Low-Rank Matrix Completion using Nuclear Norm with Facial Reduction*

Shimeng Huang † Henry Wolkowicz‡

Sunday 14th August, 2016, 17:43

Abstract

Minimization of the nuclear norm 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***). The ***SDP*** and its dual are regular in the sense that they both satisfy strict feasibility. Interior point algorithms are the current methods of choice for these problems. This means that it is difficult to solve large scale problems and difficult to get high accuracy solutions.

In this paper we take advantage of the structure at optimality for the minimum nuclear norm problem. We show that even though strict feasibility holds, the facial reduction framework can be successfully applied to obtain a proper face that contains the optimal set, and thus can dramatically reduce the size of the final nuclear norm problem while guaranteeing a low-rank solution. We include numerical tests for both exact and noisy cases. In all cases we assume that knowledge of a *target rank* is available.

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

*Presented as Part of tutorial at DIMACS Workshop on Distance Geometry: Theory and Applications, July 26-29, 2016, [15]

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

‡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 and by AFOSR;

List of Tables

List of Algorithms

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 feasibilty, 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 vectors* approach, see [3] The exposing vector approach is particularly amenable to the noisy case. Moreover, the result from facial reduction is a significant reduction in the size of the variables and a decrease in the 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., [11].

The problem of low-rank matrix completion has many applications to model reduction, sensor network localization, pattern recognition and machine learning. This problem is further related to real applications in data science, for instance, the collaborative filtering (the well known Netflix problem) and multi-tasking learning. See e.g., the recent work in [1, 14] 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., [6, 12]. The solutions can be found directly by subgradient methods or by using semidefinite programming ***SDP*** with interior point methods or low-rank methods, again see [12]. Many other methods have been developed, e.g., [10]. The two main approaches for rank minimization, convex relaxations and spectral methods, are discussed in [2, 9] along with a new algebraic combinatorial approach. A related analysis from a different viewpoint using rigidity in graphs is provided in [13].

We continue in Section 2 with the basic notions for ***LRMC*** using the nuclear norm and with the graph framework that we employ. We continue in Section 3 with the details on how to exploit *facial reduction* ***FR***, for the ***SDP*** model to minimize the nuclear norm problem. Section 4 presents the details for finding the low-rank solution after the ***FR*** has been completed. We present the numerical results in Section 5 and a comparison with results in [14]. Concluding remarks are included in Section 6.

2 Background on *LRMC*, Nuclear Norm Minimization, *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 known results we follow and include much of the known development in the literature e.g., [12, Prop. 2.1]. 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 *partial* $m \times n$ *real matrix* $Z \in \mathbb{R}^{m\times n}$ which has precise data. The *low-rank matrix completion problem* ***LRMC***, can be modeled as follows:

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

where $\hat{E}$ is the set of indices containing the known 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.

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

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

?basicnuclear?

where the nuclear norm $\|\cdot\|_*$ is the sum of the singular values, i.e., $\|M\|_* = \sum_i \sigma_i(M)$.

Moreover, we consider the primal-dual pair of problems for the *nuclear norm* minimization problem:

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

?eq:pdpairnucln

where $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^t$ is a linear mapping, $\mathcal{A}^*$ 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*.[1] In general, the linear equality constraint is an underdetermined linear system. In our case, we restrict to the case that $\mathcal{A} = \mathcal{P}_{\hat{E}}$.

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 [12, 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., [7]. □

The following Proposition shows that, we can embed the problem into an $\boldsymbol{SDP}$ and solve it efficiently.

Proposition 2.2. *The pair in* (2.3) *are equivalent to the following $\boldsymbol{SDP}$ primal-dual pair:*

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

?eq:pdpairnucln

□

[1] Note that $\mathcal{A}$ 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.

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

$$\begin{array}{ll} \min & \|Y\|_* = \mathrm{trace}(Y) \\ \text{s.t.} & \mathcal{P}_{\bar{E}}(Y) = b \\ & Y \succeq 0, \end{array} \tag{2.5}$$?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 $\mathcal{Z} = \begin{bmatrix} 0 & Z \\ Z^T & 0 \end{bmatrix} \in \mathcal{S}^{m+n}$. Here $Y \succeq 0$ denotes the Löwner partial order that Y is *positive semidefinite,* $Y \in \mathcal{S}_+^{m+n}$.

