

Robust Principal Component Pursuit via Inexact Alternating Minimization on Matrix Manifolds*

Michael Hintermüller[†]

Tao Wu[‡]

Abstract

Robust principal component pursuit (RPCP) refers to a decomposition of a data matrix into a low-rank component and a sparse component. In this work, instead of invoking a convex-relaxation model based on the nuclear norm and the ℓ^1 -norm as is typically done in this context, RPCP is solved by considering a least-squares problem subject to rank and cardinality constraints. An inexact alternating minimization scheme, with guaranteed global convergence, is employed to solve the resulting constrained minimization problem. In particular, the low-rank matrix subproblem is resolved inexactly by a tailored Riemannian optimization technique, which favorably avoids singular value decompositions in full dimension. For the overall method, a corresponding q -linear convergence theory is established. The numerical experiments show that the newly proposed method compares competitively with a popular convex-relaxation based approach.

Keywords: Matrix decomposition, low-rank matrix, sparse matrix, image processing, alternating minimization, Riemannian manifold, optimization on manifolds.

AMS subject classification: 15A83, 53B21, 65K10, 90C30, 94A08.

1 Introduction

A typical approach in understanding big and complex data in many different application areas utilizes data decomposition additively splitting the given data into several components of respective low complexity. For this purpose, *robust principal component pursuit* (RPCP), introduced in [10], aims at recovering a low-rank component and a sparse component from a possibly noisy data matrix. The low-rank component often refers to a certain intrinsically low dimensional pattern in the data, while the sparse component corresponds to either grossly corrupted measurements or pattern-irrelevant data. In this sense, RPCP is more robust in practice than the classical principal component analysis. The RPCP and its variants have found various promising applications, particularly in image and signal processing; e.g. video surveillance [10], face recognition [17], texture modeling [35], video inpainting [16], audio separation [15], latent semantic indexing [25], etc.

Concerning the numerical solution of RPCP in the large-scale setting, a popular approach [10, 12] is to solve a “relaxed” convex program, where the rank functional is relaxed by the nuclear

*This research was supported by the Austrian Science Fund (FWF) through START project Y305 “Interfaces and Free Boundaries” and through SFB project F3204 “Mathematical Optimization and Applications in Biomedical Sciences”.

[†]Department of Mathematics, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany (hint@math.hu-berlin.de).

[‡]Institute for Mathematics and Scientific Computing, Karl-Franzens-University of Graz, Heinrichstrasse 36, A-8010 Graz, Austria (tao.wu@uni-graz.at).

norm, i.e. the sum of the singular values, and the cardinality function is relaxed by the ℓ^1 -norm, i.e. the sum of all entries in absolute values. In [10], it was proven that the convex-relaxation model provides the exact recovery with dominating probability given some mild assumptions on the underlying low-rank and sparse components. A somewhat more deterministic argument can be found in [12], where a sufficient condition for exact recovery, based on the notion of rank-sparsity incoherence, is invoked. This condition holds true with high probability for random low-rank and sparse components. Based on the convex-relaxation formulation, for the numerical solution of the associated minimization problem an augmented Lagrangian method (ALM) is utilized in [10]. A related work on ALM, improving efficiency of the method and expanding its scope with respect to applications, can be found in [31]. A list of works concerning numerical solvers relevant to the convex-relaxation approach is contained in [2]. Typically, at each iteration such solvers involve the computation of a singular value decomposition (SVD) in full dimension, which becomes highly expensive in large-scale applications. Acceleration of this SVD step can be possibly done via a Lancos-based partial SVD technique (see, e.g., [3] for an efficient implementation under Matlab) as suggested in [9, 31], but its practical efficiency largely relies on the properties of the target matrix of the SVD such as relatively low rank and/or fast matrix-vector multiplication. Finally, besides the convex-relaxation based approaches, we also mention a (nonconvex) factorization-based augmented Lagrangian alternating direction method for RPCP [34], for which an online code is available [1].

In this paper, we solve RPCP by formulating a (regularized) least-squares problem with rank and cardinality constraints; see (1) below. Then an alternating minimization scheme (AMS) is employed to seek a stationary point which satisfies the first-order necessary optimality condition. If each subproblem in AMS is solved exactly by global minimization (i.e. metric projection) along the iterations, then AMS essentially becomes a heuristic method of (generalized) alternating projection onto manifolds (see the Appendix and also [21]), which is known to be locally convergent for transversal manifolds. However, the convergence of this alternating projection method can be possibly spoiled by defective initial guesses, which calls upon proper globalization (or safeguard) strategies on AMS.

For this sake, we propose a general framework sufficient for AMS to converge globally, which is then activated algorithmically. In particular, the low-rank subproblem is solved inexactly by a Riemannian (manifold) optimization step such that SVDs in full dimension can be favorably avoided. We point out that Riemannian optimization is an active research area in its own right; see [5] and the references therein for an introduction on the subject and [8] for a miscellaneous toolbox available online. Concerning the applications of Riemannian optimization related to low-rank matrices, we refer to [18, 19, 29, 7, 24, 33] among other references which appeared very recently. Nevertheless, most of these papers, if not all, address the context of *low-rank matrix completion* [11] rather than RPCP, i.e. the sparse component is of no concern. In the present work, however, we include such a sparse component (in addition to the low-rank part) by embedding a tailored Riemannian optimization technique, namely the projected dogleg step, into the overall AMS. A q -linear convergence theory is established from the perspective of an inexact Newton method on the underlying matrix manifold. For the implementation of AMS, we also propose a heuristic trimming procedure which performs a proper tuning of the underlying rank and cardinality constraints. This procedure aims at automatically identifying the appropriate rank and sparsity of the two target components within the given data.

The remainder of the paper is organized as follows. In section 2, we formulate our variational model for RPCP and investigate the existence of a solution as well as the first-order optimality condition. The overall AMS and its convergence analysis are presented in detail in section 3. Section 4 concludes the paper with a series of numerical experiments on the proposed method,

including a comparison with a currently state-of-the-art augmented Lagrangian method.

2 Robust principal component pursuit

Let the observed data Z be composed in the following way:

$$Z = A_{true} + B_{true} + N,$$

where $A_{true} \in \mathcal{M}(r) = \{A \in \mathbb{R}^{m \times n} : \text{rank}(A) \leq r\}$, $B_{true} \in \mathcal{N}(s) = \{B \in \mathbb{R}^{m \times n} : \|B\|_0 \leq s\}$, and N is an m -by- n matrix of white Gaussian noise. Moreover, $\|\cdot\|_0$ denotes the number of nonzero entries of a matrix. In what follows, we omit the arguments r and s whenever their values stay constant in the context. Let the inner product $\langle \cdot, \cdot \rangle$ be defined as $\langle A, B \rangle = \text{trace}(A^\top B)$ for any $A, B \in \mathbb{R}^{m \times n}$ and $\|\cdot\|$ be the Frobenius norm. Throughout this paper, we assume that r and s are natural numbers such that $r \ll n \leq m$ and $s \ll mn$.

Our goal is to recover the matrices A_{true} and B_{true} by solving the following optimization problem:

$$\begin{aligned} & \text{minimize} && f(A, B) = \frac{1}{2}\|A + B - Z\|^2 + \frac{\mu}{2}\|A\|^2, \\ & \text{subject to} && (A, B) \in \mathcal{M} \times \mathcal{N}. \end{aligned} \tag{1}$$

Note that a quadratic regularization on A with $0 < \mu \ll 1$ is introduced into the objective in order to enforce the existence of solution, as provided by the following theorem.

Theorem 2.1. *The variational problem (1) admits a global minimizer.*

Proof. Let $(A^k, B^k) \in \mathcal{M} \times \mathcal{N}$ form an infimizing sequence for (1), i.e.

$$\lim_{k \rightarrow \infty} f(A^k, B^k) = \inf_{(A, B) \in \mathcal{M} \times \mathcal{N}} f(A, B).$$

Since f is bounded from below and coercive with respect to A and $A + B$ (i.e. $f(A, B) \rightarrow \infty$ if either $\|A\| \rightarrow \infty$ or $\|A + B\| \rightarrow \infty$), the sequences $\{A^k\}$ and $\{A^k + B^k\}$ are both uniformly bounded and, therefore, $\{B^k\}$ is also uniformly bounded. By compactness, $\{(A^k, B^k)\}$ admits an accumulation point (A^*, B^*) . Moreover, note that the feasible set $\mathcal{M} \times \mathcal{N}$ is closed and $f : \mathcal{M} \times \mathcal{N} \rightarrow \mathbb{R}$ is continuous. Thus, we conclude that (A^*, B^*) is a global minimizer. \square

Any global minimizer $(A^*, B^*) \in \mathcal{M} \times \mathcal{N}$ of (1) satisfies the first-order necessary optimality condition:

$$\begin{cases} \langle \Delta, (1 + \mu)A^* + B^* - Z \rangle \geq 0, & \text{for any } \Delta \in T_{\mathcal{M}}(A^*), \\ \langle \Delta, A^* + B^* - Z \rangle \geq 0, & \text{for any } \Delta \in T_{\mathcal{N}}(B^*). \end{cases} \tag{2}$$

Here, $T_{\mathcal{M}}(A^*)$ denotes the tangent cone of the set \mathcal{M} at A^* , and analogously for $T_{\mathcal{N}}(B^*)$. Note that the structure of the optimality condition (2) is due to the separability of the constraints.

Whenever $\text{rank}(A^*) = r$, the set \mathcal{M} is locally (around A^*) a Riemannian manifold with the Riemannian metric $\langle \cdot, \cdot \rangle$. Hence, $T_{\mathcal{M}}(A^*)$ reduces to a linear subspace in $\mathbb{R}^{m \times n}$, namely the tangent space of \mathcal{M} at A^* , and the first variational inequality in (2) becomes

$$P_{T_{\mathcal{M}}(A^*)}((1 + \mu)A^* + B^* - Z) = 0. \tag{3}$$

Here $P_{T_{\mathcal{M}}(A^*)}$ denotes the orthogonal projection onto the linear subspace $T_{\mathcal{M}}(A^*)$. Let $U\Sigma V^\top$ be the compact singular value decomposition (SVD) of the matrix A^* . Then the tangent space $T_{\mathcal{M}}(A^*)$ is given by

$$T_{\mathcal{M}}(A^*) = \{UMV^\top + U_p V^\top + UV_p^\top : M \in \mathbb{R}^{r \times r}, U_p \in \mathbb{R}^{m \times r}, U_p^\top U = 0, V_p \in \mathbb{R}^{n \times r}, V_p^\top V = 0\};$$

see, e.g., [33]. Analogously, whenever $\|B^*\|_0 = s$, \mathcal{N} is a Riemannian manifold around B^* with the Riemannian metric $\langle \cdot, \cdot \rangle$. Hence, $T_{\mathcal{N}}(B^*)$ reduces to the tangent space of \mathcal{N} at B^* , and correspondingly the second variational inequality in (2) becomes

$$P_{T_{\mathcal{N}}(B^*)}(A^* + B^* - Z) = 0, \quad (4)$$

where the tangent space $T_{\mathcal{N}}(B^*)$ is given by

$$T_{\mathcal{N}}(B^*) = \{\Delta \in \mathbb{R}^{m \times n} : \text{supp}(\Delta) \subset \text{supp}(B^*)\}.$$

3 Alternating minimization on matrix manifolds

In this section, we investigate the numerical solution of the variational problem (1). While (1) is handled by a rather straightforward alternating minimization scheme, the respective subproblems are sophisticated due to the respective constraint sets.

3.1 Alternating minimization scheme and its convergence property

We first formulate our alternating minimization scheme in Algorithm 3.1 below. Then a rather macroscopic convergence result for this algorithm is given in Theorem 3.2. While the proof for Theorem 3.2 is straightforward, the major work is to figure out appropriate algorithmic steps for solving the respective subproblems that activate the convergence criteria, which are the subjects of sections 3.2 and 3.3.

Algorithm 3.1 (Alternating minimization scheme).

Initialize $A^0 \in \mathcal{M}$, $B^0 \in \mathcal{N}$. Set $k := 0$ and iterate:

1. Compute $A^{k+1} \in \mathcal{M}$ as an approximate solution for the A -subproblem: $\min_{A \in \mathcal{M}} \frac{1}{2} \|A + B^k - Z\|^2 + \frac{\mu}{2} \|A\|^2$.
2. Compute $B^{k+1} \in \mathcal{N}$ as an approximate solution for the B -subproblem: $\min_{B \in \mathcal{N}} \frac{1}{2} \|A^{k+1} + B - Z\|^2$.
3. If a suitable stopping criterion is satisfied, then stop; otherwise set $k := k + 1$ and return to step 1.

