assumption
Alternating Minimization [29]	(37)	—	local convergence with proper initialization and incoherence and RIP assumptions
Stochastic alg. [30]	(39)	—	convergence if the iterates are always full rank matrices, no rate given
LMafit [31]	(44)	—	convergence if difference between two consecutive iterates tends to zero, no rate given
Conditional Gradient [32]	(48)	perturbed KKT	$\mathcal{O}(1/\sqrt{k})$
ADMM [32]	(51)	perturbed KKT	$\mathcal{O}(1/\sqrt{k})$
Proximal BCD [32]	(54)	perturbed KKT	$\mathcal{O}(1/\sqrt{k})$

†Note: Some of these algorithms solve different problems and the ϵ-optimality measures are also different, so the convergence rates are not directly comparable with each other. [11] has no explicit optimization formulation but the objective is similar to (36). Moreover, global convergence is usually guaranteed for convex solvers, but only local convergence is usually guaranteed for nonconvex solvers, unless certain very strong assumptions are made.

manifold optimization are proposed in [12], and it is shown that they can reduce the dependence on the condition number of the underlying low-rank matrix theoretically. Netrapalli *et al.* [13] consider another nonconvex formulation of RPCA and analyze the iteration complexity of the proposed alternating projection method.

There are also recent survey papers [14], [15] that discuss algorithms for solving RPCA, but these papers mainly focus on its convex relaxations. In this paper, we aim to review existing algorithms for both convex and nonconvex relaxations/variants of RPCA models and point out a few possible future directions.

II. ALGORITHMS FOR CONVEX RELAXATIONS/VARIANTS OF RPCA

The earliest first-order methods for solving the convex RPCP problem are given in [16] and [17]. In [16], Lin *et al.* proposed an accelerated proximal gradient method (APGM) [33]–[35] for solving (4) in which each iteration involves computing the proximal mappings of the nuclear norm $\|L\|_*$ and the ℓ_1-norm $\|S\|_1$. In particular, the nonaccelerated proximal gradient method (PGM) for solving (4) simply updates L and S as

$$\begin{aligned} G^k &:= \mu(L^k + S^k - M) \\ L^{k+1} &:= \operatorname{argmin}_L \ \|L\|_* + \frac{1}{2\tau}\|L - (L^k - \tau G^k)\|_F^2 \\ S^{k+1} &:= \operatorname{argmin}_S \ \rho\|S\|_1 + \frac{1}{2\tau}\|S - (S^k - \tau G^k)\|_F^2. \end{aligned} \quad (9)$$

Note that G^k is the gradient of the quadratic penalty function in (4) and $\tau > 0$ denotes a step size. The two subproblems in (9) both admit easy closed-form optimal solutions. Specifically, the solution of the L-subproblem corresponds to the proximal mapping of the nuclear norm, which is given by

$$L^{k+1} = \mathcal{S}_\tau(L^k - \tau G^k)$$

where the matrix shrinkage operation is defined as

$$\mathcal{S}_\nu(Z) = U\,\mathrm{diag}((\sigma - \nu)_+)V^\top \quad (10)$$

where $Z = U\,\mathrm{diag}(\sigma)V^\top$ is the SVD of Z, and $z_+ := \max(0, z)$. The solution of the S-subproblem corresponds to the proximal mapping of the ℓ_1-norm, which is given by

$$S^{k+1} = s_{\rho\tau}(S^k - \tau G^k)$$

where the vector shrinkage operation is defined as

$$[s_\nu(Z)]_{ij} = \mathrm{sign}(Z_{ij}) \circ \max\{0, |Z_{ij}| - \nu\} \quad (11)$$

where $\mathrm{sign}(a)$ denotes the sign of a, and $\circ$ denotes the Hadamard product.

APGM incorporates Nesterov's acceleration technique and updates the variables as follows starting with

$t_{-1} = t_0 = 1$

$$\begin{aligned}
\bar{L}^k &:= L^k + \frac{t_{k-1}-1}{t_k}(L^k - L^{k-1}) \\
\bar{S}^k &:= S^k + \frac{t_{k-1}-1}{t_k}(S^k - S^{k-1}) \\
\bar{G}^k &:= \mu(\bar{L}^k + \bar{S}^k - M) \\
L^{k+1} &:= \operatorname{argmin}_L \ \|L\|_* + \frac{1}{2\tau}\|L - (\bar{L}^k - \tau\bar{G}^k)\|_F^2 \\
S^{k+1} &:= \operatorname{argmin}_S \ \rho\|S\|_1 + \frac{1}{2\tau}\|S - (\bar{S}^k - \tau\bar{G}^k)\|_F^2 \\
t_{k+1} &:= (1 + \sqrt{1 + 4t_k^2})/2. \qquad (12)
\end{aligned}$$

Results in [33]–[35] show that the proximal gradient method (9) and the accelerated proximal gradient method (12) find an ϵ-optimal solution to (4) in terms of objective value error in no more than $O(1/\epsilon)$ and $O(1/\sqrt{\epsilon})$ iterations, respectively.

When there is no noise, i.e., $\sigma = 0$, the problem of interest is (2). The drawback of the above approach for solving the unconstrained version in (4) is that (4) is equivalent to (2) only when $\mu \to +\infty$. Therefore, for any fixed $\mu > 0$, there is always a residual term which does not go to zero. To remedy this, the same group of authors [17] considered the augmented Lagrangian method (ALM) for solving (2). By associating a Lagrange multiplier Λ to the linear equality constraint, the augmented Lagrangian function of (2) can be written as

$$\begin{aligned}
\mathcal{L}_\beta(L, S; \Lambda) : &= \|L\|_* + \rho\|S\|_1 - \langle \Lambda, L + S - M\rangle \\
&+ \frac{\beta}{2}\|L + S - M\|_F^2
\end{aligned}$$

where $\beta > 0$ is a penalty parameter. A typical iteration of ALM iterates as follows:

$$(L^{k+1}, S^{k+1}) := \operatorname*{argmin}_{L,S} \mathcal{L}_\beta(L, S; \Lambda^k) \qquad (13a)$$

$$\Lambda^{k+1} := \Lambda^k - \beta(L^{k+1} + S^{k+1} - M). \qquad (13b)$$

Note that the first step in (13) requires to minimize the augmented Lagrangian function with respect to L and S simultaneously, which usually is computationally very expensive and almost as hard as solving the original problem in (2). In [17], Lin *et al.* proposed both exact and inexact versions of ALM, where the former one solves the subproblems (almost) exactly and the latter one solves the subproblems inexactly according to a particular subproblem termination criterion. Both the exact ALM and inexact ALM (IALM) employ some iterative algorithm for minimizing the augmented Lagrangian function until certain overall stopping criterion is met, which may require many iterations and is thus time consuming. Around the same time when [16], [17] appeared, the alternating direction method of multipliers (ADMM) was revisited and found very successful in solving signal processing and image processing problems [36]–[39]. It was then found that RPCP in (2) can be nicely solved by ADMM due to its special separable structure [18], [19]. The ADMM iterations for solving (2) take the following form:

$$\begin{aligned}
L^{k+1} &:= \operatorname{argmin}_L \mathcal{L}_\beta(L, S^k; \Lambda^k) \\
S^{k+1} &:= \operatorname{argmin}_S \mathcal{L}_\beta(L^{k+1}, S; \Lambda^k) \\
\Lambda^{k+1} &:= \Lambda^k - \beta(L^{k+1} + S^{k+1} - M). \qquad (14)
\end{aligned}$$

Comparing to ALM in (13), it is noted that ADMM splits the subproblem in (13a) into two smaller subproblems that correspond to computing proximal mappings of $\|L\|_*$ and $\|S\|_1$, respectively. The ADMM (14) is known as two-block ADMM as there are two block variables L and S, and hence two subproblems are solved in each iteration of the algorithm. It is now widely known that the two-block ADMM is a special case of the so-called Douglas–Rachford operator splitting method [40]–[43] applied to the dual problem, and the two-block ADMM for solving convex problems globally converges for any penalty parameter $\beta > 0$ [44] and converges with a sublinear rate $O(1/k)$ (see, e.g., [45]–[47]).

The alternating linearization method (ALM) proposed by Goldfarb *et al.* [19] is shown to be equivalent to a symmetric version of ADMM (14) with either $\|L\|_*$ or $\|S\|_1$ replaced with some suitable smooth approximation. For instance, given $\nu > 0$, define $g_\nu : \mathbb{R}^{m\times n} \to \mathbb{R}$ such that

$$g_\nu(S) = \max_{Z\in\mathbb{R}^{m\times n}} \left\{ \langle S, Z\rangle - \frac{\nu}{2}\|Z\|_F^2 : \ \|Z\|_\infty \le \rho \right\} \qquad (15)$$

and let $g(S) = \rho\|S\|_1$. Clearly, $g_\nu \to g$ uniformly as $\nu \searrow 0$. Moreover, g_ν is a differentiable convex function such that ∇g_ν is Lipschitz continuous. Indeed, given $S \in \mathbb{R}^{m\times n}$, let $Z_\nu(S)$ be the maximizer for (15), which in closed form can be written as $Z_\nu(S) = \operatorname{sign}(S) \circ \max\{\frac{1}{\nu}|S|, \ \rho\mathbf{1}_{m\times n}\}$, and $\nabla g_\nu(S) = Z_\nu(S)$ is Lipschitz continuous with constant $C_{g_\nu} = \frac{1}{\nu}$. Similarly, given $\mu > 0$, define $f_\mu : \mathbb{R}^{m\times n} \to \mathbb{R}$ such that

$$f_\mu(L) = \max_{W\in\mathbb{R}^{m\times n}} \left\{ \langle L, W\rangle - \frac{\mu}{2}\|W\|_F^2 : \ \|W\| \le 1 \right\} \qquad (16)$$

where $\|\cdot\|$ denotes the spectral norm, and let $f(L) = \lambda\|L\|_*$. Clearly, $f_\mu \to f$ uniformly as $\mu \searrow 0$. Moreover, f_μ is a differentiable convex function such that ∇f_μ is Lipschitz continuous. Indeed, given $L \in \mathbb{R}^{m\times n}$, let $W_\mu(L)$ be the maximizer for (16), which in closed form can be written as $W_\mu(L) = U\,\mathbf{diag}(\max\{(\sigma/\mu) - \mathbf{1}, \ \mathbf{0}\})V^\top$, where $L = U\,\mathbf{diag}(\sigma)V^\top$ is the singular value decomposition of L with $\sigma \in \mathbb{R}^r_{++}$ denoting the vector of singular values; moreover, $\nabla f_\mu(S) = W_\mu(L)$ is Lipschitz continuous with constant $C_{f_\mu} = (1/\mu)$.

