Low-Rank Data Completion With Very Low Sampling Rate Using Newton's Method

Morteza Ashraphijuo, Xiaodong Wang, *Fellow, IEEE*, and Junwei Zhang

***Abstract*—Newton's method is a widely applicable and empirically efficient method for finding the solution to a set of equations. The recently developed algebraic geometry analyses provide information-theoretic bounds on the sampling rate to ensure the existence of a unique completion. A remained open question from these works is to retrieve the sampled data when the sampling rate is very close to the mentioned information-theoretic bounds. This paper is concerned with proposing algorithms to retrieve the sampled data when the sampling rate is too small and close to the mentioned information-theoretic bounds. Hence, we propose a new approach for recovering a partially sampled low-rank matrix or tensor when the number of samples is only slightly more than the dimension of the corresponding manifold, by solving a set of polynomial equations using Newton's method. In particular, we consider low-rank matrix completion, matrix sensing, and tensor completion. Each observed entry contributes one polynomial equation in terms of the factors in the rank factorization of the data. By exploiting the specific structures of the resulting set of polynomial equations, we analytically characterize the convergence regions of the Newton's method for matrix completion and matrix sensing. Through extensive numerical results, we show that the proposed approach outperforms the well-known methods such as nuclear norm minimization and alternating minimization in terms of the success rate of data recovery (noiseless case) and peak signal-to-noise ratio (noisy case), especially when the sampling rate is very low.**

***Index Terms*—Low-rank matrix completion, rank factorization, matrix sensing, low-rank tensor completion, sampling rate, polynomial equations, convergence analysis.**

I. Introduction

THIS paper is concerned with the problem of reconstructing a partially sampled low-rank matrix or tensor, i.e., the low-rank data completion problem, which has many applications in various areas of engineering and applied science including image or signal processing [1], [2], network coding [3], control [4], data mining [5], data clustering [6]–[9], recommender systems and collaborative filtering [10], [11], etc. There are various techniques for tackling the low-rank data completion problem, including convex relaxations of rank [1], [12]–[18], alternating minimization [19]–[23], augmented Lagrangian method [24], generalized round-rank [25], [26], algebraic geometric analyses [27]–[34] and other heuristics [35]–[40]. Among them, the nuclear norm minimization method is the most effective and robust solution, although its complexity is very high [13]. On the other hand, the alternating minimization method is very fast, but with less satisfactory performance. The matrix sensing problem is a generalization of the matrix completion problem, where a set of linear matrix equations instead of a set of entries are given. A review on matrix sensing problem and its applications can be found in [41].

The recently developed algebraic geometric analyses in [27]–[31] make use of the rank decomposition and transform the data completion problem to the problem of solving a system of polynomial equations. In particular, each observed entry corresponds to a polynomial in terms of the entries of the factors in the rank factorization of the data. Therefore, any solution to the set of polynomial equations results in a completion of the data of the given rank constraint. As a result, the problem of low-rank matrix completion can be translated to finding a root for a system of semi-homogeneous polynomials. In addition, the mentioned works provide information-theoretic bounds on the sampling rate to ensure the existence of a unique completion with a very high probability. A remained open question from the above-mentioned works is whether we can retrieve the sampled data when the sampling rate is very close to the obtained information-theoretic bounds on the sampling rate. This work is concerned with proposing algorithms to retrieve the sampled data when the sampling rate is too small and close to the mentioned information-theoretic bounds.

In this paper, we propose to use Newton's method to solve the set of polynomial equations and to obtain the completion of the data. This can be done for any well-known data structure by using the rank factorization as we will show. By exploiting the structure of the polynomials, we can analytically characterize the convergence region of the proposed Newton's method for matrix completion and matrix sensing. In particular, our method is extremely efficient in comparison with the existing methods in the literature, when the sampling rate is too low. This can be also easily extended to homotopy methods. In the numerical experiments, we use the successful recovery rate and the peak signal-to-noise ratio as performance metrics for noiseless and noisy cases, respectively. We show that the proposed Newton's method outperforms the existing nuclear norm minimization and alternating minimization methods especially when the sampling

Manuscript received November 27, 2017; revised July 8, 2018 and January 10, 2019; accepted February 5, 2019. Date of publication February 18, 2019; date of current version March 1, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xinming Huang. This work was supported in part by the U.S. National Science Foundation under Grant CCF-1814803 and in part by the U.S. Office of Naval Research under Grant N000141410667. *(Corresponding author: Xiaodong Wang.)*

M. Ashraphijuo and X. Wang are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: ashraphijuo@ee.columbia.edu; wangx@ee.columbia.edu).

J. Zhang is with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: junweiz@andrew.cmu.edu).

Digital Object Identifier 10.1109/TSP.2019.2899315

rate is very low, i.e., when the number of samples is close to the dimension of the corresponding manifold. Note that alternating minimization is much faster than nuclear norm minimization, but nuclear norm minimization requires fewer samples to successfully recover the data. Our proposed approach outperforms nuclear norm minimization in terms of both speed and the number of samples for successful completion and it outperforms alternating minimization in terms of the number of required samples. We also provide numerical experiments on real-world datasets such as the MovieLens dataset and show that the proposed Newton's method outperforms both existing methods.

The remainder of this paper is organized as follows. In Section II, the Newton's method for matrix completion is proposed. In Section III, we characterize the convergence region of the proposed Newton's matrix completion algorithm. In Section IV, we provide numerical results to compare the proposed method with existing matrix completion algorithms. In Section V, we extend the proposed framework and propose Newton' method for solving matrix sensing and tensor completion problems. Finally, Section VI concludes the paper.

II. Matrix Completion

A. Background

Let $\mathbf{U} \in \mathbb{R}^{n_1 \times n_2}$ denote a rank-r matrix that is partially observed. Denote Ω as the set of indices (i, j) such that $\mathbf{U}(i, j)$ is observed. Moreover, define $\mathbf{U}_\Omega$ as the matrix obtained from sampling $\mathbf{U}$ according to Ω, i.e.,

$$\mathbf{U}_\Omega(i,j) = \begin{cases} \mathbf{U}(i,j) & \text{if } (i,j) \in \Omega, \\ 0 & \text{if } (i,j) \notin \Omega. \end{cases} \tag{1}$$

The matrix completion problem is to recover the original rank-r matrix $\mathbf{U}$ given its sampled version $\mathbf{U}_\Omega$. Alternating minimization [19] and nuclear norm minimization [12] are two well-known methods for matrix completion.

In the alternating minimization approach, we write $\mathbf{U} = \mathbf{XY}$ such that $\mathbf{X} \in \mathbb{R}^{n_1 \times r}$ and $\mathbf{Y} \in \mathbb{R}^{r \times n_2}$. Starting with some initial $\mathbf{X}_0$ and $\mathbf{Y}_0$, at the k-th iteration, given $\mathbf{X}_{k-1}$ and $\mathbf{Y}_{k-1}$, we first update $\mathbf{X}_k$ by solving the following convex program

$$\underset{\mathbf{X}_k \in \mathbb{R}^{n_1 \times r}}{\text{minimize}} \qquad \|\mathbf{U}_\Omega - (\mathbf{X}_k \mathbf{Y}_{k-1})_\Omega\|_{\mathcal{F}}, \tag{2}$$

and then update $\mathbf{Y}_k$ by solving

$$\underset{\mathbf{Y}_k \in \mathbb{R}^{r \times n_2}}{\text{minimize}} \qquad \|\mathbf{U}_\Omega - (\mathbf{X}_k \mathbf{Y}_k)_\Omega\|_{\mathcal{F}}, \tag{3}$$

where $\|\cdot\|_{\mathcal{F}}$ denotes the Frobenius norm. The iteration continues until both errors are below certain threshold.

On the other hand, in the nuclear norm minimization approach, we do not make use of the rank constraint and solve the following relaxed problem which is a convex program

$$\begin{aligned} \underset{\mathbf{U}' \in \mathbb{R}^{n_1 \times n_2}}{\text{minimize}} \quad & \|\mathbf{U}'\|_* \\ \text{subject to} \quad & \mathbf{U}'_\Omega = \mathbf{U}_\Omega \ , \end{aligned} \tag{4}$$

where $\|\cdot\|_*$ denotes the matrix nuclear norm, i.e., the sum of singular values.

In this paper, we compare our approach for low-rank data completion with two well-known approaches, i.e., nuclear norm minimization and alternating minimization. All these three approaches are different only in terms of formulation, which results in different performances. For example, as we will show in the paper, among the three of them alternating minimization is the fastest and our approach is slightly slower but nuclear norm minimization (the only convex formulation) is very slow. However, our formulation requires much less number of samples to recover the original sampled data in comparisons with the two other formulations. Note that one of the advantages of nuclear norm minimization approach in comparison with the non-convex approaches like alternating minimization and our proposed method is the fact that the only required input is the observed partial matrix and not the value of the rank. Whereas, in the mentioned non-convex approaches, the value of the rank is required. On the other hand, generally, these non-convex approaches are much faster than the convex approaches like nuclear norm minimization. For the alternating minimization formulation, we use the numerical method in [19] and for the nuclear norm minimization we used both CVX toolbox and also the online implementation [42].

Our method and the described alternating minimization method are both solving the same problem, i.e., Low Rank Matrix Completion (LRMC) problem, and they are only two different numerical methods.

B. Matrix Completion via Solving Polynomial Equations

The dimension of the manifold of $n_1 \times n_2$ matrices of rank r is $r(n_1 + n_2 - r)$. $\mathbf{X} \in \mathbb{R}^{n_1 \times r}$ is called a basis of $\mathbf{U}$ if each column of $\mathbf{U}$ can be written as a linear combination of the columns of $\mathbf{X}$. The following lemma gives a unique canonical decomposition of $\mathbf{U}$.

Lemma 1: Let $\mathbf{U}$ denote a generically chosen matrix from the manifold of $n_1 \times n_2$ matrices of rank r. Then, there exists a unique decomposition $\mathbf{U} = \mathbf{XY}$ such that $\mathbf{X} \in \mathbb{R}^{n_1 \times r}$, $\mathbf{Y} \in \mathbb{R}^{r \times n_2}$ and $\mathbf{X}(1:r, 1:r) = \mathbf{I}_r$, where $\mathbf{X}(1:r, 1:r)$ represents the submatrix of $\mathbf{X}$ consisting of the first r columns and the first r rows and $\mathbf{I}_r$ denotes the $r \times r$ identity matrix.

