Subspace Learning and Imputation for Streaming Big Data Matrices and Tensors

Morteza Mardani, *Student Member, IEEE*, Gonzalo Mateos, *Member, IEEE*, and Georgios B. Giannakis, *Fellow, IEEE*

***Abstract*—Extracting latent low-dimensional structure from high-dimensional 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. In this context, the present paper permeates benefits from rank minimization to scalable imputation of missing data, via tracking low-dimensional subspaces and unraveling latent (possibly multi-way) structure from *incomplete streaming* data. For low-rank matrix data, a subspace estimator is proposed based on an exponentially weighted least-squares criterion regularized with the nuclear norm. After recasting the nonseparable nuclear norm into a form amenable to online optimization, real-time algorithms with complementary strengths are developed, and their convergence is established under simplifying technical assumptions. In a stationary setting, the asymptotic estimates obtained offer the well-documented performance guarantees of the *batch* nuclear-norm regularized estimator. Under the same unifying framework, a novel online (adaptive) algorithm is developed to obtain multi-way decompositions of *low-rank tensors* with missing entries and perform imputation as a byproduct. Simulated tests with both synthetic as well as real Internet and cardiac magnetic resonance imagery (MRI) data confirm the efficacy of the proposed algorithms, and their superior performance relative to state-of-the-art alternatives.**

***Index Terms*—Low rank, matrix and tensor completion, missing data, streaming analytics, subspace tracking.**

I. Introduction

NOWADAYS ubiquitous e-commerce sites, the Web, and Internet-friendly portable devices generate massive volumes of data. The overwhelming consensus is that tremendous economic growth and improvement in quality of life can be effected by harnessing the potential benefits of analyzing this large volume of data. As a result, the task of extracting the most informative, yet low-dimensional structure from high-dimensional datasets is of paramount importance [21]. The sheer volume of data and the fact that oftentimes observations are acquired sequentially in time, motivate updating previously obtained 'analytics' rather than re-computing new ones from scratch each time a new datum becomes available [28], [36]. In addition, due to the disparate origins of the data, subsampling for faster data acquisition, or even privacy constraints, the datasets are often incomplete [2], [12].

In this context, consider streaming data comprising incomplete and noisy observations of the signal of interest $\mathbf{x}_t \in \mathbb{R}^P$ at time $t = 1, 2, \ldots$. Depending on the application, these acquired vectors could e.g., correspond to (vectorized) images, link traffic measurements collected across physical links of a computer network, or, movie ratings provided by Netflix users. Suppose that the signal sequence $\{\mathbf{x}_t\}_{t=1}^{\infty}$ lives in a *low-dimensional* ($\ll P$) linear subspace $\mathcal{L}_t$ of $\mathbb{R}^P$. Given the incomplete observations that are acquired sequentially in time, this paper deals first with (adaptive) online estimation of $\mathcal{L}_t$, and reconstruction of the signal $\mathbf{x}_t$ as a byproduct. This problem can be equivalently viewed as low-rank matrix completion with noise [12], solved online over t indexing the columns of relevant matrices, e.g., $\mathbf{X}_t := [\mathbf{x}_1, \ldots, \mathbf{x}_t]$.

Modern datasets are oftentimes indexed by three or more variables giving rise to a *tensor*, that is a data cube or a mutli-way array, in general [24]. It is not uncommon that one of these variables indexes time [32], [41], and that sizable portions of the data are missing [2], [6], [19], [27], [34]. Various data analytic tasks for network traffic, social networking, or medical data analysis aim at capturing underlying latent structure, which calls for high-order tensor factorizations even in the presence of missing data [2], [6], [27]. It is in principle possible to unfold the given tensor into a matrix and resort to either batch [19], [33], or, online matrix completion algorithms as the ones developed in the first part of this paper; see also [3], [14], [30]. 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 present paper also contributes towards fulfilling a pressing need in terms of analyzing streaming and incomplete multi-way data; namely, *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. [24] for a comprehensive tutorial treatment on tensor decompositions, algorithms, and applications.

Relation to prior work. Subspace tracking has a long history in signal processing. An early noteworthy representative is the

Manuscript received April 17, 2014; revised September 15, 2014 and January 21, 2015; accepted March 17, 2015. Date of publication March 30, 2015; date of current version April 14, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Anthony So. Work in this paper was supported by the MURI Grant No. AFOSR FA9550-10-1-0567. Part of the results in this paper were presented at the Thirty-Eighth IEEE International Conference on Acoustics, Speech, and Signal Processing, Vancouver, Canada, May 2013 and also at the Eighth IEEE Sensor Array and Multichannel Signal Processing Workshop, A Coruña, Spain, June 2014.

M. Mardani and G. B. Giannakis are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: morteza@umn.edu; georgios@umn.edu).

G. Mateos is with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: gmateosb@ur.rochester.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2015.2417491

1053-587X © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

projection approximation subspace tracking (PAST) algorithm [44]; see also [45]. Recently, an algorithm (termed GROUSE) for tracking subspaces from incomplete observations was put forth in [3], based on incremental gradient descent iterations on the Grassmannian manifold of subspaces. Recent analysis has shown that GROUSE can converge locally at an expected linear rate [5], and that it is tightly related to the incremental SVD algorithm [4]. PETRELS is a second-order recursive least-squares (RLS)-type algorithm, that extends the seminal PAST iterations to handle missing data [14]. As noted in [15], the performance of GROUSE is limited by the existence of barriers in the search path on the Grassmanian, which may lead to GROUSE iterations being trapped at local minima; see also [14]. Lack of regularization in PETRELS can also lead to unstable (even divergent) behaviors, especially when the amount of missing data is large. Accordingly, the convergence results for PETRELS are confined to the full-data setting where the algorithm boils down to PAST [14]. Relative to all aforementioned works, the algorithmic framework of this paper permeates benefits from *rank minimization* to low-dimensional subspace tracking and missing data imputation (Section III), offers provable convergence and theoretical performance guarantees in a stationary setting (Section IV), and is flexible to accommodate tensor streaming data models as well (Section V). While algorithms to impute incomplete tensors have been recently proposed in e.g., [2], [6], [19], [27], all existing approaches rely on batch processing.

Contributions. Leveraging the low dimensionality of the underlying subspace $\mathcal{L}_t$, an estimator is proposed based on an exponentially-weighted least-squares (EWLS) criterion regularized with the nuclear norm of $\mathbf{X}_t$. For a related data model, similar algorithmic construction ideas were put forth in our precursor paper [30], which dealt with real-time identification of network traffic anomalies. Here instead, the focus is on subspace tracking from incomplete measurements, and online matrix completion. Upon recasting the non-separable nuclear norm into a form amenable to online optimization as in [30], real-time subspace tracking algorithms with complementary strengths are developed in Section III, and their convergence is established under simplifying technical assumptions. For stationary data and under mild assumptions, the proposed online algorithms provably attain the global optimum of the batch nuclear-norm regularized problem (Section IV-C), whose quantifiable performance has well-appreciated merits [11], [12]. This optimality result as well as the convergence of the (first-order) stochastic-gradient subspace tracker established in Section IV-B, markedly broaden and complement the convergence claims in [30].

The present paper develops for the first time an *online* algorithm for decomposing *low-rank tensors with missing entries*; see also [32], [41] 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 [8], [25]). Leveraging stochastic gradient-descent iterations, a scalable, real-time algorithm is developed in Section V under the same rank-minimization framework utilized for the matrix case, which here entails minimizing an EWLS fitting error criterion regularized by separable Frobenius norms of the PARAFAC decomposition factors [6]. The proposed online algorithms offer 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 the main memory.

Simulated tests with synthetic as well as real Internet traffic data corroborate the effectiveness of the proposed algorithms for traffic estimation and anomaly detection, and its superior performance relative to state-of-the-art alternatives (available only for the matrix case [3], [14]). Additional tests with cardiac magnetic resonance imagery (MRI) data confirm the efficacy of the proposed tensor algorithm in imputing up to 75% missing entries. Conclusions are drawn in Section VII.