The alternating linearization method in [19] can be applied to solve the following problem, which is a smoothed version of (2):

$$\min \ \|L\|_* + g_\nu(S), \quad \text{s.t. } L + S = M. \qquad (17)$$

Denote the augmented Lagrangian function of (17) as $\tilde{\mathcal{L}}_\beta(L, S; \Lambda)$, then the alternating linearization method in [19] iterates as follows:

$$\begin{aligned} L^{k+1} &:= \operatorname{argmin}_L \tilde{\mathcal{L}}_\beta(L, S^k; \Lambda^k) \\ \Lambda^{k+\frac{1}{2}} &:= \Lambda^k - \beta(L^{k+1} + S^k - M) \\ S^{k+1} &:= \operatorname{argmin}_S \tilde{\mathcal{L}}_\beta\left(L^{k+1}, S; \Lambda^{k+\frac{1}{2}}\right) \\ \Lambda^{k+1} &:= \Lambda^{k+\frac{1}{2}} - \beta(L^{k+1} + S^{k+1} - M). \end{aligned} \tag{18}$$

Goldfarb *et al.* [19] proved that this method has a sublinear convergence rate $O(1/k)$. They also proposed an accelerated version of (18) (FALM) in [19] by adopting Nesterov's acceleration technique, and proved that the accelerated alternating linearization method has a better sublinear convergence rate $O(1/k^2)$.

Based on the success of two-block ADMM for solving (2), it is then very natural to apply ADMM to solve SPCP in (3). To do so, one has to introduce a new variable N, and rewrite (3) equivalently as

$$\begin{aligned} \min_{L,S,N \in \mathbb{R}^{m\times n}} \ & \|L\|_* + \rho\|S\|_1 \\ \text{s.t.}, \ & L + S + N = M, \quad \|N\|_F \le \sigma. \end{aligned} \tag{19}$$

The ADMM for solving (19) iterates as follows updating three block variables:

$$\begin{aligned} L^{k+1} &:= \operatorname{argmin}_L \mathcal{L}_\beta(L, S^k, N^k; \Lambda^k) \\ S^{k+1} &:= \operatorname{argmin}_S \mathcal{L}_\beta(L^{k+1}, S, N^k; \Lambda^k) \\ N^{k+1} &:= \operatorname{argmin}_N \mathcal{L}_\beta(L^{k+1}, S^{k+1}, N; \Lambda^k) \\ \Lambda^{k+1} &:= \Lambda^k - \beta(L^{k+1} + S^{k+1} + N^{k+1} - M) \end{aligned} \tag{20}$$

where the augmented Lagrangian function for (19) is defined as

$$\begin{aligned} &\mathcal{L}_\beta(L, S, N; \Lambda) \\ &\quad := \|L\|_* + \rho\|S\|_1 + \mathbf{1}(N \mid \|N\|_F \le \sigma) \\ &\qquad - \langle \Lambda, L + S + N - M\rangle + \frac{\beta}{2}\|L + S + N - M\|_F^2 \end{aligned}$$

where $\mathbf{1}(N \mid \mathcal{N})$ denotes the indicator function of the set $\{N \in \mathcal{N}\}$, i.e., $\mathbf{1}(N \mid \mathcal{N}) = 0$ if $N \in \mathcal{N}$ and $\mathbf{1}(N \mid \mathcal{N}) = \infty$ otherwise. Note that the three subproblems in (20) all have closed-form solutions. In particular, the L-subproblem corresponds to the proximal mapping of $\|L\|_*$, the S-subproblem corresponds to proximal mapping of $\|S\|_1$, and the N-subproblem corresponds to projection onto the set $\{N \mid \|N\|_F \le \sigma\}$. Similar idea was used in [2] for robust image alignments. In practice, this three-block ADMM usually works very well. However, it was later discovered that the ADMM with more than two block variables is not necessarily convergent in general [48]. Note that although (19) contains three block variables, it can be viewed as a two-block problem, if we group S and N as one (larger) block variable. One of the earliest methods for solving SPCP in (3) and (19) is a three-block ADMM algorithm, ASALM, proposed by Tao and Yuan [20], and although it does not have any convergence guarantees, it works well in practice; and slightly changing the update rule in ASALM leads to VASALM, of which iterate sequence converges to an optimal solution; but this comes at the cost of degradation in practical convergence speed when compared to ASALM. Indeed, VASALM [20] can be seen as a linearized version of two-block ADMM to solve (3) with a convergence guarantee without any convergence rate result. To remedy the shortcoming associated with the theoretical convergence of three-block ADMM, several other alternatives based on two-block ADMM were proposed [21], [22], [49].

Aybat *et al.* [21] proposed an accelerated proximal gradient method, PSPG, for solving SPCP in (3). First, (3) is reformulated with a partially smooth objective. In particular, the nuclear norm is smoothed according to (16)

$$\min_{L,S \in \mathbb{R}^{m\times n}} f_\mu(L) + \rho\|S\|_1 \quad \text{s.t. } (L, S) \in \chi \tag{21}$$

$$\chi := \{(L, S) \mid \|L + S - M\|_F \le \sigma\} \tag{22}$$

where $\mu > 0$ is a given smoothing parameter. An accelerated proximal gradient method such as [33] and [34] can be applied to solve (21), because it was shown in [21] that the following subproblem is easy to solve:

$$\min_{L,S} \|S\|_1 + \frac{1}{2\xi}\|L - \tilde{L}\|_F^2 \quad \text{s.t. } (L, S) \in \chi \tag{23}$$

where $\xi > 0$ denotes a step size of the proximal gradient step and $\tilde{L}$ denotes some known matrix. This operation requires one sorting which has $\mathcal{O}(mn\log(mn))$ complexity.

For any $\epsilon > 0$, setting $\mu = \Omega(\epsilon)$, PSPG proposed in [21] can compute an ϵ-optimal solution to (3) within $\mathcal{O}(1/\epsilon)$ iterations, and its computational complexity per iteration is comparable to the work per iteration required by ASALM and VASALM, which is mainly determined by an SVD computation. On the other hand, it is also important to emphasize that PSPG iterate sequences do not converge to an optimal solution to the SPCP problem in (3). In particular, since within PSPG the smoothing parameter μ is fixed, depending on the approximation parameter ϵ for solving (21), further iterations after reaching an ϵ-optimal solution in $\mathcal{O}(1/\epsilon)$ iterations do not necessarily improve the solution quality.

In [22] and [49], the variable penalty ADMM algorithm titled alternating direction method with increasing penalty (ADMIP) is proposed to solve the following equivalent formulation for (3) using the variable splitting trick:

$$\min_{\hat{L},L,S \in \mathbb{R}^{m\times n}} \|\hat{L}\|_* + \rho\|S\|_1 \quad \text{s.t. } (L, S) \in \chi, \quad L = \hat{L}. \tag{24}$$

The augmented Lagrangian function of (24) can be written as

$$\mathcal{L}_\beta(\hat{L}, L, S; \Lambda) := \|\hat{L}\|_* + \rho\|S\|_1 - \langle \Lambda, \hat{L} - L\rangle + \frac{\beta}{2}\|\hat{L} - L\|_F^2.$$

Given a nondecreasing penalty parameter sequence $\{\beta^k\}_{k\in\mathbb{Z}_+}$, ADMIP updates the variables as follows:

$$\hat{L}^{k+1} := \operatorname*{argmin}_{\hat{L}} \mathcal{L}_{\beta^k}(\hat{L}, L^k, S^k; \Lambda^k) \tag{25a}$$

$$(L^{k+1}, S^{k+1}) := \operatorname*{argmin}_{(L,S)\in\mathcal{X}} \mathcal{L}_{\beta^k}(\hat{L}^{k+1}, L, S; \Lambda^k) \tag{25b}$$

$$\Lambda^{k+1} := \Lambda^k - \beta^k(\hat{L}^{k+1} - L^{k+1}). \tag{25c}$$

The step in (25a) requires computing a soft thresholding on the singular values of an $m \times n$ matrix and the step in (25b) requires an operation given in (23).

Under mild conditions on the penalty parameter sequence, Aybat and Iyengar show that the primal–dual ADMIP iterate sequence converges to an optimal primal–dual solution to the SPCP problem in (24); hence, $\{(L^k, S^k)\}_{k\in\mathbb{Z}_+}$ converges to an optimal solution to (3), and when constant penalty parameter is used as a special case, it can compute an ϵ-optimal solution within $\mathcal{O}(1/\epsilon)$ iterations, of which complexity is determined by an SVD. In particular, one needs the penalty parameter sequence $\{\beta^k\}_{k\in\mathbb{Z}_+}$ to be nondecreasing and to satisfy $\sum_k(\beta^k)^{-1} = +\infty$. The main advantages of adopting an increasing sequence of penalties are as follows.

1) The algorithm is robust in the sense that there is no need to search for problem data-dependent β^* that works well in practice.
2) The algorithm is likely to achieve primal feasibility faster.
3) The complexity of initial (transient) iterations can be controlled through controlling $\{\beta^k\}$. The main computational bottleneck in ADMIP is the SVD computation in (25a). Since the optimal L^* is of low-rank, and $L_k \to L^*$, eventually the SVD computations are likely to be very efficient. However, since the initial iterates in the transient phase of the algorithm may have large rank, the complexity of the SVD in the initial iterations can be quite large. To compute the solution to the subproblem in (25a), one does not need to compute singular values smaller than $1/\beta^k$; hence, initializing ADMIP with a small $\beta^0 > 0$ will significantly decrease the complexity of initial iterations through employing partial SVD computations, e.g., Lanczos-based methods such as PROPACK [50].

In [22], Aybat and Iyengar compared ADMIP against ASALM on both randomly generated synthetic problems and surveillance video foreground extraction problems. According to numerical results reported in [22], on the synthetic problems ASALM requires about twice as many iterations for convergence, while the total runtime for ASALM is considerably larger.

Aravkin *et al.* [23] proposed solving

$$\min \psi(L, S) \quad \text{s.t.} \quad \phi(\mathcal{A}(L, S) - M) \le \sigma \tag{26}$$

where $\mathcal{A} : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is a linear operator, $\phi : \mathbb{R}^{m\times n} \to \mathbb{R}$ is a smooth convex loss function, and ψ can be set to either one of the following functions:

$$\psi_{\text{sum}}(L, S) := \|L\|_* + \rho\|S\|_1$$
$$\psi_{\max}(L, S) := \max\{\|L\|_*, \rho_{\max}\|S\|_1\}.$$

$\rho, \rho_{\max} > 0$ are some given function parameters. Note that setting $\psi = \psi_{\text{sum}}$, $\phi(.) = \|.\|_F^2$, and $\mathcal{A}(L, S) = P_\Omega L + S$ in (26), one obtains the SPCP problem in (6). This approach offers advantages over the original SPCP formulation in terms of practical parameter selection. The authors make a case that although setting $\rho = 1/\sqrt{\max\{m, n\}}$ in (3) has theoretical justification as briefly discussed in the Introduction, many practical problems may violate the underlying assumptions in (8); in those cases, one needs to tune ρ via cross validation, and selecting $\rho_{\max}$ in $\psi_{\max}$ might be easier than selecting ρ in ψ_{sum}. Instead of solving (26) directly, a convex variational framework, accelerated with a "quasi-Newton" method, is proposed. In particular, Newton's method is used to find a root of the value function

$$\upsilon(\tau) := \min_{L,S\in\mathbb{R}^{m\times n}} \phi(\mathcal{A}(L, S) - M) - \sigma \quad \text{s.t.}, \psi(L, S) \le \tau \tag{27}$$

i.e., given $\sigma > 0$ compute $\tau^* = \tau(\sigma)$ such that $\upsilon(\tau^*) = 0$. According to results in [51], if the constraint in (26) is tight at an optimal solution, then there exists $\tau^* = \tau(\sigma)$ such that $\upsilon(\tau^*) = 0$ and the corresponding optimal solution to (27) is also optimal to (26). Within Newton's method for root finding, to compute the next iterate τ^{k+1}, one can compute the derivative of the value function at the current iterate τ^k as follows: $\upsilon'(\tau_k) = -\psi^\circ(\mathcal{A}^\top\nabla\phi(\mathcal{A}(L_k, S_k) - M))$, where ψ° denotes the polar gauge to ψ and (L_k, S_k) denotes the optimal solution to (27) at $\tau = \tau_k$. Aravkin *et al.* proposed a projected "Quasi-Newton" method to solve (27). According to numerical tests reported in [51], QN-max, the quasi-Newton method running on (26) with $\psi = \psi_{\max}$ and $\phi(.) = \|.\|_F^2$, is competitive with the state-of-the-art codes: ASALM [20], PSPG [21], and ADMIP [22].

