Low-tubal-rank Tensor Completion using Alternating Minimization

Xiao-Yang Liu[1,2], Shuchin Aeron[3], Vaneet Aggarwal[4], and Xiaodong Wang[1]

[1]Department of Electrical Engineering, Columbia University, NY, USA
[2]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[3]Department of Electrical and Computer Engineering, Tufts University, Medford, MA, USA
[4]School of Industrial Engineering, Purdue University, West Lafayette, IN, USA

ABSTRACT

The low-tubal-rank tensors have been recently proposed to model real-world multidimensional data. In this paper, we study the low-tubal-rank tensor completion problem, i.e., to recover a third-order tensor by observing a subset of elements selected uniform at random. We propose a fast iterative algorithm, called *Tubal-Alt-Min*, that is inspired by similar approach for low rank matrix completion. The unknown low-tubal-rank tensor is parameterized as the product of two much smaller tensors with the low-tubal-rank property being automatically incorporated, and Tubal-Alt-Min alternates between estimating those two tensors using tensor least squares minimization. We note that the tensor least squares minimization is different from its counterpart and nontrivial, and this paper gives a routine to carry out this operation. Further, on both synthetic data and real-world video data, evaluation results show that compared with the tensor nuclear norm minimization, the proposed algorithm improves the recovery error by orders of magnitude with smaller running time for higher sampling rates.

Keywords: Data completion, Tensor, Tubal-rank, Alternating minimization, Least squares minimization

1. INTRODUCTION

The big data era calls for efficient algorithms to analyze the enormous amount of data generated by high-resolution sensors, mobile devices, online merchants, and social networks [1]. Such real-world data/signals* are naturally represented as multidimensional arrays [2], namely, vectors, matrices, high-order tensors or tensor networks. Signal recovery from partial measurements [3], exploiting the redundancy property modeled as sparse or low-rank, receives wide attention in various research and engineering communities. We are interested in fast algorithms for the problem of multilinear data completion when the measurement operator being a simple downsampling operation. Exemplar applications include MRI imaging [3], signal processing [2], big data analysis with missing entries [4], data privacy [5], network engineering [6–9], Internet of things [10, 11], machine learning [12], computer vision [13, 14], recommendation systems [15] and system identification [16].

Such diverse applications motivate/justify the developments of compressive sensing (vector case) [3, 17], matrix completion and matrix sensing [18, 19], and higher-order tensor completion [20–22]. Compressive sensing [3, 17] advocated relaxing the original NP-hard problem to its convex surrogate, i.e., replacing the ℓ_0-norm with ℓ_1-norm. Similarly, researchers introduced nuclear norm [23]/tensor-nuclear norm [24] to approximate the combinatorial *rank* function for the low-rank matrix/tensor completion problem†. Those two relaxation approaches achieve optimal results with computational sacrifice, mainly because of the time-consuming SVD (singular value decomposition) or tensor-SVD operations [24, 25].

Alternating minimization based approaches have been recently proposed for matrix completion [26–28]. The unknown low-rank matrix $M \in \mathbb{R}^{m\times n}$ is factorized into two much smaller matrices X and Y of size $m \times r$ and $n \times r$, respectively, i.e., $M = XY^\dagger$, and rank $r \ll \min(m, n)$, thus requiring much less computation to optimize. The low-tubal-rank tensor is recently proposed for modeling multilinear real-world data, such as WiFi

*In the following, we use the words "signal" and "data" interchangeably.
†A vector is a first-order tensor while a matrix is a second-order tensor.

fingerprints [6], network data [29], images [30], videos [25], and seismic data [31]. In this paper, we consider alternating based approaches for tensor completion with low-tubal-rank. The main challenge is that the algebraic structure for low-tubal-rank tensor is different and the alternating minimization based steps of matrix completion do not extend in a straightforward fashion.

In this paper, we propose a fast alternating minimization algorithm, *Tubal-Alt-Min*, for the low-tubal-rank tensor completion problem. A key novelty is solving a least square minimization for tensors, which can be of independent interest. Although non-convex, the proposed alternating minimization based approach can be much faster than its convex counterparts. Further, we evaluate the proposed algorithm on both synthetic data and real-world video data. The performance in terms of recovery error, convergence rates, and running times are compared with the convex relaxation based tensor nuclear norm minimization (TNN-ADMM) [24, 25]. Tubal-Alt-Min improves the recovery error significantly and converges in a much lower number of iterations thus achieving significantly higher convergence rates and lower run times.

2. NOTATIONS AND PRELIMINARIES

We describe the notations and models, most of them were introduced in [21, 22, 32, 33] while some of them are first introduced as specified. Throughout the paper, we will focus on real values third-order tensors denoted by uppercase calligraphic letter, $\mathcal{X} \in \mathbb{R}^{m\times n\times k}$. Lowercase boldface letters $\mathbf{x}$, $\mathbf{y} \in \mathbb{R}^n$ denote vectors, and uppercase letters X, $Y \in \mathbb{R}^{m\times n}$ denote matrices.

Let $X^\dagger$ denote the transpose of matrix. We use i, j, ℓ to index the first, second and third dimension of a tensor, and s, t for temporary indexing. $[n]$ denotes the set $\{1, 2, ..., n\}$. Usually, $i \in [m]$, $j \in [n]$, $\ell \in [k]$ unless otherwise specified.

The ℓ_2-norm of a vector is defined as $||\mathbf{a}||_2 = \sqrt{\sum_{i=1} \mathbf{a}_i^2}$, while the Frobenius norm of a matrix X is $||X||_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n X_{ij}^2}$ and a tensor is $||\mathcal{T}||_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n \sum_{\ell=1}^k \mathcal{T}_{ij\ell}^2}$.