Proof: We show that there exists exactly one decomposition $\mathbf{U} = \mathbf{XY}$ such that $\mathbf{X}(1:r, 1:r) = \mathbf{I}_r$ with probability one. Considering the first r rows of $\mathbf{U} = \mathbf{XY}$, we conclude $\mathbf{U}(1:r, :) = \mathbf{I}_r \mathbf{Y} = \mathbf{Y}$. Therefore, we need to show that there exists exactly one $\mathbf{X}(r+1:n_1, :)$ such that $\mathbf{U}(r+1:n_1, :) = \mathbf{X}(r+1:n_1, :)\mathbf{Y}$ or equivalently $\mathbf{U}(r+1:n_1, :)^\top = \mathbf{U}(1:r, :)^\top \mathbf{X}(r+1:n_1, :)^\top$. It suffices to show that each column of $\mathbf{X}(r+1:n_1, :)$ can be determined uniquely having $\mathbf{u} = \mathbf{U}(1:r, :)^\top \mathbf{x}$ where $\mathbf{u} \in \mathbb{R}^{n_2 \times 1}$ and $\mathbf{x} \in \mathbb{R}^{r \times 1}$. As $\mathbf{U}$ is a generically chosen $n_1 \times n_2$ matrix of rank r, we have rank $(\mathbf{U}(1:r, :)) = r$ with probability one. Hence, $\mathbf{u}(1:r) = \mathbf{U}(1:r, 1:r)^\top \mathbf{x}$ results in r independent degree-1 equations in terms of the r variables (entries of $\mathbf{x}$), and therefore $\mathbf{x}$ has exactly one solution with probability one. ■

Remark 1: The above lemma holds true for any permutation of rows of the identity matrix among all n_1 rows of the basis matrix $\mathbf{X}$.

Remark 2: This structure is only for simplifying the obtained system of polynomial equations. Note that we only use this structure when $\mathbf{U}$ is a generically chosen matrix from the man-

ifold of $n_1 \times n_2$ matrices of rank r. Otherwise, we do not use this structure.

Consider a rank decomposition of the original matrix, i.e., $\mathbf{U} = \mathbf{XY}$, where $\mathbf{X} \in \mathbb{R}^{n_1 \times r}$ and $\mathbf{Y} \in \mathbb{R}^{r \times n_2}$. Note that each observed entry of $\mathbf{U}$ results in a polynomial in terms of the entries of $\mathbf{X}$ and $\mathbf{Y}$ as follows

$$\sum_{k=1}^{r} \mathbf{X}(i,k)\mathbf{Y}(k,j) - \mathbf{U}(i,j) = 0, \qquad \text{for } (i,j) \in \Omega. \quad (5)$$

Given the canonical structure, to recover $\mathbf{U}$ we need to solve for the $D = r(n_1 + n_2 - r)$ unknown entries of $(\mathbf{X}, \mathbf{Y})$ from the set of second-order polynomial equations in (5). An illustrative example is provided next to show how to obtain such an equation set.

Example 1: In this example, we want to show the process of obtaining the mentioned polynomials based on the given sampled entries. Consider a 4×3 matrix $\mathbf{U}$ of rank 2 with the following observed entries

$$\mathbf{U}_\Omega = \begin{bmatrix} 4 & 7.3 & 0 \\ 0 & 0 & 0 \\ 1 & 8.3 & 22.1 \\ 0 & 12.1 & 0 \end{bmatrix}.$$

Since $\text{rank}(\mathbf{U}) = 2$, the canonical decomposition is

$$\mathbf{U} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ x_1 & x_2 \\ x_3 & x_4 \end{bmatrix} \begin{bmatrix} y_1 & y_3 & y_5 \\ y_2 & y_4 & y_6 \end{bmatrix}.$$

Denote $\mathbf{z} = [x_1, \ldots, x_4, y_1, \ldots, y_6]^\top \in \mathbb{R}^{10}$ and then the corresponding system of second-order polynomial equations is

$$\mathbf{p}(\mathbf{z}) = \mathbf{0} \Rightarrow \begin{cases} y_1 - 4 = 0, \\ y_3 - 7.3 = 0, \\ x_1 y_1 + x_2 y_2 - 1 = 0, \\ x_1 y_3 + x_2 y_4 - 8.3 = 0, \\ x_1 y_5 + x_2 y_6 - 22.1 = 0, \\ x_3 y_3 + x_4 y_4 - 12.1 = 0. \end{cases}$$

Moreover, we can simplify the polynomials by permuting the rows of the identity matrix mentioned in Remark 2 as follows

$$\mathbf{U} = \begin{bmatrix} 1 & 0 \\ x_1 & x_2 \\ 0 & 1 \\ x_3 & x_4 \end{bmatrix} \begin{bmatrix} y_1 & y_3 & y_5 \\ y_2 & y_4 & y_6 \end{bmatrix}.$$

Then, the corresponding system of second-order polynomial equations becomes

$$\bar{\mathbf{p}}(\mathbf{z}) = \mathbf{0} \Rightarrow \begin{cases} y_1 - 4 = 0, \\ y_3 - 7.3 = 0, \\ y_2 - 1 = 0, \\ y_4 - 8.3 = 0, \\ y_6 - 22.1 = 0, \\ x_3 y_3 + x_4 y_4 - 12.1 = 0. \end{cases}$$

It is easy to see that compared with $\mathbf{p}(\mathbf{z})$, $\bar{\mathbf{p}}(\mathbf{z})$ has more first-order equations and therefore it is easier to solve.

C. The Algorithm

1) Locations of Unknowns in $\mathbf{X}$*:* We choose the r rows with the most number of samples in $\mathbf{U}_\Omega$ and permute the rows of the identity matrix in $\mathbf{X}$ to those rows to have as many linear (i.e., first-order) polynomials as possible.

2) Newton's Method for Solving $\mathbf{p}(\mathbf{z}) = \mathbf{0}$*:* Define the vector $\mathbf{z} \in \mathbb{R}^{D \times 1}$ such that it contains all the D unknown elements of the decomposition. For example, if we choose $\mathbf{X}(1:r, 1:r) = \mathbf{I}_r$, then we have $\mathbf{z} = \text{vec}(\mathbf{X}(r+1:n_1, :), \mathbf{Y})$, where $\text{vec}(\cdot)$ denotes the vectorization operator. Then, it follows from (5) that each observed entry results in a polynomial that involves r entries of $\mathbf{X}$ and r entries of $\mathbf{Y}$. Hence, we have a set of second-order polynomial equations $p_i(\mathbf{z}) = 0$, $i = 1, \ldots, |\Omega|$, where $|\Omega|$ denotes the number of observed entries. Denote $\mathbf{p}(\mathbf{z}) = [p_1(\mathbf{z}), \ldots, p_{|\Omega|}(\mathbf{z})]^\top$.

In order to solve $\mathbf{p}(\mathbf{z}) = \mathbf{0}$, we use the simple Newton's method. In particular, we start with some initial $\mathbf{z}_0 \in \mathbb{R}^{D \times 1}$, and perform the following iteration

$$\mathbf{z}_n = \mathbf{z}_{n-1} - (\nabla \mathbf{p}(\mathbf{z}_{n-1}))^\dagger \, \mathbf{p}(\mathbf{z}_{n-1}), \quad (6)$$

where $\nabla \mathbf{p}(\mathbf{z}) \in \mathbb{R}^{|\Omega| \times D}$ and its (i,j)-th element denotes the partial derivate of $p_i(\mathbf{z})$ with respect to z_j, i.e., $\frac{\partial p_i(\mathbf{z})}{\partial z_j}$ and the operator $\dagger$ denotes pseudoinverse. Equivalently,

$$\nabla \mathbf{p}(\mathbf{z}_n) = [\nabla p_1(\mathbf{z}_n)^\top \ldots \nabla p_{|\Omega|}(\mathbf{z}_n)^\top]^\top \in \mathbb{R}^{|\Omega| \times D}, \quad (7)$$

where

$$\nabla p_i(\mathbf{z}_n) = \left[\frac{\partial p_i}{\partial z_1}, \ldots, \frac{\partial p_i}{\partial z_D} \right] \in \mathbb{R}^{1 \times D}, \quad (8)$$

for $i = 1, \ldots, |\Omega|$. We assume that $|\Omega| \geq D$ and $\nabla \mathbf{p}(\mathbf{z}_0)$ is full column-rank. Corollary1 in Section III states that under mild assumptions if $\nabla \mathbf{p}(\mathbf{z}_0)$ is full column-rank and $\mathbf{z}_0$ falls into a certain ball around $\mathbf{z}^*$, then $\nabla \mathbf{p}(\mathbf{z}_n)$ is also full column-rank, $n = 1, 2, \ldots$.

Note that the matrix $\nabla \mathbf{p}(\mathbf{z})$ has a very sparse structure. This is because the number of involved variables in each polynomial is either r or $2r$ (depending on the location of canonical pattern). Such sparsity enables a fast computation of its inverse or pseudoinverse, e.g., the command $\mathsf{sparse}(\nabla \mathbf{p}(\mathbf{z})) \setminus \mathbf{p}(\mathbf{z})$ in Matlab is an efficient way to calculate $(\nabla \mathbf{p}(\mathbf{z}_{n-1}))^\dagger \, \mathbf{p}(\mathbf{z}_{n-1})$. For instance, for the example $\tilde{\mathbf{p}}(\mathbf{z})$ described earlier we have

$$\nabla \bar{\mathbf{p}}(\mathbf{z}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & y_3 & y_4 & 0 & 0 & x_3 & x_4 & 0 & 0 \end{bmatrix}.$$

3) Initialization: We can use the initialization scheme for alternating minimization as in [19]. Consider the singular value decomposition $\mathbf{U}_\Omega = \mathbf{U}_0 \mathbf{S}_0 \mathbf{V}_0^\top$, where the number of nonzero entries of $\mathbf{S}_0$ can be more than r. Now, we define a decompostion corresponding to the r largest singular values, i.e., $\mathbf{U}_0(:, 1:r)\mathbf{S}_0(1:r, 1:r)\mathbf{V}_0(:, 1:r)^\top = \mathbf{LR}$, where $\mathbf{L} = \mathbf{U}_0(:, 1:r)\mathbf{S}_0(1:r, 1:r) \in \mathbb{R}^{n_1 \times r}$ and $\mathbf{R} = \mathbf{V}_0(:, 1:r)^\top \in \mathbb{R}^{r \times n_2}$.

We choose the r rows of $\mathbf{U}_\Omega$ with the most number of samples and permute the rows of the identity matrix in the basis matrix $\mathbf{X}$ to those rows to increase the number of linear polynomials as much as possible. Let $\mathcal{I} = \{i_1, \ldots, i_r\}$ denote the indices corresponding to these r rows. Let $\mathbf{L}' = \mathbf{L}\left(\mathbf{L}(\mathcal{I}, 1:r)\right)^{-1}$ and $\mathbf{R}' = \mathbf{L}(\mathcal{I}, 1:r)\mathbf{R}$. Finally, we set $(\mathbf{X}_0, \mathbf{Y}_0) = (\mathbf{L}', \mathbf{R}')$ as the initial value in Newton's method. As we will observe in the numerical results, this initialization significantly increases the efficiency and decreases the number of required observations for successful recovery. We denote the initialization procedure described above as $\mathsf{Init}(\mathbf{U}_\Omega)$.

