Imputation of Streaming Low-Rank Tensor Data

Morteza Mardani, Gonzalo Mateos, and Georgios B. Giannakis
Dept. of ECE and Digital Technology Center
University of Minnesota
Emails: {morteza,mate0058,georgios}@umn.edu

***Abstract*—Unraveling latent structure by means of multilinear models of *tensor data* is of paramount importance in timely inference tasks encountered with 'Big Data' analytics. However, increasingly noisy, heterogeneous, and incomplete datasets as well as the need for real-time processing of streaming data pose major challenges to this end. The present paper introduces a novel *online* (adaptive) algorithm to decompose low-rank tensors with missing entries, and perform imputation as a byproduct. The novel estimator minimizes an exponentially-weighted least-squares fitting error 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, real-time algorithm is developed and its convergence is established under simplifying technical assumptions. Simulated tests with cardiac magnetic resonance imagery (MRI) data confirm the efficacy of the proposed algorithm in imputing up to 75% missing entries.**

I. Introduction

As massive amounts of data become increasingly complex and heterogeneous, in many situations one encounters datasets indexed by three or more variables giving rise to a *tensor*, that is a data cube or a mutli-way array, in general. It is not uncommon that one of these variables indexes time [11], and that sizable portions of the data are missing [2], [4], [6]. Examples of time-indexed, incomplete tensor data include: (i) dynamic social networks represented through a temporal sequence of adjacency matrices, while it may be the case that not all pairwise interactions among nodes can be sampled; and (ii) multidimensional nuclear magnetic resonance (NMR) analysis, where missing data are encountered when sparse sampling is used in order to reduce the experimental time.

Various data analytics tasks aim at capturing the underlying latent structure, which calls for high-order tensor factorizations even in the presence of missing data [2], [4]. It is in principle possible to unfold the given tensor into a matrix and resort to either batch [6], [12], or online matrix completion algorithms [9]. However, tensor models preserve the multi-way nature of the data and extract the underlying factors in each mode (dimension) of a higher-order array. Accordingly, the desiderata for analyzing streaming and incomplete multi-way data are *low-complexity, real-time algorithms* capable of unraveling latent structures through parsimonious (e.g., low-rank) decompositions, such as the parallel factor analysis (PARAFAC) model; see e.g. [7] for a tutorial treatment on tensor decompositions and applications.

With this objective in mind, the present paper develops for the first time an *online* (adaptive) algorithm for decomposing *low-rank tensors with missing entries*; see also [11] for an adaptive algorithm to obtain PARAFAC decompositions with full data. Accurately approximating a given incomplete tensor allows one to *impute* those missing entries as a byproduct, by simply reconstructing the data cube from the model factors (which for PARAFAC are unique under relatively mild assumptions [3], [8]). The novel estimator relies on an exponentially-weighted least-squares fitting error criterion regularized by separable PARAFAC decomposition factors [4], to trade-off the fidelity of the approximation for complexity captured by the decomposition's rank (Section II). Leveraging stochastic gradient descent iterations, a scalable, real-time algorithm is developed in Section III, and its convergence is established under simplifying technical assumptions. The proposed online algorithm offers a viable approach to solving large-scale tensor decomposition (and completion) problems, even if the data is not actually streamed but they are so massive that do not fit in main memory. Simulated tests with cardiac magnetic resonance imagery (MRI) data confirm the efficacy of the proposed algorithm in imputing up to 75% missing entries (Section IV).

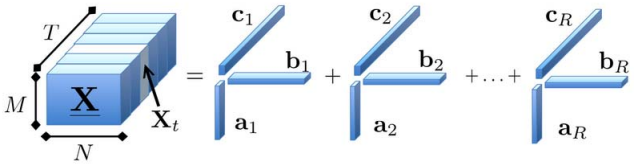

Fig. 1: A rank-R PARAFAC decomposition of the three-way tensor $\underline{\mathbf{X}}$.

