ONLINE TENSOR DECOMPOSITION AND IMPUTATION FOR COUNT DATA

Chang Ye and Gonzalo Mateos

Dept. of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA

ABSTRACT

Unveiling low-dimensional latent structure by means of multilinear decompositions of tensor data is central to data analytics tasks at the confluence of signal processing, machine learning and data mining. However, increasingly noisy, incomplete, and heterogeneous datasets (that deviate from e.g., Gaussian distributional assumptions) as well as the need for real-time processing of streaming data pose major challenges to this end. In this context, the present paper develops a novel online (adaptive) algorithm to obtain three-way decompositions of low-rank, Poisson-distributed tensors. Such (possibly incomplete) streams of count data arise with various applications including traffic engineering, computer network monitoring, genomics, photonics and satellite imaging. The proposed estimator minimizes a Poisson log-likelihood cost along with a separable regularizer of the PARAFAC decomposition factors, to trade-off fidelity for complexity of the approximation captured by the decomposition's rank. Leveraging stochastic gradient descent iterations, a scalable, online algorithm is developed to learn the decomposition factors on-the-fly and perform data imputation as a byproduct. Preliminary numerical tests with simulated data and solar flare video confirm the efficacy of the proposed tensor imputation algorithm, as well as its convergence to the batch estimator benchmark.

***Index Terms*—** Low-rank tensor, PARAFAC model, streaming count data, online imputation algorithm, missing data.

1. INTRODUCTION

As data collected from e.g., online social media, ubiquitous sensing, as well as high-throughput genome sequencing technologies become increasingly voluminous, complex and heterogeneous [1], it is not uncommon to encounter datasets indexed by three or more variables giving rise to a tensor (also known as multi-way array) [2, 3]. In various applications one of these variables indexes time [4–6], and sizable portions of the data are missing due to (intentional) subsampling for faster acquisition, privacy considerations, or sensing and communication errors [7–9]. Moreover, for non-negative integer-valued measurements and modeling of sparse multilinear count data, the workhorse Gaussian model and its induced least-squares criterion tend to be inappropriate [10–13].

Problem outline and envisioned applications. Accordingly, the desiderata for extracting actionable information from streaming and incomplete multiway data are low-complexity, online algorithms capable of unraveling latent structures through parsimonious (e.g., low-rank) decompositions; see also [2, 3] for recent tutorial treatments on tensor decompositions including the parallel factor analysis (PARAFAC) model. The focus here is to develop an online (adaptive) algorithm for decomposing low-rank tensors from possibly incomplete, streaming Poisson-distributed data. Time-indexed count data arises in application domains such as computer network monitoring, temporal recommendation systems, genome sequencing, traffic engineering, discovering latent influences among social communities, as well as with video processing whereby frames are acquired using some optical imaging technologies.

Relation to prior work and contributions. The problem of identifying low-dimensional subspace structure from streaming data has a long history in signal and array processing, e.g., [14]. Recent advances consider high-dimensional matrix data that could be incomplete and corrupted by outliers [15–19]; see [8] for a recent review. On a related note, the influential work in [20] considered online dictionary learning for image denoising (without missing data). Deviating from the Gaussian model, dimensionality reduction from streaming categorical data was considered in [21]. Algorithms for subspace tracking from Poisson-distributed count data were developed in [11]. In the tensor case, online algorithms for imputation and low-rank decomposition of multiway Gaussian data were puth forth in [5]; see also [4, 6] for early adaptive algorithms implementing PARAFAC of full tensor observations. For low-rank Poisson tensor data, completion algorithms in a batch setup have been proposed in [10].

Unlike these works, here we develop for the first time an online (adaptive) algorithm to obtain three-way decompositions of low-rank, Poisson-distributed tensors with missing entries. The proposed estimator minimizes a Poisson log-likelihood cost along with a separable regularizer of the PARAFAC decomposition factors [10], to trade-off fidelity for complexity of the approximation captured by the decomposition's rank. Leveraging stochastic gradient descent iterations, a scalable, online algorithm is developed in Section 3 to learn the decomposition factors on-the-fly and perform data imputation as a byproduct. Preliminary numerical tests with simulated data and solar flare video frames [22] confirm the efficacy of the proposed tensor imputation algorithm (Section 4), as well as its convergence to the batch estimator benchmark in [10]. Concluding remarks are given in Section 5.