As an alternative initialization method, we start by applying the nuclear norm minimization to $\mathbf{U}_\Omega$ and obtain a completion $\tilde{\mathbf{U}}$. Then, we apply the above-mentioned initialization procedure using $\tilde{\mathbf{U}}$ to obtain $(\mathbf{X}_0, \mathbf{Y}_0) = \mathsf{Init}(\tilde{\mathbf{U}})$.

4) Stopping Criterion: For the non-convex approaches (Newton and alternating minimization), we stop the program if either $\|\mathbf{z}_n\|$ converges or $\|\mathbf{z}_n\|$ becomes larger than $\max\{10^6, 10^6\|\mathbf{z}_0\|\}$. When $\|\mathbf{z}_n\|$ becomes larger than $\max\{10^6, 10^6\|\mathbf{z}_0\|\}$, we count it as failure of the algorithm for recovering data (divergence).

III. Convergence Analysis

Since the equations $\mathbf{p}(\mathbf{z}) = \mathbf{0}$ involve only polynomials which are differentiable, the following classical convergence result on Newton's method holds for our proposed algorithm. Let $\mathbf{z}^*$ denote the solution to $\mathbf{p}(\mathbf{z}) = \mathbf{0}$. Moreover, define the error vector at iteration n as $\mathbf{e}_n = \mathbf{z}_n - \mathbf{z}^* \in \mathbb{R}^D$.

Proposition 1: [43] Consider a system of polynomial equations $\mathbf{p}(\mathbf{z}) = \mathbf{0}$. There exists $\epsilon > 0$ such that if $\|\mathbf{e}_0\|_2 < \epsilon$, then we have $\|\mathbf{e}_n\|_2 \to 0$ as $n \to \infty$.

However, in the above classical result, the exact characterization of the convergence neighborhood, i.e., ϵ, is not specified. In this section, we will give an explicit expression for ϵ by exploiting the specific structure of the polynomial set $\mathbf{p}$.

We will first derive a relationship between $\|\mathbf{e}_n\|_2$ and $\|\mathbf{e}_{n+1}\|_2$ through the smallest singular value of a submatrix of $\nabla\mathbf{p}(\mathbf{z}^*)$. Then, by bounding $\|\mathbf{e}_0\|_2$ in terms of this smallest singular value, we can ensure $\|\mathbf{e}_n\|_2 \to 0$ as $n \to \infty$.

From Newton's update given in (6) we can write

$$\mathbf{p}(\mathbf{z}_n) + \nabla\mathbf{p}(\mathbf{z}_n)(\mathbf{z}_{n+1} - \mathbf{z}_n) = \mathbf{0}. \tag{9}$$

On the other hand, using Taylor's expansion and the fact that the polynomials in $\mathbf{p}$ are twice differentiable at any point, we have

$$\mathbf{p}(\mathbf{z}_n) + \nabla\mathbf{p}(\mathbf{z}_n)(\mathbf{z}^* - \mathbf{z}_n) + \mathbf{h}(\mathbf{t}_n) = \mathbf{p}(\mathbf{z}^*) = \mathbf{0}, \tag{10}$$

for some $\mathbf{t}_n$ belonging to a ball centered at $\mathbf{z}^*$ and of radius $|\mathbf{z}^* - \mathbf{z}_n|$, where

$$\begin{aligned}\mathbf{h}(\mathbf{t}_n) &= \Bigg[\frac{1}{2}(\mathbf{z}^* - \mathbf{z}_n)^\top \nabla^2 p_1(\mathbf{t}_n)(\mathbf{z}^* - \mathbf{z}_n), \ldots, \\ &\quad \frac{1}{2}(\mathbf{z}^* - \mathbf{z}_n)^\top \nabla^2 p_{|\Omega|}(\mathbf{t}_n)(\mathbf{z}^* - \mathbf{z}_n)\Bigg]^\top \in \mathbb{R}^{|\Omega|\times 1},\end{aligned} \tag{11}$$

with $\nabla^2 p_i = [\frac{\partial^2 p_i}{\partial z_j \partial z_{j'}}]_{(j,j')} \in \mathbb{R}^{D\times D}$, for $1 \le j, j' \le D$ and $i = 1, \ldots, |\Omega|$.

Using (9) and (10), we conclude that

$$\nabla\mathbf{p}(\mathbf{z}_n)(\mathbf{z}^* - \mathbf{z}_{n+1}) + \mathbf{h}(\mathbf{t}_n) = \mathbf{0}. \tag{12}$$

Note that for the proposed algorithm all polynomials are of the form (5). Consider polynomial p_i and assume that it corresponds to the observed entry $(i', j') \in \Omega$ of $\mathbf{U}$. Then, p_i involves $\mathbf{X}(i', \ell'), \mathbf{Y}(\ell', j')$ for $\ell' = 1, \ldots, r$. Note that $\mathbf{X}(i', :)$ either corresponds to r variables in $\mathbf{z}$, or it is a constant vector corresponding to a row of the identity submatrix in the canonical structure. Hence, $\nabla^2 p_i$ is a binary matrix that is independent of $\mathbf{t}_n$, where there are 1's at entries corresponding to the variables $\mathbf{X}(i', \ell')$ and $\mathbf{Y}(\ell', j')$, and zeros everywhere else.

Denote $\mathbf{e}^i_{Y,n} \in \mathbb{R}^r$ as the components of $\mathbf{e}_n$ that corresponds to the r variables $\mathbf{Y}(\ell', j')$ in p_i, $\ell' = 1, \ldots, r$. Moreover, define $\mathbf{e}^i_{X,n} \in \mathbb{R}^r$ as follows: if $\mathbf{X}(i', \ell')$ is a variable in $\mathbf{z}$, then $\mathbf{e}^i_{X,n}(\ell')$ is the component of $\mathbf{e}_n$ corresponding to this variable in p_i; otherwise $\mathbf{e}^i_{X,n}(\ell') = 0$, $\ell' = 1, \ldots, r$.

Then it is easily verified that $\frac{1}{2}(\mathbf{z}^* - \mathbf{z}_n)^\top \nabla^2 p_i(\mathbf{t}_n)(\mathbf{z}^* - \mathbf{z}_n) = (\mathbf{e}^i_{X,n})^\top \mathbf{e}^i_{Y,n}$. Hence

$$\mathbf{h}(\mathbf{t}_n) = \left[\left(\mathbf{e}^1_{X,n}\right)^\top \mathbf{e}^1_{Y,n}, \ldots, \left(\mathbf{e}^{|\Omega|}_{X,n}\right)^\top \mathbf{e}^{|\Omega|}_{Y,n}\right]^\top \in \mathbb{R}^{|\Omega|}. \tag{13}$$

Therefore, (12) can be simplified as

$$\nabla\mathbf{p}(\mathbf{z}_n)\mathbf{e}_{n+1} = \left[\left(\mathbf{e}^1_{X,n}\right)^\top \mathbf{e}^1_{Y,n}, \ldots, \left(\mathbf{e}^{|\Omega|}_{X,n}\right)^\top \mathbf{e}^{|\Omega|}_{Y,n}\right]^\top. \tag{14}$$

Assume that $\nabla\mathbf{p}(\mathbf{z}_n) \in \mathbb{R}^{|\Omega|\times D}$ is full column-rank. Hence, there exists a set of D polynomials $\tilde{\mathbf{p}} = \{p_{i_1}, \ldots, p_{i_D}\}$ such that

$$\nabla\tilde{\mathbf{p}}(\mathbf{z}_n)\mathbf{e}_{n+1} = \left[\left(\mathbf{e}^{i_1}_{X,n}\right)^\top \mathbf{e}^{i_1}_{Y,n}, \ldots, \left(\mathbf{e}^{i_D}_{X,n}\right)^\top \mathbf{e}^{i_D}_{Y,n}\right]^\top, \tag{15}$$

or

$$\mathbf{e}_{n+1} = (\nabla\tilde{\mathbf{p}}(\mathbf{z}_n))^{-1}\left[\left(\mathbf{e}^{i_1}_{X,n}\right)^\top \mathbf{e}^{i_1}_{Y,n}, \ldots, \left(\mathbf{e}^{i_D}_{X,n}\right)^\top \mathbf{e}^{i_D}_{Y,n}\right]^\top, \tag{16}$$

where $\nabla\tilde{\mathbf{p}}(\mathbf{z}_n) \in \mathbb{R}^{D\times D}$ is a full-rank matrix.

It then follows that

$$\begin{aligned}&\|\mathbf{e}_{n+1}\|_2 \\ &= \left\|(\nabla\tilde{\mathbf{p}}(\mathbf{z}_n))^{-1}\left[\left(\mathbf{e}^{i_1}_{X,n}\right)^\top \mathbf{e}^{i_1}_{Y,n}, \ldots, \left(\mathbf{e}^{i_D}_{X,n}\right)^\top \mathbf{e}^{i_D}_{Y,n}\right]^\top\right\|_2 \\ &\le \sigma_{\max}((\nabla\tilde{\mathbf{p}}(\mathbf{z}_n))^{-1}) \\ &\quad \times \left\|\left[\left(\mathbf{e}^{i_1}_{X,n}\right)^\top \mathbf{e}^{i_1}_{Y,n}, \ldots, \left(\mathbf{e}^{i_D}_{X,n}\right)^\top \mathbf{e}^{i_D}_{Y,n}\right]^\top\right\|_2 \\ &= \frac{1}{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}_n)))} \\ &\quad \times \left\|\left[\left(\mathbf{e}^{i_1}_{X,n}\right)^\top \mathbf{e}^{i_1}_{Y,n}, \ldots, \left(\mathbf{e}^{i_D}_{X,n}\right)^\top \mathbf{e}^{i_D}_{Y,n}\right]^\top\right\|_2.\end{aligned} \tag{17}$$

We can now state our convergence result.

Theorem 1: Consider the matrix $\mathbf{T}$ such that $\mathbf{T}^{-1}\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)\mathbf{T} = \mathbf{\Sigma}^*$ is the diagonal matrix consisting of the singular values of the full-rank matrix $\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)$ and define $\delta = \|\mathbf{T}^{-1}\|_{\mathcal{F}}\|\mathbf{T}\|_{\mathcal{F}}$. Assume that $\|\mathbf{e}_0\|_2 \leq \min\{\frac{\sqrt{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)))}}{2\delta\sqrt{D}}, \frac{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)))}{8r\sqrt{D}}\}$. Then, we have $\|\mathbf{e}_n\|_2 \to 0$ as $n \to \infty$.