Theorem 3.2. *Let $\{(A^k, B^k)\} \subset \mathcal{M} \times \mathcal{N}$ be the sequence generated by Algorithm 3.1. Suppose that there exists a positive constant δ and two sequences of nonnegative scalars $\{\varepsilon_a^k\}$ and $\{\varepsilon_b^k\}$ such that the following conditions are satisfied for all k :*

$$f(A^{k+1}, B^k) \leq f(A^k, B^k) - \delta \|A^{k+1} - A^k\|^2, \quad (5)$$

$$f(A^{k+1}, B^{k+1}) \leq f(A^{k+1}, B^k) - \delta \|B^{k+1} - B^k\|^2, \quad (6)$$

$$\langle \Delta, (1 + \mu)A^{k+1} + B^k - Z \rangle \geq -\varepsilon_a^k \|\Delta\|, \quad \text{for any } \Delta \in T_{\mathcal{M}}(A^{k+1}), \quad (7)$$

$$\langle \Delta, A^{k+1} + B^{k+1} - Z \rangle \geq -\varepsilon_b^k \|\Delta\|, \quad \text{for any } \Delta \in T_{\mathcal{N}}(B^{k+1}). \quad (8)$$

Furthermore, let $\{(A^{k^l}, B^{k^l})\}$ be any convergent subsequence of $\{(A^k, B^k)\}$ with the limit point $(A^*, B^*) \in \mathcal{M} \times \mathcal{N}$ such that $\text{rank}(A^*) = r$, $\|B^*\|_0 = s$, and $\lim_{l \rightarrow \infty} \varepsilon_a^{k^l} = \lim_{l \rightarrow \infty} \varepsilon_b^{k^l} = 0$. Then (A^*, B^*) satisfies the first-order optimality conditions (3)–(4).

Proof. First note that $f(A^{k+1}, B^{k+1}) \leq f(A^{k+1}, B^k) \leq f(A^k, B^k)$ for all k . Since f is bounded from below, we have $\lim_{k \rightarrow \infty} f(A^{k+1}, B^k) - f(A^k, B^k) = \lim_{k \rightarrow \infty} f(A^{k+1}, B^k) - f(A^{k+1}, B^{k+1}) = 0$, which by conditions (5)–(6) implies that $\lim_{k \rightarrow \infty} \|A^k - A^{k+1}\| = \lim_{k \rightarrow \infty} \|B^k - B^{k+1}\| = 0$.

Now let $\{(A^{k^l}, B^{k^l})\}$ be a subsequence that converges to some (A^*, B^*) with $\text{rank}(A^*) = r$ and $\|B^*\|_0 = s$. Then we have $\text{rank}(A^{k^l}) = r$ and $\|B^{k^l}\|_0 = s$ for all sufficiently large l . Since \mathcal{M} is a smooth manifold in a neighborhood of A^* and \mathcal{N} is a smooth manifold in a neighborhood B^* , conditions (7)–(8) yield that $\|P_{T_{\mathcal{M}}(A^{k^l+1})}((1+\mu)A^{k^l+1} + B^{k^l} - Z)\| \leq \varepsilon_a^{k^l}$ and $\|P_{T_{\mathcal{N}}(B^{k^l+1})}(A^{k^l+1} + B^{k^l+1} - Z)\| \leq \varepsilon_b^{k^l}$ for all sufficiently large l . Due to the continuity of the mappings $(A, M) \in \mathcal{M} \times \mathbb{R}^{m \times n} \mapsto P_{T_{\mathcal{M}}(A)}(M)$ and $(B, M) \in \mathcal{N} \times \mathbb{R}^{m \times n} \mapsto P_{T_{\mathcal{N}}(B)}(M)$, we conclude that the optimality conditions (3)–(4) hold true by passing $l \rightarrow \infty$. \square

In the following, we discuss in detail the resolution of the subproblems in Algorithm 3.1 such that the conditions (5)–(8) in Theorem 3.2 are fulfilled. We start by studying step 2 of Algorithm 3.1.

3.2 Sparse matrix (B -)subproblem

The global minimizer of the sparse matrix subproblem $\min_{B \in \mathcal{N}} \frac{1}{2} \|A^{k+1} + B - Z\|^2$ can be obtained explicitly in closed form by utilizing the projection operator $P_{\mathcal{N}}$. For this purpose, for a given matrix $M \in \mathbb{R}^{m \times n}$, one aligns its entries in decreasing order with respect to the absolute value; i.e. $|M_{i_1 j_1}| \geq |M_{i_2 j_2}| \geq \dots \geq |M_{i_{mn} j_{mn}}|$. Then one obtains $P_{\mathcal{N}}(M)$ by setting

$$(P_{\mathcal{N}}(M))_{i^l j^l} = \begin{cases} M_{i^l j^l}, & \text{if } l \leq s, \\ 0, & \text{otherwise.} \end{cases}$$

Note that $P_{\mathcal{N}}(M)$ is not unique if $M_{i^s j^s} = M_{i^{s+1} j^{s+1}}$. In this case we simply take $P_{\mathcal{N}}(M)$ to be any one of the valid candidates. With $P_{\mathcal{N}}$ at hand, the global minimizer of the sparse matrix subproblem is computed as in step 1 of Algorithm 3.3 below. On the other hand, such a global minimizer does not necessarily guarantee a sufficient decrease in the objective as required by condition (6). When the global minimizer fails to fulfill condition (6), we resort to a local minimizer as specified by step 3 in Algorithm 3.3.

Algorithm 3.3 (B -subproblem solver).

Let $(A^{k+1}, B^k) \in \mathcal{M} \times \mathcal{N}$ be given. Choose $0 < \delta \leq 1/2$.

1. Compute the global minimizer of the B -subproblem $\hat{B}^{k+1} = P_{\mathcal{N}}(Z - A^{k+1})$.
2. If $f(A^{k+1}, \hat{B}^{k+1}) \leq f(A^{k+1}, B^k) - \delta \|\hat{B}^{k+1} - B^k\|^2$, then accept $B^{k+1} = \hat{B}^{k+1}$; otherwise reject \hat{B}^{k+1} and continue with step 3.
3. Return B^{k+1} with

$$B_{ij}^{k+1} = \begin{cases} (Z - A^{k+1})_{ij}, & \text{if } B_{ij}^k \neq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

Theorem 3.4. *The solution B^{k+1} computed by Algorithm 3.3 satisfies condition (6). Moreover, if $\|B^{k+1}\|_0 = s$, condition (8) holds with $\varepsilon_b^k = 0$.*

Proof. (Case 1). We first prove the conclusion in the case that \hat{B}^{k+1} is accepted. It follows immediately from step 2 of Algorithm 3.3 that condition (6) holds.

Now assume that $\|B^{k+1}\|_0 = s$, which implies that the tangent space $T_{\mathcal{N}}(B^{k+1}) = \{\Delta \in \mathbb{R}^{m \times n} : \text{supp}(\Delta) \subset \text{supp}(B^{k+1})\}$. Then it follows that $\Delta_{ij} = 0$ whenever $(i, j) \notin \text{supp}(B^{k+1})$ and

that $B_{ij}^{k+1} = (Z - A^{k+1})_{ij}$ whenever $(i, j) \in \text{supp}(B^{k+1})$. Therefore, $\langle \Delta, A^{k+1} + B^{k+1} - Z \rangle = 0$ for any $\Delta \in T_{\mathcal{N}}(B^{k+1})$ and (8) holds with $\varepsilon_b^k = 0$.

(Case 2). Now consider the case where \hat{B}^{k+1} is not accepted. Then (9) must hold true; i.e. we have that $B_{ij}^{k+1} = (Z - A^{k+1})_{ij}$ whenever $(i, j) \in \text{supp}(B^k)$ and that $(B^k - B^{k+1})_{ij} = 0$ whenever $(i, j) \notin \text{supp}(B^k)$. Thus condition (6) is fulfilled since

$$\begin{aligned} f(A^{k+1}, B^k) - f(A^{k+1}, B^{k+1}) &= \frac{1}{2} \|B^k - B^{k+1}\|^2 + \langle B^k - B^{k+1}, A^{k+1} + B^{k+1} - Z \rangle \\ &= \frac{1}{2} \|B^k - B^{k+1}\|^2 \geq \delta \|B^k - B^{k+1}\|^2. \end{aligned}$$

The argument for the satisfaction of condition (8) with $\varepsilon_b^k = 0$ is analogous to the one given in Case 1. \square

In the numerical implementation of step 1 of Algorithm 3.3, we call the Matlab command `sort`, which is based on a *Quicksort* algorithm of complexity $O(mn \log(mn))$ in average. We remark that, according to our numerical experience, the overall cost of the alternating minimization scheme is dominated by the A -subproblem solve. Therefore, in this paper we do not pursue more advanced randomized partial ordering algorithms [20], e.g. *Quickselect*, for further CPU gain. We also remark that the choice of the parameter δ in Algorithm 3.3 represents a tradeoff between the convergence of the iterates and the global optimality of their limit. In fact, if δ is too large, then the iterates may possibly converge to one among many undesired local solutions. On the other hand, it may slow down the speed of convergence by choosing δ too close to 0.

3.3 Low-rank matrix (A -)subproblem

Now we turn our attention to solving the A -subproblem, namely the task of step 1 of Algorithm 3.1. Unlike solving the B -subproblem in section 3.2, the A -subproblem will be resolved inexactly by a single update of a gradient-based algorithm. The organization of this subsection is as follows. In section 3.3.1, we review the global minimizer via SVD and discuss its drawbacks in numerical computation. In section 3.3.2, we develop a projected dogleg method on the fixed-rank matrix manifold for resolving the A -subproblem and its convergence property is studied in detail.

3.3.1 Global minimizer via (partial) SVD

The low-rank matrix subproblem $\min_{A \in \mathcal{M}} \frac{1}{2} \|A + B^k - Z\|^2 + \frac{\mu}{2} \|A\|^2$ admits a global minimizer in closed form for any $\mu \geq 0$. It is obtained by the projection of $\frac{1}{1+\mu}(Z - B^k)$ onto \mathcal{M} , denoted by $P_{\mathcal{M}}(\frac{1}{1+\mu}(Z - B^k))$. Let $U_z^k \Sigma_z^k (V_z^k)^\top$ be the singular-value decomposition (SVD) of the matrix $\frac{1}{1+\mu}(Z - B^k)$, where $U_z^k \in \mathbb{R}^{m \times m}$ and $V_z^k \in \mathbb{R}^{n \times n}$ are both orthogonal matrices, and Σ_z^k is a diagonal matrix in $\mathbb{R}^{m \times n}$ with nonnegative diagonal elements $(\sigma_z^k)_1, (\sigma_z^k)_2, \dots, (\sigma_z^k)_m$ in decreasing order. Then, by the well-known Eckart-Young theorem [13], a global minimizer of the A -subproblem is given by $P_{\mathcal{M}}(Z - B^k) = U_z^k \hat{\Sigma}_z^k (V_z^k)^\top$, where $\hat{\Sigma}_z^k$ is a diagonal matrix in $\mathbb{R}^{m \times n}$ with diagonal elements $(\sigma_z^k)_1, (\sigma_z^k)_2, \dots, (\sigma_z^k)_r, 0, \dots, 0$.

The classical SVD of an m -by- n matrix has a complexity of $O(mn^2)$ flops [32], which is rather expensive in large-scale computation. In the context of our low-rank subproblem, however, this can be accelerated by a partial SVD technique of complexity $O(mnr)$, see e.g. the package PROPACK [3] available online, since only the first r -th singular values and vectors are needed. Although such a global minimization strategy often works quite efficiently, see the corresponding numerical tests in section 4, it does not guarantee for the overall alternating minimization scheme

(global) convergence towards a stationary point from an arbitrary initial guess. In particular, satisfaction of the sufficient conditions for global convergence, i.e. conditions (5) and (7), is not ensured. For this sake, in the following we investigate in detail an inexact-solution strategy for the low-rank subproblem based on Riemannian optimization, which fulfills conditions (5) and (7), thus admitting a global convergence theory for Algorithm 3.1. The proposed method also enjoys good practical efficiency as will be demonstrated in section 4.

3.3.2 Projected dogleg method on a fixed-rank matrix manifold

Riemannian optimization techniques have been developed in the past two decades; see, e.g., [30, 14]. More recently, these methods have been successfully applied to optimization problems related to low-rank matrices [29, 7, 33]. In the following, we develop a tailored Riemannian optimization approach, namely a projected dogleg method, on a rank- r matrix manifold $\bar{\mathcal{M}}(r) = \{A \in \mathbb{R}^{m \times n} : \text{rank}(A) = r\}$. We emphasize that the ultimate goal of the projected dogleg method under consideration is to fulfill conditions (5) and (7) with $\lim_{k \rightarrow \infty} \varepsilon_a^k = 0$ in order to guarantee the global convergence of the alternating minimization scheme. Other Riemannian approaches, e.g. Riemannian trust-region method [5, 8], may also be applicable in the context, but require a rather different, perhaps more involved, analysis.