Notation. Bold uppercase (lowercase) letters are used to represent matrices (column vectors). Underlined bold uppercase letters denote tensors. As an example, $\mathbf{x}$, $\mathbf{X}$ are $\underline{\mathbf{X}}$, denote a vector, matrix, and tensor, respectively. Matrices with a subscript are used to denote a slice of the tensor, e.g., $\mathbf{X}_t$. Symbols $\otimes, \odot, \circ, \circledast$ denote the Kronecker, Khatri-Rao, outer product, and Hadamard (entry-wise) product, respectively. For matrices, $\|\cdot\|_F$ and $\|\cdot\|_*$ stand for the Frobenius and nuclear norms.

2. PRELIMINARIES AND PROBLEM STATEMENT

2.1. PARAFAC Decomposition

In matrix case, a rank-one matrix $\mathbf{Z} \in \mathbb{R}^{M\times N}$ can be decomposed as $\mathbf{a} \circ \mathbf{b}^T$, where $\mathbf{a} \in \mathbb{R}^{M\times 1}$ and $\mathbf{b} \in \mathbb{R}^{N\times 1}$ are two vectors. Bollowing the concept from that, if given the third vector $\mathbf{c} \in \mathbb{R}^{T\times 1}$, a rank-one tensor $\underline{\mathbf{Z}} \in \mathbb{R}^{M\times N\times T}$ can be formed as $\underline{\mathbf{Z}} = \mathbf{a}\circ\mathbf{b}\circ\mathbf{c}$ with the (m, n, t)-th entry given by $Z_{mnt} = a_m b_n c_t$.

Then a R-rank tensor $\underline{\mathbf{M}} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$, where R is the minimum number of rank-one tensors for the decomposition.

Work in this paper was supported in part by the NSF award CCF-1750428. Emails: cye7@ur.rochester.edu, gmateosb@ece.rochester.edu.

The paralle factor decomposition (PARAFAC) is the most basic tensor model because it directly considers the low rank approximation of a tensor

$$\underline{\mathbf{X}} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \tag{1}$$

2.2. Poisson Distributed Tensor Model For Count Data

A natural way to formulate the non-negative integer-value data observed from independent count event is to assume that the observation $\underline{\mathbf{Y}}$ satisfies the Poisson distribution with parameter $\underline{\mathbf{X}}$, e.g. $\underline{\mathbf{Y}} \sim \text{Pois}(\underline{\mathbf{X}})$. Pois$(\cdot)$ denotes the tensor Poisson distribution,

$$\Pr\left[y_{mnt} = k\right] = \frac{x_{mnt}^{k} e^{-x_{mnt}}}{k!} \tag{2}$$

In reality, we may only have access to a subset of $\underline{\mathbf{Y}}$, e.g. $\{\underline{\mathbf{\Delta}} \circledast \underline{\mathbf{Y}}, \underline{\mathbf{\Delta}}\}$, where the observation mask $\underline{\mathbf{\Delta}}$ is a binary tensor and $\delta_{mnt} = 1$, when Y_{mnt} is observed, otherwise $\delta_{mnt} = 0$. In many applications, we receive observations continuously, such as a live video. To process the steaming data online, we consider the expectation or the average of the log likelihood instead of the whole log likelihood function. which can be formulate as

$$\mathbf{L}_T(\underline{\mathbf{Y}}, \underline{\mathbf{\Delta}}; \underline{\mathbf{X}}) = \mathbf{E}_t[\mathbf{l}_t(\mathbf{Y}_t, \mathbf{\Delta}_t; \mathbf{X}_t)] = \frac{1}{T}\sum_{t=1}^{T} \mathbf{l}_t(\mathbf{Y}_t, \mathbf{\Delta}_t; \mathbf{X}_t)$$
(3)

