

SET: AN ALGORITHM FOR CONSISTENT MATRIX COMPLETION

Wei Dai and Olgica Milenkovic

Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign

Email: {weidai07,milenkov}@illinois.edu

Abstract—A new algorithm, termed subspace evolution and transfer (SET), is proposed for solving the consistent matrix completion problem. For this problem, one is given a subset of the entries of a low-rank matrix, and asked to find *one* low-rank matrix consistent with the given observations. We show that this problem can be solved by searching for a column space that matches the observations. The corresponding algorithm consists of two parts - subspace evolution and subspace transfer. In the evolution part, we use a line search procedure to refine the column space. However, line search can not guarantee convergence as there may exist barriers along the search path that prevent the algorithm from reaching a global optimum. To address this problem, in the transfer part, we design mechanisms to detect barriers and transfer the estimated column space from one side of the barriers to the other. The SET algorithm exhibits excellent empirical performance.

I. INTRODUCTION

Suppose that we observe a subset of entries of a matrix. The matrix completion problem asks when and how the matrix can be *uniquely* recovered based on the observed entries. This reconstruction task is ill-posed and computationally intractable. However, if the data matrix is known to have low-rank, exact recovery can be accomplished in efficient manners, provided that sufficiently many entries are revealed. Low-rank matrix completion problem has received considerable interests due to its wide applications, see for example [5] for more details.

An efficient way to solve the completion problem is via convex relaxation. Instead of looking at rank-restricted matrices, one can search for the matrix with minimum nuclear norm, subject to the data consistency constraint. Although in general nuclear norm minimization is not equivalent to rank minimization, the former will give us the same solution as the latter if the data matrix satisfies certain incoherence conditions [6]. More importantly, nuclear norm minimization can be accomplished by polynomial complexity algorithms, for example, semi-definite programming or singular value thresholding (SVT) [1].

There are other low-complexity alternatives. Based on the subspace pursuit (SP) and CoSaMP algorithms for compressive sensing [7], [8], the authors of [2] developed the so called ADMiRA algorithm. A modification of the power factorization algorithm was used for matrix completion in [3]. Another

approach for solving this problem, termed OptSpace, was described in [4].

The problem considered in this paper and its algorithmic solution differ from all previous approaches. The problem at hand is to identify *one* low-rank matrix *consistent* with the observations. The solution may or may not be unique. In contrast, most results in matrix completion deal with the somewhat more restrictive requirement that sampling is performed in such a way that the reconstruction is *unique*. Hence, our approach can be applied to scenarios where the matrix is highly under-sampled, and where potentially many consistent solutions exist. The relaxation on the uniqueness allows for the empirically observed performance improvement over other completion techniques.

To solve the consistent matrix completion problem, we propose an algorithm, termed subspace evolution and transfer (SET). We show that the matrix completion problem can be solved by searching for a column (or row) space that matches the observations. As a result, optimization on the Grassmann manifold, i.e., subspace evolution, plays the central role in the algorithm. However, there may exist “barriers” along the search path that prevent subspace evolution from converging to a global optimum. To address this problem, in the subspace transfer part, we design mechanisms to detect and cross barriers. Empirical simulations demonstrate the excellent performance of the proposed algorithm.

Despite resembling the OptSpace algorithm [4] in terms of using optimization over Grassmann manifolds, our approach substantially differs from this technique. Searching over only one space (column or row space) represents one of the most significant differences between SET and OptSpace. There, one needs to search *both* the column and the row spaces, which introduces numerical and analytical difficulties. Moreover, when optimizing over the column space, one has to take care of “barriers” that prevent the search procedure from converging to a global optimum, an issue that was not addressed before.

II. CONSISTENT MATRIX COMPLETION

Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be an unknown matrix with rank $r \ll \min(m, n)$, and let $\Omega \subset [m] \times [n]$ be the set of indices of the observed entries, where $[K] = \{1, 2, \dots, K\}$. Define the projection operator

$$\mathfrak{P}_\Omega : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$$

$$\mathbf{X} \mapsto \mathbf{X}_\Omega, \text{ where } (\mathbf{X}_\Omega)_{i,j} = \begin{cases} \mathbf{X}_{i,j} & \text{if } (i,j) \in \Omega \\ 0 & \text{if } (i,j) \notin \Omega \end{cases}.$$

The authors would like to thank Dayu Huang for his help in designing the employed line-search procedure, acknowledge the useful discussion with Yoram Bresler, Justin Haldar, Ely Kerman, Angelia Nedich, and Zoi Rapti, and acknowledge the authors of [1]–[4] for providing online software packages for their matrix completion algorithms.