Proof: Since each polynomial $p_i(\mathbf{z})$ is second-order as given in (5), $\nabla p_i(\mathbf{z})$ is a vector with entries either equal to zero or one of the elements of $\mathbf{z}$. For example, if $\mathbf{z} \in \mathbb{R}^6$ and $p_i(\mathbf{z}) = z_1z_4 + z_2z_5 - 4 = 0$, then we have $\nabla p_i(\mathbf{z}) = [z_4, z_5, 0, z_1, z_2, 0]$. Define $\mathbf{E}_n = \nabla\tilde{\mathbf{p}}(\mathbf{z}_n) - \nabla\tilde{\mathbf{p}}(\mathbf{z}^*)$. It is easy to observe that each element of $\mathbf{E}_n$ is either zero or one element of the error vector $\mathbf{e}_n$. As a result, $\|\nabla p_i(\mathbf{z}_n) - \nabla p_i(\mathbf{z}^*)\|_2 \leq \|\mathbf{e}_n\|_2$, and therefore $\|\mathbf{E}_n\|_{\mathcal{F}} \leq \sqrt{D}\|\mathbf{e}_n\|_2$. Hence, $\nabla\tilde{\mathbf{p}}(\mathbf{z}_n) = \nabla\tilde{\mathbf{p}}(\mathbf{z}^*) + \mathbf{E}_n$, where $\|\mathbf{E}_n\|_{\mathcal{F}} \leq \sqrt{D}\|\mathbf{e}_n\|_2$.

The perturbation analysis [44] results that for any eigenvalue λ of $\nabla\tilde{\mathbf{p}}(\mathbf{z}_n)$, there exists $i \in \{1, \ldots, D\}$ such that

$$|\lambda - \lambda_i| \leq \|\mathbf{T}^{-1}\|_{\mathcal{F}}\|\mathbf{T}\|_{\mathcal{F}}\|\mathbf{E}_n\|_{\mathcal{F}}, \tag{18}$$

where λ_i's are the eigenvalues of $\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)$ and $\mathbf{T}^{-1}\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)\mathbf{T} = \mathbf{\Sigma}^*$ is the diagonal matrix consisting of the singular values of $\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)$. Recall that $\|\mathbf{E}_n\|_{\mathcal{F}} \leq \sqrt{D}\|\mathbf{e}_n\|_2$, and according to the assumption we have $\|\mathbf{e}_n\|_2 \leq \frac{\sqrt{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)))}}{2\delta\sqrt{D}}$, with $\delta = \|\mathbf{T}^{-1}\|_{\mathcal{F}}\|\mathbf{T}\|_{\mathcal{F}}$. Therefore, for any eigenvalue λ of $\nabla\tilde{\mathbf{p}}(\mathbf{z}_n)$, there exists $i \in \{1, \ldots, D\}$ such that

$$|\lambda - \lambda_i| \leq \frac{\sqrt{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)))}}{2} = \frac{1}{2}\min_{1\leq j\leq D}|\lambda_j|. \tag{19}$$

It simply follows from (19) that for any eigenvalue λ of $\nabla\tilde{\mathbf{p}}(\mathbf{z}_n)$ we have $|\lambda| \geq \frac{1}{2}\min_{1\leq j\leq D}|\lambda_j|$, and therefore

$$\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}_n))) \geq \frac{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)))}{4}. \tag{20}$$

Therefore, $\nabla\tilde{\mathbf{p}}(\mathbf{z}_n)$ is full-rank, i.e., $\nabla\mathbf{p}(\mathbf{z}_n)$ is full column-rank. Then, using (17) we conclude

$$\|\mathbf{e}_{n+1}\|_2 \leq \frac{4}{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)))} \times \left\|\left[\left(\mathbf{e}_{X,n}^{i_1}\right)^\top \mathbf{e}_{Y,n}^{i_1}, \ldots, \left(\mathbf{e}_{X,n}^{i_D}\right)^\top \mathbf{e}_{Y,n}^{i_D}\right]^\top\right\|_2. \tag{21}$$

On the other hand, by definition, each element of $\mathbf{e}_{X,n}^i$ and $\mathbf{e}_{Y,n}^i$ is either 0 or an element of $\mathbf{e}_n$. Hence we can write

$$\left\|\left[\left(\mathbf{e}_{X,n}^{i_1}\right)^\top \mathbf{e}_{Y,n}^{i_1}, \ldots, \left(\mathbf{e}_{X,n}^{i_D}\right)^\top \mathbf{e}_{Y,n}^{i_D}\right]^\top\right\|_2 \leq \sqrt{Dr^2\left(\max_{1\leq j\leq D}\{|\mathbf{e}_n(j)|\}\right)^4}, \tag{22}$$

where $\mathbf{e}_n(j)$ denotes the j-th element of the vector $\mathbf{e}_n$. As a result, we have

$$\|\mathbf{e}_{n+1}\|_2 \overset{(a)}{\leq} \frac{4r\sqrt{D}\|\mathbf{e}_n\|_2^2}{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)))} \overset{(b)}{<} \frac{\|\mathbf{e}_n\|_2}{2}, \tag{23}$$

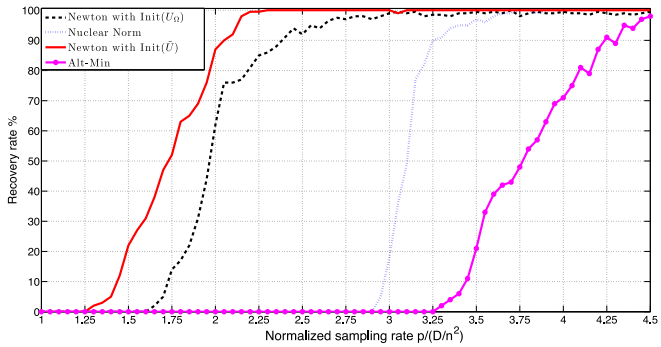


Fig. 1. Comparison of recovery rates for $\mathbf{U} \in \mathbb{R}^{500\times500}$ of rank 3.

where (a) follows from the fact that $(\max_{1\leq j\leq D}\{|\mathbf{e}_n(j)|\})^2 \leq \|\mathbf{e}_n\|_2^2$ and (b) follows from the assumption that $\|\mathbf{e}_n\|_2 \leq \frac{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)))}{8r\sqrt{D}}$.

Starting from $\mathbf{e}_0$ that satisfies the condition in the statement of the theorem, we have $\|\mathbf{e}_1\|_2 \leq \frac{\|\mathbf{e}_0\|_2}{2}$. Hence $\mathbf{e}_1$ also satisfies the condition and $\|\mathbf{e}_2\|_2 \leq \frac{\|\mathbf{e}_1\|_2}{2}$, and so on. Therefore $\|\mathbf{e}_n\|_2 \to 0$ as $n \to \infty$. ■

Corollary 1: Consider the matrix $\mathbf{T}$ such that $\mathbf{T}^{-1}\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)\mathbf{T} = \mathbf{\Sigma}^*$ is the diagonal matrix consisting of the singular values of the full-rank matrix $\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)$ and define $\delta = \|\mathbf{T}^{-1}\|_{\mathcal{F}}\|\mathbf{T}\|_{\mathcal{F}}$. Assume that $\|\mathbf{e}_0\|_2 \leq \min\{\frac{\sqrt{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)))}}{2\delta\sqrt{D}}, \frac{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)))}{8r\sqrt{D}}\}$. Then, $\nabla\tilde{\mathbf{p}}(\mathbf{z}_n)$ is full-rank, $i = 1, 2, \ldots$. Hence, $\nabla\mathbf{p}(\mathbf{z}_n)$ is full column-rank, $i = 1, 2, \ldots$.

Proof: The proof is straightforward since (20) simply results that $\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}_n))) > 0$. ■

IV. Numerical Results

A. Noiseless Matrix

Here, we are interested in generating a random low-rank matrix to randomly sample it by a sampling probability and compare the proposed algorithms with the existing algorithms in the literature. To this end, we first generate $\mathbf{X} \in \mathbb{R}^{n_1\times r}$ and $\mathbf{Y} \in \mathbb{R}^{r\times n_2}$ by choosing each entry of them uniformly from some interval on real numbers, e.g., [1, 10]. Then, we generate a random $n_1 \times n_2$ matrix of rank r, i.e., $\mathbf{U} = \mathbf{XY}$. We sample each entry independently with probability $0 < p < 1$. We say the sampled matrix $\mathbf{U}$ is recovered if $\frac{\|\hat{\mathbf{U}}-\mathbf{U}\|_{\mathcal{F}}}{\|\mathbf{U}\|_{\mathcal{F}}} < 0.01$, where $\hat{\mathbf{U}}$ denotes the obtained matrix through the corresponding completion algorithm. For each value of p, we complete 200 matrices and calculate the average recovery rate of each algorithm.

For the first example, we consider a 500×500 matrix of rank 3. The dimension of the manifold is $D = n_1r + n_2r - r^2 = 2991$. Since there are D unknowns, at least D samples are needed and therefore $\frac{D}{n_1n_2} = \frac{2991}{250000} = 0.011964$ is the absolute lower bound on the sampling rate. Figure 1 shows the recovery rate of different completion algorithms in terms of the normalized sampling rate $\frac{p}{D/(n_1n_2)}$. We have the following observations:

- Our proposed Newton's method with initialization $\mathsf{Init}(\tilde{\mathbf{U}})$ outperforms all other methods in the sense of requiring the

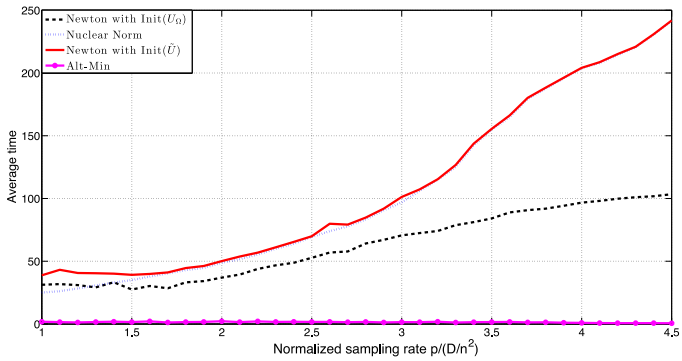


Fig. 2. Comparison of running time for completing $\mathbf{U} \in \mathbb{R}^{500\times 500}$ of rank 3.