When the data are 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.,

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

We emphasize that since there is no constraint on the diagonal blocks of Y, 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

For our needs, we furthermore 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

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

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

$$A_{ij} = \begin{cases} 1 & \text{if } ij \in E \text{ or } ji \in E \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$?eq:Aadj?

Recall that a *clique* in the graph G is a complete subgraph in G. We have the trivial cliques $C = \{i_1, \ldots, i_k\} \subset \{1, \ldots, m\}$ and $C = \{j_1, \ldots, j_k\} \subset \{m+1, \ldots, m+n\}$, which are not of interest to our algorithm. The nontrivial cliques of interest correspond to (possibly after row and column permutations) a full (specified) submatrix X in 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?

This means that we have found

$$X \equiv \{Z_{i(j-m)} : ij \in C\}, \quad \text{specified (fully known) } p \times q \text{ rectangular matrix.} \tag{2.9}$$?eq:Xspecif?

These non-trivial cliques are at the center of our considerations.

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 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*

$$0 \preceq Y = \begin{bmatrix} W_1 & M^* \\ (M^*)^T & W_2 \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} D \begin{bmatrix} U \\ V \end{bmatrix}^T = \begin{bmatrix} UDU^T & UDV^T \\ VDU^T & VDV^T \end{bmatrix}, \quad D \in \mathcal{S}^{r_Z}_{++}, \operatorname{rank} Y =: r_Y = r_M. \tag{3.1}$$

?eq:YDUV?

We get

$$W_1 = UDU^T, \quad W_2 = VDV^T, \quad M^* = UDV^T, \qquad \|M^*\|_* = \frac{1}{2}\operatorname{trace}(Y) = \frac{1}{2}\operatorname{trace}(D). \tag{3.2}$$

?eq:Winorm?

Proof. Let $M^* = U_M \Sigma_M V_M^T$ be the compact 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 (3.2) 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 *specified submatrix,* $X \in \mathbb{R}^{p \times q}$, of $Z \in \mathbb{R}^{m \times n}, \operatorname{rank}(X) = r_X$. Without loss of generality, after row and column permutations if needed, we can assume that

$$Z = \begin{bmatrix} Z_1 & Z_2 \\ X & Z_3 \end{bmatrix},$$

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

$$X = \bar{P}\bar{Q}^T = U_X D_X V_X^T, \, D_Z \in \mathcal{S}^{r_X}_{++}, \quad \bar{P} = U_X D_X^{1/2}, \, \bar{Q} = V_X D_X^{1/2}.$$

Note that a desirable X that corresponds to a clique in G is given by

$$C_X = \{i, \ldots, m, m+1, \ldots, m+k\}, \qquad r < \max\{p, q\},$$

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

By abuse of notation, we can rewrite the optimality form in (3.1) as

$$0 \preceq Y = \begin{bmatrix} U \\ P \\ Q \\ V \end{bmatrix} D \begin{bmatrix} U \\ P \\ Q \\ V \end{bmatrix}^T = \left[\begin{array}{c|ccc} UDU^T & UDP^T & UDQ^T & UDV^T \\ \hline PDU^T & PDP^T & PDQ^T & PDV^T \\ QDU^T & QDP^T & QDQ^T & QDV^T \\ \hline VDU^T & VDP^T & VDQ^T & VDV^T \end{array}\right]. \tag{3.3}$$?eq:Ypartit?

We see that $X = PDQ^T = \bar{P}\bar{Q}^T$. Since X is *big enough*, we conclude that generically $r_X = r_Y = r$, see Lemma 3.2 below, and that the ranges satisfy

$$\mathcal{R}(X) = \mathcal{R}(P) = \mathcal{R}(\bar{P}), \qquad \mathcal{R}(X^T) = \mathcal{R}(Q) = \mathcal{R}(\bar{Q}). \tag{3.4}$$?eq:PQbar?

