Tensor Completion through Total Variation with Initialization from Weighted HOSVD

Zehan Chao, Longxiu Huang and Deanna Needell

Abstract—In our paper, we have studied the tensor completion problem when the sampling pattern is deterministic. We first propose a simple but efficient weighted HOSVD algorithm for recovery from noisy observations. Then we use the weighted HOSVD result as an initialization for the total variation. We have proved the accuracy of the weighted HOSVD algorithm from theoretical and numerical perspectives. In the numerical simulation parts, we also showed that by using the proposed initialization, the total variation algorithm can efficiently fill the missing data for images and videos.

I. INTRODUCTION

Tensor, a high-dimensional array which is an extension of matrix, plays an important role in a wide range of real-world applications [3], [12]. Due to the high-dimensional structure, tensor could preserve more information compared to the unfolded matrix. For instance, a k frame, m × n video stored as an m × n × k tensor will keep the connection between each frames, splitting the frames or unfolding this tensor may lose some conjunctural information. A brain MRI would also benefit from the 3D structure if stored as tensor instead of randomly arranging several snapshots as matrices. On the other hand, most of the real-world datasets are partially missing and incomplete data which can lead to extremely low performance of downstream applications. The linear dependency and redundancy between missing and existing data can be leveraged to recover unavailable data and improve the quality and scale of the incomplete dataset. The task of recovering missing elements from partially observed tensor is called tensor completion and has attracted widespread attention in many applications. e.g., image/video inpainting [23], [25], recommendation systems [26]. Matrix completion problem, [1], [2], [4], [5], [6], [7], [10], [15], [17], [18] as a special case of tensor completion problem has been well-studied in recent decades, which enlightened researchers on the tensor completion problem. First of all, we will introduce notations, basic operations and definitions for tensor.

A. Preliminaries and Notations

Tensors, matrices, vectors and scalars are denoted in different typeface for clarity below. Throughout this paper, calligraphic boldface capital letters are used for tensors, capital letters are used for matrices, lower boldface letters for vectors, and regular letters for scalars. The set of the first d natural numbers will be denoted by [d] := {1, · · · , d}. Let T ∈ Rd1×···×dn and α ∈ R, T(α) represents the pointwise power operator i.e., (T(α))i1···in = (Tij···in)α. We use Ω > 0 to denote the tensor with Tij···in > 0 for all i1, · · · , in. 1Ω denotes the tensor with all entries equal to 1 on Ω and 0 otherwise.

Definition 1 (Tensor). A tensor is a multidimensional array. The dimension of a tensor is called the order (also called the mode). The space of a real tensor of order n and of size d1 × · · · × dn is denoted as R d1 × · · · × dn. The elements of a tensor T ∈ R d1 × · · · × dn are denoted by Tij···in.

For an order n tensor T can be matricized in n ways by unfolding it along each of the n modes, next we will give the definition for the matricization of a given tensor.

Definition 2 (Matricization of a tensor). The mode-k matricization of tensor T ∈ R d1 × · · · × dn is the matrix,
which is denoted as $T_k \in \mathbb{R}^{d_k \times \prod_j d_j}$, whose columns are composed of all the vectors obtained from $T$ by fixing all indices but $i$th.

In order to illustrate the matricization of a tensor, let us consider the following example.

**Example 1.** Let $T \in \mathbb{R}^{3 \times 4 \times 2}$ with the following frontal slices:

$$T_1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \quad T_2 = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix},$$

then the three mode-$n$ matricizations are

$$T_{(1)} = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \quad T_{(2)} = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix}, \quad T_{(3)} = \begin{bmatrix} 1 & 2 & 3 & \ldots & 10 & 11 & 12 \\ 13 & 14 & 15 & \ldots & 22 & 23 & 24 \end{bmatrix}.$$

**Definition 3 (Folding Operator).** Suppose $T$ be a tensor. The mode-$k$ folding operator of a matrix $M = T(k)$, denoted as $fold_k(M)$, is the inverse operator of the unfolding operator.