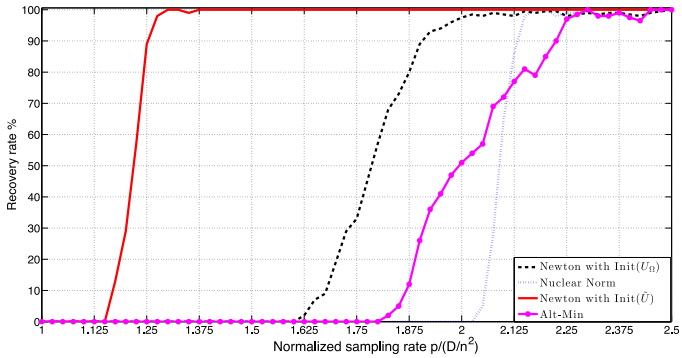


Fig. 3. Comparison of recovery rates for $\mathbf{U} \in \mathbb{R}^{200\times 200}$ of rank 10.

least number of samples to recover the original matrix $\mathbf{U}$. Newton's method with initialization $\mathsf{Init}(\mathbf{U}_\Omega)$ is the second best and outperforms nuclear norm minimization and alternating minimization.

- With $2D$ samples the proposed Newton's method can achieve recovery rates of 0.9 and 0.65 using the initializations $\mathsf{Init}(\tilde{\mathbf{U}})$ and $\mathsf{Init}(\mathbf{U}_\Omega)$, respectively; whereas the recovery rate is zero for the conventional methods (i.e., nuclear norm minimization and alternating minimization).
- The average number of samples required for a recovery rate of at least 90% is $2.3D$, $3.25D$ and $4.25D$ using Newton's method with initialization $\mathsf{Init}(\mathbf{U}_\Omega)$, nuclear norm minimization and alternating minimization, respectively. This leads to $\frac{3.25-2.3}{3.25} \times 100\% = 29.2\%$ and $\frac{4.25-2.3}{4.25} = 45.8\%$ reductions in the required number of samples, respectively.

We also show the average running time for each algorithm in Figure 2. It is seen that alternating minimization is the fastest method with running time almost independent of the sample size. The proposed Newton's method with initialization $\mathsf{Init}(\mathbf{U}_\Omega)$ is significantly faster than nuclear norm minimization especially when the sample size is large.

For the second example, we consider a 200×200 matrix of rank 10. The dimension is $D = n_1 r + n_2 r - r^2 = 3900$ and the lower bound on the sampling rate is $\frac{D}{n_1 n_2} = \frac{3900}{40000} = 0.0975$. Figure 3 shows the recovery rates for different completion algorithms. We have the following observations:

- Newton's method with $\mathsf{Init}(\mathbf{U}_\Omega)$ outperforms nuclear norm minimization and alternating minimization. The average number of samples required for a recovery rate of at least 90% is $1.9D$, $2.125D$ and $2.2D$ using Newton's method with $\mathsf{Init}(\mathbf{U}_\Omega)$, nuclear norm minimization and alternating minimization, respectively. This leads to $\frac{2.125-1.9}{2.125} \times 100\% = 10.5\%$ and $\frac{2.2-1.9}{2.2} = 13.6\%$ reductions in the required number of samples, respectively.
- Using Newton's method with $\mathsf{Init}(\tilde{\mathbf{U}})$, we are able to recover the data with $1.3D$ samples with probability of almost one, whereas $2D$, $2.15D$, $2.25D$ samples are required for Newton's method with $\mathsf{Init}(\mathbf{U}_\Omega)$, nuclear norm minimization and alternating minimization, respectively. This leads to $\frac{2-1.3}{2} \times 100\% = 35\%$, $\frac{2.15-1.3}{2.15} \times 100\% = 39.5\%$ and $\frac{2.25-1.3}{2.25} \times 100\% = 42.2\%$ reductions in the required number of samples, respectively.

In these comparisons, we have compared the non-convex approaches that are able to incorporate and take advantage of the rank value with nuclear norm minimization. Note that the calculated time for Newton's method with Init$(\tilde{\mathbf{U}})$ includes the time spent for solving the nuclear norm minimization.

B. Noisy Matrix

1) Additive Gaussian Noise: We first generate a random matrix $\mathbf{U} \in \mathbb{R}^{n_1\times n_2}$ of rank r and then we add a noise matrix $\mathbf{E}$ to $\mathbf{U}$, where $\mathbf{E}$ consists of i.i.d. $\mathcal{N}(0, \sigma^2)$ entries. We sample each entry of $\mathbf{E} + \mathbf{U}$ with probability p. We use the peak signal-to-noise ratio (PSNR) as the completion performance metric, defined as

$$\mathrm{PSNR}\left(\hat{\mathbf{U}}, \mathbf{U}\right) = 20\log_{10}(M) - 10\log_{10}\left(\frac{1}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\left(\hat{\mathbf{U}}(i,j) - \mathbf{U}(i,j)\right)^2\right), \quad (24)$$

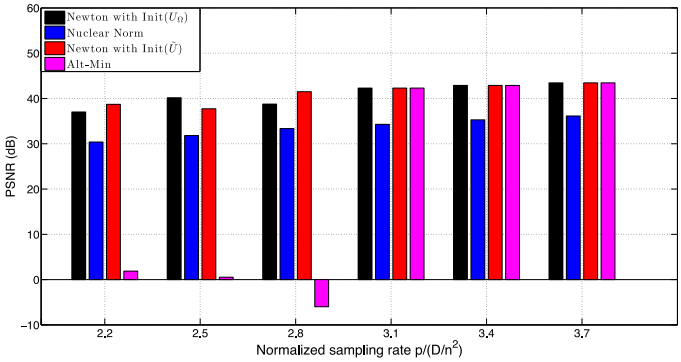
where M denotes upper bound of the entries in $\mathbf{U}$ (since $\mathbf{U} = \mathbf{XY}$ and each element of $\mathbf{X}$ and $\mathbf{Y}$ is uniform in [1, 10], we have $M = 100r$ here).

We consider $\mathbf{U} \in \mathbb{R}^{500\times 500}$ of rank 3 with $D = n_1 r + n_2 r - r^2 = 2991$ and $\frac{D}{n_1 n_2} = 0.011964$. The PSNR performances of different completion algorithms are shown in Figures 4(a) and 4(b) for two SNR (defined as $10\log_{10}\left(\frac{\sigma_U}{\sigma}\right)$, where σ_U denotes the variance of each entry of $\mathbf{U}$) values 17 dB and 12.5 dB, respectively. As we can observe, for the non-convex approaches (Newton's method and alternating minimization) performance becomes less stable as we decrease SNR. However, for moderate SNR, Newton's method mainly outperforms alternating minimization and nuclear norm minimization with small number of samples.

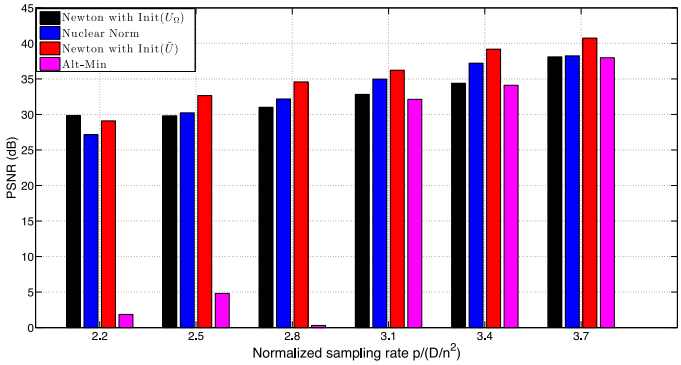
2) Netflix Rating Table Real Data (Recommendation Systems): The "MovieLens-1M" dataset found online at "grouplens" website includes 6040 rows representing the users and 3706 columns representing the movies and each entry is between 1 and 5, which represents the rating of a particular user for a particular movie from Netflix rating table. For each single experiment we choose a random 2000×2000 submatrix of this dataset and sample each entry with probability p. Note that for the non-convex approaches, i.e., Newton's method and

TABLE I
COMPARISON OF RMSE FOR MOVIELENS DATA

Algorithm	$p = 0.05$	$p = 0.07$	$p = 0.09$	$p = 0.11$	$p = 0.13$
Newton's method	1.4019	1.2401	1.1395	1.0871	1.0498
Alternating minimization	24.2515	8.2742	6.9766	2.7103	1.9240
Nuclear norm minimization	1.6027	1.4373	1.3282	1.2648	1.2139



(a) SNR = 17 dB.



(b) SNR = 12.5 dB.

Fig. 4. Comparison of PSNR for $\mathbf{U} \in \mathbb{R}^{500\times 500}$ of rank 3.

alternating minimization, we need a rank value. We start with a small number, e.g., $r = 1$ and if the algorithm diverges, we increase r and repeat this trial a few times.

We use the root-mean-square error (RMSE) for the unobserved entries as the performance metric, given by

$$\text{RMSE}\left(\hat{\mathbf{U}}, \mathbf{U}\right) = \sqrt{\frac{1}{|\bar{\Omega}|}\sum\nolimits_{(i,j)\in\bar{\Omega}} \left(\hat{\mathbf{U}}(i,j) - \mathbf{U}(i,j)\right)^2}, \tag{25}$$

where $\bar{\Omega}$ denotes the set of unobserved entries. For each value of p, we repeat the experiment 20 times and obtain the average RMSE. For this example, Newton's method with the two different initializations give the same solution. The results are summarized in Table I. It is seen that the proposed Newton's method outperforms both alternating minimization and nuclear norm minimization.

In this example, the average running times of one experiment for Newton's method with $\mathsf{Init}(\mathbf{U}_\Omega)$, alternating minimization, nuclear norm minimization and Newton's method with $\mathsf{Init}(\tilde{\mathbf{U}})$ are 12, 9, 457 and 470 seconds, respectively.

V. EXTENSIONS

A. Matrix Sensing

Denote the inner product between two $n_1 \times n_2$ matrices $\mathbf{U}$ and $\mathbf{M}$ as $\langle \mathbf{U}, \mathbf{M}\rangle = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \mathbf{U}(i,j)\mathbf{M}(i,j)$. The matrix sensing problem is to recover a low-rank matrix $\mathbf{U}$ with $\text{rank}(\mathbf{U}) = r$, given a set of linear measurements $\langle \mathbf{U}, \mathbf{M}_i\rangle = b_i$, $i = 1, \ldots, L$, where L denotes the total number of measurements. Hence we have a set of L polynomial equations $\mathbf{p}(\mathbf{z}) = \mathbf{0}$, whose elements are $p_i(\mathbf{z}) = \langle \mathbf{XY}, \mathbf{M}_i\rangle - b_i$.

