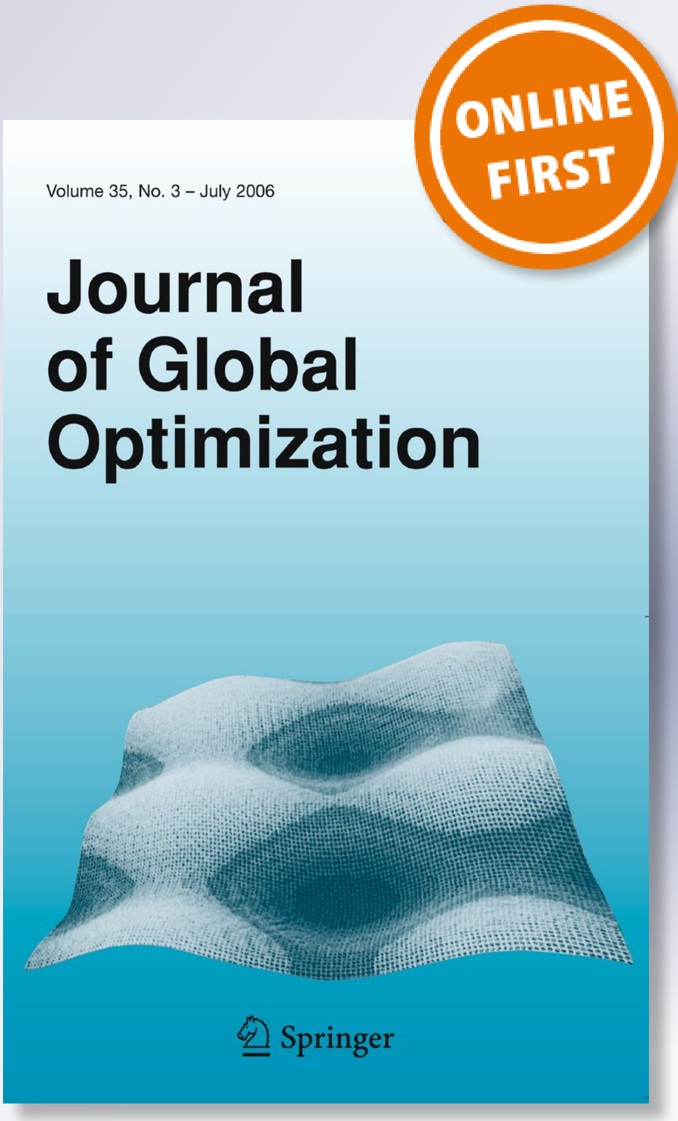
Low-rank matrix completion using nuclear norm minimization and facial reduction

Shimeng Huang & Henry Wolkowicz

Journal of Global Optimization
An International Journal Dealing with Theoretical and Computational Aspects of Seeking Global Optima and Their Applications in Science, Management and Engineering

ISSN 0925-5001

J Glob Optim
DOI 10.1007/s10898-017-0590-1



J Glob Optim
https://doi.org/10.1007/s10898-017-0590-1

Low-rank matrix completion using nuclear norm minimization and facial reduction

Shimeng Huang[1] · **Henry Wolkowicz**[1]

Received: 16 August 2016 / Accepted: 19 November 2017

Abstract Minimization of the nuclear norm, **NNM**, is often used as a surrogate (convex relaxation) for finding the minimum rank completion (recovery) of a *partial matrix*. The minimum nuclear norm problem can be solved as a trace minimization semidefinite programming problem, **SDP**. Interior point algorithms are the current methods of choice for this class of problems. This means that it is difficult to: solve large scale problems; exploit sparsity; and get high accuracy solutions. The **SDP** and its dual are regular in the sense that they both satisfy strict feasibility. In this paper we take advantage of the structure of low rank solutions in the **SDP** embedding. We show that even though strict feasibility holds, the facial reduction framework used for problems where strict feasibility fails can be successfully applied to *generically* obtain a proper face that contains all minimum low rank solutions for the original completion problem. This can dramatically reduce the size of the final **NNM** problem, while simultaneously guaranteeing a low-rank solution. This can be compared to identifying part of the active set in general nonlinear programming problems. In the case that the graph of the sampled matrix has sufficient bicliques, we get a low rank solution independent of any nuclear norm minimization. We include numerical tests for both exact and noisy cases. We illustrate that our approach yields lower ranks and higher accuracy than obtained from just the **NNM** approach.

Keywords Low-rank matrix completion · Matrix recovery · Semidefinite programming (**SDP**) · Facial reduction · Bicliques · Slater condition · Nuclear norm · Compressed sensing

Mathematics Subject Classification 65J22 · 90C22 · 65K10 · 52A41 · 90C46

Presented as Part of tutorial at DIMACS Workshop on Distance Geometry: Theory and Applications, July 26–29, 2016, [25]. Shimeng Huang: Research supported by the Undergraduate Student Research Awards Program, Natural Sciences and Engineering Research Council of Canada. Henry Wolkowicz: Research supported by The Natural Sciences and Engineering Research Council of Canada.

✉ Henry Wolkowicz
hwolkowicz@uwaterloo.ca

1 Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

Published online: 27 November 2017

Contents

1 Introduction

We consider the intractable *low-rank matrix completion problem*, **LRMC**, i.e., the problem of finding the missing elements of a given *partial matrix* so that the completion has low-rank. This problem can be relaxed using the nuclear norm that can then be solved using a *semidefinite programming*, **SDP**, model. Though the resulting **SDP** and its dual satisfy strict feasibility, we show that it is implicitly highly degenerate and amenable to *facial reduction*, **FR**. This is done by taking advantage of the special structure at minimum low rank solutions of the completion problem within the **SDP** formulation. This often results in improved low rank solutions compared to just using the nuclear norm relaxation. In addition, we get improved accuracy and efficiency in the algorithm. A key in the success is the use of the *exposing vector* approach, see [5], that is particularly amenable to the noisy case. Moreover, from **FR** we get a significant reduction in the size of the variables of the nuclear norm relaxation and a corresponding decrease in the possible rank of the solution of the **LRMC**. If the data is exact, then **FR** results in redundant constraints that we remove before solving for the low-rank solution. While if the data is contaminated with noise, **FR** yields an overdetermined semidefinite least squares problem. We *flip* this problem to minimize the nuclear norm using a Pareto frontier approach. Instead of removing constraints from the overdetermined problem, we exploit the notion of *sketch matrix* to reduce the size of the overdetermined problem. The sketch matrix approach is studied in e.g., [19].

The problem of **LRMC** has many applications to real applications in data science, model reduction, collaborative filtering (the well known Netflix problem) sensor network localization, pattern recognition and various other machine learning scenarios, e.g., [22,23]. See also the recent work in [2,20,24] and the references therein. Of particular interest is the case where the data is contaminated with noise. This falls into the area of *compressed sensing* or *compressive sampling*. An extensive collection of papers, books, codes is available at the Compressed Sensing 2.0 Community, sites.google.com/site/compressedsensing/.

The convex relaxation of minimizing the rank using the nuclear norm, the sum of the singular values, is studied in e.g., [10,11,20]. The solutions can be found directly by subgradient methods or by using **SDP** with interior point methods or low-rank methods, again see [20]. Many other methods have been developed, e.g., [17]. The two main approaches for rank minimization, convex relaxations and spectral methods, are discussed in [4,15] along with a new algebraic combinatorial approach. A related analysis from a different viewpoint using rigidity in graphs is provided in [21].

1.1 Outline

We continue in Sect. 2 with the basic notions for **LRMC** using the nuclear norm and with the graph framework that we employ. Then in Sect. 3 we include preliminaries on cone facial structure and the details on how to exploit **FR**, for the **SDP** model to minimize the rank. The main result for the reduction is in Lemma 3.4.

The results for the noiseless case are given in Sect. 4.1. This includes an outline of the basic approach in Algorithm 3.1 and empirical results from randomly generated problems. The noisy case follows in Sect. 4.2 with empirical results. We include a comparison against using CVX [13] and minimizing the nuclear norm directly in Sect. 4.3. Concluding remarks are given in Sect. 5.

2 Background on LRMC, NNM, SDP

We now consider our problem within the known framework on relaxing the low-rank matrix completion problem using the nuclear norm minimization and then using **SDP** to solve the relaxation. For the standard results we follow and include much of the known development in the literature e.g., [10,11,20]. In this section we also include several useful tools and a graph theoretic framework that allows us to exploit **FR** at the optimum.