Where $\mathbf{l}_t(\mathbf{X}_t; \mathbf{Y}_t, \mathbf{\Delta}_t)$ is the log likelihood function of the partial observation $\{\mathbf{\Delta}_t \circledast \mathbf{Y}_t, \mathbf{\Delta}_t\}$ at time t,

$$\mathbf{l}_t(\mathbf{Y}_t, \mathbf{\Delta}_t; \mathbf{X}_t) = \sum_{mn} \delta_{mnt}(x_{mnt} - y_{mnt}\log x_{mnt}) \tag{4}$$

We also assume that tensor $\underline{\mathbf{X}}$ has low-rank structure, so we want to solve the following problem

$$\begin{aligned} \hat{\underline{\mathbf{X}}} &= \underset{\underline{\mathbf{X}}, \mathbf{A}, \mathbf{B}, \mathbf{C}}{\text{argmin}}\ \mathbf{L}_T(\underline{\mathbf{Y}}, \underline{\mathbf{\Delta}}; \underline{\mathbf{X}}) \\ \text{s.t. } \underline{\mathbf{X}} &= \mathbf{A} \circ \mathbf{B} \circ \mathbf{C}, \text{rank}(\underline{\mathbf{X}}) \le R \end{aligned} \tag{5}$$

2.3. A Separable Low-Rank Regularization

Problem (5) is NP-hard because of the rank constrain. For matrix case, low rank condition can be approximate be the nuclear norm $\|\mathbf{X}\|_*$. And it can be further transormed by solving the following problem [23].

$$\|\mathbf{X}\|_* = \min_{\mathbf{B}, \mathbf{C}} \frac{1}{2}(\|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2) \ \text{ s.to } \ \mathbf{X} = \mathbf{B}\mathbf{C}^T \tag{6}$$

Bollowing the idea from [23], we expect to introduce a Frobenius norm regularization term to approximate the low rank constrain in (5).

$$h(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \frac{1}{2}(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2) \tag{7}$$

Then the loss function becomes

$$\begin{aligned} & G_T(\mathbf{A}, \mathbf{B}, \mathbf{C}; \underline{\mathbf{Y}}, \underline{\mathbf{\Delta}}) \\ & = \mathbf{L}_T(\underline{\mathbf{Y}}, \underline{\mathbf{\Delta}}; \underline{\mathbf{X}}) + \lambda h(\mathbf{A}, \mathbf{B}, \mathbf{C}) \\ & = \frac{1}{T}\sum_{\tau=1}^{T}\sum_{mn} \delta_{mn\tau}(\sum_{r=1}^{R} a_{mr}b_{nr}c_{\tau r} - y_{mn\tau}\log\sum_{r'=1}^{R} a_{mr}b_{nr}c_{\tau r}) \\ & + \frac{\lambda}{2}(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2) \end{aligned}$$
(8)

3. ONLINE TENSOR COMPLETION FOR POISSON DATA

The minimizer of (8) can be solved by the LRPTI algorithm in [?]. However, it requires us to have assess to all of the data steam, which hinders applicability to memory limited case. To begin with, let's rewrite the rank-regularized empirical loss function in (8)

$$\begin{aligned} & G_T(\mathbf{A}, \mathbf{B}, \mathbf{C}; \underline{\mathbf{Y}}, \underline{\mathbf{\Delta}}) := \mathbf{L}_T(\underline{\mathbf{Y}}, \underline{\mathbf{\Delta}}; \underline{\mathbf{X}}) + \lambda h(\mathbf{A}, \mathbf{B}, \mathbf{C}) \\ & = \frac{1}{T}\sum_{\tau=1}^{T}(\mathbf{l}_\tau(\mathbf{Y}_\tau, \mathbf{\Delta}_\tau; \mathbf{X}_\tau) + \frac{1}{2\mu}(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2) + \frac{\mu}{2}\|\mathbf{c}_\tau\|_2^2) \\ & := \frac{1}{T}\sum_{\tau=1}^{T} g_\tau(\mathbf{A}, \mathbf{B}, \mathbf{c}_\tau) \end{aligned}$$
(9)

Where $\mu = \lambda T$. Then problem (8) becomes

$$\min_{\mathbf{A}, \mathbf{B}} C_T(\mathbf{A}, \mathbf{B}) := \frac{1}{T}\sum_{\tau=1}^{T} \bar{g}_\tau(\mathbf{A}, \mathbf{B}) \tag{10}$$