In a recent work [24], Lin *et al.* considered the penalty formulation of the SPCP problem, which is equivalent to solving (3) for certain noise level $\sigma > 0$

$$\min \|L\|_* + \rho\|S\|_1 + \mu\|N\|_F^2 \quad \text{s.t. } L + S + N = M \tag{28}$$

where $\rho > 0$ is the sparsity tradeoff parameter and $\mu > 0$ is a suitable penalty parameter depending on the noise level $\sigma > 0$. The authors showed that the following three-block

ADMM for solving (28) globally converges for any penalty parameter $\beta > 0$:

$$\begin{aligned} L^{k+1} &:= \operatorname{argmin}_L \ \mathcal{L}_\beta(L, S^k, N^k; \Lambda^k) \\ S^{k+1} &:= \operatorname{argmin}_S \ \mathcal{L}_\beta(L^{k+1}, S, N^k; \Lambda^k) \\ N^{k+1} &:= \operatorname{argmin}_N \ \mathcal{L}_\beta(L^{k+1}, S^{k+1}, N; \Lambda^k) \\ \Lambda^{k+1} &:= \Lambda^k - \beta(L^{k+1} + S^{k+1} + N^{k+1} - M) \end{aligned} \tag{29}$$

where the augmented Lagrangian function is

$$\begin{aligned} &\mathcal{L}_\beta(L, S, N; \Lambda) \\ &:= \|L\|_* + \rho\|S\|_1 + \mu\|N\|_F^2 \\ &\quad - \langle \Lambda, L + S + N - M\rangle + \frac{\beta}{2}\|L + S + N - M\|_F^2. \end{aligned}$$

Note that the three subproblems in (29) are all easy to solve. Specifically, the L-subproblem corresponds to the proximal mapping of nuclear norm $\|L\|_*$, the S-subproblem corresponds to the proximal mapping of ℓ_1 norm, and the N-subproblem admits a very easy analytical solution.

The Frank–Wolfe method (also known as the conditional gradient method) [52] was revisited recently for solving large-scale machine learning problems [53], [54]. RPCA is a representative example that is suitable for the Frank–Wolfe method. Note that algorithms discussed above usually involve computing the proximal mapping of the nuclear norm, which is given by an SVD in (10). Computing full SVD for a large matrix in every iteration can be very time consuming. In contrast, the Frank–Wolfe method deals with nuclear norm in a much simpler manner, which only computes the largest singular value of a matrix in each iteration. The Frank–Wolfe method for solving RPCA was proposed by Mu *et al.* [25], who considered the penalized variant of RPCA (4). However, this problem cannot be directly solved by the Frank–Wolfe method, because the Frank–Wolfe method requires a bounded constraint set. Therefore, the authors further reformulated (4) to the following problem for properly chosen constants λ_L, λ_S, U_L, and U_S:

$$\begin{aligned} &\min \ \frac{1}{2}\|L + S - M\|_F^2 + \lambda_L t_L + \lambda_S t_S \\ &\text{s.t.}, \ \|L\|_* \le t_L \le U_L, \ \|S\|_1 \le t_S \le U_S. \end{aligned} \tag{30}$$

The Frank–Wolfe method iterates as follows:

$$\begin{aligned} G^k &:= L^k + S^k - M \\ (d_L^k, d_{t_L}^k) &:= \operatorname{argmin}_{\|L\|_* \le t_L \le U_L} \ \langle G^k, \ L\rangle + \lambda_L t_L \\ (d_S^k, d_{t_S}^k) &:= \operatorname{argmin}_{\|S\|_1 \le t_S \le U_S} \ \langle G^k, \ S\rangle + \lambda_S t_S \\ \gamma^k &:= 2/(k+2) \\ L^{k+1} &:= (1-\gamma^k)L^k + \gamma^k d_L^k \\ t_L^{k+1} &:= (1-\gamma^k)t_L^k + \gamma^k d_{t_L}^k \\ S^{k+1} &:= (1-\gamma^k)S^k + \gamma^k d_S^k \\ t_S^{k+1} &= (1-\gamma^k)t_S^k + \gamma^k d_{t_S}^k. \end{aligned} \tag{31}$$

It was shown in [25] that the two minimization subproblems in (31) are easy to solve. In particular, solving the subproblem for (d_L, d_{t_L}) requires only to compute the largest singular value and its corresponding singular vector of an $m \times n$ matrix. This is a big saving with computing the full SVD as required for computing the proximal mapping of the nuclear norm. As a result, the Frank–Wolfe method has better per-iteration complexity than the proximal gradient method and ADMM algorithms discussed above, and thus may have better scalability for very large-scale problems. On the other hand, as pointed out in [25], one clear disadvantage of the Frank–Wolfe method on (30) is that at every iteration only one entry of the sparse component is updated. This leads to very slow convergence in practice. Hence, Mu *et al.* [25] proposed combining Frank–Wolfe iterations with an additional proximal gradient step in S-block. In particular, they proposed that after Frank–Wolfe iterate (L_{k+1}, S_{k+1}) is computed, an extra proximal gradient step is computed and S-block is updated again. Moreover, the authors also showed that this hybrid method obtained by combining Frank–Wolfe and proximal gradient steps enjoys a sublinear convergence rate $O(1/k)$ similar to the Frank–Wolfe method given in (31).

As a special case of RPCA, one can consider that all columns of the low-rank matrix L are identical. That is, the given matrix M is a superposition of a special rank-one matrix L and a sparse matrix S. This special RPCA finds many interesting applications in practice such as video processing [55], [56] and bioinformatics [57]. For instance, in the background extraction of surveillance video, if the background is static, then the low-rank matrix L that corresponds to the background should have identical columns. As a result, the background and foreground can be separated by solving the following convex program:

$$\begin{aligned} &\min_{x,S} \ \|S\|_1 \\ &\text{s.t.} \ [x, x, \ldots, x] \circ \mathbf{E} + S = M \end{aligned} \tag{32}$$

where $[x, x, \ldots, x]$ denotes the $m \times n$ matrix with all columns being x, and $\mathbf{E}$ denotes the $m \times n$ matrix with all ones. Note that the optimal x of (32) corresponds to the static background for all frames and S corresponds to the moving foreground. The advantage of (32) is that it does not involve nuclear norm. As a result, SVD can be avoided when designing algorithms for solving it which makes the resulting algorithms very efficient. Yang *et al.* [58] adopted the similar idea and designed variants of the ADMM algorithm for solving a more general model where the sparsity function of S is allowed to be a nonconvex function. Convergence of the proposed ADMM was proved under the assumption of KL property [59], [60] being satisfied. We will discuss these topics in more details in the next section.

III. ALGORITHMS FOR NONCONVEX RELAXATIONS/VARIANTS OF RPCA

In this section, we discuss nonconvex relaxations and variants of RPCA given in (1) and algorithms for solving them. Some researchers aim to (approximately) solve RPCA in (1) directly without convexifying the rank function and/or the ℓ_0-norm. In [26], Zhou and Tao considered a variant of (1)

$$\min \ \|L+S-M\|_F^2 \quad \text{s.t. } \mathbf{rank}(L) \le \tau_r, \ \|S\|_0 \le \tau_s \tag{33}$$

where τ_r and τ_s are given parameters to control the rank of L and sparsity of S. The authors proposed the GoDec algorithm which alternatingly minimizes the objective function in one variable while fixing the other, which is a special case of alternating projection method analyzed in [61]. In particular, a naive version of GoDec algorithm iterates as follows:

$$\begin{aligned} L^{k+1} &:= \operatorname{argmin}_L \|L+S^k-M\|_F^2 \quad \text{s.t. } \mathbf{rank}(L) \le \tau_r \\ S^{k+1} &:= \operatorname{argmin}_S \|L^{k+1}+S-M\|_F^2 \quad \text{s.t. } \|S\|_0 \le \tau_s. \end{aligned} \tag{34}$$

The two subproblems correspond to two projections. Although the projection for S is easy, the projection for L requires computing a partial SVD, which may be time consuming when the matrix size is large. The authors proposed to use a low-rank approximation based on bilateral random projections to approximate this projection operation which can significantly speed up the computation. The authors showed that the iterate sequence converges to a local minimum provided that the initial point is close to some point in the intersection of the two manifolds $\{L \mid \mathbf{rank}(L) \le \tau_r\}$ and $\{S \mid \|S\|_0 \le \tau_s\}$. The convergence of GoDec follows from the results in [61].

In [28], Hintermüller and Wu considered a regularized version of (33)

$$\begin{aligned} \min_{L,S\in\mathbb{R}^{m\times n}} \ & \|L+S-M\|_F^2 + \frac{\rho}{2}\|L\|_F^2 \\ \text{s.t.} \quad & \mathbf{rank}(L) \le \tau_r, \quad \|S\|_0 \le \tau_s \end{aligned} \tag{35}$$

where $\tau_r, \tau_s > 0$ are given model parameters as in (33), and $0 \le \rho \ll 1$ is a given regularization parameter. An inexact alternating minimization method (R2PCP) on matrix manifolds is proposed to solve (35). The iterates L_{k+1} and S_{k+1} are computed as "inexact" solutions to subproblems $\min_L\{\|L+S_k-M\|_F^2+\rho\|L\|_F^2 : \mathbf{rank}(L) \le \tau_r\}$ and $\min_S\{\|L_{k+1}+S-M\|_F^2 : \|S\|_0 \le \tau_s\}$, respectively. Provided that a limit point of the iterate sequence exists, under some further restrictive technical assumptions, it is shown that first-order necessary optimality conditions are satisfied.