Given $B^k \in \mathcal{N}$, define the smooth mapping $f_A^k : \bar{\mathcal{M}} \rightarrow \mathbb{R}$ with $f_A^k(A) = f(A, B^k)$ for all $A \in \bar{\mathcal{M}}$. The *Riemannian gradient* of f_A^k at A on $\bar{\mathcal{M}}$, denoted by $\text{grad} f_A^k(A)$, is defined as a tangent vector in the tangent space $T_{\bar{\mathcal{M}}}(A)$ such that $\langle \text{grad} f_A^k(A), \Delta \rangle = Df_A^k(A)[\Delta]$ for all $\Delta \in T_{\bar{\mathcal{M}}}(A)$. Here $Df_A^k(A)[\Delta]$ is the directional derivative of f_A^k at A along the direction Δ . Let ∇ be the (unique) *Riemannian connection* on $\bar{\mathcal{M}}$, and let $\nabla_{\eta(A)} \xi(A) \in T_{\bar{\mathcal{M}}}(A)$ denote the *covariant derivative* of two smooth vector fields ξ and η on $\bar{\mathcal{M}}$ at A . Then the *Riemannian Hessian* of f_A^k at A on $\bar{\mathcal{M}}$, denoted by $\text{Hess} f_A^k(A)$, is a linear mapping from $T_{\bar{\mathcal{M}}}(A)$ to $T_{\bar{\mathcal{M}}}(A)$ such that $\text{Hess} f_A^k(A)[\Delta] = \nabla_{\Delta} \text{grad} f_A^k(A)$ for any $\Delta \in T_{\bar{\mathcal{M}}}(A)$. By considering $\bar{\mathcal{M}}$ as an embedded submanifold in the Euclidean space $(\mathbb{R}^{m \times n}, \langle \cdot, \cdot \rangle)$, the Riemannian gradient is derived as

$$\text{grad} f_A^k(A) = P_{T_{\bar{\mathcal{M}}}(A)}(\nabla f_A^k(A)) = P_{T_{\bar{\mathcal{M}}}(A)}((1 + \mu)A + B^k - Z),$$

see section 3.6.1 in [5]. The derivation of the Riemannian Hessian is more involved in general. For the rank- r matrix manifold, the following Hessian formula can be calculated by constructing a factorization-based second-order retraction [33]:

$$\begin{aligned} \text{Hess} f_A^k(A)[\Delta] &= (1 + \mu)\Delta + (I - UU^{\top})\nabla f_A^k(A)(I - VV^{\top})\Delta^{\top}U\Sigma^{-1}V^{\top} \\ &\quad + U\Sigma^{-1}V^{\top}\Delta^{\top}(I - UU^{\top})\nabla f_A^k(A)(I - VV^{\top}) \\ &= (1 + \mu)\Delta + (I - UU^{\top})(B^k - Z)(I - VV^{\top})\Delta^{\top}U\Sigma^{-1}V^{\top} \\ &\quad + U\Sigma^{-1}V^{\top}\Delta^{\top}(I - UU^{\top})(B^k - Z)(I - VV^{\top}), \end{aligned} \quad (10)$$

where $A = U\Sigma V^{\top}$ is the compact SVD of A with a full-rank diagonal matrix $\Sigma \in \mathbb{R}^{r \times r}$ and two orthonormal matrices $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$. It is worth noting that the Hessian formula (10) should be handled in a matrix-free fashion so that computing each matrix-vector product $\text{Hess} f_A^k(A)[\Delta]$ requires $O(mnr)$ flops. To ease our presentation, in the remainder of section 3.3.2, we use the notations $g^k := \text{grad} f_A^k(A^k)$, $H^k := \text{Hess} f_A^k(A^k)$, and assume that $g^k \neq 0$.

One can approximate f_A^k around A^k in the tangent space $T_{\bar{\mathcal{M}}}(A^k)$ by a quadratic function $h^k(\Delta^k) := f_A^k(A^k) + \langle g^k, \Delta^k \rangle + \frac{1}{2}\langle \Delta^k, H^k[\Delta^k] \rangle$ for $\Delta^k \in T_{\bar{\mathcal{M}}}(A^k)$. Presuming that H^k is positive definite on $T_{\bar{\mathcal{M}}}(A^k)$, based on the *Cauchy point*

$$\Delta_C^k := -\frac{\|g^k\|^2}{\langle g^k, H^k[g^k] \rangle} g^k, \quad (11)$$

and the *Newton point*

$$\Delta_N^k := -(H^k)^{-1}[g^k], \quad (12)$$

we define the dogleg path in the tangent space $T_{\bar{\mathcal{M}}}(A^k)$ as follows:

$$\Delta^k(\tau^k) = \begin{cases} \tau^k \Delta_C^k, & \text{if } 0 \leq \tau^k \leq 1, \\ \Delta_C^k + (\tau^k - 1)(\Delta_N^k - \Delta_C^k), & \text{if } 1 \leq \tau^k \leq 2. \end{cases} \quad (13)$$

Lemma 3.5. *For the statements:*

- i.* H^k is positive definite on $T_{\bar{\mathcal{M}}}(A^k)$; i.e. $\langle \Delta, H^k[\Delta] \rangle > 0$ for any nonzero $\Delta \in T_{\bar{\mathcal{M}}}(A^k)$;
- ii.* $\langle \Delta_C^k, \Delta_N^k - \Delta_C^k \rangle \geq 0$;
- iii.* $\|\Delta^k(\tau^k)\|$ is an increasing function in $\tau^k \in [0, 2]$;

the following implication holds true: (i) \Rightarrow (ii) \Rightarrow (iii).

Proof. The proof is analogous to the one of Lemma 4.2 in [27]. \square

Given the current iterate $A^k \in \bar{\mathcal{M}}$, in order to generate the next iterate from the update step in the tangent space at A^k , we use the metric projection $P_{\bar{\mathcal{M}}} : \mathbb{R}^{m \times n} \rightarrow \bar{\mathcal{M}}$ defined by $P_{\bar{\mathcal{M}}}(Z) = \arg \min_{A \in \bar{\mathcal{M}}} \|A - Z\|$, which makes a smooth mapping locally around A^k ; see, e.g., [21]. Different from the scenario in section 3.3.1, given $A \in \bar{\mathcal{M}}$ and $\Delta \in T_{\bar{\mathcal{M}}}(A)$, the projection $P_{\bar{\mathcal{M}}}(A + \Delta)$ can be computed via a *reduced SVD* on a $2r$ -by- $2r$ matrix thanks to unitary invariance; see, e.g., [26, 33]. The reduction of the computational cost, compared to the approach in section 3.3.1, is significant in practice where r is typically much smaller than m and n . For the reader's convenience, we describe the implementation of the projection operation in Algorithm 3.6 below.

Algorithm 3.6 (Projection onto fixed-rank matrix manifold via reduced SVD).

Let $A \in \bar{\mathcal{M}}(r)$, represented in the compact SVD form $A = U\Sigma V^\top$, and $\Delta \in T_{\bar{\mathcal{M}}(r)}(A)$ be given. Choose $0 < \epsilon_s \ll 1$.

1. Compute $M = U^\top \Delta V$, $U_p = \Delta V - UM$, $V_p = \Delta^\top U - VM^\top$.
2. Perform the QR-factorization of U_p and V_p such that $U_p = Q_u R_u$ and $V_p = Q_v R_v$ with two orthonormal matrices $Q_u \in \mathbb{R}^{m \times r}$, $Q_v \in \mathbb{R}^{n \times r}$ and two upper-triangular matrices $R_u, R_v \in \mathbb{R}^{r \times r}$.
3. Perform an SVD of the $2r$ -by- $2r$ matrix on the left-hand side of the following equation:

$$\begin{bmatrix} \Sigma + M & R_v^\top \\ R_u & 0 \end{bmatrix} = \tilde{U} \tilde{\Sigma} \tilde{V}^\top,$$

where $\tilde{\Sigma} = \text{diag}(\{\tilde{\sigma}_j\}_{j=1}^{2r}) \in \mathbb{R}^{2r \times 2r}$ is some diagonal matrix with positive diagonal entries $\{\tilde{\sigma}_j\}_{j=1}^{2r}$ in descending order, and $\tilde{U}, \tilde{V} \in \mathbb{R}^{2r \times 2r}$ are two orthogonal matrices.

4. Set $\hat{\Sigma} = \text{diag}(\{\max(\tilde{\sigma}_j, \epsilon_s)\}_{j=1}^r) \in \mathbb{R}^{r \times r}$, $\hat{U} = [U \ Q_u][\{\tilde{U}_j\}_{j=1}^r] \in \mathbb{R}^{m \times r}$, and $\hat{V} = [V \ Q_v][\{\tilde{V}_j\}_{j=1}^r] \in \mathbb{R}^{n \times r}$, where \tilde{U}_j and \tilde{V}_j denote the j -th columns of \tilde{U} and \tilde{V} , respectively. Return $P_{\bar{\mathcal{M}}(r)}(A + \Delta) = \hat{U} \hat{\Sigma} \hat{V}^\top$.

Concerning the QR-factorization of the matrices U_p and V_p required in step 2 of Algorithm 3.6, we remark that in a MATLAB environment one may call the command `qr` with the “economy-size” option. In addition, we note that a small positive parameter ϵ_s is introduced in step 4 in order to prevent rank deficiency of the projection. Ideally, it suffices to choose $\epsilon_s > 0$ which is significantly smaller than the minimal nonzero singular value of the underlying low-rank matrix A^* that, together with B^* , solves (1). Throughout our numerical experiments in section 4, we shall fix $\epsilon_s = 10^{-3}$. For a proper tuning of the underlying rank r along the overall iterative algorithm, we refer to the trimming procedure presented in section 3.4.

It is known [21, 6] that for any point A on the smooth manifold $\bar{\mathcal{M}}$, the projection $P_{\bar{\mathcal{M}}}$ is a smooth diffeomorphism in a neighborhood of A , and moreover the differentiation rule

$$DP_{\bar{\mathcal{M}}}(A)[\Delta] = P_{T_{\bar{\mathcal{M}}}(A)}(\Delta)$$

holds for any $\Delta \in \mathbb{R}^{m \times n}$. Thus, the projected dogleg path $\tau^k \mapsto P_{\bar{\mathcal{M}}}(A^k + \Delta^k(\tau^k))$ is a well-defined smooth function in a neighborhood of 0. We remark that, in the context of [5, 6], $P_{\bar{\mathcal{M}}}$ induces a *second-order retraction* on $\bar{\mathcal{M}}$ near A given by $\Delta \in T_{\bar{\mathcal{M}}}(A) \mapsto P_{\bar{\mathcal{M}}}(A + \Delta) \in \bar{\mathcal{M}}$, which locally fits the exponential mapping up to second order.

We are now in a position to present the projected dogleg method for solving the low-rank matrix subproblem. Below, we have chosen a specific sequence of trial step sizes \mathcal{F}^k , but obviously other choices may be considered as well.

Algorithm 3.7 (A -subproblem solver via projected dogleg method).

Let $(A^k, B^k) \in \bar{\mathcal{M}} \times \mathcal{N}$ be given. Choose $\delta > 0$.

- 0.[#] (Optional) Compute the global minimizer of the A -subproblem $\hat{A}^{k+1} = P_{\mathcal{M}}(\frac{1}{1+\mu}(Z - B^k))$. If $f_A^k(\hat{A}^{k+1}) \leq f_A^k(A^k) - \delta \|\hat{A}^{k+1} - A^k\|^2$, then accept $A^{k+1} = \hat{A}^{k+1}$; otherwise, reject \hat{A}^{k+1} and continue with step 1.
1. Compute g^k, H^k . If $\langle g^k, H^k[g^k] \rangle > 0$, then compute Δ_C^k by formula (11); otherwise, set $\Delta^k(\tau^k) := -\tau^k g^k$, $\mathcal{F}^k := \{1, 1/2, 1/4, 1/8, 1/16, \dots\}$, and go to step 3.
2. Compute Δ_N^k by formula (12). If $\langle \Delta_C^k, \Delta_N^k - \Delta_C^k \rangle < 0$ or any non-positive definiteness of H^k is detected during the computation, then set $\Delta^k(\tau^k) := \tau^k \Delta_C^k$ and $\mathcal{F}^k := \{1, 1/2, 1/4, 1/8, 1/16, \dots\}$; otherwise define the dogleg path $\Delta^k : [0, 2] \rightarrow T_{\bar{\mathcal{M}}}(A^k)$ as in (13) and set $\mathcal{F}^k := \{2, 3/2, 1, 1/2, 1/4, 1/8, 1/16, \dots\}$.
3. Set τ^k to be the largest element in \mathcal{F}^k that fulfills

$$f_A^k(A^k) - f_A^k(P_{\bar{\mathcal{M}}}(A^k + \Delta^k(\tau^k))) \geq \delta \|A^k - P_{\bar{\mathcal{M}}}(A^k + \Delta^k(\tau^k))\|^2. \quad (14)$$

Return $A^{k+1} = P_{\bar{\mathcal{M}}}(A^k + \Delta^k(\tau^k))$.

We remark that step 0 is included in the above algorithm as an optional trial step, only recommended for utility when the global minimizer \hat{A}^{k+1} tends to be accepted and can be computed at low cost, say e.g. via partial SVD [3]. Since the projected dogleg method works practically well in its own right, unless otherwise specified, this trial step is skipped in our subsequent algorithmic development and analysis. Nevertheless, in section 4.2 we shall numerically compare the performances of Algorithm 3.7 both with and without step 0, together with the augmented Lagrangian method based on a convex variational model.

[#] This trial step is optional, which is only recommended for utility if the global minimizer tends to be accepted and can be computed at low cost. Unless otherwise specified, this step is skipped in our algorithmic development and analysis.