Where $\bar{g}_\tau(\mathbf{A}, \mathbf{B}) := \min_{\mathbf{c}_\tau} g_\tau(\mathbf{A}, \mathbf{B}, \mathbf{c}_\tau)$. However, solving (10) is complex, so let's solve the approximate problem

$$\min_{\mathbf{A}, \mathbf{B}} \bar{C}_T(\mathbf{A}, \mathbf{B}) := \frac{1}{T}\sum_{\tau=1}^{T} g_\tau(\mathbf{A}, \mathbf{B}, \hat{\mathbf{c}}_\tau) \tag{11}$$

Where $\hat{\mathbf{c}}_\tau$ is the estimation of $\mathbf{c}_\tau$ based on the current data $\{\mathbf{Y}_\tau, \mathbf{\Delta}_\tau\}$ and the factor matrices obtained at previous step $\{\mathbf{A}[\tau-1], \mathbf{B}[\tau-1]\}$.

$$\hat{\mathbf{c}}_\tau = \underset{\mathbf{c}_\tau}{\text{argmin}}\ g_\tau(\mathbf{A}[\tau-1], \mathbf{B}[\tau-1], \mathbf{c}_\tau) \tag{12}$$

Problem (12) can be solved efficiently because when $\{\mathbf{Y}_\tau, \mathbf{\Delta}_\tau\}$ and $\{\mathbf{A}[\tau-1], \mathbf{B}[\tau-1]\}$ are given, it becomes a convex problem with respect to $\mathbf{c}_\tau\tau$. In order to solve (11), let's look at an quadratic upperbound of $g_\tau(\mathbf{A}[\tau-1], \mathbf{B}[\tau-1], \mathbf{c})$.

$$\begin{aligned} & \tilde{g}_\tau(\mathbf{A}, \mathbf{B}; \mathbf{A}[\tau-1], \mathbf{B}[\tau-1], \mathbf{c}_\tau) := g_\tau(\mathbf{A}[\tau-1], \mathbf{B}[\tau-1], \mathbf{c}_\tau) \\ & + \frac{1}{2\alpha_\tau}(\|\mathbf{A} - \mathbf{A}[\tau-1]\|_F^2 + \|\mathbf{B} - \mathbf{B}[\tau-1]\|_F^2) \\ & + \langle \nabla_{\mathbf{A}} g_\tau(\mathbf{A}, \mathbf{B}[\tau-1], \mathbf{c}_\tau)|_{\mathbf{A}=\mathbf{A}[\tau-1]}, \mathbf{A} - \mathbf{A}[\tau-1]\rangle \\ & + \langle \nabla_{\mathbf{B}} g_\tau(\mathbf{A}[\tau-1], \mathbf{B}], \mathbf{c}_\tau)|_{\mathbf{B}=\mathbf{B}[\tau-1]}, \mathbf{B} - \mathbf{B}[\tau-1]\rangle \end{aligned}$$
(13)

With $\alpha_\tau^{-1} \ge \max(\|\nabla_{\mathbf{A}}^2 g_\tau(\mathbf{A}, \mathbf{B}[\tau-1], \hat{\mathbf{c}}_\tau)|_{\mathbf{A}=\mathbf{A}[\tau-1]}\|, \|\nabla_{\mathbf{B}}^2 g_\tau(\mathbf{A}[\tau-1], \mathbf{B}, \hat{\mathbf{c}}_\tau)|_{\mathbf{B}=\mathbf{B}[\tau-1]}\|)$, (13) is a tight approximation for g_τ. Because one can verify