This is the key for facial reduction as we can use an *exposing vector* formed from $\bar{P}$ and/or $\bar{Q}$.

Lemma 3.1 (Basic ***FR***). *Let $r < \min\{p, q\}$ and let* (3.3), (3.4) *hold with $X = PDQ^T = \bar{P}\bar{Q}^T$, found using the full rank factorization. Let Y be an optimal solution of the primal problem in* (2.4). *Define $(\bar{U}, \bar{V}) = \boldsymbol{FR}(\bar{P}, \bar{Q})$ by*

$$\boldsymbol{FR}(\bar{P}, \bar{Q}): \quad \bar{P}\bar{P}^T + \bar{U}\bar{U}^T \succ 0, \quad \bar{P}^T\bar{U} = 0, \qquad \bar{Q}\bar{Q}^T + \bar{V}\bar{V}^T \succ 0, \quad \bar{Q}^T\bar{V} = 0. \tag{3.5}$$?eq:qpuv?

By abuse of notation, suppose that both matrices $\bar{U} \leftarrow \bar{U}\bar{U}^T, \bar{V} \leftarrow \bar{V}\bar{V}^T$ are filled out with zeros above and below so their size is that of Y and let $W = \bar{U} + \bar{V}$. Then $\bar{U}, \bar{V}, W$ are all exposing vectors for the optimal face, i.e., for W we have $W \succeq 0, WY = 0$. Moreover, if T is a full column rank matrix with the columns forming a basis for $\mathcal{N}(W)$, the null space of W, then a facial reduction step for the optimal face is the substitution

$$Y = TRT^T,\ R \in \mathcal{S}_+^{(n+m)-(p+q-2r)}.$$

Proof. That $\bar{U}, \bar{V}$ are exposing vectors is by construction. The result follows from the fact that the sum of exposing vectors is an exposing vector. Moreover, the block diagonal structure of the exposing matrices guarantees that the ranks add up to get the size of R. (More details are available in [3, 4].) □

3.2 Cliques, Weights and Final Exposing Vector

Given a partial matrix $Z \in \mathbb{R}^{m\times n}$, we need to find nontrivial cliques according to the definition 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 that can cover as many vertices as possible. 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 expensive computational cost of the eigenvalue calculation for $\bar{U}, \bar{V}$ in (3.5) when the clique 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 sampling matrix 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 *correct rank* and correct size. The correct 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 $\bar{U}, \bar{V}$ 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 cliques corresponding to a specified submatrix $X \in \mathbb{R}^{p\times q}$ such that $\min\{p,q\} \geq r$ without losing rank magnitude, where p and q are defined in (2.8). This means if either $p > r$ or $q > r$, we can obtain a useful exposing vector based on that part of the clique.

Lemma 3.2. *Let $Z \in \mathbb{R}^{m\times n}$ be a random matrix where the entries come from a continuous distribution. Suppose that* $\operatorname{rank}(Z) = r$ *and $X \in \mathbb{R}^{p\times q}$ is a specified partial matrix obtained from Z with* $\min\{p,q\} \geq r$. *Then* $\operatorname{rank}(X) = r$ *with probability* 1 *(generically).*

In terms of the above notation, let

$$Y, Z, r_Y, r_Z, r_X, p, q,$$

be defined as above with $r_Z \leq \min\{p,q\}$. Then generically

$$\operatorname{rank}\left(\left[PDP^T\right]\right) = \operatorname{rank}\left(\left[QDQ^T\right]\right) = r_Z = r_X.$$

Proof. Recall that the $\operatorname{rank} : \mathbb{R}^{m\times n} \to \mathbb{N}$ is a lower semi-continuous function. 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.

More precisely, without loss of generality, we can suppose that $X = [x^1, ..., x^r]$, where $x^i \in \mathbb{R}^p$ are the column vectors of X, with $p \geq r$. If $\operatorname{rank}(X) < r$, then there exists $\{a_1, ..., a_r\} \subset \mathbb{R}$ such that $y = \sum_{i=1}^r a_i x^i = 0$.

