Low-Rank Matrix Completion using Nuclear Norm Minimization and Facial Reduction[*]

Shimeng Huang [†]   Henry Wolkowicz[‡]

Wednesday 18th January, 2017, 14:03, REVISION 1

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 at optimality for the **NNM**. We show that even though strict feasibility holds, the facial reduction framework used for problems where strict feasibility fails can be successfully applied to obtain a proper face that contains the optimal set. This can dramatically reduce the size of the final **NNM** problem, while simultaneously guaranteeing a low-rank solution. This can be compared to identifying the active set in general nonlinear programming problems.

We include numerical tests for both exact and noisy cases.

Keywords: Low-rank matrix completion, matrix recovery, semidefinite programming (**SDP**), facial reduction, cliques, Slater condition, nuclear norm, compressed sensing.

AMS subject classifications: 65J22, 90C22, 65K10, 52A41, 90C46

Contents

1 Introduction **3**

[*] Presented as Part of tutorial at DIMACS Workshop on Distance Geometry: Theory and Applications, July 26-29, 2016, [22]

[†] 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.

1.1 Outline 3

2 Background on LRMC, NNM, SDP 3
2.1 Models 4
2.2 Graph Representation of the Problem 5

3 Facial Reduction, Cliques, Exposing Vectors 7
3.1 Preliminaries on faces 7
3.2 Structure at Optimum 8
3.3 Cliques, Weights and Final Exposing Vector 11

4 Noiseless Case 13
4.1 Numerics Noiseless Case 14

5 Noisy Case 16
5.1 Refinement Step with Dual Multiplier 18
5.2 Numerics Noisy Case 19

6 Conclusion 19

Acknowledgment 20

Index 21

Bibliography 22

List of Tables

4.1 noiseless: $r = 2$; $m \times n$ size; density p; mean 20 instances. 16
4.2 noiseless: $r = 3$; $m \times n$ size; density p; mean 20 instances. 16
4.3 noiseless: $r = 5$; $m \times n$ size; density p; mean 20 instances. 16
4.4 noiseless: $r = 6$; $m \times n$ size; density p; mean 20 instances. 16
4.5 noiseless: $r = 8$; $m \times n$ size; density p; mean 20 instances. 17
4.6 noiseless: $r = 3$; $m \times n$ size; density p; mean 20 instances. 17
4.7 noiseless: $r = 4$; 100% recovered; nullspace with eigs; mean 5 instances. 17
5.1 noisy: $r = 2$; $m \times n$ size; density p; mean 20 instances. 19
5.2 noisy: $r = 3$; $m \times n$ size; density p; mean 20 instances. 19
5.3 noisy: $r = 4$; $m \times n$ size; density p; mean 20 instances. 19

List of Algorithms

3.1 (finding the final exposing vector) 13

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 be then 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 the optimum* and by using the *exposing vector* approach, see [3]. The exposing vector approach is particularly amenable to the noisy case. Moreover, from **FR** we get a significant reduction in the size of the variables and a corresponding decrease in the possible rank of the solution. 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., [16].

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., [19, 20]. See also the recent work in [1, 17, 21] 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 Compressive Sensing Resources, http://dsp.rice.edu/cs.

The convex relaxation of minimizing the rank using the nuclear norm, the sum of the singular values, is studied in e.g., [7, 17]. The solutions can be found directly by subgradient methods or by using **SDP** with interior point methods or low-rank methods, again see [17]. Many other methods have been developed, e.g., [14]. The two main approaches for rank minimization, convex relaxations and spectral methods, are discussed in [2, 12] along with a new algebraic combinatorial approach. A related analysis from a different viewpoint using rigidity in graphs is provided in [18].

1.1 Outline

We continue in Section 2 with the basic notions for **LRMC** using the nuclear norm and with the graph framework that we employ. Then in Section 3 we include preliminaries on cone facial structure and the details on how to exploit **FR**, for the **SDP** model to minimize the nuclear norm problem. The main result for the reduction is in Lemma 3.1.

The results for the noiseless case are given in Section 4. This includes an outline of the basic approach in Algorithm 3.1 and empirical results from randomly generated problems. The noisy case follows in Section 5 with empirical results and a comparison with results in [21]. Concluding remarks are given in Section 6.

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., [17]. 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:

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

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 $b = \mathcal{P}_{\hat{E}}(Z)$ 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., [11].