In the alternating minimization approach, we again write $\mathbf{U} = \mathbf{XY}$ such that $\mathbf{X} \in \mathbb{R}^{n_1\times r}$ and $\mathbf{Y} \in \mathbb{R}^{r\times n_2}$. Starting with some initial $\mathbf{X}_0$ and $\mathbf{Y}_0$ ($\mathsf{Init}(\mathbf{U}_M)$ described later), at the k-th iteration, given $\mathbf{X}_{k-1}$ and $\mathbf{Y}_{k-1}$, we first update $\mathbf{X}_k$ by solving the following convex program

$$\underset{\mathbf{X}_k\in\mathbb{R}^{n_1\times r}}{\text{minimize}} \quad \sqrt{\sum\nolimits_{i=1}^{L} \left(\langle \mathbf{X}_k\mathbf{Y}_{k-1}, \mathbf{M}_i\rangle - b_i\right)^2}, \tag{26}$$

and then update $\mathbf{Y}_k$ by solving

$$\underset{\mathbf{Y}_k\in\mathbb{R}^{r\times n_2}}{\text{minimize}} \quad \sqrt{\sum\nolimits_{i=1}^{L} \left(\langle \mathbf{X}_k\mathbf{Y}_{k}, \mathbf{M}_i\rangle - b_i\right)^2}. \tag{27}$$

The iteration continues until both errors are below certain threshold. In the nuclear norm minimization approach, we solve the following relaxed convex problem

$$\begin{aligned} \underset{\mathbf{U}'\in\mathbb{R}^{n_1\times n_2}}{\text{minimize}} \quad & \|\mathbf{U}'\|_* \\ \text{subject to} \quad & \langle \mathbf{U}', \mathbf{M}_i\rangle = b_i\,, \quad i = 1, \ldots, L. \end{aligned} \tag{28}$$

Similar to the algorithm described in Section II-C, we can use Newton's method to solve the system of polynomial equations $\mathbf{p}(\mathbf{z}) = \mathbf{0}$. We use the same initialization for alternating minimization as in [19] for the matrix sensing problem. We define the matrix $\mathbf{U}_M = \sum_{i=1}^{L} b_i\mathbf{M}_i$. Then, we apply the initialization procedure described in Section II-C using $\mathbf{U}_M$ instead of $\mathbf{U}_\Omega$ and denote this initialization by $\mathsf{Init}(\mathbf{U}_M)$.

Or we can first obtain a completion $\tilde{\mathbf{U}}$ by using the nuclear norm minimization method, and then apply the initialization procedure in Section II-C using $\tilde{\mathbf{U}}$ to obtain $(\mathbf{X}_0, \mathbf{Y}_0) = \mathsf{Init}(\tilde{\mathbf{U}})$.

Similar to Theorem 1, we can characterize the convergence region of the Newton's algorithm for matrix sensing as follows.

Theorem 2: Consider the matrix $\mathbf{T}$ such that $\mathbf{T}^{-1}\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)\mathbf{T} = \mathbf{\Sigma}^*$ is the diagonal matrix consisting of the singular values of $\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)$ and define $\delta = \|\mathbf{T}^{-1}\|_{\mathcal{F}}\|\mathbf{T}\|_{\mathcal{F}}$. Suppose that the number of non-zero elements of $\mathbf{M}_i$ (denoted by K_i), i.e., $\|\mathbf{M}_i\|_0$, is upper bounded by a constant number K, $i = 1, \ldots, L$. Assume that $\|\mathbf{e}_0\|_2 \le \min\{\frac{\sqrt{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)))}}{2\delta\sqrt{D}}, \frac{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)))}{8rK^2\sqrt{D}}\}$. Then, we have $\|\mathbf{e}_n\|_2 \to 0$ as $n \to \infty$.

Proof: The proof is similar to the proof of Theorem 1 with the only difference that each polynomial in this scenario can be written as a linear combination of at most K polynomials in the matrix completion scenario. It is easily verified that for matrix sensing (12) holds true. Moreover, instead of (13) we have the following

$$\mathbf{h}(\mathbf{t}_n) = \left[\left(\mathbf{w}_{X,n}^1\right)^\top \mathbf{w}_{Y,n}^1, \ldots, \left(\mathbf{w}_{X,n}^L\right)^\top \mathbf{w}_{Y,n}^L\right]^\top \in \mathbb{R}^L, \quad (29)$$

where $\mathbf{w}_{X,n}^i = \sum_{j=1}^{K_i} \mathbf{e}_{X,n}^j \in \mathbb{R}^r$ for each of the K_i involved polynomials in the i-th linear measurement. Recall that we have $K_i \leq K$. Therefore, (21) becomes as follows

$$\|\mathbf{e}_{n+1}\|_2 \leq \frac{4}{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}_n)))} \times \left\|\left[\left(\mathbf{w}_{X,n}^{i_1}\right)^\top \mathbf{w}_{Y,n}^{i_1}, \ldots, \left(\mathbf{w}_{X,n}^{i_D}\right)^\top \mathbf{w}_{Y,n}^{i_D}\right]^\top\right\|_2. \quad (30)$$

Note that since $K_i \leq K$, each element of $\mathbf{h}(\mathbf{t}_n)$ changes to the summation of at most K terms in the form of $(\mathbf{e}_{X,n}^1)^\top \mathbf{e}_{Y,n}^1$. As a result, the RHS of (22) becomes $\sqrt{Dr^2K^4\left(\max_{1\leq j\leq D}\{|\mathbf{e}_n(j)|\}\right)^4}$, i.e., we have

$$\left\|\left[\left(\mathbf{w}_{X,n}^{i_1}\right)^\top \mathbf{w}_{Y,n}^{i_1}, \ldots, \left(\mathbf{w}_{X,n}^{i_D}\right)^\top \mathbf{w}_{Y,n}^{i_D}\right]^\top\right\|_2 \leq \sqrt{Dr^2K^4\left(\max_{1\leq j\leq D}\{|\mathbf{e}_n(j)|\}\right)^4}. \quad (31)$$

Therefore (23) changes to

$$\|\mathbf{e}_{n+1}\|_2 \leq K^2 \frac{4r\sqrt{D}\|\mathbf{e}_n\|_2^2}{\sigma_{\min}((\nabla\tilde{\mathbf{p}}(\mathbf{z}^*)))} \overset{(a)}{<} \frac{\|\mathbf{e}_n\|_2}{2}, \quad (32)$$

where (a) follows from the modified assumption on $\|\mathbf{e}_0\|_2$. ■

1) Column-Wise Measurements: In the first experiment, we assume that each matrix $\mathbf{M}_i$ has only one non-zero column, which is chosen uniformly from the n_2 columns. The n_1 elements of the non-zero column are i.i.d. uniformly chosen from the interval $[-1, 1]$. We say the matrix $\mathbf{U}$ is recovered if $\frac{\|\hat{\mathbf{U}}-\mathbf{U}\|_\mathcal{F}}{\|\mathbf{U}\|_\mathcal{F}} < 0.01$, where $\hat{\mathbf{U}}$ denotes the output of the corresponding algorithm. For each value of L, we repeat the experiment 200 times and calculate the average recovery rates of different algorithms.

For the first example, we consider a 50×50 matrix $\mathbf{U}$ of rank 3. The dimension of corresponding manifold $D = n_1 r + n_2 r - r^2 = 291$. Figure 5 shows the recovery rates of different algorithms in terms of the normalized sample size $\frac{L}{D}$. It is seen that the Newton's method with both initialization schemes outperforms the nuclear norm minimization and alternating minimization.

In these comparisons, we have compared the non-convex approaches that are able to incorporate and take advantage of the rank value with nuclear norm minimization. Note that the calculated time for Newton's method with Init($\tilde{\mathbf{U}}$) includes the time spent for solving the nuclear norm minimization.

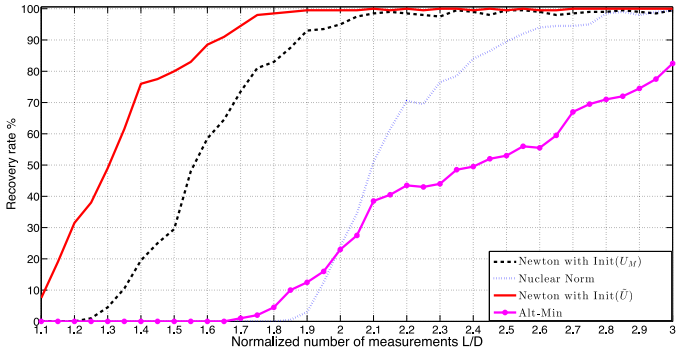


Fig. 5. Comparison of recovery rates for $\mathbf{U} \in \mathbb{R}^{50\times 50}$ of rank 3.

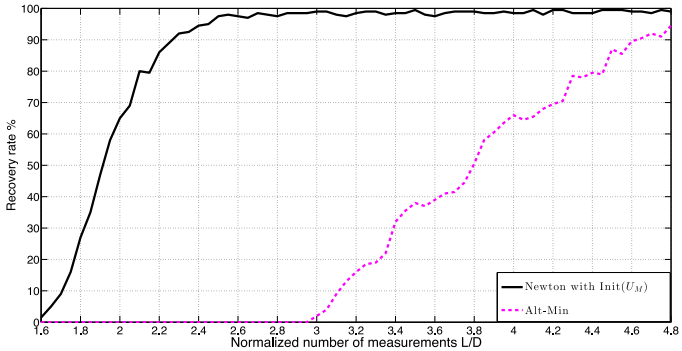


Fig. 6. Comparison of recovery rates for $\mathbf{U} \in \mathbb{R}^{400\times 400}$ of rank 3.

For the second numerical experiment, we consider randomly generated $\mathbf{U} \in \mathbb{R}^{400\times 400}$ of rank 3 and hence $D = n_1 r + n_2 r - r^2 = 2391$. Since nuclear norm minimization is significantly slower than Newton's method with Init($\mathbf{U}_M$) and alternating minimization (it almost gets stuck using a regular laptop) for this experiment, we only compare the two non-convex algorithms in Figure 6. Again the Newton's method significantly outperforms alternating minimization.

In this example, the average running times of one experiment for Newton's method with Init($\mathbf{U}_M$) and alternating minimization are 702 and 129 seconds, respectively (we stopped nuclear norm minimization after 2 hours, as it takes a very long time to run on a regular computer).

Then we add a noise matrix $\mathbf{E} \in \mathbb{R}^{400\times 400}$ to $\mathbf{U}$, where $\mathbf{E}$ consists of i.i.d. $\mathcal{N}(0, \sigma^2)$ entries. Again, we use the PSNR as the performance metric for matrix sensing problem. The PSNR performances of different algorithms are shown in Figure 7 for SNR = 18 dB.

2) Random Measurements: In this experiment, each entry of $\mathbf{M}_i$ is non-zero with probability p' independent of other entries. And nonzero entries are chosen uniformly from the interval $[-1, 1]$.