The first element of this vector is $y_1 = \sum_{i=1}^r a_i x_1^i$. Since x_1^i comes from a continuous distribution then so does y_1, and the probability $\mathrm{P}(\operatorname{rank}(X) < r) \leq \mathrm{P}(y_1 = 0) = 0$. Thus, it must be that $\operatorname{rank}(X) = r$ for X of the appropriate size given in the lemma. □

With the existence of noise, 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 know the correct rank of X, we can adjust the exposing vector so that it will not over-expose the completion matrix.

After finding a clique corresponding to a sampled submatrix and its full rank factorization $X = \bar{P}\bar{Q}^T$, we then construct a sized *clique weight,* u_X^i, to measure how *noisy* the corresponding exposing vector is. We essentially use the *Eckart-Young distance* to the nearest matrix of rank r on the semidefinite cone and include the size. If the problem is *noiseless*, then generically we expect this distance to be 0, since submatrices of sufficient size yield either $\bar{P}\bar{P}^T$ or $\bar{Q}\bar{Q}^T$ to be rank r.

Definition 3.1 (clique weights)**.** *Let $X = \bar{P}\bar{Q}^T$ denote a $p \times q$ sampled submatrix with its full rank factorization. If $p > r$ let $B = \bar{P}\bar{P}^T = UDU^T$ be the spectral decomposition with eigenvalues $\lambda_j, j = 1, \ldots, p$ in nondecreasing order. Define the clique weight, u_X^i, with $i = p$*

$$u_X^p := \frac{\sum_{i=1}^{p-r} \lambda_i^2 + \sum_{i=p-r+1}^{p} (\min\{0, \lambda_i\})^2}{0.5p(p-1)}.$$

If $q > r$, repeat with $\bar{Q}$ and q.

Definition 3.2 (exposed vector weights)**.** *Define the exposed vector weight,* w_X^i*, as*

$$w_X^i = 1 - \frac{u_X^i}{\textit{sum of all existing clique weights}}, \quad i = p, q.$$

Algorithm 3.1 summarizes how to find an exposing vector Y_{expo} for our optimal Y for the minimum nuclear norm problem. This exposing vector locates a face containing Y which is

$$F_Y = V\mathcal{S}_+^{r_v}V^T$$

where V is from the spectral decomposition of Y_{expo}

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

such that $\Sigma \succ 0$. In other words, $\boldsymbol{FR}$ yields a representation of Y as

$$Y = VRV^T, \quad \text{for some} \quad R \in \mathcal{S}_+^{r_v}, \tag{3.6}$$?eq:VRVt?

where we hope that we have found enough cliques to get the reduction $r_v = r$. We now aim to find an appropriate R.

4 Dimension Reduction and Refinement

After $\boldsymbol{FR}$ the original Y can be expressed as $Y = VRV^T$, where $R \in \mathcal{S}^{r_v}$ and $V^TV = I$. This means the problems (2.5) and (2.4) are in general reduced to the much smaller dimension r_v. And if we find enough cliques we expect a reduction to $r_v = r$, the target rank.

4.1 Noiseless Case

The expression of Y after $\boldsymbol{FR}$ means we now turn to solve the nuclear norm minimization problem

$$\begin{array}{ll} \min & \operatorname{trace}(R) \quad (= \operatorname{trace}(VRV^T)) \\ \text{s.t.} & \mathcal{P}_{\bar{E}}(VRV^T) = b \\ & R \succeq 0, \end{array} \tag{4.1}$$?frnuclearnoise

where $b = \mathcal{P}_{\hat{E}}(Z)$. The $\boldsymbol{FR}$ typically results in many of the linear equality constraints becoming redundant. We use the compact QR decomposition[2] 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

$$\mathcal{M}(R) := \mathcal{P}_{\tilde{E}}(VRV^T) = \tilde{b}, \text{ for some } \tilde{E} \subseteq \bar{E}, \tag{4.2}$$?eq:smallsyst?

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

[2]We use $[\sim, R, E] = qr(\Phi, 0)$ to find the list of constraint for a well conditioned representation, where Φ denotes the matrix of constraints.