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

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

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

$$\begin{array}{ll} \min_M & \|M\|_* \\ \text{s.t.} & \mathcal{A}(M) = b \end{array} \qquad\qquad \begin{array}{ll} \max_z & \langle b, z\rangle \\ \text{s.t.} & \|\mathcal{A}^*(z)\| \leq 1, \end{array} \tag{2.3}$$?eq:pdpairnucln

where $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^t$ is a linear mapping, $b \in \mathbb{R}^t$, $\mathcal{A}^*$ 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 there exists $\hat{M}$ with $\mathcal{A}(\hat{M}) = b$. 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 [17, 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 $M = \hat{M}, z = 0$. □

Corollary 2.1. *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 the primal satisfies the *Mangasarian-Fromovitz constraint qualification*; while $z = 0$ shows that the dual satisfies strict feasibility. It is well known that this constraint qualification is equivalent to the dual problem having a nonempty, convex, compact optimal set, e.g., [9]. □

[1] Note that the linear mapping $\mathcal{A} = \mathcal{P}_{\hat{E}}$ 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.

The following proposition shows that 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 2.2. *The optimal primal-dual solution set in* (2.3) *is the same as that in the* **SDP** *primal-dual pair:*

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

?eq:pdpairnucln

□

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

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

?sdpnuclear?

where $\bar{E}$ is the set of indices in Y that correspond to $\hat{E}$, the known entries of the upper right block of $\begin{bmatrix} 0 & Z \\ Z^T & 0 \end{bmatrix} \in \mathcal{S}^{m+n}$.

When the data is contaminated with noise, we reformulate the strict 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.,

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

?sdpnuclearinex

We emphasize that since 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.

2.2 Graph Representation of the Problem

Our sampling yields elements $b = \mathcal{P}_{\hat{E}}(Z)$. 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 apply the clique algorithms from [3, 13].

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

$$\big\{\{ij \in V \times V : i < j \le m\} \cup \{ij \in V \times V : m+1 \le i < j \le m+n\}\big\} \subseteq E \subseteq \{ij \in V \times V : i < j\},$$

and *weights* for all $ij \in E$

$$W_{ij} = \begin{cases} Z_{i(j-m)}, & \forall ij \in \bar{E} \\ 0, & \forall ij \in E \backslash \bar{E}. \end{cases}$$

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

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

We associate a biclique in G_Z with a clique in G in the obvious way. Each results in a sampled submatrix X.

We can now construct the *adjacency matrix*, A, for the graph G as follows

$$A_{ij} = \left\{ \begin{array}{ll} 1 & \text{if } ij \in E \text{ or } ji \in E \\ 0 & \text{otherwise.} \end{array} \right. \tag{2.7}$$

?eq:Aadj?

Recall that a *clique* in the graph G is a complete subgraph in G. We have the 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. The nontrivial cliques of interest correspond to (possibly after row and column permutations) a full (sampled) submatrix X in Z, i.e., to bicliques in G_Z. The cliques of interest are $C = \{i_1, \ldots, i_k\}$ with cardinalities

$$|C \cap \{1, \ldots, m\}| = p \neq 0, \quad |C \cap \{m+1, \ldots, m+n\}| = q \neq 0. \tag{2.8}$$

?eq:cardspq?

The submatrix of Z for the corresponding biclique from the clique C is

$$X \equiv \{Z_{i(j-m)} : ij \in C\}, \quad \text{sampled } p \times q \text{ rectangular submatrix.} \tag{2.9}$$

?eq:Xspecif?

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

Example 2.1 (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 $\mathcal{P}_{\hat{E}}(Z)$ denotes a vector representation of the known entries.

Suppose that our algorithm found a biclique with indices

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

The corresponding submatrix is

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

The sampled large matrix Y containing the sampled Z is filled in with ones on the diagonal blocks to emphasize that these blocks are unknown during the algorithm. Then the clique C_X corresponding to the biclique and the corresponding principal submatrix corresponding to X are, respectively,

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

and

$$\begin{bmatrix} 1 & 1 & 1 & 3 & 8 & -2 & 4 \\ 1 & 1 & 1 & -5 & -20 & 10 & -6 \\ 1 & 1 & 1 & 4 & 4 & 4 & 6 \\ 3 & -5 & 4 & 1 & 1 & 1 & 1 \\ 8 & -20 & 4 & 1 & 1 & 1 & 1 \\ -2 & 10 & 4 & 1 & 1 & 1 & 1 \\ 4 & -6 & 6 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

3 Facial Reduction, Cliques, Exposing Vectors

In this section we look at the details of solving the **SDP** formulation of the nuclear norm relaxation for **LRMC**. In particular we show how to exploit cliques in the graph G and the *special structure at the optimum.* 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., [3,13,15].