The *consistent matrix completion* problem is to find one rank- r matrix $\mathbf{X}'$ that is consistent with the observations $\mathbf{X}_\Omega$, i.e.,

$$(P0) : \text{find } \mathbf{X}' \text{ such that} \\ \text{rank}(\mathbf{X}') \leq r \text{ and } \mathfrak{P}_\Omega(\mathbf{X}') = \mathfrak{P}_\Omega(\mathbf{X}) = \mathbf{X}_\Omega. \quad (1)$$

This problem is well defined as $\mathbf{X}_\Omega$ is generated from the matrix $\mathbf{X}$ with rank r and therefore there must exist *at least one solution*. In this paper, like in other heuristic approaches described in [2]–[4], we assume that the rank r is given. This will not introduce serious problems in practice: if r is unknown, we can try different values of r and choose the appropriate one.

III. THE SET ALGORITHM

A. Why optimize over column spaces only?

In this section, we show that the consistent matrix completion problem is equivalent to finding a column (or row) space consistent with the observed entries.

Let $\mathcal{U}_{m,r}$ be the set of $m \times r$ matrices with r orthonormal columns, i.e., $\mathcal{U}_{m,r} = \{\mathbf{U} \in \mathbb{R}^{m \times r} : \mathbf{U}^T \mathbf{U} = \mathbf{I}_r\}$. Define a function

$$f : \mathcal{U}_{m,r} \rightarrow \mathbb{R} \\ \mathbf{U} \mapsto \min_{\mathbf{W} \in \mathbb{R}^{n \times r}} \|\mathbf{X}_\Omega - \mathfrak{P}_\Omega(\mathbf{U}\mathbf{W}^T)\|_F^2, \quad (2)$$

where $\|\cdot\|_F$ denotes the Frobenius norm, i.e., $\|\mathbf{A}\|_F^2 = \sum_{i,j} \mathbf{A}_{i,j}^2$ for an $\mathbf{A} \in \mathbb{R}^{m \times n}$. The function f captures the consistency between $\mathbf{U}$ and the observations $\mathbf{X}_\Omega$: let the minimum in the definition of $f(\mathbf{U})$ be achieved by $\mathbf{W}_U$; if $f(\mathbf{U}) = 0$, then the matrix $\mathbf{U}\mathbf{W}_U^T$ has rank r and satisfies $\mathfrak{P}_\Omega(\mathbf{U}\mathbf{W}_U^T) = \mathbf{X}_\Omega$. Hence, the consistent matrix completion problem is equivalent to

$$(P1) : \text{find } \mathbf{U} \in \mathcal{U}_{m,r} \text{ such that } f(\mathbf{U}) = 0. \quad (3)$$

An important property of f is that $f(\mathbf{U}) = f(\mathbf{U}\mathbf{V})$ for any r -by- r orthogonal matrix $\mathbf{V}$ as $\mathbf{U}\mathbf{W}^T = (\mathbf{U}\mathbf{V})(\mathbf{W}\mathbf{V}^T)^T$. Hence, the function f depends only on the subspace spanned by the columns of $\mathbf{U}$, i.e., the $\text{span}(\mathbf{U})$. Note that $\text{span}(\mathbf{U})$ is the column space of the matrix $\mathbf{U}\mathbf{W}^T$ for any $\mathbf{W} \in \mathbb{R}^{n \times r}$ with full column rank. The consistent matrix completion problem is essentially *finding a column space that is consistent with the observed entries*.