For tensor $\mathcal{T} \in \mathbb{R}^{m\times n\times k}$, the (i, j, ℓ)-th entry is $\mathcal{T}(i, j, \ell)$, or concisely represented as $\mathcal{T}_{ij\ell}$.

Tubes, fibers and slices of a tensor: A *tube* (also called a fiber) is a 1-D section defined by fixing all indices but one, while a *slice* is a 2-D section defined by fixing all but two indices. We use $\mathcal{T}(:, j, \ell)$, $\mathcal{T}(i, :, \ell)$, $\mathcal{T}(i, j, :)$ to denote the (i, j)-th mode-1, mode-2, mode-3 tubes, which are vectors, and $\mathcal{T}(:, :, \ell)$, $\mathcal{T}(:, j, :)$, $\mathcal{T}(i, :, :)$ to denote the frontal, lateral, horizontal slices, which are matrices. For simplicity, sometimes we denote $\mathcal{T}^{(i)} = \mathcal{T}(:, :, i)$.

Tensor transpose: Tensor transpose, $\mathcal{T}^\dagger$ is obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through k, i.e., for $2 \le \ell \le k$, $\mathcal{T}^\dagger(:, :, \ell) = \mathcal{T}^{\dagger(\ell)} = (\mathcal{T}(:, :, k + 2 - \ell))^\dagger$ (the transpose of matrix $\mathcal{T}(:, :, k + 2 - \ell)$). For reasons to become clear soon, we define a tensor $\widetilde{\mathcal{T}}$, which is obtained by taking the Fourier transform along the third dimension of $\mathcal{T}$, i.e., $\widetilde{\mathcal{T}}(i, j, :) = \text{fft}(\mathcal{T}(i, j, :))$. In MATLAB notation, $\widetilde{\mathcal{T}} = \text{fft}(\mathcal{T}, [\,], 3)$, and one can also compute $\mathcal{T}$ from $\widetilde{\mathcal{T}}$ via $\mathcal{T} = \text{ifft}(\widetilde{\mathcal{T}}, [\,], 3)$.

We now define the linear algebraic model for tensor decomposition that is used in this paper. For two tubes/vectors of the same size, i.e., $\mathbf{a}$, $\mathbf{b} \in \mathbb{R}^k$, let $\mathbf{a} * \mathbf{b}$ denote the *circular convolution* between these two vectors, which preserves the size. Using this one can define multiplication between two tensors, that is referred to as the t-product, as stated below.

t-product. The tensor-product $\mathcal{C} = \mathcal{A} * \mathcal{B}$ of $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times k}$ and $\mathcal{B} \in \mathbb{R}^{n_2\times n_3\times k}$ is a tensor of size $n_1 \times n_3 \times k$, $\mathcal{C}(i, j, :) = \sum_{s=1}^{n_2} \mathcal{A}(i, s, :) * \mathcal{B}(s, j, :)$, for $i \in [n_1]$ and $j \in [n_3]$.

We now note the following definitions under this construction.

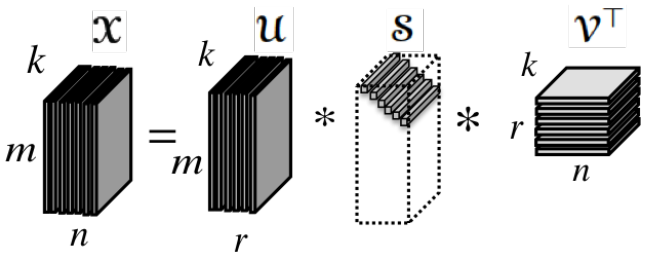

Figure 1. t-SVD of an $m \times n \times k$ tensor of tubal rank r.

Identity tensor. The identity tensor $\mathcal{I} \in \mathbb{R}^{n\times n\times k}$ is a tensor whose first frontal slice $\mathcal{I}(:,:,1)$ is the $n \times n$ identity matrix and all other frontal slices are zero.

Orthogonal tensor. A tensor $\mathcal{Q} \in \mathbb{R}^{n\times n\times k}$ is orthogonal if it satisfies $\mathcal{Q}^\dagger * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^\dagger = \mathcal{I}$.

Inverse. The inverse of a tensor $\mathcal{T} \in \mathbb{R}^{n\times n\times k}$ is written as $\mathcal{T}^{-1} \in \mathbb{R}^{n\times n\times k}$ and satisfies $\mathcal{T}^{-1} * \mathcal{T} = \mathcal{T} * \mathcal{T}^{-1} = \mathcal{I}$.

Block diagonal form of third-order tensor. Let $\overline{\mathcal{A}}$ denote the block-diagonal matrix of the tensor $\mathcal{A}$ in the Fourier domain, i.e.,

$$\overline{\mathcal{A}} \doteq \text{blkdiag}(\hat{\mathcal{A}}) \doteq \begin{bmatrix} \hat{\mathcal{A}}(:,:,1) & & & \\ & \hat{\mathcal{A}}(:,:,2) & & \\ & & \ddots & \\ & & & \hat{\mathcal{A}}(:,:,k) \end{bmatrix} \in \mathbb{C}^{mk\times nk}. \tag{1}$$

It is easy to verify that the block diagonal matrix of $\mathcal{A}^\dagger$ is equal to the transpose of the block diagonal matrix of $\mathcal{A}$: $\overline{\mathcal{A}^\dagger} = \overline{\mathcal{A}}^\dagger$.

f-diagonal tensor. A tensor is called f-diagonal if each frontal slice of the tensor is a diagonal matrix, i.e., $\Theta(i,j,\ell) = 0$ for $i \neq j, \forall \ell$.