We consider $\mathbf{U} \in \mathbb{R}^{200\times 200}$ of rank 3. Note that $D = n_1 r + n_2 r - r^2 = 1191$. Observe that $p' = \frac{1}{200}$ results in an average $C = p' n_1 n_2 = 200$ nonzero entries, which is the number of entries of each column. In order to observe the impact of p' on the recovery rate, we provide the results for $p' = 2 \times \frac{1}{200}$ and $p' = 0.5 \times \frac{1}{200}$ for different values of L. Since nuclear norm minimization is significantly slower than Newton's method and

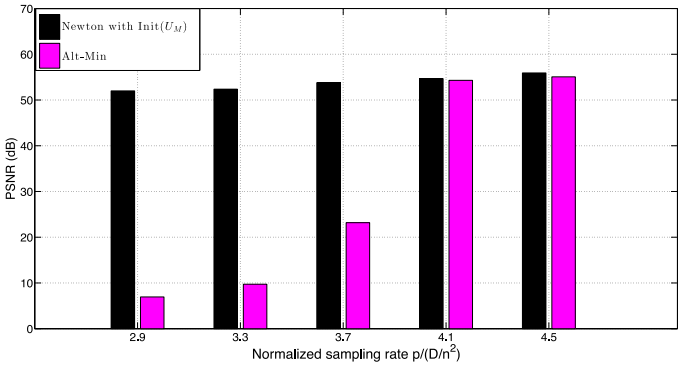
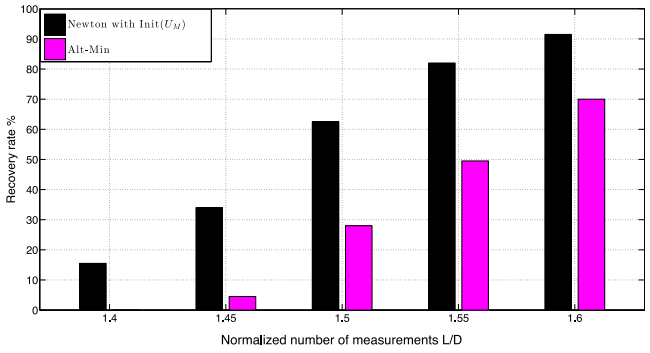
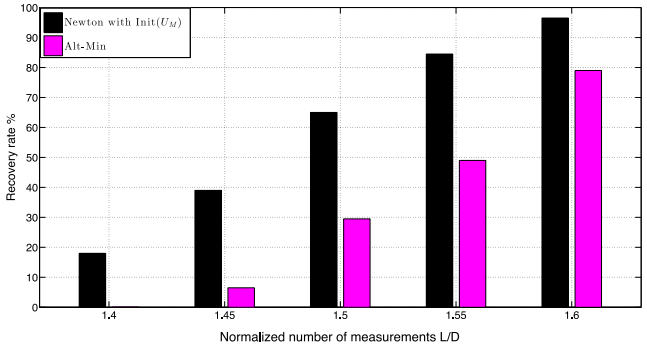


Fig. 7. Comparison of PSNR for $\mathbf{U} \in \mathbb{R}^{400\times 400}$ of rank 3 for SNR $= 18$ dB.



(a) $C = 400$ ($p' = 0.01$).



(b) $C = 100$ ($p' = 0.0025$).

Fig. 8. Recovery rate for $\mathbf{U} \in \mathbb{R}^{200\times 200}$ of rank 3.

alternating minimization for this example, we only compare the latter two methods in Figures 8(a) and 8(b).

B. Tensor Completion

Assume that a d-way tensor $\mathcal{U} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ is sampled. The CP rank of a tensor $\mathcal{U}$ is defined as the minimum number r such that there exist $\mathbf{a}_i^l \in \mathbb{R}^{n_i}$ for $1 \le i \le d$ and $1 \le l \le r$, such that

$$\mathcal{U} = \sum_{l=1}^{r} \mathbf{a}_1^l \otimes \mathbf{a}_2^l \otimes \cdots \otimes \mathbf{a}_d^l, \tag{33}$$

or equivalently,

$$\mathcal{U}(x_1, x_2, \ldots, x_d) = \sum_{l=1}^{r} \mathbf{a}_1^l(x_1)\mathbf{a}_2^l(x_2)\ldots\mathbf{a}_d^l(x_d), \tag{34}$$

where $\otimes$ denotes the tensor product (outer product) and $\mathbf{a}_i^l(x_i)$ denotes the x_i-th entry of vector $\mathbf{a}_i^l$. Note that $\mathbf{a}_1^l \otimes \mathbf{a}_2^l \otimes \cdots \otimes \mathbf{a}_d^l \in \mathbb{R}^{n_1\times\cdots\times n_d}$ is a rank-1 tensor, $l = 1, 2, \ldots, r$. Denote Ω as the set of indices such that $\vec{x} \in \Omega$ if $\mathcal{U}(\vec{x})$ is observed, where $\mathcal{U}(\vec{x})$ represents an entry of tensor $\mathcal{U}$ with coordinate $\vec{x} = (x_1, \ldots, x_d)$. Moreover, define $\mathcal{U}_\Omega$ as the tensor obtained from sampling $\mathcal{U}$ according to Ω, i.e.,

$$\mathcal{U}_\Omega(\vec{x}) = \begin{cases} \mathcal{U}(\vec{x}) & \text{if } \vec{x} \in \Omega, \\ 0 & \text{if } \vec{x} \notin \Omega. \end{cases} \tag{35}$$

In the alternating minimization approach, we write $\mathcal{U} = \sum_{l=1}^{r} \mathbf{a}_1^l \otimes \mathbf{a}_2^l \otimes \cdots \otimes \mathbf{a}_d^l$ such that $\mathbf{a}_i^l \in \mathbb{R}^{n_i}$ for $1 \le i \le d$ and $1 \le l \le r$. Starting with some initial $\mathbf{a}_{i_0}^l \in \mathbb{R}^{n_i}$ for $1 \le i \le d$ and $1 \le l \le r$ (described later), at the k-th iteration, we update all $\mathbf{a}_{i,k}^l$'s by solving the following convex programs

$$\underset{\mathbf{a}_{i',k}^{l'} \in \mathbb{R}^{n_i}}{\text{minimize}} \left\| \mathcal{U}_\Omega - \left(\sum_{l=1}^{r} \mathbf{a}_{1,k_{(1,l)}}^l \otimes \mathbf{a}_{2,k_{(2,l)}}^l \otimes \cdots \otimes \mathbf{a}_{d,k_{(d,l)}}^l \right)_\Omega \right\|_{\mathcal{F}}, \tag{36}$$

where $k_{(i,l)} = k$ if $i \le i'$ and $l \le l'$ and $k_{(i,l)} = k - 1$ otherwise. The iteration continues until both errors are below certain threshold.

Define $\mathbf{z} \in \mathbb{R}^{(n_1+\ldots+n_d)r}$ as the vector that contains $\mathbf{a}_i^l$ for $1 \le i \le d$ and $1 \le l \le r$. Then the set of observed entries result in a set of $|\Omega|$ r-th order polynomial equations $\mathbf{p}$ such that each polynomial has the form

$$p_i(\mathbf{z}) = \sum_{l=1}^{r} \mathbf{a}_1^l(x_1)\mathbf{a}_2^l(x_2)\ldots\mathbf{a}_d^l(x_d) - \mathcal{U}(\vec{\mathbf{x}}), \qquad \vec{\mathbf{x}} \in \Omega. \tag{37}$$

Again we can apply Newton's method described in Section II-C to solve the set of equations $\mathbf{p}(\mathbf{z}) = \mathbf{0}$. In particular, for initialization, we use the Matlab toolbox "Tensorlab" found online to calculate the CP decomposition of $\mathcal{U}_\Omega$ and the leading r rank-1 components to obtain $\mathbf{z}_0$.

1) Noiseless Data: For the numerical experiments, we consider $\mathcal{U} \in \mathbb{R}^{100\times 100\times 100}$ of rank 3 (by generating random vectors $\mathbf{a}_i^l \in \mathbb{R}^{100}$ for $1 \le i \le 3$ and $1 \le l \le 3$) and hence the number of unknowns is $D = n_1 r + n_2 r + n_3 r = 900$. The lower bound on the sampling probability is $\frac{D}{n_1 n_2 n_3} = \frac{900}{10^6} = 0.0009$. Figure 9 shows the recovery rate comparison between Newton's method and alternating minimization, as a function of the normalized sampling rate $\frac{p}{D/(n_1 n_2 n_3)}$.

Note that the average number of samples required for a recovery rate of at least 50% is $14.9D$ and $17D$ using Newton's method with $\mathsf{Init}(\mathbf{U}_\Omega)$ and alternating minimization, respectively. This leads to $\frac{17-14.9}{17} \times 100\% = 12.3\%$ reduction in the required number of samples. Moreover, the average running times of one experiment for Newton's method with $\mathsf{Init}(\mathbf{U}_\Omega)$ and alternating minimization are 152 and 72 seconds, respectively.

2) Noisy Data: We again consider a tensor $\mathcal{U} \in \mathbb{R}^{100\times 100\times 100}$ of rank 3. Then, we add a noise tensor $\mathcal{E}$ to $\mathcal{U}$,

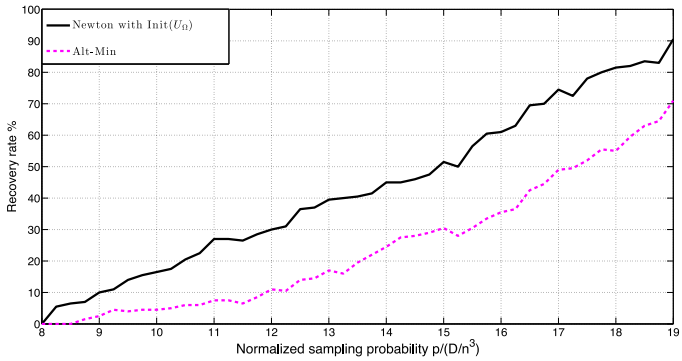


Fig. 9. Comparison of recovery rates for $\mathcal{U} \in \mathbb{R}^{100\times100\times100}$ of rank 3.

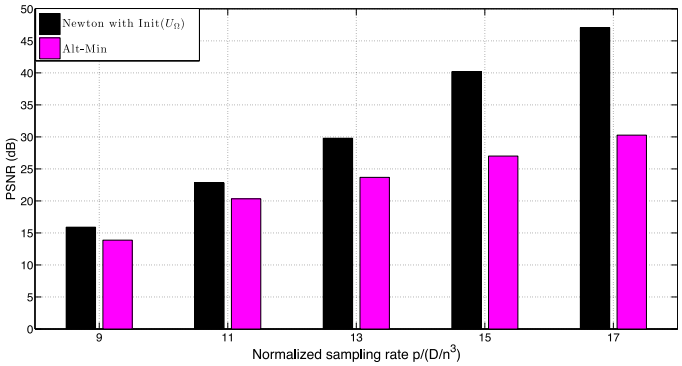


Fig. 10. Comparison of PSNR for $\mathcal{U} \in \mathbb{R}^{100\times100\times100}$ of rank 3 for SNR $= 22.4$ dB.

where $\mathcal{E}$ consists of i.i.d. $\mathcal{N}(0, \sigma^2)$ entries. We sample each entry of $\mathcal{E} + \mathcal{U}$ with probability p and use the PSNR as the performance metric of each algorithm in recovering the data $\mathcal{U}$. Figure 10 shows the PSNR of different algorithms in terms of the normalized sampling probability $\frac{p}{D/(n_1 n_2 n_3)} = \frac{p}{0.0009}$ for SNR $= 22.4$ dB. As we can observe, Newton's method outperforms alternating minimization.