We find the following definitions useful for the exposition to follow. The set of all r -dimensional linear subspaces in $\mathbb{R}^n$ is called the Grassmann manifold, and is denoted by $\mathcal{G}_{m,r}$. Given a subspace $U \in \mathcal{G}_{m,r}$, one can always find a $\mathbf{U} \in \mathcal{U}_{m,r}$ such that $U = \text{span}(\mathbf{U})$. The matrix $\mathbf{U}$ is referred to as a generator matrix of U , and we say $\mathbf{U}$ generates U . Although a given subspace $U \in \mathcal{G}_{m,r}$ has multiple generator matrices, a given matrix $\mathbf{U} \in \mathcal{U}_{m,r}$ uniquely defines a subspace. For this reason, we henceforth use $\mathbf{U}$ to represent its generated subspace.

B. The SET algorithm: a high level description

Our algorithm aims to minimize the objective function $f(\mathbf{U})$, provided that the minimum value of $f(\mathbf{U})$ is *known to be zero*. Ideally, a solution can be obtained by using a line search procedure on the Grassmann manifold. Here, line search refers to as iterative refinements of the interval in which the function attains its minimum. Hence, the “subspace evolution” part of the algorithm reduces to a well studied optimization method.

The main difficulty that arises during line search, and makes the SET algorithm highly non-trivial, is when during the search, one encounters “barriers”. Careful inspection reveals that the objective function f can be decomposed into a sum of atomic functions, each of which involves only one column of $\mathbf{X}_\Omega$ (see Section III-D for details). Along the steepest descent path, these atomic functions may not agree with each other: some decrease and some increase. Rapid increases of some atomic functions may result in “bumps” in the f curve, which block the linear search procedure from global optima and are therefore referred to as *barriers*. The main component of the “transfer” part of the algorithm is to identify, before performing line search, whether there exist barriers along the steepest descent path between the current column space estimate and the global optimizer. Detecting barriers is in general a very difficult task, since one does not know the locations of global minima. Nevertheless, we observe that barriers can be detected by the existence of atomic functions with inconsistent descent directions. When such a scenario is encountered, the algorithm “transfers” the starting point of line search to the other side of the barriers, and proceeds from there. Such a transfer does not overshoot global minima as we enforce consistency of the steep descent directions at the points before and after the transfer.

In summary, we start with a randomly generated $\mathbf{U} \in \mathcal{U}_{m,r}$ and then refine it until $f(\mathbf{U}) = 0$. In each iteration, we first detect and cross barriers if there are any, and then perform line search. The details of subspace evolution and transfer are given in Section III-C and III-D. Simulation results are presented in Section IV.

C. Subspace evolution

Due to space limitation, we focus on the $r = 1$ case in Sections III-C and III-D. Furthermore, our exposition aims to make the algorithmic details as transparent to the readers as possible. The highly technical *performance and complexity analysis* of SET for both $r = 1$ and $r > 1$ is deferred to the journal version of the paper.

For the optimization problem at hand, we shall refine the current column space estimate $\mathbf{u}$ along some geodesic curve on the Grassmann manifold $\mathcal{G}_{m,1}$. Here, the lowercase letter $\mathbf{u}$ is used to emphasize that the $\mathbf{U}$ matrix is a vector when $r = 1$. A geodesic curve on the Grassmann manifold is an analogy of a straight line in the Euclidean space: given two points on the manifold, the geodesic curve connecting them is the path of the shortest length on the manifold. To define the geodesic curve, we first compute the gradient of f as follows. Suppose that $\mathbf{w}_u$ is a length- n column vector that achieves

$f(\mathbf{u})$. Define

$$\mathbf{X}_r = \mathbf{X}_\Omega - \mathfrak{P}_\Omega(\mathbf{u}\mathbf{w}_u^T), \quad (4)$$

referred to as the residue matrix. Then the gradient¹ of f at $\mathbf{u}$ is given by

$$\nabla_{\mathbf{u}} f = -2\mathbf{X}_r \mathbf{w}_u. \quad (5)$$

Note that the gradient describes the tangent vector along which the function f increases the fastest. The tangent vector $\mathbf{h} = -\nabla_{\mathbf{u}} f$ is the vector along which f decreases the fastest. According to [9, Theorem 2.3], the geodesic curve on the Grassmann manifold starting from $\mathbf{u}$, along the tangent vector $\mathbf{h}$, is given by

$$\mathbf{u}(t) = \mathbf{u} \cos t + \mathbf{h} \sin t, \quad t \in [0, \pi]. \quad (6)$$

We restrict t to the interval $[0, \pi)$ because

$$f(\mathbf{u}(t + \pi)) = f(-\mathbf{u}(t)) = f(\mathbf{u}(t)),$$

i.e., $f(\mathbf{u}(t))$ has period π . Interested readers are referred to [9] for more details on geodesics on the Grassmann manifold.