Using all these definitions and constructs, in [21] the following singular value decomposition, referred to as t-SVD was obtained for dimensionality reduction of third order data.

t-SVD: For $\mathcal{T} \in \mathbb{R}^{m\times n\times k}$, the t-SVD of $\mathcal{T}$ is given by $\mathcal{T} = \mathcal{U} * \Theta * \mathcal{V}^\dagger$, where $\mathcal{U}$ and $\mathcal{V}$ are orthogonal tensors of sizes $m \times m \times k$ and $n \times n \times k$, respectively. Θ is a f-diagonal tensor of size $m \times n \times k$ and the tubes are called the eigentubes of $\mathcal{T}$. This is depicted in Figure 1. An algorithm for computing the t-SVD is outlines as Algorithm 1.

Algorithm 1 t-SVD

Input: $\mathcal{X} \in \mathbb{C}^{n_1\times n_2\times n_3}$
$\widetilde{\mathcal{X}} \leftarrow \texttt{fft}(\mathcal{X}, [\,], 3)$; Take DFT along 3-rd direction
for $i = 1$ to n_3 **do**
  $[\hat{U}, \hat{S}, \hat{V}] = \texttt{SVD}(\widetilde{\mathcal{X}}(:,:,i))$
  $\widetilde{\mathcal{U}}(:,:,i) = \hat{U}; \widetilde{\mathcal{S}}^{(i)} = \hat{S}; \widetilde{\mathcal{V}}^{(i)} = \hat{V}$;
end for
$\mathcal{U} \leftarrow \texttt{ifft}(\widetilde{\mathcal{U}}, [\,], 3)$; $\mathcal{S} \leftarrow \texttt{ifft}(\widetilde{\mathcal{S}}, [\,], 3)$; $\mathcal{V} \leftarrow \texttt{ifft}(\widetilde{\mathcal{V}}, [\,], 3)$;

Tensor tubal-rank. The tensor tubal-rank of a third-order tensor $\mathcal{T}$ is the number of non-zero tubes of Θ in the t-SVD, denoted as r.

Suppose $\mathcal{T}$ has tubal-rank r, then the reduced t-SVD of $\mathcal{T}$ is given by $\mathcal{T} = \mathcal{U} * \Theta * \mathcal{V}^\dagger$, where $\mathcal{U} \in \mathbb{R}^{m\times r\times k}$ and $\mathcal{V} \in \mathbb{R}^{n\times r\times k}$ satisfying $\mathcal{U}^\dagger * \mathcal{U} = \mathcal{I}$, $\mathcal{V}^\dagger * \mathcal{V} = \mathcal{I}$, and Θ is a f-diagonal tensor of size $r \times r \times k$. This reduced version of t-SVD will be used throughout the paper unless otherwise noted.

Best rank-r approximation. Let the t-SVD of $\mathcal{T} \in \mathbb{R}^{m\times n\times k}$ be $\mathcal{T} = \mathcal{U} * \Theta * \mathcal{V}^\dagger$ and positive integer r, define $\mathcal{T}_r = \sum_{s=1}^{r} \mathcal{U}(:, s, :) * \Theta(s, s, :) * \mathcal{V}^\dagger(:, s, :)$, then

$$\mathcal{T}_r = \arg \min_{\overline{\mathcal{T}}\in\mathbb{T}} ||\mathcal{T} - \overline{\mathcal{T}}||_F,$$

where $\mathbb{T} = \{\mathcal{X} * \mathcal{Y}^\dagger | \mathcal{X} \in \mathbb{R}^{m\times r\times k}, \mathcal{Y} \in \mathbb{R}^{n\times r\times k}\}$.

Remark: In the proposed framework, the main advantage, in terms of modeling multidimensional data comes from the *linear algebraic* nature of the construction where the tensors can be seen as linear operators on other tensors. This is in contrast to the classical multilinear algebraic approach [2, 20] where one considers the tensor as a multilinear operator with different notions of rank.

3. PROBLEM STATEMENT AND PROPOSED ALGORITHM

3.1 Problem Statement

In this paper we consider the problem of completing a 3-D tensor under the assumption that the 3-D tensor has low tubal rank, which is defined in the previous section. Specifically, assume that data tensor $\mathcal{T} \in \mathbb{R}^{m\times n\times k}$ has tubal-rank $r \ll \min(m, n)$. By observing a set $\Omega \subset [m] \times [n] \times [k]$ of $\mathcal{T}$'s elements, our aim is to recover $\mathcal{T}$. That is, knowing the elements $\mathcal{T}_{ij\ell}$ for $(i, j, \ell) \in \Omega$, we want to estimate the elements outside of Ω as accurately as possible.

