A Nonconvex Projection Method for Robust PCA

Aritra Dutta * Filip Hanzely † Peter Richtárik ‡

Abstract

Robust principal component analysis (RPCA) is a well-studied problem with the goal of decomposing a matrix into the sum of low-rank and sparse components. In this paper, we propose a nonconvex feasibility reformulation of RPCA problem and apply an alternating projection method to solve it. To the best of our knowledge, we are the first to propose a method that solves RPCA problem without considering any objective function, convex relaxation, or surrogate convex constraints. We demonstrate through extensive numerical experiments on a variety of applications, including shadow removal, background estimation, face detection, and galaxy evolution, that our approach matches and often significantly outperforms current state-of-the-art in various ways.

1 Introduction

Principal component analysis (PCA) [31] addresses the problem of best approximation of a matrix $A \in \mathbb{R}^{m \times n}$ by a matrix of rank $r$:

$$X^* = \arg \min_{X \in \mathbb{R}^{m \times n}} \| A - X \|_F^2,$$

where $\| \cdot \|_F$ denotes the Frobenius norm of matrices. The solution to (1) is given by

$$X^* = H_r(A) \overset{\text{def}}{=} U \Sigma_r V^\top,$$

where $A$ has singular value decompositions $A = U \Sigma V^\top$, and $\Sigma_r(A)$ is the diagonal matrix obtained from $\Sigma$ by keeping the $r$ largest singular values and replacing the remaining singular values by zeros.

1.1 Robust PCA

In many real-world problems, if sparse large errors or outliers are present in the data matrix, PCA fails to deal with it. Therefore, it is natural to consider a robust matrix decomposition model in
which we wish to decompose $A$ into the sum of a low-rank matrix $L$ and an error matrix $S$. However, without further assumptions, the problem is ill-posed. We assume the error matrix $S$ is sparse, but allow the entries to have arbitrarily large magnitudes. That is, given $A$, we consider the problem of finding a low rank matrix $L$ and a sparse matrix $S$ such that

$$A = L + S.$$  \hspace{1cm} (3)

In this context, the celebrated principal component pursuit (PCP) formulation of the problem uses the $\ell_0$ norm (cardinality) to address the sparsity constraint and (3). In particular, PCP is the constrained minimization problem [11, 15]

$$\min_{L,S} \text{rank}(L) + \lambda \|S\|_{\ell_0} \quad \text{subject to} \quad A = L + S,$$  \hspace{1cm} (4)

where $\lambda > 0$ is a balancing parameter. Since both $\text{rank}(L)$ and $\|S\|_{\ell_0}$ are non-convex, one often replaces the rank function by the (convex) nuclear norm and $\ell_0$ by the (convex) $\ell_1$ norm. This leads to the immensely popular robust principal component analysis (RPCA) [53, 38, 11], which can be seen as a convex relaxation of (4):

$$\min_{L,S} \|L\|_* + \lambda \|S\|_{\ell_1} \quad \text{subject to} \quad A = L + S,$$  \hspace{1cm} (5)

where $\|\cdot\|_*$ denotes the nuclear norm (sum of the singular values) of matrices. Under some reasonable assumptions on the low-rank and sparse components, [15, 11] showed that (4) can be provably solved via (5). A vast literature is dedicated to solving the RPCA problem, including an inexact augmented Lagrangian method of multipliers [38], accelerated proximal gradient method [53], alternating direction method [56], alternating projection with intermediate denoising [44], dual approach [39], and SpaRCS [52]. Recently, Yi et al. [55], Zhang and Yang [57] proposed a manifold optimization to RPCA. We refer the reader to [6] for a comprehensive review of the RPCA algorithms.

1.2 Robust matrix completion

Besides formulation (5), other tractable reformulations of (4) were proposed in the literature. For instance, by relaxing the equality constraint in (4) and moving it to the objective as a penalty, and adding explicit constraints on the target rank $r$ and target sparsity $s$, we get the formulation

$$\min_{L,S} \|A - L - S\|_F^2 \quad \text{subject to} \quad \text{rank}(L) \leq r \quad \text{and} \quad \|S\|_0 \leq s.$$  \hspace{1cm} (6)

One can extend the above model to the case of partially observed data, which leads to the robust matrix completion (RMC) problem [17, 50, 16]:

$$\min_{L,S} \|\mathcal{P}_\Omega(A - L - S)\|_F^2 \quad \text{subject to} \quad \text{rank}(L) \leq r \quad \text{and} \quad \|\mathcal{P}_\Omega(S)\|_0 \leq s'.$$  \hspace{1cm} (7)

Above, $\Omega \subseteq [m] \times [n]$ is the set of observed data entries, and $\mathcal{P}_\Omega$ is the restriction operator defined by

$$(\mathcal{P}_\Omega[X])_{ij} = \begin{cases} X_{ij} & (i,j) \in \Omega \\ 0 & \text{otherwise}. \end{cases}$$

One can also think of the matrix completion (MC) problem as a special case of (7) [12, 40, 30, 10, 29, 13, 33, 14]. For MC problems, $S = 0$. Therefore, (7) is a generalization of two fundamental problems: RPCA and MC. Further recent work on RMC problem and outlier based PCA can be found in [18, 17].
1.3 Contributions

In this paper, we propose a novel approach to solve the RPCA and RMC problems by addressing the original decomposition problem (3) directly, without introducing any optimization objective or surrogate constraints. Instead, we aim to find a point at the intersection of three sets, two of which are non-convex. We formulate both RPCA and RMC as set feasibility problems and propose alternating projection algorithm to solve them. This leads to Algorithm 2 and 3. Our approach is described in Section 2.

In comparison with existing models, we completely dispense the need for hard to interpret parameters (such as $\lambda$) and surrogate functions (such as the nuclear norm, or $\ell_1$ norm) in our approach. Instead, we rely on two direct parameters: the target rank $r$ and the desired level of sparsity $s$. By performing extensive numerical experiments on both synthetic and real datasets, we show that our approach can match or outperform state-of-the-art methods to solve the RPCA and RMC problems. More precisely, when the sparsity level is low, our feasibility approach can provide a viable reconstruction for any target low rank, which the RPCA algorithms can not. Moreover, our approach can tolerate denser outliers compared to the state-of-the-art RPCA algorithms when the original matrix has a low-rank structure (see details in Section 4). These attributes make our approach attractive to solve many real-world problems where we match or outperform state-of-the-art RPCA algorithms in solution quality, and do this in comparable or less time.