The subspace evolution part is designed to search for a minimizer of the function f along the geodesic curve. It includes two steps. The goal of the first step is to find a $0 < t_{\max} \leq \pi$ such that $f(\mathbf{u}(t))$ must have a minimizer (in most cases, a local minimizer) in the interval $(0, t_{\max})$. The second step is devoted to locating the minimizer $t^* \in (0, t_{\max})$ accurately by iteratively applying the golden section rule [10]. These two steps are described in Algorithm 1. The constants are set to $\epsilon = 10^{-9}$, $c_1 = (\sqrt{5} - 1)/2$, $c_2 = c_1/(1 - c_1)$ and $itN = 10$. Ideally, the number $\epsilon > 0$ should be chosen adaptively. We set it to a fixed constant as it is already sufficiently small in all our experiments.

Algorithm 1 Subspace evolution.

Input: $\mathbf{X}_\Omega$, Ω , $\mathbf{u}$, and itN .

Output: t^* and $\mathbf{u}(t^*)$.

Step A: find $t_{\max}$.

Let $t' = \epsilon\pi$, where $\epsilon > 0$ is a constant.

- 1) Let $t'' = c_2 \cdot t'$.
- 2) If $t'' > \pi$, then $t_{\max} = \pi$ and quit Step A.
- 3) If $f(\mathbf{u}(t'')) > f(\mathbf{u}(t'))$, then $t_{\max} = t''$ and quit Step A.
- 4) Otherwise, $t' = t''$. Go back to step 1).

Step B: numerically search for t^* .

Let $t_1 = t_{\max}/c_2^2$, $t_2 = t_{\max}/c_2$, $t_4 = t_{\max}$, and $t_3 = t_1 + c_1(t_4 - t_1)$. Let $itn = 1$. Perform the following iterations.

- 1) If $f(\mathbf{u}(t_1)) > f(\mathbf{u}(t_2)) > f(\mathbf{u}(t_3))$, then $t_1 = t_2$, $t_2 = t_3$, and $t_3 = t_1 + c_1(t_4 - t_1)$.
- 2) Else, $t_4 = t_3$, $t_3 = t_2$ and $t_2 = t_1 + (1 - c_1)(t_4 - t_1)$.
- 3) $itn = itn + 1$.
- 4) If $itn > itN$, then quit the iterations. Otherwise, go back to step 1).

Let $t^* = \arg \min_{t \in \{t_1, \dots, t_4\}} f(\mathbf{u}(t))$ and compute $\mathbf{u}(t^*)$.

¹In fact, the gradient is not defined at finitely many points in $\mathcal{U}_{m,r}$. However, the probability of hitting those points during search is zero.

D. Subspace transfer

In the matrix completion problem, the objective function $f(\mathbf{u})$ may not be a convex function of $\mathbf{u}$. The linear search procedure in Algorithm 1 may not converge to a global minimum. The reason behind this problem is that the search path may be blocked by what we call “barriers”. We show next how to overcome the problem introduced by barriers.

At this point, we formally introduce the decoupling principle. Let $\mathbf{x}_{\Omega_j} \in \mathbb{R}^{m \times 1}$ be the j^{th} column of the matrix $\mathbf{X}_\Omega$. Define vectors $\mathbf{u}_{\Omega_j} \in \mathbb{R}^{m \times 1}$ and $\mathbf{h}_{\Omega_j} \in \mathbb{R}^{m \times 1}$ such that

$$\begin{aligned} (\mathbf{u}_{\Omega_j})_i &= \mathbf{u}_i, \quad (\mathbf{h}_{\Omega_j})_i = \mathbf{h}_i, \quad \text{if } (i, j) \in \Omega; \\ (\mathbf{u}_{\Omega_j})_i &= 0, \quad (\mathbf{h}_{\Omega_j})_i = 0, \quad \text{if } (i, j) \notin \Omega. \end{aligned}$$