Let $\mathcal{P}^\Omega(\cdot)$ denote the projection of a tensor onto the observed set Ω, i.e.,

$$\mathcal{P}^\Omega(\mathcal{T})_{ij\ell} = \begin{cases} \mathcal{T}_{ij\ell}, & \text{if } (i, j, \ell) \in \Omega, \\ 0, & \text{otherwise,} \end{cases}$$

where the (i, j, ℓ)-th entry of $\mathcal{P}^\Omega(\mathcal{X})$ equals to $\mathcal{X}_{ij\ell}$ if $(i, j, \ell) \in \Omega$ and zero otherwise. Since $\mathcal{T}$ is known to be a low-tubal-rank tensor and the estimated $\hat{\mathcal{T}}$ should be in consistent with $\mathcal{T}$ on the set Ω, the low-tubal-rank tensor completion problem is formulated as the following optimization problem:

$$\begin{aligned} \hat{\mathcal{T}} = \arg \min_{\mathcal{X}\in\mathbb{R}^{m\times n\times k}} & \ \text{rank}(\mathcal{X}) \\ s.t. \ & \ \mathcal{P}^\Omega(\mathcal{X}) = \mathcal{P}^\Omega(\mathcal{T}) \end{aligned} \tag{2}$$

where $\mathcal{X} \in \mathbb{R}^{m\times n\times k}$ is the decision variable, and the function rank$(\cdot)$ refers to the tensor tubal-rank. Problem (2) is known to be NP-hard in general and existing works seek to relax the rank function to its convex surrogate, namely, the tensor-nuclear norm [24, 25]. In [24], it was shown that given a sufficient number of observations ($|\Omega| = \mathcal{O}(\max\{m, n\}kr \log mn)$), it was proved that tensor-nuclear norm minimization results in exact recovery under random sampling if the tensors satisfy certain tensor incoherency conditions.

However, the computational cost of the algorithm in [24] is relatively high due to two key factors: 1) Each iteration requires computation of SVD for large block diagonal matrices, and 2) The iterations are jointly done in both the time domain and frequency domain, thus involving frequent and large number of Fourier and Inverse Fourier transforms. Therefore, in the following section, we will introduce an alternating minimization algorithm, inspired by the alternating minimization approach's empirical and theoretical successes in low-rank matrix completion [26–28].

Algorithm 2 Tubal Alternating Minimization: AM($\mathcal{P}^{\Omega}(\mathcal{T}), \Omega, L, r, \epsilon, \mu$)

Input: Observation set $\Omega \in [n] \times [m] \times [k]$ and the corresponding elements $\mathcal{P}^{\Omega}(\mathcal{T})$, number of iterations L, target tubal-rank r.

$\mathcal{X}_0 \leftarrow$ Initialize($\mathcal{P}^{\Omega}(\mathcal{T}), \Omega, r$),

For $\ell = 1$ to L

$\mathcal{Y}_\ell \leftarrow \text{LS}(\mathcal{P}^{\Omega}(\mathcal{T}), \Omega, \mathcal{X}_{\ell-1}, r)$,

$\mathcal{X}_\ell \leftarrow \text{LS}(\mathcal{P}^{\Omega}(\mathcal{T}), \Omega, \mathcal{Y}_\ell, r)$,

Output: Pair of tensors $(\mathcal{X}_L, \mathcal{Y}_L)$.

3.2 The Alternating Minimization Algorithm for low tubal rank tensor completion

The main idea of the algorithm is very similar to that of the alternating minimization for matrix completion [26] and tries to iteratively estimate two low tubal rank factors $\mathcal{X} \in \mathbb{R}^{m \times r \times k}$ and $\mathcal{Y} \in \mathbb{R}^{r \times n \times k}$, each of tubal rank r, such that

$$\mathcal{T} = \mathcal{X} * \mathcal{Y}.$$

The key ingredient resulting in good performance is a clever initialization procedure that makes sure that one of the initial factors is close (in spectral norm) to the original tensor. The algorithm is outlined in Algorithm 2 and consists of the following steps.

- Initialization: Alg. 3 describes the procedure for finding a good starting point that can be shown to have a bounded distance from the optimal result. Similar to the alternating minimization for matrix completion, our initialization procedure computes the top-r singular singular slices of $\mathcal{P}^{\Omega}(\mathcal{T})$.
- Alternating Least-Squares (LS) update: Alg. 4 describes the least-squares update whose implementation details are subsequently presented in section 3.2.1.

Algorithm 3 Initialization Algorithm: Initialize($\mathcal{P}^{\Omega}, \Omega, r$)

Input: Observation set $\Omega \in \mathbb{R}^{[m] \times [n] \times [k]}$ and elements $\mathcal{P}^{\Omega}(\mathcal{T})$, target dimension r.

Compute the first r eigenslices $\mathcal{W} \in \mathbb{R}^{m \times r \times k}$ of $\mathcal{P}^{\Omega}(\mathcal{T})$ using the t-SVD computed using Algorithm 1 .

Output: Orthonormal tensor $\mathcal{X} \in \mathbb{R}^{m \times r \times k}$.

Algorithm 4 Least Squares Update: LS($\mathcal{P}^{\Omega}(\mathcal{T}), \Omega, \mathcal{X}, r$)

Input: target dimension r, observation set $\Omega \in [m] \times [n] \times [k]$ and elements $\mathcal{P}^{\Omega}(\mathcal{T})$, tensor $\mathcal{X} \in \mathbb{R}^{[m] \times [r] \times [k]}$

$\mathcal{Y} = \arg\min_{\mathcal{Y} \in \mathbb{R}^{[r] \times [n] \times [k]}} ||\mathcal{P}^{\Omega}(\mathcal{T} - \mathcal{X} * \mathcal{Y})||_F^2$

Output: tensor $\mathcal{Y}$.