(1) $\tilde{g}_\tau(\mathbf{A}, \mathbf{B}; \mathbf{A}[\tau-1], \mathbf{B}[\tau-1], \mathbf{c}_\tau) \ge g_\tau(\mathbf{A}, \mathbf{B}, \mathbf{c}_\tau), \forall \mathbf{A}, \mathbf{B}$
(2) $\tilde{g}_\tau(\mathbf{A}[\tau-1], \mathbf{B}[\tau-1]; \mathbf{A}[\tau-1], \mathbf{B}[\tau-1], \mathbf{c}_\tau)$
$= g_\tau(\mathbf{A}[\tau-1], \mathbf{B}[\tau-1], \mathbf{c}_\tau)$
(3a) $\nabla_{\mathbf{A}}\tilde{g}_t(\mathbf{A}, \mathbf{B}; \mathbf{A}[\tau-1], \mathbf{B}[\tau-1], \mathbf{c}_\tau)|_{\mathbf{A}=\mathbf{A}[\tau-1], \mathbf{B}=\mathbf{B}[\tau-1]}$
$= \nabla_{\mathbf{A}} g_\tau(\mathbf{A}, \mathbf{B}, \mathbf{c}_\tau)|_{\mathbf{A}=\mathbf{A}[\tau-1], \mathbf{B}=\mathbf{B}[\tau-1]}$
(3b) $\nabla_{\mathbf{B}}\tilde{g}_t(\mathbf{A}, \mathbf{B}; \mathbf{A}[\tau-1], \mathbf{B}[\tau-1], \mathbf{c}_\tau)|_{\mathbf{A}=\mathbf{A}[\tau-1], \mathbf{B}=\mathbf{B}[\tau-1]}$
$= \nabla_{\mathbf{B}} g_\tau(\mathbf{A}, \mathbf{B}, \mathbf{c}_\tau)|_{\mathbf{A}=\mathbf{A}[\tau-1], \mathbf{B}=\mathbf{B}[\tau-1]}$
(14)

We intend to solve

$$\min_{\mathbf{A}, \mathbf{B}} \tilde{C}_T(\mathbf{A}, \mathbf{B}) := \frac{1}{T}\sum_{\tau=1}^{T} \tilde{g}_\tau(\mathbf{A}, \mathbf{B}; \mathbf{A}[\tau-1], \mathbf{B}[\tau-1], \hat{\mathbf{c}}_\tau)$$
(15)

The optimizer for (15) is

$$\begin{bmatrix} \mathbf{A}[\tau] \\ \mathbf{B}[\tau] \end{bmatrix} = \begin{bmatrix} \mathbf{A}[\tau-1] \\ \mathbf{B}[\tau-1] \end{bmatrix} - \alpha_\tau \begin{bmatrix} \nabla_{\mathbf{A}} g_\tau(\mathbf{A}, \mathbf{B}[\tau-1], \hat{\mathbf{c}}_\tau) \\ \nabla_{\mathbf{B}} g_\tau(\mathbf{A}[\tau-1], \mathbf{B}, \hat{\mathbf{c}}_\tau) \end{bmatrix} \tag{16}$$

Algorithm 1 Online SGD for Poisson tensor decomposition and imputation

Input: Tensor $\{(\mathbf{Y}_\tau \circledast \mathbf{\Delta}_\tau, \mathbf{\Delta}_\tau)\}_{\tau=1,..,T}$, $\mu > 0$, $0 < \beta \leq 1$ and $\gamma > 0$, rank R

Initialize $\tau = 1$. $\mathbf{A}[0], \mathbf{B}[0]$ at random.

repeat

  Solve

$$\hat{\mathbf{c}}_\tau = \underset{\mathbf{c}_\tau}{\operatorname{argmin}}\, g_\tau(\mathbf{A}[\tau-1], \mathbf{B}[\tau-1], \mathbf{c}_\tau)$$

  $\alpha_\tau \leftarrow \gamma$

  repeat

    Update

$$\mathbf{A}[\tau] = \max(\mathbf{A}[\tau-1] - \alpha_\tau \nabla_{\mathbf{A}} g_\tau(\mathbf{A}, \mathbf{B}[\tau-1], \hat{\mathbf{c}}_\tau), \mathbf{0}_{M\times R})$$
$$\mathbf{B}[\tau] = \max(\mathbf{B}[\tau-1] - \alpha_\tau \nabla_{\mathbf{B}} g_\tau(\mathbf{A}[\tau-1], \mathbf{B}, \hat{\mathbf{c}}_\tau), \mathbf{0}_{M\times R})$$
$$\alpha_\tau \leftarrow \alpha_\tau \beta$$

  until $g_t(\mathbf{A}[\tau], \mathbf{B}[\tau], \hat{\mathbf{c}}_\tau) \leq g_t(\mathbf{A}[\tau-1], \mathbf{B}[\tau-1], \hat{\mathbf{c}}_\tau)$

  $\tau \leftarrow \tau + 1$.

until $\tau > T$

return $\underline{\hat{\mathbf{X}}} = \mathbf{A} \circ \mathbf{B} \circ \mathbf{C}$.