**Definition 4 ($\infty$ norm).** Let $T \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_n}$, the $\|T\|_\infty$ is defined as

$$\|T\|_\infty = \max_{i_1, i_2, \ldots, i_n} |T_{i_1i_2\cdots i_n}|.$$

The unit ball under $\infty$ norm is denoted by $B_\infty$.

**Definition 5 (Frobenius norm).** The Frobenius norm for tensor $T \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_n}$ is defined as

$$\|T\|_F = \sqrt{\sum_{i_1, i_2, \ldots, i_n} T_{i_1i_2\cdots i_n}^2}.$$

**Definition 6 (Product Operations).**

- **Mode-$k$ Product:** Mode-$k$ product of tensor $T \in \mathbb{R}^{d_1 \times \cdots \times d_n}$ and matrix $A \in \mathbb{R}^{d_k \times d_k}$ is defined by
  $$T \times_k A = fold_k(A T(k)),$$
  i.e.,
  $$(T \times_k A)_{i_1\ldots i_k-1 j k+1 \ldots i_n} = \sum_{i_k=1}^{d_k} T_{i_1 i_2 \cdots i_k} A_{j i_k}.$$

- **Outer Product:** Let $a_1 \in \mathbb{R}^{d_1}, \ldots, a_n \in \mathbb{R}^{d_n}$. The outer product among these $n$ vectors is a tensor $T \in \mathbb{R}^{d_1 \times \cdots \times d_n}$ defined as:
  $$T = a_1 \otimes \cdots \otimes a_n, \quad T_{i_1, \ldots, i_n} = \prod_{k=1}^n a_k(i_k).$$

- **Kronecker product of matrices:** The Kronecker product of $A \in \mathbb{R}^{k_1 \times l_1}$ and $B \in \mathbb{R}^{k_2 \times l_2}$ is denoted by $A \otimes B$. The result is a matrix of size $(k_1 l_1) \times (k_2 l_2)$ and defined by
  $$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1l}B \\ A_{21}B & A_{22}B & \cdots & A_{2l}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{l1}B & A_{l2}B & \cdots & A_{ll}B \end{bmatrix}.$$

- **Khatri-Rao product:** Given matrices $A \in \mathbb{R}^{d_1 \times r}$ and $B \in \mathbb{R}^{d_2 \times r}$, their Khatri-Rao product is denoted by $A \odot B$. The result is a matrix of size $(d_1 d_2) \times r$ defined by
  $$A \odot B = [a_1 \otimes b_1 \cdots a_r \otimes b_r],$$
  where $a_i$ and $b_i$ stands for the $i$-th column of $A$ and $B$ respectively.

- **Hadamard product:** Given two tensors $T$ and $Y$, both of size $d_1 \times \cdots \times d_n$, their Hadamard product is denoted by $T \odot Y$. The result is also of the size $d_1 \times \cdots \times d_n$, and the elements of $T \odot Y$ are defined as the elementwise tensor product i.e.,
  $$(T \odot Y)_{i_1,\ldots,i_n} = X_{i_1} Y_{i_1 \ldots i_n}.$$

**Definition 7 (Rank-one Tensors).** An $n$-order tensor $T \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_n}$ is rank one if it can be written as the outer product of $n$ vectors, i.e.,

$$T = a_1 \otimes \cdots \otimes a_n.$$

**Definition 8 (Tensor (CP) rank) [20], [21].** The rank of a tensor $X$, denoted rank($X$), is defined as the smallest number of rank-one tensors that generate $X$ as their sum. We use $K_r$ to denote the cone of rank-$r$ tensors.

Different from the case of matrices, the rank of a tensor is presently not understood well. And the problem of computing the rank of a tensor is NP-hard. Next we will introduce a new rank definition related to the tensor.

**Definition 9 (Tucker rank) [21].** Let $X \in \mathbb{R}^{d_1 \times \cdots \times d_n}$. The tuple $(r_1, \ldots, r_n) \in \mathbb{N}^n$, with $r_k = \text{rank}(X(k))$ is called tensor Tucker rank of $X$. We use $K_r$ to denote the cone of Tucker rank $r$ tensors.

**B. Problem Statement**

In order to find a good initialization for total variation method, we would like to solve the following questions.