Notation: Operators $()'$, $\circ$, $\odot$, and $\sigma_{\max}(\cdot)$ denote transposition, outer product, Hadamard product, and maximum singular value, respectively; $\mathrm{diag}(\mathbf{x})$ is a diagonal matrix with diagonal entries $\mathbf{x}$; $|\cdot|$ is the magnitude of a scalar, $\|\cdot\|_2$ the ℓ_2-norm of a vector; and $\|\cdot\|_F$ the Frobenius norm of a matrix.

II. Preliminaries and Problem Statement

For three vectors $\mathbf{a} \in \mathbb{R}^{M\times 1}$, $\mathbf{b} \in \mathbb{R}^{N\times 1}$, and $\mathbf{c} \in \mathbb{R}^{T\times 1}$, the outer product $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$ is an $M \times N \times T$ rank-one three-way array with (m, n, t)-th entry given by [1]$\mathbf{a}(m)\mathbf{b}(n)\mathbf{c}(t)$. Note that this comprises a generalization to the two vector (matrix) case, where $\mathbf{a} \circ \mathbf{b} := \mathbf{a}\mathbf{b}'$ is a rank-one matrix. The rank of a

[1]The focus here is on three-way tensors for simplicity in exposition, but extensions to higher-way arrays are possible.

978-1-4799-1481-4/14/$31.00 ©2014 IEEE

tensor $\underline{\mathbf{X}}$ is defined as the minimum number of outer products required to synthesize $\underline{\mathbf{X}}$.

The PARAFAC model, also known as canonical decomposition (CANDECOMP) or CP (CANDECOMP/PARAFAC), is arguably the most basic tensor model because of its direct relationship to tensor rank. Based on the previous discussion it is natural to form a *low-rank approximation* of tensor $\underline{\mathbf{X}} \in \mathbb{R}^{M\times N\times T}$ as

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

When the decomposition is exact, (1) is the PARAFAC decomposition of $\underline{\mathbf{X}}$; see also Fig. 1. Accordingly, the minimum value R for which the exact decomposition is possible is (by definition) the rank of $\underline{\mathbf{X}}$. PARAFAC is the model of choice when one is primarily interested in revealing latent structure. Considering the analysis of a dynamic social network for instance, each of the rank-one factors in Fig. 1 could correspond to communities that e.g., persist or form and dissolve periodically across time. Different from the matrix case, there is no straightforward algorithm to determine the rank of a given tensor, a problem that has been shown to be NP-hard [7]. For a survey of algorithmic approaches to obtain approximate PARAFAC decompositions, the reader is referred to [7].

With reference to (1), introduce the factor matrix $\mathbf{A} := [\mathbf{a}_1, \ldots, \mathbf{a}_R] \in \mathbb{R}^{M\times R}$, and likewise for $\mathbf{B} \in \mathbb{R}^{N\times R}$ and $\mathbf{C} \in \mathbb{R}^{T\times R}$. Let $\mathbf{X}_t,\ t = 1, \ldots, T$ denote the t-th slice of $\underline{\mathbf{X}}$ along its third (tube) dimension, such that $\mathbf{X}_t(m, n) = \underline{\mathbf{X}}(m, n, t)$; see also Fig. 1. The following compact matrix form of the PARAFAC decomposition in terms of slice factorizations will be used in the sequel

$$\mathbf{X}_t = \mathbf{A}\text{diag}[\boldsymbol{\gamma}_t]\mathbf{B}' = \sum_{r=1}^{R} \boldsymbol{\gamma}_t(r)\mathbf{a}_r\mathbf{b}_r', \quad t = 1, \ldots, T \tag{2}$$

where $\boldsymbol{\gamma}_t'$ denotes the t-th row of $\mathbf{C}$ (recall that $\mathbf{c}_r$ instead denotes the r-th column of $\mathbf{C}$). It is apparent that each slice $\mathbf{X}_t$ can be represented as a linear combination of R rank-one matrices $\{\mathbf{a}_r\mathbf{b}_r'\}_{r=1}^R$, which constitute the bases for the tensor fiber subspace. The PARAFAC decomposition is symmetric [cf. (1)], and one can likewise write $\mathbf{X}_m = \mathbf{B}\text{diag}[\boldsymbol{\alpha}_m]\mathbf{C}'$, or, $\mathbf{X}_n = \mathbf{C}\text{diag}[\boldsymbol{\beta}_n]\mathbf{A}'$ in terms of slices along the first (row), or, second (column) dimensions – once more, $\boldsymbol{\alpha}_m'$ stands for the m-th row of $\mathbf{A}$, and likewise for $\boldsymbol{\beta}_n'$.