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

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

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*, $F \trianglelefteq K$, 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 $F^* = F^\perp \cap K^*$. 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 S be a subset of the convex cone K, then $\operatorname{face}(S)$ 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 intersection of exposed faces $F_i \trianglelefteq K, i = 1, \ldots, k$, see [3],

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

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

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

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

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

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

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

i.e. the face and the exposing vectors are characterized by the eigenspace of any $\hat{X}$ in the relative interior of the face. Moreover, if $\bar{X} \in \operatorname{face}(X)$ and the columns of V are a basis for $\operatorname{Null}(\bar{X})$, then

$$\operatorname{face}(X) \subseteq \mathcal{S}_+^n \cap (VV^T)^\perp, \tag{3.1}$$

?eq:exposnonuni

i.e., we obtain an exposing vector using the nullspace of any matrix $\bar{X}$ in $\operatorname{face}(X)$.

3.2 Structure at Optimum

The results in Section 2 can now be used to prove the following special structure at the optimum. This structure is essential in our **FR** scheme.

Corollary 3.1. *Let M^* be optimal for the primal in* (2.4) *with* $\operatorname{rank}(M^*) = r_M$. *Then there exist variables W_1, W_2, z to complete the primal-dual pair for* (2.4) *such that the compact spectral decomposition of the corresponding optimal Y in* (2.4) *can be written as*

$$0 \preceq Y = \begin{bmatrix} W_1 & M^* \\ (M^*)^T & W_2 \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} D \begin{bmatrix} U \\ V \end{bmatrix}^T, \quad D \in \mathcal{S}_{++}^{r_Y}, \operatorname{rank}(Y) =: r_Y = r_M. \tag{3.2}$$

?eq:YDUV?

We get

$$W_1 = UDU^T, \quad W_2 = VDV^T, \quad M^* = UDV^T, \qquad \|M^*\|_* = \frac{1}{2}\operatorname{trace}(Y) = \frac{1}{2}\operatorname{trace}(D). \tag{3.3}$$

?eq:Winorm?

Proof. Let $M^* = U_M \Sigma_M V_M^T$ be the compact *singular value decomposition,* **SVD**, with $\Sigma_M \in \mathcal{S}_{++}^{r_M}$ on the diagonal. Let

$$D = 2\Sigma_M, \quad U = \frac{1}{\sqrt{2}} U_M,\, V = \frac{1}{\sqrt{2}} V_M, \qquad Y = \begin{bmatrix} U \\ V \end{bmatrix} D \begin{bmatrix} U \\ V \end{bmatrix}^T.$$

Then the matrix $\begin{bmatrix} U \\ V \end{bmatrix}$ has orthonormal columns and $\operatorname{trace} Y = 2 \operatorname{trace}(\Sigma_M) = 2\|M\|_*$. Therefore (3.3) holds. Since Y is now primal optimal and Slater's condition holds for the primal problem, there must exist z optimal for the dual. □

Now suppose that there is a biclique of G_Z and a corresponding *sampled submatrix,* $X \in \mathbb{R}^{p \times q}$, 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

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

?eq:ZX?

Let the **SVD** be

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

?eq:svdX?

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

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

We see below that such a desirable X (after a permutation if needed), that corresponds to a clique C_X in G and at least one nontrivial exposing vector, is characterized by

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

?eq:suffbig?

Here we use the *target rank, r*. We can exploit the information using these cliques to obtain exposing vectors of the *optimal face, F^**, i.e., the smallest face of $\mathcal{S}_+^{m+n}$ that contains the set of optimal solutions of (2.4).

By abuse of notation, we can rewrite the optimality form in (3.2) as

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

?eq:Ypartit?

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 3.2 below, and that the ranges satisfy

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

?eq:PQbar?

This is the key for facial reduction as we can use an *exposing vector* formed as $U_X U_X^T$ as well as $V_X V_X^T$.