Algorithm 3.1 Finding Exposing Vectors

INPUT: A partial matrix $Z \in \mathcal{M}^{m\times n}$, target rank r, clique size range $\{minsize, maxsize\}$;
OUTPUT: A final exposing vector that exposes a face containing the matrix $Y \in \mathcal{S}^{m+n}$ formed by Z
PREPROCESSING:
1. form the corresponding adjacency matrix A;
2. find a set of cliques Θ from A of size within the given range;
for each clique $X \in \Theta$ **do**
  $[p, q] \leftarrow \text{size}(X)$;
  $[P, Q] \leftarrow \text{FullrankDecompose}(X)$;
  if $p > r$ **then**
    $W \leftarrow PP^T$;
    $[U_X^p, D] \leftarrow \text{eig}(W)$, eigenvalues in nondecreasing order;
    calculate clique weight u_X^p;
  end if
  if $q > r$ **then**
    $W \leftarrow QQ^T$;
    $[U_X^q, D] \leftarrow \text{eig}(W)$, eigenvalues in nondecreasing order
    calculate clique weight u_X^q;
  end if
end for
calculate all the exposing vector weights $w_X^i, i = p, q, X \in \Theta$ from existing clique weights;
sum over existing weights using nullity eigenspaces

$$Y_{expo} \leftarrow \sum_{\substack{i=p,q \\ w_X^i exists, X\in\Theta}} w_X^i \left(U_X^i(:, 1 : i-r)(U_X^i(:, 1 : i-r))^T\right);$$

return Y_{expo}

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

$$\min_{R\in\mathcal{S}_+^{r_v}} \|(VRV^T)_{\tilde{E}} - \tilde{b})\|$$

If the optimal R has the correct target rank, then the exactness of the data implies that necessarily the optimal value is zero; and we are done.

2. If R does not have the correct rank in Item 1 above, then we solve (4.1) for our minimum nuclear norm solution. We note that the linear transformation $\mathcal{M}$ in (4.2) may not be one-one. Therefore, we often need to add a small regularizing term to the objective, i.e., we use $\min \text{trace}(R) + \gamma\|R\|_F$ with small $\gamma > 0$.

4.2 Noisy Case

4.2.1 Base Step after Facial Reduction

As for the noiseless case we complete $\boldsymbol{FR}$ and expect the dimension of R, r_v, to be reduced dramatically. We again begin and solve the simple semidefinite constrained least squares problem

$$\delta_0 = \min_{R \in \mathcal{S}_+^{r_v}} \|(VRV^T)_{\bar{E}} - b)\|, \quad b = Z_{\hat{E}}.$$

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., [11]. 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 problem. As noted in [11], 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.

4.2.2 Refinement Step with Dual Multiplier

If the result from the base step does not have the correct rank, we now use this δ_0 as a best target value for our parametric approach as done in [3]. Denoting $b = Z_{\hat{E}}$ as the vector of known entries in Z in column order, our minimum nuclear norm problem can be stated as:

$$\begin{array}{ll} \min & \operatorname{trace}(R) \\ \text{s.t.} & \|(VRV^T)_{\hat{E}} - b)\| \leq \delta_0 \\ & R \succeq 0. \end{array} \tag{4.3}$$?frsdpnuclear?

To ensure a lower rank solution is obtained through this process, we use the approach in [3] and *flip* this problem:

$$\begin{array}{rll} \varphi(\tau) := \min & \| \left(\hat{V}R\hat{V}^T\right)_{\hat{E}} - b\| + \gamma\|R\|_F \\ \text{s.t.} & \operatorname{trace}(R) \leq \tau \\ & R \succeq 0. \end{array} \tag{4.4}$$?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 shrink the trace of R to reduce the resulting 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 *dual multiplier* of the inquality 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 Numerics

We now present experiments with the algorithm on random instances. Averages (Times, Rank, Residuals) on **five** random instances are included in the table [3]. In the noisy cases we include the output for both before refinement and after refinement. (Total time for both is give after refinement.) We see that in most cases with sufficient density refinement is *not* needed. And, we see that near perfect completion (recovery) is obtained relative to the noise. In particular, the low target rank was attained most times.