Furthermore, let

$$\mathbf{u}_{\Omega_j}(t) = \mathbf{u}_{\Omega_j} \cos t + \mathbf{h}_{\Omega_j} \sin t.$$

Then the objective function $f(\mathbf{u}(t))$ can be written as a sum of n atomic functions:

$$\begin{aligned} f(\mathbf{u}(t)) &= \min_{\mathbf{w} \in \mathbb{R}^{n \times 1}} \|\mathbf{X}_\Omega - \mathfrak{P}_\Omega(\mathbf{u}\mathbf{w}^T)\|_F^2 \\ &= \sum_{j=1}^n \underbrace{\|\mathbf{x}_{\Omega_j} - \mathfrak{P}(\mathbf{x}_{\Omega_j}, \mathbf{u}_{\Omega_j}(t))\|_F^2}_{f_j(\mathbf{u}(t))}, \end{aligned} \quad (7)$$

where $\mathfrak{P}(\mathbf{x}_{\Omega_j}, \mathbf{u}_{\Omega_j}(t))$ is the projection of the vector $\mathbf{x}_{\Omega_j}$ on $\text{span}(\mathbf{u}_{\Omega_j}(t))$.

The following example illustrates the concept of a barrier. Let $r = 1$, $\Omega = [3] \times [2] - \{(1, 1), (2, 2)\}$, and $\mathbf{X}_\Omega = \begin{bmatrix} 0, 2, 2 \\ 2, 0, 1 \end{bmatrix}^T$. It is clear that the global minimizer is given by $\mathbf{u}_X = \frac{1}{\sqrt{6}}[2, 1, 1]^T$, i.e., $f(\mathbf{u}_X) = 0$. Let $\mathbf{u} \in \mathcal{U}_{m,1}$ be a generator matrix of the current estimate of the column space, and $\mathbf{u}_i$ be the i^{th} entry of $\mathbf{u}$. Suppose that $\mathbf{u} = \frac{1}{\sqrt{102}}[-10, 1, 1]^T$. Fig. 1a gives the contour of f_1 , projected on the plane spanned by $\mathbf{u}_2$ and $\mathbf{u}_3$. Along the line search direction $\mathbf{h}$, the atomic function f_1 increases very fast. Careful tracking of several line search procedures (Fig. 1b) shows that the estimate $\mathbf{u}$ will approach $[-1, 0, 0]^T$, but will never cross the contour $f_1 = 8$. That is, the contour $f_1 = 8$ forms a “barrier” for the line search procedure.

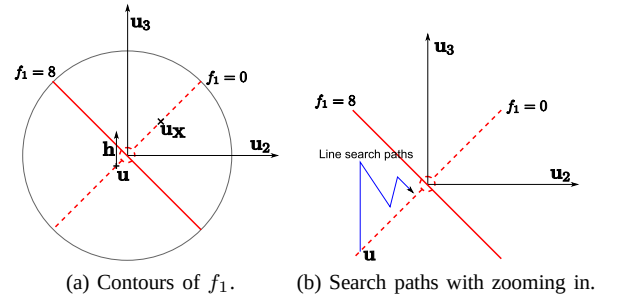


Fig. 1: An illustrative example for barriers.

To define barriers formally, we need to study the atomic functions $f_j(\mathbf{u}(t))$'s. If

$$\text{rank}([\mathbf{u}_{\Omega_j}, \mathbf{h}_{\Omega_j}]) = 1,$$

then $f_j(\mathbf{u}(t))$ is a constant almost everywhere. Suppose that

$$\text{rank}([\mathbf{u}_{\Omega_j}, \mathbf{h}_{\Omega_j}]) = 2.$$

It can be verified that $f_j(\mathbf{u}(t))$ has unique minimizer and maximizer, given by

$$t_{\max,j} = \arg \max_{t \in [0, \pi)} \|\mathbf{x}_{\Omega_j} - \mathfrak{P}(\mathbf{x}_{\Omega_j}, \mathbf{u}_{\Omega_j}(t))\|_F^2, \quad (8)$$

$$t_{\min,j} = \arg \min_{t \in [0, \pi)} \|\mathbf{x}_{\Omega_j} - \mathfrak{P}_{\Omega_j}(\mathbf{x}_{\Omega_j}, \mathbf{u}_{\Omega_j}(t))\|_F^2, \quad (9)$$

respectively. These two quantities can be easily computed.