Note that the convex relaxation in (2) involves the nuclear norm $\|\cdot\|_*$ in the objective. Algorithms dealing with the nuclear norm (like the ones discussed in Section II) usually require to compute its proximal mapping, which then require an SVD. This can be very time consuming when the problem size is large, even when $\min\{m, n\}$ is in the order of thousands. This has motivated researchers to consider nonconvex relaxations of RPCA that avoid SVD calculations. One way to achieve SVD-free methods is to factorize the low-rank matrix $L \in \mathbb{R}^{m\times n}$ as a product of two low-rank matrices, i.e., factorize $L = UV^\top$, where $U \in \mathbb{R}^{m\times r}$, $V \in \mathbb{R}^{n\times r}$, and $r \ll \min\{m,n\}$ such that r is an upper bound on $\mathbf{rank}(L^o)$. This leads to many different nonconvex relaxations of RPCA.

In [27], Zhou and Tao considered a regularized version of (33)

$$\begin{aligned} \min_{U,V,S} \ & \rho_1\|S\|_1 + \left\|UV^\top+S-M\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{rank}(U) = \mathbf{rank}(V) \le \tau_r \end{aligned} \tag{36}$$

where $\rho_1 > 0$ and $\tau_r \in \mathbb{Z}_+$ such that $\tau_r \ge \mathbf{rank}(L^o)$. The authors propose a three-block alternating minimization algorithm, GreBsmo, for solving (36). The proposed algorithm lacks theoretical convergence guarantees; but, on the other hand, according to numerical results reported in [27], GreBsmo performs considerably better than both GoDec [26] and the inexact ALM method [17] (around 30–100 times faster than both) when applied to foreground extraction problems.

A nonconvex model of RPCA, similar to the one in [27], is studied in [11] and a GD method with proper initialization is proposed to solve it. The algorithm proposed in [11] has two phases and in both phases the objective is to reduce the function $Q(U,V,S) := \|UV^\top+S-M\|_F^2$. In the first phase, a sorting-based sparse estimator is used to generate a rough initial estimate S_0 to the unknown sparse target matrix S^o, and then U_0 and V_0 are generated via an SVD of $M-S_0$ such that $U_0V_0^\top$ forms a rough initial estimate to the unknown low-rank target matrix L^o. In the second phase, the algorithm alternatingly performs two operations: taking gradient steps for U and V, and computing a sparse estimator to adjust S. The sparse estimator is to guarantee that the fraction of nonzero entries in each column and row of S is bounded above so that the nonzero entries are spread out in S. The authors showed that the proposed two-phase algorithm recovers the target decomposition and linear convergence is achieved with proper initialization and step size, under the incoherence assumptions similar to (8). For a more detailed description of the assumptions and the results, see [11].

In [29], assuming that the data matrix M is observed indirectly through compressive measurements, Gu *et al.* considered the following variant of RPCA:

$$\min \ H(U,V,S) := \|A(UV^\top+S)-M\|_F^2 \ \text{ s.t. } \|S\|_0 \le \tau_s \tag{37}$$

where A is a sensing matrix. The alternating minimization algorithm proposed in [29] iterates as follows:

$$\begin{aligned} U^{k+1} &:= \operatorname{argmin}_U \ H(U, V^k, S^k) \\ V^{k+1} &:= \operatorname{argmin}_V \ H(U^{k+1}, V, S^k) \\ S^{k+1} &:= \operatorname{argmin}_S \ H(U^{k+1}, V^{k+1}, S) \quad \text{s.t. } \|S\|_0 \le \tau_s. \end{aligned} \tag{38}$$

It is noted that the U and V subproblems in (38) correspond to solving linear systems and the S-subproblem admits an easily computable closed-form solution. The authors showed that under incoherence assumption on L^o and A satisfying the restricted isometry property (RIP), (38) converges globally. However, note that $A = I$ does not satisfy the RIP condition, and therefore the convergence is not guaranteed for the RPCA problem in (1) for which $A = I$. A similar idea was also investigated in [62], assuming the RIP condition on the sensing matrix A, and it thus does not apply to the RPCA problem either.

In [30], Feng *et al.* considered the scenario such that the columns of the data matrix M are observed in an online fashion. This is suitable for many real applications, e.g., in surveillance video background separation. To handle this problem, the authors proposed a stochastic algorithm, which solves a nonconvex variant of RPCA

$$\min_{U,V,S} \frac{1}{2}\|UV^\top + S - M\|_F^2 + \frac{\rho_1}{2}(\|U\|_F^2 + \|V\|_F^2) + \rho_2\|S\|_1 \tag{39}$$

where ρ_1 and ρ_2 are some weight parameters. The formulation in (39) exploits the representation of the nuclear norm established in [8]. In particular, for any given $L \in \mathbb{R}^{m\times n}$ such that $\operatorname{rank}(L) \le r$, $\|L\|_*$ can be computed as follows:

$$\|L\|_* := \inf_{U\in\mathbb{R}^{m\times r}, V\in\mathbb{R}^{n\times r}} \left\{ \frac{1}{2}\|U\|_F^2 + \frac{1}{2}\|V\|_F^2 : UV^\top = L \right\}. \tag{40}$$

From (40), we know that (4) is equivalent to

$$\min_{U,V,S} \frac{\mu}{2}\|UV^\top + S - M\|_F^2 + \frac{1}{2}(\|U\|_F^2 + \|V\|_F^2) + \rho\|S\|_1. \tag{41}$$

As a result, (39) is a nonconvex reformulation of the penalized variant of RPCA in (4). For a given matrix $M = [M_1, \ldots, M_n] \in \mathbb{R}^{m\times n}$, solving (39) is the same as the following empirical risk minimization problem:

$$\min_U \frac{1}{n}\sum_{i=1}^n \ell(M_i, U) + \rho_1\|U\|_F^2 \tag{42}$$

where $\ell(M_i, U)$ is defined as

$$\begin{aligned} \ell(M_i, U) := \ & \min_{V_i\in\mathbb{R}^r, S_i\in\mathbb{R}^m} \frac{1}{2}\|UV_i + S_i - M_i\|_2^2 \\ & + \frac{\rho_1}{2}\|V_i\|_2^2 + \rho_2\|S_i\|_1. \end{aligned} \tag{43}$$

The empirical risk minimization (42) favors stochastic GD algorithm. Of course, every time to compute the gradient of $\ell(M_i, U)$, another minimization problem in (43) needs to be solved. Therefore, the algorithm proposed in [30] is an alternating minimization method with a subproblem for U being solved using stochastic GD. The authors showed that the proposed method converges to the correct low-dimensional subspace asymptotically under certain assumptions.

The following nonconvex variant of RPCA was proposed by Shen *et al.* in [31]:

$$\min_{U,V} \|UV^\top - M\|_1. \tag{44}$$

This simple reformulation can be viewed as a nonconvex reformulation of (2) but without any regularization terms on U and V. In particular, (44) can be rewritten as

$$\min \|S\|_1 \quad \text{s.t. } S + UV^\top = M. \tag{45}$$

Shen *et al.* [31] proposed an ADMM algorithm (named LMafit) for solving (45). By associating a Lagrange multiplier Λ to the constraint, the augmented Lagrangian function for (45) can be written as

$$\begin{aligned} \mathcal{L}_\beta(U, V, S; \Lambda) := \ & \|S\|_1 - \langle \Lambda, UV^\top + S - M\rangle \\ & + \frac{\beta}{2}\|UV^\top + S - M\|_F^2 \end{aligned}$$

where $\beta > 0$ is a penalty parameter. The nonconvex ADMM for solving (45) iterates the updates as follows:

$$\begin{aligned} U^{k+1} &:= \operatorname{argmin}_U \ \mathcal{L}_\beta(U, V^k, S^k; \Lambda^k) \\ V^{k+1} &:= \operatorname{argmin}_V \ \mathcal{L}_\beta(U^{k+1}, V, S^k; \Lambda^k) \\ S^{k+1} &:= \operatorname{argmin}_S \ \mathcal{L}_\beta(U^{k+1}, V^{k+1}, S; \Lambda^k) \\ \Lambda^{k+1} &:= \Lambda^k - \beta(U^{k+1}V^{k+1\top} + S^{k+1} - M). \end{aligned} \tag{46}$$

Note that all three subproblems in (46) are easy to solve. In particular, the U and V-subproblems correspond to solving linear systems, and the S-subproblem corresponds to the soft-shrinkage operation of the ℓ_1-norm (11). However, this nonconvex ADMM lacks convergence guarantees.

In [32], Jiang *et al.* studied some variants of the conditional gradient method and ADMM for solving nonconvex and nonsmooth optimization problems. Consider a general nonconvex optimization problem

$$\min_x \ f(x) + r(x) \quad \text{s.t. } x \in \mathcal{X} \tag{47}$$

where f is smooth and possibly nonconvex, $r(x)$ is convex and nonsmooth, and $\mathcal{X}$ is a convex compact set. Moreover, f satisfies the so-called Hölder condition

$$f(y) \le f(x) + \nabla f(x)^\top(y - x) + \frac{\gamma}{2}\|y - x\|_p^p, \quad \forall x, y \in \mathcal{X}$$

where $p > 1$ and $\gamma > 0$. The definition of the ϵ-stationary solution given in [32] is as follows.

Definition 1: $x \in \mathcal{X}$ is called an ϵ-stationary solution ($\epsilon \geq 0$) for (47) if the following holds:

$$\Psi_{\mathcal{X}}(x) := \inf_{y \in \mathcal{X}} \{\nabla f(x)^\top (y - x) + r(y) - r(x)\} \geq -\epsilon.$$

The authors commented that this definition is stronger than the one used by Ghadimi *et al.* in [63]. Now we briefly discuss how to apply the algorithms analyzed in [32] to solve nonconvex RPCA variants. Consider the nonconvex RPCA variant given in (39), then the generalized conditional gradient method proposed in [32] can be customized to solve (39). Since the generalized conditional gradient method requires a compact constraint set, one can equivalently reformulate (39) in the following form:

$$\begin{aligned} \min_{U,V,S} \quad & f(U,V,S) + \rho_2 \|S\|_1 \\ \text{s.t.} \quad & \max\{\|U\|_F,\ \|V\|_F\} \leq \frac{1}{\sqrt{\rho_1}} \|M\|_F \\ & \|S\|_1 \leq \frac{1}{2\rho_2} \|M\|_F^2 \end{aligned} \tag{48}$$

where $f(U,V,S) := (1/2)\|UV^\top + S - M\|_F^2 + (\rho_1/2)(\|U\|_F^2 + \|V\|_F^2)$ denotes the smooth part of the objective function. It is easy to see that ∇f is Lipschitz continuous. Let $\gamma > 0$ denote the Lipschitz constant.