The tests were run on MATLAB version R2016a, on a Dell Optiplex 9020, with Windows 7, Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz and 16 GB RAM. For the semidefinite constrained least problems we used the MATLAB addon CVX [8] for simplicity. This means our cputimes could be improved if we replaced CVX with a recent ***SDP*** solver.

5.1 Simulated Data

We generate the instances as done in the recent work [5]. 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 $N(0,1)$. For the noisy data, we 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.

We evaluate our results using the same measurement as in [5], 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 [5] 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. We could combine our preprocessing approach using the cliques before the method in [5] is applied.

5.2 Noiseless Instances

In Tables 5.1 and 5.2 we present the results with noiseless data with target rank $r = 2$ and $r = 4$, respectively. We have left the density of the data relatively high. Note that we set the density in MATLAB at .35 and .4 and obtained .30 and .36, respectively, as the average of the actual densities for the 5 instances. 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 cputime is almost entirely spent on a QR factorization that is used as a heuristic for finding a correct subset of well-conditioned linear constraints. However, we do not need any refinement steps as the high density guarantees that we have enough cliques to cover the nodes in

[3] 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.

the corresponding graph. Table 5.3 illustrates a different approach for lower density problems. We remove the rows and columns of the original data matrix corresponding to zero diagonal elements of the final exposing matrix. 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 quite high for this recovered submatrix, it can then be used as data with another heuristic, such as the nuclear norm heuristic, to recover the complete matrix.

Note that the largest problems have $50,000,000$ data entries in Z with approximately $5,000,000$ unkown values that were recovered successfully. The correct rank was recovered in every instance.

Table 5.1: noiseless: $r = 2$; $m \times n$ size; density p

Specifications			Time (s)	Rank	Residual ($\%Z$)
m	n	mean(p)			
700	2000	0.30	9.00	2.0	4.4605e-14
1000	5000	0.30	28.76	2.0	3.0297e-13
1400	9000	0.30	77.59	2.0	7.8674e-14
1900	14000	0.30	192.14	2.0	6.7292e-14
2500	20000	0.30	727.99	2.0	4.2753e-10

Table 5.2: noiseless: $r = 4$; $m \times n$ size; density p.

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

Table 5.3: sparse data; noiseless: $r = 3$; $m \times n$ size; density p

Specifications			Recover ($\%Z$)	Time (s)	Rank	Residual ($\%Z$)
m	n	p				
700	1000	0.36	100.00	5.21	3.00	4.32e-11
700	1000	0.33	100.00	5.13	3.00	5.69e-11
700	1000	0.30	100.00	4.78	3.00	7.04e-11
700	1000	0.26	99.69	4.79	3.00	3.11e-10
700	1000	0.22	97.77	4.36	3.00	7.66e-05
1100	8000	0.36	100.00	325.58	3.00	6.53e-10
1100	8000	0.33	100.00	321.58	3.00	3.72e-11
1100	8000	0.30	100.00	316.28	3.00	1.92e-10
1100	8000	0.26	100.00	313.04	3.00	5.60e-10
1100	8000	0.22	100.00	307.48	3.00	9.24e-10

5.3 Noisy Instances

The first noisy cases follow in Tables 5.4 and 5.5. As above for the noiseless case we consider problems with relatively high density to ensure that we can find enough cliques. There were some instances where the *eigs* command failed in MATLAB. The algorithm avoids these cases by rounding *tiny* numbers to zero and finding fewer eigenvalues.

In Table 5.4 we consider first increasing noise and then increasing size. In Table 5.5 we allow for larger size and have decreasing density.