2 Nonconvex Feasibility and Alternating Projections

Set feasibility problem aims to find a point in the intersection of a collection of closed sets, that is:

$$\text{Find } x \in \mathcal{X} \text{ where } \mathcal{X} \overset{\text{def}}{=} \bigcap_{i=1}^{m} \mathcal{X}_i \neq \emptyset,$$

for closed sets $\mathcal{X}_i$. Usually, sets $\mathcal{X}_1, \ldots, \mathcal{X}_m$ are assumed to be simple and easy to project on. A special case of the above setting for convex sets $\mathcal{X}_i$ – convex feasibility problem – is already well studied. In particular, a very efficient convex feasibility algorithm is known as the alternating projection algorithm [32, 3], in which each iteration picks one set $\mathcal{X}_i$ and projects the current iterate on it. There are two main approaches on how sets $\mathcal{X}_i$ might be chosen – traditional cyclic method and randomized method [49, 25, 43], where the later, in general, is faster and not vulnerable to adversarial set order.

We also note that the alternating projection algorithm for convex feasibility problem does not converge in general to the projection of the starting point onto $\mathcal{X}$, but rather finds a close-to feasible point in $\mathcal{X}$, except the case when the sets $\mathcal{X}_i$ are affine spaces. However, once an exact projection onto $\mathcal{X}$ is desired, Dykstra’s algorithm [7] should be applied.

On the other hand, for general nonconvex sets $\mathcal{X}_i$, projection algorithms might not converge. In some special settings, some forms of convergence (for example, local convergence) can be guaranteed even without convexity [36, 35, 28, 19, 45]. In this paper we do not aim to study theoretical convergence properties of our method; we instead investigate convergence from a practical perspective.
Algorithm 1: Alternating projection method for set feasibility

1. **Input**: \( \Pi_i(\cdot) \) – Projector onto \( X_i \) for each \( i \in \{1, \ldots, m\} \), starting point \( x_0 \)
2. for \( k = 0, 1, \ldots \) do
   3. Choose via some rule \( i \) (e.g., cyclically or randomly)
   4. \( x_{k+1} = \Pi_i(x_k) \)
end
5. **Output**: \( x_{k+1} \)

2.1 Set feasibility for RPCA

In this scope, we define \( \alpha \)-sparsity as it appears in the last convex constraint \( X_3 \). We do it so that we are directly comparable to the approaches from [55, 57]. However, we note that the \( \ell_0 \)-ball constraint can be applied as well.

**Definition 2.1** (\( \alpha \)-sparsity). A matrix \( S \in \mathbb{R}^{m \times n} \) is said to be \( \alpha \)-sparse if each row and column of \( S \) contains at most \( \alpha n \) and \( \alpha m \) nonzero entries, respectively. That is, the cardinality of the support set of each row and column of the matrix \( S \) do not exceed \( \alpha n \) and \( \alpha m \), respectively. Formally, we can write
\[
\| S_{(i,\cdot)} \|_0 \leq \alpha n \quad \text{and} \quad \| S_{(\cdot,j)} \|_0 \leq \alpha m \quad \text{for all} \quad i \in [m], j \in [n],
\]
where the \( i^{\text{th}} \) row and \( j^{\text{th}} \) column of \( S \) are \( S_{(i,\cdot)} \) and \( S_{(\cdot,j)} \), respectively.

Now we consider the following reformulation of RPCA:

\[
\text{Find} \quad M \overset{\text{def}}{=} [L, S] \in \mathcal{X} \overset{\text{def}}{=} \bigcap_{i=1}^{3} \mathcal{X}_i \neq \emptyset, \quad (9)
\]

where

\[
\begin{align*}
\mathcal{X}_1 & \quad \overset{\text{def}}{=} \{ M \mid L + S = A \} \quad (10) \\
\mathcal{X}_2 & \quad \overset{\text{def}}{=} \{ M \mid \text{rank}(L) \leq r \} \\
\mathcal{X}_3 & \quad \overset{\text{def}}{=} \{ M \mid \| S_{(i,\cdot)} \|_0 \leq \alpha n \quad \text{and} \quad \| S_{(\cdot,j)} \|_0 \leq \alpha m \quad \text{for all} \quad i \in [m], j \in [n]. \}
\end{align*}
\]

Clearly, \( \mathcal{X}_1 \) is convex, however, \( \mathcal{X}_2 \) and \( \mathcal{X}_3 \) are not. Nevertheless, the algorithm we propose – alternating Frobenius norm projection on \( \mathcal{X}_i \) – performs well for solving RPCA in practice. Due to the non-convex nature of the problem, we do not provide any theoretical guarantees. However, we compare our method to other state-of-the-art RPCA approaches on various practical problems in Section 4. We also study the empirical convergence properties and show that despite the non-convex nature of the problem, the algorithms we propose often behave surprisingly well.

In Appendix A we also empirically show that:

1. Convergence speed is not significantly influenced by starting point.
2. Convergence is usually fastest for small true sparsity level \( \alpha \) and small true rank \( r \), which is the situation with many practical applications.
3. Convergence of Algorithm 3 is slow for medium sized number of observable entries, that is, when $|\Omega| \approx 0.5(m \cdot n)$, and fast for smaller and bigger sizes.

4. If sparsity and rank levels ($\alpha$ and $r$) are set to be smaller than their true values at the optimum incorrectly, Algorithm 2 does not converge (as in this case, $\cap \mathcal{X}_i$ might not exist). Moreover, the performance of the algorithm is sensitive to the choice of $r$, particularly so if we underestimate the true value (see Figure 13).

Finally, in Appendix C we give two examples for the convex version of the problem (9) with the same block structure, where the alternating projection algorithm either converges extremely fast or does not even converge linearly.

3 Alternating Projection Methods for RPCA and RMC

Denote $\Pi_i$ to be projector onto $\mathcal{X}_i$. Note that $\Pi_2$ does not include $S$ and projection onto $\Pi_3$ does not include $L$. Consequently, $\Pi_2 \Pi_3$ is a projector onto $\mathcal{X}_2 \cap \mathcal{X}_3$. Since we now want only to find a point at the intersection of two sets, we shall employ a cyclic projection method (note that randomized method does not make sense). Indeed, steps 4 and 5 of Algorithm 2 perform projection onto $\mathcal{X}_1$, step 6 performs projection onto $\mathcal{X}_2$, and finally, step 7 performs projection onto $\mathcal{X}_3$. In Sections 3.1, 3.2, and 3.3 we describe the exact implementation and prove correctness of the above mentioned steps.

Algorithm 2: Alternating projection method for RPCA

1. **Input**: $A \in \mathbb{R}^{m \times n}$ (the given matrix), rank $r$, sparsity level $\alpha \in (0, 1]$  

2. **Initialize**: $L_0, S_0$

3. **for** $k = 0, 1, \ldots$ **do**

4. $\tilde{L} = \frac{1}{2}(L_k - S_k + A)$

5. $\tilde{S} = \frac{1}{2}(S_k - L_k + A)$

6. $L_{k+1} = H_r(\tilde{L})$

7. $S_{k+1} = T_\alpha(\tilde{S})$

**end**

8. **Output**: $L_{k+1}, S_{k+1}$

Next, we propose an algorithm to solve the RMC problem (7). Note that we use the generic hard thresholding operator as in (2) in step 6 of Algorithm 3. In practice, however, one can perform cheap SVD.