At the kth iteration of the generalized conditional gradient method [32], implemented on (48), one needs to solve the following subproblem:

$$\begin{aligned} \min_{U,V,S} \ & \langle \nabla_U f(U^k, V^k, S^k), U \rangle + \langle \nabla_V f(U^k, V^k, S^k), V \rangle \\ & + \langle \nabla_S f(U^k, V^k, S^k), S \rangle + \rho_2 \|S\|_1 \\ \text{s.t.} \ & \max\{\|U\|_F,\ \|V\|_F\} \leq \frac{1}{\sqrt{\rho_1}} \|M\|_F \\ & \|S\|_1 \leq \frac{1}{2\rho_2} \|M\|_F^2. \end{aligned} \tag{49}$$

Suppose $(\bar{U}^k, \bar{V}^k, \bar{S}^k)$ denotes the solution of (49), then a typical iteration of the generalized conditional gradient method is given as follows:

$$\begin{cases} \text{Solve} & (49) \text{ to obtain } (\bar{U}^k, \bar{V}^k, \bar{S}^k) \\ \Delta U^k & := \bar{U}^k - U^k \\ \Delta V^k & := \bar{V}^k - V^k \\ \Delta S^k & := \bar{S}^k - S^k \\ \Delta^k & := [\Delta U^k, \Delta V^k, \Delta S^k] \\ \alpha_k & := \text{argmin}_{\alpha \in [0,1]} \alpha(\langle \nabla f(U^k, V^k, S^k), \Delta^k \rangle \\ & \quad + \frac{\alpha^2 \gamma}{2} \|\Delta^k\|_F^2 + (1-\alpha)\rho_2 \|[U^k, V^k, S^k]\|_1 \\ & \quad + \alpha \rho_2 \|[\bar{U}^k, \bar{V}^k, \bar{S}^k]\|_1 \\ U^{k+1} & := (1-\alpha_k) U^k + \alpha_k \bar{U}^k \\ V^{k+1} & := (1-\alpha_k) V^k + \alpha_k \bar{V}^k \\ S^{k+1} & := (1-\alpha_k) S^k + \alpha_k \bar{S}^k. \end{cases} \tag{50}$$

Note that the generalized conditional gradient method in [32] involves a line search step for computing α_k as shown in (50), which can be efficiently computed. It is shown in [32] that the generalized conditional gradient method in (50) can compute an ϵ-stationary solution of (48) in $O(1/\epsilon^2)$ iterations.

Jiang *et al.* [32] also proposed some ADMM variants that can solve various nonconvex RPCA formulations, and the authors provided a convergence rate analysis to compute an ϵ-stationary solution. The definition of ϵ-stationarity employed to analyze the ADMM algorithm is given in Definition 2. We here discuss the ADMM-g algorithm in [32] which can solve the following RPCA variant:

$$\begin{aligned} \min \ & \frac{1}{2}\|L - UV^\top\|_F^2 + \frac{\rho_1}{2}(\|U\|_F^2 + \|V\|_F^2) \\ & + \rho_2 \|S\|_1 + \rho_3 \|N\|_F^2 \\ \text{s.t.} \ & L + S + N = M. \end{aligned} \tag{51}$$

This is a nonsmooth and nonconvex problem with five block variables L, S, N, U, and V, and it can be viewed as a variant of (39) with linear constraints. Treating N as the last block variable, a typical iteration of ADMM-g for solving (51) iterates as follows:

$$\begin{aligned} L^{k+1} &:= \text{argmin}_L \ \tilde{\mathcal{L}}_{L^k}(L, U^k, V^k, S^k, N^k; \Lambda^k) \\ U^{k+1} &:= \text{argmin}_U \ \tilde{\mathcal{L}}_{U^k}(L^{k+1}, U, V^k, S^k, N^k; \Lambda^k) \\ V^{k+1} &:= \text{argmin}_V \ \tilde{\mathcal{L}}_{V^k}(L^{k+1}, U^{k+1}, V, S^k, N^k; \Lambda^k) \\ S^{k+1} &:= \text{argmin}_S \ \tilde{\mathcal{L}}_{S^k}(L^{k+1}, U^{k+1}, V^{k+1}, S, N^k; \Lambda^k) \\ N^{k+1} &:= N^k - \eta \nabla_N \mathcal{L}(L^{k+1}, U^{k+1}, V^{k+1}, S^{k+1}, N; \Lambda^k) \\ \Lambda^{k+1} &:= \Lambda^k - \beta(L^{k+1} + S^{k+1} + N^{k+1} - M) \end{aligned} \tag{52}$$

where $\eta > 0$ is a step size and the augmented Lagrangian function $\mathcal{L}$ is defined as

$$\begin{aligned} & \mathcal{L}(L, U, V, S, N; \Lambda) \\ & := \frac{1}{2}\|L - UV^\top\|_F^2 + \frac{\rho_1}{2}(\|U\|_F^2 + \|V\|_F^2) + \rho_2 \|S\|_1 \\ & \quad + \rho_3 \|N\|_F^2 - \langle \Lambda, L + S + N - M \rangle \\ & \quad + \frac{\beta}{2}\|L + S + N - M\|_F^2 \end{aligned}$$

and $\tilde{\mathcal{L}}$ denotes $\mathcal{L}$ plus a proximal term. For example, $\tilde{\mathcal{L}}_{L^k}$ is defined as

$$\begin{aligned} & \tilde{\mathcal{L}}_{L^k}(L, U^k, V^k, S^k, N^k; \Lambda^k) \\ & := \mathcal{L}(L, U^k, V^k, S^k, N^k; \Lambda^k) + \frac{1}{2}\|L - L^k\|_H^2 \end{aligned}$$

where H denotes a prespecified positive-definite matrix which needs to satisfy certain conditions to guarantee the convergence of the method as stated in (52). It is noted that the last block variable N is treated specially. It is not updated by minimizing the augmented Lagrangian function, but by taking a gradient step on it. The results in [32] indicate that ADMM-g (52) finds an ϵ-stationary solution for (51) in no more than $O(1/\epsilon^2)$ iterations. Since (51)

is a constrained problem, the definition of its ϵ-stationary solution is different from the one in Definition 1. Here we briefly discuss how it is defined for constrained problems in [32]. We consider the following constrained nonsmooth and nonconvex problem:

$$\begin{aligned} \min\ & f(x_1,\ldots,x_p)+\sum_{i=1}^{p-1} r_i(x_i) \\ \text{s.t.}\ & \sum_{i=1}^{p-1} A_i x_i + x_p = b \\ & x_i \in \mathcal{X}_i, \quad i=1,\ldots,p-1 \end{aligned} \tag{53}$$

where $x_i \in \mathbb{R}^{n_i}$, f is differentiable and possibly nonconvex, each r_i is possibly nonsmooth and nonconvex, and each $\mathcal{X}_i$ is a convex set. Note again that the last block variable x_p is treated differently, which is needed in the analysis of convergence rate. The ϵ-stationary solution to (53) is defined as follows.

Definition 2: $(x_1^*,\ldots,x_p^*) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_{p-1} \times \mathbb{R}^{n_p}$ is called an ϵ-stationary solution to (53), if there exists λ^* such that the following holds for any $(x_1,\ldots,x_p) \in \mathcal{X}_1 \times \ldots \times \mathcal{X}_{p-1} \times \mathbb{R}^{n_p}$:

$$\begin{aligned} (x_i - x_i^*)^\top (g_i^* + \nabla_i f(x_1^*,\ldots,x_p^*) - A_i^\top \lambda^*) \geq -\epsilon & \\ i=1,\ldots,p-1 & \\ \|\nabla_p f(x_1^*,\ldots,x_{p-1}^*,x_p^*) - \lambda^*\| \leq \epsilon & \\ \|\sum_{i=1}^{p-1} A_i x_i^* + x_p^* - b\| \leq \epsilon & \end{aligned}$$

where g_i^* is a general subgradient of r_i at point x_i^*. This set of inequalities can be viewed as a perturbed KKT system.

It is also interesting to note that (51) is equivalent to the following unconstrained problem, and thus can be solved by block coordinate descent method (BCD):

$$\begin{aligned} \min\ & \frac{1}{2}\|M-S-N-UV^\top\|_F^2 + \frac{\rho_1}{2}(\|U\|_F^2+\|V\|_F^2) \\ & + \rho_2\|S\|_1 + \rho_3\|N\|_F^2. \end{aligned} \tag{54}$$

Most existing BCD-type algorithms for solving nonconvex problems lack convergence rate analysis. In [32], Jiang *et al.* proposed a proximal BCD method that can find an ϵ-stationary solution in $O(1/\epsilon^2)$ iterations. Denoting the objective function in (54) as $F(U,V,S,N)$, the proximal BCD given in [32] for solving (54) updates the variables as follows:

$$\begin{aligned} U^{k+1} &:= \operatorname{argmin}_U F(U,V^k,S^k,N^k) + \frac{1}{2}\|U-U^k\|_H^2 \\ V^{k+1} &:= \operatorname{argmin}_V F(U^{k+1},V,S^k,N^k) + \frac{1}{2}\|V-V^k\|_H^2 \\ S^{k+1} &:= \operatorname{argmin}_S F(U^{k+1},V^{k+1},S,N^k) + \frac{1}{2}\|S-S^k\|_H^2 \\ N^{k+1} &:= \operatorname{argmin}_N F(U^{k+1},V^{k+1},S^{k+1},N) + \frac{1}{2}\|N-N^k\|_H^2 \end{aligned} \tag{55}$$

where H denotes a prespecified positive-definite matrix.

IV. PRELIMINARY NUMERICAL EXPERIMENTS

In this section, we provide some elementary numerical results of different algorithms for solving RPCA. We selected eight different solvers, five for solving convex problems: IALM [17], ADM [18], ADMIP [22], fastRPCA-max [23], and fastRPCA-sum [23], and three for solving nonconvex problems: LMafit [31], R2PCP [28], and GD [11]. We tested their performance on some standard synthetic data used in many RPCA papers. The synthetic data were generated in the following manner.

1) $L^o = UV^\top$ such that $U \in \mathbb{R}^{n\times r}$ and $V \in \mathbb{R}^{n\times r}$ for $r = c_r n$ and $c_r \in \{0.05, 0.1\}$. Moreover, $U_{ij} \sim \mathcal{N}(0,1)$ and $V_{ij} \sim \mathcal{N}(0,1)$ for all i,j are independent standard Gaussian variables.
2) $\Omega \subset \{(i,j) : 1 \leq i,j \leq n\}$ was chosen uniformly at random such that its cardinality $|\Omega| = c_p n^2$ and $c_p \in \{0.05, 0.1\}$.
3) $S_{ij}^o \sim \mathcal{U}[-\sqrt{8r/\pi}, \sqrt{8r/\pi}]$ for all $(i,j) \in \Omega$ are independent uniform random variables.
4) $N_{ij}^o \sim \varrho\mathcal{N}(0,1)$ for all i,j are independent Gaussian variables, where for a given signal-to-noise ratio (SNR) of M, ϱ is computed from

$$\begin{aligned} \text{SNR}(M) &= 10\log_{10}\left(\frac{\mathbb{E}[\|L^o+S^o\|_F^2]}{\mathbb{E}[\|N^o\|_F^2]}\right) \\ &= 10\log_{10}\left(\frac{c_r n + c_s\frac{8r}{3\pi}}{\varrho^2}\right) \end{aligned}$$

and $\text{SNR}(M) \in \{50\text{ dB}, 100\text{ dB}\}$.
5) The data matrix $M = L^o + S^o + N^o$.