Table 5.4: noisy: $r = 3$; $m \times n$ size; density p

Specifications				Time (s)		Rank		Residual ($\%Z$)	
m	n	% noise	p	initial	refine	initial	refine	initial	refine
700	1000	0.00	0.36	4.50	6.35	3.00	3.00	4.14e-14	3.99e-14
700	1000	1.00	0.36	4.22	9.00	3.00	3.00	2.52e-02	2.52e-02
700	1000	2.00	0.36	4.27	9.13	2.60	2.60	4.05e-01	4.04e-01
700	1000	3.00	0.36	4.15	9.43	2.20	2.20	4.31e-01	4.30e-01
700	1000	4.00	0.36	4.30	12.34	1.60	1.60	9.64e-01	9.61e-01
700	1000	1.00	0.36	4.23	8.88	3.00	3.00	2.52e-02	2.52e-02
800	2000	1.00	0.36	11.79	20.60	3.00	3.00	1.91e-02	1.91e-02
900	4000	1.00	0.36	43.27	65.41	3.00	3.00	1.86e-02	1.85e-02
1000	8000	1.00	0.36	156.81	204.76	3.00	3.00	1.46e-02	1.46e-02
1100	16000	1.00	0.36	528.60	673.97	3.00	3.00	1.51e-02	1.51e-02

Table 5.5: noisy: $r = 3$; $m \times n$ size; density p

Specifications				Time (s)		Rank		Residual ($\%Z$)	
m	n	% noise	p	initial	total	initial	refine	initial	refine
700	1000	0.00	0.40	2.22	1.82	2.40	2.40	3.961e-14	3.961e-14
700	1000	0.01	0.40	4.16	8.79	3.20	3.20	9.242e-01	9.360e-01
700	1000	0.15	0.40	3.64	6.32	2.40	2.40	9.416e-01	9.517e-01
700	1000	0.30	0.40	3.46	7.09	8.40	8.40	9.862e-01	9.862e-01
700	1000	0.45	0.40	3.45	4.26	3.80	3.80	9.539e-01	9.539e-01
1500	2000	10.00	0.40	14.07	19.13	2.40	2.40	9.281e-01	9.360e-01
1600	2100	10.00	0.35	13.85	18.03	2.40	2.40	9.535e-01	9.535e-01
1700	2200	10.00	0.30	10.48	30.81	11.00	11.00	8.000e-01	8.000e-01
1800	2300	10.00	0.25	4.22	15.22	4.60	4.60	4.000e-01	4.000e-01
1900	2500	10.00	0.40	21.39	29.03	2.20	2.20	9.506e-01	9.546e-01
2000	2600	10.00	0.35	18.58	50.70	10.20	10.20	9.894e-01	9.894e-01
2100	2700	10.00	0.30	22.75	40.97	6.40	6.40	9.759e-01	9.759e-01
2200	2800	10.00	0.25	6.61	26.14	5.20	5.20	4.000e-01	4.000e-01

5.4 Sparse Noisy Instances

We now consider our last case - noisy instances but with lower density, see Tables 5.6, 5.7.

In this case there may not be enough cliques to cover the entire graph for the problem instance. Moreover, the covered nodes are not covered *well* and so we do not expect good recovery from this poor data in the presence of noise. We report on the percentage of the matrix Z that has been recovered. If we wanted to recover more then we could solve a larger $\boldsymbol{SDP}$ problem as done in [3].

Table 5.6: sparse noisy: $r = 2$; $m \times n$ size; density p

Specifications				Recover (%Z)	Time (s)		Rank		Residual (%Z)	
m	n	% noise	mean(p)		initial	refine	initial	refine	initial	refine
700	1000	0.0e+00	0.18	99.89	3.18	7.91	2.00	2.00	2.32e-12	1.29e-12
700	1000	1.0e-01	0.18	99.89	3.10	13.87	2.40	2.00	1.53e+01	1.11e+00
700	1000	2.0e-01	0.18	99.89	2.98	14.15	2.40	2.00	2.79e+01	2.22e+00
700	1000	3.0e-01	0.18	99.89	3.00	14.36	2.40	2.00	3.83e+01	3.12e+00
700	1000	4.0e-01	0.18	99.89	2.98	15.53	2.40	2.00	4.76e+01	4.23e+00
700	1000	1.0e-03	0.33	100.00	3.64	13.08	2.80	2.00	4.11e-03	4.11e-03
700	1000	1.0e-03	0.30	100.00	3.27	11.71	2.50	2.00	4.49e-03	4.49e-03
700	1000	1.0e-03	0.26	100.00	3.78	15.18	2.20	2.00	3.52e-03	3.52e-03
700	1000	1.0e-03	0.22	100.00	3.88	16.54	2.00	2.00	6.62e-03	6.62e-03
700	1000	1.0e-03	0.18	99.89	3.66	10.34	2.40	2.00	1.35e-01	1.35e-01
900	2000	1.0e-04	0.18	100.00	8.74	25.18	2.60	2.00	1.09e-02	1.09e-02
900	2000	1.0e-04	0.16	100.00	8.39	26.91	2.00	2.00	9.66e-04	9.66e-04
900	2000	1.0e-04	0.14	99.96	7.94	25.85	2.60	2.00	2.84e-02	2.84e-02
900	2000	1.0e-04	0.11	98.89	7.86	24.00	2.40	2.00	1.17e-01	1.16e-01
900	2000	1.0e-04	0.09	92.26	6.48	26.28	2.80	2.00	8.51e-01	3.47e-01