3.1 Projection on the linear constraint

The next lemma provides an explicit formula for the projection onto $\mathcal{X}_1$, which corresponds to steps 4 and 5 of Algorithm 2.

**Lemma 3.1.** Solutions to

$$
\min_{L,S} \|L - L_0\|_F^2 + \|S - S_0\|_F^2 \text{ subject to } L + S = A
$$

is $L^* = \frac{1}{2}(L_0 - S_0 + A)$ and $S^* = \frac{1}{2}(S_0 - L_0 + A)$. 

5
Algorithm 3: Alternating projection method for RMC

1. **Input**: $A \in \mathbb{R}^{m \times n}$ (the given matrix), rank $r$, sparsity level $\alpha \in (0, 1]$
2. **Initialize**: $L_0, S_0$
3. **for** $k = 0, 1, \ldots$ **do**
   4. $\tilde{L} = \frac{1}{2} P_{\Omega}(L_k - S_k + A)$
   5. $\tilde{S} = \frac{1}{2} P_{\Omega}(S_k - L_k + A)$
   6. $L_{k+1} = H_r(\tilde{L} + P_{\Omega}c(L_k))$
   7. $S_{k+1} = T_\alpha(\tilde{S})$
4. **end**
8. **Output**: $L_{k+1}, S_{k+1}$

We also provide an analogy to Lemma 3.1 for the RMC problem (steps 4 and 5 of Algorithm 3).

**Lemma 3.2.** Solutions to

$$
\min_{L,S} \| P_{\Omega}(L - L_0) \|_F^2 + \| P_{\Omega}(S - S_0) \|_F^2 \quad \text{subject to} \quad P_{\Omega}(L + S) = P_{\Omega}(A)
$$

are $L^* = \frac{1}{2} P_{\Omega}(L_0 - S_0 + A)$ and $S^* = \frac{1}{2} P_{\Omega}(S_0 - L_0 + A)$.

### 3.2 Projection on the low rank constraint

Consider $L^{(r)}$ to be the projection of $L$ onto the rank $r$ constraint, that is,

$$
L^{(r)} = \arg \min_{L'} \| L' - L \|_F \quad \text{subject to} \quad \text{rank}(L') \leq r.
$$

It is known that $L^{(r)}$ can be computed as $r$-SVD of $L$. There have been a great improvement on fast $r$-SVD solvers in recent years [26, 42, 47, 1]. Unfortunately, in our setting we are not able to apply the most recent approaches [47, 1], which are inefficient since they need to compute $LL^\top$ (or $L^\top L$), which is expensive. Instead, we use block Krylov approach from [42] (for completeness, we include a pseudocode of the algorithm in Appendix D). Regarding the computational complexity, it was shown that block Krylov SVD outputs $Z$ satisfying $\| L - ZZ^\top L \|_F \leq (1 + \epsilon) \| L - L^{(r)} \|_F$ in

$$
\mathcal{O}\left(\| L \|_0 \frac{r \log n}{\sqrt{\epsilon}} + \frac{mr^2 \log^2 n}{\epsilon} + \frac{r^3 \log^3 n}{\epsilon^{3/2}}\right)
$$

flops. Therefore, projection on the low-rank constraint is not an issue for relatively small rank $r$.

### 3.3 Projection on the sparsity constraint

Projection onto $X_3$ simply keeps the $\alpha$-fraction of the largest elements in absolute value in each row and column and set the rest to zero. Formally, the following operator projects onto $X_3$:

$$
T_\alpha[S] \overset{\text{def}}{=} \{ P_{\Omega_\alpha}(S) \in \mathbb{R}^{m \times n} : (i,j) \in \Omega_\alpha \text{ if } |S_{ij}| \geq |S_{(i\cdot)}^{(\alpha)}| \text{ and } |S_{ij}| \geq |S_{(\cdot j)}^{(\alpha)}|\}, \quad (11)
$$

where $S_{(i\cdot)}^{(\alpha)}$ and $S_{(\cdot j)}^{(\alpha)}$ denote the $\alpha$ fraction of largest entries of $S$ along the $i$th row and $j$th column, respectively. We note that the operator $T_\alpha(\cdot)$ is similar to that defined in [55, 57]. Projection on
sparsity constraint (11) can be implemented in $O(nd)$ time: for each row and each column we find $\alpha n$-th largest element (or $\alpha d$) and simultaneously (for other rows/columns) mask the rest. In our experiments, we use fast implementation of $n$-th element computation from [37].

4 Numerical Experiments

In this section we perform numerical experiments to explore the strengths and flexibility of our feasibility approach. First, we work with synthetic data and subsequently apply our method to four real-world problems.

4.1 Results on synthetic data

To perform our numerical simulations, first, we construct the test matrix $A$. We follow the seminal work of Wright et al. [53] to design our experiment. To this end, we construct $A$ as a low-rank matrix, $L$, corrupted by sparse large noise, $S$, with arbitrary large entries such that $A = L + S$. We generate $L$ as a product of two independent full-rank matrices of size $m \times r$ whose elements are independent and identically distributed (i.i.d.) $\mathcal{N}(0, 1)$ random variables and $\text{rank}(L) = r$. We generate $S$ as a noise matrix whose elements are sparsely supported by using the operator (11) and lie in the range $[-500, 500]$. We fix $m = 200$ and define $\rho_r = \text{rank}(L)/m$ where $\text{rank}(L)$ varies. We choose the sparsity level $\alpha \in (0, 1)$. For each pair of $(\rho_r, \alpha)$ we apply iEALM, APG, and our algorithm to recover the pair $(\hat{L}, \hat{S})$ such that $\hat{A} = \hat{L} + \hat{S}$ be the recovered matrix. For both APG and iEALM, we set $\lambda = 1/\sqrt{m}$ and for iEALM we use $\mu = 1.25/\|A\|_2$ and $\rho = 1.5$, where $\|A\|_2$ is the spectral norm (maximum singular value) of $A$. If the recovered matrix pair $(\hat{L}, \hat{S})$ satisfies the relative error $\|\hat{L} - L\|_F + \|\hat{S} - S\|_F / \|A\|_F < 0.01$ then we consider the construction is viable. In Figure 1 we show the fraction of perfect recovery, where white denotes success and black denotes failure. As mentioned in [53], the success of APG is approximately below the line $\rho_r + \alpha = 0.35$. However, the success of iEALM is not as good as APG. To conclude, when the sparsity level $\alpha$ is low, our feasibility approach can provide a feasible reconstruction for any $\rho_r$. We note that for low sparsity level, the RPCA algorithms can only provide a feasible reconstruction for $\rho_r \leq 0.4$. On the other hand, for low $\rho_r$, our feasibility approach can tolerate sparsity level approximately up to 63%. In contrast, RPCA algorithms can tolerate sparsity up to 50% for low $\rho_r$. Therefore, taken together,
Figure 2: Phase transition diagram for Relative error for RMC problems: (a) $|\Omega_C| = 0.5(m.n)$, (b) $|\Omega_C| = 0.75(m.n)$, (c) $|\Omega_C| = 0.9(m.n)$. Here, $\rho_r = \text{rank}(L)/m$ and $\alpha$ is the sparsity measure. We have $(\rho_r, \alpha) \in (0.025, 1] \times (0, 1)$ with $r = 5 : 25 : 200$ and $\alpha = \text{linspace}(0, 0.99, 8)$.

Figure 3: Background and foreground separation on Stuttgart dataset Basic video. Except RPCA GD and our method, all other methods fail to remove the static foreground object.

we can argue that our method can be proved useful to solve real-world problems when one wants to recover a moderately sparse matrix having any inherent low-rank structure present in it or in case of a low-rank matrix corrupted with dense outliers of arbitrary large magnitudes.

4.2 Results on synthetic data: RMC problem

For experiments in this section, we used a similar technique as in Section 4.1 to generate the test matrix $A$. We fixed $m = 200$ and denote $\rho_r$ and $\alpha$ same as in Section 4.1. We randomly select the set of observable entries in $A$. We compare our method against the RPCA gradient descent (RPCA GD) by Yi et al. [55] and use the relative error for the low-rank component recovered as performance measure, that is, if $\|L - \hat{L}\|_F / \|L\|_F < \tilde{\epsilon}$ then we consider the construction is viable. Note that $L$ is the original low-rank matrix and $\hat{L}$ is the low-rank matrix recovered. For $|\Omega_C| = 0.5(m.n), 0.75(m.n),$ and $0.9(m.n)$ we consider $\tilde{\epsilon} = 0.2, 0.6,$ and $1$, respectively. In Figure 2, for the phase transition diagram white denotes success and black denotes failure. From Figure 2 we observe that irrespective of the cardinalities of the set of the observed entries our feasibility approach outperforms RPCA GD. However, as the cardinality of the set of the observable entries, that is, $|\Omega|$ decreases, the performance of our feasibility approach gets better (see Figure 2c).

Next, we use the root mean square error (RMSE), that is, $\|L - \hat{L}\|_F / \sqrt{mn}$ as a performance measure for these set of results. Note that $L$ is the original low-rank matrix and $\hat{L}$ is the low-rank matrix recovered. From Figure 4 we observe that when the cardinality of the set of the observable entries $\Omega$ is 50% and 75% of $[m] \times [n]$, respectively, RPCA GD has slightly better RMSE than our method as $\rho_r$ increases. However, as the cardinality of the set of the observable entries, that is, $|\Omega|$ decreases, we outperform RPCA GD (see Figure 4c-4d). Therefore, we further validate that for
Figure 4: RMSE for RMC problems: (a) $|\Omega_C| = 0.5(m.n)$, (b) $|\Omega_C| = 0.75(m.n)$, (c) $|\Omega_C| = 0.9(m.n)$, (d) $|\Omega_C| = 0.95(m.n)$. Here, $\rho_r = \text{rank}(L)/m$ and $\alpha$ is the sparsity measure. We have $(\rho_r, \alpha) \in (0.025, 1) \times (0, 1)$ with $r = 5 : 25 : 200$ and $\alpha = \text{linspace}(0, 0.99, 8)$.

Figure 5: Background and foreground separation on Stuttgart dataset Basic video. We used 90% sample. GRASTA forms a fragmentary background and exhausts around 540 frames to form a stable video. We also note that RPCA GD has more false positives in the foreground.
Figure 6: (a) Comparison of relative error vs. iteration between RPCA F, iEALM, APG, and RPCA GD on Basic video, frame size 144 × 176. iEALM takes 55.41 seconds, RPCA GD takes 36.08 seconds (30 iterations), APG takes 51.14 seconds, and RPCA CF takes 42.72 seconds (30 iterations). The threshold $\epsilon$ for all algorithms is set to $2 \times 10^{-4}$. (b) Comparison of relative error (log scale) vs. iteration between RPCA F, iEALM, APG, and RPCA GD on Shadow removal, Yale Extended Face dataset, subject B12. iEALM takes 2.05 seconds (threshold $10^{-7}$), RPCA GD takes 4.73 seconds (30 iterations, threshold $2 \times 10^{-4}$), APG takes 10.9 seconds (threshold $10^{-7}$), and RPCA CF takes 4.71 seconds (30 iterations, threshold $2 \times 10^{-4}$). For APG and iEALM we plot every fifth iteration.

RMC problems, when $|\Omega|$ is small the feasibility approach is better to recover a low-rank matrix.

4.3 Applications to real-world problem

In this section we demonstrate the robustness of our feasibility approach to solve four classic real-world problems: i) background and foreground estimation from fully and partially observed data, ii) shadow removal from face images captured under varying illumination and camera position, iii) inlier subspace detection, iv) processing astronomical data.

4.3.1 Background and foreground estimation from fully observed data

In this section, we show our results on the background estimation problem. In the past decade, one of the most prevalent approaches used to solve background estimation problem is to treat it as a low-rank and sparse matrix decomposition problem [4, 5, 48, 21, 41, 51, 27, 54, 20]. Given a sequence of $n$ video frames with each frame mapped into a vector $a_i \in \mathbb{R}^m$, $i = 1, 2, ..., n$, the data matrix $A \in \mathbb{R}^{m \times n}$ in the collection of all the frame vectors is expected to be split into $L + S$. By using the above idea, RPCA [11, 38, 53] was introduced by considering the background frames, $L$, having a low-rank structure and the foreground, $S$, sparse. The convex relaxation of the problem is (5).

For simulations, we used the Basic sequence of the Stuttgart artificial dataset [8]. We compare our methods against inexact augmented Lagrange methods of multiplier (iEALM) of Lin et al. [38], accelerated proximal gradient (APG) of Wright et al. [53], and RPCA GD. We downsampled the video frames to a resolution of 144 × 176 and for iEALM we use $\mu = 1.25/\|A\|_2$ and $\rho = 1.5$. For
both APG and iEALM we set $\lambda = 1/\sqrt{\max\{m, n\}}$. For RPCA GD and our method we use target rank $r = 2$, sparsity $\alpha = 0.1$. The threshold $\epsilon$ for all methods are kept to $2 \times 10^{-4}$. The qualitative analysis on the background and foreground recovered on the sample frame of the Basic sequence in Figure 3 suggests that our method and RPC GD recover a visually better quality background and foreground compared with the other methods. We also note that RPCA GD recovers a foreground with more false positives compared to our method and iEALM and APG cannot remove the static foreground object.

**4.3.2 Background and foreground estimation from partially observed data**

We randomly select the set of observable entries in the data matrix $A$ and tested our algorithm against Grassmannian Robust Adaptive Subspace Tracking Algorithm (GRASTA) \[27\] and RPCA GD. In Figure 5, we demonstrate the performance on the Basic sequence of the Stuttgart dataset with $|\Omega| = 0.9(m.n)$. The parameters for our algorithm and RPCA GD are set as same as in Section 4.3.1. For GRATSA we set the parameters same as mentioned in the authors’ website\[1\]. Next, in Figure 7 we show the background and foreground separated by different methods on the same sample frame of the Basic sequence of Stuttgart dataset for different subsample rate. It is evident that RPCA GD and our approach has the best background reconstruction. However, when compared with the foreground ground truth our method has a better quantitative measure as RPCA GD has a higher number of false positives in the foreground (see Figure 8).

**Background and foreground estimation from partially observed data: Quantitative measure.** Let $X = (X_1, \ldots, X_n)$ and $y = (Y_1, \ldots, Y_n)$ be two video sequences (reconstructed foreground and ground truth foreground), where $X_i, Y_i \in \mathbb{R}^m$ are vectors corresponding to frame $i$, each containing $m$ pixels. We scale all pixel values to $[0, 1]$. To compare the video sequences, we define an $\epsilon$-proximity measure of $X$ and $Y$ as

$$d^\epsilon(X, Y) \overset{\text{def}}{=} \frac{1}{nm} \sum_{i=1}^{n} \sum_{k=1}^{m} d^\epsilon(X_{ik}, Y_{ik}),$$

\[1\]https://sites.google.com/site/hejunzz/grasta
Figure 8: Quantitative comparison of foreground recovered by RPCA GD and RPCA F on Basic video, frame size 144 × 176 with observable entries: (a) $|\Omega| = 0.9(m.n)$, (b) $|\Omega| = 0.8(m.n)$, (c) $|\Omega| = 0.7(m.n)$, (d) $|\Omega| = 0.6(m.n)$, (e) $|\Omega| = 0.5(m.n)$, and (f) $|\Omega| = 0.4(m.n)$. The performance of RPCA GD drops significantly as $|\Omega|$ decreases. In contrast, the performance of RPCA F stays stable irrespective of the size of $|\Omega|$.

Figure 9: Shadow and specularities removal from face images captured under varying illumination and camera position. Our feasibility approach provides comparable reconstruction to that of iEALM and APG.
where
\[
d^{\epsilon}(u, v) \overset{\text{def}}{=} \begin{cases} 1 \\ 0 \end{cases} \begin{cases} |u - v| \leq \epsilon, \\ \text{otherwise}, \end{cases}
\]
and \( \epsilon \in [0, 1] \) is a threshold. Clearly, \( 0 \leq d^{\epsilon}(X, Y) \leq 1 \), \( \epsilon \mapsto d^{\epsilon}(X, Y) \) is increasing, and \( d^{1}(X, Y) = 1 \). If \( d^{\epsilon}(X, Y) = \alpha \), then \( \alpha \times 100\% \) of pixels in the recovered video are within \( \epsilon \) distance, in absolute value, from the ground truth.

In Figure 8, we plot \( d^{\epsilon}(X, Y) \) as a function of \( \epsilon \) for our method and RPCA GD. We use the Basic sequence of the Stuttgart dataset and vary the cardinality of the set of observable entries \( \Omega \). Our feasibility approach outperforms RPCA GD for all values of \( |\Omega| \) and \( \epsilon \), and the difference is striking; in particular, our method recovers more than 95% pixels correctly even for under accuracy (i.e., small \( \epsilon \)) requirements.

### 4.3.3 Shadow removal

The images of a face exposed to a wide variety of lighting conditions can be approximated accurately by a low-dimensional linear subspace. More specifically, the images under distant, isotropic lighting lie close to a 9-dimensional linear subspace which is known as the harmonic plane [2]. We used the Extended Yale Face Database for our experiments [23]. We used iEALM, APG, and RPCA GD to compare against our algorithm. We downscaled each image to a resolution of 120 \( \times \) 160 and use 63 images of a subject in each test. For APG and iEALM, we set the parameters same as in Section 4.3.1. For RPCA GD and our method, we set target rank \( r = 9 \) and sparsity level \( \alpha = 0.1 \).

The qualitative analysis on the recovered images shows that our feasibility approach provides a comparable reconstruction similar to that of iEALM and APG (see Figure 9). In contrast, the reconstructed face images by RPCA GD are of poor visual quality.

### 4.3.4 Inlier detection

Our next set of experiments demonstrate the power of our method in detecting the inliers and the outliers from a composite dataset. For this purpose, we artificially create a dataset that contains both inliers and outliers. We used the Yale Extended Face Database to construct a data set that contains images of faces under different illuminations. We denote this as inliers. With these inliers, we infused 400 random natural images from the BACKGROUND/Google folder of the Caltech101 database [22] that serve as outliers. Both inlier and outlier images were converted to grayscale and the resolution is downsampled to 20 \( \times \) 20 pixels. For the inliers, we are looking for the 9-dimensional linear subspace where the images of the same face lie. That is, similar to [24] we consider a low-dimensional model to the set of all faces aka inliers. We note that the seven algorithms proposed in [24] are designed to explicitly find a low-rank subspace. In [24] the authors used different objective functions and used SGD, incremental approach, and mirror descent algorithms to find the low-dimensional subspace. However, we approach the problem slightly differently. We split the dataset, \( A \), into a 9-dimensional low-rank subspace \( L \) and expect the specularities and outliers to be in the sparse set \( S \). Once we find \( L \), we extract the basis of \( L \) and project the faces on it. In Figure 10, we show the qualitative results of our experiments\(^2\).

\(^2\)The codes and datasets for experiments in Section 4.3.4 and 4.3.5 are obtained from https://github.com/jwgoes/RSPCA
Figure 10: Inliers and outliers detection. Face images captured in different lighting conditions are inliers. We project different faces to 9 dimensional subspaces found by different methods.

Table 1: Quantitative performance of different algorithms in inlier detection experiment. Except R-SGD2 all methods are highly competitive.

| Metric Used | SGD  | R-SGD1 | R-SGD2 | Inc  | R-Inc | MD   | R-MD | RPCA-F |
|-------------|------|--------|--------|------|-------|------|------|--------|
| $\frac{\|P_L - P_{L^*}\|_F}{3\sqrt{2}}$ | 0.6985 | 0.8603 | 4.6607 | 0.7703 | 0.7214 | 0.6711 | 0.6679 | 0.7764 |

4.3.5 Processing astronomical data

In this experiment, we use the VIMOS Very Large Telescope (VIMOS-VLT) Deep Survey dataset [34] to understand the evolution of the galaxies. We compare the first 4 eigenspectra obtained by our feasibility approach with those of the state-of-the-art methods, such as RE-PCA of [9], online PCA, and robust online PCA of [24]. Similar to Section 4.3.4, we split the dataset, $A$, into a 4-dimensional low-rank subspace $L$ and after we find $L$, we extract the orthogonal basis of $L$ and plot them. From Figure 11, visually, robust online PCA and our method are close relatively best fit to the ground
Figure 11: The top four eigenspectra for the VVDS galaxies. The top three rows are the state-of-the-art algorithms RE-PCA of [9], online PCA, and robust online PCA of [24], respectively. The last row is our feasibility approach.

truth RE-PCA of [9]. For details of the data, motivation, and experimental setup we refer the readers to [34, 9, 24].

4.4 Further experiments

We show many of our experimental results in the Appendix. On synthetic data, we empirically validate the sensitivity of Algorithm 2 with respect to the initialization, the choices of $r$, and sparsity level $\alpha$; and the effect of the cardinality of $\Omega$ for Algorithm 3 (see A.1, A.2, and A.3).

5 Conclusion

In this paper, we propose a simplistic and novel approach to solve the classic RPCA and RMC problems. We consider an alternating projection algorithm based on the set feasibility approach to solve these problems in their crude form, without considering any further heuristics, such as loss functions, convex and surrogate constraints. Although we did not rigorously study convergence of our method theoretically; we investigated the convergence through numerical simulations on synthetic and real-world data and extensively compared with the current state-of-the-art methods. Our feasibility approach can open a new direction of potential research on online algorithms based on RPCA framework [46, 27, 54] that vastly used in video analysis, segmentation, subspace detection, and only a few to mention.
References

[1] Z. A.-Zhu and Y. Li. LazySVD: Even faster SVD decomposition yet without agonizing pain. In Advances in Neural Information Processing Systems, pages 974–982, 2016.

[2] R. Basri and D. Jacobs. Lambertian reflection and linear subspaces. IEEE Transaction on Pattern Analysis and Machine Intelligence, 25(3):218–233, 2003.

[3] H. H. Bauschke and J. M. Borwein. On projection algorithms for solving convex feasibility problems. SIAM review, 38(3):367–426, 1996.

[4] T. Bouwmans. Traditional and recent approaches in background modeling for foreground detection: An overview. Computer Science Review, 11–12:31 – 66, 2014.

[5] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah. Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset. Computer Science Review, 23:1–71, 2017.

[6] T. Bouwmans and E.-H. Zahzah. Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance. Computer Vision and Image Understanding, 122:22–34, 2014.

[7] J. P. Boyle and R. L. Dykstra. A method for finding projections onto the intersection of convex sets in Hilbert spaces. In Advances in order restricted statistical inference, pages 28–47. Springer, 1986.

[8] S. Brutzer, B. Höferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. IEEE Computer Vision and Pattern Recognition, pages 1568–1575, 2012.

[9] T. Budavari, V. Wild, A. S. Szalay, L. Dobos, and C.-W. Yip. Reliable eigenspectra for new generation surveys. Monthly Notices of the Royal Astronomical Society, 394(3):1496–1502, 2009.

[10] J. F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization, 20(4):1956–1982, 2010.

[11] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? Journal of the Association for Computing Machinery, 58(3):11:1–11:37, 2011.

[12] E. J. Candès and Y. Plan. Matrix completion with noise. Proceedings of the IEEE, 98(6):925–936, 2009.

[13] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. Foundations of Computational Mathematics, 9(6):717–772, 2009.

[14] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. IEEE Transactions on Information Theory, 56(5):2053–2080, 2010.

[15] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. SIAM Journal on Optimization, 21(2):572–596, 2011.

[16] Y. Chen, H. Xu, C. Caramanis, and S. Sanghavi. Robust matrix completion and corrupted columns. In Proceedings of the 28th International Conference on International Conference on Machine Learning, pages 873–880, 2011.

[17] Y. Cherapanamjeri, K. Gupta, and P. Jain. Nearly optimal robust matrix completion. In Proceedings of the 34th International Conference on Machine Learning (ICML), pages 797–805, 2017.

[18] Y. Cherapanamjeri, P. Jain, and P. Netrapalli. Thresholding based outlier robust PCA. In Proceedings of the 30th Conference on Learning Theory (COLT), pages 593–628, 2017.

[19] D. Drusvyatskiy, A. D. Ioffe, and A. S. Lewis. Transversality and alternating projections for nonconvex sets. Foundations of Computational Mathematics, 15(6):1637–1651, 2015.
[20] A. Dutta. Weighted Low-Rank Approximation of Matrices: Some Analytical and Numerical Aspects. PhD thesis, University of Central Florida, 2016.

[21] A. Dutta, B. Gong, X. Li, and M. Shah. Weighted singular value thresholding and its application to background estimation, 2017. arXiv:1707.00133.

[22] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. Computer Vision and Image Understanding, 106(1):59–70, 2007.

[23] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Transactions on PAMI, 23(6):643–660, 2001.

[24] J. Goes, T. Zhang, R. Arora, and G. Lerman. Robust stochastic principal component analysis. In Proceedings of the 17th International Conference on Articial Intelligence and Statistics, pages 266–274, 2014.

[25] R. M. Gower and P. Richtárik. Randomized iterative methods for linear systems. SIAM Journal on Matrix Analysis and Applications, 36(4):1660–1690, 2015.

[26] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM review, 53(2):217–288, 2011.

[27] J. He, L. Balzano, and A. Szlam. Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video. IEEE Computer Vision and Pattern Recognition, pages 1937–1944, 2012.

[28] R. Hesse and D. R. Luke. Nonconvex notions of regularity and convergence of fundamental algorithms for feasibility problems. SIAM Journal on Optimization, 23(4):2397–2419, 2013.

[29] P. Jain and P. Netrapalli. Fast exact matrix completion with finite samples. In Proceedings of The 28th Conference on Learning Theory (COLT), pages 1007–1034, 2015.

[30] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing, pages 665–674, 2013.

[31] I. T. Jolliffe. Principal component analysis, 2002. Second edition.

[32] S. Kaczmarz. Angenaherte auflosung von systemen linearer gleichungen. Bulletin International de l’Académie Polonaise des Sciences et des Lettres, A, 35:355–357, 1937.

[33] R. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. IEEE Transactions on Information Theory, 56(6):2980–2998, 2010.

[34] O. L. Fèvre, G. Vettolani, B. Garilli, L. Tresse, D. B. V. L. Brun, D. Maccagni, J. P. Picat, R. Scaramella, M. Scodeggio, et al. The VIMOS VLT deep survey-First epoch VVDS-deep survey: 11564 spectra with 17 : 5 < I(AB) < 24, and the redshift distribution over 0 < z < 5. Astronomy & Astrophysics, 439(3):845–862, 2005.

[35] A. S. Lewis, R. Luke, and J. Malick. Local linear convergence for alternating and averaged non-convex projections. Foundations of Computational Mathematics, 9(4):485–513, 2009.

[36] A. S. Lewis and J. Malick. Alternating projections on manifolds. Mathematics of Operations Research, 33(1):216–234, 2008.

[37] P. Li. Nth element. https://www.mathworks.com/matlabcentral/fileexchange/29453-nth-element, 2013.

[38] Z. Lin, M. Chen, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices, 2010. arXiv1009.5055.
[39] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Yi Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. *UIUC Technical Report UILU-ENG-09-2214*, 2009.

[40] J. Mareček, P. Richtárik, and M. Takáč. Matrix completion under interval uncertainty. *European Journal of Operational Research*, 256(1):35 – 43, 2017.

[41] G. Mateos and G. Giannakis. Robust PCA as bilinear decomposition with outlier-sparsity regularization. *IEEE Transaction on Signal Processing*, 60(10):5176–5190, 2012.

[42] C. Musco and C. Musco. Randomized block Krylov methods for stronger and faster approximate singular value decomposition. In *Advances in Neural Information Processing Systems*, pages 1396–1404, 2015.

[43] I. Necoara, P. Richtárik, and A. Patrascu. Randomized projection methods for convex feasibility problems: conditioning and convergence rates. *arXiv preprint arXiv:1801.04873*, 2018.

[44] P. Netrapalli, U. N. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain. Non-convex robust PCA. In *Advances in Neural Information Processing Systems 27*, pages 1107–1115. 2014.

[45] C. H. J. Pang. Nonconvex set intersection problems: From projection methods to the newton method for super-regular sets. *arXiv:1506.08246*, 2015.

[46] P. Rodriguez and B. Wohlberg. Incremental principal component pursuit for video background modeling. *Journal of Mathematical Imaging and Vision*, 55(1):1–18, 2016.

[47] O. Shamir. A stochastic PCA and SVD algorithm with an exponential convergence rate. In *International Conference on Machine Learning*, pages 144–152, 2015.

[48] A. Sobral and A. Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122:4 – 21, 2014.

[49] T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262, 2009.

[50] M. Tao and J. Yang. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization*, 21(1):57–81, 2011.

[51] N. Wang, T. Yao, J. Wang, and D.-Y. Yeung. A probabilistic approach to robust matrix factorization. In *Proceedings of 12th European Conference on Computer Vision*, pages 126–139, 2012.

[52] A. E. Waters, A. C. Sankaranarayanan, and R. Baraniuk. SpaRCS: Recovering low-rank and sparse matrices from compressive measurements. *Proceedings of 24nd Advances in Neural Information Processing systems*, pages 1089–1097, 2011.

[53] J. Wright, Y. Peng, Y. Ma, A. Ganesh, and S. Rao. Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization. *Proceedings of 22nd Advances in Neural Information Processing systems*, pages 2080–2088, 2009.

[54] J. Xu, V. K. Ithapu, L. Mukherjee, J. M. Rehg, and V. Singh. Gosus: Grassmannian online subspace updates with structured sparsity. In *Proceedings of IEEE International Conference on Computer Vision*, pages 3376–3383, 2013.

[55] X. Yi, D. Park, Y. Chen, and C. Caramanis. Fast algorithms for robust PCA via gradient descent. *Advances in Neural Information Processing systems*, pages 361–369, 2016.

[56] X. Yuan and J. Yang. Sparse and low-rank matrix decomposition via alternating direction methods. *Pacific Journal of Optimization*, 9(1):167–180, 2013.

[57] T. Zhang and Y. Yang. Robust PCA by manifold optimization, 2017. *arXiv:1708.00257v3*.  

18
A Additional Numerical Experiments

In this section we empirically study the convergence of Algorithm 2.

A.1 Algorithm 2: Sensitivity to initialization

First, we examine how the starting point influences the convergence. We construct $A \in \mathbb{R}^{100 \times 100}$ and perform 50 runs of Algorithm 2 for various values of $\alpha$ and $r$. In all cases, we set

$$A \overset{\text{def}}{=} T_\alpha(S') + H_r(L')$$

for $S', L'$ with independent random entries from $\mathcal{N}(0,1)$, and run Algorithm 2 with (correct) parameters $\alpha, r$. Figure 12 shows the worst, the best, and the median case for each iteration, and illustrates that the convergence (and convergence speed) of the algorithm for the vast majority of cases is independent of the initial point. Moreover, when both rank and sparsity are not too big (sparsity level is 10% or less and rank is 15% or less), we observe very fast convergence.

![Figure 12: Sensivity of Algorithm 2 to initialization. The best, the worst, and the median case are plotted for each iteration.](image)
A.2 Algorithm 3: The effect of the number of observable entries on convergence

In this section, we study convergence properties of Algorithm 3. For different choices of $\alpha$ and $r$, Figure 13 shows how fast does Algorithm 3 converge to the optimum. We observe extremely fast convergence for both small ($< 0.1$) and large ($\approx 1$) fraction of observable entries. However, for medium fractions of observable entries, Algorithm 3 seems to often do not converge. This is an interesting phenomenon that could be studied more deeply in future research. However, for example, as Figure 8 shows, we demonstrate that Algorithm 3 still outperforms the other methods even for the critical medium sized $\Omega$.

A.3 Algorithm 2: Sensitivity to the choice of $\alpha$ and $r$

In this section we study the sensitivity of Algorithm 2 to the degree at which we choose the rank and sparsity level parameters correctly, compare with their true values at the optimum. We first generate a matrix $A$ as described in Section A.1 for a fixed choice of $\hat{\alpha}, \hat{r}$. We then run Algorithm 2 various choices of $\alpha, r$, including the correct choice. Figure 14 shows the results. If sparsity and rank
levels ($\alpha$ and $r$) are set to be smaller than their true values at the optimum incorrectly, Algorithm 2 does not converge (as in this case, $\mathcal{X}_c$ might not exist). Moreover, the performance of the algorithm is sensitive to the choice of $r$, particularly so if we underestimate the true value (see Figure 13). However, overestimating the parameters only leads to a slower convergence.

## B Proof of Lemma 3.1 and Lemma 3.2

We start with proof of Lemma 3.1. Note that for $I \in \mathbb{R}^{mn \times mn}$, $L + S = A$ can be rewritten as

$$(I \ I) \begin{pmatrix} L \\ S \end{pmatrix} = A.$$  

Define $x \overset{\text{def}}{=} \text{vec} \left( \begin{pmatrix} L \\ S \end{pmatrix} \right)$ and $\text{vec}(A) = a$. Therefore, the above is equivalent to

$$(I \otimes (I \ I)) x = a,$$

which is just a projection on a particular linear system. Recall that the projection of $x_0$ in the Frobenius norm (for the vectors it is equivalent to the $\ell_2$ norm) onto $Ax = b$ is given as $x_0 - A^\top (AA^\top)^\dagger (Ax - b)$. Therefore,

$$x = x_0 - (I \otimes (I \ I))^\top \left( (I \otimes (I \ I)) (I \otimes (I \ I))^\top \right)^\dagger \left( (I \otimes (I \ I)) x - a \right)$$

$$= x_0 - \frac{1}{2} (I \otimes (I \ I))^\top (I \otimes I) \left( (I \otimes (I \ I)) x - a \right)$$

$$= x_0 - \frac{1}{2} \left( I \otimes \begin{pmatrix} I & I \\ I & I \end{pmatrix} \right) x + \frac{1}{2} (I \otimes (I \ I))^\top a$$

$$= \text{vec} \left( \begin{pmatrix} L_0 \\ S_0 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} L_0 + S_0 \\ L_0 + S_0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} A \\ A \end{pmatrix} \right),$$

which is equivalent to $L^* = \frac{1}{2}(L_0 - S_0 + A)$ and $S^* = \frac{1}{2}(S_0 - L_0 + A)$. Therefore, Lemma 3.1 is established.

To get Lemma 3.2, it remains to note that the problem is coordinate wise separable. Therefore, the solution behaves as in Lemma 3.1. on set $\Omega$, otherwise the coordinates of $L, S$ remain unchanged.

## C Two Examples of Convergence

In this section we give two examples of a convex version of the alternating projection method on a problem with similar (block) structure as (9). The first example shows that the convergence might be extremely fast and independent on $A$, and the second one demonstrates that the rate might not be linear even under convexity.

**Example 1.** Consider problem (9) with $\mathcal{X}_1$ defined as (10) and both $\mathcal{X}_2, \mathcal{X}_3$ satisfy the same linear constraint.

**Lemma C.1.** Alternating projection algorithm applied on Example 1 converges in 1 iteration.
Figure 14: Convergence of Algorithm 3 for the different choices of \( \omega \). Each line corresponds to a random percentage of observable entries and shows normalized \( \ell_2 \) norm of \((A - L_k - S_k)\omega\).
Proof. For simplicity, let us vectorize \( L, S \): \( x \overset{\text{def}}{=} \text{vec}(L), y \overset{\text{def}}{=} \text{vec}(S) \) and denote \( I \) to be \( mn \times mn \) identity matrix. Since the constraints are linear, the alternating projection algorithm applied on them converges as fast as alternating projection applied on any affine translation of them such that nonempty intersection property holds. Let us therefore, without loss of generality consider the following linearly translated problem for some matrix \( Q \):

\[
X_1 \overset{\text{def}}{=} \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \mid (I \ I) \begin{pmatrix} x \\ y \end{pmatrix} = 0 \right\}
\]

\[
X_2 \overset{\text{def}}{=} \left\{ \begin{pmatrix} x \\ 0 \end{pmatrix} \mid Qx = 0 \right\}
\]

\[
X_3 \overset{\text{def}}{=} \left\{ \begin{pmatrix} 0 \\ y \end{pmatrix} \mid Qy = 0 \right\}
\]

Therefore we have for some projection matrix \( P = P(Q) \)

\[
\pi_{X_2 \cap X_3} \left( \pi_{X_1} \begin{pmatrix} x \\ y \end{pmatrix} \right) = \underbrace{\begin{pmatrix} P & 0 \\ 0 & P \end{pmatrix}}_{R} \begin{pmatrix} \frac{1}{2}I & -\frac{1}{2}I \\ \frac{1}{2}I & \frac{1}{2}I \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}
\]

and the convergence of the algorithm is determined by the maximal eigenvalue of \( R \) which is not 0 or 1 in the absolute value. Clearly, vectors of type

\[
\begin{pmatrix} p_1 \\ p_1 \end{pmatrix}, \begin{pmatrix} p_2 \\ -p_2 \end{pmatrix}, \begin{pmatrix} p_1' \\ p_1' \end{pmatrix}, \begin{pmatrix} p_2' \\ -p_2' \end{pmatrix},
\]

might form an orthonormal basis of the space for \( p_1, p_2 \in \text{Range}(P) \) and \( p_1', p_2' \perp \text{Range}(P) \). However, each of them is an eigenvector of \( R \) with eigenvalue 0 or 1, which finishes the proof.

We will now present an example where linear convergence rate cannot be attained.

**Example 2.** Consider problem (9) for \( A \in \mathbb{R}^2 \) with \( X_1 \) defined as (10) and both \( X_2, X_3 \) are unit balls.

The next lemma shows that there exist a problem of structure (9), for which alternating projection algorithm does not attain a linear convergence rate.

**Lemma C.2.** Suppose that \( X \) is nonempty. There exists a starting point such that for Example 2, alternating projection algorithm does not converge linearly.

**Proof.** Choose

\[
A = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \quad L_0 = \begin{pmatrix} \sqrt{2} \\ \sqrt{2} \end{pmatrix}, \quad \text{and} \quad S_0 = \begin{pmatrix} \sqrt{2} \\ -\sqrt{2} \end{pmatrix}.
\]

Clearly, in optimum we must have \( L^* = e_1, S^* = e_1 \). It is a simple exercise to notice that \( L, S \) are projected each iteration onto line \( x = 1 \) and then back to the unit circle. Therefore, alternating projection onto (9) converges as fast as alternating projection onto unit ball and its tangent line. However, it is easy to see that the latter algorithm does not enjoy a linear convergence.
Algorithm 4: Block Krylov SVD [42] (BKSVD)

Input : $L \in \mathbb{R}^{m \times n}$, tolerance $\tilde{\epsilon} \in (0, 1)$, rank $r \leq m, n$

1. $q \overset{\text{def}}{=} \Theta\left(\frac{\log d}{\sqrt{\epsilon}}\right)$, $\Pi \sim \mathcal{N}(0, 1)^{n \times r}$
2. $K \overset{\text{def}}{=} [LL, (LL^\top)L, L, \ldots, (LL^\top)^qL, \ldots]$\Pi]
3. Orthonormalize the columns of $K$ to obtain $Q \in \mathbb{R}^{m \times qr}$
4. Compute $M \overset{\text{def}}{=} Q^\top LL^\top Q \in \mathbb{R}^{qr \times qr}$
5. Set $\bar{U}_r$ to the top $r$ singular vectors of $M$
6. Output : $Z = Q\bar{U}_r$