3.2.1 Implementation of Least Squares Minimization

For simplicity, denote $\mathcal{T}^{\Omega} = \mathcal{P}^{\Omega}(\mathcal{T})$ and let $\mathcal{P}^{\Omega}$ be the sampling tensor with ones at places where the tensor is sampled and zero otherwise. Then we have $\mathcal{T}^{\Omega} = \mathcal{P}^{\Omega} \odot \mathcal{T}$ where $\odot$ is the element-wise multiplication of same size arrays. For the (i, j)-th tube, $\mathcal{T}_{ij} = \mathcal{P}^{\Omega}_{ij} \odot \mathcal{T}_{ij}$. According to the *Convolution Theorem*, we can transform the least squares minimization in Alg. 4 to the following frequency domain version:

$$\widetilde{\mathcal{Y}} = \underset{\widetilde{\mathcal{Y}} \in \mathbb{R}^{r \times n \times k}}{\arg\min} \; ||\widetilde{\mathcal{T}}_{\Omega} - \widetilde{\mathcal{P}}_{\Omega} \cdot \otimes (\widetilde{\mathcal{X}} \; \cdot \S \; \widetilde{\mathcal{Y}})||_F^2, \tag{3}$$

where $\cdot\otimes$ denotes tube-wise circular convolution, and $\cdot\S$ denotes front-slice-wise matrix multiplication. The operator $\cdot\S$ is introduced to replace the block diagonal matrix in [24, 25] since we need to preserve the three-way data array structure. In the following, we will show how to compute (3) by providing methods to transform (3) into n separate standard least squares minimization problem. In other words, each lateral slice of $\widetilde{\mathcal{Y}}$ is a standard

least squares minimization problem and will be estimated separately. Note that the above problem (3) can be split into n separate subproblems:

$$\widetilde{\mathcal{Y}}(:,j,:) = \underset{\widetilde{\mathcal{Y}}(:,j,:)\in\mathbb{R}^{r\times 1\times k}}{\arg\min} ||\widetilde{\mathcal{T}}_\Omega(:,j,:) - \widetilde{\mathcal{P}}_\Omega(:,j,:)\cdot\otimes(\widetilde{\mathcal{X}} \ \cdot\S\ \widetilde{\mathcal{Y}}(:,j,:))||_F^2, \tag{4}$$

where each subproblem corresponds to estimating a lateral slice $\widetilde{\mathcal{Y}}(:,j,:),\ j\in[n]$. Similarly, we can estimate $\widetilde{\mathcal{X}}$ in the following way:

$$\widetilde{\mathcal{X}} = \underset{\widetilde{\mathcal{X}}\in\mathbb{R}^{n\times r\times k}}{\arg\min} ||\widetilde{\mathcal{T}}_\Omega^T - \widetilde{\mathcal{P}}_\Omega^T\cdot\otimes(\widetilde{\mathcal{Y}}^T \ \cdot\S\ \widetilde{\mathcal{X}}^T)||_F^2, \tag{5}$$

where $\mathcal{X}^T$ denotes the tube-wise transpose (the transpose of a matrix of vectors). To solve this one performs the following steps.

1. A lateral slice, $\widetilde{\mathcal{T}_\Omega}(:,j,:)$ of size $m\times 1\times k$, is squeezed into a vector b of size $mk\times 1$ in the following way:

$$b = [\text{squeeze}(\widetilde{\mathcal{T}_\Omega}(1,j,:)); \text{squeeze}(\widetilde{\mathcal{T}_\Omega}(2,j,:)); ...; \text{squeeze}(\widetilde{\mathcal{T}_\Omega}(m,j,:))], \tag{6}$$

   where $\text{squeeze}(\widetilde{\mathcal{T}_\Omega}(i,j,:))$ squeezes the i-th tube of the j-th lateral slice of $\widetilde{\mathcal{T}_\Omega}$ into vector of size $k\times 1$. Similarly $\widetilde{\mathcal{Y}_\Omega}(:,j,:)$ is transformed into a vector $\mathbf{x}$ of size $rk\times 1$:

$$\mathbf{x} = [\text{squeeze}(\widetilde{\mathcal{Y}_\Omega}(1,j,:)); \text{squeeze}(\widetilde{\mathcal{Y}_\Omega}(2,j,:)); ...; \text{squeeze}(\widetilde{\mathcal{Y}_\Omega}(m,j,:))]; . \tag{7}$$

2. $\widetilde{\mathcal{X}}$ is transformed into a block diagonal matrix A_1 of size $mk\times rk$ like so,

$$A_1 = \begin{bmatrix} \widetilde{\mathcal{X}}(:,:,1) & & \\ & \widetilde{\mathcal{X}}(:,:,2) & \\ & & \widetilde{\mathcal{X}}(:,:,k) \end{bmatrix}. \tag{8}$$

3. The j-th lateral slice $\widetilde{\mathcal{P}}_\Omega(:,j,:)$ is transformed into a tensor A_2 of size $k\times k\times m$ first, and then into a matrix A_3 of size $mk\times mk$

$$A_2(:,:,\ell) = \mathbf{circ}(\widetilde{\mathcal{P}}_\Omega(\ell,j,:)),\ \ell\in[m], \tag{9}$$

   where $\mathbf{circ}(\mathbf{v})$ is a square circulant matrix formed from a vector $\mathbf{v}$, where the i-th column is a circularly shifted version of vector $\mathbf{v}$, whose elements are circularly shifted down by amount i.

$$A_3 = \begin{bmatrix} \text{diag}(A_2(1,1,:)) & \text{diag}(A_2(1,2,:)) & \dots & \text{diag}(A_2(1,k,:)) \\ \text{diag}(A_2(2,1,:)) & \text{diag}(A_2(2,2,:)) & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \text{diag}(A_2(k,1,:)) & \dots & \dots & \text{diag}(A_2(k,k,:)) \end{bmatrix}, \tag{10}$$

   where the operator $\text{diag}(\cdot)$ transform a tube into a diagonal matrix by putting the elements in the diagonal.