Note that the nonzero entries of the sparse component and the entries of the low-rank component have approximately the same magnitude in expectation. Indeed, for $n \gg 1$, $L_{ij}^o \approx \sqrt{r}\mathcal{N}(0,1)$; hence, $\mathbb{E}[L_{ij}^o] = \sqrt{2r/\pi}$ for all i,j and $\mathbb{E}[S_{ij}^o] = \sqrt{2r/\pi}$ for $(i,j) \in \Omega$. We created ten random problems of size $n \in \{500, 1500\}$, i.e., $M \in \mathbb{R}^{n\times n}$, for each of the two choices of $\text{SNR}(M)$, c_r and c_p, using the procedure described above. We plot the figures showing the averaged relative errors of the iterates over ten runs versus CPU times (in seconds) in Fig. 1, where the relative error of (L,S) is defined as

$$\text{Error}(L,S) := \frac{\|L-L^o\|_F}{\|L^o\|_F} + \frac{\|S-S^o\|_F}{\|S^o\|_F}.$$

For all the eight algorithms, we used their default stopping criteria and default parameters setting if the output is of good quality; otherwise, we tuned some parameters so that the algorithm becomes competitive for our experimental setting. It is worth emphasizing that fastRPCA-sum and fastRPCA-max solve (27) with $\psi = \psi_{\text{sum}}$ and $\psi = \psi_{\max}$, respectively; and $\rho_{\max}$, the tradeoff parameter for $\psi_{\max}$, is set to $\|L^o\|_*/\|S^o\|_1$, i.e., this model needs an oracle that provides an ideal separation to tune the parameter.

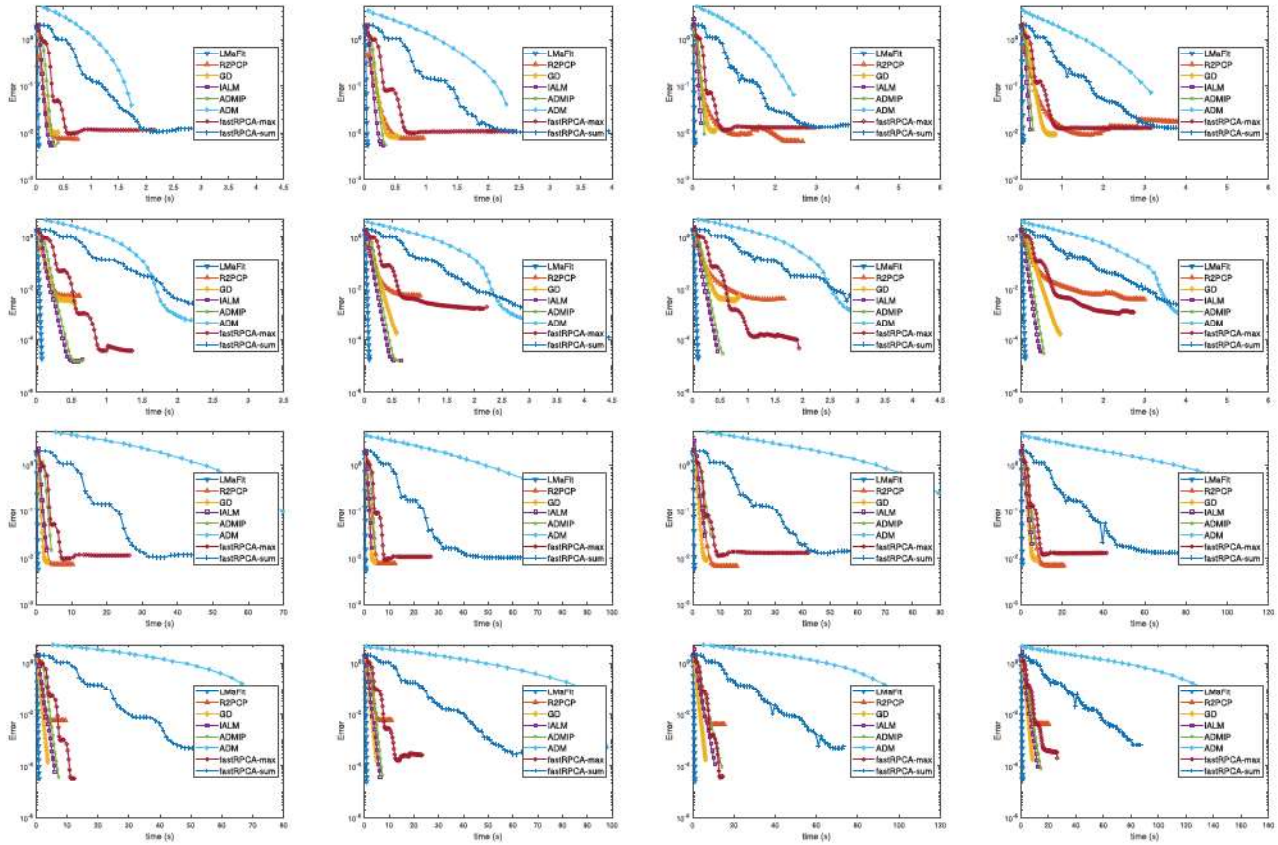

Fig. 1. *Comparison of different algorithms. The first row shows the comparison results for n=500, SNR = 50 dB, and (c_r, c_p)=(0.05,0.05), (0.05,0.1), (0.1,0.05), (0.1,0.1), respectively, from left to right. The second row corresponds to n=500, SNR = 100 dB; the third row corresponds to n=1500, SNR = 50 dB; and the fourth row corresponds to n=1500, SNR = 100 dB. All are with (c_r, c_p) set the same as the first row from left to right. We remark again that these comparison results only reflect the performance of the algorithms under the current setting of parameters, stopping criteria, initial points, input data, and so on. For other settings and data, the performance can be very different. See Remark 4.1.*

Remark 4.1 We remark that comparing different algorithms for solving RPCA is not an easy task for the following reasons. i) The algorithms are designed for solving related but different formulations. For example, IALM solves (2) and LMafit solves (45), so it is difficult to compare which algorithm is better. ii) The performance of all the algorithms depends on the parameter settings, initial points, and data structures. For example, from Fig. 1 we see that LMafit consistently outperforms fastRPCA-sum. However, this is only based on the data and parameters we tested. For other data sets, initial points, and parameter settings, fastRPCA-sum may be better than LMafit.

V. FUTURE DIRECTIONS

Although (2) is a convex problem with nice statistical properties, $\|\cdot\|_*$ causes problem for large-scale instances as standard algorithms for (2) cannot easily exploit multicore or multiserver computing environments as (2) is not amenable to distributed optimization. Some future work in this direction is related to the nonconvex formulation which uses an equivalent representation of $\|\cdot\|_*$ by Recht *et al.* [8] as given in (40). For instance, given some $\nu > 0$, consider $g_\nu : \mathbb{R}^{m\times n} \to \mathbb{R}$ defined in (15). Given a data matrix $M \in \mathbb{R}^{m\times n}$, using g_ν, we can formulate a smooth nonconvex optimization problem

$$\min_{U\in\mathbb{R}^{m\times r}, V\in\mathbb{R}^{n\times r}} \Psi_\nu(U,V) \triangleq \frac{1}{2}\|U\|_F^2 + \frac{1}{2}\|V\|_F^2 + \rho g_\nu(UV^\top - M) \quad (56)$$

where $\rho > 0$ and $\mathbb{Z}_+ \ni r \geq \text{rank}(L^\circ)$ are given parameters. Here, one can use the PALM algorithm [64] to generate a sequence that converges to a critical point of Ψ_ν, which is a KL function; also see [32] and [65] for some other related work on nonconvex optimization.

Note that instead of solving the smooth approximation given in (56), it is preferable to solve the following nonconvex formulation, which is equivalent to (2):

$$\min_{U\in\mathbb{R}^{m\times r}, V\in\mathbb{R}^{n\times r}} \frac{1}{2}\|U\|_F^2 + \frac{1}{2}\|V\|_F^2 + \rho\|UV^\top - M\|_1. \quad (57)$$

To the best of authors' knowledge, there do not exist efficient methods with strong convergence guarantees to solve (57).

Note that the third term in (57) is a composite function of the form $g(h(\cdot))$ where g is a nonsmooth convex function such that g is Lipschitz continuous, and h is a differentiable function such that its Jacobian h' is Lipschitz. Thus, one possible direction is to design the trust region algorithm for (57); see [66, Sec. 7.7]. As an alternative to the trust region algorithm, one might also consider the augmented Lagrangian (AL) method. It is known that for constrained nonconvex problems, provided that the second-order KKT conditions hold, the AL will have a saddle point for the penalty parameter chosen sufficiently large; therefore, the duality gap encountered in Lagrangian formulations does not pose a problem for augmented Lagrangian-based methods, thus, AL methods might prove useful to establish convergence to local minima when initialized sufficiently close to the local minimum [66], [67].

Due to the close relationship between low-rank matrix completion and RPCA, some algorithms for solving low-rank matrix completion problems can possibly be extended to solve variants of RPCA. New methods based on manifold optimization have been recently studied for solving low-rank matrix completion problems; see, e.g., [68]–[72]. It is noted that all these works consider a matrix completion variant/reformulation which is a manifold optimization problem with a smooth objective function. For example, assume that the matrix M is observed partially, i.e., only entries that in a subset Ω are observed, the low-rank matrix completion model considered in [70] is

$$\min_{U\in\mathcal{G}(m,r),V\in\mathbb{R}^{n\times r}} \frac{1}{2}\sum_{(i,j)\in\Omega} C_{ij}^2((UV^\top)_{ij}-M_{ij})^2+\frac{\mu^2}{2}(UV^\top)_{ij}^2 \tag{58}$$

where $\mathcal{G}(m,r)$ denotes Grassmann manifold, C_{ij} denotes some weighting parameter, and $\mu>0$ is a penalty parameter. It is noted that (58) is a manifold optimization problem with a smooth objective function. Many existing algorithms can be used to solve a manifold optimization problem with a smooth objective, for example, the Riemannian gradient method [73], the Riemannian trust region method [70], the Riemannian conjugate gradient method [72], and so on.