Lemma 3.8. *There exists $\bar{\tau}^k > 0$ such that condition (14) is fulfilled for all $\tau^k \in (0, \bar{\tau}^k]$. Consequently, step 3 in Algorithm 3.7 always returns some admissible step size $\tau^k > 0$ fulfilling condition (5) after finitely many trials.*

Proof. Let $\phi(\tau^k) := f_A^k(A^k) - f_A^k(P_{\bar{\mathcal{M}}}(A^k + \Delta^k(\tau^k))) - \delta \|A^k - P_{\bar{\mathcal{M}}}(A^k + \Delta^k(\tau^k))\|^2$, which is a well-defined smooth function in a neighborhood of 0. Then it follows that $\phi(0) = 0$ and

$$\phi'(0) = -\langle g^k, (\Delta^k)'(0) \rangle \geq \min\left(1, \frac{\|g^k\|^2}{\langle g^k, H^k[g^k] \rangle}\right) \|g^k\|^2 \geq \min\left(1, \frac{1}{\lambda_{\max}(H^k)}\right) \|g^k\|^2 > 0.$$

Since ϕ is continuously differentiable in a neighborhood of 0, there exists some $\bar{\tau}^k > 0$ such that $\phi'(\cdot) > 0$ on the interval $(0, \bar{\tau}^k]$. By utilizing the mean value theorem, we conclude that $\phi(\cdot) \geq 0$ on the interval $(0, \bar{\tau}^k]$. \square

Lemma 3.9. *Let $\{A^k\} \subset \bar{\mathcal{M}}$ be generated by Algorithm 3.7 along with some sequence $\{B^k\} \subset \mathcal{N}$ satisfying condition (6). Then the following statements hold true:*

- i. $\lim_{k \rightarrow \infty} \|A^k - A^{k+1}\| = 0$.
- ii. $\lim_{k \rightarrow \infty} \|\Delta^k(\tau^k)\| = 0$.
- iii. Any convergent subsequence $\{A^{k^l}\}$ of $\{A^k\}$ satisfies $\lim_{l \rightarrow \infty} \|g^{k^l}\| = 0$.

Proof. Owing to Lemma 3.8, the proof of (i) essentially resembles the first part of the proof for Theorem 3.2.

Concerning (ii), note that $A^{k+1} = P_{\bar{\mathcal{M}}}(A^k + \Delta^k(\tau^k))$, which satisfies the necessary condition $P_{T_{\bar{\mathcal{M}}}(A^{k+1})}(A^{k+1} - A^k - \Delta^k(\tau^k)) = 0$. Then it follows from the reverse triangle inequality that

$$\|P_{T_{\bar{\mathcal{M}}}(A^{k+1})}(\Delta^k(\tau^k))\| \leq \|P_{T_{\bar{\mathcal{M}}}(A^{k+1})}(A^{k+1} - A^k - \Delta^k(\tau^k))\| + \|A^k - A^{k+1}\| \rightarrow 0,$$

and therefore

$$\begin{aligned} \|\Delta^k(\tau^k)\| &= \|P_{T_{\bar{\mathcal{M}}}(A^k)}(\Delta^k(\tau^k))\| \\ &\leq \|P_{T_{\bar{\mathcal{M}}}(A^{k+1})}(\Delta^k(\tau^k))\| + \|P_{T_{\bar{\mathcal{M}}}(A^k)}(\Delta^k(\tau^k)) - P_{T_{\bar{\mathcal{M}}}(A^{k+1})}(\Delta^k(\tau^k))\| \rightarrow 0, \end{aligned}$$

as $k \rightarrow \infty$.

We prove (iii) by contradiction. For this purpose, let $\{A^{k^l}\}$ be a convergent subsequence of $\{A^k\}$ and $\varepsilon > 0$ such that $\|g^{k^l}\| \geq \varepsilon$ for all l . Based on an observation of the structure of the Riemannian Hessian given in (10), the sequence $\{H^{k^l}\}$ is uniformly bounded, and we denote $\kappa_h := \sup_l \lambda_{\max}(H^{k^l})$. Making use of Lemma 3.5(iii), we obtain a lower bound for $\|\Delta^{k^l}(\tau^{k^l})\|$ as follows:

$$\|\Delta^{k^l}(\tau^{k^l})\| \geq \min\left(\tau^{k^l} \|g^{k^l}\|, \min(1, \tau^{k^l}) \frac{\|g^{k^l}\|^3}{\langle g^{k^l}, H^{k^l}[g^{k^l}] \rangle}\right) \geq \varepsilon \min(1, 1/\kappa_h) \min(1, \tau^{k^l}).$$

Then the result in (ii) yields that $\lim_{l \rightarrow \infty} \tau^{k^l} = 0$. Due to the nature of the backtracking dogleg search in step 3 of Algorithm 3.7, this further implies that the trial step $2\tau^{k^l}$ is not admissible at iteration l for all sufficiently large l ; i.e.

$$f_A^{k^l}(A^{k^l}) - f_A^{k^l}(P_{\bar{\mathcal{M}}}(A^{k^l} + \Delta^{k^l}(2\tau^{k^l}))) < \delta \|A^{k^l} - P_{\bar{\mathcal{M}}}(A^{k^l} + \Delta^{k^l}(2\tau^{k^l}))\|^2,$$

for all l . Dividing both sides above by $2\tau^{kl}$ and passing $l \rightarrow \infty$, we find

$$0 \geq \lim_{l \rightarrow \infty} -\langle g^{kl}, (\Delta^{kl})'(0) \rangle \geq \min \left(1, \frac{1}{\kappa_h} \right) \lim_{l \rightarrow \infty} \|g^{kl}\|^2,$$

which leads to a contradiction. \square

Lemma 3.10. *Let $(A^*, B^*) \in \bar{\mathcal{M}} \times \mathcal{N}$ satisfy the first-order optimality conditions (3)–(4) with $\|B^*\|_0 = s$. Further assume that the Riemannian Hessian $\text{Hess}f(A^*, B^*) : T_{\bar{\mathcal{M}}}(A^*) \times T_{\mathcal{N}}(B^*) \rightarrow T_{\bar{\mathcal{M}}}(A^*) \times T_{\mathcal{N}}(B^*)$ is strictly positive definite when $\mu = 0$. Then the following statements hold true:*

- i. The Riemannian Hessian of f with respect to the first argument, denoted by $\text{Hess}_A f(A^*, B^*) : T_{\bar{\mathcal{M}}}(A^*) \rightarrow T_{\bar{\mathcal{M}}}(A^*)$, is strictly positive definite for any $\mu \geq 0$.*
- ii. The following tangent space transversality holds true:*

$$T_{\bar{\mathcal{M}}}(A^*) \cap T_{\mathcal{N}}(B^*) = \{0\}. \quad (15)$$

- iii. The linear operator $P_{T_{\bar{\mathcal{M}}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$ is a contraction; i.e. there exists a constant $\kappa_p \in [0, 1)$ such that*

$$\|(P_{T_{\bar{\mathcal{M}}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)})(\Delta)\| \leq \kappa_p \|\Delta\|, \quad (16)$$

for all $\Delta \in \mathbb{R}^{m \times n}$.

Proof. Given an arbitrary nonzero element Δ_A in $T_{\bar{\mathcal{M}}}(A^*)$, we have

$$\begin{aligned} & 0 < \langle (\Delta_A, 0), \text{Hess}f(A^*, B^*)[(\Delta_A, 0)] \rangle_{T_{\bar{\mathcal{M}}}(A^*) \times T_{\mathcal{N}}(B^*)} \Big|_{\mu=0} \\ & = \langle (\Delta_A, 0), \nabla_{(\Delta_A, 0)} \text{grad}f(A^*, B^*) \rangle_{T_{\bar{\mathcal{M}}}(A^*) \times T_{\mathcal{N}}(B^*)} \Big|_{\mu=0} \\ & = \langle \Delta_A, \nabla_{\Delta_A} \text{grad}_A f(A^*, B^*) \rangle_{T_{\bar{\mathcal{M}}}(A^*)} \Big|_{\mu=0} \\ & = \langle \Delta_A, \text{Hess}_A f(A^*, B^*)[\Delta_A] \rangle_{T_{\bar{\mathcal{M}}}(A^*)} \Big|_{\mu=0} \\ & \leq \langle \Delta_A, \text{Hess}_A f(A^*, B^*)[\Delta_A] \rangle_{T_{\bar{\mathcal{M}}}(A^*)} \Big|_{\mu \geq 0}. \end{aligned}$$

The last inequality follows from an observation of the Hessian formula (10). Thus, (i) is proven.

We prove (ii) by contradiction. For this purpose, assume that $\mu = 0$ and there exists a nonzero element $\Delta \in T_{\bar{\mathcal{M}}}(A^*) \cap T_{\mathcal{N}}(B^*)$. Since $\bar{\mathcal{M}}$ is an embedded submanifold of $\mathbb{R}^{m \times n}$, we have

$$\begin{aligned} & \text{Hess}f(A^*, B^*)[(\Delta, -\Delta)] \\ & = P_{T_{\bar{\mathcal{M}}}(A^*) \times T_{\mathcal{N}}(B^*)} (D_{(A,B)} \text{grad}f(A^*, B^*)[(\Delta, -\Delta)]) \\ & = P_{T_{\bar{\mathcal{M}}}(A^*) \times T_{\mathcal{N}}(B^*)} \left(D_{(A,B)} (P_{T_{\bar{\mathcal{M}}}(A)}(A + B - Z), P_{T_{\mathcal{N}}(B)}(A + B - Z)) [(\Delta, -\Delta)] \Big|_{(A,B)=(A^*, B^*)} \right) \\ & = (0, 0). \end{aligned} \quad (17)$$

In the above formulae, note that the first equality follows from Proposition 5.3.2 in [5]. The last equality is a consequence of conditions (3)–(4) and the chain rule of differentiation. Thus,

(17) yields $\langle (\Delta, -\Delta), \text{Hess}f(A^*, B^*)[(\Delta, -\Delta)] \rangle = 0$, which contradicts the positive definiteness of $\text{Hess}f(A^*, B^*)$.

We prove (iii) again by contradiction. Note that the composition of projections $P_{T_{\bar{\mathcal{M}}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)}$ is *nonexpansive*, and therefore condition (16) always hold true for $\kappa_p = 1$. Now assume that (iii) does not hold. Then there must exist a sequence $\{\kappa_p^l\}_{l=1}^\infty \in [0, 1)$ with $\lim_{l \rightarrow \infty} \kappa_p^l = 1$ and correspondingly $\{\Delta^l\} \subset \mathbb{R}^{m \times n}$ with $\|\Delta^l\| = 1$ for all l such that it holds for all l that

$$\kappa_p^l \|\Delta^l\| < \|(P_{T_{\bar{\mathcal{M}}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)})(\Delta^l)\| \leq \|\Delta^l\|.$$

Upon the extraction of a subsequence of $\{\Delta^l\}$, whose limit point $\Delta \in \mathbb{R}^{m \times n}$ satisfies $\|\Delta\| = 1$, we have

$$\|(P_{T_{\bar{\mathcal{M}}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)})(\Delta)\| = \|\Delta\|.$$

Then it follows from the self-adjointness and idempotence of an orthogonal projection onto a linear subspace that

$$\begin{aligned} \langle \Delta, \Delta \rangle &= \langle (P_{T_{\bar{\mathcal{M}}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)})(\Delta), (P_{T_{\bar{\mathcal{M}}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)})(\Delta) \rangle \\ &= \langle \Delta, (P_{T_{\mathcal{N}}(B^*)} \circ P_{T_{\bar{\mathcal{M}}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)})(\Delta) \rangle, \end{aligned}$$

or equivalently $\langle \Delta, (\text{id} - P_{T_{\mathcal{N}}(B^*)} \circ P_{T_{\bar{\mathcal{M}}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)})(\Delta) \rangle = 0$. By the self-adjointness we have $(\text{id} - P_{T_{\mathcal{N}}(B^*)} \circ P_{T_{\bar{\mathcal{M}}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)})(\Delta) = 0$, and thus

$$(P_{T_{\mathcal{N}}(B^*)} \circ P_{T_{\bar{\mathcal{M}}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)})(\Delta) = P_{T_{\mathcal{N}}(B^*)}(\Delta). \quad (18)$$

In particular, note that $P_{T_{\mathcal{N}}(B^*)}(\Delta) \in T_{\mathcal{N}}(B^*)$ and $P_{T_{\mathcal{N}}(B^*)}(\Delta) \neq 0$. Further manipulation of (18) yields that $\langle P_{T_{\mathcal{N}}(B^*)}(\Delta), (\text{id} - P_{T_{\bar{\mathcal{M}}}(A^*)})(P_{T_{\mathcal{N}}(B^*)}(\Delta)) \rangle = 0$, and therefore by the self-adjointness

$$P_{T_{\mathcal{N}}(B^*)}(\Delta) = (P_{T_{\bar{\mathcal{M}}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)})(\Delta).$$

Thus, we have found $P_{T_{\mathcal{N}}(B^*)}(\Delta) \neq 0$ such that $P_{T_{\mathcal{N}}(B^*)}(\Delta) \in T_{\bar{\mathcal{M}}}(A^*) \cap T_{\mathcal{N}}(B^*)$, which contradicts the tangent space transversality in (ii). \square

We remark that in [12] the tangent space transversality condition (15) is discussed in detail, and a sufficient condition on A^* and B^* , which holds with high probability in practice, is also provided for ensuring the tangent space transversality. From Lemma 3.10, we see that tangent space transversality can be naturally regarded as a consequence of the second-order sufficient optimality condition.