We say that the k^{th} column of $\mathbf{X}_{\Omega}$ forms a barrier for the j^{th} column if the following conditions are satisfied:

- 1) $\text{rank}([\mathbf{u}_{\Omega_k}, \mathbf{h}_{\Omega_k}]) = \text{rank}([\mathbf{u}_{\Omega_j}, \mathbf{h}_{\Omega_j}]) = 2$, alternatively, neither f_k nor f_j is a constant function;
- 2) $t_{\max,k} < t_{\min,j} < t_{\max,j}$, which means that the maximizer of f_k appears before the minimizer of f_j ;
- 3) $\frac{d}{dt} f(\mathbf{u}(t))|_{t=t_{\max,k}} < 0$, i.e., the vectors of the steepest descent of f at $t = 0$ and $t_{\max,k}$ are consistent (form a sharp angle).

Note that for a given column j of $\mathbf{X}_{\Omega}$, there may exist multiple barriers for it.

When barriers are detected, we transfer $\mathbf{u}$ across them. In our implementation, we focus on the closest barriers to $\mathbf{u}$. Define

$$\mathcal{J} = \{j : \text{the } j^{\text{th}} \text{ column of } \mathbf{X}_{\Omega} \text{ admits barriers}\},$$

$$j^* = \arg \min_{j \in \mathcal{J}} t_{p,j}, \quad (10)$$

$$\mathcal{K}_j = \{k : \text{the } k^{\text{th}} \text{ column of } \mathbf{X}_{\Omega} \text{ forms a barrier for the } j^{\text{th}} \text{ column of } \mathbf{X}_{\Omega}\},$$

and

$$k^* = \arg \max_{k \in \mathcal{K}_{j^*}} t_{o,k}. \quad (11)$$

The subspace transfer part is described in Algorithm 2.

Algorithm 2 Subspace transfer

Input: $\mathbf{X}_{\Omega}$, Ω , and $\mathbf{u}$.

Output: t_{st} and $\mathbf{u}(t_{st})$.

Steps:

- 1) Compute $t_{o,j}$ and $t_{p,j}$ for each column j satisfying $\text{rank}([\mathbf{u}_{\Omega_j}, \mathbf{h}_{\Omega_j}]) = 2$.
 - 2) Suppose that there exist barriers.
 - a) find j^* and k^* according to (10) and (11) respectively.
 - b) Let $t_{st} = t_{o,k^*}$ and compute $\mathbf{u}(t_{st})$.
 - 3) Otherwise, $t_{st} = 0$ and $\mathbf{u}(t_{st}) = \mathbf{u}$.
-

IV. PERFORMANCE EVALUATION

Here, we introduce an error tolerance parameter $\epsilon_e > 0$. In practice, instead of requiring exact data matching, it usually suffices to have $\|\mathfrak{P}_{\Omega}(\mathbf{X}') - \mathbf{X}_{\Omega}\|_F^2 < \epsilon_e \|\mathbf{X}_{\Omega}\|_F^2$ for some small ϵ_e . In our simulations, we set $\epsilon_e = 10^{-6}$.

We tested the SET algorithm by randomly generating low-rank matrices $\mathbf{X}$ and index sets Ω . Specifically, we decompose the matrix $\mathbf{X}$ into $\mathbf{X} = \mathbf{U}_{\mathbf{X}} \mathbf{S}_{\mathbf{X}} \mathbf{V}_{\mathbf{X}}^T$, where $\mathbf{U}_{\mathbf{X}} \in \mathcal{U}_{m,r}$, $\mathbf{V}_{\mathbf{X}} \in \mathcal{U}_{n,r}$, and $\mathbf{S}_{\mathbf{X}} \in \mathbb{R}^{r \times r}$. We generate $\mathbf{U}_{\mathbf{X}}$ and $\mathbf{V}_{\mathbf{X}}$ from the isotropic distribution on the set $\mathcal{U}_{m,r}$ and $\mathcal{U}_{n,r}$, respectively. The entries of the $\mathbf{S}_{\mathbf{X}}$ matrix are independently drawn from the standard Gaussian distribution $\mathcal{N}(0, 1)$. This step is important in order to guarantee the randomness in the singular values of $\mathbf{X}$. The index set Ω is randomly generated from the uniform distribution over the set $\{\Omega' \subset [m] \times [n] : |\Omega'| = |\Omega|\}$.