**Question 1.** Given a deterministic sampling pattern $\Omega$ and corresponding (possibly noisy) observations from the tensor, what type of recovery error can we expect, in what metric, and how may we efficiently implement this recovery?

**Question 2.** Given a sampling pattern $\Omega$, and noisy observations $T_\Omega + Z_\Omega$, for what rank-one weight tensor $H$
can we efficiently find a tensor $\hat{T}$ so that $\|\mathcal{H}(\hat{T} - T)\|_F$ is small compared to $\|\mathcal{H}\|_F$? And how can we efficiently find such weight tensor $\mathcal{H}$, or certify that a fixed $\mathcal{H}$ has this property?

In order to find weight tensor, we consider the optimization problem

$$W := \arg\min_{\mathcal{X} > 0, \text{rank}(\mathcal{X}) = 1} \|\mathcal{X} - \mathbf{1}_\Omega\|_F.$$ 

$W$ can be estimated by using the least square CP algorithm [8], [19]. After we find $W$, then we consider the following optimization problem to estimate $T$:

$$\hat{T} = Y_\Omega^{(1/2)} \square \arg\min_{\text{Tucker}_\text{rank}(\mathcal{T}) = r} \|T - Y_\Omega^{(1/2)} \square Y_\Omega\|_F,$$

(1)

where $Y_\Omega = T_\Omega + Z_\Omega$. As we know, to solve problem (1) is NP-hard. In order to solve (1) in polynomial time, we consider the HOSVD process [13]. Assume that $T$ has Tucker rank $r = [r_1, \cdots, r_n]$. Let

$$\hat{A}_i = \arg\min_{\text{rank}(A) = r_i} \|A - (Y_\Omega^{(1/2)} \square Y_\Omega^{(1/2)})\|_F,$$

and set $\hat{U}_i$ to be the left singular vector matrix of $\hat{A}_i$. Then the estimated tensor is of the form

$$\hat{T} = Y_\Omega^{(1/2)} \square ((Y_\Omega^{(1/2)} \square Y_\Omega^{(1/2)}) \times \hat{U}_1 \hat{U}_1^T \times \cdots \times \hat{U}_n \hat{U}_n^T).$$

In the following, we call our algorithm weighted HOSVD algorithm.

With the output from weighted HOSVD, $\hat{T}$, we can solve the following total variation problem with $\hat{T}$ as initialization:

$$\min_{\mathcal{X}} \|\mathcal{X}\|_{TV}$$

$$\text{s.t. } \mathcal{X}_\Omega = Y_\Omega.$$

This total variation minimization problem is solved by iterative method. We will discuss the details of algorithm and numerical performance in section IV and V.

III. RELATED WORK

In this section, we will briefly step through the history of matrix completion and introduce several relevant studies on tensor completion.

Suppose we have a partially observed matrix $M$ under the low-rank assumption and give a deterministic sampling pattern $\Omega$, the most intuitive optimization problem raised here is:

$$\min_{\mathcal{X}} \text{Rank}(\mathcal{X}),$$

$$\text{s.t. } \mathcal{X}_\Omega = M_\Omega.$$

However, due to the computational complexity (NP-hard) of this minimization problem, researchers developed workarounds by defining new optimization problems which could be done in polynomial time. Two prominent substitutions are the nuclear norm minimization (NMM) and low rank matrix factorization (LRMF):

$$\min_{\mathcal{X}} \|\mathcal{X}\|_*$$

$$\text{s.t. } \mathcal{X}_\Omega = M_\Omega,$$

where $\|\cdot\|_*$ stands for the sum of singular values.

$$\min_{A,B} \|\mathcal{X}_\Omega - AB\|_F$$

$$\text{s.t. } A \in \mathbb{R}^{m \times r}, B \in \mathbb{R}^{r \times n},$$

where $A \in \mathbb{R}^{m \times r}, B \in \mathbb{R}^{r \times n}$ are restricted to be the low rank components.