Theorem 3.11. *Let $\{A^k\} \subset \bar{\mathcal{M}}$ be a sequence generated by Algorithm 3.7 along with some sequence $\{B^k\} \subset \mathcal{N}$ generated by Algorithm 3.3. At iteration k , assume that the iterate (A^k, B^k) is sufficiently close to some $(A^*, B^*) \in \bar{\mathcal{M}} \times \mathcal{N}$ with $\|B^*\|_0 = s$ satisfying the first-order optimality conditions (3)–(4). Moreover, assume that the Riemannian Hessian $\text{Hess}f(A^*, B^*) \Big|_{\mu=0}$ is strictly positive definite as in Lemma 3.10 and that $0 < \delta < \lambda_{\min}(\text{Hess}_{A^*}f(A^*, B^*))/4$. Then it follows:*

- i. For all sufficiently large k , $\Delta^k(\tau^k) = \Delta_N^k$ is admissible in the backtracking dogleg search in step 3 of Algorithm 3.7; i.e. $A^{k+1} = P_{\bar{\mathcal{M}}}(A^k + \Delta_N^k)$ satisfies condition (5).
- ii. The sequence $\{A^k\}$ converges q -linearly to A^* at rate κ_p ; i.e.

$$\limsup_{k \rightarrow \infty} \frac{\|A^{k+1} - A^*\|}{\|A^k - A^*\|} \leq \kappa_p,$$

where $\kappa_p \in [0, 1)$ is a qualified constant in Lemma 3.10(iii) such that condition (16) holds.

iii. $\lim_{k \rightarrow \infty} \|P_{T_{\bar{\mathcal{M}}}(A^{k+1})}((1 + \mu)A^{k+1} + B^k - Z)\| = 0$. Consequently, condition (7) is fulfilled with $\lim_{k \rightarrow \infty} \varepsilon_a^k = 0$.

Proof. By the continuity of the mapping $(A^k, B^k) \mapsto H^k$ and Lemma 3.10(i), we have $\lambda_{\min}(H^k) \geq \lambda_{\min}(\text{Hess}_A f(A^*, B^*))/2 > 0$ for all sufficiently large k . Thus the backtracking dogleg search in step 3 of Algorithm 3.7 is initiated with $\tau^k = 2$, or $\Delta^k(\tau^k) = \Delta_N^k = -(H^k)^{-1}[g^k]$. Note that, due to Lemma 3.9(iii), both $\|g^k\|$ and $\|\Delta_N^k\|$ can be assumed to be sufficiently close to 0. Since $\Delta \in T_{\bar{\mathcal{M}}}(A) \mapsto P_{\bar{\mathcal{M}}}(A + \Delta) \in \bar{\mathcal{M}}$ is a second-order retraction on $\bar{\mathcal{M}}$ near A (see Example 18 in [6]), we have the following Taylor expansion:

$$\begin{aligned} f_A^k(P_{\bar{\mathcal{M}}}(A^k + \Delta_N^k)) &= f_A^k(A^k) + \langle g^k, \Delta_N^k \rangle + \frac{1}{2} \langle \Delta_N^k, H^k[\Delta_N^k] \rangle + o(\|\Delta_N^k\|^2) \\ &= f_A^k(A^k) - \frac{1}{2} \langle \Delta_N^k, H^k[\Delta_N^k] \rangle + o(\|\Delta_N^k\|^2), \quad \text{as } k \rightarrow \infty. \end{aligned}$$

Meanwhile, it follows from $A^{k+1} = P_{\bar{\mathcal{M}}}(A^k + \Delta_N^k) = A^k + \Delta_N^k + o(\|\Delta_N^k\|)$ that

$$\|A^{k+1} - A^k\|^2 = \|\Delta_N^k\|^2 + o(\|\Delta_N^k\|^2), \quad \text{as } k \rightarrow \infty.$$

Thus, altogether we have

$$\begin{aligned} f(A^k, B^k) - f(A^{k+1}, B^k) &= f_A^k(A^k) - f_A^k(A^{k+1}) = \frac{1}{2} \langle \Delta_N^k, H^k[\Delta_N^k] \rangle + o(\|\Delta_N^k\|^2) \\ &\geq \frac{\lambda_{\min}(\text{Hess}_A f(A^*, B^*))}{4} \|\Delta_N^k\|^2 + o(\|\Delta_N^k\|^2) \\ &= \frac{\lambda_{\min}(\text{Hess}_A f(A^*, B^*))}{4} \|A^{k+1} - A^k\|^2 + o(\|\Delta_N^k\|^2) \geq \delta \|A^{k+1} - A^k\|^2, \end{aligned}$$

i.e. $\Delta^k(\tau^k) = \Delta_N^k$ is admissible.

Now consider $A^{k+1} = \phi^A(A^k, B^k)$, where $\phi^A : \bar{\mathcal{M}} \times \mathcal{N} \rightarrow \bar{\mathcal{M}}$ is defined by the following system of equations:

$$\phi^A(A, B) = P_{\bar{\mathcal{M}}}(A + \Delta) =: \rho(A, \Delta), \quad (19)$$

$$\nabla_{\Delta} \text{grad}_A f(A, B) = -\text{grad}_A f(A, B). \quad (20)$$

If $(A^k, B^k) = (A^*, B^*)$, then we have $\text{grad}_A f(A^*, B^*) = 0$. Moreover, since $\text{Hess}_A f(A^*, B^*)$ is invertible, we have $\Delta = 0$ and thus $A^{k+1} = \rho(A^*, 0) = A^*$. Let us perturb ϕ^A at (A^*, B^*) with respect to the first argument along some $\Lambda_A \in T_{\bar{\mathcal{M}}}(A^*)$, which yields

$$D_A \phi^A(A^*, B^*)[\Lambda_A] = D_A \rho(A^*, 0)[\Lambda_A] + D_{\Delta} \rho(A^*, 0)[D_A \Delta(A^*, B^*)[\Lambda_A]].$$

Since $D_A \rho(A^*, 0)[\Lambda_A] = \Lambda_A$ and $D_{\Delta} \rho(A^*, 0)[\cdot] = \text{id}_{T_{\bar{\mathcal{M}}}(A^*)}(\cdot)$ on $T_{\bar{\mathcal{M}}}(A^*)$, we have

$$D_A \phi^A(A^*, B^*)[\Lambda_A] = \Lambda_A + D_A \Delta(A^*, B^*)[\Lambda_A].$$

The function $\Delta(A, B)$ is implicitly defined through equation (20), and in particular $\Delta(A^*, B^*) = 0$.

Next we use a calculus approach to show the following identity:

$$D_A \text{grad}_A f(A^*, B^*)[D_A \Delta(A^*, B^*)[\Lambda_A]] = -D_A \text{grad}_A f(A^*, B^*)[\Lambda_A]. \quad (21)$$

Let Γ_A denote the matrix-form Christoffel symbols of $\bar{\mathcal{M}}$ around A (see, e.g., [14]) such that Γ_A is symmetric, bilinear, and $\nabla_{\eta(A)}\xi(A) = D\xi(A)[\eta(A)] + \Gamma_A[\xi(A), \eta(A)] \in T_{\bar{\mathcal{M}}}(A)$ for any two smooth vector fields ξ and η on $\bar{\mathcal{M}}$. Then equation (20) can be rewritten as follows:

$$D_A \text{grad}_A f(A, B)[\Delta(A, B)] + \Gamma_A[\text{grad}_A f(A, B), \Delta(A, B)] = -\text{grad}_A f(A, B).$$

By perturbing the above equation at (A^*, B^*) along $(\Lambda_A, 0) \in T_{\bar{\mathcal{M}}}(A^*) \times T_{\mathcal{N}}(B^*)$, we have

$$\begin{aligned} & D_A^2 \text{grad}_A f(A^*, B^*)[\Delta(A^*, B^*), \Lambda_A] + D_A \text{grad}_A f(A^*, B^*)[D_A \Delta(A^*, B^*)[\Lambda_A]] \\ & + \Gamma_{A^*}[D_A \text{grad}_A f(A^*, B^*)[\Lambda_A], \Delta(A^*, B^*)] + \Gamma_{A^*}[\text{grad}_A f(A^*, B^*), D_A \Delta(A^*, B^*)[\Lambda_A]] \\ & = -D_A \text{grad}_A f(A^*, B^*)[\Lambda_A]. \end{aligned}$$

Crossing out the vanishing terms, we obtain (21) as claimed. Note that $\text{grad}_A f(A, B) = P_{T_{\bar{\mathcal{M}}}(A)}((1 + \mu)A + B - Z)$ and thus $D_A \text{grad}_A f(A^*, B^*)[\cdot] = (1 + \mu)\text{id}_{T_{\bar{\mathcal{M}}}(A^*)}(\cdot)$ on $T_{\bar{\mathcal{M}}}(A^*)$. Thus we have

$$D_A \phi^A(A^*, B^*) = 0.$$

Analogously, we perturb ϕ^A at (A^*, B^*) with respect to the second argument along $(0, \Lambda_B) \in T_{\bar{\mathcal{M}}}(A^*) \times T_{\mathcal{N}}(B^*)$. This leads to

$$D_B \phi^A(A^*, B^*)[\Lambda_B] = D_B \Delta(A^*, B^*)[\Lambda_B].$$

Again by the calculus approach, we derive

$$D_A \text{grad}_A f(A^*, B^*)[D_B \Delta(A^*, B^*)[\Lambda_B]] = -D_B \text{grad}_A f(A^*, B^*)[\Lambda_B].$$

Note that $D_B \text{grad}_A f(A^*, B^*)[\Lambda_B] = P_{T_{\bar{\mathcal{M}}}(A^*)}(\Lambda_B)$. Then it follows that

$$D_B \phi^A(A^*, B^*)[\Lambda_B] = P_{T_{\bar{\mathcal{M}}}(A^*)}(\Lambda_B).$$

By the Taylor expansion of ϕ^A at (A^*, B^*) , we have the following estimate

$$\begin{aligned} & \|A^{k+1} - A^*\| = \|\phi^A(A^k, B^k) - \phi^A(A^*, B^*)\| \\ & \leq \|D_A \phi^A(A^*, B^*)(A^k - A^*)\| + \|D_B \phi^A(A^*, B^*)(B^k - B^*)\| + o(\|A^k - A^*\|) + o(\|B^k - B^*\|) \\ & = \|P_{T_{\bar{\mathcal{M}}}(A^*)}(B^k - B^*)\| + o(\|A^k - A^*\|) + o(\|B^k - B^*\|), \quad \text{as } k \rightarrow \infty. \end{aligned} \quad (22)$$

In order to obtain an estimate on $B^k - B^*$, consider the mapping $\phi^B(A, B) := P_{T_{\mathcal{N}}(B)}(A + B - Z)$. Let B^k be sufficiently close to B^* such that $\|B^k\|_0 = s$ and $B^k - B^* \in T_{\mathcal{N}}(B^*)$. Due to our assumption on the sequence $\{B^k\}$ and Theorem 3.4, we have $\phi^B(A^k, B^k) = \phi^B(A^*, B^*) = 0$. Moreover, the derivatives of ϕ^B are given by $D_A \phi^B(A^*, B^*) = P_{T_{\mathcal{N}}(B^*)}$ and $D_B \phi^B(A^*, B^*) = \text{id}_{T_{\mathcal{N}}(B^*)}$. Thus the Taylor expansion of ϕ^B at (A^*, B^*) appears as

$$\begin{aligned} \phi^B(A^k, B^k) &= \phi^B(A^*, B^*) + D_A \phi^B(A^*, B^*)(A^k - A^*) + D_B \phi^B(A^*, B^*)(B^k - B^*) \\ &+ o(\|A^k - A^*\|) + o(\|B^k - B^*\|), \quad \text{as } k \rightarrow \infty, \end{aligned}$$

which further implies that

$$B^k - B^* = -P_{T_{\mathcal{N}}(B^*)}(A^k - A^*) + o(\|A^k - A^*\|) + o(\|B^k - B^*\|), \quad \text{as } k \rightarrow \infty. \quad (23)$$

In particular, we have $\|B^k - B^*\| \leq O(\|A^k - A^*\|)$ as $k \rightarrow \infty$.

By plugging (23) into (22), it follows from Lemma 3.10(iii) that

$$\begin{aligned}\|A^{k+1} - A^*\| &\leq \|(P_{T_{\bar{\mathcal{M}}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)})(A^k - A^*)\| + o(\|A^k - A^*\|) \\ &\leq \kappa_p \|A^k - A^*\| + o(\|A^k - A^*\|),\end{aligned}$$

for all sufficiently large k . This proves our claim (ii).