However, it is more challenging to design algorithms for manifold optimization reformulations of RPCA variants. The reason is that RPCA variants always involve nonsmooth terms in the objective. In fact, all RPCA variants we discussed so far involve $\|S\|_1$ in the objective. As a result, any manifold optimization reformulation of RPCA variants will involve the nonsmooth ℓ_1 term $\|S\|_1$ as well, unless one can bear with smoothing it, which can potentially degrade the sparsity of S. Algorithms for solving the manifold optimization problem with a nonsmooth objective function have been very limited, and most of them lack convergence guarantees. Nonetheless, some of these algorithms can still be adopted to solve manifold optimization reformulations of RPCA variants, although their efficiency in practice needs further investigations. For instance, Hintermüller and Wu [28] and Podosinnikova *et al.* [74] propose optimization methods on matrix manifolds. In particular, in [74], Podosinnikova *et al.* proposed a new RPCA model by minimizing the trimmed reconstruction error, which reduces to minimizing a nonsmooth function over the Stiefel manifold. The method lacks theoretical convergence guarantees such as convergence to a critical point. That said, Podosinnikova *et al.* [74] numerically demonstrate that their method exhibits good empirical recovery and it is competitive against other nonconvex formulations and convex optimization-based methods. One simple manifold optimization reformulation of RPCA is given as follows:

$$\min_{U\in\mathcal{S}_t(m,r),V\in\mathbb{R}^{n\times r}} \|UV^\top-M\|_1 \tag{59}$$

where $\mathcal{S}_t(m,r)$ denotes Stiefel manifold. The advantages of (59) are as follows: i) it does not involve the nuclear norm and thus avoids SVD; and ii) the sizes of U and V are $m\times r$ and $n\times r$, respectively, which are much smaller than the size of M when $r\ll\min(m,n)$. One may also note that (59) differs from (44) only with the Stiefel manifold constraint. The drawback of (44) is that its optimal solution (U^*,V^*) is not unique, because $(U^*W,V^*W^{-\top})$ is also optimal for any invertible matrix $W\in\mathbb{R}^{r\times r}$. This drawback is fixed nicely in (59). There are several ways to solve (59). For example, one can reformulate (59) as the following one and then apply ADMM to solve it:

$$\begin{aligned}&\min\ \|S\|_1\\ &\text{s.t., } S+UV^\top=M,\quad U\in\mathcal{S}_t(m,r).\end{aligned} \tag{60}$$

The ADMM iterates the updates as follows:

$$\begin{aligned}U^{k+1}&:=\operatorname{argmin}_U\ \mathcal{L}_\beta(U,V^k,S^k;\Lambda^k),\quad \text{s.t., } U\in\mathcal{S}_t(m,r)\\ V^{k+1}&:=\operatorname{argmin}_V\ \mathcal{L}_\beta(U^{k+1},V,S^k;\Lambda^k)\\ S^{k+1}&:=\operatorname{argmin}_S\ \mathcal{L}_\beta(U^{k+1},V^{k+1},S;\Lambda^k)\\ \Lambda^{k+1}&:=\Lambda^k-\beta(U^{k+1}V^{k+1^\top}+S^{k+1}-M)\end{aligned} \tag{61}$$

where the augmented Lagrangian function is defined as

$$\begin{aligned}\mathcal{L}_\beta(U,V,S;\Lambda):&=\|S\|_1-\langle UV^\top+S-M\rangle\\ &\quad+\frac{\beta}{2}\|UV^\top+S-M\|_F^2.\end{aligned}$$

The U-subproblem in (61) is a smooth manifold optimization problem and can be solved by existing methods [73]. This method should be very efficient, but the main issue is under what kind of conditions it is guaranteed to converge.

Zhang *et al.* studied some ADMM variants for Riemannian manifold optimization in [75], which can be used to solve manifold optimization reformulations of

some RPCA variants. We here briefly discuss this work. We consider the following RPCA variant, which minimizes a nonsmooth function over Stiefel manifold:

$$\begin{aligned} &\min \ \frac{1}{2}\|L-UV^\top\|_F^2+\rho\|S\|_1+\frac{\mu}{2}\|N\|_F^2 \\ &\text{s.t. } L+S+N=M \\ &\quad U\in\mathcal{S}_t(m,r) \end{aligned} \tag{62}$$

where $\rho>0$ and $\mu>0$ are tradeoff parameters. One of the ADMM variants for solving (62) proposed in [75] iterates the updates as follows:

$$\begin{aligned} L^{k+1} :&= \operatorname{argmin}_L \ \tilde{\mathcal{L}}_{L^k}(L,U^k,V^k,S^k,N^k;\Lambda^k), \\ &\quad \text{s.t.}, U\in\mathcal{S}_t(m,r) \\ U^{k+1} :&= \operatorname{argmin}_U \ \tilde{\mathcal{L}}_{U^k}(L^{k+1},U,V^k,S^k,N^k;\Lambda^k) \\ V^{k+1} :&= \operatorname{argmin}_V \ \tilde{\mathcal{L}}_{V^k}(L^{k+1},U^{k+1},V,S^k,N^k;\Lambda^k) \\ S^{k+1} :&= \operatorname{argmin}_S \ \tilde{\mathcal{L}}_{S^k}(L^{k+1},U^{k+1},V^{k+1},S,N^k;\Lambda^k) \\ N^{k+1} :&= N^k-\eta\nabla_N\mathcal{L}(L^{k+1},U^{k+1},V^{k+1},S^{k+1},N;\Lambda^k) \\ \Lambda^{k+1} :&= \Lambda^k-\beta(L^{k+1}+S^{k+1}+N^{k+1}-M) \end{aligned} \tag{63}$$

where $\eta>0$ is a step size, the augmented Lagrangian function $\mathcal{L}$ is defined as

$$\begin{aligned} &\mathcal{L}(L,U,V,S,N;\Lambda) \\ &:=\frac{1}{2}\|L-UV^\top\|_F^2+\rho_1\|S\|_1 \\ &\quad+\frac{\mu}{2}\|N\|_F^2-\langle\Lambda,L+S+N-M\rangle \\ &\quad X+\frac{\beta}{2}\|L+S+N-M\|_F^2 \end{aligned}$$

and $\tilde{\mathcal{L}}$ denotes $\mathcal{L}$ plus a proximal term. For example, $\tilde{\mathcal{L}}_{L^k}$ is defined as

$$\begin{aligned} &\tilde{\mathcal{L}}_{L^k}(L,U^k,V^k,S^k,N^k;\Lambda^k) \\ &:=\mathcal{L}(L,U^k,V^k,S^k,N^k;\Lambda^k)+\frac{1}{2}\|L-L^k\|_H^2 \end{aligned}$$

where H denotes a prespecified positive-definite matrix which needs to satisfy certain conditions to guarantee the convergence of (63). Zhang *et al.* showed in [75] that the algorithm described in (63) finds an ϵ-stationary solution to (62) in no more than $O(1/\epsilon^2)$ iterations under certain conditions on β, η, and H.

One thing that we need to note is that the term $(\mu/2)\|N\|_F^2$ in (62) is very crucial here. Without this squared term, the convergence results in [75] do not apply. For example, if one considers the following RPCA variant without the noisy term N:

$$\begin{aligned} &\min \ \frac{1}{2}\|L-UV^\top\|_F^2+\rho\|S\|_1 \\ &\text{s.t. } L+S=M \\ &\quad U\in\mathcal{S}_t(m,r) \end{aligned} \tag{64}$$

then the ADMM variants proposed in [75] are not guaranteed to converge if they are applied to solve (64). How to extend and generalize the results in [75] so that they can be used to solve other manifold optimization reformulations of RPCA variants definitely deserves more investigations.

It is known that the nuclear norm minimization problem can be equivalently written as an SDP; see [8]. Though the problem size of the SDP is larger than the original nuclear norm minimization problem, it is recently found that the facial reduction technique [76] can reduce the size of the SDP significantly. This idea has been explored in low-rank matrix completion [77] and RPCA [78]. In particular, in [78], Ma *et al.* showed that RPCA with partial observation

$$\min \ \mathbf{rank}(L)+\rho\|S\|_0, \ \text{s.t.}, \ P_\Omega(L+S-M)=0 \tag{65}$$

is equivalent to another optimization problem with semidefinite constraint. By applying the facial reduction technique to the semidefinite cone, the size of this reformulation can be significantly reduced, and then it can be solved very efficiently to high accuracy. Extending the facial reduction technique to other RPCA variants is an interesting topic for future research.

VI. CONCLUSION

In this paper, we gave a comprehensive review of algorithms for solving relaxations and variants of robust PCA. Algorithms for solving convex and nonconvex models were discussed. We elaborated in details on the applicability of the algorithms and their convergence behaviors. We also proposed several new directions in the hope that they may shed some light for future research in this area. □

Acknowledgments

The authors would like to thank three anonymous referees for their insightful and constructive comments that led to an improved version of this paper.

REFERENCES

[1] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 1, pp. 1–37, 2009.

[2] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.

[3] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[4] N. Shahid, V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Robust principal component

analysis on graphs," in *Proc. ICCV*, Dec. 2015, pp. 2812–2820.

[5] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM J. Optim.*, vol. 21, no. 2, pp. 572–596, 2011.

[6] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in Neural Information Processing Systems*, vol. 22, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Red Hook, NY, USA: Curran Associates, 2009, pp. 2080–2088. [Online]. Available: http://papers.nips.cc/paper/3704-robust-principal-component-analysis-exact-recovery-of-corrupted-low-rank-matrices-via-convex-optimization.pdf

[7] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Advances in Neural Information Processing Systems*, vol. 23, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Red Hook, NY, USA: Curran Associates, 2010, pp. 2496–2504. [Online]. Available: http://papers.nips.cc/paper/4005-robust-pca-via-outlier-pursuit.pdf

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

[9] Z. Zhou, J. Wright, X. Li, E. J. Candés, and Y. Ma, "Stable principal component pursuit," in *Proc. Int Symp. Inf. Theory*, 2010, pp. 1518–1522.

[10] H. Zhang, Y. Zhou, and Y. Liang, "Analysis of robust PCA via local incoherence," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[11] X. Yi, D. Park, Y. Chen, and C. Caramanis (2016). "Fast algorithms for robust PCA via gradient descent." [Online]. Available: https://arxiv.org/abs/1605.07784

[12] T. Zhang and Y. Yang (2017). "Robust PCA by manifold optimization." [Online]. Available: https://arxiv.org/abs/1708.00257

[13] P. Netrapalli, U. N. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain (2014). "Non-convex robust PCA." [Online]. Available: https://arxiv.org/abs/1410.7660

[14] T. Bouwmans and E. H. Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *Comput. Vis. Image Understand.*, vol. 122, pp. 22–34, May 2014.

[15] N. S. Aybat, "Algorithms for stable PCA," in *Handbook of Robust Low Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. Boca Raton, FL, USA: CRC Press, 2016.

[16] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," in *Proc. Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process.*, 2009.

[17] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," Univ. Illinois Urbana-Champaign, Champaign, IL, USA, Tech. Rep. UILU-ENG-09-2215, 2009.

[18] X. Yuan and J. Yang, "Sparse and low-rank matrix decomposition via alternating direction methods," *Pacific J. Optim.*, vol. 9, no. 1, pp. 167–180, 2013.

[19] D. Goldfarb, S. Ma, and K. Scheinberg, "Fast alternating linearization methods for minimizing the sum of two convex functions," *Math. Program. A*, vol. 141, nos. 1–2, pp. 349–382, 2013.

[20] M. Tao and X. Yuan, "Recovering low-rank and sparse components of matrices from incomplete and noisy observations," *SIAM J. Optim.*, vol. 21, no. 2, pp. 57–81, 2011.

[21] N. S. Aybat, D. Goldfarb, and S. Ma, "Efficient algorithms for robust and stable principal component pursuit problems," *Comput. Optim. Appl.*, vol. 58, no. 1, pp. 1–29, 2014.

[22] N. S. Aybat and G. Iyengar, "An alternating direction method with increasing penalty for stable principal component pursuit," *Comput. Optim. Appl.*, vol. 61, no. 3, pp. 635–668, 2015.