Therefore, the estimation of the j-th lateral slice is transformed into the following standard least squares minimization problem:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\mathbb{R}^{rk\times 1}}{\arg\min} ||b - A_3A_1\mathbf{x}||_F^2. \tag{11}$$

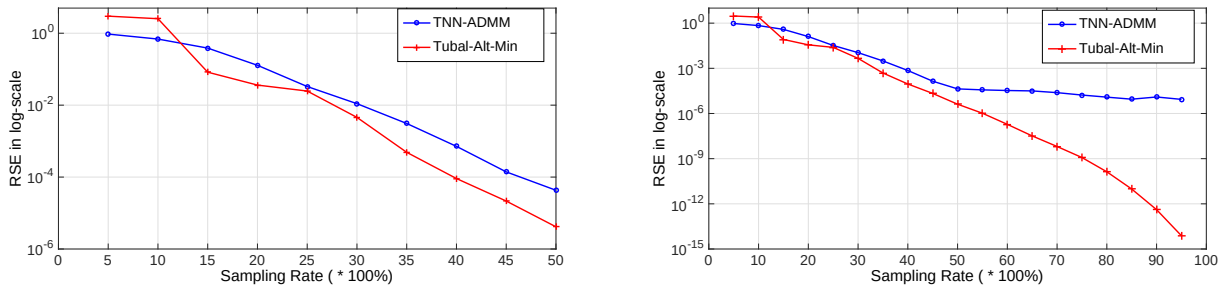

Figure 2. Recovery error RSE in log scale for different sampling rates.

4. EVALUATION

4.1 Experiment Setup

All evaluations are run in Matlab on a server with Linux operating system. The parameters of the server is: Intel® Xeon® Processor E5-2650 v3, 2.3GHz clock speed, 2 CPUs with 10 physical cores each, virtually maximum 40 threads, 25 MB cache, and 64 GB memory. We test our alternating minimization algorithm (Tubal-Alt-Min) on both synthetic and real dataset. The synthetic data, generated according to low-tubal-rank tensor model, serves as well-controlled inputs for understanding Tubal-Alt-Min's performance. On the other hand, the real data tests the applicability of our low-tubal-rank tensor model, as compared with other tensor models.

For synthetic data, we compare our Tubal-Alt-Min algorithm with the tensor-nuclear norm minimization method based on Alternating Direction Method of Multipliers (TNN-ADMM) for low-tubal-rank tensors proposed in [25] to compare our non-convex approach with the convex relaxation based approach. We complete a third-order tensor of size $n_1 \times n_2 \times k$ with tubal-rank r from $|\Omega|$ observed elements. Three metrics are adopted for comparison - the recovery error, the running time, and the convergence speed.

- For recovery error, we adopt the metric relative square error, defined as RSE= $||\hat{\mathcal{T}} - \mathcal{T}||_F / ||\mathcal{T}||_F$.
- For the running time, varying the tensor size and fixing other parameters, we measure cpu time in seconds.
- For the convergence speed, we measure decreasing rate of the RSE across iterations by linearly fitting the measured RSEs. These depict the convergence rates of the two iterative algorithms.

For real dataset, we choose a basketball video of size $144 \times 256 \times 40$ (source: YouTube, as used in [25]), with a non-stationary panning camera moving from left to right horizontally following the running players.

4.2 Synthetic Data

To demonstrate improved recovery error, we use a low-tubal-rank tensor of size $200 \times 200 \times 50$ and tubal-rank 10 which is generated by a tensor product of two Guassian random tensors of sizes $200 \times 10 \times 50$ and $10 \times 200 \times 50$. We set the maximum iteration number to be 10 for Tubal-Alt-Min, and 500 for TNN-ADMM, respectively. Varying the sampling rate as $5\%,\ 10\%, ..., 95\%$ by uniform randomly select entries, we test each sampling rate 5 times and then compute the average recovery error results.

Fig. 2 depicts the recovery error performance (RSE in log scale) of Tubal-Alt-Min and TNN-ADMM for varying sampling rates. For a clear comparison, we draw two plots for sampling rate $\leq 50\%$ and sampling rates $5\% \sim 95\%$, respectively. Tubal-Alt-Min achieves lower orders of error magnitude for sampling rates higher than 50%, while the RSE does not decrease much for TNN-ADMM. For very low sampling rates ($5\% \sim 10\%$), Tubal-Alt-Min does not perform as good as TNN-ADMM, but the performance of Tubal-Alt-Min is much better at sampling rates of 15% or higher.

We next fix the sampling rate to be 50%, and see the convergence rates of the two algorithms in Fig. 3. We only use 10 iterations for Tubal-Alt-Min, and TNN-ADMM terminates at the 134-th iteration based on the

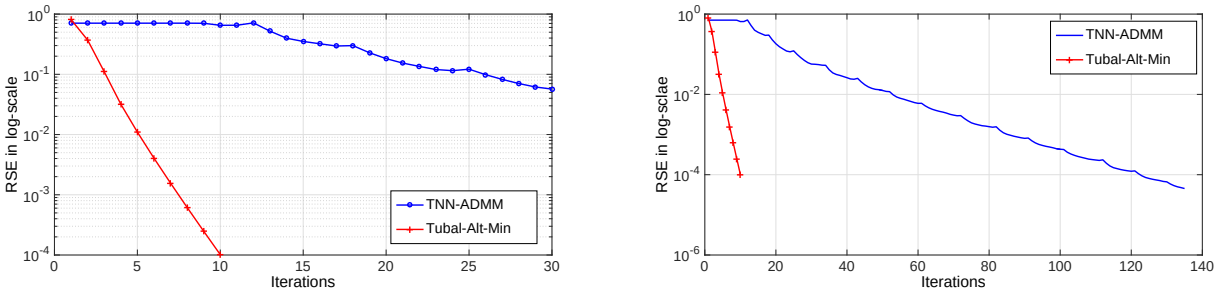

Figure 3. Convergence speed for the two algorithms on synthetic data.