Where α_τ can be considered as a proper stepsize which guarantees a true descent on the $g_\tau(\mathbf{A}, \mathbf{B}, \hat{\mathbf{c}}_\tau)$. In the proposed algorithm 1, we apply the line search approach to find $\{\alpha_\tau\}$.

4. NUMERICAL TESTS

The performance and convergence of the proposed algorithm 1 is assessed in this section via computer simulation on synthetic data and the SDO data set.

4.1. Synthetic tensor data tests

The input signal $\underline{\mathbf{X}}$ of dimension $M \times N \times T = 20 \times 15 \times 400$ is generated i.i.d, using the low-rank approximation (1), where the entries of $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$, with rank $R = 10$, are drawn from a uniform distribution and scaled yielding to $\mathbb{E}[x_{mnt}] = 100]$. Also, we assume the observation mask $\underline{\mathbf{\Delta}}$ is generated i.i.d. via the Bernoulli distribution with the parameter $p \in [0, 1]$. Relative error $\{e_t^{(re)}\}$ and normalized subspace reconstruction error (NSRE) $\{e_t^{(ss)}\}$ are adopted to measure performance. They are computed as $e_t^{(re)} = \|\hat{\mathbf{X}}_t - \mathbf{X}_t\|_F / \|\mathbf{X}_t\|_F$, where $\hat{\mathbf{X}}_t = (\hat{\mathbf{B}}[t] \odot \hat{\mathbf{A}}[t])\mathbf{c}_t^T$ is the estimation of $\mathbf{X}_t$; and $e_t^{(ss)} = \|P_{\hat{\mathbf{\Pi}}_{Ct}^\perp} \mathbf{\Pi}_C\|_F / \|\mathbf{\Pi}_C\|_F$, where $P_{\hat{\mathbf{\Pi}}_{Ct}^\perp} = (\mathbb{I} - \hat{\mathbf{\Pi}}_{Ct}\hat{\mathbf{\Pi}}_{Ct}^\dagger)$ is the projection operator onto the orthogonal complement of estimation of the subspace (with respect to $\{\mathbf{c}_t\}$), $\hat{\mathbf{\Pi}}_{Ct} = (\hat{\mathbf{B}}[t] \odot \hat{\mathbf{A}}[t])$. And $\mathbf{\Pi}_C = (\mathbf{B} \odot \mathbf{A})$ is the true subspace. We compare our result with the batch algorithm LRPTI in [10]. LRPTI is not an online algorithm, since it is assessd using all of the data steam. As a result, the performance of LRPTI will later be considered as the baseline. The relative error for LRPTI is computed as $\{\|\hat{\mathbf{A}}\text{diag}(\mathbf{c}_t)\hat{\mathbf{B}} - \mathbf{X}_t\|_F / \|\mathbf{X}_t\|_F\}_{t=1,\ldots,T}$, and the normalized subspace reconstrucion error will be only a number $\hat{\mathbf{\Pi}}_C = (\hat{\mathbf{B}} \odot \hat{\mathbf{A}})$, depicted by the yellow line in Fig.1 (b).

Fig.**??** depicts the relative error with respect to data steam index T for the proposed algorithm (both fully and 50% observed) and batch alogrithm LRPTI when the data is fully observed. The optimality of our approach is demonstrated by showing the conver-