VI. Conclusions

We have studied the problem of retrieving a partially sampled low-rank data, when the sampling rate is very close to the information-theoretic bounds on the sampling rate for existence of a unique solution, i.e., we have considered the problem of low-rank data completion where the number of samples is comparable to the dimension of the corresponding manifold. By using rank factorization, each observed entry gives a polynomial equation of the factor entries and the solution to the set of such polynomial equations constitutes a completion of the data. We have proposed to employ Newton's method to solve the set of polynomial equations. The convergence regions of the proposed Newton's methods for matrix completion and matrix sensing are analytically characterized. Extensive numerical results have been provided to demonstrate that the proposed Newton's method for data completion outperforms the well-known existing methods, such as nuclear norm minimization and alternating minimization, especially when the sampling rate is very low, i.e., comparable to the normalized dimension of the corresponding manifold. Our numerical experiments include both noiseless and noisy made up low-rank data (both two-dimensional data and higher dimensions) and also real-world low-rank data. Moreover, the proposed method is significantly faster than nuclear norm minimization but slower than alternating minimization. Hence, our proposed method is the most efficient when the sampling rate is very low and almost close to the information-theoretic bounds for existence of a unique completion of the sampled data.

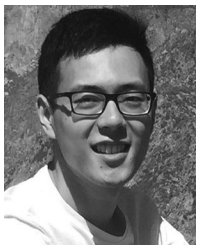
References

[1] E. J. Candès, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 199–225, 2013.
[2] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust video denoising using low rank matrix completion," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2010, pp. 1791–1798.
[3] N. J. Harvey, D. R. Karger, and K. Murota, "Deterministic network coding by matrix completion," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, 2005, pp. 489–498.
[4] M. Mesbahi and G. P. Papavassilopoulos, "On the rank minimization problem over a positive semidefinite linear matrix inequality," *IEEE Trans. Autom. Control*, vol. 42, no. 2, pp. 239–243, Feb. 1997.
[5] L. Eldén, *Matrix Methods in Data Mining and Pattern Recognition*, vol. 4. Philadelphia, PA, USA: SIAM, 2007.
[6] M. Ashraphijuo and X. Wang, "Clustering a union of low-rank subspaces of different dimensions with missing data," *Pattern Recognit. Lett.*, vol. 120, pp. 31–35, 2019.
[7] D. Pimentel-Alarcon and R. Nowak, "The information-theoretic requirements of subspace clustering with missing data," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 802–810.
[8] D. L. Pimentel-Alarcón, N. Boston, and R. D. Nowak, "Deterministic conditions for subspace identifiability from incomplete sampling," in *Proc. IEEE Int. Symp. Inf. Theory*, 2015, pp. 2191–2195.
[9] D. Pimentel-Alarcón, L. Balzano, and R. Nowak, "Necessary and sufficient conditions for sketched subspace clustering," in *Proc. Annual Allerton Conf. Commun., Control, Comput.*, 2016, pp. 1335–1343.
[10] J. D. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proc. Int. Conf. Mach. Learn.*, 2005, pp. 713–719.
[11] K. Azizzadenesheli, A. Lazaric, and A. Anandkumar, "Reinforcement learning of POMDPs using spectral methods," in *Proc. Conf. Learn. Theory*, 2016, pp. 1639–1642.
[12] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.
[13] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053–2080, May 2010.
[14] J. F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.
[15] M. Ashraphijuo, R. Madani, and J. Lavaei, "Characterization of rank-constrained feasibility problems via a finite number of convex programs," in *Proc. IEEE Conf. Decision Control*, 2016, pp. 6544–6550.
[16] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, pp. 1–19, 2011.
[17] R. Tomioka, K. Hayashi, and H. Kashima, "Estimation of low-rank tensors via convex optimization," 2010, arXiv:1010.0789.
[18] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. A. Suykens, "Learning with tensors: A framework based on convex optimization and spectral regularization," *Mach. Learn.*, vol. 94, no. 3, pp. 303–351, 2014.
[19] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proc. ACM Symp. Theory Comput.*, 2013, pp. 665–674.
[20] P. Jain and S. Oh, "Provable tensor factorization with missing data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1431–1439.
[21] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Imag. Sci.*, vol. 1, no. 3, pp. 248–272, 2008.
[22] W. Wang, V. Aggarwal, and S. Aeron, "Tensor completion by alternating minimization under the tensor train (TT) model," Sep. 2016, arXiv:1609.05587.

[23] X.-Y. Liu, S. Aeron, V. Aggarwal, and X. Wang, "Low-tubal-rank tensor completion using alternating minimization," *Proc. SPIE*, vol 9848, 2016, Art. no. 984809.
[24] J. Yang and X. Yuan, "Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization," *Math. Comput.*, vol. 82, no. 281, pp. 301–329, 2013.
[25] P. Pezeshkpour, C. Guestrin, and S. Singh, "Compact factorization of matrices using generalized round-rank," 2018, arXiv:1805.00184.
[26] P. Pezeshkpour, L. Chen, and S. Singh, "Embedding multimodal relational data for knowledge base completion," 2018, arXiv:1809.01341.
[27] D. Pimentel-Alarcón, N. Boston, and R. Nowak, "A characterization of deterministic sampling patterns for low-rank matrix completion," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 623–636, Jun. 2016.
[28] M. Ashraphijuo and X. Wang, "Fundamental conditions for low-CP-rank tensor completion," *J. Mach. Learn. Res.*, vol. 18, no. 63, pp. 1–29, 2017.
[29] M. Ashraphijuo, V. Aggarwal, and X. Wang, "A characterization of sampling patterns for low-Tucker-rank tensor completion problem," in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 531–535.
[30] M. Ashraphijuo, X. Wang, and V. Aggarwal, "A characterization of sampling patterns for low-rank multi-view data completion problem," in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 1147–1151.
[31] M. Ashraphijuo, V. Aggarwal, and X. Wang, "On deterministic sampling patterns for robust low-rank matrix completion," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 343–347, Mar. 2018.
[32] M. Ashraphijuo, X. Wang, and V. Aggarwal, "Rank determination for low-rank data completion," *J. Mach. Learn. Res.*, vol. 18, no. 98, pp. 1–29, 2017.
[33] D. L. Pimentel-Alarcón, "A simpler approach to low-rank tensor canonical polyadic decomposition," in *Proc. Annu. Allerton Conf. Commun., Control, Comput.*, 2016, pp. 474–481.
[34] D. Pimentel-Alarcón, G. Ongie, L. Balzano, R. Willett, and R. Nowak, "Low algebraic dimension matrix completion," in *Proc. Annu. Allerton Conf. Commun., Control, Comput.*, 2017, pp. 790–797.
[35] X. Y. Liu, S. Aeron, V. Aggarwal, X. Wang, and M. Y. Wu, "Adaptive sampling of RF fingerprints for fine-grained indoor localization," *IEEE Trans. Mobile Comput.*, vol. 15, no. 10, pp. 2411–2423, Oct. 2016.
[36] D. Kressner, M. Steinlechner, and B. Vandereycken, "Low-rank tensor completion by Riemannian optimization," *BIT Numer. Math.*, vol. 54, no. 2, pp. 447–468, 2014.
[37] M. Ashraphijuo and X. Wang, "A characterization of sampling patterns for union of low-rank subspaces retrieval problem." in *Proc. Int. Symp. Artif. Intelligence Math.*, 2018, pp. 1–8.
[38] A. Krishnamurthy and A. Singh, "Low-rank matrix and tensor completion via adaptive sampling," in *Proc. Adv. Neural Inf. Process. Syst.s*, 2013, pp. 836–844.
[39] M. Ashraphijuo and X. Wang, "Scaled nuclear norm minimization for low-rank tensor completion," 2017, *arXiv:1707.07976*.
[40] M. Ashraphijuo, X. Wang, and V. Aggarwal, "An approximation of the CP-rank of a partially sampled tensor," in *Proc. Annu. Allerton Conf. Commun., Control, Comput.*, 2017, pp. 604–611.
[41] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, 2010.
[42] G. Shabat, "Matrix completion using nuclear norm, spectral norm or weighted nuclear norm minimization," 2015. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/50056-matrix-completion-using-nuclear-norm-spectral-normor-weighted-nuclear-norm-minimization
[43] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*. Reading, MA, USA: Addison-Wesley, 1973, vol. 28.
[44] A. Deif, "Rigorous perturbation bounds for eigenvalues and eigenvectors of a matrix," *J. Comput. Appl. Math.*, vol. 57, no. 3, pp. 403–412, 1995.

Morteza Ashraphijuo received the B.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, and the M.Sc. degree in industrial engineering & operations research from the University of California, Berkeley, Berkeley, CA, USA, in 2015 and 2016, respectively. He is currently working toward the Ph.D. degree in electrical engineering from Columbia University in the City of New York, New York, NY, USA. His research interests lie in the general areas of machine learning, optimization theory, and statistical learning. He has won multiple awards including a Silver Medal in the International Mathematical Olympiad (IMO) in 2010 and the Katta G. Murty Best Paper Prize from University of California, Berkeley, in 2016.

Xiaodong Wang (S'98–M'98–SM'04–F'08) received the Ph.D degree in electrical engineering from Princeton University, Princeton, NJ, USA. He is currently a Professor of Electrical Engineering with Columbia University in the City of New York, New York, NY, USA. His research interests fall in the general areas of computing, signal processing, and communications, and he has published extensively in these areas. Among his publications is a book entitled *Wireless Communication Systems: Advanced Techniques for Signal Reception* (Prentice-Hall, 2003). His current research interests include wireless communications, statistical signal processing, and genomic signal processing. He is the recipient of the 1999 NSF CAREER Award, the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award, and the 2011 IEEE Communication Society Award for Outstanding Paper on New Communication Topics. He was an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON SIGNAL PROCESSING, and IEEE TRANSACTIONS ON INFORMATION THEORY. He is listed as an ISI Highly-cited Author.



Junwei Zhang received the B.Sc. degree in electrical engineering from Peking University, Beijing, China, in 2018. He is currently pursuing the M.Sc. degree at the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA.