ROBUST AND PARALLELIZABLE TENSOR COMPLETION BASED ON TENSOR FACTORIZATION AND MAXIMUM CORRENTROPY CRITERION

Yicong He, George K. Atia

Department of Electrical and Computer Engineering
University of Central Florida, Orlando, FL, 32816, USA.
{Yicong.He, George.Atia}@ucf.edu.

ABSTRACT

Robust tensor completion aims to recover a tensor from partially observed noisy entries that may be contaminated with large outliers by exploiting its low-rank property. While there exist several robust tensor completion algorithms, their reliance on singular value decomposition (SVD) limits their scalability. In this paper, we propose a new robust and parallelizable tensor completion method using the tubal rank model. The proposed method rests on tensor factorization, thus averts the costly SVD iterations, and leverages a differentiable, robust correntropy error measure to mitigate the effect of outliers. Leveraging a half-quadratic technique and an alternating steepest descent method, we develop a new SVD-free and parallelizable robust tensor completion algorithm. Numerical results using both synthetic and real data demonstrate the robustness and efficiency of the proposed algorithm.

Index Terms— Robust tensor completion, tensor factorization, correntropy

1. INTRODUCTION

Multi-way data processing has attracted much attention in recent years given the variety and sheer amount of data created from diverse sources at an unprecedented scale. Tensor completion is a popular problem in multi-way data analysis in which the aim is to estimate the missing information from partially observed data. An essential characteristic of real-world multi-way data (such as image and video data) is its low-rank property since it often contains redundant information. This property underlies the ability to successfully perform tensor completion. To date, many tensor completion algorithms have been proposed based on various notions of the tensor rank and were shown to exhibit desirable performance in recovering the missing entries of the tensor data [1, 2, 3, 4, 5, 6, 7].

In dealing with real-world applications, several challenges emerge in tensor completion. One challenge is robustness; the observed data is often fraught with large outliers due to system or human errors. In such settings, the traditional second-order statistics-based algorithms may suffer from serious performance degradation. To enhance robustness, several robust tensor completion algorithms have been proposed to alleviate the negative impact of outliers [8, 9, 10, 11].

Another problem is scalability through parallelism. Parallel computation can greatly improve the algorithm efficiency. Although existing robust tensor completion algorithms exhibit robust performance in the presence of outliers, they largely utilize the matrix nuclear norm for regularization, which requires performing a singular value decomposition (SVD) operation in every iteration. However, because of the high computational burden of SVD on large matrices, these algorithms are generally not amenable to parallel implementation on multi-core systems such as GPUs.

Motivated by these challenges, we propose a parallelizable robust tensor completion method. The approach utilizes tensor factorization [5, 6], so the SVD computation can be avoided. Tensor factorization is theoretically underpinned on the fact that a best tubal rank-r approximation can be obtained from truncation of the tensor singular value decomposition (t-SVD) [12]. Further, to improve the robustness, a correntropy error measure is introduced for tensor completion. Correntropy is a non-linear information-theoretic similarity measure that can provably handle the negative effect of large outliers [13, 14, 15]. Compared with the ℓ_1-norm, correntropy is everywhere differentiable [16]. By utilizing tensor factorization and correntopy, we propose a novel objective function for robust tensor completion under the tubal rank (t-SVD) model. Then, an SVD-free and parallelizable algorithm based on a half-quadratic (HQ) optimization technique [17] and alternating steepest descent [18] is developed.

2. PRELIMINARIES

2.1. Definitions and Notation

In this paper, uppercase script letters are used to denote tensors (e.g., $\mathcal{X}$), and boldface letters to denote matrices (e.g., $\boldsymbol{X}$). Unless stated otherwise, we focus on third order ten-

This work was supported by NSF CAREER Award CCF-1552497 and NSF Award CCF-2106339.

sors, i.e., $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ where n_1, n_2, n_3 are the dimensions of each way of the tensor. The notation $\mathcal{X}(i,:,:), \mathcal{X}(:,i,:), \mathcal{X}(:,:,i)$ denotes the frontal, lateral, horizontal slices of $\mathcal{X}$, respectively. $\mathcal{X}_{ijk}$ denotes the (i,j,k)-th entry of tensor $\mathcal{X}$. The Frobenius norm of tensor $\mathcal{X}$ is defined as $\|\mathcal{X}\|_F = \sqrt{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} |\mathcal{X}_{ijk}|^2}$. $\bar{\mathcal{X}} = \text{fft}(\mathcal{X}, [\,], 3)$ denotes the Fourier transform along the third dimension. Similarly, $\mathcal{X} = \text{ifft}(\bar{\mathcal{X}}, [\,], 3)$ denotes the inverse transform. The block diagonal matrix $\bar{\boldsymbol{X}} \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$ is obtained from tensor $\bar{\mathcal{X}}$ with each diagonal block $\boldsymbol{X}^{(i)} := \mathcal{X}(:,:,i)$.

Definition 1 (t-product [12])**.** *The t-product $\mathcal{A} * \mathcal{B}$ of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is the tensor of size $n_1 \times n_4 \times n_3$ given by*

$$\mathcal{A} * \mathcal{B} = \text{fold}(\text{bcirc}(\mathcal{A}) \cdot \text{unfold}(\mathcal{B})) \,,$$

where the operator $\text{unfold}(\cdot)$ *maps the tensor $\mathcal{X}$ to a matrix $\tilde{\boldsymbol{X}} = \text{unfold}(\mathcal{X}) = \left[\boldsymbol{X}^{(1)\top} \boldsymbol{X}^{(2)\top} \cdots \boldsymbol{X}^{(n_3)\top}\right]^\top$ and its inverse operator* $\text{fold}(\cdot)$ *is defined as* $\text{fold}(\tilde{\boldsymbol{X}}) = \mathcal{X}$.

The block circulant operator $\text{bcirc}(\cdot)$ is defined as in [12].

Theorem 1 (t-SVD [12, 19])**.** *The tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ can be factorized as $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*$, where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$, $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal, and $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor, i.e., each of the frontal slices of $\mathcal{S}$ is a diagonal matrix. The diagonal entries in $\mathcal{S}(:,:,1)$ are called the singular values of $\mathcal{A}$.*

Definition 2 (Tensor tubal-rank and tensor multi-rank [20])**.** *For any tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tensor tubal-rank is the number of non-zero singular tubes of $\mathcal{S}$ from the t-SVD, i.e.,*

$$\text{rank}_t(\mathcal{A}) = \#\{i : \mathcal{S}(i,i,:) \neq 0\} \,.$$

The multi-rank rank of $\mathcal{A}$ is a vector $\boldsymbol{r} \in \mathbb{R}^{n_3 \times 1}$ with i-th element $r_i = \text{rank}(\bar{\boldsymbol{A}}^{(i)})$. Specifically, $\text{rank}_t(\mathcal{A}) = \max(r_1, \cdots, r_{n_3})$.

2.2. Tensor Completion using Tensor Factorization

Given a partially observed tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with observed set of entries $\boldsymbol{\Omega} \subseteq [n_1] \times [n_2] \times [n_3]$, tensor completion solves the following minimization problem leveraging the low-rank property of the tensor

$$\min_{\mathcal{Z} \in \mathbb{R}^{n_1 \times n_2 \times n_3}} \text{rank}_t(\mathcal{Z}), \text{ s.t. } \mathcal{P} \circ (\mathcal{Z} - \mathcal{M}) = 0 \,, \quad (1)$$

where $\circ$ denotes the Hadamard (element-wise) product of two tensors of the same size. For indicator tensor $\mathcal{P}$, $\mathcal{P}_{ijk}$ is set to 1 if $(i,j,k) \in \boldsymbol{\Omega}$ and 0 otherwise.

The problem in (1) is NP-hard. To address this problem, several methods were proposed using convex relaxation [4]. Specifically, (1) is relaxed to obtain a nuclear norm-based convex optimization problem. However, nuclear norm-based algorithms need to compute an SVD in each iteration, which may incur a high computational cost. Recently, a tensor factorization based method was developed in [5, 6] under the tubal rank (t-SVD) model. Specifically, the recovered tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ can be factorized into the t-product of two tensors $\mathcal{X} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\mathcal{Y} \in \mathbb{R}^{r \times n_2 \times n_3}$, where r is the tubal rank of $\mathcal{M}$ [5]. The tensor factorization then solves tensor completion by utilizing the objective function

$$\min_{\mathcal{X}, \mathcal{Y}} J(\mathcal{X}, \mathcal{Y}) := \|\mathcal{P} \circ (\mathcal{M} - \mathcal{X} * \mathcal{Y})\|_F^2 \,. \quad (2)$$

2.3. Correntropy

Correntropy is a local and non-linear similarity measure between two random variables within a 'window' in the joint space determined by the kernel width. Given two random variables X and Y with a finite number of samples $\{x_i, y_i\}_{i=1}^N$, and using the Gaussian kernel as the kernel function, the correntropy can be approximated by

$$\hat{V}(X, Y) = \frac{1}{N} \sum\nolimits_{i=1}^{N} \exp\left(-\frac{e_i^2}{2\sigma^2}\right) \,, \quad (3)$$

where $e_i = x_i - y_i$. Compared with the ℓ_2-norm based second-order statistic of the error, the correntropy involves all the even moments of the difference between X and Y and is insensitive to outliers [13].

3. PROPOSED ALGORITHM

In this section, we develop our robust and parallelizable algorithm using tensor factorization and correntropy. By replacing the ℓ_2 error measure in (2) with correntropy, we obtain the maximum correntropy criterion-based optimization problem

$$\max_{\mathcal{X}, \mathcal{Y}} J(\mathcal{X}, \mathcal{Y}) := \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \mathcal{P}_{ijk} G_\sigma(\mathcal{M}_{ijk} - (\mathcal{X} * \mathcal{Y})_{ijk}) \quad (4)$$

where $G_\sigma(x) = \exp(-\frac{x^2}{2\sigma^2})$. In general, (4) is non-convex and is difficult to be directly optimized. To tackle this difficulty, we utilize a half-quadratic (HQ) optimization technique and alternating steepest descent to solve the problem.

3.1. Optimization via Half-quadratic Minimization

According to the Proposition 1 in [14], we have

$$\max_{e} \sigma^2 G_\sigma(e) = \min_{e,w} \left(e^2 w + \varphi(w)\right) \,, \quad (5)$$

where φ is a convex conjugated function of $G_\sigma(e)$ and w is an auxiliary variable. Specifically, for a fixed e, the minimum of w is reached at $w = G_\sigma(e)$. Thus, optimizing the non-convex correntropy in terms of e is equivalent to minimizing an augmented cost function in an enlarged parameter space $\{e, w\}$. Therefore, by substituting (5) in (4), the correntropy-based objective function in (4) can be transformed to

$$\max_{\mathcal{X},\mathcal{Y}} \sigma^2 J(\mathcal{X},\mathcal{Y}) = \min_{\mathcal{X},\mathcal{Y},\mathcal{W}} J_{\mathrm{HQ}}(\mathcal{X},\mathcal{Y},\mathcal{W}) \ , \tag{6}$$

where

$$J_{\mathrm{HQ}}(\mathcal{X},\mathcal{Y},\mathcal{W}) = \|\sqrt{\mathcal{W}}\circ\mathcal{P}\circ(\mathcal{M}-\mathcal{X}*\mathcal{Y})\,\|_F^2+\Phi_{\mathbf{\Omega}}(\mathcal{W}) \tag{7}$$

and $\Phi_{\mathbf{\Omega}}\left(\mathcal{W}\right)=\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\mathcal{P}_{ijk}\varphi\left(\mathcal{W}_{ijk}\right)$. Therefore, the optimization is reformulated as a weighted tensor factorization problem. Next, we develop a method based on alternating steepest descent (ASD) [18] to efficiently solve (7).