Finally, in view of the convergence of $\{(A^k, B^k)\}$ to (A^*, B^*) as well as Lemma 3.9(i), we conclude that $\lim_{k \rightarrow \infty} \|P_{T_{\bar{\mathcal{M}}}(A^{k+1})}((1 + \mu)A^{k+1} + B^k - Z)\| = 0$ and that condition (7) is fulfilled with $\lim_{k \rightarrow \infty} \varepsilon_a^k = 0$. \square

We end this subsection by noting that the dependence of δ on (A^*, B^*) is certainly delicate. In our numerics, however, the choice of δ turned out to be rather unproblematic, even for $\mu = 0$ as in section 4. Concerning the complexity of the low-rank subproblem solver, note that the computation of Δ_N^k in step 2 of Algorithm 3.7 possibly requires solving the linear system involving H^k . Under the assumption on the positive definiteness of the Riemannian Hessian in Theorem 3.11, which imitates the second-order sufficient optimality condition in classical (unconstrained, Euclidean) optimization, each H^k -system solve, up to certain fixed tolerance of the error, can be carried out by the conjugate gradient method within a uniformly bounded number of iterations. Thus, the overall complexity for the low-rank subproblem solver at each iteration is no more than $O(mnr)$ flops. In addition, we remark that the constant κ_p in Lemma 3.10(iii), which in fact measures the angle between the tangent spaces $T_{\bar{\mathcal{M}}}(A^*)$ and $T_{\mathcal{N}}(B^*)$, is an intrinsic quantification of the local identifiability [12] at (A^*, B^*) . Even though our alternating minimizer scheme solves its subproblems only inexactly, its asymptotical convergence rate (i.e. κ_p) is equally fast as that attained by the (exact) alternating projection method. When $0 \leq \kappa_p < 1$, (A^*, B^*) is a strict local minimizer, and $\{A^k\}$ converges to A^* q -linearly at rate κ_p , as shown in Theorem 3.11(ii). In case $\kappa_p = 0$, or equivalently $T_{\bar{\mathcal{M}}}(A^*)$ and $T_{\mathcal{N}}(B^*)$ are perpendicular to each other, the convergence of $\{A^k\}$ to A^* is even superlinear.

3.4 Alternating minimization scheme with trimming

The favorable performance of Algorithm 3.1 depends on a proper choice of r and s . If either r or s is too small, the constraint will rule out the desired solution. On the other hand, if either r or s is too large, the convergence property of Algorithm 3.1 is in danger due to the rank- or cardinality-deficiency at the desired solution. In this subsection we resolve this issue by incorporating a heuristic trimming procedure into the alternating minimization scheme which allows an adaptive tuning of r and s . The trimming of the matrix A^k is based on the k -means clustering algorithm [28], and the trimming of the matrix B^k is based on a hard-thresholding.

In brief, we initialize the algorithm by some safe choices of r^1 and s^1 that are larger than the underlying r and s , respectively. As the iterates $A^k \in \bar{\mathcal{M}}(r^k)$ tend to settle, we partition the r^k largest singular values of A^k (in logarithmic scale) into two clusters by the k -means algorithm. If the gap between the means of the two clusters is larger than some prescribed threshold, then we set r^{k+1} to be the cardinality of the cluster of the larger mean, and replace the old A^k by its projection onto $\bar{\mathcal{M}}(r^{k+1})$. On the other hand, when the iterates $B^k \in \mathcal{N}(s^k)$ tend to stabilize along the sequence, we replace those entries of B^k , which are less than some threshold in absolute value, by 0 and set $s^{k+1} := \|B^k\|_0$. The detailed implementation of the alternating minimization scheme with trimming is specified in the following.

Algorithm 3.12 (Alternating minimization scheme with trimming).

Choose $\delta > 0$, $\nu_a > 0$, $\nu_b > 0$, $\theta_a > 0$, $\theta_b > 0$. Initialize $r^1 \in \mathbb{N}$, $s^1 \in \mathbb{N}$, $A^0 \in \bar{\mathcal{M}}(r^1)$, $B^0 = P_{\mathcal{N}(s^1)}(Z - A^0)$. Set $k = 1$ and iterate:

1. Compute A^k as an approximate solution of the A -subproblem $\min_{A \in \bar{\mathcal{M}}(r^k)} \frac{1}{2} \|A + B^{k-1} - Z\|^2$ by Algorithm 3.7, which is represented in the compact SVD form $A^k = U^k \Sigma^k (V^k)^\top$.
2. Compute B^k as an approximate solution of the B -subproblem $\min_{B \in \mathcal{N}(s^k)} \frac{1}{2} \|A^k + B - Z\|^2$ by Algorithm 3.3.
3. If $\|A^k - A^{k-1}\| / \|A^{k-1}\| > \nu_a$, then set $r^{k+1} := r^k$; otherwise trim A^k as follows:
 - (a) Partition the logarithms of the r^k largest singular values of A^k , namely $\{\log \sigma_j^k\}_{j=1}^{r^k}$, into two disjoint sets $\{\log \sigma_j^k\}_{j \in \mathcal{I}_1}$ and $\{\log \sigma_j^k\}_{j \in \mathcal{I}_2}$ by the k -means clustering algorithm (with $|\mathcal{I}_1| + |\mathcal{I}_2| = r^k$).
 - (b) Evaluate the means of the two clusters; i.e. $m_1 := (\sum_{j \in \mathcal{I}_1} \log \sigma_j^k) / |\mathcal{I}_1|$ and $m_2 := (\sum_{j \in \mathcal{I}_2} \log \sigma_j^k) / |\mathcal{I}_2|$. Assume $m_1 \geq m_2$ without loss of generality.
 - (c) If $m_1 - m_2 > \theta_a$, then set $r^{k+1} := |\mathcal{I}_1|$, $U^k := [\{U_j^k\}_{j \in \mathcal{I}_1}]$, $V^k := [\{V_j^k\}_{j \in \mathcal{I}_1}]$, $\Sigma^k := \text{diag}(\{\sigma_j^k\}_{j \in \mathcal{I}_1})$, and $A^k := U^k \Sigma^k (V^k)^\top$.
4. If $\|B^k - B^{k-1}\| / \|B^{k-1}\| > \nu_b$, then set $s^{k+1} := s^k$; otherwise set $B_{ij}^k := 0$ whenever $|B_{ij}^k| < \theta_b$ and update $s^{k+1} := \|B^k\|_0$.
5. If a suitable stopping criterion is satisfied, then stop; otherwise increase k by 1 and return to step 1.

4 Numerical experiments

In this section, we study the numerical performance of Algorithm 3.12. The following parameters in the algorithm are fixed throughout the experiments: $\mu = 0$, $\delta = 0.1$, $\nu_a = \nu_b = 0.2$. Note that although it is favorable to consider $\mu > 0$ so as to guarantee the existence of a solution (see Theorem 2.1), we experience no troubles in our numerical experiments when choosing $\mu = 0$. Concerning the initialization, given any $A^0 \in \bar{\mathcal{M}}(r^1)$, we always take $B^0 = P_{\mathcal{N}(s^1)}(Z - A^0)$ accordingly. The inversion of the linear system (12) for computing the Newton step Δ_N^k is carried out by the conjugate gradient method with fixed residual tolerance 0.01. It turns out that this (approximate) Newton step in resolving the A -subproblem is so good that it is admissible, i.e. $\Delta^k(\tau^k) = \Delta_N^k$ fulfills condition (14), in almost every iteration. In addition, all partial SVDs are performed using the PROPACK routine `1ansvd` [3], which should be distinguished from the (full) SVDs using the MATLAB routine `svd`.

The experiments are performed under MATLAB R2011b on a 2.66 GHz Intel Core Laptop with 4 GB RAM. All CPU-time reported in this section is measured in seconds.

4.1 Numerical behavior

We apply our algorithm to a test example of robust principal component pursuit. Let $m = n = 400$, $r = 0.05n$, $s = 0.05n^2$, and the observation matrix is generated by $Z = A_{true} + B_{true} + N$. The rank- r matrix $A_{true} = L_{true} R_{true}^\top$ is generated by the product of two matrices $L_{true} \in \mathbb{R}^{m \times r}$ and $R_{true} \in \mathbb{R}^{n \times r}$, both of which have entries independently sampled from a normal distribution of mean 0 and standard deviation 1. The sparse matrix B_{true} has s nonzero entries, whose locations are randomly chosen and whose values are independently sampled from $\{\pm\sqrt{n}\}$ with uniform probability. The matrix N contains white Gaussian noise of mean 0 and standard deviation 0.001. In this example, we choose $\theta_a = \log 5$, $\theta_b = 0.2\sqrt{n}$.

Since our algorithm intends to find a local solution for the nonconvex minimization problem (1), it is important to check the quality of such a local solution as well as the dependence on the initial guess (A^0, B^0) . In the following test, we consider two different choices for A^0 , namely $A^0 = P_{\bar{\mathcal{M}}(r^1)}(Z)$ and A^0 being the projection of a random Gaussian matrix onto $\bar{\mathcal{M}}(r^1)$. Meanwhile, we also investigate the effectiveness of the trimming procedure for tuning r^k and s^k , provided that the true values of r and s are not available at the beginning. In this test, we allow r^1 and s^1 to be overestimations with respect to the true r and s up to 100%. The iterative algorithm is terminated once the relative error $\|A^k - A_{true}\|/\|A_{true}\|$ drops below 2×10^{-4} .

In Table 1, we report the corresponding relative error and the CPU-time. It is observed that the quality of the solutions produced by Algorithm 3.12, measured by the relative error, is robust to different initializations. Nonetheless, we remark that the efficiency of the algorithm is correlated to the choices of r^1 , s^1 , and A^0 . As it can be expected, the initial guess $A^0 = P_{\bar{\mathcal{M}}(r^1)}(Z)$ is superior to a randomly chosen A^0 with respect to CPU-time, while choosing r^1 and s^1 closer to the underlying r and s yields faster convergence.

We further illustrate the numerical behavior of the algorithm, for instance, when $r^1 = 1.5r$, $s^1 = 1.5s$, and $A^0 = P_{\bar{\mathcal{M}}(r^1)}(Z)$. In Figure 1, we provide the semi-logarithmic plots of the objective value $f(A^k, B^k)$, the residual norm $\|\text{grad}_A f(A^k, B^k)\|$, and the convergence errors $\|A^k - A^*\|/\|A^*\|$ and $\|B^k - B^*\|/\|B^*\|$. The limit points A^* and B^* are precomputed with sufficiently high accuracy. It is observed from Figure 1(c) that, as is theoretically justified in Theorem 3.11, the sequence $\{A^k\}$ indeed exhibits a linear convergence, and the asymptotical convergence rate in this example is about 0.26.

r^1	s^1	$A^0 = P_{\bar{\mathcal{M}}(r^1)}(Z)$		random A^0	
		error	CPU	error	CPU
r	s	1.78e-4	1.32	1.19e-4	1.76
$1.25r$	$1.25s$	1.73e-4	1.74	1.67e-4	2.05
$1.5r$	$1.5s$	1.84e-4	1.77	1.35e-4	2.36
$1.75r$	$1.75s$	1.32e-4	2.03	1.01e-4	2.64
$2r$	$2s$	1.16e-4	2.17	1.83e-4	2.75

Table 1: Initialization study.

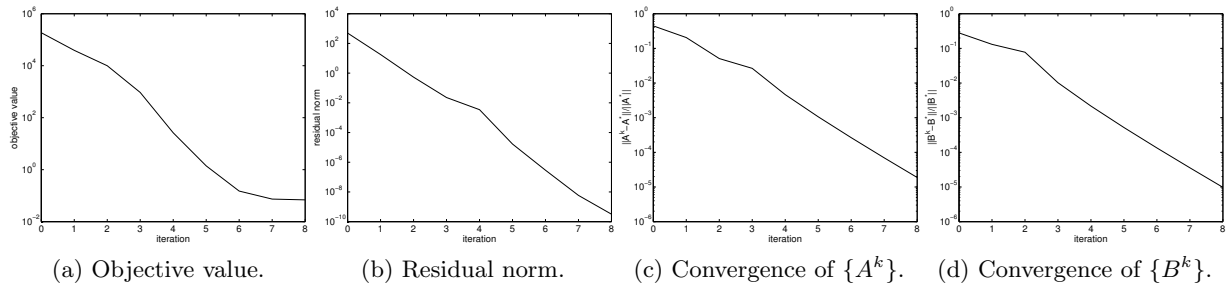


Figure 1: Convergence behavior.

4.2 Comparison with an augmented Lagrangian method

A comprehensive comparison of numerical solvers on the (convex) nuclear-plus- ℓ^1 -norm model for robust principal component pursuit can be found on the webpage [2]. Among those solvers,

the augmented Lagrangian method [10, 23, 31] seems to be the most efficient one in practice. Hence, in the following we compare the performances of our alternating minimization scheme and the augmented Lagrangian method with implementation-wise variations on both methods. More specifically, we implement Algorithm 3.12 both with and without the global minimization trial step for the low-rank subproblem (i.e. step 0 of Algorithm 3.7), which are abbreviated by “AMS#” and “AMS” respectively. The implementation of the augmented Lagrangian method essentially follows Algorithm 1 in [10]. The major computational cost of this algorithm lies in an SVD in full dimension for performing a “singular value thresholding” at each iteration. As pointed out by [9], it is possible to accelerate the singular value thresholding via partial SVD [3]. Different from the context in [9], however, the target matrix (for SVD) in our matrix decomposition problem is dense and unstructured in general, and thus this acceleration strategy should only be utilized when the rank of the target matrix is predictably low. In our experiments, we implement the augmented Lagrangian method with full SVDs only (abbreviated by “fSVD-ALM”), and also its partial-SVD variant (abbreviated by “pSVD-ALM”) where one switches from full SVD to partial SVD once the rank of the low-rank component A^k in the previous iteration drops below an empirical threshold equal to $0.2n$.