While the task went up to tensor completion, the low rank assumption became even harder to approach. Some of the recent studies performed matrix completion algorithms on the unfolded tensor and obtained considerable results. For example, [25] introduced nuclear norm to unfolded tensors and took the weighted average for loss function. They proposed several algorithms such as FaLRTC and SiLRTC to solve the minimization problem:

$$\min_{\mathcal{X}} \sum_{i=1}^n \alpha_i \|\mathcal{X}(i)\|_*$$

$$\text{s.t. } \mathcal{X}_\Omega = \mathcal{T}_\Omega.$$

[28] applied low-rank matrix factorization (LRMF) to all-mode unfolded tensors and defined the minimization problem as following:

$$\min_{\mathcal{X},A,B} \sum_{i=1}^n \alpha_i \|\mathcal{X}(i) - A_i B_i\|_F^2$$

$$\text{s.t. } \mathcal{X}_\Omega = \mathcal{T}_\Omega.$$

Where $A = \{A_1, \ldots, A_n\}, B = \{B_1, \ldots, B_n\}$ are the set of low rank matrices with different size according to the unfolded tensor. This method is called TMac and could be solved using alternating minimization.

While researchers often test the performance of their tensor completion algorithms on image/video/MRI data, they started to combine NMM and LRMF with total variation norm minimization when dealing with relevant recovery tasks. For example, [22] introduced the TV regularization into the tensor completion problem:

$$\min_{\mathcal{X},A,B} \sum_{i=1}^n \alpha_i \|\mathcal{X}(i) - A_i B_i\|_F^2 + \mu \|B_3\|_{TV}$$

$$\text{s.t. } \mathcal{X}_\Omega = \mathcal{T}_\Omega.$$

Note that the $B_3$ here only compute the TV-norm for the first 2 modes. For example, assume that $\mathcal{X}$ is a video which can be treated as a 3-order tensor, then this TV-norm only counts the variation within each frame without the variation between frames.
For specific tensors - RGB image data, [24] unfolded the tensor in 2 ways (the horizontal and vertical dimension) and minimized the TV and nuclear norms of each unfolded matrix:

$$\min_{\mathcal{X}} \sum_{i=1}^{2} (\alpha_i \|X_{(i)}\|_2^2 + \mu \|X_{(i)}\|_{\text{TV}}),$$

s.t. $\mathcal{X}_\Omega = \mathcal{T}_\Omega$.

In our experiment, we noticed that for a small percentage of observations, for instance, 50% or more entries are missing, the TV-minimization recovery will produce a similar structure of the original tensor, in the sense of singular values, but NNM will force a large portion of smaller singular values to be zero, which cannot be ignored in the original tensor. Therefore one should be really careful with the choice of minimization problem when performing the completion tasks on a specific dataset. We will discuss the details in the experiment section.

IV. TV MINIMIZATION ALGORITHM

A. Matrix Denoising Algorithm

Total Variation Norm is often discretized by [16]

$$\|u\|_{TV} \approx \sum_{i,j} \sqrt{(\nabla_x u)^2_{i,j} + (\nabla_y u)^2_{i,j}}.$$  

Hence the image denoising problem is defined as:

$$\min_{\mathcal{X}} \sum_{i,j} \sqrt{(\nabla_x M_{i,j})^2 + (\nabla_y M_{i,j})^2} + \lambda \|M - X\|_F.$$  

This can be solved by implementing the algorithm in [9].

B. Tensor Completion with TV

Similar to the image denoising algorithm, we first compute the divergence at each entry and move each entry towards the divergence direction. To keep the existing entries unchanged, we project the observed entries to their original values at each step. We consider the following minimization problem:

$$\min_{\mathcal{X}} \|\mathcal{X}\|_{TV},$$

s.t. $\mathcal{X}_\Omega = \mathcal{T}_\Omega$.

The related algorithm is summarized in Algorithm 1:

**Algorithm 1: Tensor Completion through TV minimization**

**Input:** Incomplete tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$, Sampling pattern $\Omega \subseteq \{0, 1\}^{d_1 \times \cdots \times d_n}$; stepsize $h_k$, threshold $\lambda$; $\mathcal{X}^0 \in \mathbb{R}^{d_1 \times \cdots \times d_n}$.

Set $\mathcal{X}^0 = \mathcal{X}^0 + (\mathcal{T}_\Omega - \mathcal{X}^0)_\Omega$.

for $k = 0 : K$ do

for $i = 1 : n$ do

Find $\nabla_i (\mathcal{X}_{(i)}^{k+1}) = \nabla_i \mathcal{X}_{(i)}^{k+1} - \Delta_i (\mathcal{X}_{(i)}^{k+1})$

$$\Delta_i (\mathcal{X}_{(i)}^{k+1}) = \Delta_i (\mathcal{X}_{(i)}^{k+1}) + \lambda \left(\frac{\nabla_i (\mathcal{X}_{(i)}^{k+1})}{\nabla_i (\mathcal{X}_{(i)}^{k+1})} - 1\right).$$

Hence the divergence is defined as:

$$\sum_{i,j} \sqrt{(\nabla_x M_{i,j})^2 + (\nabla_y M_{i,j})^2} + \lambda \|M - X\|_F.$$  

In Algorithm 1, the Laplacian operator computes the divergence of all-dimension gradients for each point. The shrink operator simply moves the input towards 0 with distance $\lambda$, or formally defined as:

$$\text{shrink}(x, \lambda) = \text{sign}(x) \cdot \max(|x| - \lambda, 0).$$  

For $\mathcal{X}^0$ initialization, simple tensor completion with total variation (TVTC) method would set $\mathcal{X}^0$ to be a zero tensor, i.e. $\mathcal{X}^0 = 0^{d_1 \times \cdots \times d_n}$, but our proposed method will set $\mathcal{X}^0$ to be the result from w-HOSVD. We will show the theoretical and experimental advantage of w-HOSVD in the following section.

V. MAIN RESULTS

In order to show the efficiency $\hat{T}$ as the initialization for total variation algorithm, we only need to show that $\hat{T}$ is close to $\mathcal{T}$. In the following, the bound of $\|\mathcal{W} \ominus (\mathcal{T} - \hat{T})\|_F$ is estimated.

**Theorem 1.** Let $\mathcal{W} = w_1 \otimes \cdots \otimes w_n \in \mathbb{R}^{d_1 \times \cdots \times d_n}$ have strictly positive entries, and fix $\Omega \subseteq [d_1] \times \cdots \times [d_n]$. Suppose that $\mathcal{T} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$ has Tucker-rank $r = [r_1, \ldots, r_n]$ for problem (1). Suppose that $\mathcal{Z}_{(i_1, \ldots, i_n)} \sim \mathcal{N}(0, \sigma^2)$. Then with probability at least $1 - 2^{-|\Omega|/2}$ over the choice of $\mathcal{Z}$,

$$\|\mathcal{W}^{(1/2)} \ominus (\mathcal{T} - \hat{T})\|_F \leq 4\sigma \mu \sqrt{\Omega} \ln(2)$$

$$+ 2\|\mathcal{T}\|_\infty \|\mathcal{W}^{(1/2)} - \mathcal{W}^{(-1/2)} \|_\infty \|\hat{T}\|_F,$$

where $\mu^2 = \max_{(i_1, \ldots, i_n) \in \Omega} \frac{1}{\mathcal{W}_{i_1 \cdots i_n}}$. 


Notice that the upper bound in Theorem 1 is for the optimal output $\hat{T}$ for problems (3), which is general. But the upper bound in Theorem 1 contains no rank information of the underlying tensor $T$. In order to introduce the rank information of the underlying tensor $T$, we restrict our analysis for Problem (8) by considering the HOSVD process, we have the following results.

**Theorem 2** (General upper bound for Tucker rank $r$ tensor). Let $W = w_1 \otimes \cdots \otimes w_n \in \mathbb{R}^{d_1 \times \cdots \times d_n}$ have strictly positive entries, and fix $\Omega \subseteq [d_1] \times \cdots \times [d_n]$. Suppose that $T \in \mathbb{R}^{d_1 \times \cdots \times d_n}$ has Tucker rank $r = [r_1, \ldots, r_n]$. Suppose that $Z_{i_1, \ldots, i_n} \sim \mathcal{N}(0, \sigma^2)$ and let

$$\hat{T} = \mathcal{W}(-1/2) \circ ((\mathcal{W}(-1/2) \circ \mathcal{Y}_{\Omega}) \times \hat{U}_1 \hat{U}_1^T \times \cdots \times \hat{U}_n \hat{U}_n^T)$$

where $\hat{U}_1, \ldots, \hat{U}_n$ are obtained by considering the HOSVD approximation process. Then with probability at least

$$1 - \frac{1}{d_1} \sum_{\substack{i_1, \ldots, i_n \neq \hat{i} \in \mathbb{R} \times \cdots \times \mathbb{R}^n}} \frac{1}{d_j} \cdot \mathbb{P}(\mathcal{W}(\frac{1}{2}) \circ \mathcal{Y}_{\Omega}) \leq$$

$$\left(\frac{\sum_{k=1}^n r_k \log(d_k + \prod_{j \neq k} d_j) \mu_k}{\sigma}\right) \cdot \frac{\|\mathcal{W}(\frac{1}{2}) \circ (\mathcal{T} - \mathcal{T})\|_F}{\|\mathcal{T} - \mathcal{T}\|_F}$$

where

$$\mu_k^2 = \max_{i_1, \ldots, i_n} \left\{ \frac{1}{\mathcal{W}_{i_1, i_2, \ldots, i_n}} \right\} \cdot \frac{1}{\mathcal{W}_{i_1, i_2, \ldots, i_n}}$$

VI. SIMULATIONS

In this section, we conducted numerical simulations to show the efficiency of the proposed weighted HOSVD algorithm first. Then, we will include the experiment results to show that using the weighted HOSVD algorithm results as a initialization of TV minimization algorithm can accelerate the convergence speed of the original TV minimization.

**A. Simulations for Weighted HOSVD**

In this simulation, we have tested our weighted HOSVD algorithm for 3-order tensor of the form $\mathcal{T} = C \times U_1 \times U_2 \times U_3 \times U_4$ under uniform and nonuniform sampling patterns, where $U_i \in \mathbb{R}^{d_i \times r_i}$ and $C \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ with $r_i < d_i$. First of all, we generate $\mathcal{T}$ of the size $100 \times 100 \times 100$ with Tucker rank $r = [r, r, r]$ with $r$ varies from 2 to 10. Then we add Gaussian random noises with $\sigma = 10^{-2}$ to $\mathcal{T}$. Next we generate a sampling pattern $\Omega$ which subsample 10% of the entries. We consider estimates $\hat{T}$, $\hat{T}_p$ and $\hat{T}_w$ by considering

$$\hat{T}_o = \arg \min_{\text{Tucker rank}(X)=r} \|X - \mathcal{Y}_\Omega\|_F,$$

$$\hat{T}_p = \arg \min_{\text{Tucker rank}(X)=r} \|X - \mathcal{Y}_\Omega\|_F, p = \frac{|\Omega|}{d_1 d_2 d_3},$$

$$\hat{T}_w = \mathcal{W}(-1/2) \circ \arg \min_{\text{Tucker rank}(X)=r} \|X - \mathcal{Y}_\Omega\|_F,$$

respectively by using truncated HOSVD algorithm. We give names HOSVD associated with $\hat{T}_o$, HOSVD $p$ associated with $\hat{T}_p$, and w-HOSVD associated with $\hat{T}_w$. For an estimate $\mathcal{T}$, we consider both the weighted and unweighted relative errors:

$$\frac{\|\mathcal{W}(-1/2) \circ (\hat{T} - \mathcal{T})\|_F}{\|\mathcal{W}(-1/2) \circ \mathcal{T}\|_F}$$

We average each error over 20 experiments.

When each entry is sampled randomly uniformly with probability $p$, then we have $\mathbb{E}(\mathcal{Y}_\Omega) = p \mathcal{T}$ which implies that the estimate $\hat{T}_o$ should perform better than $\hat{T}_p$. In Figure 1 we take the sampling pattern to be uniform at random. The estimates $\hat{T}_p$ and $\hat{T}_w$ perform significantly better than $\hat{T}_o$ as expected.

In Figure 2 the set-up is the same as the one in Figure 1 except that the sampling pattern is non-uniformly at random. Then using $\frac{1}{p}$ is not a good weight tensor which as shown in Figure 2 $\hat{T}_p$ works terrible. But $\hat{T}_w$ still works better that $\hat{T}_o$.

**B. Simulations for TV with Initialization from Weighted HOSVD (wHOSVD-TV)**

1) Experimental Setup: In order to show the advantage of weighted HOSVD, we test our proposed algorithm and simple TV minimization method, along with a baseline algorithm on video data. We mask a specific ratio of entries and conduct each completion algorithm in order to obtain the completion results $\hat{T}$. The tested sampling rates (SR) are 0.1, 0.3, 0.5, 0.8. We then compute the relative root mean square error (RSE):

$$RSE = \frac{\|\hat{T} - \mathcal{T}\|_F}{\|\mathcal{T}\|_F}$$

for each method to evaluate their performance. Meanwhile, we compare the average running time until algorithm converges to some preset threshold.

2) Data: In this part, we have tested our algorithm on three video data, see [14]. The video data are tennis serve data from an Olympic Sports Dataset. The three videos are color videos. In our simulation, we use the
same set-up like the one in [14], we pick 30 frames evenly from each video. For each image frame, the size is scaled to $360 \times 480 \times 3$. So each video is transformed into a 4-D tensor data of size $360 \times 480 \times 3 \times 30$. The first frame of each video after preprocessed is illustrated in Figure 3.

3) Numerical Results: The simulation results on the videos are reported in Table I and Figure 4. In existent studies, there are studies performed the same completion task on the same dataset (see [14].) In [14], the ST-HOSVD [27] had the best performance among several low-rank based tensor completion algorithm.

We record the completion error and running time for each completion task and compare them with the previous low-rank based algorithms. One can observe that the TV-based algorithm is more compatible with video data most of the time.

On the other hand, we have implemented total variations with zero filling initialization for the entries which are not observed and with the tensor obtained from weighted HOSVD which are termed TVTC and wHOSVD-TV respectively. The iterative results are shown in Figure 4, which shows that using the result from weighted HOSVD as initialization could notably reduce the iterations of TV-minimization for achieving the convergence threshold ($\|X^k - X^{k-1}\|_F < 10^{-4}$).

C. Discussion

The relation between smoothness pattern and low-rank pattern is mysterious. When studying image-related data,
Table I: The Relative Square Error (RSE) and time spend for different algorithms on video data

| Video | Method   | SR  | RSE    | Time(S) |
|-------|----------|-----|--------|---------|
|       | wHOSVD-TV| 10% | 0.2080 | 82.26   |
|       |          | 30% | 0.1418 | 50.40   |
|       |          | 50% | 0.1045 | 41.31   |
|       |          | 80% | 0.0566 | 33.21   |
|       | ST-HOSVD | 10% | N/A    | N/A     |
|       |          | 30% | 0.1941 | 521.94  |
|       |          | 50% | 0.1381 | 175.82  |
|       |          | 80% | 0.0667 | 128.68  |
|       | wHOSVD-TV| 10% | 0.2694 | 35.21   |
|       |          | 30% | 0.1888 | 21.08   |
|       |          | 50% | 0.1411 | 16.55   |
|       |          | 80% | 0.0767 | 12.88   |
|       | ST-HOSVD | 10% | N/A    | N/A     |
|       |          | 30% | 0.2249 | 1130.77 |
|       |          | 50% | 0.1480 | 1304.31 |
|       |          | 80% | 0.0749 | 976.65  |
|       | wHOSVD-TV| 10% | 0.2198 | 156.5   |
|       |          | 30% | 0.1394 | 87.89   |
|       |          | 50% | 0.0955 | 72.15   |
|       |          | 80% | 0.0470 | 18.44   |
|       | ST-HOSVD | 10% | N/A    | N/A     |
|       |          | 30% | 0.1734 | 560.97  |
|       |          | 50% | 0.1105 | 158.74  |
|       |          | 80% | 0.0594 | 52.09   |

Figure 4: Comparison Between TVTC and wHOSVD-TV on video 1 with SR = 50%.

both patterns are usually taken at the same time and converted to a mixed optimization structure. Through experiments, we find that, with uniform random missing entries, result from total variation minimization on image-like data has singular values closer to the original image.