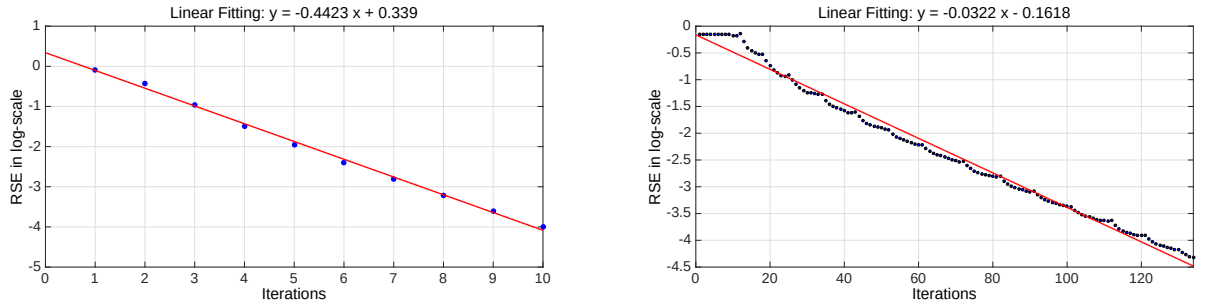

Figure 4. Using linear fitting to estimate the convergence rate for the two algorithms, Left - Tubal-Alt-Min, Right - TNN-ADMM.

preset threshold. For a clear comparison, we draw two plots at iteration number 30 and 140 for TNN-ADMM, respectively. Clearly, recovery error for Tubal-Alt-Min decreases much faster than TNN-ADMM depicting much faster convergence rates for the proposed Tubal-Alt-min approach. The convergence rates are found by a linear fitting of the observed RSE (in log form) as a function of iterations in Fig. 4. For Tubal-Alt-Min, the fitted function is $y = -0.4423x + 0.339$. For TNN-ADMM, the fitted function is $y = -0.0322x - 0.1618$. Thus, the slope of Tubal-Alt-Min is a multiple of 14 more than TNN-ADMM demonstrating significantly better convergence rate.

Fig. 5 shows the running time comparison, measured as CPU time in seconds, for Tubal-Alt-Min and TNN-ADMM for a tensor of size $n \times n \times 20$ of tubal-rank 5, where n is varied from 25 to 200. The sampling rate is set to 50%, while the target RSE is less than 10^{-5}. For $n > 75$, the proposed Tubal-Alt-Min algorithm beats TNN-ADMM. The speed-up is 2 times for $n = 125$ and 5 times for $n = 200$.

4.3 Real Data

We compare Tubal-Alt-Min and TNN-ADMM in terms of the recovery error for a real video data set in Fig. 6. For a clear comparison, we draw two plots for sampling rate $\leq 50\%$ and sampling rates $5\% \sim 95\%$, respectively. Tubal-Alt-Min achieves much lower orders of error magnitude for sampling rates higher than 35%. This demonstrates the performance of the algorithms when the data is not exactly low-tubal-rank thus showing that Tubal-Alt-Min is robust with respect to noise.

5. CONCLUSION AND FUTURE WORK

This paper proposes a fast iterative algorithm, called *Tubal-Alt-Min*, that is based on the alternating minimization approach for low-tubal rank tensor completion. The unknown low-tubal-rank tensor is parameterized as the product of two much smaller tensors with the low-tubal-rank property being automatically incorporated, and Tubal-Alt-Min alternates between estimating those two tensors using tensor least squares minimization. The performance of the proposed algorithm is demonstrated for both synthetic and real-world data. As part of future work, we will derive theoretical performance guarantees on the success of the proposed algorithm under the tensor

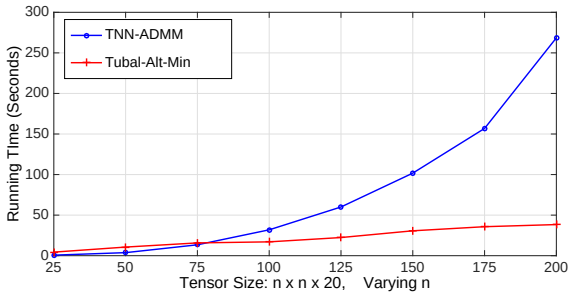

Figure 5. Comparison of running time of the two algorithms for varying tensor size.