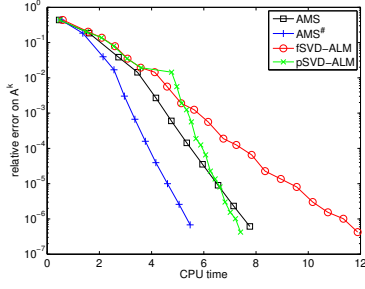
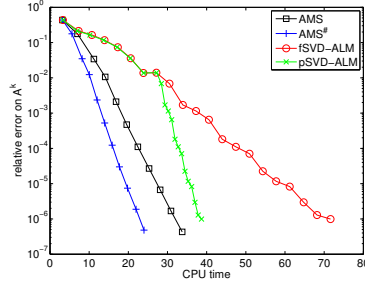
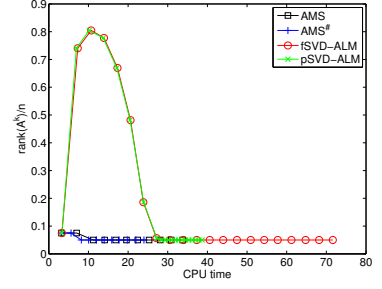
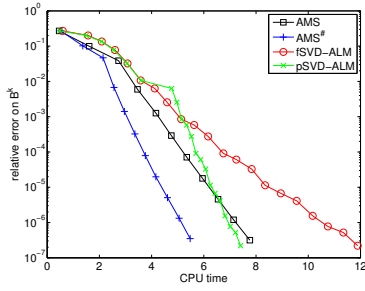
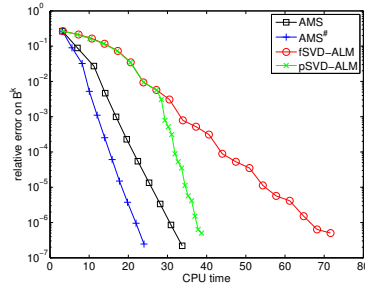
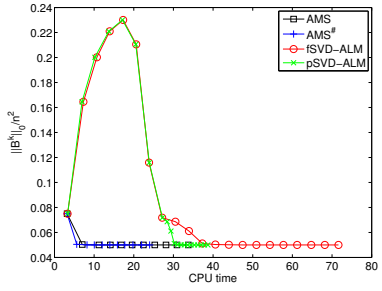
The test data is generated in the same way as described in the first paragraph of section 4.1, except for $N = 0$. Thus, the exact recovery of A_{true} and B_{true} is expected for all candidate methods, namely AMS, AMS#, fSVD-ALM, and pSVD-ALM. In this example, we choose $\theta_a = \log 5$, $\theta_b = 0.2\sqrt{n}$ in AMS and AMS#. Besides, we assume a moderate initial estimate (rather than the exact knowledge) of r and s such that $r^1 = 1.5r$, $s^1 = 1.5s$. For a fair comparison, we use the same initial guesses, i.e. $A^0 = P_{\mathcal{M}(r^1)}(Z)$, $B^0 = P_{\mathcal{N}(s^1)}(Z - A^0)$, for all candidate methods. The experiments are performed with different combinations of n , r , and s .

The corresponding comparisons among the four candidate methods with respect to relative errors, measured by $\|A^k - A_{true}\|/\|A_{true}\|$ and $\|B^k - B_{true}\|/\|B_{true}\|$, and CPU time are demonstrated in Figure 2. It is observed in the experiments that AMS# always accepts the global minimizers from both the A - and B -subproblems, and essentially behaves like a heuristic alternating projection method (see the Appendix for a description) known to be locally linearly convergent. In this particular example, AMS# works extremely well owing to a good initial guess so that the local convergence of the alternating projection method is immediately activated from the beginning. Nevertheless, the reader should be cautioned that in general such convergence behavior is not guaranteed for the alternating projection method with arbitrary initial guesses, and under such circumstances the global minimization trial steps are most likely wasteful. On the other hand, the plots on the relative errors in Figure 2 indicate that AMS, with guaranteed global convergence, has rather close performance to AMS#, especially for larger scales. Although partial SVDs typically improve the augmented Lagrangian method over the asymptotical convergence rate, expensive full SVDs are inevitable at early iterations; see the plots of the rank transitions of $\{A^k\}$ (when $n = 2000$) in the rightmost column of Figure 2. In comparison, AMS and AMS# capture the rank of the low-rank component and the cardinality of the sparse component efficiently, thanks to the heuristic trimming procedure, and thus outperform fSVD-ALM and pSVD-ALM for large scales.

4.3 Application to background-foreground separation of surveillance video

We apply our algorithm to background-foreground separation of surveillance videos. Our first test video, which is taken from [22, 10] and also publicly available [4], is a sequence of 200 frames taken in an airport. Each frame is a gray-level image of resolution 144×176 , and is stacked as one column in the data matrix $Z \in \mathbb{R}^{25344 \times 200}$; i.e. $m = 25344$, $n = 200$. Our goal is to extract

$$r/n = s/n^2 = 0.05$$

 $n = 1000$
 $n = 2000$

 (a) Relative error on $\{A^k\}$.

 (b) Relative error on $\{A^k\}$.

 (c) Rank transition of $\{A^k\}$.

 (d) Relative error on $\{B^k\}$.

 (e) Relative error on $\{B^k\}$.

 (f) Cardinality transition of $\{B^k\}$.

$$r/n = s/n^2 = 0.1$$

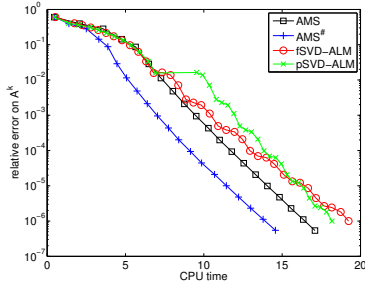
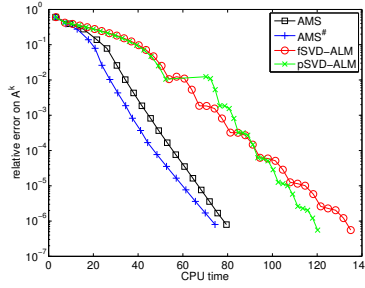
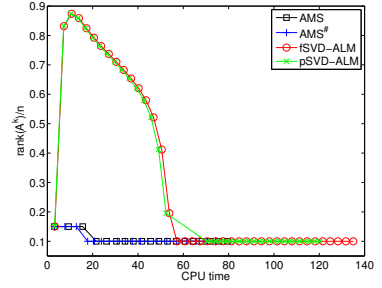
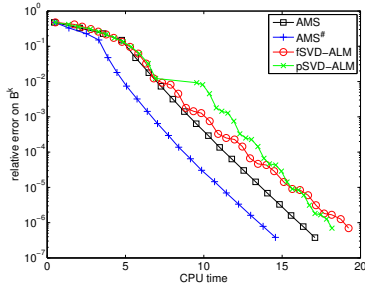
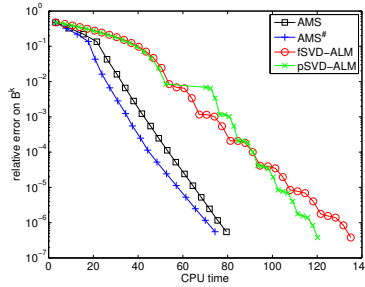
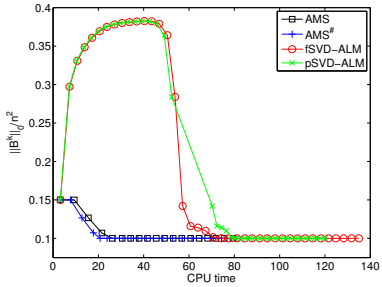
 $n = 1000$
 $n = 2000$

 (g) Relative error on $\{A^k\}$.

 (h) Relative error on $\{A^k\}$.

 (i) Rank transition of $\{A^k\}$.

 (j) Relative error on $\{B^k\}$.

 (k) Relative error on $\{B^k\}$.

 (l) Cardinality transition of $\{B^k\}$.

Figure 2: Comparison with augmented Lagrangian method.

from Z the static background (as the low-rank matrix A) and the moving foreground (as the sparse matrix B).

We implement the alternating minimization scheme (AMS) with $\theta_a = \log 10$, $\theta_b = 0.12$, $A^0 = P_{\mathcal{M}(r^1)}(Z)$, $r^1 = 5$, and $s^1 \approx 0.1mn$, which is terminated once the residual norm $\|\text{grad}_A f(A^k, B^k)\|$ is reduced by a factor of 10^{-4} . It takes 39.4 seconds for AMS to converge, and the ultimate value of r^k is equal to 1 and $s^k \approx 0.0483mn$. The corresponding extractions for three selected frames are displayed in columns (b) and (c) in Figure 3. For comparison, we also perform the extraction using the augmented Lagrangian method (ALM). The implementation of ALM again follows [10], and we terminate the iterations once $\|Z - A^k - B^k\|/\|Z\| \leq 10^{-4}$. We note that only full SVDs are implemented in ALM, as partial SVDs do not lead to CPU gain in this problem. The results by ALM are shown in columns (d) and (e), and it takes 124.4 seconds for ALM to converge.

Our second example is a 400-frame sequence taken in a lobby with varying illumination [22, 10, 4]. Each frame is of resolution 128×160 , and the data matrix Z is formulated as a 20480-by-400 matrix (i.e. $m = 20480$, $n = 400$). We run AMS with the same parameters as in the previous example except for $\theta_b = 0.06$, which is smaller than before, so that we allow more information in the sparse matrix. The algorithm converges after 69.19 seconds, and the ultimate value of r^k is equal to 2 and $s^k \approx 0.00413mn$. We also implement ALM using the same setting as before, for which it takes 193.5 seconds to converge. The separation results of both methods are displayed in Figure 4.

We conclude from the experiments that AMS performs well in background-foreground separation of surveillance videos, which is robust to the variation of illumination. In comparison with ALM, AMS typically eliminates the moving shadows in the backgrounds that occur in ALM, and provides sharper extractions of the moving foregrounds. Moreover, AMS has considerable advantage over ALM with respect to CPU-time.

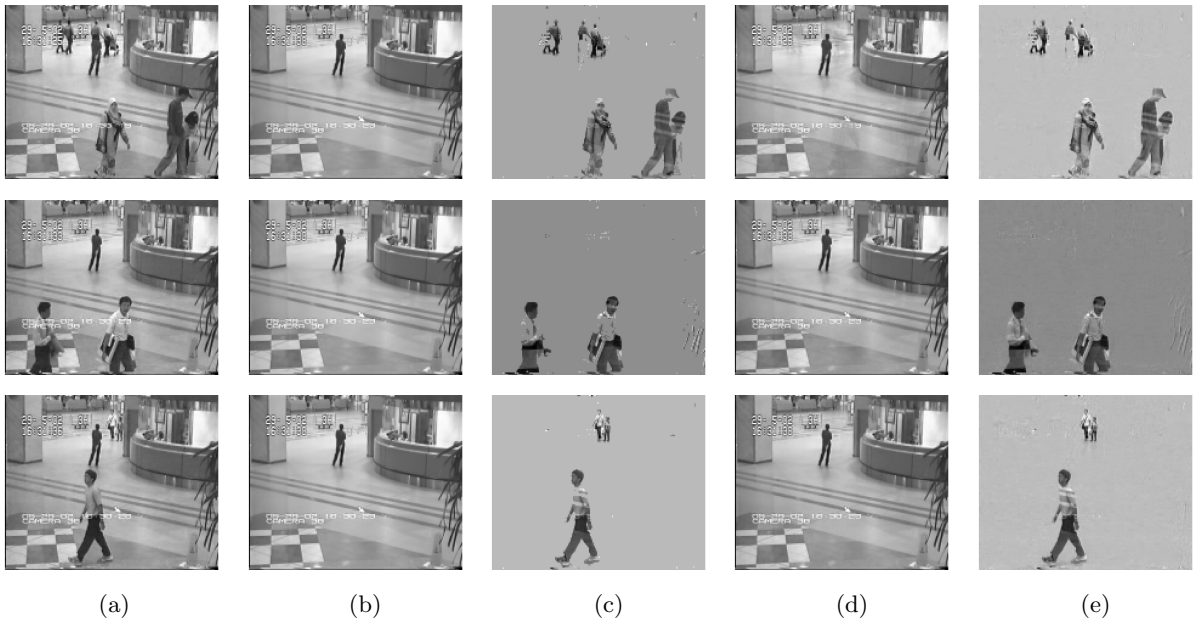


Figure 3: Background-foreground separation (airport): (a) original frames; (b) background via AMS; (c) foreground via AMS; (d) background via ALM; (e) foreground via AMS. The CPU-time consumed by AMS and ALM is 39.4 and 124.4 seconds, respectively.