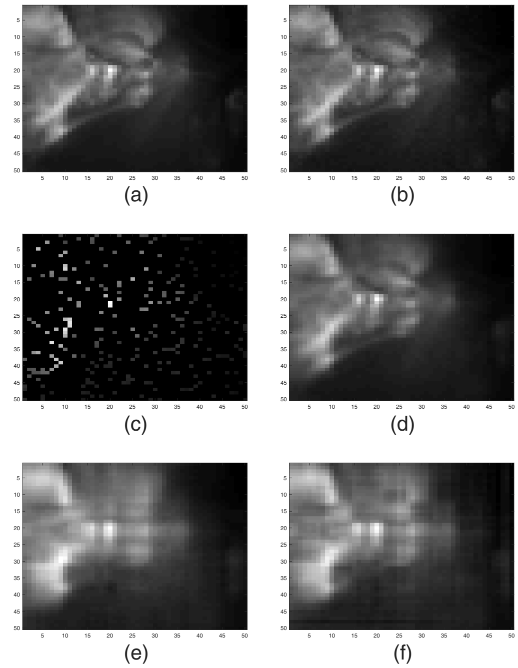

Fig. 2: Solar flare video at $t = 200$. (a) True video, (b) video corrupted by Poisson noise, (c) partial observation of (b) with $p = 0.1$, (d) reconstructed video via batch algorithm for full Poisson corrupted observation (b), (e) reconstructed video via proposed algorithm for full Poisson corrupted observation (b), (f) reconstructed video via proposed algorithm for patial observation (c). For (d)(e)(f), we set the rank for testing $R = 20$. (f) depicts that only with a small sample of the video, the proposed algorithm will still be able to reconstruct most of the details of the original video.

gence of the relative error to the baseline. Even with a 50% observed data steam, the relative error for our approach still converges to the baseline fast. Fig. **??** shows the normalized subspace reconstruction error for the proposed algorithm with respect to data steam index T, and the batch algorithm. It can be seen that the normalized subspace reconstruction error for our approach also converges to the baseline, which implies that our approach can also achieve the optimal subspace.

4.2. Real tensor data

We also apply the proposed algorithm on a Poisson currupted version of a real solar flare video captured by the NASA SDO satellite (see [22] for detailed information). The video is of resolution 50×50 with 300 frames. We let the rank for testing $R = 20$, and model parameter $\mu = 0.01$. Fig.**??** depicts the relative error for both of our approach and the batch algorithm. Unlike the symthetic data in section 4.1, we can not compute the normalized subspace reconstruction error since there is no ground truth factor matrices $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ for the real video. In Fig.2, we illustrate the original solar flare video, the Poisson corrupted version, the and their reconstruction via our approach and the batch algorithm. Fig.2f shows that few samples are sufficient for the reconstruction using the proposed algorithm.

5. CONCLUSIONS

In this paper, we have studied online tensor decomposition and imputation problem when the steaming data are Poisson random counts. We consider a maximum likelihood formulation regularized with a separable surrogate of the nuclear norm. Applying stochastic gradient descent on an approximated loss function, an efficient

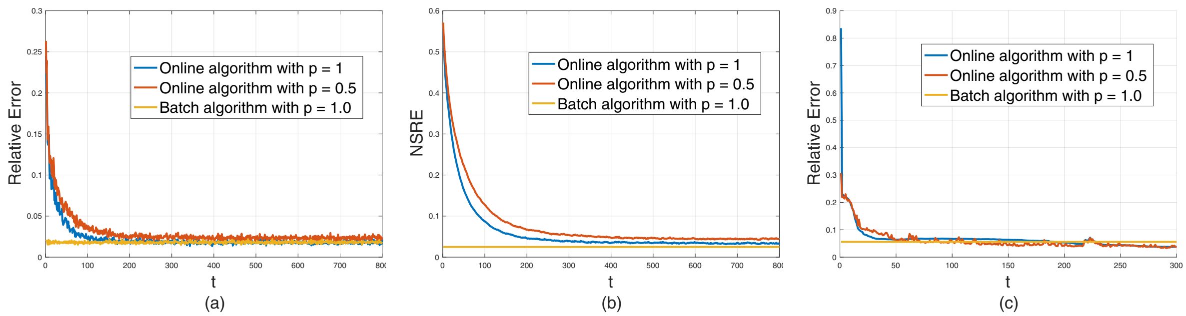

Fig. 1: (a) Relative error for symthetic data with 10 realizations and $\mu = 0.01$; (b) Normalized subspace reconstruction error for symthetic data with 10 realizations and $\mu = 0.01$; (c) Relative error for SDO dataset, with $\mu = 0.01$. In all of those figures, the blue and red curves converge to the base line yellow curve after only a few time steps, e.g. $t < 150$, which indicates the convergence and optimality of the proposed approach.

online algorithm was developed. With the capability to memory-limited implementation, the proposed online approach offered an attractive alternative to batch algorithm. Convergency and optimality of the proposed algorithm was demonstrated numerically via tests on symthetic data and a real data example of solar flare video. The proof of the convergency and optimality for the proposed algorithm will be persuing as future research.