Given $\underline{\mathbf{X}}$, under some technical conditions, $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ are unique up to a common column permutation and scaling (meaning PARAFAC is identifiable); see e.g. [3], [8].

A. Batch Low-Rank Tensor Decomposition and Imputation

Let $\underline{\mathbf{Y}} \in \mathbb{R}^{M\times N\times T}$ be a three-way tensor, and likewise let $\underline{\boldsymbol{\Omega}}$ denote a $M \times N \times T$ binary $\{0, 1\}$-tensor with (m, n, t)-th entry equal to 1 if $\underline{\mathbf{Y}}(m, n, t)$ is observed, and 0 otherwise. One can thus represent the incomplete data tensor compactly as $\mathcal{P}_{\underline{\Omega}}(\underline{\mathbf{Y}}) = \underline{\boldsymbol{\Omega}} \odot \underline{\mathbf{Y}}$; see also Fig. 2 (left). The objective is, given the data $\{\mathcal{P}_{\underline{\Omega}}(\underline{\mathbf{Y}}), \underline{\boldsymbol{\Omega}}\}$, impute the missing entries, and possibly denoise the available entries. Building on the intuition

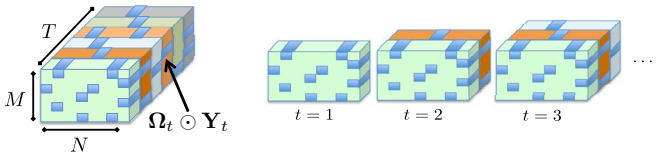

Fig. 2: Tensor data with missing entries. (Left) Batch data, and slice $\boldsymbol{\Omega}_t \odot \mathbf{Y}_t$ along the time (tube) dimension. (Right) Streaming data, where slices $\boldsymbol{\Omega}_t \odot \mathbf{Y}_t$ become available for $t = 1, 2, \ldots$.

for the matrix case, feasibility of the imputation task relies fundamentally on assuming a low-dimensional PARAFAC model for the data, to couple the available and missing entries as well as reduce the effective degrees of freedom in the problem. Under the low-rank assumption for instance, a rough idea on the fraction p_m of missing data that can be afforded is obtained by comparing the number of unknowns $R(M+N+T)$ in (1) with the number of available data samples $(1 - p_m)MNT$. Ensuring that $(1 - p_m)MNT \geq R(M + N + T)$, roughly implies that the tensor can be potentially recovered even if a fraction $p_m \leq 1 - R(M + N + T)/(MNT)$ of entries is missing. Different low-dimensional tensor models would lead to alternative imputation methods, such as the unfolded tensor regularization in [6] for *batch* tensor completion.

For matrices, nuclear-norm regularization $\|\mathbf{X}\|_* := \sum_i \sigma_i(\mathbf{X})$, with $\sigma_i(\mathbf{X})$ denoting the i-th singular value, is known as a convex surrogate for the rank [12]. However, generalizing the nuclear-norm regularization technique from low-rank matrices to tensor completion is not straightforward if one also desires to unveil the latent structure in the data. The notion of singular values of a tensor (given by the Tucker3 decomposition) are not related to the rank [7]. Interestingly, it was argued in [4] that the Frobenius-norm regularization of the tensor factors, namely

$$h(\mathbf{A}, \mathbf{B}, \mathbf{C}) := \|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2 \tag{3}$$

offers a viable option for *batch* low-rank tensor completion under the PARAFAC model, by solving

$$\text{(P1)} \quad \min_{\{\mathbf{X},\mathbf{A},\mathbf{B},\mathbf{C}\}} \frac{1}{2}\|\underline{\boldsymbol{\Omega}} \odot (\underline{\mathbf{Y}} - \underline{\mathbf{X}})\|_F^2 + \frac{\lambda}{2}\, h(\mathbf{A}, \mathbf{B}, \mathbf{C})$$
$$\text{s.to} \quad \mathbf{X}_t = \mathbf{A}\text{diag}[\boldsymbol{\gamma}_t]\mathbf{B}', \quad t = 1, 2 \ldots, T.$$

The regularizer (3) encourages low-rank tensor decompositions, with controllable rank by tuning the parameter λ [4].