2.1 Models

Suppose that we are given a (random) low rank $m \times n$ real matrix $Z \in \mathbb{R}^{m\times n}$ where a subset of entries are *sampled*. The **LRMC** can be modeled as follows:

$$(\textbf{LRMC}) \qquad \begin{array}{ll} \min & \operatorname{rank}(M) \\ \text{s.t.} & \mathcal{P}_{\hat{E}}(M) = z, \end{array} \tag{2.1}$$

where $\hat{E}$ is the set of indices containing the known (*sampled*) entries of Z, $\mathcal{P}_{\hat{E}}(\cdot) : \mathbb{R}^{m\times n} \to \mathbb{R}^{|\hat{E}|}$ is the projection onto the corresponding entries in $\hat{E}$, and $z = \mathcal{P}_{\hat{E}}(Z)$ is the vector of known entries formed from Z. However, the rank function is not a convex function and the **LRMC** is computationally intractable, e.g., [14].

To set up the problem as a convex optimization problem, we can relax the rank minimization using *nuclear norm minimization*, **NNM**:

$$(\textbf{NNM}) \qquad \begin{array}{ll} \min & \|M\|_* \\ \text{s.t.} & \mathcal{P}_{\hat{E}}(M) = z, \end{array} \tag{2.2}$$

where the nuclear norm $\|\cdot\|_*$ is the sum of the singular values, i.e., $\|M\|_* = \sum_i \sigma_i(M)$. The general primal–dual pair of problems for the **NNM** problem is

$$\begin{array}{ll} \min_M & \|M\|_* \\ \text{s.t.} & \mathcal{A}(M) = z, \end{array} \qquad\qquad \begin{array}{ll} \max_y & \langle z, y\rangle \\ \text{s.t.} & \|\mathcal{A}^*(y)\| \le 1, \end{array} \tag{2.3}$$

where $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^t$ is a linear mapping, $z \in \mathbb{R}^t$, $\mathcal{A}^*$ is the *adjoint of* $\mathcal{A}$, and $\|\cdot\|$ is the operator norm of a matrix, i.e., the largest singular value. The matrix norms $\|\cdot\|_*$ and $\|\cdot\|$

are a dual pair of matrix norms akin to the vector ℓ_1, ℓ_∞ norms on the vector of singular values. Without loss of generality, we further assume that $\mathcal{A}$ is *surjective*. In general, the linear equality constraint is an underdetermined linear system. In our case, we restrict to the case that $\mathcal{A} = \mathcal{P}_{\hat{E}}$.[1]

Proposition 2.1 *Suppose that, in the primal–dual pair (2.3), there exists $\hat{M}$ with $\mathcal{A}(\hat{M}) = z$. Then the pair of programs in (2.3) are a convex primal–dual pair and they satisfy both primal and dual strong duality, i.e., the optimal values are equal and both values are attained.*

Proof This is shown in [20, Prop. 2.1]. That primal and dual strong duality holds can be seen from the fact that the generalized Slater condition trivially holds for both programs using $M = \hat{M}$, $y = 0$, respectively. □

Corollary 2.2 *The optimal sets for the primal–dual pair in (2.3) are nonempty, convex, compact sets.*

Proof This follows since both problems are regular, i.e., since $\mathcal{A}$ is surjective, we conclude that the primal satisfies the *Mangasarian–Fromovitz constraint qualification*; while $y = 0$ shows that the dual satisfies strict feasibility. It is well known that these two constraint qualifications are equivalent to their respective dual problems having nonempty, convex, compact optimal sets, e.g., [12]. □

The following proposition shows that the nuclear norm minimization problem is **SDP** representable, i.e., we can embed the problem into an **SDP** and solve it efficiently. Here $Y \succeq 0$ denotes the Löwner partial order that Y is symmetric and positive semidefinite, denoted $Y \in \mathcal{S}_+^{m+n}$. We let $\succ 0$, $\mathcal{S}_{++}^n$ denote *positive definite*.

Proposition 2.3 *The optimal primal–dual solution set in (2.3) is the same as that in the* **SDP** *primal–dual pair:*

$$\begin{array}{ll} \min & \frac{1}{2}(\operatorname{trace}(W_1) + \operatorname{trace}(W_2)) \\ s.t. & Y = \begin{bmatrix} W_1 & M \\ M^T & W_2 \end{bmatrix} \succeq 0 \\ & \mathcal{A}(M) = z \end{array} \qquad \begin{array}{ll} \max_y & \langle z, y \rangle \\ s.t. & \begin{bmatrix} I_m & \mathcal{A}^*(y) \\ \mathcal{A}^*(y)^T & I_n \end{bmatrix} \succeq 0. \end{array} \tag{2.4}$$

□

This means that after ignoring the $\frac{1}{2}$ in the objective function, we can further transform the **NNM** problem as:

$$\begin{array}{ll} \min & \|Y\|_* = \operatorname{trace}(Y) \\ \text{s.t.} & \mathcal{P}_{\bar{E}}(Y) = z \\ & Y \succeq 0, \end{array} \tag{2.5}$$

where $\bar{E}$ is the set of indices in Y that correspond to $\hat{E}$, the known entries of the upper right block of $\begin{bmatrix} 0 & Z \\ Z^T & 0 \end{bmatrix} \in \mathcal{S}^{m+n}$. We emphasize that there is no constraint on the diagonal blocks of Y in (2.4) or in (2.5). Therefore, we can always obtain a positive definite feasible solution in this exact case by setting the diagonal elements of Y to be large enough. Therefore strict feasibility, the *Slater constraint qualification*, always holds. Further, we recall that the

[1] Note that the linear mapping $\mathcal{A} = \mathcal{P}_{\hat{E}} : \mathbb{R}^{m \times n} \to \mathbb{R}^{|\hat{E}|}$ corresponding to sampling is surjective as we can consider $\mathcal{A}(M)_{ij \in \hat{E}} = \operatorname{trace}(E_{ij} M)$, where E_{ij} is the ij-unit matrix.

original (nonconvex) objective function is the rank and that the nuclear norm provides the convex relaxation. Our aim is to minimize the rank of Y over the feasible set and resort to the relaxation using the trace only if needed at the end.

When the data is contaminated with noise, we reformulate the equality constraint by allowing the observed entries in the output matrix to be perturbed within a tolerance δ for the norm, where δ is normally a known noise level of the data, i.e.,

$$\begin{array}{ll} \min & \|Y\|_* = \text{trace}(Y) \\ \text{s.t.} & \|\mathcal{P}_{\bar{E}}(Y) - z\| \leq \delta \\ & Y \succeq 0. \end{array} \tag{2.6}$$

2.2 Graph representation of the problem

Our sampling yields elements $z = \mathcal{P}_{\hat{E}}(Z)$. With the matrix Z and the sampled elements we can associate a bipartite graph $G_Z = (U_m, V_n, \hat{E})$, where

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

Our algorithm exploits finding complete bipartite subgraphs, *bicliques*, in G_Z. We now relate this approach to finding cliques by using the larger symmetric matrix Y in (2.4). This allows us to exploit **FR** and apply the clique algorithms from [5,16]. However, we keep the biclique notation as much as possible.

Therefore, for our needs we associate Z with the *undirected graph*, $G = (V, E)$, with node set $V = \{1, \ldots, m, m+1, \ldots, m+n\}$ and edge set E that satisfies

$$\begin{aligned} &\big\{\{ij \in V \times V : i < j \leq m\} \cup \{ij \in V \times V : m+1 \leq i < j \leq m+n\}\big\} \subseteq E \\ &\quad \subseteq \{ij \in V \times V : i < j\}. \end{aligned}$$

Note that as above, $\bar{E}$ is the set of edges excluding the trivial ones, that is,

$$\bar{E} = E \backslash \big\{\{ij \in V \times V : i \leq j \leq m\} \cup \{ij \in V \times V : m+1 \leq i \leq j \leq m+n\}\big\}.$$

Recall that a *biclique* α in the graph G_Z is a complete bipartite subgraph in G_Z with corresponding complete submatrix $z[\alpha]$. This corresponds to a nontrivial[2] *clique* in the graph G, a complete subgraph in G. The cliques of interest are $C = \{i_1, \ldots, i_k\}$ with cardinalities

$$|C \cap \{1, \ldots, m\}| = p \neq 0, \quad |C \cap \{m+1, \ldots, m+n\}| = q \neq 0. \tag{2.7}$$

The submatrix $z[\alpha]$ of Z for the corresponding biclique from the clique C is

$$z[\alpha] \equiv X \equiv \{Z_{i(j-m)} : ij \in C\}, \quad \text{sampled } p \times q \text{ rectangular submatrix.} \tag{2.8}$$

These non-trivial cliques in G that correspond to bicliques of G_Z are at the center of our algorithm.