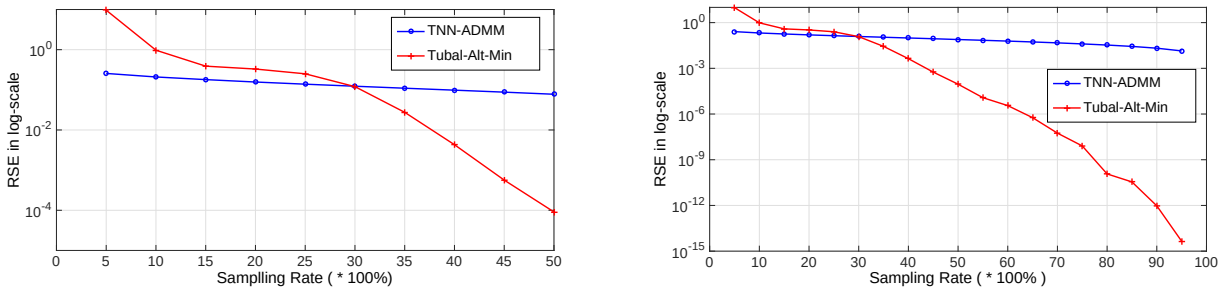

Figure 6. Recovery error RSE in log scale for varying sampling rates.

incoherency conditions as derived in [24]. We will also compare the performance of the proposed algorithm with other tensor factorization approaches.

6. ACKNOWLEDGEMENTS

V. Aggarwal gratefully acknowledges financial support from the Air Force Research Laboratory Faculty Research Extension Grant Award. S. Aeron gratefully acknowledges the support by NSF:CCF:1319653.

References

[1] R. Baraniuk, "More Is less: Signal processing and the data deluge," *Science*, 331(6018), pp. 717-719, 2011.

[2] A. Cichocki, C. Mandic, AH. Phan, C. Caiafa, G. Zhou, Q. Zhao, and L. Lathauwer, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, 32(2), pp. 145-163, 2015.

[3] E.J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, 52(2): 489-509, 2006.

[4] R. Little and D.B. Rubin, "Statistical analysis with missing data," *John Wiley & Sons*, 2014.

[5] L. Kong, L. He, X.-Y. Liu, Y. Gu, M.-Y. Wu, and X. Liu, "Privacy-preserving compressive sensing for crowdsensing based trajectory recovery," *IEEE ICDCS*, 2015.

[6] X.-Y. Liu, S. Aeron, V. Aggarwal, X. Wang, and M. Wu, "Adaptive sampling of RF fingerprints for fine-grained indoor localization," *IEEE Transactions on Mobile Computing*, 2015

[7] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," *ACM SIGCOMM*, 39(4): 267-278, 2009.

[8] S. Rallapalli, L. Qiu, Y. Zhang, Y.-C. Chen, "Exploiting temporal stability and low-rank structure for localization in mobile networks," *ACM MobiCom*, 2010.

[9] Y.-C. Chen, L. Qiu, Y. Zhang, Z. Hu, and G. Xue, "Robust network compressive sensing," *ACM MobiCom*, 2014.

[10] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, "Data loss and reconstruction in sensor networks," *IEEE INFOCOM*, pp. 1654-1662, 2013.

[11] X.-Y. Liu, Y. Zhu, L. Kong, C. Liu, Y. Gu, and M.-Y. Wu, "CDC: Compressive data collection for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, pp. 2188-2197, 2014.

[12] M. Signoretto, Q.T. Dinh, L.D. Lathauwer, and J. Suykens, "Learning with tensors: a framework based on convex optimization and spectral regularization," *Springer Machine Learning*, 94(3): 303-351, 2014.

[13] P. Chen and D. Suter, "Recovering the missing components in a large noisy low-rank matrix: Application to SFM," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(8): 1051-1063, 2004.

[14] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1): 208-220, 2013.

[15] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, (8): 30-37, 2009.

[16] Z. Liu and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification," *SIAM Journal on Matrix Analysis and Applications*, 31(3): 1235-1256, 2009.

[17] E. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. on Information Theory*, 52(12), pp. 5406-5425, 2006.

[18] E. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, 9(6), pp. 717-772, 2009.

[19] E. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Transactions on Information Theory*, 56(5), pp. 2053-2080, 2010.

[20] T.G. Kolda and B.W. Bader, "Tensor decompositions and applications," *SIAM review*, 51(3), pp. 455-500, 2009.

[21] M.E. Kilmer, C.D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and its Applications*, 435(3), pp. 641-658, 2011.

[22] M.E. Kilmer, K. Braman, N. Hao, and R.C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM Journal on Matrix Analysis and Applications*, 34(1), pp. 148-172, 2013.

[23] M Fazel, "Matrix rank minimization with applications," *Diss. PhD thesis, Stanford University*, 2002.

[24] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *arXiv:1502.04689v2*, Feb., 2015.

[25] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-SVD," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3842-3849, 2014.

[26] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," *ACM symposium on Theory of computing*, pp. 665-674, 2013.

[27] M. Hardt and Mary Wootters, "Fast matrix completion without the condition number," *Springer, COLT*, 2014.

[28] M. Hardt. "Understanding alternating minimization for matrix completion," IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS), pp. 651-660, 2014.

[29] V. Aggarwal, H. Ma, A. Mahimkar, W. Willinger, Z. Zhang, and S. Aeron, "FlashFill: inferring fine-grained service quality in cellular networks," submitted.

[30] E. Kernfeld, S. Aeron, and M. Kilmer, "Clustering multi-way data: A novel algebraic approach, 2014.

[31] G. Ely, S. Aeron, N. Hao, and M.E. Kilmer, "5D seismic data completion and denoising using a novel class of tensor decompositions, *Geophysics*, 2015.

[32] D.F. Gleich, C. Greif, and J.M. Varah, "The power and Arnoldi methods in an algebra of circulants," *Numerical Linear Algebra with Applications*, 20(5): 809-831, 2013.

[33] K. Braman, "Third-order tensors as linear operators on a space of matrices," *Linear Algebra and its Applications*, 433(7): 1241-1253, 2010.