The performance of the SET algorithm is surprisingly good. For the results presented in Table I, we tested different matrices with different ranks. For each size and each rank, we tried different sampling rates, defined as $|\Omega| / (m \times n)$. The SET algorithm succeeds for all the realizations that we generated. We also compare the SET algorithm to other matrix completion algorithms. As shown in Figure 2, the SET algorithm outperforms all known completion approaches². In simulations, the number of iterations of the SET algorithm did not exceed 500, while that of other completion algorithms were set to 2000. All tested algorithms had run-time of the same order.

Matrix Size	Ranks	Sample Rates	# of Realizations	Recovery Rate
9 × 9	1, 2	1/27, 2/27, ..., 1	500	100%
50 × 50	1, 2, 3, 4	0.04, 0.08, ..., 1	200	100%

TABLE I: The performance of the SET algorithm.

REFERENCES

- [1] J. Cai, E. J. Candes, and Z. Shen, "A singular value thresholding algorithm for matrix completion," Oct. 2008.
- [2] K. Lee and Y. Bresler, "ADMiRA: atomic decomposition for minimum rank approximation," Apr. 2009.
- [3] J. Haldar and D. Hernando, "Rank-constrained solutions to linear matrix equations using powerfactorization," *IEEE Signal Processing Letters*, pp. 16:584–587, 2009.
- [4] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," 2009.
- [5] E. Candes and B. Recht, "Exact matrix completion via convex optimization," *Submitted for publication*, 2008.
- [6] E. J. Candes and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," Mar. 2009.

²Note that the SVT algorithm is not designed to solve the problem (P0). We include it for completeness. In the standard SVT algorithm, although the unknown matrix $\mathbf{X}$ has rank r , the *reconstructed* matrix may have a rank larger than r . In order to have the output matrix have rank at most r , we did the following. We set the error tolerance of the standard SVT algorithm to $\epsilon_e/2$, i.e., the output matrix $\hat{\mathbf{X}}$ satisfies $\|\mathbf{X}_{\Omega} - \mathfrak{P}_{\Omega}(\hat{\mathbf{X}})\|_F^2 \leq \epsilon_e \|\mathbf{X}_{\Omega}\|_F^2 / 2$. Let $s_1 \geq s_2 \geq \dots \geq s_{\min(m,n)}$ be the singular values of $\hat{\mathbf{X}}$. We performed singular value decomposition of $\hat{\mathbf{X}}$, set the singular values $s_{r+1} \geq \dots \geq s_{\min(m,n)}$ to zero, and denoted the resulting matrix by $\mathbf{X}'$. Clearly, we have $\text{rank}(\mathbf{X}') \leq r$. We claimed a successful reconstruction if $\|\mathbf{X}_{\Omega} - \mathfrak{P}_{\Omega}(\mathbf{X}')\|_F^2 \leq \epsilon_e \|\mathbf{X}_{\Omega}\|_F^2$, and a failure otherwise.

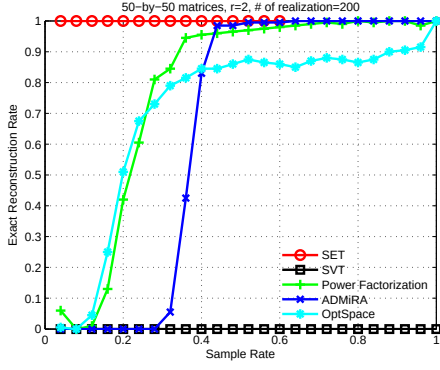


Fig. 2: Performance comparison.

- [7] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inform. Theory*, vol. 55, pp. 2230 – 2249, May 2009.
- [8] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, pp. 301–321, May 2009.
- [9] A. Edelman, T. Arias, S. T. Smith, Steven, and T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, pp. 303–353, April 1999.
- [10] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. Academic Press, 1982.