Notation: Bold uppercase (lowercase) letters will denote matrices (column vectors), and calligraphic letters will be used for sets. Tensors or multi-way arrays are denoted by bold, underlined uppercase letters. Operators $(\cdot)'$, $\mathrm{tr}(\cdot)$, $\mathbb{E}[\cdot]$, $\sigma_{\max}(\cdot)$, $\odot$, and $\circ$ will denote transposition, matrix trace, statistical expectation, maximum singular value, Hadamard product, and outer product, respectively; $|\cdot|$ will be used for the cardinality of a set, and the magnitude of a scalar. The positive semidefinite matrix $\mathbf{M}$ will be denoted by $\mathbf{M} \succeq \mathbf{0}$. The ℓ_p-norm of $\mathbf{x} \in \mathbb{R}^n$ is $\|\mathbf{x}\|_p := (\sum_{i=1}^n |x_i|^p)^{1/p}$ for $p \geq 1$. For two matrices $\mathbf{M}, \mathbf{U} \in \mathbb{R}^{n \times p}$, $\langle \mathbf{M}, \mathbf{U} \rangle := \mathrm{tr}(\mathbf{M}'\mathbf{U})$ denotes their trace inner product, and $\|\mathbf{M}\|_F := \sqrt{\mathrm{tr}(\mathbf{M}\mathbf{M}')}$ is the Frobenious norm. The $n \times n$ identity matrix will be represented by $\mathbf{I}_n$, while $\mathbf{0}_n$ will stand for the $n \times 1$ vector of all zeros, $\mathbf{0}_{n \times p} := \mathbf{0}_n \mathbf{0}_p'$, and $[n] := \{1, 2, \ldots, n\}$.

II. Preliminaries and Problem Statement

Consider a sequence of high-dimensional data vectors, which are corrupted with additive noise and some of their entries may be missing. At time t, the incomplete streaming observations are modeled as

$$\mathcal{P}_{\omega_t}(\mathbf{y}_t) = \mathcal{P}_{\omega_t}(\mathbf{x}_t + \mathbf{v}_t), \quad t = 1, 2, \ldots \tag{1}$$

where $\mathbf{x}_t \in \mathbb{R}^P$ is the signal of interest, and $\mathbf{v}_t$ stands for the noise. The set $\omega_t \subset \{1, 2, \ldots, P\}$ contains the indices of available observations, while the corresponding sampling operator $\mathcal{P}_{\omega_t}(\cdot)$ sets the entries of its vector argument not in ω_t to zero, and keeps the rest unchanged; note that $\mathcal{P}_{\omega_t}(\mathbf{y}_t) \in \mathbb{R}^P$. Suppose that the sequence $\{\mathbf{x}_t\}_{t=1}^{\infty}$ lives in a *low-dimensional* ($\ll P$) linear subspace $\mathcal{L}_t$, which is allowed to change slowly over time. Given the incomplete observations $\{\mathcal{P}_{\omega_\tau}(\mathbf{y}_\tau)\}_{\tau=1}^t$, the first part of this paper deals with online (adaptive) estimation of $\mathcal{L}_t$, and reconstruction of $\mathbf{x}_t$ as a byproduct. The reconstruction here involves imputing the missing elements, and denoising the observed ones.

A. Challenges Facing Large-Scale Nuclear Norm Minimization

Collect the indices of available observations up to time t in the set $\Omega_t := \cup_{\tau=1}^t \omega_\tau$, and the actual batch of observations in the matrix $\mathcal{P}_{\Omega_t}(\mathbf{Y}_t) := [\mathcal{P}_{\omega_1}(\mathbf{y}_1), \ldots, \mathcal{P}_{\omega_t}(\mathbf{y}_t)] \in \mathbb{R}^{P \times t}$; see also Fig. 1. Likewise, introduce matrix $\mathbf{X}_t$ containing the signal of interest. Since $\mathbf{x}_t$ lies in a low-dimensional subspace, $\mathbf{X}_t$ is (approximately) a *low-rank* matrix. A natural estimator leveraging the low rank property of $\mathbf{X}_t$ attempts to fit the incomplete data $\mathcal{P}_{\Omega_t}(\mathbf{Y}_t)$ to $\mathbf{X}_t$ in the least-squares (LS) sense, as well as