Different solvers can be developed to solve the batch problem (P1) for reasonable size datasets. However, when it comes to streaming high-dimensional observations with huge slices of size MN acquired over the time dimension, where the time horizon T grows, scalable algorithms need to be devised, which overcome the growing problem size. This is dealt with in the next section, which offers an approach for decomposing and imputing low-rank *streaming* tensors.

III. Online Tensor Decomposition and Imputation

Consider now a real-time setting where the incomplete tensor slices $\mathcal{P}_{\Omega_t}(\mathbf{Y}_t) = \boldsymbol{\Omega}_t \odot \mathbf{Y}_t$ are acquired sequentially

over time $t = 1, 2, \ldots$; i.e., streaming data as depicted in Fig. 2 (right). Leveraging the batch formulation (P1) one can naturally broaden the subspace tracking framework introduced in [9], to devise adaptive algorithms capable of factorizing tensors 'on the fly'. The basic idea is to estimate the PARAFAC model factors $\{\mathbf{A}[t], \mathbf{B}[t], \mathbf{C}[t]\}$ as the minimizers of the following exponentially-weighted least-squares cost

$$\text{(P2)} \quad \min_{\{\mathbf{A},\mathbf{B},\mathbf{C}\}} \frac{1}{2}\sum_{\tau=1}^{t} \mu^{t-\tau}\Big[\|\mathbf{\Omega}_\tau \odot (\mathbf{Y}_\tau - \mathbf{A}\text{diag}(\boldsymbol{\gamma}_\tau)\mathbf{B}')\|_F^2 + \bar{\lambda}_t(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2) + \lambda_t\|\boldsymbol{\gamma}_\tau\|^2\Big].$$

where the forgetting factor $\mu \in (0, 1)$ downweighs the data in the distant past to facilitate tracking in nonstationary environments. The normalization $\bar{\lambda} := \lambda_t / \sum_{\tau=1}^{t} \mu^{t-\tau}$ also ensures that for the infinite memory setting ($\mu = 1$) and $t = T$, (P2) coincides with the batch estimator (P1).

A. Online alternating minimization algorithm

Towards deriving a real-time, computationally efficient, and recursive solver of (P2), an alternating minimization technique is adopted in which iterations coincide with the time-scale t of data acquisition. A justification in terms of minimizing a suitable approximate cost function is discussed later. Per time t, given the previously updated $\{\mathbf{A}[t-1], \mathbf{B}[t-1]\}$, a new datum $\{\mathbf{\Omega}_t, \mathcal{P}_{\Omega_t}(\mathbf{Y}_t)\}$ is drawn and $\boldsymbol{\gamma}[t]$ is estimated as $\boldsymbol{\gamma}[t] = \arg\min_{\boldsymbol{\gamma}} g_t(\mathbf{A}[t-1], \mathbf{B}[t-1], \boldsymbol{\gamma})$, where

$$g_t(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) := \frac{1}{2}\|\mathbf{\Omega}_t \odot (\mathbf{Y}_t - \mathbf{A}\text{diag}(\boldsymbol{\gamma})\mathbf{B}')\|_F^2 + \frac{\lambda_t}{2}\|\boldsymbol{\gamma}\|^2$$

The minimizer is readily obtained in closed form $\boldsymbol{\gamma}_t(\mathbf{A}[t-1], \mathbf{B}[t-1])$, where

$$\boldsymbol{\gamma}_t(\mathbf{A}, \mathbf{B}) = \big[\lambda\mathbf{I}_R + \sum_{(m,n)\in\Omega_t} (\boldsymbol{\alpha}_m \odot \boldsymbol{\beta}_n)(\boldsymbol{\alpha}_m \odot \boldsymbol{\beta}_n)'\big]^{-1} \times \sum_{(m,n)\in\Omega_t} \underline{\mathbf{Y}}_t(m,n)(\boldsymbol{\alpha}_m \odot \boldsymbol{\beta}_n). \quad (4)$$

Accordingly, given the projection coefficients $\{\boldsymbol{\gamma}[\tau]\}_{\tau=1}^{T}$, the factor matrices $\{\mathbf{A}, \mathbf{B}\}$ that can be interpreted as bases for the fiber subspace are the minimizers of the cost function

$$C_t(\mathbf{A}, \mathbf{B}) := \sum_{\tau=1}^{t} \mu^{t-\tau} g_\tau(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}[\tau]) + \frac{\bar{\lambda}_t}{2}(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2). \quad (5)$$

Note that $\boldsymbol{\gamma}[t] := \boldsymbol{\gamma}_t(\mathbf{A}[t-1], \mathbf{B}[t-1])$ as per (4), so minimizing $C_t(\mathbf{A}, \mathbf{B})$ becomes increasingly complex computationally as t grows.

As discussed in [9], one can approximate $g_t(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}_t)$ with a tight upper bound $g_t(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}_t(\mathbf{A}[t-1], \mathbf{B}[t-1]))$ given the past updates $\{\mathbf{A}[t-1], \mathbf{B}[t-1]\}$, to develop a second-order algorithm that circumvents the aforementioned increasing complexity roadblock. Unlike the matrix case in [9] however, (5) is a nonconvex problem due to the bilinear nature of the PARAFAC decomposition (when, say, $\mathbf{C}$ is fixed); thus finding its global optimum efficiently may prove challenging.

Algorithm 1 : Online SGD algorithm for tensor decomposition and imputation

input $\{\mathbf{\Omega}_t \odot \mathbf{Y}_t, \mathbf{\Omega}_t\}_{t=1}^{\infty}$, R and $\{\lambda_t\}_{t=1}^{\infty}$.
initialize $\{\mathbf{A}[0], \mathbf{B}[0]\}$ at random, and $\tilde{\mu}[0] > 0$.
for $t = 0, 1, 2, \ldots$ **do**
$\mathbf{A}'[t] := [\boldsymbol{\alpha}_1[t], \ldots, \boldsymbol{\alpha}_M[t]]$ and $\mathbf{B}'[t] := [\boldsymbol{\beta}_1[t], \ldots, \boldsymbol{\beta}_N[t]]$
$\boldsymbol{\gamma}[t] = \Big[\lambda\mathbf{I}_R + \sum_{(m,n)\in\Omega_t}(\boldsymbol{\alpha}_m[t] \odot \boldsymbol{\beta}_n[t])(\boldsymbol{\alpha}_m[t] \odot \boldsymbol{\beta}_n[t])'\Big]^{-1} \times \sum_{(m,n)\in\Omega_t} \mathbf{Y}_t(m,n)(\boldsymbol{\alpha}_m[t] \odot \boldsymbol{\beta}_n[t])$
$\mathbf{P}[t] = [\mathbf{\Omega}_t \odot (\mathbf{Y}_t - \mathbf{A}[t]\text{diag}(\boldsymbol{\gamma}[t])\mathbf{B}'[t])]\mathbf{B}[t]\text{diag}(\boldsymbol{\gamma}[t])$
$\mathbf{A}[t+1] = \mathbf{A}[t] + \tilde{\mu}[t]\mathbf{P}[t] - \tilde{\mu}[t]\frac{\lambda_t}{t}\mathbf{A}[t]$
$\mathbf{Q}[t] = [\mathbf{\Omega}_t \odot (\mathbf{Y}_t - \mathbf{A}[t]\text{diag}(\boldsymbol{\gamma}[t])\mathbf{B}'[t])]'\mathbf{A}[t]\text{diag}(\boldsymbol{\gamma}[t])$
$\mathbf{B}[t+1] = \mathbf{B}[t] + \tilde{\mu}[t]\mathbf{Q}[t] - \tilde{\mu}[t]\frac{\lambda_t}{t}\mathbf{B}[t]$
end for
return $\hat{\mathbf{X}}[t] := \mathbf{A}[t]\text{diag}(\boldsymbol{\gamma}[t])\mathbf{B}'[t]$.

One could instead think of carrying out alternating minimizations with respect to each of the tree factors per time instant t, namely updating: (i) $\boldsymbol{\gamma}[t]$ first, given $\{\mathbf{A}[t-1], \mathbf{B}[t-1]\}$; (ii) then $\mathbf{B}[t]$ given $\mathbf{A}[t-1]$ and $\{\boldsymbol{\gamma}[\tau]\}_{\tau=1}^{t}$; and (iii) finally $\mathbf{A}[t]$ with fixed $\mathbf{B}[t]$ and $\{\boldsymbol{\gamma}[\tau]\}_{\tau=1}^{t}$. While each of these subtasks boils down to a convex optimization problem, the overall procedure does not necessarily lead to an efficient algorithm since one can show that updating $\mathbf{A}[t]$ and $\mathbf{B}[t]$ recursively is impossible.

Acknowledging the aforementioned challenges and the desire of computationally-efficient updates compatible with Big Data requirements, it is prudent to seek instead a (first-order) stochastic gradient descent (SGD) alternative. Let $f_t(\mathbf{A}, \mathbf{B}) := g_t(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}[t]) + \frac{\lambda_t}{2t}(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2)$ denote the t-th summand in (5), for $t = 1, 2, \ldots$ and $\mu = 1$. Matrices $\mathcal{L}[t] := \{\mathbf{A}[t], \mathbf{B}[t]\}$ are obtained via the SGD iteration

$$\begin{aligned}\mathcal{L}[t] &= \arg\min_{\mathcal{L}} Q_{(1/\tilde{\mu}[t]),t}(\mathcal{L}, \mathcal{L}[t-1]) \\ &= \mathcal{L}[t-1] - \tilde{\mu}[t]\nabla f_t(\mathcal{L}[t-1]) \end{aligned} \quad (6)$$

where $\tilde{\mu}[t] \geq \sigma_{\max}[\nabla^2 f_t(\mathcal{L}[t-1])]$ is a stepsize, and $Q_{\nu,t}(\mathcal{L}_1, \mathcal{L}_2) := f_t(\mathcal{L}_2) + \langle \mathcal{L}_1 - \mathcal{L}_2, \nabla_t(\mathcal{L}_2)\rangle + \frac{\nu}{2}\|\mathcal{L}_1 - \mathcal{L}_2\|_F^2$. It is instructive to recognize that the quadratic surrogate $Q_{\nu,t}$ has the following properties: (i) it majorizes $f_t(\mathbf{A}, \mathbf{B})$, namely $f_t(\mathcal{L}) \leq Q_{\nu,t}(\mathcal{L}, \mathcal{L}[t-1]),\ \forall \mathbf{A} \in \mathbb{R}^{M\times R}, \mathbf{B} \in \mathbb{R}^{N\times R}$; while it is locally tight meaning that (ii) $f_t(\mathcal{L}[t-1]) = Q_{\nu,t}(\mathcal{L}[t-1], \mathcal{L}[t-1])$, and (iii)$\nabla f_t(\mathcal{L}[t-1]) = \nabla Q_{\nu,t}(\mathcal{L}[t-1], \mathcal{L}[t-1])$. Accordingly, the minimizer of $Q_{(1/\tilde{\mu}[t]),t}(\mathcal{L}, \mathcal{L}[t-1])$ amounts to a correction along the negative gradient $\nabla f_t(\mathcal{L}[t-1])$, with stepsize $\tilde{\mu}[t]$ [cf. (6)].