[2] For G we have the additional trivial cliques of size k, $C = \{i_1, \ldots, i_k\} \subset \{1, \ldots, m\}$ and $C = \{j_1, \ldots, j_k\} \subset \{m+1, \ldots, m+n\}$, that are not of interest to our algorithm.

Example 2.4 (Biclique for X) Let the $m \times n$ data matrix of rank r with $m = 7, n = 6, r = 2$ be

$$Z = \begin{bmatrix} -5 & 15 & 10 & -20 & -21 & -6 \\ 4 & 0 & 4 & 4 & 6 & 6 \\ -3 & -35 & -38 & 32 & 27 & -8 \\ 5 & -5 & 0 & 10 & 12 & 7 \\ 0 & -30 & -30 & 30 & 27 & -3 \\ 3 & -5 & -2 & 8 & 9 & 4 \\ 5 & 5 & 10 & 0 & 3 & 8 \end{bmatrix}.$$

After sampling we have unknown entries denoted by NA and known entries in

$$\begin{bmatrix} -5 & NA & 10 & -20 & NA & -6 \\ 4 & 0 & 4 & 4 & 6 & 6 \\ -3 & NA & NA & 32 & 27 & NA \\ 5 & NA & 0 & 10 & 12 & NA \\ NA & -30 & NA & NA & 27 & NA \\ 3 & -5 & -2 & 8 & NA & 4 \\ 5 & 5 & NA & 0 & 3 & NA \end{bmatrix}.$$

Then $z = \mathcal{P}_{\hat{E}}(Z)$ denotes a vector representation of the known entries. $\bar{E}$ denotes the corresponding indices for $\hat{E}$ when Z is considered in the big matrix Y and E is formed from $\bar{E}$ by adding on the indices corresponding to the diagonal blocks.

Suppose that our algorithm found a biclique α with indices

$$\bar{U}_m = \{6, 1, 2\}, \quad \bar{V}_n = \{1, 4, 3, 6\}.$$

The corresponding submatrix is

$$z[\alpha] \equiv X = \begin{bmatrix} 3 & 8 & -2 & 4 \\ -5 & -20 & 10 & -6 \\ 4 & 4 & 4 & 6 \end{bmatrix}.$$

The sampled large matrix Y containing the sampled Z is filled in with the word *free* on the diagonal blocks to emphasize that these blocks are free during the algorithm. Then the clique C_X corresponding to the biclique and the corresponding principal submatrix of Y corresponding to X are, respectively,[3]

$$C_X = \{6, 1, 2|, |1+7, 4+7, 3+7, 6+7\} = \{6, 1, 2|, |8, 11, 10, 13\},$$

and

$$\begin{bmatrix} & \text{FREE} & & 3 & 8 & -2 & 4 \\ & & & -5 & -20 & 10 & -6 \\ & & & 4 & 4 & 4 & 6 \\ 3 & -5 & 4 & & & & \\ 8 & -20 & 4 & & & \text{FREE} & \\ -2 & 10 & 4 & & & & \\ 4 & -6 & 6 & & & & \end{bmatrix}$$

[3] We have a bar | to emphasize the end/start of the row/column indices.

3 Facial reduction, bicliques, exposing vectors

In this section we look at the details of **FR** and how to solve the facially reduced **SDP** formulation for **LRMC**. In particular we show how to exploit bicliques in the graph G_Z and the *special structure at low rank solutions*. We note again that though strict feasibility holds for the **SDP** formulation, we can take advantage of facial reduction and efficiently obtain low-rank solutions.

3.1 Preliminaries on faces

We now present some of the geometric facts we need. More details can be found in e.g., [5, 16, 18].

Suppose that $K \subseteq R^n$. Then K is a cone if $\lambda K \subseteq K, \forall \lambda \geq 0$. It is a proper closed convex cone, if it is a closed set and

$$K + K \subseteq K,\ \ \lambda K \subseteq K, \forall \lambda \geq 0,\ \ \text{int}\,(K) \neq \emptyset,\ \ K \cap (-K) = \{0\}.$$

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*, $F \unlhd K$, 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 $F \unlhd K$ is an *exposed face* if there exists $\phi \in K^*$ such that $F = \phi^\perp \cap K$; and ϕ is an *exposing vector*. Let S be a subset of the convex cone K, then face(S) is the smallest face of K containing S. 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 exposing vector for the intersection of exposed faces $F_i \unlhd K, i = 1, \ldots, k$, see [5],

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

For $K = \mathcal{S}_+^n$ 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

$$X = [P\ \ Q] \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} [P\ \ Q]^T$$

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

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

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

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

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 3.1 ([6, Theorem 4.1]) *Consider a linear transformation* $\mathcal{M}\colon \mathcal{S}^n \to \mathbb{R}^m$ *and a nonempty feasible set*

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

for some point $b \in \mathbb{R}^m$. *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*

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

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

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

$$v \textit{ exposes } N \iff \mathcal{M}^* v \textit{ exposes } \operatorname{face}(\mathcal{F}). \tag{3.1}$$

□

The result in (3.1) 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 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. We let Mat : $\mathbb{R}^{t(n)} \to \mathcal{S}^n$ denote the inverse mapping.

Corollary 3.2 *Suppose that* $1 < k < n$ *and* $\mathcal{M}$ *in Theorem 3.1 is the coordinate projection onto the leading principal submatrix of order* k, $m = t(k)$. *Let* $B \in \mathcal{S}^k_+$, $b = \operatorname{vec}(B) \in \mathbb{R}^{t(k)}$, *i.e., for* $X \in \mathcal{S}^n$, *we have*

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

Let

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

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

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

Proof The proof follows immediately from Theorem 3.1 as v exposes N and $Y = \mathcal{M}^* v$ is an exposing vector for face$(\mathcal{F})$. □

3.2 Structure at low rank solutions

The results in Sect. 2 can now be used to prove the following special structure at low rank feasible solutions. This structure is essential in our **FR** scheme.

Corollary 3.3 *Let* M^* *be optimal for the primal in* (2.3) *with* $\operatorname{rank}(M^*) = r_M$. *Let* $M^* = UDV^T$ *be the compact SVD,* $D \in \mathcal{S}^{r_Y}_{++}$. *Define*

$$W_1 = UDU^T, \quad W_2 = VDV^T, \tag{3.2}$$

and

$$Y = \begin{bmatrix} W_1 & M^* \\ (M^*)^T & W_2 \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} D \begin{bmatrix} U \\ V \end{bmatrix}^T. \tag{3.3}$$

Then we have $Y \succeq 0$ *and optimal in* (2.4) *with* $\operatorname{rank}(Y) =: r_Y = r_M$ *and*

$$\|M^*\|_* = \frac{1}{2}\operatorname{trace}(Y) = \operatorname{trace}(D).$$

Proof The matrices U, V have orthonormal columns. Therefore $\operatorname{trace}(Y) = 2\operatorname{trace}(D) = 2\|M\|_*$. □

Now suppose that there is a biclique α of G_Z and a corresponding *sampled submatrix*, $z[\alpha] \equiv X \in \mathbb{R}^{p\times q}$, of $Z \in \mathbb{R}^{m\times n}$, with $\operatorname{rank}(X) = r_X$. Without loss of generality, after row and column permutations if needed, we can assume that

$$Z = \begin{bmatrix} Z_1 & Z_2 \\ X & Z_3 \end{bmatrix}. \tag{3.4}$$

Let the **SVD** be

$$X = \begin{bmatrix} U_1 & U_X \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 & V_X \end{bmatrix}^T, \quad \Sigma \in \mathcal{S}^r_{++}; \tag{3.5}$$

and we have a full rank factorization $X = \bar{P}\bar{Q}^T$ obtained using the compact **SVD**

$$X = \bar{P}\bar{Q}^T = U_1\Sigma V_1^T, \quad \bar{P} = U_1\Sigma^{1/2}, \; \bar{Q} = V_1\Sigma^{1/2}.$$

We see below that such a desirable X (after a permutation if needed), that corresponds to a biclique $\alpha \in G_Z$, $z[\alpha] \equiv X$, and at least one nontrivial exposing vector, is characterized by

$$C_X = \{m-p+1, \ldots, m, m+1, \ldots, m+q\}, \qquad r \le \min\{p, q\} < \max\{p, q\}. \tag{3.6}$$

Here we use the *target rank*, r. We can exploit the information using these bicliques to obtain exposing vectors of the *optimal face*, F^*, i.e., the smallest face of $\mathcal{S}^{m+n}_+$ that contains the set of low rank solutions, see Lemma 3.4, below.