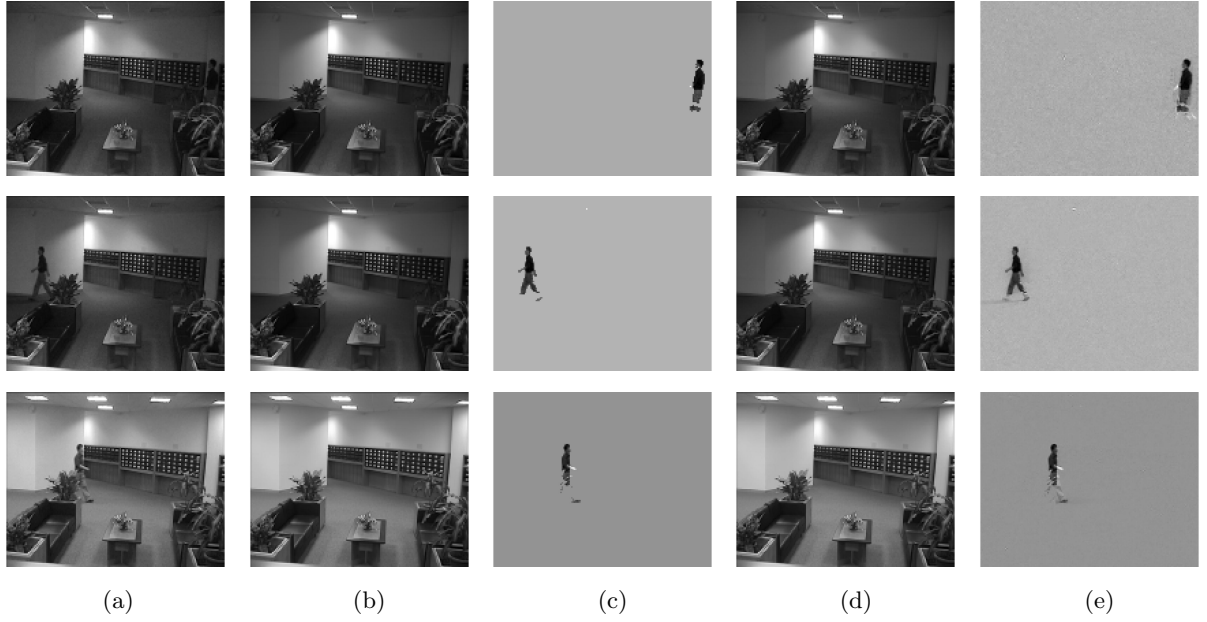


Figure 4: Background-foreground separation (lobby): (a) original frames; (b) background via AMS; (c) foreground via AMS; (d) background via ALM; (e) foreground via AMS. The CPU-time consumed by AMS and ALM is 69.19 and 193.5 seconds, respectively.

Appendix: Local convergence of an alternating projection method.

Here we consider a heuristic alternating projection method for the RPCP problem. This method, which can be interpreted as an exact alternating minimizer scheme for the optimization problem (1) with $\mu = 0$, can be shortly described as follows. Given $A^k \in \mathcal{M}$, one generates

$$\begin{cases} B^{k+1} := P_{\mathcal{N}}(Z - A^k), \\ A^{k+1} := P_{\mathcal{M}}(Z - B^{k+1}). \end{cases} \quad (24)$$

The name ‘‘alternating projection method’’ is termed, since the iterative procedure (on $\{A^k\}$) can be expressed as

$$A^{k+1} = \psi(A^k) := (P_{\mathcal{M}} \circ \iota \circ P_{\mathcal{N}} \circ \iota)(A^k), \quad (25)$$

with $\iota : A \mapsto Z - A$, and thus generalizes the classical alternating projection (where ι is the identity map) in, e.g, [21]. The following theorem asserts the local convergence of the alternating projection method. However, we note that the global convergence for this method is not guaranteed in general.

Theorem A.1. *Given $A^0 \in \mathcal{M}$, let the sequence $\{(A^k, B^k)\}$ be iteratively generated by formula (24). Assume that (A^k, B^k) is sufficiently close to some (A^*, B^*) such that $\text{rank}(A^*) = r$, $\|B^*\| = s$, $T_{\mathcal{M}}(A^*) \cap T_{\mathcal{N}}(B^*) = \{0\}$, and moreover*

$$\begin{cases} B^* := P_{\mathcal{N}}(Z - A^*), \\ A^* := P_{\mathcal{M}}(Z - B^*). \end{cases} \quad (26)$$

Then $\{(A^k, B^k)\}$ converges to (A^, B^*) q -linearly at rate κ_p ; i.e.*

$$\limsup_{k \rightarrow \infty} \frac{\|(A^{k+1}, B^{k+1}) - (A^*, B^*)\|}{\|(A^k, B^k) - (A^*, B^*)\|} \leq \kappa_p,$$

where $\kappa_p \in [0, 1)$ is a constant (same as in Lemma 3.10) such that

$$\|(P_{T_{\mathcal{M}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)})(\Delta)\| \leq \kappa_p \|\Delta\|,$$

for all $\Delta \in \mathbb{R}^{m \times n}$.

Proof. We only prove the q -linear convergence on $\{A^k\}$, as the proof for $\{B^k\}$ is almost identical. Note that \mathcal{M} and \mathcal{N} are two smooth manifolds near A^* and B^* , respectively. For the existence of a qualified constant κ_p , we refer to Lemma 3.10(iii).

In the following, we perturb both equations in (26) with respect to A^* by an arbitrarily fixed $\Delta \in \mathbb{R}^{m \times n}$. The perturbation of the first equation gives

$$P_{\mathcal{N}}(Z - A^* - \Delta) = B^* + P_{\mathcal{N}}(Z - A^* - \Delta) - P_{\mathcal{N}}(Z - A^*) = B^* + P_{T_{\mathcal{N}}(B^*)}(-\Delta) + O(\|\Delta\|^2).$$

Since A^* is a fixed point of the map ψ in (25), the second equation in (26) can be written as $\psi(A^*) = P_{\mathcal{M}}(Z - B^*)$. Then we have

$$\begin{aligned} \psi(A^* + \Delta) &= P_{\mathcal{M}}(Z - P_{\mathcal{N}}(Z - A^* - \Delta)) = P_{\mathcal{M}}(Z - B^* - P_{T_{\mathcal{N}}(B^*)}(-\Delta) + O(\|\Delta\|^2)) \\ &= A^* + P_{\mathcal{M}}(Z - B^* - P_{T_{\mathcal{N}}(B^*)}(-\Delta) + O(\|\Delta\|^2)) - P_{\mathcal{M}}(Z - B^*) \\ &= A^* + (P_{T_{\mathcal{M}}(A^*)} \circ P_{T_{\mathcal{N}}(B^*)})(\Delta) + O(\|\Delta\|^2). \end{aligned}$$

Thus, by considering $\Delta = A^k - A^*$ and passing $\Delta \rightarrow 0$, we conclude that

$$\limsup_{k \rightarrow \infty} \frac{\|A^{k+1} - A^*\|}{\|A^k - A^*\|} \leq \kappa_p.$$

□

References

- [1] *LMaFit: Low-rank matrix fitting*. <http://lmafit.blogs.rice.edu>.
- [2] *Low-rank matrix recovery and completion via convex optimization*. http://perception.csl.illinois.edu/matrix-rank/sample_code.html.
- [3] *PROPACK—Software for large and sparse SVD calculations*. <http://sun.stanford.edu/~rmunk/PROPACK/>.
- [4] *Statistical modeling of complex background for foreground object detection*. http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html.
- [5] P.-A. ABSIL, R. MAHONY, AND R. SEPULCHRE, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, New Jersey, 2008.
- [6] P.-A. ABSIL AND J. MALICK, *Projection-like retractions on matrix manifolds*, *SIAM J. Optim.*, 22 (2012), pp. 135–158.
- [7] N. BOUMAL AND P.-A. ABSIL, *RTRMC: A Riemannian trust-region method for low-rank matrix completion*, in *Advances in Neural Information Processing Systems*, 2011, pp. 406–414.

- [8] N. BOUMAL, B. MISHRA, P.-A. ABSIL, AND R. SEPULCHRE, *Manopt, a Matlab toolbox for optimization on manifolds*, J. Mach. Learn. Res., 15 (2014), pp. 1455–1459. <http://www.manopt.org>.
- [9] J. F. CAI, E. J. CANDÈS, AND Z. SHEN, *A singular value thresholding algorithm for matrix completion*, SIAM J. Optim., 20 (2010), pp. 1956–1982.
- [10] E. J. CANDÈS, X. LI, Y. MA, AND J. WRIGHT, *Robust principal component analysis?*, J. ACM, 58 (2011), pp. 1–37.
- [11] E. J. CANDÈS AND B. RECHT, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), pp. 717–772.
- [12] V. CHANDRASEKARAN, S. SANGHAVI, P. A. PARRILO, AND A. S. WILLSKY, *Rank-sparsity incoherence for matrix decomposition*, SIAM J. Optim., 21 (2011), pp. 572–596.
- [13] C. ECKART AND G. YOUNG, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.
- [14] A. EDELMAN, T. A. ARIAS, AND S. T. SMITH, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.
- [15] P.-S. HUANG, S. D. CHEN, P. SMARAGDIS, AND M. HASEGAWA-JOHNSON, *Singing-voice separation from monaural recordings using robust principal component analysis*, in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 57–60.
- [16] H. JI, S. HUANG, Z. SHEN, AND Y. XU, *Robust video restoration by joint sparse and low rank matrix approximation*, SIAM J. Imaging Sci., 4 (2011), pp. 1122–1142.
- [17] K. JIA, T.-H. CHAN, AND Y. MA, *Robust and practical face recognition via structured sparsity*, in Proceedings of the 12th European Conference on Computer Vision, vol. 4, 2012, pp. 331–344.
- [18] R. H. KESHAVAN, A. MONTANARI, AND S. OH, *Matrix completion from a few entries*, IEEE Trans. Inform. Theory, 56 (2010), pp. 2980–2998.
- [19] ———, *Matrix completion from noisy entries*, J. Mach. Learn. Res., 11 (2010), pp. 2057–2078.
- [20] D. KNUTH, *The Art of Computer Programming*, vol. 3: *Sorting and Searching*, Addison-Wesley, 3rd ed., 1997.
- [21] A. S. LEWIS AND J. MALICK, *Alternating projections on manifolds*, Math. Oper. Res., 33 (2008), pp. 216–234.
- [22] L. LI, W. HUANG, I. Y.-H. GU, AND Q. TIAN, *Statistical modeling of complex backgrounds for foreground object detection*, IEEE Trans. Image Process., 13 (2004), pp. 1459–1472.
- [23] Z. LIN, M. CHEN, L. WU, AND Y. MA, *The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices*, Technical Report UILU-ENG-09-2215, UIUC, 2009.
- [24] G. MEYER, S. BONNABEL, AND R. SEPULCHRE, *Regression on fixed-rank positive semidefinite matrices: A Riemannian approach*, J. Mach. Learn. Res., 12 (2011), pp. 593–625.

- [25] K. MIN, Z. ZHANG, J. WRIGHT, AND Y. MA, *Decomposing background topics from keywords by principal component pursuit*, in Proceedings of the 19th ACM international conference on Information and knowledge management, 2010, pp. 269–278.
- [26] T. T. NGO AND Y. SAAD, *Scaled gradients on Grassmann manifolds for matrix completion*, Advances in Neural Information Processing Systems, 25 (2012), pp. 1421–1429.
- [27] J. NOCEDAL AND S. WRIGHT, *Numerical Optimization*, Springer, New York, 2nd ed., 2006.
- [28] G. A. F. SEBER, *Multivariate Observations*, John Wiley & Sons, Hoboken, NJ, 1984.
- [29] L. SIMONSSON AND L. ELDÉN, *Grassmann algorithms for low rank approximation of matrices with missing values*, BIT Numer. Math., 50 (2010), pp. 173–191.
- [30] S. T. SMITH, *Geometric Optimization Methods for Adaptive Filtering*, PhD thesis, Harvard University, 1993.
- [31] M. TAO AND X. YUAN, *Recovering low-rank and sparse components of matrices from incomplete and noisy observations*, SIAM J. Optim., 21 (2011), pp. 57–81.
- [32] L. N. TREFETHEN AND D. BAU, III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [33] B. VANDEREYCKEN, *Low-rank matrix completion by Riemannian optimization*, SIAM J. Optim., 23 (2013), pp. 1214–1236.
- [34] Z. WEN, W. YIN, AND Y. ZHANG, *Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm*, Math. Prog. Comp., 4 (2012), pp. 333–361.
- [35] Z. ZHANG, A. GANESH, X. LIANG, AND Y. MA, *TILT: Transform invariant low-rank textures*, Int. J. Comput. Vis., 99 (2012), pp. 1–24.