Putting together (4) and (6), while observing that the components of $\nabla f_t(\mathcal{L})$ are expressible as

$$\nabla_{\mathbf{A}} f_t(\mathbf{A}, \mathbf{B}) = -[\mathbf{\Omega}_t \odot (\mathbf{Y}_t - \mathbf{A}\text{diag}(\boldsymbol{\gamma}[t])\mathbf{B}')]\mathbf{B}\text{diag}(\boldsymbol{\gamma}[t]) + \frac{\lambda_t}{t}\mathbf{A}, \quad (7)$$

$$\nabla_{\mathbf{B}} f_t(\mathbf{A}, \mathbf{B}) = -[\mathbf{\Omega}_t \odot (\mathbf{Y}_t - \mathbf{A}\text{diag}(\boldsymbol{\gamma}[t])\mathbf{B}')]'\mathbf{A}\text{diag}(\boldsymbol{\gamma}[t]) + \frac{\lambda_t}{t}\mathbf{B} \quad (8)$$

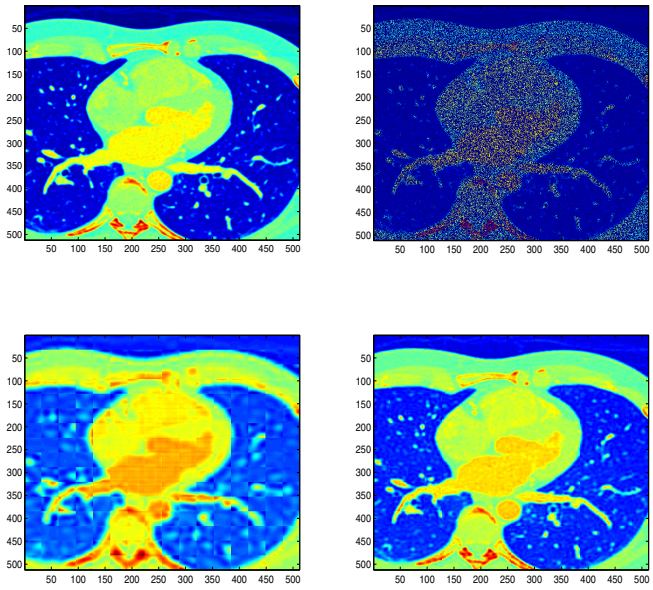

Fig. 3: Results of applying Algorithm 1 to cardiac MRI images. (top) Ground truth (left) and acquired image with only 25% available pixels (right). (bottom) Reconstructed image for rank $R = 10$ (left) and $R = 50$ (right).

one arrives at the SGD iterations tabulated under Algorithm 1.

Remark [Computational cost]: Updating $\mathbf{A}[t]$ and $\mathbf{B}[t]$ demands $\mathcal{O}(|\Omega_t|R)$ operations (R signifies rank), while updating $\boldsymbol{\gamma}[t]$ incurs a cost of $\mathcal{O}(|\Omega_t|R^2)$. The overall complexity per iteration is thus $\mathcal{O}(|\Omega_t|R^2)$.

Convergence of Algorithm 1 is formalized in the next proposition whose proof can be found in [10]. Furthermore, empirical observations in Section IV suggest that the convergence rate can be linear.

Proposition 1: *Suppose slices $\{\mathbf{Y}_t\}_{t=1}^{\infty}$ and the corresponding sampling sets $\{\Omega_t\}_{t=1}^{\infty}$ are i.i.d., and $\mu = 1$ while $\lambda_t = \lambda$, $\forall t$. If (c1) $\{\mathbf{A}[t], \mathbf{B}[t]\}_{t=1}^{\infty}$ live in a compact set, and (c2) $\sum_{\tau=1}^{t}(\tilde{\mu}[\tau])^{-1} \geq ct$ for some constant $c > 0$, then $\lim_{t\to\infty} \nabla C_t(\mathbf{A}[t], \mathbf{B}[t]) = \mathbf{0}$; i.e., iterates of Algorithm 1 converge to the stationary-point set of (P1) as $t \to \infty$ almost surely.*

Notice that for convergence, (c2) demands small step sizes. However, $\tilde{\mu}[t]$ cannot be too small since it must respect $\tilde{\mu}[t] \geq \sigma_{\max}[\nabla^2 f_t(\mathcal{L}[t-1])]$, $\forall t$, guaranteeing that $Q_{(1/\tilde{\mu}[t]),t}$ upper bounds f_t.

IV. Numerical Tests

The effectiveness and convergence of Algorithm 1 is assessed in this Section via computer simulations on real cardiac MRI data. Cardiac MRI is a major imaging modality for noninvasive diagnosis of heart diseases in clinic practice [5]. However, quality of MRI images is degraded as a result of fast acquisition process which is mainly due to patient's breath-holding time. The acquired image pixels thus can be considered as a subset of pixels of the high-resolution "ground truth" cardiac image. With this in mind, recovering the "ground truth" image amounts to imputing the missing pixels. This is well motivated by the low-intrinsic dimensionality of cardiac MRI images [5].

The FOURDIX dataset containing 263 cardiac scans with 10 steps of entire cardiac cycle is considered [1]. Each scan is an image of size 512×512 pixels, which is divided into 64 patches of 32×32 pixels. The 32×32 size patches then form slices of the tensor $\underline{\mathbf{Y}} \in \quad^{32\times 32\times 67328}$. A large percentage 75% of the entries of $\underline{\mathbf{Y}}$ is uniformly randomly dropped to simulate the missing entries, i.e., each entry of $\underline{\boldsymbol{\Omega}}$ is a binary $\{0, 1\}$-random variable which is one with probability 0.25, and zero otherwise. Clearly, imputing such a huge tensor by solving (P1) is infeasible as the tensor may not even fit the memory. The online Algorithm 1 is however a viable alternative, which performs $256R^2$ operations per iteration on average, and requires storing only about $256 + 64R$ scalar variables.

For a candidate cardiac image, the results of Algorithm 1 are depicted in Fig. 3 for various choices of the tensor rank, namely $R = 10, 50$. The constant step size $\tilde{\mu}[t] = 10^{-6}$, $\forall t$ is also picked along with $\lambda = 0.01$. For the reconstruction error $e_x := \|\hat{\mathbf{X}}_t - \mathbf{X}_t\|_F / \|\mathbf{X}_t\|_F$ with $\mathbf{X}_t$ and $\hat{\mathbf{X}}_t$ denoting the t-th true and estimated slices, respectively, different values of rank $R = 10, 50$ lead to $e_x = 0.14, 0.046$, respectively. Top figure shows the "ground truth" image on the left and the acquired one with only 25% available pixels on the right, where the missing entries are set to zero. Bottom figure also illustrates the reconstructed images after learning the tensor low-rank subspace for $R = 10$ (left) and $R = 50$ (right).

References

[1] http://http://www.osirix-viewer.com/datasets.

[2] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mrup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems,* vol. 106, no. 1, pp. 41-56, 2011.

[3] J. M. F. T. Berge and N. D. Sidiropoulos, "On uniqueness in CANDECOMP/PARAFAC," *Psychometrika,* vol. 67, no. 3, pp. 399-409, 2002.

[4] 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, pp. 5689–5703, Nov. 2013.

[5] H. Gao, "Prior rank, intensity and sparsity model (PRISM): A divide-and-conquer matrix decomposition model with low-rank coherence and sparse variation," *SPIE Opt. Eng. Appl,* 2012.

[6] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems,* vol. 27, no. 2, pp. 1-19, 2011.

[7] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455-500, Sep. 2009.

[8] J. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Lin. Alg. Applicat.*, vol. 18, no. 2, pp. 95-138, 1977.

[9] M. Mardani, G. Mateos, and G. B. Giannakis, "Dynamic anomalography: Tracking network anomalies via sparsity and low-rank," *IEEE J. Sel. Tpoics in Signal Process.*, vol. 07, no. 1, pp. 50-66, Feb. 2013.

[10] M. Mardani, G. Mateos, and G. B. Giannakis, "Streaming algorithms for imputation of Big Data matrices and tensors," *IEEE Trans. Signal Process.*, 2014 (submitted).

[11] 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, Jun. 2009.

[12] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization," in *SIAM Rev.*, vol. 52, pp. 471–501, 2010.