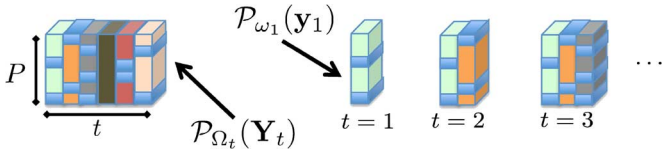

Fig. 1. Matrix data with missing entries. (Left) Batch data $\mathcal{P}_{\Omega_t}(\mathbf{Y}_t)$ available at time t. (Right) Streaming data, where vectors $\mathcal{P}_{\omega_t}(\mathbf{y}_t)$ become available for $t = 1, 2, \ldots$.

minimize the rank of $\mathbf{X}_t$. Unfortunately, albeit natural the rank criterion is in general NP-hard to optimize [33]. This motivates solving for [12]

$$\text{(P1)} \quad \hat{\mathbf{X}}_t := \arg\min_{\mathbf{X}} \left\{ \frac{1}{2} \|\mathcal{P}_{\Omega_t}(\mathbf{Y}_t - \mathbf{X})\|_F^2 + \lambda_t \|\mathbf{X}\|_* \right\}$$

where the nuclear norm $\|\mathbf{X}_t\|_* := \sum_k \sigma_k(\mathbf{X}_t)$ (σ_k is the k-th singular value) is adopted as a convex surrogate to $\text{rank}(\mathbf{X}_t)$ [16], and λ_t is a (possibly time-varying) rank-controlling parameter. Scalable imputation algorithms for streaming observations should effectively overcome the following challenges: (c1) the problem size can easily become quite large, since the number of optimization variables Pt grows with time; (c2) existing batch iterative solvers for (P1) typically rely on costly SVD computations per iteration; see e.g., [11]; and (c3) (columnwise) nonseparability of the nuclear-norm challenges online processing when new columns $\{\mathcal{P}_{\omega_t}(\mathbf{y}_t)\}$ arrive sequentially in time. In the following subsection, the 'Big Data' challenges (c1)–(c3) are dealt with to arrive at an efficient online algorithm in Section III.

B. A Separable Low-Rank Regularization

To limit the computational complexity and memory storage requirements of the algorithm sought, it is henceforth assumed that the dimensionality of the underlying time-varying subspace $\mathcal{L}_t$ is bounded by a known quantity ρ. Accordingly, it is natural to require $\text{rank}(\hat{\mathbf{X}}_t) \leq \rho$. As argued later in Remark 1, the smaller the value of ρ, the more efficient the algorithm becomes. Because $\text{rank}(\hat{\mathbf{X}}_t) \leq \rho$, one can factorize the matrix decision variable as $\mathbf{X} = \mathbf{L}\mathbf{Q}'$, where $\mathbf{L}$ and $\mathbf{Q}$ are $P \times \rho$ and $t \times \rho$ matrices, respectively. Such a bilinear decomposition suggests $\mathcal{L}_t$ is spanned by the columns of $\mathbf{L}$, while the rows of $\mathbf{Q}$ are the projections of $\{\mathbf{x}_t\}$ onto $\mathcal{L}_t$.

To address (c1) and (c2) [along with (c3) as it will become clear in Section III], consider the following alternative characterization of the nuclear norm [39], [40]

$$\|\mathbf{X}\|_* := \min_{\{\mathbf{L},\mathbf{Q}\}} \frac{1}{2} \left\{ \|\mathbf{L}\|_F^2 + \|\mathbf{Q}\|_F^2 \right\}, \quad \text{s. to } \mathbf{X} = \mathbf{L}\mathbf{Q}'. \quad (2)$$

The optimization (2) is over all possible bilinear factorizations of $\mathbf{X}$, so that the number of columns ρ of $\mathbf{L}$ and $\mathbf{Q}$ is also a variable. Leveraging (2), the following nonconvex reformulation of (P1) provides an important first step towards obtaining an online algorithm:

$$\text{(P2)} \quad \min_{\{\mathbf{L},\mathbf{Q}\}} \frac{1}{2} \left\| \mathcal{P}_{\Omega_t}(\mathbf{Y}_t - \mathbf{L}\mathbf{Q}') \right\|_F^2 + \frac{\lambda_t}{2} \left\{ \|\mathbf{L}\|_F^2 + \|\mathbf{Q}\|_F^2 \right\}.$$

The number of variables is reduced from Pt in (P1) to $\rho(P+t)$ in (P2), which can be significant when ρ is small, and both P and t are large. Most importantly, it follows that adopting the separable (across the time-indexed columns of $\mathbf{Q}$) Frobenius-norm regularization in (P2) comes with no loss of optimality relative to (P1), provided $\rho \geq \text{rank}(\hat{\mathbf{X}}_t)$.