Lemma 3.1 (basic **FR**)**.** *Let Y be an optimal solution of the primal problem in* (2.4) *with* $\operatorname{rank}(Y) = r = r_Z$*, i.e., the* **NNM** *heuristic yields an optimal Y that successful solves the* **LRMC** *problem. Let $X \in \mathbb{R}^{p \times q}$ be a sampled submatrix of Z be as in* (3.4) *(after a permutation if needed) with* $\operatorname{rank}(X) = r_X = r_Z \leq \min\{p, q\} < \max\{p, q\}$*, and with* **SVD** *as in* (3.5)*. By abuse of notation, suppose that both matrices $U_X \leftarrow U_X U_X^T, V_X \leftarrow V_X V_X^T$ are filled out with zeros above and below so their size is that of Y and let $W_X := U_X + V_X$.*

$$W_X = \left[\begin{array}{c|ccc|c} 0 & 0 & & 0 & 0 \\ \hline 0 & U_X & & 0 & 0 \\ 0 & 0 & & 0 & 0 \\ \hline 0 & 0 & & 0 & 0 \end{array}\right] + \left[\begin{array}{c|ccc|c} 0 & 0 & & 0 & 0 \\ \hline 0 & 0 & & 0 & 0 \\ 0 & 0 & & V_X & 0 \\ \hline 0 & 0 & & 0 & 0 \end{array}\right] = \left[\begin{array}{c|ccc|c} 0 & 0 & & 0 & 0 \\ \hline 0 & U_X & & 0 & 0 \\ 0 & 0 & & V_X & 0 \\ \hline 0 & 0 & & 0 & 0 \end{array}\right].$$

Then U_X, V_X, W_X are all exposing vectors for the optimal face, i.e., 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 optimal face, F^*, the minimal face containing the optimal set, yields*

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

Proof. That U_X, V_X provide exposing vectors is by construction based on the argument for (3.8). The result follows from the fact that the sum of exposing vectors is an exposing vector. Moreover, the block diagonal structure of the exposing vectors guarantees that the ranks add up to get the size of the smaller face containing F^*. □

Example 3.1 (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 3.2, 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 Cliques, Weights and Final Exposing Vector

Given a partial matrix $Z \in \mathbb{R}^{m\times n}$, we need to find nontrivial cliques and corresponding sampled submatrices X according to the properties in (2.8) and (2.9). Intuitively, we may want to find cliques 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 cliques. Instead we find it is more efficient to find many medium-size cliques, satisfying the size-rank condition $r \leq \min\{p,q\} < \max\{p,q\}$. We can then add the exposing vectors obtained from these cliques to finally expose a small face containing the optimal Y. This is equivalent to dealing with a small number of large cliques. This consideration also comes from the expense of the singular value decomposition for the sampled submatrix X for U_X, V_X in (3.5) when the biclique is large.

The cliques are found through using the *adjacency matrix* defined above in (2.7). We can then use these cliques to find a set of exposing vectors. Specifically, we can obtain at most two useful exposing vectors from each of the cliques we found. Exposing vectors are useful only if they are nonzero. To get a nonzero exposing vector we need the sizes of the sampled submatrix X from the bicliques to be sufficiently large, i.e., a useful exposing vector requires that the diagonal block formed from one of the full rank decomposed parts of this clique has a *target rank* and appropriate size for the rank. The target rank and size of the diagonal block depend on the size and rank of the submatrix X. In particular, we want at least one of U_X, V_X in (3.5) to be nonzero, in which case, we say the clique is useful. We illustrate this in detail in Algorithm 3.1.

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 3.2 (generic rank property). *Let $Z \in \mathbb{R}^{m\times n}$ be a continuous random variable with i.i.d. elements. Suppose that* $\operatorname{rank}(Z) = r$ *and $X \in \mathbb{R}^{p\times q}$ is a sampled submatrix corresponding to a biclique obtained from Z with* $\min\{p,q\} \geq r$. *Then* $\operatorname{rank}(X) = r$ *with probability* 1 *(generically), i.e., the conditional probability*

$$P\big(\operatorname{rank}(X) = r | \operatorname{rank}(Z) = r\big) = 1.$$

Proof. First we note that the trivial case $r = 0$ is clear. Now, recall that the rank is a lower semi-continuous function on $\mathbb{R}^{m\times n}$. Therefore, arbitrary small perturbations can increase the rank but not decrease it. The result now follows since the rank of a submatrix is bounded above by r.