[23] A. Aravkin, S. Becker, V. Cevher, and P. Olsen, "A variational approach to stable principal component pursuit," in *Proc. 30th Conf. Uncertain. Artif. Intell. (UAI)*, Jun. 2014, paper EPFL-CONF-199542. [Online]. Available: https://arxiv.org/abs/1406.1089

[24] T. Lin, S. Ma, and S. Zhang, "Global convergence of unmodified 3-block ADMM for a class of convex minimization problems," *J. Sci. Comput.*, vol. 76, no. 1, pp. 69–88, Jul. 2018.

[25] C. Mu, Y. Zhang, J. Wright, and D. Goldfarb, "Scalable robust matrix recovery: Frank–Wolfe meets proximal methods," *SIAM J. Sci. Comput.*, vol. 38, no. 5, pp. A3291–A3317, 2016.

[26] T. Zhou and D. Tao, "GoDec: Randomized low-rank and sparse matrix decomposition in noisy case," in *Proc. ICML*, 2011.

[27] T. Zhou and D. Tao, "Greedy bilateral sketch, completion & smoothing," in *Proc. 16th Int. Conf. Artif. Intell. Statist.*, vol. 31, C. M. Carvalho and P. Ravikumar, Eds. Scottsdale, AZ, USA: PMLR, Apr./May 2013, pp. 650–658.

[28] M. Hintermüller and T. Wu, "Robust principal component pursuit via inexact alternating minimization on matrix manifolds," *J. Math. Imag. Vis.*, vol. 51, no. 3, pp. 361–377, 2015.

[29] Q. Gu, Z. Wang, and H. Liu, "Low-rank and sparse structure pursuit via alternating minimization," in *Proc. AISTATS*, 2016, pp. 600–609.

[30] J. Feng, H. Xu, and S. Yan, "Online robust PCA via stochastic optimization," in *Advances in Neural Information Processing Systems*, 2013.

[31] Y. Shen, Z. Wen, and Y. Zhang, "Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization," *Optim. Methods Softw.*, vol. 29, no. 2, pp. 239–263, 2014.

[32] B. Jiang, T. Lin, S. Ma, and S. Zhang (2016). "Structured nonconvex and nonsmooth optimization: Algorithms and iteration complexity analysis." [Online]. Available: https://arxiv.org/abs/1605.02408

[33] Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Program.*, vol. 140, no. 1, pp. 125–161, Aug. 2013.

[34] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[35] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," unpublished, 2008. [Online]. Available: http://www.mit.edu/~dimitrib/PTseng/papers/apgm.pdf

[36] P. L. Combettes and J.-C. Pesquet, "A Douglas–Rachford splitting approach to nonsmooth convex variational signal recovery," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 564–574, Dec. 2007.

[37] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

[38] J. Yang, Y. Zhang, and W. Yin, "A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 288–297, Apr. 2010.

[39] J. Yang and Y. Zhang, "Alternating direction algorithms for l1-problems in compressive sensing," *SIAM J. Sci. Comput.*, vol. 33, no. 1, pp. 250–278, 2011.

[40] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Comput. Math. Appl.*, vol. 2, no. 1, pp. 17–40, 1976.

[41] D. Gabay, "Applications of the method of multipliers to variational inequalities," in *Augmented Lagrangian Methods: Applications to the Solution of Boundary Value Problems*, M. Fortin and R. Glowinski, Eds. Amsterdam, The Netherlands: North-Holland, 1983.

[42] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*. Amsterdam, The Netherlands: North-Holland, 1983.

[43] P. L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM J. Numer. Anal.*, vol. 16, no. 6, pp. 964–979, 1979.

[44] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, no. 3, pp. 293–318, Jun. 1992.

[45] B. He and X. Yuan, "On the $O(1/n)$ convergence rate of Douglas–Rachford alternating direction method," *SIAM J. Numer. Anal.*, vol. 50, no. 2, pp. 700–709, 2012.

[46] R. D. C. Monteiro and B. F. Svaiter, "Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers," *SIAM J. Optim.*, vol. 23, no. 1, pp. 475–507, 2013.

[47] T. Lin, S. Ma, and S. Zhang, "On the sublinear convergence rate of multi-block ADMM," *J. Oper. Res. Soc. China*, vol. 3, no. 3, pp. 251–274, 2015.

[48] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Math. Program.*, vol. 155, nos. 1–2, pp. 57–79, 2016.

[49] N. S. Aybat, D. Goldfarb, and G. Iyengar (2011). "Fast first-order methods for stable principal component pursuit." [Online]. Available: https://arxiv.org/1105.2126

[50] R. M. Larsen, "Lanczos bidiagonalization with partial reorthogonalization," *DAIMI Rep.*, vol. 27, p. 537, Sep. 1998.

[51] A. Y. Aravkin, J. V. Burke, and M. P. Friedlander, "Variational properties of value functions," *SIAM J. Optim.*, vol. 23, no. 3, pp. 1689–1717, 2013.

[52] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Res. Logistics Quart.*, vol. 3, nos. 1–2, pp. 95–110, 1956.

[53] M. Jaggi, "Revisiting Frank–Wolfe: Projection-free sparse convex optimization," in *Proc. ICML*, 2013, pp. 427–435.

[54] M. Jaggi, "Sparse convex optimization methods for machine learning," Ph.D. dissertation, ETH Zurich, Zürich, Switzerland, Oct. 2011.

[55] X. Li, M. K. Ng, and X. Yuan, "Median filtering-based methods for static background extraction from surveillance video," *Numer. Linear Algebra Appl.*, vol. 22, no. 5, pp. 845–865, 2015.

[56] L. Li, P. Wang, Q. Hu, and S. Cai, "Efficient background modeling based on sparse representation and outlier iterative removal," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 26, no. 2, pp. 278–289, Feb. 2016.

[57] S. Ma *et al.*, "SPARCoC: A new framework for molecular pattern discovery and cancer gene identification," *PLoS ONE*, vol. 10, no. 3, p. e0117135, 2015.

[58] L. Yang, T. K. Pong, and X. Chen, "Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction," *SIAM J. Imag. Sci.*, vol. 10, pp. 74–110, 2017.

[59] K. Kurdyka, "On gradients of functions definable in o-minimal structures," *Annales l'Institut Fourier*, vol. 146, pp. 769–783, 1998.

[60] S. Łojasiewicz, "Une propriete topologique des sous-ensembles analytiquesreels," in *Les Équations aux Dérivées Partielles*. Paris: Editions du centre National de la Recherche Scientifique, 1963.

[61] A. S. Lewis and J. Malick, "Alternating projections on manifolds," *Math. Oper. Res.*, vol. 33, no. 1, pp. 216–234, Feb. 2008.

[62] A. E. Waters, A. C. Sankaranarayanan, and R. Baraniuk, "SpaRCS: Recovering low-rank and sparse matrices from compressive measurements," in *Proc. NIPS*, 2011, pp. 1089–1097.

[63] S. Ghadimi, G. Lan, and H. Zhang, "Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization," *Math. Program.*, vol. 155, no. 1, pp. 267–305, 2016.

[64] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Math. Program.*, vol.

146, nos. 1–2, pp. 459–494, 2014.
[65] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for L_1-subspace signal processing," *IEEE Trans. Signal Process.*, vol. 62, no. 19, pp. 5046–5058, Oct. 2014.
[66] A. P. Ruszczyński, *Nonlinear Optimization*, vol. 13. Princeton, CA, USA: Princeton Univ. Press, 2006.
[67] G. Cohen and D. L. Zhu, "Decomposition and coordination methods in large scale optimization problems: The nondifferentiable case and the use of augmented Lagrangians," *Adv. Large Scale Syst.*, vol. 1, pp. 203–266, 1984.
[68] R. H. Keshavan and S. Oh (2009). "Optspace: A gradient descent algorithm on the Grassman manifold for matrix completion." [Online]. Available: https://arxiv.org/0910.5260v2
[69] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Proc. Allerton Conf.*, 2010, pp. 704–711.
[70] N. Boumal and P.-A. Absil, "RTRMC: A Riemannian trust-region method for low-rank matrix completion," in *NIPS*, 2011, pp. 406–414.
[71] B. Vandereycken, "Low-rank matrix completion by Riemannian optimization," *SIAM J. Optim.*, vol. 23, pp. 1214–1236, Jun. 2013.
[72] K. Wei, J.-F. Cai, T. F. Chan, and S. Leung (2016). "Guarantees of Riemannian optimization for low rank matrix completion." [Online]. Available: https://arxiv.org/abs/1603.06610
[73] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, CA, USA: Princeton Univ. Press, 2008.
[74] A. Podosinnikova, S. Setzer, and M. Hein, "Robust PCA: Optimization of the robust reconstruction error over the stiefel manifold," in *Proc. GCPR*, 2014, pp. 121–131.
[75] J. Zhang, S. Ma, and S. Zhang (2017). "Primal-dual optimization algorithms over Riemannian manifolds: An iteration complexity analysis." [Online]. Available: https://arxiv.org/abs/1710.02236
[76] D. Drusvyatskiy and H. Wolkowicz, "The many faces of degeneracy in conic optimization," *Found. Trends Optim.*, vol. 3, no. 2, pp. 77–170, 2016.
[77] S. Huang and H. Wolkowicz, "Low-rank matrix completion using nuclear norm with facial reduction," *J. Global Optim.*, pp. 1–22, Nov. 2017.
[78] S. Ma, F. Wang, L. Wei, and H. Wolkowicz (2018). "Robust principal component analysis using facial reduction." [Online]. Available: http://www.optimization-online.org/DB_HTML/2018/03/6535.html

ABOUT THE AUTHORS

Shiqian Ma received the B.S. degree in computational mathematics from Peking University, Beijing, China, in 2003, the M.S. degree in computational mathematics from the Chinese Academy of Sciences, Beijing, China, in 2006, and the Ph.D. degree in operations research from Columbia University, New York, NY, USA, in 2011.

He joined the Department of Mathematics, University of California Davis, Davis, CA, USA, in 2017. Prior to that he was with the Chinese University of Hong Kong, Hong Kong, as an Assistant Professor. His current research interests include theory and algorithms for large-scale optimization and its applications in big data analytics, statistics, and machine learning.

Dr. Ma received the *Journal of the Operations Research Society of China* Excellent Paper Award in 2016, the INFORMS Optimization Society Best Student Paper Prize in 2010, honorable mention of INFORMS George Nicholson Student Paper Competition in 2011, and was one of the finalists of the 2011 IBM Herman Goldstine fellowship.

Necdet Serhat Aybat received the B.S. and M.S. degrees in industrial engineering from Bogazici University, Istanbul, Turkey, in 2003 and 2005, respectively, and the Ph.D. degree in operations research from Columbia University, New York, NY, USA, in 2011.

He joined the Department of Industrial Engineering, Pennsylvania State University, State College, PA, USA, in August 2011. His current research interests include developing first-order algorithms for large-scale convex optimization problems from diverse application areas, such as compressed sensing, matrix completion, convex regression, and distributed optimization.

Dr. Aybat was a runner-up of the INFORMS Computing Society Student Paper Award in 2010, and received the SIAM Student Paper Prize in 2011.