By finding the global minimum of (P2), one can recover the optimal solution of (P1). However, since (P2) is nonconvex, it may have stationary points which need not be globally optimum. Interestingly, results in [10], [29] offer a global optimality certificate for stationary points of (P2). Specifically, if $\{\bar{\mathbf{L}}_t, \bar{\mathbf{Q}}_t\}$ is a stationary point of (P2) (obtained with any practical solver) satisfying the qualification inequality $\sigma_{\max}[\mathcal{P}_{\Omega_t}(\mathbf{Y}_t - \bar{\mathbf{L}}_t\bar{\mathbf{Q}}_t')] \leq \lambda_t$, then $\hat{\mathbf{X}}_t := \bar{\mathbf{L}}_t\bar{\mathbf{Q}}_t'$ is the globally optimal solution of (P1) [10], [29].

III. Online Rank Minimization for Matrix Imputation

In 'Big Data' applications the collection of massive amounts of data far outweigh the ability of modern computers to store and analyze them as a batch. In addition, in practice (possibly incomplete) observations are acquired sequentially in time which motivates updating previously obtained estimates rather than re-computing new ones from scratch each time a new datum becomes available. As stated in Section II, the goal is to recursively track the low-dimensional subspace $\mathcal{L}_t$, and subsequently estimate $\hat{\mathbf{x}}_t$ per time t from historical observations $\{\mathcal{P}_{\omega_\tau}(\mathbf{y}_\tau)\}_{\tau=1}^t$, naturally placing more importance on recent measurements. To this end, one possible adaptive counterpart to (P2) is the exponentially-weighted LS (EWLS) estimator found by minimizing the empirical cost

$$\text{(P3)} \quad \min_{\{\mathbf{L},\mathbf{Q}\}} \sum_{\tau=1}^{t} \theta^{t-\tau} \left[\frac{1}{2} \|\mathcal{P}_{\omega_\tau}(\mathbf{y}_\tau - \mathbf{L}\mathbf{q}_\tau)\|_2^2 + \frac{\bar{\lambda}_t}{2} \|\mathbf{L}\|_F^2 + \frac{\lambda_t}{2} \|\mathbf{q}_\tau\|_2^2 \right]$$

where $\mathbf{Q} := [\mathbf{q}_1, \ldots, \mathbf{q}_t]$, $\bar{\lambda}_t := \lambda_t / \sum_{\tau=1}^t \theta^{t-\tau}$, and $0 < \theta \leq 1$ is the so-termed forgetting factor. When $\theta < 1$, data in the distant past are exponentially downweighted, which facilitates tracking in nonstationary environments. In the case of infinite memory ($\theta = 1$), the formulation (P3) coincides with the batch estimator (P2). This is the reason for the time-varying factor $\bar{\lambda}_t$ weighting $\|\mathbf{L}\|_F^2$.

We first introduced the basic idea of performing online rank-minimization leveraging the separable nuclear-norm regularization (2) in [30] (and its conference precursor), in the context of unveiling network traffic anomalies. Since then, the approach has gained popularity in real-time non-negative matrix factorization for singing voice separation from its music accompaniment [38], and online robust PCA [17], to name a few examples. Instead, the novelty here is on subspace tracking from incomplete measurements, as well as online low-rank matrix and tensor completion.

A. Alternating Recursive LS for Subspace Tracking From Incomplete Data

Towards deriving a real-time, computationally efficient, and recursive solver of (P3), an alternating-minimization (AM) method 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 in detail in Section IV-A. Per time instant t, a new datum $\{\mathcal{P}_{\omega_t}(\mathbf{y}_t)\}$ is drawn and $\mathbf{q}_t$ is estimated via

$$\mathbf{q}[t] = \arg\min_{\mathbf{q}} \left[\frac{1}{2} \|\mathcal{P}_{\omega_t}(\mathbf{y}_t - \mathbf{L}[t-1]\mathbf{q})\|_2^2 + \frac{\lambda_t}{2} \|\mathbf{q}\|_2^2 \right] \quad (3)$$

which is an ