Robust Principal Component Analysis using Facial Reduction

Shiqian Ma* Fei Wang† Linchuan Wei‡ Henry Wolkowicz§

Tuesday 20th March, 2018

Abstract

We study algorithms for robust principal component analysis (RPCA) for a partially observed data matrix. The aim is to recover the data matrix as a sum of a low-rank matrix and a sparse matrix so as to eliminate erratic noise (outliers). This problem is known to be NP-hard in general. A classical way to solve RPCA is to consider convex relaxations, e.g., to minimize the sum of the nuclear norm, to obtain a low-rank component, and the ℓ_1 norm, to obtain a sparse component. This is a well-structured convex problem that can be efficiently solved by modern first-order methods. However, first-order methods often yield low accuracy solutions. In this paper, we propose a novel nonconvex reformulation of the original NP-hard RPCA model. The new model imposes a semidefinite cone constraint and utilizes a facial reduction technique. we are thus able to reduce the size significantly. This makes the problem amenable to efficient algorithms in order to obtain high-level accuracy. We include numerical results that confirm the efficacy of our approach.

Keywords: Robust Principal Component Analysis, Semidefinite Cone, Facial Reduction, Biclique

AMS subject classifications: 65J22, 90C22, 65K10, 52A41

Contents

*Department of Mathematics, University of California, Davis, CA 95616, USA. Research supported by a startup package from Department of Mathematics at UC Davis.

†Department of Mathematics, Royal Institute of Technology, Sweden

‡Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1. Research supported by the Undergraduate Student Research Awards Program, Natural Sciences and Engineering Research Council of Canada.

§Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1; Research supported by The Natural Sciences and Engineering Research Council of Canada.

List of Tables

List of Algorithms

1 Introduction

Principal component analysis (PCA) seeks a low-rank approximation to an input data matrix. It is arguably the most popular statistical tool in data analysis and is very effective for reducing the force of random Gaussian noise. PCA can be obtained easily via the singular value decomposition (SVD). However, it is also well-known that PCA lacks robustness with respect to gross erratic noise, i.e., a single corrupted entry can result in an approximation that is far away from the true solution. Robust PCA has been proposed to remove the effect of sparse gross errors. In e.g., [5–7], the intractable RPCA problem and its convex relaxation are introduced and it is shown that they are equivalent with high probability under certain conditions of the input data. Specifically, RPCA aims to express a given data matrix $Z \in \mathbb{R}^{m\times n}$ as the sum $Z = L^* + S^*$, where L^* is the low-rank approximation to Z, and S^* is a sparse matrix that captures the additive erratic noise. Throughout this paper, we assume that $r := \text{rank}(L^*) \ll \min(m, n)$ and $\mu > 0$ is a given positive parameter.

RPCA can be formulated as the following optimization problem:

$$\begin{array}{ll} \min & \operatorname{rank}(L) + \mu\|S\|_0 \\ \text{s.t.} & L + S = Z, \end{array} \tag{1.1}$$

where the cardinality function $\|S\|_0$ counts the number of nonzeros of S. This problem is NP-hard due to the combinatorial nature of the rank and the cardinality functions. It is shown in e.g., [5,6] that under certain conditions, (1.1) is equivalent, with high probability, to the following convex program:

$$\begin{array}{ll} \min & \|L\|_* + \mu\|S\|_1 \\ \text{s.t.} & L + S = Z. \end{array} \tag{1.2}$$

Here $\|L\|_*$ is the nuclear norm of L, the sum of the singular values of L, and the ℓ_1 norm is $\|S\|_1 := \sum_{ij} |S_{ij}|$. The convex program (1.2) is known as robust principal component pursuit (RPCP).

In practice, it is possible that the matrix Z is only partially observed. That is, there exists a subset $\hat{E}$ of the indices such that only entries Z_{ij} for $(i,j) \in \hat{E}$ are given. In this case, RPCA (1.1) and RPCP (1.2) need to be changed, respectively, to

$$F_1(L,S) := \begin{array}{ll} \min & \operatorname{rank}(L) + \mu\|S\|_0 \\ \text{s.t.} & \mathcal{P}_{\hat{E}}(L + S) = z, \end{array} \tag{1.3}$$

and

$$\begin{array}{ll} \min & \|L\|_* + \mu\|S\|_1 \\ \text{s.t.} & \mathcal{P}_{\hat{E}}(L + S) = z, \end{array} \tag{1.4}$$

where $\mathcal{P}_{\hat{E}} : \mathbb{R}^{m\times n} \to \mathbb{R}^{\hat{E}}$ denotes the projection that only keeps the components with indices in $\hat{E}$, and $z = \mathcal{P}_{\hat{E}}(Z)$.

The convex programs (1.2) and (1.4) can be reformulated as semidefinite programming problems (SDP), e.g., [5, 19]. Thus, they can be efficiently solved by e.g., interior point methods. However, RPCA arising from real applications is usually of huge scale and interior point methods generally do not scale well. As a result, current research on algorithms for RPCA and RPCP is focused on first-order methods, see e.g., the recent survey papers [1, 4, 17]. Nonetheless, it is also known that the first-order methods cannot generally provide as high accuracy solutions as the higher-order methods such as the interior point methods.

1.1 Main contributions

In this paper we propose a new approach to solve RPCA with partially observed data (1.3). We use semidefinite programming and a novel cone facial reduction (FR) technique employed to a reformulation of (1.3). As a result, the size of the feasible region can be significantly reduced and a more efficient algorithm can be used to get a highly accurate solution. This extends the approach in [12] for low-rank matrix completion. A key ingredient here is the use of the exposing vector approach for characterizing faces of semidefinite cones; see [8, 10, 12]. With FR we generally obtain a significant reduction in the size of the problem that is sent to a solver. Our preliminary numerical results indicate that in many cases, and even under relatively low density for the sampled data, we get exact recovery without using any SDP solver. This provides an *improvement* over simply using the convex relaxation approach, RPCP.

Organization.

We introduce further background on RPCA in Section 2; this includes SDP reformulations and graph representations. In Section 3 we discuss how to obtain accurate solutions for small fully sampled submatrices. This is the key in obtaining high accurate exposing vectors for faces used in reducing the size of the complete problem, while still maintaining low error growth that could arise due to noise. Section 4 presents the main theorems used in the paper. This includes facial reduction, bicliques and exposing vectors, as well as the algorithm for solving the RPCA problem accurately. In Section 5 we discuss a heuristic on growing the size of bicliques using a submatrix approach. We present the numerical results in Section 6 for both noiseless and noisy data. We conclude in Section 7.

2 Background

In this paper we focus on model (1.3). Our approach starts with a reformulation of (1.3) using an additional semidefinite cone constraint.

2.1 Reformulating RPCA with SDP

In this section we show that RPCA (1.3) is equivalent to the following SDP problem:

$$\begin{array}{rll} F_2(Y,S) := & \min\limits_{Y,S} & \operatorname{rank}(Y) + \mu\|S\|_0 \\ & \text{s.t.} & \mathcal{P}_{\hat{E}}(L+S) = z \\ & & Y = \begin{bmatrix} W_1 & L \\ L^\top & W_2 \end{bmatrix} \succeq 0. \end{array} \tag{2.1}$$

Lemma 2.1. *Problems* (1.3) *and* (2.1) *are equivalent in the sense that they have the same optimal solution pair* (L^*, S^*) *and the same optimal objective value. In particular, the optimal* L^* *is a submatrix of the optimal* Y^**.*

Proof. Suppose (L^*, S^*) solves (1.3) with $\operatorname{rank}(L^*) = r < \min(m,n)$, and the compact SVD of L^* is given by $L^* = U\Sigma V^\top$, where both $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ have orthonormal columns, and $\Sigma \in \mathbb{R}^{r\times r}_{++}$ is a diagonal matrix with the positive singular values of L^* on its diagonal. We now see that this is equivalent to (Y^*, S^*) solves (2.1) with

$$Y^* = \begin{bmatrix} U \\ V \end{bmatrix} \Sigma \begin{bmatrix} U \\ V \end{bmatrix}^\top = \begin{bmatrix} U\Sigma U^\top & U\Sigma V^\top \\ V\Sigma U^\top & V\Sigma V^\top \end{bmatrix} = \begin{bmatrix} U\Sigma U^\top & L^* \\ (L^*)^\top & V\Sigma V^\top \end{bmatrix}, \tag{2.2}$$

from which we get that $\operatorname{rank}(Y^*) = \operatorname{rank}(L^*) = r$, and L^* is the upper right corner block of Y^*. This shows that for an optimal Y^* we have $\operatorname{rank}(Y^*) = \operatorname{rank}(L^*) = r$, and emphasizes that the only change in the two problems is the addition of the semidefinite constraint.

The equivalence follows from: if there exists a better solution $(\hat{Y}, \hat{S})$, with the spectral decomposition of $\hat{Y}$ given by

$$\hat{Y} = \begin{bmatrix} \hat{U} \\ \hat{V} \end{bmatrix} \hat{\Sigma} \begin{bmatrix} \hat{U} \\ \hat{V} \end{bmatrix}^\top = \begin{bmatrix} \hat{U}\hat{\Sigma}\hat{U}^\top & \hat{U}\hat{\Sigma}\hat{V}^\top \\ \hat{V}\hat{\Sigma}\hat{U}^\top & \hat{V}\hat{\Sigma}\hat{V}^\top \end{bmatrix},$$

then we have

$$F_1(\hat{U}\hat{\Sigma}\hat{V}^\top, \hat{S}) \le F_2(\hat{Y}, \hat{S}) < F_2(Y^*, S^*) = r + \mu\|S^*\|_0 = F_1(L^*, S^*),$$

where the first inequality is due to the fact $\text{rank}(\hat{U}\hat{\Sigma}\hat{V}^\top) \leq \text{rank}(\hat{\Sigma}) = \text{rank}(\hat{Y})$. This is a contradiction. □

Remark 2.2. *Lemma 2.1 indicates that in order to solve* (1.3)*, we can solve* (2.1) *instead to obtain* (Y^*, S^*)*, and then* (L^*, S^*) *solves* (1.3)*, where* L^* *is obtained from the upper right corner block of* Y^**. Seemingly,* (2.1) *is harder than* (1.3) *because it is much larger. However, the fact that the optimal* L^* *and* Y^* *are both of low rank enables us to reduce the size of* (2.1) *greatly to a new problem whose size is much smaller than* (1.3)*, by pursuing the facial structure of the semidefinite cone in* (2.1)*. This is the main motivation for our approach.*

2.2 Outline of our algorithm

We now present a brief outline of our algorithm based on the graph representation of the partially observed matrix Z and the facial reduction of the reformulation (2.1).

For a given partially observed data matrix $Z \in \mathbb{R}^{m\times n}$, we aim to find the low-rank-plus-sparse decomposition $Z = L^* + S^*$ with $r = \text{rank}(L^*) \ll \min(m, n)$. The main steps of our algorithm are as follows:

(i) Associate a bipartite graph to Z using the observed entries, and find a biclique associated to a completely determined submatrix of Z. Denote this submatrix as $\bar{Z} \in \mathbb{R}^{p\times q}$, where p, q are generally much smaller than m, n, respectively.

(ii) Find the low-rank-plus-sparse decomposition for $\bar{Z}$, i.e., $\bar{Z} = \bar{L} + \bar{S}$. Note that since $p \ll m, q \ll n$, this problem is much easier than the original problem.

(iii) From $\bar{L}$ we are able to find an exposing vector for a face of the SDP cone that contains an optimal Y^* that solves (2.1), i.e., an exposing vector of the optimal face. Note that $\bar{L}$ is a submatrix of L^* and thus a submatrix of Y^*.

(iv) We use the fact that the sum of exposing vectors of faces is an exposing vector of the intersection of the faces, see (4.1) below. We can then reformulate (2.1) into a new problem whose size is much smaller than (1.3). This new problem can then be solved efficiently and accurately.

2.3 Graph representation of a partially observed matrix Z

We associate a bipartite graph $G_Z((U_m, V_n), \hat{E})$ to Z, whose node set corresponds to the union of the two sets of rows and columns of Z

$$U_m = \{1, \ldots, m\} \cup V_n = \{1, \ldots, n\},$$

and there is an edge $(i, j) \in \hat{E}$, with $i \in U_m$ and $j \in V_n$ if Z_{ij} is observed. Note that a biclique of G_Z corresponds to a submatrix of Z whose entries are all observed. To find a biclique of $G_Z((U_m, V_n), \hat{E})$, we can relate it to finding cliques in the graph $G = (V, E)$ whose node set is $V = \{1, \ldots, m, m + 1, \ldots, m + n\}$ and the edge set E is

$$E := \hat{E} \cup \{(i, j) \in V \times V : 1 \leq i < j \leq m\} \cup \{(i, j) \in V \times V : m + 1 \leq i < j \leq m + n\}.$$

Suppose we find a non-trivial clique of G denoted by $C = \{i_1, \ldots, i_k\}$ whose cardinality $k = p + q$ satisfies

$$|C \cap \{1, \ldots, m\}| = p \neq 0, \quad |C \cap \{m+1, \ldots, m+n\}| = q \neq 0.$$

By removing the edges in C that have both nodes in $\{1, \ldots, m\}$ or $\{m+1, \ldots, m+n\}$, we obtain a biclique of G_Z. We use $\bar{C}$ to denote this biclique. This biclique then corresponds to a submatrix of Z, with entries corresponding to edges in $\bar{C}$ being kept and other entries of Z are removed. We use $\bar{Z}$ to denote this matrix whose size is $p \times q$. Note that $\bar{Z}$ is a submatrix of Z, and all entries of $\bar{Z}$ are observed. Moreover, we generally maintain the size of $\bar{Z}$ much smaller than the size of Z, $p \ll m$ and $q \ll n$.

2.4 Heuristics for finding cliques

Finding all the cliques from a graph is a NP-hard problem, [2]; the clique decision problem is one of Karp's 21 NP-complete problems, [13]. Moreover, the cost of finding the SVD needed for each submatrix corresponding to each clique found becomes expensive for large cliques and large submatrices. Therefore, we use a heuristic algorithm that is proposed in [14] and [8, Algorithm 2] to efficiently find many small cliques. This heuristic algorithm is briefly outlined in Algorithm 2.1.

Algorithm 2.1 Heuristics for finding a set of cliques

Input: A Graph $G = (V, E)$, maximum clique size K_{max}, minimum clique size K_{min}
Initialization: Set the clique set Θ as an empty set
for each node $v \in G$ **do**
    Initialize $\alpha \leftarrow \{v\}$
    while $|\alpha| < K_{max}$ and there is a w that is connected to every node in α **do**
        $\alpha \leftarrow \alpha \cup \{w\}$
    end while
    if $|\alpha| \geq K_{min}$ **then** $\Theta \leftarrow \Theta \cup \{\alpha\}$
    end if
end for
Output: The clique set Θ

Algorithm 2.1 is very efficient in practice because it only goes through each node in the graph G once. In total, we need to add at most $K_{max}|V|$ nodes which corresponds to $K_{max}|V|$ row operations on the adjacency matrix of the graph.

3 Decomposing the submatrix $\bar{Z}$ using PALM

From Sections 2.3 and 2.4 we know that by finding bicliques of G_Z, we get a submatrix of Z, denoted by $\bar{Z} \in \mathbb{R}^{p \times q}$ with $K_{min} \leq p + q \leq K_{max}$, whose entries are all known (sampled). Now we want to find a low-rank-plus-sparse decomposition of $\bar{Z}$. This problem can be formulated as

$$\begin{array}{ll} \min\limits_{\bar{L}, \bar{S}} & \frac{1}{2}\|\bar{L} + \bar{S} - \bar{Z}\|_F^2 \\ \text{s.t.} & \text{rank}(\bar{L}) \leq \bar{r}, \quad \|\bar{S}\|_0 \leq \bar{s}, \end{array} \tag{3.1}$$

where $\bar{r}$ and $\bar{s}$ are given parameters to control the rank of $\bar{L}$ and sparsity of $\bar{S}$, respectively. There are two reasons that (3.1) is much easier to solve than (1.3). The first reason is that all entries

of $\bar{Z}$ are known, while those in Z are only partially known. The second reason is that the size of $\bar{Z}$ is much smaller than the size of Z. We adopt the proximal alternating linearization method (PALM) [3] to solve (3.1). This is summarized in Algorithm 3.1.

Algorithm 3.1 PALM for Solving (3.1)

INPUT: $\bar{Z} \in \mathbb{R}^{p\times q}$, $\bar{r}$, $\bar{s}$, initial matrices $\bar{L}^0, \bar{S}^0$, step sizes $\gamma_1 > 1$, $\gamma_2 > 1$.
for $k = 0, 1, \ldots,$ **do**
  $G_L^k := \bar{L}^k - \frac{1}{\gamma_1}(\bar{L}^k + \bar{S}^k - \bar{Z})$.
  $\bar{L}^{k+1} := \operatorname{argmin}_{\bar{L}\in\mathbb{R}^{p\times q}}\{\|\bar{L} - G_L^k\|_F^2 : \operatorname{rank}(\bar{L}) \le \bar{r}\}$.
  $G_S^k := \bar{S}^k - \frac{1}{\gamma_2}(\bar{L}^{k+1} + \bar{S}^k - \bar{Z})$.
  $\bar{S}^{k+1} := \operatorname{argmin}_{\bar{S}\in\mathbb{R}^{p\times q}}\{\|\bar{S} - G_S^k\|_F^2 : \|\bar{S}\|_0 \le \bar{s}\}$.
end for
Set $\bar{L} := \bar{L}^k$ and $\bar{S} := \bar{S}^k$.
OUTPUT: Low-rank component $\bar{L}$ and sparse component $\bar{S}$.

By introducing indicator functions, (3.1) can be equivalently written as

$$\min_{\bar{L},\bar{S}} \ \Psi(\bar{L},\bar{S}) := \frac{1}{2}\|\bar{L} + \bar{S} - \bar{Z}\|_F^2 + \mathbb{1}(\bar{L} \mid \operatorname{rank}(\bar{L}) \le \bar{r}) + \mathbb{1}(\bar{S} \mid \|\bar{S}\|_0 \le \bar{s}), \tag{3.2}$$

where the indicator function $\mathbb{1}(X \mid \mathcal{X})$ equals 0 if $X \in \mathcal{X}$, and equals $+\infty$ otherwise. It is easy to verify that the objective function $\Psi(\bar{L},\bar{S})$ satisfies the so-called Kurdyka-Łojasiewicz property [3, 15,16]. As a result, we have the following convergence result for Algorithm 3.1 that follows directly from [3, Theorem 1].

Theorem 3.1. *The sequence $\{\bar{L}^k, \bar{S}^k\}_{k\in\mathbb{N}}$ generated by Algorithm 3.1 converges to a critical point of problem* (3.1).

Suppose we solve (3.1) and obtain the optimal value zero that guarantees a global optimum. To ensure the correct $\bar{L}$ is recovered, we need this global optimum to be unique. This happens with high probability when the rank of $\bar{L}$ is much smaller than the size of the matrix, and the incoherence conditions hold, see e.g., [5, 6]. Therefore, we make the following assumption throughout the remainder of the paper.

Assumption 3.2. *Assume $\bar{r} \ll \min(p, q)$ and assume that* (3.1) *has a unique global optimal solution $\bar{L}$ that is a submatrix of L^* and has the same rank as L^*.*

The uniqueness also happens with higher probability when $\bar{s}$ is smaller. Suppose D_S is the density of the sparsity matrix S^*, in order to control the sparsity part in problem (3.1), we vary $\bar{s}$ from 1 to $\max\{1, pqD_S\}$. We also set the target rank $\bar{r} = r$. Thus we have the following algorithm:

4 Facial Reduction, Bicliques and Exposing Vectors

4.1 Preliminaries on faces

We now present some of the geometric facts we need. More details can be found in e.g., [8,10,14,18]. Recall that the set K is a proper convex cone if

$$K + K \subseteq K,\ \lambda K \subseteq K, \forall \lambda \ge 0,\ \operatorname{int}(K) \neq \emptyset.$$

Algorithm 3.2 Decompose $\bar{Z}$

INPUT: $\bar{Z} \in \mathbb{R}^{p\times q}$, target rank r such that $r < \min\{p, q\}$, density D_S of the sparsity matrix.
$T := \max\{1, pqD_S\}$
for $\bar{s} = 1, \ldots, T$ **do**
    Decompose $\bar{Z}$ by solving (3.1) using PALM (Algorithm 3.1).
    if The optimal value of (3.1) is zero **then**
        Return success and $\bar{L}, \bar{S}$.
    end if
end for
Return failure.
OUTPUT: Success with $\bar{L}, \bar{S}$ or failure

The *dual cone,* K^*, is defined by

$$K^* = \{\phi \in R^n : \langle \phi, k\rangle \geq 0, \forall k \in K\}.$$

A subcone $F \subseteq K$ is a *face,* $F \trianglelefteq K$, of the convex cone K if

$$x, y \in K, x + y \in F \implies x, y \in F.$$

The *conjugate face,* F^*, is defined by $F^* = F^\perp \cap K^*$, where $F^\perp$ denotes the *orthogonal complement* of F. A face $F \trianglelefteq K$ is an *exposed face* if there exists $\phi \in K^*$ such that $F = \phi^\perp \cap K$; and ϕ is an *exposing vector*. Let T be a subset of the convex cone K, then face(T) is the smallest face of K containing T. It is known that: a face of a face is a face; an intersection of faces is a face; and essential for our algorithm is the following for finding an intersection of exposed faces $F_i \trianglelefteq K, i = 1, \ldots, k$, see [8],

$$F_i = K \cap \phi_i^\perp, \forall i \implies \cap_{i=1}^k F_i = \left(\sum_{i=1}^k \phi_i\right)^\perp \cap K. \tag{4.1}$$

If $K = \mathcal{S}_+^n$, i.e., the positive semidefinite cone, then the facial structure is well understood. Faces are characterized by the ranges or nullspaces of the matrices in the face. Let $X \in \mathcal{S}_+^n$ be rank r and

$$X = \begin{bmatrix} P & Q \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P & Q \end{bmatrix}^\top$$

be the (orthogonal) spectral decomposition with $D \in \mathcal{S}_{++}^r$ being a diagonal matrix. Then the smallest face containing X is

$$\text{face}(X) = P\mathcal{S}_+^r P^\top = \mathcal{S}_+^n \cap (QQ^\top)^\perp.$$

The matrix $QQ^\top$ is an *exposing vector* for face(X). Moreover, the relative interior satisfies

$$\text{relint}(\text{face}(X)) = P\mathcal{S}_{++}^r P^\top = \text{relint}(\text{face}(\hat{X})), \quad \forall \hat{X} \in \text{relint}(\text{face}(X)),$$

i.e., the face and the exposing vectors are characterized by the eigenspace of any $\hat{X}$ in the relative interior of the face.

For our application we use the following view of facial reduction and exposed faces.

Theorem 4.1 ([9, Theorem 4.1], [10]). *Consider a linear transformation $\mathcal{M}\colon \mathcal{S}^n \to \mathbb{R}^m$ and a nonempty feasible set*

$$\mathcal{F} := \{X \in \mathcal{S}^n_+ : \mathcal{M}(X) = b\},$$

for some $b \in \mathbb{R}^m$. Then a vector v exposes a proper face of $\mathcal{M}(\mathcal{S}^n_+)$ containing b if, and only if, v satisfies the auxiliary system

$$0 \neq \mathcal{M}^* v \in \mathcal{S}^n_+ \quad \text{and} \quad \langle v, b\rangle = 0.$$

Let N denote the smallest face of $\mathcal{M}(\mathcal{S}^n_+)$ containing b. Then the following statements are true.

1. *We always have $\mathcal{S}^n_+ \cap \mathcal{M}^{-1}N = \text{face}(\mathcal{F})$, the smallest face containing $\mathcal{F}$.*
2. *For any vector $v \in \mathbb{R}^m$ the following equivalence holds:*

$$v \text{ exposes } N \iff \mathcal{M}^* v \text{ exposes } \text{face}(\mathcal{F}). \tag{4.2}$$

The result in (4.2) details the facial reduction process for the matrix completion problem using exposing vectors. More precisely, if $B \succeq 0$ is a principal submatrix of the data and $\text{trace}(VB) = 0, V \succeq 0$, then V provides an exposing vector for the image of the coordinate map. We can then complete V with zeros to get $Y \in \mathcal{S}^n_+$ an exposing vector for $\mathcal{F}$. Define the *triangular number,* $t(n) = n(n+1)/2$, and the projection vec: $\mathcal{S}^n \to \mathbb{R}^{t(n)}$ that vectorizes the upper-triangular part of a symmetric matrix columnwise.

Corollary 4.2. *Suppose that $1 < k < n$ and $\mathcal{M}$ in Theorem 4.1 is the coordinate projection onto the leading principal submatrix of order $k, m = t(k)$. Let $B \in \mathcal{S}^k_+$, $b = \text{vec}(B) \in \mathbb{R}^{t(k)}$, i.e., for $X \in \mathcal{S}^n$, we have*

$$\mathcal{M}(X)_{ij} = b_{ij}, \quad \forall 1 \leq i \leq j \leq k.$$

Let

$$V \in \mathcal{S}^k_+, \ \text{trace}(VB) = 0, \ v = \text{vec}\, V.$$

Then $Y = \mathcal{M}^* v$ is an exposing vector for the feasible set $\mathcal{F}$, i.e.,

$$\text{trace}(Y(\mathcal{F})) = 0.$$

Proof. The proof follows immediately from Theorem 4.2 as v exposes N and $Y = \mathcal{M}^* v$ is an exposing vector for face($\mathcal{F}$). □

4.2 Exposing vector

The following lemma shows a generic rank property of a matrix with its submatrix.

Lemma 4.3 (Generic rank property (see Lemma 3.6 in [12])). *Let r be a positive integer and $Z_1 \in \mathbb{R}^{m\times r}$ and $Z_2 \in \mathbb{R}^{n\times r}$ be continuous random variables with i.i.d. entries. Set $Z = Z_1 Z_2^\top$ and let $X \in \mathbb{R}^{p\times q}$ be any submatrix of Z with $\min(p, q) \geq r$. Then $\text{rank}(X) = r$ with probability 1.*

Based on Lemma 4.3, we can assume that $\bar{L}$ returned by Algorithm 3.2 has the same rank as the targeting low-rank matrix L^* if $\min(p, q) > r$, i.e.,

$$\bar{Z} = \bar{L} + \bar{S}, \qquad \text{rank}(\bar{L}) = r. \tag{4.3}$$

That is, we solved (3.1) to global optimality with objective value being 0.

Proposition 4.4. *Under Assumption 3.2, suppose that PALM (Algorithm 3.1) returns $\bar{L}$ and $\bar{S}$ such that* (4.3) *holds. Without loss of generality, we further assume the targeting low-rank matrix L^* can be partitioned as $L^* = \left[\begin{array}{c|c} L_1 & L_2 \\ \hline \bar{L} & L_3 \end{array}\right]$ (after a permutation if needed), where $\bar{L} \in \mathbb{R}^{p\times q}$ and $r \le \min(p,q)$, and the SVD of $\bar{L}$ is given by*

$$\bar{L} = \begin{bmatrix} \bar{P} & \bar{U} \end{bmatrix} \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{Q} & \bar{V} \end{bmatrix}^\top, \tag{4.4}$$

where $\Sigma_r \in \mathcal{S}^r_{++}$. By adding appropriate blocks of zeros to $\bar{U}\bar{U}^\top$ and $\bar{V}\bar{V}^\top$ (after a permutation if needed), we get the following matrix W, which is an exposing vector for face(Y^*)*, i.e., $Y^* \cdot W = 0$, where Y^* is the optimal solution of* (2.1).

$$W = \left[\begin{array}{c|ccc|c} 0 & 0 & 0 & 0 \\ \hline 0 & \bar{U}\bar{U}^\top & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right] + \left[\begin{array}{c|ccc|c} 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & \bar{V}\bar{V}^\top & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right] = \left[\begin{array}{c|ccc|c} 0 & 0 & 0 & 0 \\ \hline 0 & \bar{U}\bar{U}^\top & 0 & 0 \\ 0 & 0 & \bar{V}\bar{V}^\top & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right].$$

Proof. Lemma 2.1 shows that $\operatorname{rank}(Y^*) = \operatorname{rank}(L^*) = r$. Without loss of generality, we can assume that (after a permutation if needed)

$$Y^* = \begin{bmatrix} U \\ P \\ Q \\ V \end{bmatrix} D \begin{bmatrix} U \\ P \\ Q \\ V \end{bmatrix}^\top = \left[\begin{array}{c|c|c|c} UDU^\top & UDP^\top & UDQ^\top & UDV^\top \\ \hline PDU^\top & PDP^\top & PDQ^\top & PDV^\top \\ \hline QDU^\top & QDP^\top & QDQ^\top & QDV^\top \\ \hline VDU^\top & VDP^\top & VDQ^\top & VDV^\top \end{array}\right]$$

and $L^* = \left[\begin{array}{c|c} UDQ^\top & UDV^\top \\ \hline PDQ^\top & PDV^\top \end{array}\right]$ with $PDQ^\top = \bar{L}$ and $D \in \mathcal{S}^r_{++}$.

Since $PDQ^\top = \bar{L}$, we have $\text{Range}(\bar{L}) \subseteq \text{Range}(P)$, $\text{Range}(\bar{L}^\top) \subseteq \text{Range}(Q)$. From $\bar{L} = \bar{P}\Sigma_r\bar{Q}^\top$ we have $\text{Range}(\bar{L}) \subseteq \text{Range}(\bar{P})$, $\text{Range}(\bar{L}^\top) \subseteq \text{Range}(\bar{Q})$. Since $\bar{P}, P, \bar{Q}, Q$ all have only r columns and $\operatorname{rank}(\bar{L}) = r$, it follows that $\text{Range}(P) = \text{Range}(\bar{P}) = \text{Range}(\bar{L})$ and $\text{Range}(Q) = \text{Range}(\bar{Q}) = \text{Range}(\bar{L}^\top)$. Therefore $P^\top\bar{U}\bar{U}^\top = \bar{P}^\top\bar{U}\bar{U}^\top = 0$ and $Q^\top\bar{V}\bar{V}^\top = \bar{Q}^\top\bar{V}\bar{V}^\top = 0$. It is then easy to check $Y^* \cdot W = 0$ and therefore W is an exposing vector of the optimal face that contains Y^* – the optimal solution of (2.1). □

4.3 Reducing the problem size using facial reduction for SDP cone

Using Proposition 4.4, we now present Algorithm 4.1 to find an exposing vector Y_{expo} for face(Y^*).

Remark 4.5. *For bicliques in Θ, we do not need them to be large ones, because we can take advantage of the fact that adding exposing vectors results in an exposing vector. As a result, Θ is just a set of very small bicliques, and therefore very easy to be found.*

From Y_{expo} we can also find the blocked basis for $\text{Null}(Y_{expo})$ which is given by the columns of

$$V = \text{Null}(Y_{expo}) = \begin{bmatrix} V_P & 0 \\ 0 & V_Q \end{bmatrix}, \quad V_P^\top V_P = I_{r_p},\ V_Q^\top V_Q = I_{r_q}, \tag{4.5}$$

Algorithm 4.1 Finding the final exposing vector

INPUT: partially observed matrix $Z \in \mathcal{M}^{m\times n}$, target rank r, density D_S, a set of bicliques Θ, minimum clique size $K_{min} > 2r$, set the set of bicliques $\bar{\Theta}$ as an empty set.
OUTPUT: final *block exposing vector* Y_{expo} and biclique set $\bar{\Theta}$.
for each biclique $\alpha \in \Theta$ and the corresponding submatrix $\bar{Z} \in \mathbb{R}^{p\times q}$ **do**
  if $\min\{p, q\} \geq \frac{K_{min}}{2}$ **then**
    Find the decomposition of $\bar{Z} = \bar{L} + \bar{S}$ satisfying (4.3) using Algorithm 3.2;
    $[\bar{U}, \bar{V}] \leftarrow$ from **SVD** of $\bar{L}$ (4.4)
    $W_\alpha^P \leftarrow \bar{U}\bar{U}^\top$;
    $W_\alpha^Q \leftarrow \bar{V}\bar{V}^\top$;
    add the biclique α to the set $\bar{\Theta}$
  end if
end for
sum over the bicliques with block submatrices filled in with appropriate size of zero matrices:

$$0 \neq Y_{expo} \leftarrow \left[\begin{array}{c|c} \sum_{\alpha\in\bar{\Theta}} W_\alpha^P & 0 \\ \hline 0 & \sum_{\alpha\in\bar{\Theta}} W_\alpha^Q \end{array}\right].$$

return Y_{expo} and $\bar{\Theta}$.

where $V_P \in \mathbb{R}^{m\times r_p}$ and $V_Q \in \mathbb{R}^{n\times r_q}$. We denote $r_v = r_p + r_q < m + n$. Therefore Y^* (optimal solution of (2.1)) can be expressed as

$$Y^* = VRV^\top = \begin{bmatrix} V_P R_p V_P^\top & V_P R_{pq} V_Q^\top \\ V_Q R_{pq}^\top V_P^\top & V_Q R_q V_Q^\top \end{bmatrix}, \quad \text{for some } R = \begin{bmatrix} R_p & R_{pq} \\ R_{pq}^\top & R_q \end{bmatrix} \in \mathcal{S}^{r_v}. \tag{4.6}$$

From Lemma 2.1 and (4.5) we have $\operatorname{rank}(Y^*) = \operatorname{rank}(V_P R_{pq} V_Q^\top) = \operatorname{rank}(R_{pq})$. Therefore, (2.1) reduces to the following problem which has a much smaller size for the low-rank component:

$$\begin{array}{ll} \min\limits_{R_{pq}\in\mathbb{R}^{r_p\times r_q}, S\in\mathbb{R}^{m\times n}} & \operatorname{rank}(R_{pq}) + \mu\|S\|_0 \\ \text{s.t.} & \mathcal{P}_{\hat{E}}(V_P R_{pq} V_Q^\top) + \mathcal{P}_{\hat{E}}(S) = z. \end{array} \tag{4.7}$$

4.4 Further Reducing the size of the problem

Note that the size of S in (4.7) can be further reduced. In the decomposition (4.3), we know that $\bar{L}$ and $\bar{S}$ are exactly recovered by PALM when successful. Therefore, there is a subset of entries of S^* that has been successfully recovered. We can remove this subset of entries from the linear constraints in (4.7). We let $\hat{E}_S$ denote the set of indices of entries of S^* that are exactly recovered by PALM, and let $\hat{E}_{S^c} := \hat{E}\backslash\hat{E}_S$ and get:

$$\begin{array}{ll} \min & \operatorname{rank}(R_{pq}) + \mu\|s\|_0 \\ \text{s.t.} & \mathcal{P}_{\hat{E}_S}(V_P R_{pq} V_Q^\top) = z_{\hat{E}_S} \\ & \mathcal{P}_{\hat{E}_{S^c}}(V_P R_{pq} V_Q^\top) + s = z_{\hat{E}_{S^c}} \\ & R_{pq} \in \mathbb{R}^{r_p\times r_q},\ s \in \mathbb{R}^{\hat{E}_{S^c}}. \end{array} \tag{4.8}$$

Moreover, the number of linear constraints in (4.8) can, and should, be further reduced to remove any redundant linear constraints that arose. We use A and B to denote the matrix representations

of the coefficient matrices in $\mathcal{P}_{\hat{E}_S}(V_P R_{pq} V_Q^\top)$ and $\mathcal{P}_{\hat{E}_{S^c}}(V_P R_{pq} V_Q^\top)$, respectively. If the ith row of B (denoted by B_i) is in the row space of A, then B_i can be removed so the size of B gets smaller. The algorithm that we use for this purpose is outlined in Algorithm 4.2.

Algorithm 4.2 Removing rows of B that are in the row space of A

1: **INPUT:** two matrices $A \in \mathbb{R}^{s_1 \times t}$, $B \in \mathbb{R}^{s_2} \times t$
2: Compute the rank of A, denote $r_A = \text{rank}(A)$.
3: Compute the LU decomposition of $\left[A^\top, B^\top\right]$
4: Remove the first r_A rows and first s_1 columns from the upper triangular matrix U and obtain $\bar{U}$.
5: Let I_c be the index set of the nonzero columns in $\bar{U}$.
6: Let $\bar{B}$ be the submatrix of B by removing rows of B corresponding to indices in I_c.
7: **OUTPUT:** $\bar{B}$

Since FR typically results in very small values of r_p and r_q in (4.5), the first set of linear constraints in (4.8) , i.e.,

$$\mathcal{P}_{\hat{E}_S}(V_P R_{pq} V_Q^\top) = z_{\hat{E}_S}, \tag{4.9}$$

is usually an overdetermined linear system. As a result, we can obtain L^* by only solving this linear system.

Lemma 4.6. *The target matrix L^* is unique if $\mathcal{P}_{\hat{E}_S}(V_P R_{pq} V_Q^\top) = z_{\hat{E}_{S^c}}$ in (4.8) has a unique solution.*

Lemma 4.7. *If L^* is the unique optimal solution of (1.3), then (4.8) has a unique optimal solution R_{pq}^* that recovers L^*:*

$$L^* = V_P R_{pq}^* V_Q^\top.$$

When we have enough bicliques, our numerical tests generally successfully recover L^* by solving the linear system (4.9). If we do not have enough bicliques, then we have to solve the nonconvex problem (4.8). In this case, we can solve the following convex relaxation of (4.8):

$$\begin{array}{ll} \min\limits_{R_{pq} \in \mathbb{R}^{r_p \times r_q}, s \in \mathbb{R}^{\hat{E}_{S^c}}} & \|R_{pq}\|_* + \mu \|s\|_1 \\ \text{s.t.} & \mathcal{P}_{\hat{E}_S}(V_P R_{pq} V_Q^\top) = z_{\hat{E}_S} \\ & \mathcal{P}_{\hat{E}_{S^c}}(V_P R_{pq} V_Q^\top) + s = z_{\hat{E}_{S^c}}. \end{array} \tag{4.10}$$

Standard first-order methods such as alternating direction method of multipliers can be used to solve (4.10). Moreover, since the size of (4.10) is very small, we can also use interior point methods to solve it. The convex relaxation (4.10) after facial reduction is usually tighter than (1.4) and is more likely to successfully recover L^*. We do not go into more details in this paper.

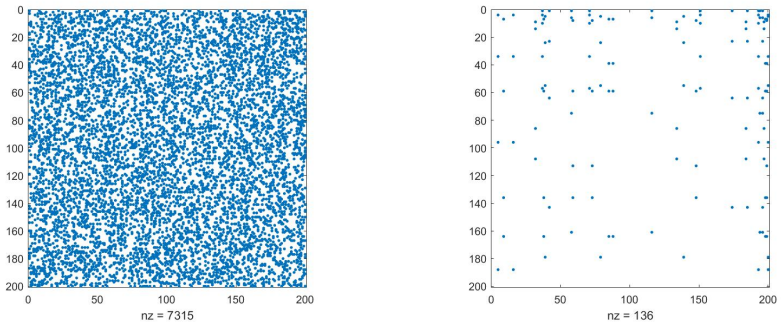
5 Growing bicliques by submatrix completion

We observe that when there are not enough bicliques, we may have many zero rows and zero columns in the exposing vector Y_{expo}. Let the set of indices for nonzero rows and columns in the upper-left block of Y_{expo} be J_1 and the set of indices for nonzero rows and columns in the bottom-right block of Y_{expo} be J_2. Now, if we consider the submatrix Z_{J_1,J_2} which is obtained by taking

the rows of Z whose indices are in J_1 and columns of Z whose indices are in J_2, we see that the size of the problem is much smaller, i.e., since we remove all the zero rows and columns and the exposing vectors now cover all the rows and columns of Z_{J_1,J_2}. Hence we have a better chance of recovering Z_{J_1,J_2}. Usually we can even get a unique solution.

After we recover the submatrix Z_{J_1,J_2}, we can add the indices for all the entries in Z_{J_1,J_2} to the sampled indices set $\hat{E}$, The sampled indices set $\hat{E}$ contains a bigger biclique $J_1 \times J_2$ and has more elements. We then use Algorithm 2.1 again to find more bicliques. This process can be repeated until all the rows and columns in Z are recovered.

We illustrate the growth of bicliques and the exposing vector Y_{expo} with an example. We choose the size of Z to be 200×200, the density for the sampled data is 0.18 and the minimum size for the bicliques is 4×4 with the target rank $r = 2$. The original low rank matrix L is recovered after 3 iterations by our algorithm. The example is shown in Figures 1 and 2.



(a) sampled elements (b) iteration 1

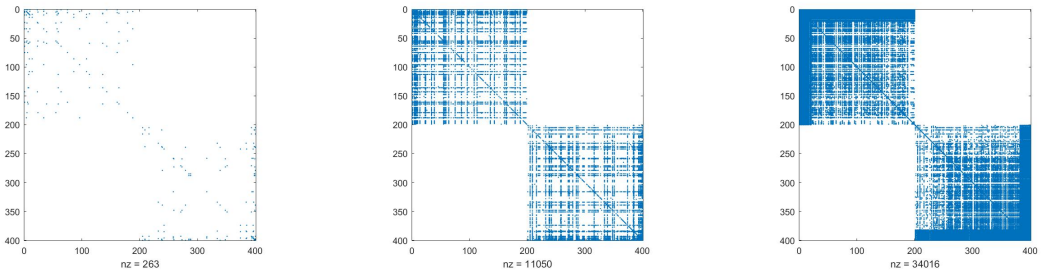
(c) iteration 2 (d) iteration 3

Figure 1: Growth of the bicliques: it takes 3 iterations for the set of bicliques to grow from very small to big enough to cover all the rows and columns.

Our main algorithm for recovering the low rank matrix is summarized in Algorithm 5.1.

6 Numerical Experiments

We applied our Algorithm 5.1 to solving random noiseless instances of (1.3). The input data were generated in the following manner. The low-rank matrix L^* was integer valued and obtained using



(a) iteration 1 (b) iteration 2 (c) iteration 3

Figure 2: Growth of Y_{expo}

Algorithm 5.1 Recovery of the low rank component L in RPCA with partially observed data (1.3)

INPUT: Partial matrix $Z \in \mathbb{R}^{m\times n}$ with sampling indices set $\hat{E}$, target rank r, density D_S, limit of iteration T, tolerance for the rank tol, range of clique size $[K_{min}, K_{max}]$;
Set $J_1 = J_2 = \emptyset$, $\hat{E}_S = \emptyset$, $W_{expo} = 0$, $j = 0$
while $J_1 \times J_2$ is growing with $|J_1| + |J_2| < m + n$ and $\text{rank}(L, tol) = r$ and $j < T$ **do**
  Find a set of bicliques $\Theta = \{\alpha_1, ..., \alpha_k\}$ in $\hat{E}$ by Algorithm 2.1.
  Calculate the exposing vector Y_{expo} and biclique set $\bar{\Theta}$ by Algorithm 4.1.
  Update

$$W_{expp} := W_{expo} + Y_{expo}.$$

  Set $Y_P = W_{expo}[1 : m, 1 : m]$ and $Y_Q = W_{expo}[m + 1 : m + n, m + 1 : m + n]$.
  Calculate the indices sets J_1 and J_2: $J_1 := \text{find}(\text{diag}(Y_P) > 0)$, $J_2 := \text{find}(\text{diag}(Y_Q) > 0)$.
  Update $Y_P = Y_P[J_1, J_1]$ and $Y_Q = Y_Q[J_2, J_2]$.
  Calculate $V_P = \text{Null}(Y_P)$ and $V_Q = \text{Null}(Y_Q)$ such that

$$V_P^\top V_P = I_{r_p},\ V_Q^\top V_Q = I_{r_q},$$

  Update $\hat{E}_S = \hat{E}_S \cup \bar{\Theta}$ and $\hat{E}_{S^c} = (\hat{E} \backslash \hat{E}_S) \cap \{J_1 \times J_2\}$.
  Solve problem (4.8) with the sampling indices $\hat{E}_S, \hat{E}_{S^c}$.
  Calculate the low rank matrix completion $L = V_P R_{pq} V_Q^\top$.
  Set $\hat{E} = \hat{E} \cup \{J_1 \times J_2\}$ and set $Z[J_1, J_2] = L$.
  $j = j + 1$.
end while
return L or FAILURE if $|J_1| + |J_2| < m + n$ or $\text{rank}(L, tol) \neq r$.

$L^* = \text{round}(L_r L_c^\top)$, where $L_r \in \mathbb{R}^{m\times r}$ and $L_c \in \mathbb{R}^{r\times n}$ are from the standard normal distribution, $N(0, 10)$. The density of the sparse matrix $S^* \in \mathbb{R}^{m\times n}$ is denoted by D_S, and the nonzero elements were integer valued and obtained by rounding the uniformly distributed random elements. Matrix Z is set as $Z = L^* + S^*$. We then randomly sample elements in Z using the density δ, i.e., the ratio of the observed components of Z is δ. We use $\hat{L}$ to denote the recovery returned by our Algorithm 5.1.

For each set of data parameters (m, n, δ), we randomly generated 20 instances and reported the averaged performance in Tables 6.1 and 6.2. In these tables, $D_S = 0.01$, $\mu = c/(\delta\sqrt{m})$ where

c is a chosen constant in $[1, 10]$, the cpu times are in seconds, K_{min} denotes the smallest size of the cliques we found, K_{max} denotes the largest size of the cliques we found, $\hat{r} = \text{rank}(\hat{L})$ and "Succ" denotes the number of successfully recovered instances out of 20 randomly generated ones. Here we claim that matrix L^* is successfully recovered if there is no difference between $\hat{L}$ and L^*, i.e., $\|\hat{L} - L^*\|_F = 0$. Note that this is possible because all entries of L^* are integer valued. From these results we see that we can get exact recovery with very low density. Moreover, our algorithm is very efficient, and for very large problems with size 5000×5000, we can solve it in about two minutes.

Table 6.1: Results of Algorithm 5.1 for solving (1.3) with target rank $r = 2$

Specifications			Succ	CPU	$\hat{r}$	K_{min}	K_{max}	μ
m	n	δ						
800	800	0.18	14	11.55	2	10	50	0.88
1000	1000	0.18	19	14.55	2	10	50	0.79
1200	1200	0.18	18	23.52	2	10	50	0.29
1500	1500	0.18	17	27.77	2	10	50	0.26
1800	1800	0.18	17	33.43	2	10	50	0.12
2100	2100	0.18	19	37.84	2	10	50	0.11
3000	3000	0.18	17	54.61	2	10	50	0.09
5000	5000	0.18	16	106.87	2	10	50	0.07

Table 6.2: Results of Algorithm 5.1 for solving (1.3) with target rank $r = 3$

Specifications			Succ	CPU	$\hat{r}$	K_{min}	K_{max}	μ
m	n	δ						
800	800	0.26	18	17.74	3	10	60	0.12
1000	1000	0.26	14	21.20	3	10	60	0.11
1500	1500	0.26	14	30.32	3	10	60	0.09
2000	2000	0.26	18	41.83	3	12	60	0.07
2500	2500	0.26	18	48.74	3	12	60	0.07
3000	3000	0.26	19	67.17	3	12	60	0.06
4000	4000	0.26	16	97.61	3	12	60	0.05
5000	5000	0.26	16	119.99	3	12	60	0.05

We also compare our method with the fast alternating linearization method (FALM) proposed in [11]. The results are reported in Table 6.3, where $D_S = 0.01$. From these results we see that after preprocessing by facial reduction, our reduced model (4.8) has a higher probability for recovering the low-rank matrix than solving the convex relaxation (1.4) by FALM. Moreover, our algorithm, although using a second-order interior point solver, is significantly faster than the first order method FALM.

Table 6.3: Comparison of our Algorithm 5.1 and FALM

Specifications				Succ FR	Succ FALM	CPU FR	CPU FALM
m	n	δ	r				
500	500	0.33	4	18	0	15.88	11.07
500	500	0.36	4	19	0	9.33	11.11
500	500	0.45	4	20	8	5.00	10.92
500	500	0.5	4	20	18	5.38	11.66
600	600	0.22	3	18	0	25.62	15.03
1000	1000	0.22	3	16	0	18.16	72.22
1000	1000	0.33	3	17	0	23.81	72.61
1000	1000	0.45	3	20	20	11.73	76.22
1000	1000	0.36	5	20	1	29.36	75.49
1000	1000	0.39	5	20	14	27.88	77.53
2000	2000	0.28	5	14	6	67.76	615.62
2000	2000	0.33	5	20	18	56.74	651.81
2000	2000	0.26	3	18	0	41.83	662.70
2000	2000	0.36	3	20	20	36.74	673.05
2100	2100	0.18	2	19	0	37.84	728.17
3000	3000	0.18	2	17	0	54.61	2216.66

7 Conclusion

In this paper we have shown that we can apply a facial reduction approach in combination with the nuclear norm heuristic to efficiently solve the robust PCA problem with partially observed data. This exploits the implicit degeneracy at the optimal solution resulting from the low-rank and sparse structure.

Specifically, whenever enough complete bipartite subgraphs in the data are available, we are able to find a proper face of the semidefinite cone that contains the optimal solution and results in a significant reduction in dimension. If we cannot find enough bicliques, the matrix can still be partially completed. Having an insufficient number of bicliques is indicative of not having enough initial data to recover the unknown elements for our algorithm. This is particularly true for large rank r, where larger bicliques are needed. Throughout we see that the facial reduction both regularizes the problem and reduces the size and often allows for a solution without any refinement or need for an SDP solver.

Our preliminary numerical results are promising as they efficiently and accurately recover large scale problems. The numerical tests are ongoing with improvements in using biclique algorithms rather than clique algorithms. In our paper we have started our tests with knowing the target rank r. In forthcoming tests we plan on using estimates for the target rank obtained from sampled submatrices.

Theoretical results on exact recovery are discussed in many papers, e.g., [5, 6]. They use the so-called incoherence conditions, which are difficult to verify. It appears from our work above that exact recovery guarantees can be found from rigidity results in the graph of Z, i.e., in the number and density of the bicliques. Moreover, there are interesting questions on how to extend these results from the simple matrix completion to general solutions of linear equations, $\mathcal{A}(Z) = b$, where

$\mathcal{A}$ is some linear transformation.

References

[1] N.S. Aybat. Algorithms for stable pca. In T. Bouwmans, N. S. Aybat, and E. Zahzah, editors, *Handbook of Robust Low Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. CRC Press, Taylor and Francis Group, 2016. 3

[2] J.R.S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. In *Graph theory and sparse matrix computation*, volume 56 of *IMA Vol. Math. Appl.*, pages 1–29. Springer, New York, 1993. 6

[3] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014. 7

[4] 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. 3

[5] E.J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis. *Journal of ACM*, 58(1):1–37, 2010. 2, 3, 7, 16

[6] V. Chandrasekaran, S. Sanghavi, P.A. Parrilo, and A.S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM J. Optim.*, 21(2):572–596, 2011. 2, 3, 7, 16

[7] J. Cheng, K. Chen, and M.D. Sacchi. Application of robust principal component analysis (RPCA) to suppress erratic noise in seismic records. In *SEG Technical Program Expanded*, pages 4646–4651. SEG, 2015. 2

[8] D. Drusvyatskiy, N. Krislock, Y-L. Cheung Voronin, and H. Wolkowicz. Noisy Euclidean distance realization: robust facial reduction and the Pareto frontier. *SIAM Journal on Optimization*, 27(4):2301–2331, 2017. 3, 6, 7, 8

[9] D. Drusvyatskiy, G. Pataki, and H. Wolkowicz. Coordinate shadows of semidefinite and Euclidean distance matrices. *SIAM J. Optim.*, 25(2):1160–1178, 2015. 9

[10] D. Drusvyatskiy and H. Wolkowicz. The many faces of degeneracy in conic optimization. *Foundations and Trends® in Optimization*, 3(2):77–170, 2017. 3, 7, 9

[11] D. Goldfarb, S. Ma, and K. Scheinberg. Fast alternating linearization methods for minimizing the sum of two convex functions. *Math. Program.*, 141(1-2, Ser. A):349–382, 2013. 15

[12] S. Huang and H. Wolkowicz. Low-rank matrix completion using nuclear norm with facial reduction. *J. Global Optim.*, 23 pages, to appear, 2017. 3, 9

[13] R.M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computation*, pages 85–103. Plenum Press, New York, 1972. 6

[14] N. Krislock and H. Wolkowicz. Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM Journal on Optimization*, 20(5):2679–2708, 2010. 6, 7

[15] K. Kurdyka. On gradients of functions definable in o-minimal structures. *Annales de l'institut Fourier*, 146:769–783, 1998. 7

[16] S. Łojasiewicz. *Une propriété topologique des sous-ensembles analytiques réels, Les Équations aux Dérivées Partielles*. Éditions du centre National de la Recherche Scientifique, Paris, 1963. 7

[17] S. Ma and N.S. Aybat. Efficient optimization algorithms for robust principal component analysis and its variants. *to appear*, 2018. 3

[18] G. Pataki. Geometry of Semidefinite Programming. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook OF Semidefinite Programming: Theory, Algorithms, and Applications*. Kluwer Academic Publishers, Boston, MA, 2000. 7

[19] B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, 2010. 3