6. REFERENCES

[1] K. Slavakis, G. B. Giannakis, and G. Mateos, "Modeling and optimization for big data analytics: (Statistical) learning tools for our era of data deluge," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 18–31, 2014.

[2] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, 2017.

[3] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. PHAN, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, 2015.

[4] D. Nion and N. D. Sidiropoulos, "Adaptive algorithms to track the PARAFAC decomposition of a third-order tensor," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2299–2310, June 2009.

[5] M. Mardani, G. Mateos, and G. B. Giannakis, "Subspace learning and imputation for streaming big data matrices and tensors," *IEEE Trans. Signal Process.*, vol. 63, no. 10, pp. 2663–2677, 2015.

[6] J. Sun, D. Tao, and C. Faloutsos, "Beyond streams and graphs: Dynamic tensor analysis," in *Proc. of the 12th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, Philadelphia, USA, Aug. 2006, pp. 374–383.

[7] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mrup, "Scalable tensor factorizations for incomplete data," *Chemom. Intell. Lab. Syst.*, vol. 106, no. 1, pp. 41–56, Mar. 2011.

[8] L. Balzano, Y. Chi, and Y. M. Lu, "Streaming PCA and subspace tracking: The missing data case," *Proc. IEEE*, vol. 106, no. 8, pp. 1293–1310, 2018.

[9] E. J. Candès and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, pp. 925–936, June 2009.

[10] J. A. Bazerque, G. Mateos, and G. B. Giannakis, "Rank regularization and Bayesian inference for tensor completion and extrapolation," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5689–5703, Nov. 2013.

[11] L. Wang and Y. Chi, "Stochastic approximation and memory-limited subspace tracking for poisson streaming data," *IEEE Trans. Signal Process.*, vol. 66, no. 4, pp. 1051–1064, 2018.

[12] E. C. Chi and T. G. Kolda, "On tensors, sparsity, and nonnegative factorizations," *SIAM J. Matrix Anal. Appl.*, vol. 33, no. 4, pp. 12721299, Dec. 2012.

[13] Y. Cao and Y. Xie, "Poisson matrix recovery and completion," *IEEE Trans. Signal Process.*, vol. 64, no. 6, pp. 1609–1620, 2016.

[14] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, pp. 95–107, Jan. 1995.

[15] Y. Chi, Y. C. Eldar, and R. Calderbank, "PETRELS: Parallel subspace estimation and tracking by recursive least squares from partial observations," *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5947–5959, Dec. 2013.

[16] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Providence, Rhode Island, June 2012.

[17] C. Qiu, N. Vaswani, B. Lois, and L. Hogben, "Recursive robust pca or recursive sparse recovery in large but structured noise," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 5007–5039, Aug. 2014.

[18] M. Mardani, G. Mateos, and G. B. Giannakis, "Dynamic anomalography: tracking network anomalies via sparsity and low rank," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 11, pp. 50–66, Feb. 2013.

[19] J. Feng, H. Xu, and S. Yan, "Online robust PCA via stochastic optimization," in *Proc. Advances in Neural Information Processing Systems*, Lake Tahoe, NV, Dec. 2013.

[20] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Jan. 2010.

[21] Y. Shen, M. Mardani, and G. B. Giannakis, "Online categorical subspace learning for sketching big data with misses," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4004–4018, 2017.

[22] Y. Xie, J. Huang, and R. Willett, "Change-point detection for high-dimensional time series with missing data," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 1, pp. 12–27, 2013.

[23] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Proc. of Advances in Neural Information Processing Systems*, 2005, pp. 1329–1336.