3.2. Alternating Steepest Descent-based Algorithm

The algorithm is described as follows:

1. Optimizing $\mathcal{W}$: According to [14, 17] and (5), given the fixed factors $\mathcal{X}$ and $\mathcal{Y}$, the optimal solution of $\mathcal{W}_{ijk}$ for $(i,j,k)\in\mathbf{\Omega}$ can be obtained as

$$\mathcal{W}_{ijk} = G_\sigma(\mathcal{M}_{ijk} - (\mathcal{X}*\mathcal{Y})_{ijk}), (i,j,k)\in\mathbf{\Omega} \ . \tag{8}$$

Since computing $\mathcal{W}_{ijk}$ for $(i,j,k)\notin\mathbf{\Omega}$ does not affect the solution of (4) due to the multiplication with $\mathcal{P}$, in the following parts we use $\mathcal{W}_{ijk}$ for all the entries to simplify the expressions.

2. Update $\mathcal{X}$ and $\mathcal{Y}$: By fixing $\mathcal{W}$, we solve the following problem

$$\min_{\mathcal{X},\mathcal{Y}} \|\sqrt{\mathcal{W}}\circ\mathcal{P}\circ(\mathcal{M}-\mathcal{X}*\mathcal{Y})\,\|_F^2 \ . \tag{9}$$

Then, according to Definition 1, (9) can be rewritten as

$$\min_{\mathcal{X},\mathcal{Y}} \|\sqrt{\tilde{\boldsymbol{W}}}\circ\tilde{\boldsymbol{P}}\circ(\tilde{\boldsymbol{M}}-\mathrm{bcirc}(\mathcal{X})\tilde{\boldsymbol{Y}})\|_F^2 \ . \tag{10}$$

Using the block-circulant diagonalization [20], we obtain

$$\mathrm{bcirc}(\mathcal{X})\tilde{\boldsymbol{Y}} = (\boldsymbol{F}_{n_3}^{-1}\otimes\boldsymbol{I}_{n_1})\bar{\boldsymbol{X}}\hat{\boldsymbol{Y}} = \boldsymbol{U}\hat{\boldsymbol{Y}} \ , \tag{11}$$

where $\boldsymbol{U} = \boldsymbol{F}^{-1}\bar{\boldsymbol{X}}$, $\boldsymbol{F}^{-1} = \boldsymbol{F}_{n_3}^{-1}\otimes\boldsymbol{I}_{n_1}$ and $\hat{\boldsymbol{A}} = \mathrm{unfold}(\bar{\mathcal{A}})$. $\boldsymbol{F}_{n_3}\in\mathbb{C}^{n_3\times n_3}$ is the Discrete Fourier Transform (DFT) matrix, $\otimes$ is the Kronecker product and $\boldsymbol{I}_{n_1}\in\mathbb{R}^{n_1\times n_1}$ is the identity matrix. $\boldsymbol{F}_{n_3}^{-1}$ can be computed as $\boldsymbol{F}_{n_3}^{-1} = \boldsymbol{F}_{n_3}^{*}/n_3$, where $\boldsymbol{X}^*$ denotes the Hermitian transpose of $\boldsymbol{X}$. Finally, (10) can be reformulated as

$$\min J(\boldsymbol{U},\hat{\boldsymbol{Y}}) := \frac{1}{2}\|\sqrt{\tilde{\boldsymbol{W}}}\circ\tilde{\boldsymbol{P}}\circ(\tilde{\boldsymbol{M}}-\boldsymbol{U}\hat{\boldsymbol{Y}})\|_F^2 \ . \tag{12}$$

Here, a multiplicative factor of $\frac{1}{2}$ is added for convenience. The partial derivative of $J(\boldsymbol{U},\hat{\boldsymbol{Y}})$ with respect to $\boldsymbol{U}$ can be computed as

$$\boldsymbol{g}_{\boldsymbol{U}} = \partial J/\partial\boldsymbol{U} = -\tilde{\boldsymbol{W}}\circ\tilde{\boldsymbol{P}}\circ(\tilde{\boldsymbol{M}}-\boldsymbol{U}\hat{\boldsymbol{Y}})\hat{\boldsymbol{Y}}^* \ . \tag{13}$$

Note that $\bar{\boldsymbol{X}} = \boldsymbol{F}\boldsymbol{U}$ is a block diagonal matrix with $\boldsymbol{F} = \boldsymbol{F}^{-1}\times n_3$. Then, we follow the method in [7] and force $\bar{\boldsymbol{X}}$ to be block diagonal. In particular, by defining the operator $\mathrm{bdiagz}(\cdot)$, which sets the non-block-diagonal entries of a matrix to zero, the updated gradient can be obtained as

$$\boldsymbol{g}'_{\boldsymbol{U}} = \boldsymbol{F}^{-1}\,\mathrm{bdiagz}(\boldsymbol{F}\boldsymbol{g}_{\boldsymbol{U}}) \ . \tag{14}$$

The steepest descent step size $\mu'_{\boldsymbol{U}}$ for $\boldsymbol{g}'_{\boldsymbol{U}}$ can be obtained using exact line-search

$$\mu'_{\boldsymbol{U}} = \|\boldsymbol{g}'_{\boldsymbol{U}}\|_F^2/\|\sqrt{\tilde{\boldsymbol{W}}}\circ\tilde{\boldsymbol{P}}\circ(\boldsymbol{g}'_{\boldsymbol{U}}\hat{\boldsymbol{Y}})\|_F^2 \tag{15}$$

and the matrix $\boldsymbol{U}$ can be updated as

$$\boldsymbol{U}^{t+1} = \boldsymbol{U}^t - \mu'^{t}_{\boldsymbol{U}}\boldsymbol{g}'^{t}_{\boldsymbol{U}} \ . \tag{16}$$

Similarly, by fixing $\boldsymbol{U}$, the partial derivative of $\boldsymbol{J}$ w.r.t. $\hat{\boldsymbol{Y}}$ and the corresponding step size can be obtained as

$$\begin{aligned} \boldsymbol{g}_{\hat{\boldsymbol{Y}}} &= \partial J/\partial\hat{\boldsymbol{Y}} = -\boldsymbol{U}^*(\tilde{\boldsymbol{W}}\circ\tilde{\boldsymbol{P}}\circ(\tilde{\boldsymbol{M}}-\boldsymbol{U}\hat{\boldsymbol{Y}})) \ , \\ \mu_{\hat{\boldsymbol{Y}}} &= \|\boldsymbol{g}_{\hat{\boldsymbol{Y}}}\|_F^2/\|\sqrt{\tilde{\boldsymbol{W}}}\circ\tilde{\boldsymbol{P}}\circ(\boldsymbol{U}\boldsymbol{g}_{\hat{\boldsymbol{Y}}})\|_F^2 \ . \end{aligned} \tag{17}$$

The above update process could suffer from slow convergence speed when directly applied to image and video completion tasks. To tackle this problem, following [18], we scale the gradient descent direction for $\hat{\boldsymbol{Y}}$ in (17) by $(\boldsymbol{U}^*\boldsymbol{U})^{-1}$, i.e., $\boldsymbol{g}'_{\hat{\boldsymbol{Y}}} = (\boldsymbol{U}^*\boldsymbol{U})^{-1}\boldsymbol{g}_{\hat{\boldsymbol{Y}}}$, and the corresponding step size $\mu'_{\hat{\boldsymbol{Y}}}$ with exact line-search is

$$\mu'_{\hat{\boldsymbol{Y}}} = \langle\boldsymbol{g}_{\hat{\boldsymbol{Y}}},\boldsymbol{g}'_{\hat{\boldsymbol{Y}}}\rangle/\|\sqrt{\tilde{\boldsymbol{W}}}\circ\tilde{\boldsymbol{P}}\circ(\boldsymbol{U}\boldsymbol{g}'_{\hat{\boldsymbol{Y}}})\|_F^2 \ , \tag{18}$$

where $\langle\boldsymbol{A},\boldsymbol{B}\rangle := \sum_{1\le i,j\le n} A_{ij}^* B_{ij}$. Therefore, the matrix $\hat{\boldsymbol{Y}}$ at the t-th iteration can be updated as

$$\hat{\boldsymbol{Y}}^{t+1} = \hat{\boldsymbol{Y}}^{t} - (1-\lambda)\mu^{t}_{\hat{\boldsymbol{Y}}}\boldsymbol{g}^{t}_{\hat{\boldsymbol{Y}}} - \lambda\mu'^{t}_{\hat{\boldsymbol{Y}}}\boldsymbol{g}'^{t}_{\hat{\boldsymbol{Y}}} \ , \tag{19}$$

where $0\le\lambda\le 1$ is a free parameter to be chosen.

We call the above algorithm 'Half-Quadratic based Tensor Completion by Alternating Steepest Descent' (HQ-TCASD), and the pseudocode is summarized in Algorithm 1. We should remark that the matrices $\boldsymbol{U}(\bar{\boldsymbol{X}})$ and $\hat{\boldsymbol{Y}}$ have a block structure, so the matrix computation can be processed block-by-block in parallel. Also, for a tensor $\mathcal{A}$ we have $\boldsymbol{F}\tilde{\boldsymbol{A}} = \mathrm{unfold}(\mathrm{fft}(\mathcal{A},[\,],3))$, thus the conventional Fast Fourier transform (FFT) operation can be used in (14) instead of matrix multiplication to further speed up the computation.

Algorithm 1 HQ-TCASD

```
Input: Indicator tensor $\mathcal{P}$, partially observed tensor $\mathcal{P}\circ\mathcal{M}$,
    rank $r$ and parameter $\sigma$, $\lambda$.
initial matrices $\boldsymbol{U}^0$ and $\hat{\boldsymbol{Y}}^0$, $t=0$
repeat
    compute $\mathcal{W}^{t+1}$ using (8).
    compute $\boldsymbol{U}^{t+1}$ using (16).
    compute $\hat{\boldsymbol{Y}}^{t+1}$ using (19).
    $t=t+1$
until stopping criterion is satisfied
Output: Recovered tensor $\hat{\mathcal{M}} = \mathrm{fold}(\boldsymbol{U}^t\hat{\boldsymbol{Y}}^t)$.
```

4. EXPERIMENTS

In this section, we evaluate the performance of the proposed HQ-TCASD algorithm. We compare to existing tensor completion algorithms, including TCTF [5] and TNN [4], and robust tensor completion algorithms, including SNN-L1 [8], SNN with Welsch loss (W-ST) [9], TRNN-L1 [10] and TNN-L1 [11]. In the experiments, the adaptive kernel width selection method [15] is applied to HQ-TCASD and W-ST. The run-time of the proposed methods on a GPU (designated with suffix '-P') is reported by simply using the 'gpuArray' data structure in MATLAB. All algorithms are implemented using MATLAB r2019b on a standard 16-GB memory PC with a 2.6-GHz CPU and an NVIDIA RTX3070 GPU, and the parameters for each algorithm are tuned to achieve the best performance in each task. More experimental results are included in an extended version of this work [21].

4.1. Synthetic Data

First, we compare the performance of tubal rank-based methods using synthetic data. The tensor $\mathcal{M}$ with tubal rank r is obtained by the t-product of two tensors whose entries are generated from a zero mean Gaussian distribution with unit variance. The observed entries are randomly and uniformly selected with sampling rate 0.5. The performance is evaluated using the relative error $\|\hat{\mathcal{M}} - \mathcal{M}\|_F / \|\mathcal{M}\|_F$, where $\hat{\mathcal{M}}$ is the recovered tensor. The tensor size is set such that $n_1 = n_2$ and $n_3 = 20$. The observed entries are perturbed by noise generated from a Gaussian mixture model (GMM) with probability density function (pdf) $0.9N(0, 0.01) + 0.1N(0, 1)$. The rank r is set to $0.05n_1$. We gradually increase n_1 from 100 to 1000 and average the relative error over 20 Monte Carlo runs. The average relative error and running time are shown in Fig. 1. The proposed algorithm consistently yields significantly lower relative error and smaller computation time than other algorithms. Further, the parallel computation can speed up the computation of HQ-TCASD by an order of magnitude.

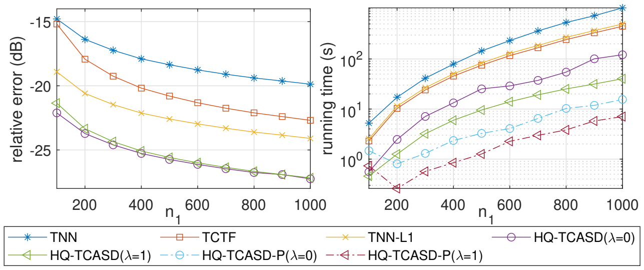

Fig. 1. Average relative error (left) and average run-time (right) as function of n_1 under GMM noise.

4.2. Real Data

Second, we investigate the performance using real image and video data. We evaluate the robust completion performance of the different algorithms using two images 'cow' and 'horse' with resolution 1920 × 1080, and two 30-frame gray-scale videos 'flamingo' and 'scooter' with resolution 1280 × 720 from the DAVIS 2016 dataset [22] . The observed pixels are randomly and uniformly selected with sampling rate 0.5. For 'cow' and 'flamingo', GMM noise with pdf $0.8N(0, 0.001) + 0.2N(0, 1)$ is added to the observed pixels. For 'horse' and 'scooter', 10% of the observed pixels are perturbed by Salt and Pepper noise, while the remaining observed pixels are contaminated with Gaussian noise with pdf $N(0, 0.001)$.

The multi-rank for HQ-TCASD is set to $[120, 20, 20]$ and $[80, 80, \ldots, 80]$ for the image and video data, respectively. The parameter λ is set to 0.2. The average PSNR and run-time over 20 Monte Carlo runs are shown in Fig. 2. HQ-TCASD achieves the highest PSNR for all data, and the parallel computation reduces the computational cost considerably compared with non-parallel algorithms. Examples of the recovered images for 'cow' and 'flamingo' are shown in Fig. 3. As shown, HQ-TCASD yields visually clearer texture than the other methods.

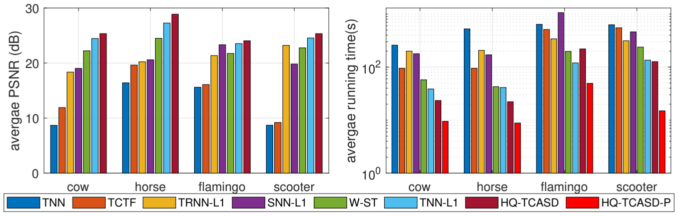

Fig. 2. Average PSNR (left) and run-time (right) on four data.

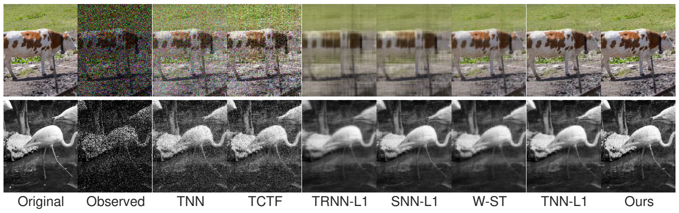

Fig. 3. Images/frames (cropped) using different algorithms.

5. CONCLUSION

We proposed a novel robust and parallelizable tensor completion method. The approach rests on tensor factorization, thereby avoids the costly computation of SVD, and leverages the robust correntropy measure to alleviate the negative impact of large outliers. An efficient robust tensor completion algorithm, HQ-TCASD, is proposed based on a half-quadratic minimization technique and an alternating steepest descent method. Experiments with both synthetic and real data demonstrate the efficiency and superior performance of the proposed method compared to existing robust completion algorithms.

6. REFERENCES

[1] Ledyard R Tucker, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[2] Hiroyuki Kasai and Bamdev Mishra, “Low-rank tensor completion: a riemannian manifold preconditioning approach,” in *International Conference on Machine Learning*, 2016, pp. 1012–1021.

[3] Yangyang Xu, Ruru Hao, Wotao Yin, and Zhixun Su, “Parallel matrix factorization for low-rank tensor completion,” *Inverse Problems and Imaging*, vol. 9, no. 2, pp. 601–624, 2015.

[4] Zemin Zhang and Shuchin Aeron, “Exact tensor completion using t-svd,” *IEEE Transactions on Signal Processing*, vol. 65, no. 6, pp. 1511–1526, 2016.

[5] Pan Zhou, Canyi Lu, Zhouchen Lin, and Chao Zhang, “Tensor factorization for low-rank tensor completion,” *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1152–1163, 2017.

[6] Xiao-Yang Liu, Shuchin Aeron, Vaneet Aggarwal, and Xiaodong Wang, “Low-tubal-rank tensor completion using alternating minimization,” *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1714–1737, 2019.

[7] Kyle Gilman and Laura Balzano, “Grassmannian optimization for online tensor completion and tracking in the t-svd algebra,” *arXiv preprint arXiv:2001.11419*, 2020.

[8] Donald Goldfarb and Zhiwei Qin, “Robust low-rank tensor recovery: Models and algorithms,” *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 1, pp. 225–253, 2014.

[9] Yuning Yang, Yunlong Feng, and Johan AK Suykens, “Robust low-rank tensor recovery with regularized redescending m-estimator,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 9, pp. 1933–1946, 2015.

[10] Huyan Huang, Yipeng Liu, Zhen Long, and Ce Zhu, “Robust low-rank tensor ring completion,” *IEEE Transactions on Computational Imaging*, vol. 6, pp. 1117–1126, 2020.

[11] Qiang Jiang and Michael Ng, “Robust low-tubal-rank tensor completion via convex optimization.,” in *IJCAI*, 2019, pp. 2649–2655.

[12] Misha E Kilmer and Carla D Martin, “Factorization strategies for third-order tensors,” *Linear Algebra and its Applications*, vol. 435, no. 3, pp. 641–658, 2011.

[13] Weifeng Liu, Puskal P Pokharel, and José C Príncipe, “Correntropy: Properties and applications in non-gaussian signal processing,” *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5286–5298, 2007.

[14] Ran He, Wei-Shi Zheng, and Bao-Gang Hu, “Maximum correntropy criterion for robust face recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1561–1576, 2010.

[15] Yicong He, Fei Wang, Yingsong Li, Jing Qin, and Badong Chen, “Robust matrix completion via maximum correntropy criterion and half-quadratic optimization,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 181–195, 2019.

[16] Badong Chen, Xi Liu, Haiquan Zhao, and Jose C Principe, “Maximum correntropy kalman filter,” *Automatica*, vol. 76, pp. 70–77, 2017.

[17] Mila Nikolova and Michael K Ng, “Analysis of half-quadratic minimization methods for signal and image recovery,” *SIAM Journal on Scientific computing*, vol. 27, no. 3, pp. 937–966, 2005.

[18] Jared Tanner and Ke Wei, “Low rank matrix completion by alternating steepest descent methods,” *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 417–429, 2016.

[19] Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan, “Tensor robust principal component analysis with a new tensor nuclear norm,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 925–938, 2019.

[20] Misha E Kilmer, Karen Braman, Ning Hao, and Randy 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*, vol. 34, no. 1, pp. 148–172, 2013.

[21] Yicong He and George K Atia, “Robust low-tubal-rank tensor completion based on tensor factorization and maximum correntopy criterion,” *arXiv preprint arXiv:2010.11740*, 2022.

[22] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung, “A benchmark dataset and evaluation methodology for video object segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 724–732.