Alternatively, without loss of generality, we can assume that $X \in \mathbb{R}^{p\times r}$, with $p \geq r$. If $\operatorname{rank}(X) < r$, then there exists $\alpha \in \mathbb{R}^r, \|\alpha\|_1 = 1$, such that $y = X\alpha = 0$. The first element of this vector as a function of α is

$$y_1(\alpha) = e_1^T X\alpha = X_{(1,:)}\alpha,$$

where e_1 is the first unit vector. Since the row vector $X_{(1,:)}$ comes from a continuous distribution then so does y_1. Therefore the probabilities

$$P\big(\operatorname{rank}(X) < r | \operatorname{rank}(Z) = r\big) \leq P\big(y_1 = 0 | \operatorname{rank}(Z) = r\big) = 0.$$

The conclusion follows. □

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 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 each sampled submatrix X we estimate and increase the target rank $r = \operatorname{rank}(X)$ if X is full rank. As soon as we find $r = \operatorname{rank}(X) < \min\{p, q\}$ then we have found our estimated target rank r.

After finding a clique corresponding to a sampled submatrix X and its full rank factorization $X = \bar{P}\bar{Q}^T$, we then construct *clique weights* u_X^P and u_X^Q to measure how *noisy* the corresponding exposing vectors are. We essentially use the *Eckart-Young distance* [5] 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 3.1 (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 3.2 (biclique weights)**.** *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 3.1. Let*

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

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 3.1, 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 $\Lambda \in \mathbb{S}^{r_e}_{++}$, 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.

Algorithm 3.1 (finding the final exposing vector)

```
INPUT: partial matrix $Z \in \mathcal{M}^{m\times n}$, target rank $r$, clique size range $\{minsize, maxsize\}$;
OUTPUT: final blocked exposing vector $Y_{expo}$ that exposes the optimal face for (2.5)
PREPROCESSING:
    1. form the corresponding adjacency matrix $A$;
    2. find a set of cliques $\Theta$ from $A$ of size within the given range;
for each clique $C_X \in \Theta$ and corresponding $X$ do
    $[U_X, V_X] \leftarrow$ from SVD of $X$ in 3.5    (nullspaces)
    $W_X^P \leftarrow U_X U_X^T$;
    calculate clique noise $u_X^P$;
    $W_X^Q \leftarrow V_X V_X^T$;
    calculate clique noise $u_X^Q$;
end for
calculate all the clique weights $w_X^i, i = P, Q, X \in \Theta$, from clique noise;
sum over weights using nullspaces filled in with appropriate zeros.

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

return $Y_{expo}$
```

4 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 $\text{Null}(Y_{expo})$[2] given by the columns of

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

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

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

?eq:VRVT?

[2] 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*.

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 (2.5) and (2.4) are in general reduced to the much smaller dimension $\mathbb{R}^{r_p \times r_q}$. And if we find enough cliques 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.

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

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

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 underdetermined linear system. We use the compact QR decomposition[3] 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

$$\mathcal{M}(R_{pq}) := \mathcal{P}_{\tilde{E}}(V_P R_{pq} V_Q^T) = \tilde{b}, \text{ for some } \tilde{E} \subseteq \hat{E}, \qquad (4.3)$$?eq:smallsyst?

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

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

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

   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 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[4].

The tests were run with MATLAB version R2016a, and Windows 8, on a Dell Optiplex 9030, Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz and 16 GB RAM. [5] The times we present are the

[3] 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.

[4] 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.

[5] The Tables 4.4 with rank 6 and 4.5 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.

wall-clock times in seconds. For the semidefinite constrained least squares problems we used the MATLAB addon CVX [10] 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 [6]. 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 [6], which we call "Residual" in our tables. It 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 [6] 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 cliques before the method in [6] is applied.

In Tables 4.1 to 4.7 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, 4.6 and 4.7, 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 4.6 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 4.7 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.

Table 4.1: 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 4.2: 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 4.3: 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 4.4: 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

5 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, refinement is *not*

Table 4.5: 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 4.6: 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 4.7: 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

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 @ 3.60GHz 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

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

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., [16]. 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

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

As noted in [16], 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.

5.1 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 [3]. Our **NNM** problem can be stated as:

$$\begin{array}{lrcl} \min & & \operatorname{trace}(R) & \\ \text{s.t.} & \left\| A\left(\mathcal{P}_{\hat{E}}(V_P R_{pq} V_Q^T) - b\right)\right\| & \leq & \delta_0 \\ & R & \succeq & 0. \end{array} \tag{5.1}$$

?frsdpnuclear?