By abuse of notation, we can express any feasible Y, and so any optimal Y, that has the correct rank, as in (3.3), to get

$$0 \preceq Y = \begin{bmatrix} U \\ P \\ Q \\ V \end{bmatrix} D \begin{bmatrix} U \\ P \\ Q \\ V \end{bmatrix}^T = \left[\begin{array}{c|cc|c} UDU^T & UDP^T & UDQ^T & UDV^T \\ \hline PDU^T & PDP^T & PDQ^T & PDV^T \\ QDU^T & QDP^T & QDQ^T & QDV^T \\ \hline VDU^T & VDP^T & VDQ^T & VDV^T \end{array}\right], \quad D \succ 0. \tag{3.7}$$

We see that $X = PDQ^T = \bar{P}\bar{Q}^T$. Since we assume that X satisfies (3.6) and so is *big enough*, we conclude that generically $r_X = r_Y = r$, see Lemma 3.6 below, and that the ranges satisfy

$$\begin{aligned} \operatorname{Range}(X) &= \operatorname{Range}(P) = \operatorname{Range}(\bar{P}) = \operatorname{Range}(U_1), \\ \operatorname{Range}(X^T) &= \operatorname{Range}(Q) = \operatorname{Range}(\bar{Q}) = \operatorname{Range}(V_1). \end{aligned} \tag{3.8}$$

This is the key for facial reduction as we can use an *exposing vector* formed as $U_X U_X^T$ as well as $V_X V_X^T$, and then add them together to get a third exposing vector.

Lemma 3.4 (Basic **FR**) *Let* $X \in \mathbb{R}^{p\times q}$ *be a sampled submatrix of* Z*, and let* Z, X *be as in* (3.4) (*after a permutation if needed*) *with*

$$r_X := \operatorname{rank}(X) \le \min\{p, q\} < \max\{p, q\}.$$

Let X have a **SVD** *as in (3.5). We now add appropriate blocks of zeros to the block exposing vectors $U_X U_X^T$, $V_X V_X^T$ and get*

$$W_X = \left[\begin{array}{c|cc|c} 0 & 0 & 0 & 0 \\ \hline 0 & U_X U_X^T & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right] + \left[\begin{array}{c|cc|c} 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & V_X V_X^T & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right] = \left[\begin{array}{c|cc|c} 0 & 0 & 0 & 0 \\ \hline 0 & U_X U_X^T & 0 & 0 \\ 0 & 0 & V_X V_X^T & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right]. \tag{3.9}$$

Let Y be any **feasible** *solution of the primal problem in (2.4) with correct rank,* $\operatorname{rank}(Y) = r_X$*. Then all three matrices in (3.9) are exposing vectors for the minimal face containing Y. In particular, for W_X we have $0 \neq W_X \succeq 0$, $W_X Y = 0$. Moreover, if T is a full column rank matrix with the columns forming a basis for* $\operatorname{Null}(W_X)$*, the nullspace of W_X, then a facial reduction step for the minimal face containing all feasible Y with the correct rank, yields*

$$\operatorname{face}(Y) \trianglelefteq T \mathcal{S}_+^{(n+m)-(p+q-2r)} T^T.$$

Proof Since we have assumed that Y has the correct rank, we can use, for example the compact spectral decomposition, and get (3.7), $D \in \mathcal{S}_{++}^{r_X}$, with both P, Q having r_X columns. Since the rank of a product of matrices is at most the maximum of the ranks of the matrices, we see that $\operatorname{rank}(P) = \operatorname{rank}(Q) = r_X$ and the range equations in (3.8) hold by construction of the SVD. Therefore,

$$U_X^T P = 0,\ V_X^T Q = 0 \implies U_X U_X^T P D P^T = 0,\ V_X V_X^T Q D Q^T = 0,$$

i.e., $U_X U_X^T$, $V_X V_X^T$ provide exposing vectors as desired. We can now fill in with zeros and add to get the conclusion about the three exposing vectors in (3.9). Moreover, the block diagonal structure of the exposing vectors guarantees that the ranks add up to get the size of the smaller face. □

Example 3.5 (Pair of exposing vectors) We now present a matrix $Y \in \mathcal{S}_+^{11}$ with $\operatorname{rank}(Y) = 2$. Here $(m, n) = (6, 5)$.

$$Y = \begin{bmatrix} 0.0059877 & 0.10551 & -0.011994 & -0.036276 & -0.073807 & -0.049863 & -0.049795 & -0.02602 & 0.01314 & 0.022035 & -0.012187 \\ 0.10551 & 2.1638 & 0.035252 & -0.6439 & -1.5417 & -0.77074 & -1.9215 & -0.13496 & -0.23004 & 0.13318 & 0.239 \\ -0.011994 & 0.035252 & 0.22366 & 0.068878 & -0.04733 & 0.18725 & -0.74543 & 0.31405 & -0.39999 & -0.25065 & 0.39174 \\ -0.036276 & -0.6439 & 0.068878 & 0.21984 & 0.45085 & 0.30043 & 0.31772 & 0.15267 & -0.072518 & -0.12958 & 0.066865 \\ -0.073807 & -1.5417 & -0.04733 & 0.45085 & 1.1006 & 0.52923 & 1.4401 & 0.064661 & 0.20335 & -0.069711 & -0.2089 \\ -0.049863 & -0.77074 & 0.18725 & 0.30043 & 0.52923 & 0.45348 & 0.044817 & 0.33131 & -0.27295 & -0.27387 & 0.26224 \\ -0.049795 & -1.9215 & -0.74543 & 0.31772 & 1.4401 & 0.044817 & 3.9923 & -0.89251 & 1.4727 & 0.69104 & -1.4538 \\ -0.02602 & -0.13496 & 0.31405 & 0.15267 & 0.064661 & 0.33131 & -0.89251 & 0.45673 & -0.54736 & -0.3667 & 0.53491 \\ 0.01314 & -0.23004 & -0.39999 & -0.072518 & 0.20335 & -0.27295 & 1.4727 & -0.54736 & 0.72824 & 0.43489 & -0.71429 \\ 0.022035 & 0.13318 & -0.25065 & -0.12958 & -0.069711 & -0.27387 & 0.69104 & -0.3667 & 0.43489 & 0.29471 & -0.42483 \\ -0.012187 & 0.239 & 0.39174 & 0.066865 & -0.2089 & 0.26224 & -1.4538 & 0.53491 & -0.71429 & -0.42483 & 0.7007 \end{bmatrix}$$

We sample the elements in rows 4, 5, 6 and columns 7, 8, 9, 10 to obtain the $(p = 3) \times (q = 4)$ matrix X. We let U_X, V_X, denote orthonormal bases for the nullspaces of X, X^T, respectively, i.e.,

$$X U_X = 0, \quad X^T V_X = 0.$$

Then the two exposing vectors are $U_X U_X^T$ and $V_X V_X^T$, filled in with zeros. After adding them together, we get

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.81985 & -0.17015 & -0.34459 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.17015 & 0.035313 & 0.071516 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.34459 & 0.071516 & 0.14483 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.023237 & -0.058066 & -0.12587 & 0.059006 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.058066 & 0.57988 & 0.34589 & 0.34729 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.12587 & 0.34589 & 0.68409 & -0.28395 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.059006 & 0.34729 & -0.28395 & 0.71279 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

We see that

$$\|WY\| = 7.67638e{-}16,$$

thus verifying to 15 decimals that the sum of the two exposing vectors is indeed an exposing vector for face(Y).

We emphasize that here we knew the two principal diagonal blocks of Y that corresponded to the clique $C = \{4, 5, 6, 7, 8, 9, 10\}$. But in general we do not and only know the sampled X. However, generically (Lemma 3.6, below), we get the exposing vectors correctly as done here. Moreover, here we only had a single sampled X and could permute it to an easy position to illustrate the exposing vector. In general, we will have many of these that are identified by the indices determining the corresponding clique. We then add them up to get a final exposing vector which is used for the **FR** step.