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 cliques are available for our graph description, 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 the efficiency of exploiting the block structure of the cliques and with solving the lower dimensional flipped problems. In

Table 5.7: sparse noisy: $r = 3$; $m \times n$ size; density p

Specifications				Recover (%Z)	Time (s)		Rank		Residual (%Z)	
m	n	% noise	mean(p)		initial	refine	initial	refine	initial	refine
700	1000	0.0e+00	0.18	86.90	1.87	4.85	3.20	3.00	3.76e-07	3.66e-07
700	1000	1.0e-01	0.18	86.75	2.81	31.12	4.25	3.75	2.07e+03	9.78e+01
700	1000	2.0e-01	0.18	86.75	2.77	31.60	4.25	3.50	2.88e+03	2.70e+02
700	1000	3.0e-01	0.18	86.75	2.69	32.35	4.00	3.25	2.96e+03	1.59e+02
700	1000	4.0e-01	0.18	86.90	2.61	35.53	4.00	3.60	6.09e+04	4.10e+02
700	1000	1.0e-03	0.33	100.00	4.93	13.04	3.00	3.00	2.43e-03	2.43e-03
700	1000	1.0e-03	0.30	100.00	4.62	14.36	3.80	3.00	1.77e-02	1.77e-02
700	1000	1.0e-03	0.26	99.69	4.19	16.04	3.00	3.00	6.94e-02	6.94e-02
700	1000	1.0e-03	0.22	97.77	3.81	13.91	3.40	3.00	9.74e-01	8.42e-01
700	1000	1.0e-03	0.18	86.75	2.93	13.23	4.75	3.00	3.54e+00	1.65e+00
900	2000	1.0e-04	0.18	96.81	8.01	26.45	4.60	3.00	9.72e-02	9.71e-02
900	2000	1.0e-04	0.16	92.60	6.04	18.93	4.80	3.00	2.10e+00	7.20e-01
900	2000	1.0e-04	0.16	89.45	5.41	31.18	4.00	3.25	5.72e+01	7.57e-01
900	2000	1.0e-04	0.15	83.53	4.71	18.10	5.00	3.00	2.61e-01	2.60e-01
900	2000	1.0e-04	0.14	74.94	7.21	28.49	4.00	3.00	4.05e+01	5.70e+00

addition, there are many theoretical questions about the complexity of exact recovery guarantees and the relation to the number and size of the cliques.

Acknowledgement

The authors would like to thank Nathan Krislock for his help with parts of the MATLAB coding.

Index

References

[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

[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

[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. 2, 7, 11, 15

[4] D. Drusvyatskiy, G. Pataki, and H. Wolkowicz. Coordinate shadows of semidefinite and Euclidean distance matrices. *SIAM J. Optim.*, 25(2):1160–1178, 2015. 7

[5] 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. 12

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

[7] A.V. Fiacco. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, volume 165 of *Mathematics in Science and Engineering*. Academic Press, 1983. 4

[8] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. `http://cvxr.com/cvx`, April 2011. 12

[9] 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

[10] 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

[11] M. Pilanci and M.J. Wainwright. Randomized sketches of convex programs with sharp guarantees. *IEEE Trans. Infor. Theory*, 61(9):5096–5115, 2015. 3, 11

[12] 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

[13] 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

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

[15] 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, by D. Drusvyatskiy and H. Wolkowicz. 1