To attempt to find a lower rank solution, we use the approach in [3] and *flip* this problem:

$$\begin{array}{ll} \varphi(\tau) := \min & \left\| A\left(\mathcal{P}_{\hat{E}}(V_P R_{pq} V_Q^T) - b\right)\right\| + \gamma \|R\|_F \\ \text{s.t.} & \operatorname{trace}(R) \leq \tau \\ & R \succeq 0. \end{array} \tag{5.2}$$

?eq:flipNoisyfi

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.

5.2 Numerics Noisy Case

The noisy case results with increasing ranks $2, 3, 4$ and various sizes and densities follow in Tables 5.1, 5.2, 5.3. 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.

Table 5.1: 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 5.2: 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 5.3: 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

6 Conclusion

In this paper we have shown that we can apply facial reduction through the exposing vector approach used in [3] in combination with the nuclear norm heuristic to efficiently find low-rank matrix completions. This exploits the degenerate structure of the optimal solution set even though the nuclear norm heuristic problem 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. 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 cliques, the matrix can still be partially completed. Having an insufficient number of cliques is indicative of not having enough initial data to recover the unknown elements. Throughout we see that the facial reduction both regularizes the problem and reduces the size and often allows for a solution without any refinement.

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 using heuristics to estimate this rank and then reduce the target rank during the algorithm.

In addition, theoretical results on *exact recovery* are discussed in many papers, e.g., [2, 17]. 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.

Acknowledgment

The authors would like to thank Nathan Krislock for his help with parts of the MATLAB coding; and we would also like to thank two anonymous referees for carefully reading the paper and helping us improve the details and the presentation.

Index

A, adjacency matrix, 6
$C_X = \{m-p+1,\dots,m,m+1,\dots,m+q\}$, 9
E, edge set, 5
$F \trianglelefteq K$, face, 7
F^*, optimal face, 10
$G=(V,E,w)$, weighted undirected graph, 5
$G_Z=(U_m,V_n,\hat{E})$, 5
K^*, dual cone, 7
P-biclique weight, w_X^P, 12
R_p, 13
R_q, 13
R_{pq}, 13
V, node set, 5
$X \in \mathbb{R}^{p\times q}$, sampled submatrix, 9
$Y \succeq 0$, positive semidefinite, 5
Y_{expo}, 13
$\mathcal{S}^n$, space of real symmetric matrices, 5
$\mathcal{S}_+^n$, positive semidefinite matrices, 5
$\mathcal{S}_{++}^n$, positive definite matrices, 5
$\bar{E}$, 5
$\hat{E}$, 4
$\|\cdot\|_*$, nuclear norm, 4
$\|\cdot\|_F$, Frobenius norm, 15
e_1, first unit vector, 12
r, target rank, 9
r_X, 9
r_Y, 8
r_p, 13
r_q, 13
u_X^P, biclique noise, 12
u_X^P, clique weight, 12
w_X^P, P-biclique weight, 12
FR, facial reduction, 3
LRMC, low-rank matrix completion problem, 3, 4
NNM, nuclear norm minimization, 4
RIP, restricted isometry property, 20
SDP, semidefinite programming, 3
SVD, singular value decomposition, 9

adjacency matrix, 11
adjacency matrix, A, 6
adjoint of $\mathcal{A}$, 4
biclique, 5
biclique noise, u_X^P, 12

clique, 6
clique weight, u_X^P, 12
conjugate face, F^*, 8

dual cone, K^*, 7
dual multiplier, 18

Eckart-Young distance, 12
edge set E, 5
exposed face, 8
exposing vector, 3, 8, 9

face, $F \trianglelefteq K$, 7
x **FR**, 3
first unit vector, e_1, 12
Frobenius norm, $\|\cdot\|_F$, 15

low-rank matrix completion problem, **LRMC**, 4
x **LRMC**, 3

Mangasarian-Fromovitz constraint qualification, 4

node set, 5
x **NNM**, 4
nuclear norm, $\|\cdot\|_*$, 4

optimal face, F^*, 9, 10

partial matrix, 3
positive definite matrices, $\mathcal{S}_{++}^n$, 5
positive semidefinite matrices, $\mathcal{S}_+^n$, 5

x **RIP**, 20

sampled submatrix, $X \in \mathbb{R}^{p\times q}$, 9
x **SDP**, 3
x **SVD**, 9
sketch matrix, 3, 18
Slater constraint qualification, 5
space of real symmetric matrices, $\mathcal{S}^n$, 5

target rank, r, 9

weighted undirected graph, $G=(V,E,W)$, 5
weights, 6

References

MR3436116 [1] J.D. Blanchard, J. Tanner, and K. Wei. CGIHT: conjugate gradient iterative hard thresholding for compressed sensing and matrix completion. *Inf. Inference*, 4(4):289–327, 2015. 3