3.3 Bicliques, weights and final exposing vector

Given a partial matrix $Z \in \mathbb{R}^{m\times n}$, we need to find nontrivial bicliques α and corresponding sampled submatrices $z[\alpha] = X$ according to the properties in (2.7) and (2.8). Intuitively, we may want to find bicliques with size as large as possible so that we can expose Y immediately. However, we do not want to spend a great deal of time finding large bicliques. Instead we find it is more efficient to find many medium-size bicliques, satisfying the size-rank condition $r \leq \min\{p, q\} < \max\{p, q\}$. This rank condition guarantees that at least one of the two exposing vectors found from the biclique is not zero. We can then add the exposing vectors obtained from the equivalent cliques for these bicliques to finally expose a small face containing the optimal Y. This is equivalent to dealing with a small number of large bicliques. This consideration also comes from the expense of the singular value decomposition for the sampled submatrix $z[\alpha] = X$ for U_X, V_X in (3.5) when the biclique is large.

The following lemma shows that, generically, we can restrict the search to bicliques corresponding to a sampled submatrix $X \in \mathbb{R}^{p\times q}$ that satisfies the rank condition $r \leq \min\{p, q\} < \max\{p, q\}$ without losing rank magnitude.

Lemma 3.6 (Generic rank property) *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. elements. Set $Z = Z_1 Z_2^T$ and let $X \in \mathbb{R}^{p\times q}$ be any submatrix of Z with* $\min\{p, q\} \geq r$. *Then* $\operatorname{rank}(X) = r$ *with probability* 1.

Proof Without loss of generality, we can assume that X is the top left corner of Z. Therefore, $X = \bar{Z}_1 \bar{Z}_2$ for appropriate (top part) submatrices $\bar{Z}_i$ of Z_i, $i = 1, 2$. By the rank condition, we have that $X = \bar{Z}_1 \bar{Z}_2^T$ is a full rank factorization of X generically. □

Remark 3.7 In our numerical tests we generate our matrices $Z = Z_1 Z_2^T$ as done in the above Lemma 3.6. Therefore, it clear that submatrices X with the specified size restriction have $\operatorname{rank}(X) = \operatorname{rank}(Z)$ generically. It is not clear if the converse is true, i.e., whether a given random matrix Z with $\operatorname{rank}(Z) = r$ and full rank factorization $Z = Z_1 Z_2^T$ implies that Z_1, Z_2 have random elements.[4]

With the existence of noise (e.g., Gaussian), we know that generically the X found can only have a higher rank but not a lower rank than r. In this case, since we assume that we know the target rank of X, we can adjust the exposing vector so that it will not over-expose

[4] The authors thank Dmitriy Drusvyatskiy for the simplification of our original proof of Lemma 3.6. Further discussions are given in [7].

the completion matrix. If the target rank is not known, then it can be estimated during the algorithm up to a given tolerance, i.e., for a give sampled $p \times q$ submatrix X we estimate the rank. If the estimated rank $r < \min\{p, q\}$, then by our (generic) Lemma 3.6, we can assume that we have found our target rank for Z. If this is not the case, then we need to look for bicliques of larger size. As soon as we find $r = \operatorname{rank}(X) < \min\{p, q\}$ then we have found our estimated target rank r.

After finding a biclique α corresponding to a sampled submatrix X and its full rank factorization $X = \bar{P}\bar{Q}^T$, we then construct *biclique weights* u_X^P and u_X^Q to measure how *noisy* the corresponding exposing vectors are. We essentially use the *Eckart–Young distance* [8] to the nearest matrix of rank r and include the size of the submatrix. If the problem is *noiseless* and we know the target rank for Z, then these distances for the submatrices are 0.

Definition 3.8 (Biclique noise) Suppose that $X \in \mathbb{R}^{p\times q}$, with singular values $\sigma_1 \geq \ldots \geq \sigma_{\min\{p,q\}}$, is a given sampled submatrix corresponding to a biclique of the graph of the partial matrix Z. Let r be the target rank. Define the *biclique noise*

$$u_X^P := \frac{\sum_{i=r+1}^{\min\{p,q\}} \sigma_i^2}{0.5p(p-1)}, \quad u_X^Q := \frac{\sum_{i=r+1}^{\min\{p,q\}} \sigma_i^2}{0.5q(q-1)}.$$

Definition 3.9 (Biclique weights) Let Θ be the set of all bicliques. For each biclique $X \in \Theta$ of the partial matrix Z, let p, q, u_X^P, u_X^Q be defined as in Definition 3.8. Let

$$S = \sum_{X\in\Theta} \left(u_X^P + u_X^Q\right).$$

Define the *biclique weight*

$$w_X^P = 1 - \frac{u_X^P}{S}, \qquad w_X^Q = 1 - \frac{u_X^Q}{S}.$$

Using Lemma 3.4, we now present Algorithm 3.1, page 13, to find an exposing vector Y_{expo} for the optimal face, i.e., we get the block diagonal

$$0 \neq \left[\begin{array}{c|c} \sum_{X\in\Theta} w_X^P U_X U_X^T & 0 \\ \hline 0 & \sum_{X\in\Theta} w_X^Q V_X V_X^T \end{array}\right] = Y_{expo} \succeq 0, \quad Y_{expo} Y^* = 0, \ \forall \text{ optimal } Y^*.$$

Note that if

$$Y_{expo} = \begin{bmatrix} U & V \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U & V \end{bmatrix}^T,$$

is the (orthogonal) spectral decomposition of Y_{expo}, with $\Lambda \in S_{++}^{r_e}$, then the optimal face satisfies

$$F^* \trianglelefteq V \mathcal{S}_+^{m+n-r_e} V^T, \quad V = \begin{bmatrix} V_P & 0 \\ 0 & V_Q \end{bmatrix}.$$

Thus this **FR** process reduces the size of the problem.

Remark 3.10 We do not need to look for large bicliques in Algorithm 3.1 since we can take advantage of the fact that adding exposing vectors results in an exposing vector. Moreover, finding a biclique is equivalent to finding a clique in G. Therefore, we use the algorithms for finding cliques given in [16] and [5, Algorithm 2].

Algorithm 3.1 finding the final exposing vector

1: **INPUT:** partial matrix $Z \in \mathcal{M}^{m\times n}$, target rank r;
2: **OUTPUT:** final *blocked exposing vector* Y_{expo} that exposes the optimal face for (2.5)
3: **PREPROCESSING:**
find a set of bicliques Θ of size within the given range {minsize, maxsize} with $r <$ minsize;
4: **for** each biclique $\alpha \in \Theta$ and corresponding $z[\alpha] = X$ **do**
5: $[U_X, V_X] \leftarrow$ from **SVD** of X in 3.5
6: $W_X^P \leftarrow U_X U_X^T$;
calculate biclique noise u_X^P;
7: $W_X^Q \leftarrow V_X V_X^T$;
calculate biclique noise u_X^Q;
8: **end for**
9: calculate all the biclique weights w_X^i, $i = P, Q, \alpha \in \Theta$, from biclique noise;
10: sum over bicliques the weighted blocked matrices filled in with appropriate zeros.

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

11: **return** Y_{expo}

4 Numerics

4.1 Noiseless case

In the noiseless case, the biclique noise is 0 and the weights are all 1 and so ignored. The **FR** step finds the *blocked* exposing vector Y_{expo} and the *blocked* basis for $\text{Null}(Y_{expo})$[5] given by the columns of

$$V = \begin{bmatrix} V_P & 0 \\ 0 & V_Q \end{bmatrix}, \quad V_P^T V_P = I_{r_p}, \quad V_Q^T V_Q = I_{r_q},$$

thus defining the dimensions $r_p + r_q = r_v$. Therefore an original feasible Y can be expressed as

$$Y = VRV^T = \begin{bmatrix} V_P R_p V_P^T & V_P R_{pq} V_Q^T \\ V_Q R_{pq}^T V_P^T & V_Q R_q V_Q^T \end{bmatrix} \tag{4.1}$$

where the blocked

$$R = \begin{bmatrix} R_p & R_{pq} \\ R_{pq}^T & R_q \end{bmatrix} \in \mathcal{S}^{r_v}, r_v < m + n.$$

This means the problems (2.4) and (2.5) are in general reduced to the much smaller dimension $\mathbb{R}^{r_p \times r_q}$. And if we find enough bicliques we expect a reduction to $r_p = r_q = r, r_v = 2r$, twice the target rank. If this is the case then we have exact recovery that can be obtained by a simple least squares solution. Otherwise, we have to rely on the **NNM** heuristic.

[5] The MATLAB command *null* was used to find an orthonormal basis for the nullspace. However, this requires an SVD decomposition and fails for huge problems. In that case, we used the Lanczos approach with *eigs*.