We randomly mask 70% entries for several grey-scale images and performed both nuclear norm minimization and total variation minimization on the images. Then the nuclear norms for original image, masked image, TV estimates, and NNM estimates are computed, see Figure 5. From this figure, we can see that the image recovered from TV minimization already gives a smaller nuclear norm compared to the original image, while NNM will bring this further away. By observing the singular values of the TV-recovered matrix and of the NNM recovered matrix, we can see that TV-minimization could better capture the smaller singular values, hence better preserve the overall structures of the original matrix. Since both

\[ \| \cdot \|_* \text{ and } \| \cdot \|_{TV} \] are convex functions, the mixed minimization problem with restricted observation entries

\[
\hat{X}_{\text{mix}} = \min_{X} \sum_{i} \alpha_i \|X_{(i)}\|_* + \lambda \|X\|_{TV}
\]
will produces a result whose nuclear norm is between $\|X_{\text{NNM}}\|$ and $\|X_{\text{TV}}\|_*$, where $\hat{X}_{\text{NNM}}$ and $\hat{X}_{\text{TV}}$ are the results from each individual minimization problem (with the same constraint):

$$\hat{X}_{\text{NNM}} = \min_{\hat{X}} \sum_i \alpha_i \|X_{(i)}\|_* ,$$

$$\hat{X}_{\text{TV}} = \min_{\hat{X}} \|X\|_{TV}.$$ 

Unlike user-rating data and synthetic low-rank tensor, the image-like data tends to have a non-trivial tail of singular values. Figure 6 shows the similarity between the original image and TV-recovered image, which gives us hints about the performance comparison between TV-recovery and low-rank recovery.

**ACKNOWLEDGEMENT**

Needell was partially supported by NSF CAREER #1348721 and NSF BIGDATA #1740325.

**REFERENCES**

[1] Jacob Abernethy, Francis Bach, Theodoros Evgeniou, and Jean-Philippe Vert. Low-rank matrix factorization with attributes. arXiv preprint cs/0611124, 2006.

[2] Yonatan Amit, Michael Fink, Nathan Srebro, and Shimon Ullman. Uncovering shared structures in multiclass classification. In Proceedings of the 24th international conference on Machine learning, pages 17–24. ACM, 2007.

[3] Carl J Appellof and Ernest R Davidson. Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents. Analytical Chemistry, 53(13):2053–2056, 1981.

[4] Jian-Feng Cai, Emmanuel J Candés, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. SIAM Journal on optimization, 20(4):1956–1982, 2010.

[5] T Tony Cai, Wen-Xin Zhou, et al. Matrix completion via max-norm constrained optimization. Electronic Journal of Statistics, 10(1):1493–1525, 2016.

[6] Emmanuel J Candès and Yanniv Plan. Matrix completion with noise. Proceedings of the IEEE, 98(6):925–936, 2010.

[7] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. Foundations of Computational Mathematics, 9(6):717, 2009.

[8] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young” decomposition. Psychometrika, 35(3):283–319, 1970.

[9] Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. Theoretical foundations and numerical methods for sparse recovery, 9(263-340):227, 2010.

[10] Pei Chen and David Suter. Recovering the missing components in a large noisy low-rank matrix: Application to sfm. IEEE transactions on pattern analysis and machine intelligence, 26(8):1051–1063, 2004.

[11] Miaomiao Cheng, Liping Jing, and Michael Kwok-Po Ng. A weighted tensor factorization method for low-rank tensor completion. In 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM), pages 30–38. IEEE, 2019.

[12] P Comon. Tensor decompositions: state of the art and applications. math. Signal Process. V (Coventry, 2000), pages 1–24.

[13] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. SIAM journal on Matrix Analysis and Applications, 21(4):1253–1278, 2000.