MR2723472 [2] E.J. Candès and T. Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Inform. Theory*, 56(5):2053–2080, 2010. 3, 20

ChDrWo:14 [3] D. Drusvyatskiy, N. Krislock, Y-L. Cheung Voronin, and H. Wolkowicz. Noisy sensor network localization: robust facial reduction and the Pareto frontier. Technical report, University of Waterloo, Waterloo, Ontario, 2014. arXiv:1410.6852, 20 pages. 3, 5, 7, 8, 18, 19

?DrPaWo:14? [4] D. Drusvyatskiy, G. Pataki, and H. Wolkowicz. Coordinate shadows of semidefinite and Euclidean distance matrices. *SIAM J. Optim.*, 25(2):1160–1178, 2015.

EckartYoung:36 [5] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrica*, 1(3):211–218, 1936. 12

angLiuTohZhou:15 [6] E.X. Fang, H. Liu, K.-C. Toh, and W.-X. Zhou. Max-norm optimization for robust matrix recovery. Technical report, Department of Operations Research and Financial Engineering, Princeton University, Princeton, 2015. 15

Fazel:02 [7] M. Fazel. *Matrix Rank Minimization with Applications.* PhD thesis, Stanford University, Stanford, CA, 2001. 3

?Fia:83? [8] A.V. Fiacco. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, volume 165 of *Mathematics in Science and Engineering.* Academic Press, 1983.

MR0489903 [9] J. Gauvin. A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming. *Math. Programming*, 12(1):136–138, 1977. 4

cvxunpubl [10] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. `http://cvxr.com/cvx`, April 2011. 15

ekaRaghWeigz:14 [11] M. Hardt, R. Meka, P. Raghavendra, and B. Weitz. Computational limits for matrix completion. In *JMLR Proceedings*, pages 1–23. JMLR, 2014. 4

MR3417786 [12] F.J. Király, L. Theran, and R. Tomioka. The algebraic combinatorial approach for low-rank matrix completion. *J. Mach. Learn. Res.*, 16:1391–1436, 2015. 3

kriswolk:09 [13] N. Krislock and H. Wolkowicz. Explicit sensor network localization using semidefinite representations and facial reductions. *SIAM Journal on Optimization*, 20(5):2679–2708, 2010. 5, 7

aJainDhillon:09 [14] R. Meka, P. Jain, and I.S. Dhillon. 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. 3

PatakiSVW:99 [15] 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

ciWainwright:15 [16] M. Pilanci and M.J. Wainwright. Randomized sketches of convex programs with sharp guarantees. *IEEE Trans. Infor. Theory*, 61(9):5096–5115, 2015. 3, 18

chtparrilofazel [17] 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, 4, 20

MR2595541 [18] A. Singer and M. Cucuringu. Uniqueness of low-rank matrix completion by rigidity theory. *SIAM J. Matrix Anal. Appl.*, 31(4):1621–1641, 2009/10. 3

MR2717223 [19] N. Srebro. *Learning with matrix factorizations.* ProQuest LLC, Ann Arbor, MI, 2004. Thesis (Ph.D.)–Massachusetts Institute of Technology. 3

MR2203286 [20] N. Srebro and A. Shraibman. Rank, trace-norm and max-norm. In *Learning theory*, volume 3559 of *Lecture Notes in Comput. Sci.*, pages 545–560. Springer, Berlin, 2005. 3

MR3440180 [21] J. Tanner and K. Wei. Low rank matrix completion by alternating steepest descent methods. *Appl. Comput. Harmon. Anal.*, 40(2):417–429, 2016. 3

ktalkdimacs:16 [22] H. Wolkowicz. Tutorial: Facial reduction in cone optimization with applications to matrix completions, at: Dimacs workshop on distance geometry: Theory and applications, 2016. Based on survey paper: The many faces of degeneracy in conic optimization, (with D. Drusvyatskiy). 1