The reduced model for Y after **FR** with **NNM** is

$$\begin{array}{lll} \min & \operatorname{trace}(R) & (= \operatorname{trace}(VRV^T)) \\ \text{s.t.} & \mathcal{P}_{\bar{E}}(V_P R_{pq} V_Q^T) = z & (= \mathcal{P}_{\hat{E}}(Z)) \\ & R = \begin{bmatrix} R_p & R_{pq} \\ R_{pq}^T & R_q \end{bmatrix} \succeq 0. & \end{array} \tag{4.2}$$

The **FR** typically results in low values for r_p, r_q and in the exact data case *many* of the linear equality constraints become redundant, i.e., we generally end up with an overdetermined linear system. We use the compact QR decomposition[6] to identify which constraints to choose that result in a linearly independent set with a relatively low condition number. Thus we have eliminated a portion of the sampling and we get the linear system

$$\mathcal{M}(R_{pq}) := \mathcal{P}_{\tilde{E}}\left(V_P R_{pq} V_Q^T\right) = \tilde{z}, \text{ for some } \tilde{E} \subseteq \hat{E}, \tag{4.3}$$

and $\tilde{z}$ is the vector of corresponding elements in z.

1. We first solve the simple semidefinite constrained least squares problem

$$\min_{R \in \mathcal{S}_+^{r_v}} \left\| \mathcal{P}_{\tilde{E}}\left(V_P R_{pq} V_Q^T\right) - \tilde{z}) \right\|.$$

   If the optimal R has attained the target rank, then the exactness of the data implies that necessarily the optimal value is zero; and we are done. (In fact, the **SDP** constraint is redundant here as R can always be completed using an **SVD** decomposition of R_{pq}.)
2. If R does not have the target rank in Item 1 above, then we solve (4.2) for our minimum nuclear norm solution. We note that the linear transformation $\mathcal{M}$ in (4.3) is not one–one. Therefore, we often need to add a small regularizing term to the objective, i.e., we use $\min \operatorname{trace}(R) + \gamma \|R\|_F$ with small $\gamma > 0$.

4.1.1 Numerics noiseless case

We now present experiments with the algorithm on random noiseless instances. Averages (computer times, rank, residuals) on **twenty** random instances are included in the tables[7].

The tests were run with MATLAB version R2016a, and Windows 8, on a Dell Optiplex 9030, Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz and 16 GB RAM. [8] The times we present are the wall-clock times in seconds. For the semidefinite constrained least squares problems we used the MATLAB addon CVX [13] for simplicity. This means our computer times could be improved if we replaced CVX with a recent **SDP** solver.

We generate the instances as done in the recent work [9]. The target matrices are obtained from $Z = Z_L Z_R^T$, where $Z_L \in \mathbb{R}^{m \times r}$ and $Z_R \in \mathbb{R}^{r \times n}$. Each entry of the two matrices Z_L and Z_R is generated independently from a standard normal distribution $N(0, 1)$. We then generate a sparse $m \times r$ matrix to obtain the random indices that are sampled. We evaluate our results using the same measurement as in [9], which we call "Residual" in our tables. It

[6] The MATLAB economical version function $[\sim, R, E] = qr(\Phi, 0)$ finds the list of constraints for a well conditioned representation, where Φ denotes the matrix of constraints.

[7] The density p in the tables are reported as "mean(p)" because the real density obtained is usually not the same as the one set for generating the problem. We report the mean of the real densities over the five instances.

[8] The Tables 4 with rank 6 and 5 with rank 8 were done using a MacBookPro12,1, Intel Core i5, 2.7 GHz with two cores and 8 GB RAM. The version of MATLAB was the same R2016a.

is calculated as:

$$\text{Residual} = \frac{\|\hat{Z} - Z\|_F}{\|Z\|_F},$$

where Z is the target matrix, $\hat{Z}$ is the output matrix that we find, and $\|\cdot\|_F$ is the Frobenius norm.

We observe that we far outperform the results in [9] both in accuracy and in time; and we solve much larger problems. We are not as competitive for the low density problems as our method requires a sufficient number of cliques in G (bicliques in G_Z). We could combine our preprocessing approach using the bicliques before the method in [9] is applied.

In Tables 1, 2, 3, 4, 5, 6, 7 we present the results with noiseless data with target ranks ranging from $r = 2$ to $r = 6$. We see that we get efficient *high* accuracy recovery in *every* instance. The accuracy is significantly higher than what one can expect from an **SDP** interior point solver. The computer time is almost entirely spent on finding the matrix representation and on its QR factorization that is used as a heuristic for finding a correct subset of well-conditioned linear constraints. However, we do not use any refinement steps for these tests. For higher rank and sparse problems we end up with a larger **FR** problem and a a large matrix representation. This can be handled using the sketch matrix and refinement described in the noisy case. For the lower density problems, we remove the rows and columns of the original data matrix corresponding to zero diagonal elements of the final exposing matrix. These rows and columns have no sampled entries in them and so it does not make sense to include them in the algorithm. We include the percentage of the number of elements of the original data matrix that are recovered and the corresponding percentage residual. Since the accuracy is high for this recovered submatrix, it can then be used with further sampling to recover the complete original matrix.

These problems involved relatively low target ranks $r = 2$ to $r = 8$. Larger ranks mean that we need to find larger bicliques/cliques, e.g., $r = 20$ means that the cliques need to be of size bigger than 40. This means that the values for r_p, r_q can be large and we need to solve a large **SDP** least squares problem. We include a purify step to do this in the noisy case discussed below.

Note that the largest problems in the last of the noiseless Tables 6 and 7, have, respectively, 48,000,000 and 50,000,000 data entries in Z with approximately 35,000,000 unknown values that were recovered successfully with **extremely** high accuracy. The target rank was recovered in every instance. We used the MATLAB command *null* in Table 6 to find the nullspaces to derive V in (4.1). This is based on an **SVD** decomposition of a full matrix and is expensive. We used MATLAB *eigs* rather than *null* in Table 7 which resulted in lower computer times but lower accuracy. We could not use *null* in the noisy case as this results in essentially full rank each time due to the noise. We changed to a sparse QR decomposition which estimates the rank, has the lowest computer times while still maintaining high accuracy.

Though we have not made a comprehensive comparison with results in the literature, our results compare well with e.g., those in [24]. We obtain a significant increase in accuracy and speed of solution.

4.2 Noisy case

This case is similar to the noiseless case but with the addition of a refinement step. (The refinement step can also be used for the noiseless case when the **FR** problem dimension r_v is too large.) We include the rank and residual outputs for both before refinement and the total of both after refinement. We see that in most cases when the graph is sufficiently dense,

Table 1 *Noiseless*: $r = 2$; $m \times n$ size; density p; mean 20 instances

Specifications			r_v	Rcvrd (%Z)	Time (s)	Rank	Residual (%Z)
m	n	Mean (p)					
2100	4000	0.33	4.00	100.00	46.35	2.0	1.4298e−13
2100	4000	0.26	4.00	100.00	44.69	2.0	4.3546e−14
2100	4000	0.22	4.00	100.00	43.43	2.0	9.8758e−14
2100	4000	0.18	4.00	100.00	42.66	2.0	1.4409e−13
2100	4000	0.14	4.00	99.78	42.16	2.0	8.9667e−14

Table 2 *Noiseless*: $r = 3$; $m \times n$ size; density p; mean 20 instances

Specifications			r_v	Rcvrd (%Z)	Time (s)	Rank	Residual (%Z)
m	n	Mean (p)					
2100	4000	0.33	6.00	100.00	50.46	3.0	8.6855e−13
2100	4000	0.26	6.00	100.00	49.88	3.0	1.0738e−12
2100	4000	0.22	6.00	100.00	48.56	3.0	1.1436e−12
2100	4000	0.18	6.00	99.81	47.90	3.0	2.5695e−12
2100	4000	0.14	6.20	95.15	46.69	3.0	8.5525e−12

Table 3 *Noiseless*: $r = 5$; $m \times n$ size; density p; mean 20 instances

Specifications			r_v	Rcvrd (%Z)	Time (s)	Rank	Residual (%Z)
m	n	Mean (p)					
2100	4000	0.45	10.00	100.00	52.48	5.0	2.2232e−10
2100	4000	0.42	10.00	100.00	53.16	5.0	2.3748e−11
2100	4000	0.39	10.00	100.00	52.45	5.0	1.5950e−10
2100	4000	0.36	10.00	99.99	49.78	5.0	4.5280e−11
2100	4000	0.33	10.00	99.79	47.60	5.0	2.5057e−10

Table 4 *Noiseless*: $r = 6$; $m \times n$ size; density p; mean 20 instances

Specifications			r_v	Rcvrd (%Z)	Time (s)	Rank	Residual (%Z)
m	n	Mean (p)					
2100	4000	0.48	12.00	100.00	84.83	6.0	4.4311e−10
2100	4000	0.45	12.00	99.98	78.81	6.0	7.2856e−10
2100	4000	0.42	12.00	99.78	76.11	6.0	1.3813e−11
2100	4000	0.39	12.00	98.46	73.48	6.0	2.8688e−10
2100	4000	0.36	13.65	92.08	74.52	6.0	5.6545e−08

Table 5 *Noiseless*: $r = 8$; $m \times n$ size; density p; mean 20 instances

Specifications			r_v	Rcvrd (%Z)	Time (s)	Rank	Residual (%Z)
m	n	Mean (p)					
1000	3000	0.53	16.10	96.39	37.29	8.0	1.1072e−10
1000	3000	0.50	17.65	88.99	36.50	8.0	4.6569e−10
1000	3000	0.48	32.15	71.66	72.14	8.5	2.0413e−07

Table 6 *Noiseless*: $r = 3$; $m \times n$ size; density p; mean 20 instances

Specifications			r_v	Rcvrd (%Z)	Time (s)	Rank	Residual (%Z)
m	n	Mean (p)					
700	2000	0.33	6.00	100.00	5.58	3.0	2.6857e−13
1000	5000	0.33	6.00	100.00	58.31	3.0	3.0256e−12
1400	9000	0.33	6.00	100.00	296.91	3.0	1.4185e−12
1900	14000	0.33	6.00	100.00	1043.46	3.0	1.9995e−12
3000	16000	0.33	6.00	100.00	1758.76	3.0	2.5250e−12

Table 7 *Noiseless*: $r = 4$; 100% recovered; nullspace with eigs; mean 5 instances

Specifications			Time (s)	Rank	Residual (%Z)
m	n	Mean (p)			
700	2000	0.36	12.80	4.0	1.5217e−12
1000	5000	0.36	49.66	4.0	1.0910e−12
1400	9000	0.36	131.53	4.0	6.0304e−13
1900	14000	0.36	291.22	4.0	3.4847e−11
2500	20000	0.36	798.70	4.0	7.2256e−08

refinement is *not* needed, and near perfect completion (recovery) is obtained relative to the noise. In particular, the low target rank was attained most times.

We generate the data as in the noiseless case and then perturb the known entries by additive noise, i.e.,

$$Z_{ij} \leftarrow Z_{ij} + \sigma \xi_t \|Z\|_\infty, \quad \forall ij \in \bar{E},$$

where $\xi_t \sim N(0, 1)$ and σ is a noise factor that can be changed. The computer and software were similar as in the noiseless case. The tests were run on MATLAB version R2016a as above, but on a Dell Optiplex 9030, with Windows 8, Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz and 16 GB RAM.

As above we proceed to first complete **FR** in order to reduce the dimension of Y, i.e., the dimension of R, r_v, is dramatically smaller. In the low density and/or high rank cases it is difficult to find enough cliques and in this case the final exposing vector Y_{expo} contains many zero rows. This essentially means that we have not sampled rows and/or columns of Z. In these cases we have ignored the rows and columns that used no sampled entries.

After **FR** we first solve the simple semidefinite constrained least squares problem

$$\delta_0 = \min_{R \in \mathcal{S}_+^{rv}} \left\| \mathcal{P}_{\hat{E}} \left(V_P R_{pq} V_Q^T \right) - z \right\|, \quad z = \mathcal{P}_{\hat{E}}(Z).$$

However, unlike in the noiseless case, we cannot remove redundant constraints, even though there may be many. This problem is now highly overdetermined and may also be ill-posed in that the constraint transformation may not be one-one. We use the notion of *sketch matrix* to reduce the size of the system, e.g., [19]. The matrix A is a random matrix of appropriate size with a relatively small number of rows in order to dramatically decrease the size of the constrained least squares problem

$$\delta_0 = \min_{R \in \mathcal{S}_+^{rv}} \left\| A \left(\mathcal{P}_{\hat{E}} \left(V_P R_{pq} V_Q^T \right) - z \right) \right\|.$$

As noted in [19], this leads to surprisingly good results. If s is the dimension of R, then we use a random sketch matrix of size $2t(s) \times |\hat{E}|$, where $t(\cdot)$ is the number of variables on and above the diagonal of a symmetric matrix, i.e., the triangular number

$$t(s) = \frac{s(s+1)}{2}.$$

If the optimal R has the correct target rank, then we are done.

4.2.1 Refinement step with dual multiplier

If the result from the constrained least squares problem does not have the target rank, we now use this δ_0 as a best target value for our parametric approach as done in [5]. Our **NNM** problem can be stated as:

$$\begin{array}{lrcl} \min & & \text{trace}(R) & \\ \text{s.t.} & \left\| A \left(\mathcal{P}_{\hat{E}} \left(V_P R_{pq} V_Q^T \right) - z \right) \right\| & \leq & \delta_0 \\ & R & \succeq & 0. \end{array} \tag{4.4}$$

To attempt to find a lower rank solution, we use the approach in [5] and *flip* this problem:

$$\begin{array}{rl} \varphi(\tau) := \min & \left\| A \left(\mathcal{P}_{\hat{E}} \left(V_P R_{pq} V_Q^T \right) - z \right) \right\| + \gamma \|R\|_F \\ \text{s.t.} & \text{trace}(R) \leq \tau \\ & R \succeq 0. \end{array} \tag{4.5}$$

As in the noiseless case, the least squares problem may be underdetermined. We add a regularizing term $+\gamma \|R\|_F$ to the objective with $\gamma > 0$ small. The starting value of τ is obtained from the unconstrained least squares problem, and from which we can reduce the value of the trace of R to reduce the nuclear norm and so heuristically reduce the rank. We refer to this process as the refinement step.

This process requires a tradeoff between low-rank and low-error. Specifically, the trace constraint may not be tight at the starting value of τ, which means we can lower the trace of R without sacrificing accuracy, however, if the trace is pushed lower than necessary, the error starts to get larger. To detect the balance point between low-rank and low-error, we exploit the role as sensitivity coefficient for the *dual multiplier* of the inequality constraint. The value of the dual variable indicates the rate of increase of the objective function. When the the dual multiplier becomes positive then we know that decreasing τ further will increase the residual value. We have used the value of .01 to indicate that we should stop decreasing τ.

Table 8 *Noisy*: $r = 2$; $m \times n$ size; density p; mean 20 instances

Specifications				Rcvd (%Z)	Time (s)		Rank		Residual (%Z)	
m	n	% noise	p		Initial	Refine	Initial	Refine	Initial	Refine
1100	3000	0.50	0.33	100.00	33.72	48.53	2.00	2.00	8.53e−03	8.53e−03
1100	3000	1.00	0.33	100.00	33.67	49.09	2.00	2.00	2.70e−02	2.70e−02
1100	3000	2.00	0.33	100.00	34.13	48.84	2.00	2.00	9.75e−02	9.75e−02
1100	3000	3.00	0.33	100.00	36.34	92.73	5.00	5.00	5.48e−01	1.40e−01
1100	3000	4.00	0.33	100.00	51.45	186.28	11.00	8.00	1.25e+00	1.28e−01

Table 9 *Noisy*: $r = 3$; $m \times n$ size; density p; mean 20 instances

Specifications				Rcvd (%Z)	Time (s)		Rank		Residual (%Z)	
m	n	% noise	p		Initial	Refine	Initial	Refine	Initial	Refine
700	1000	1.00	0.33	99.99	2.58	16.54	3.35	3.35	1.29e+00	1.07e+00
800	2000	1.00	0.33	100.00	10.72	29.59	3.75	3.75	1.15e+00	1.07e+00
900	4000	1.00	0.33	100.00	61.92	94.40	3.25	3.20	1.47e+00	1.07e+00
1000	8000	1.00	0.33	100.00	404.26	672.60	8.70	6.45	3.94e+00	7.11e−01
1100	16000	1.00	0.33	100.00	3553.81	4230.73	9.00	6.65	4.00e+00	6.66e−01

Table 10 *Noisy*: $r = 4$; $m \times n$ size; density p; mean 20 instances

Specifications				Rcvd (%Z)	Time (s)		Rank		Residual (%Z)	
m	n	% noise	p		Initial	Refine	Initial	Refine	Initial	Refine
1100	3000	0.00	0.36	100.00	30.27	42.44	4.00	4.00	9.04e−13	9.04e−13
1200	3500	1.00	0.33	100.00	52.48	198.22	8.20	6.70	6.45e+00	1.08e+00
1300	4000	2.00	0.32	100.00	81.09	388.68	11.80	7.85	1.88e+01	1.28e+00
1400	4500	3.00	0.31	100.00	117.40	573.87	12.00	7.40	2.51e+01	1.45e+00
1500	5000	4.00	0.31	100.00	142.86	699.06	12.00	6.90	2.42e+01	1.61e+00

4.2.2 Numerics noisy case

The noisy case results with increasing ranks 2, 3, 4 and various sizes and densities follow in Tables 8, 9, 10. With the densities we use the recovery is essentially 100%. We consider problems with relatively high density to ensure that we can find enough cliques. We have not included tests with higher rank as those are done in the noiseless case and are similar here.

4.3 Comparison with direct NNM

We conclude with Table 11 that compares our approach with **FR** against using CVX and minimizing the nuclear norm directly.[9] We clearly see that **FR** consistently yields significant improvements with obtaining lower rank, higher accuracy in the residual, and efficiency in

[9] We used CVX version 2.1 with the MOSEK solver, e.g., [1].

Table 11 *Noiseless*: $r = 2$; $m \times n$ size; density p; mean 20 instances

Specifications			FR result					CVX result		
m	n	mean(p)	r_v	Rcvrd (%Z)	Time (s)	Rank	Residual (%Z)	Time (s)	Rank	Residual (%Z)
60	100	0.33	4.00	93.73	0.24	2.0	2.1061e−13	5.14	2.1	4.9836e−08
60	100	0.26	4.55	79.89	0.25	2.0	1.9728e−12	2.87	2.5	4.0214e−08
60	100	0.22	6.00	63.64	0.23	2.1	1.8306e−11	2.33	7.0	3.7404e−08
60	100	0.18	9.55	50.86	0.28	3.2	1.9193e−10	1.87	19.8	3.5576e−08
60	100	0.14	21.35	31.15	0.40	7.7	7.6125e−11	1.23	18.0	2.9111e−08

time. This emphasizes that our **FR** approach does more than exploit the structure of the **NNM** model but actually improves on this model.

5 Conclusion

In this paper we have shown that we can apply facial reduction with an exposing vector approach used in [5] in combination with the structure at low rank solutions of the semidefinite embedding to efficiently find low-rank matrix completions. This exploits the singular structure of the optimal solution set of the minimum rank completion problem even though the feasible set itself satisfies strict feasibility.

Specifically, whenever enough complete bipartite subgraphs are available for the graph of the matrix of the problem, we are able to find a proper face with a *significantly reduced dimension* that contains the optimal solution set of minimum rank. We then solve this smaller minimum trace problem by *flipping* the problem and using a refinement with a parametric point approach. 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 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, i.e., without need for solving a nuclear norm minimization problem.

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 thus exploiting the block structure of the cliques; and with solving the lower dimensional flipped problems. In our paper we have started our tests with knowing the target rank r. In forthcoming tests we plan on estimating the target rank using sampled submatrices. Our tests illustrate that the facial reduction approach significantly improves on just relying on the nuclear norm relaxation.

In addition, theoretical results on *exact recovery* are discussed in many papers, e.g., [3, 4, 20]. They use the so-called *restricted isometry property*, **RIP**, for vectors extended to the matrix case. However, the **RIP** condition is difficult to verify. It appears from our work above that exact recovery guarantees can be guaranteed from rigidity questions 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.

Acknowledgements The authors thank Nathan Krislock for his help with parts of the MATLAB coding and also thank Fei Wang for useful discussions related to the facial reduction steps. We also thank two anonymous referees for carefully reading the paper and helping us improve the details and the presentation.

References

1. Andersen, E.D., Andersen, K.D.: The Mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In: High performance optimization, vol. 33 of Applied Optimization, pp. 197–232. Kluwer Academic Publishers, Dordrecht (2000)
2. Blanchard, J.D., Tanner, J., Wei, K.: CGIHT: conjugate gradient iterative hard thresholding for compressed sensing and matrix completion. Inf. Inference **4**(4), 289–327 (2015)
3. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. Found. Comput. Math. **9**(6), 717–772 (2009)
4. Candès, E.J., Tao, T.: The power of convex relaxation: near-optimal matrix completion. IEEE Trans. Inf. Theory **56**(5), 2053–2080 (2010)
5. Drusvyatskiy, D., Krislock, N., Voronin, Y.-L.C. Wolkowicz, H.: Noisy Euclidean distance realization: robust facial reduction and the Pareto frontier. SIAM J. Optim. **27**(4), 2301–2331 (2017). arXiv:1410.6852. (to appear)
6. Drusvyatskiy, D., Pataki, G., Wolkowicz, H.: Coordinate shadows of semidefinite and Euclidean distance matrices. SIAM J. Optim. **25**(2), 1160–1178 (2015)
7. Drusvyatskiy, D., Wolkowicz, H.: The many faces of degeneracy in conic optimization. Found. Trends Optim. **3**(2), 77–170 (2016). https://doi.org/10.1561/2400000011
8. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrica **1**(3), 211–218 (1936)
9. Fang, E.X., Liu, H., Toh, K.-C., Zhou, W.-X.: Max-norm optimization for robust matrix recovery. Technical report, Department of Operations Research and Financial Engineering, Princeton University, Princeton (2015)
10. Fazel, M.: Matrix rank minimization with applications. Ph.D. thesis, Stanford University, Stanford, CA (2001)
11. Fazel, M., Hindi, H., Boyd, S.P.: A rank minimization heuristic with application to minimum order system approximation. In: Proceedings American Control Conference, pp. 4734–4739 (2001)
12. Gauvin, J.: A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming. Math. Program. **12**(1), 136–138 (1977)
13. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 1.21 (2011). http://cvxr.com/cvx
14. Hardt, M., Meka, R., Raghavendra, P., Weitz, B.: Computational limits for matrix completion. In: JMLR Proceedings, pp. 1–23. JMLR (2014)
15. Király, F.J., Theran, L., Tomioka, R.: The algebraic combinatorial approach for low-rank matrix completion. J. Mach. Learn. Res. **16**, 1391–1436 (2015)
16. Krislock, N., Wolkowicz, H.: Explicit sensor network localization using semidefinite representations and facial reductions. SIAM J. Optim. **20**(5), 2679–2708 (2010)
17. Meka, R., Jain, P., Dhillon, I.S.: Guaranteed rank minimization via singular value projection. Technical report advanced studies in pure mathematics, Department of Computer Science, University of Texas at Austin, Austin (2009)
18. Pataki, G.: Geometry of semidefinite programming. In: Wolkowicz, H., Saigal, R., Vandenberghe, L. (eds.) Handbook of Semidefinite Programming: Theory, Algorithms, and Applications. Kluwer Academic Publishers, Boston, MA (2000)
19. Pilanci, M., Wainwright, M.J.: Randomized sketches of convex programs with sharp guarantees. IEEE Trans. Inf. Theory **61**(9), 5096–5115 (2015)
20. Recht, B., Fazel, M., Parrilo, P.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev. **52**(3), 471–501 (2010)
21. Singer, A., Cucuringu, M.: Uniqueness of low-rank matrix completion by rigidity theory. SIAM J. Matrix Anal. Appl. **31**(4), 1621–1641 (2009/10)
22. Srebro, N.: Learning with matrix factorizations. (Ph.D.) Thesis, ProQuest LLC, Ann Arbor, MI. Massachusetts Institute of Technology (2004)
23. Srebro, N., Shraibman, A.: Rank, trace-norm and max-norm. In: Learning theory, vol. 3559 of Lecture Notes in Computer Science, pp. 545–560. Springer, Berlin (2005)

24. Tanner, J., Wei, K.: Low rank matrix completion by alternating steepest descent methods. Appl. Comput. Harmon. Anal. **40**(2), 417–429 (2016)
25. Wolkowicz, H.: Tutorial: facial reduction in cone optimization with applications to matrix completions. In: Dimacs workshop on distance geometry: theory and applications. Based on survey paper: the many faces of degeneracy in conic optimization, (with D. Drusvyatskiy) (2016)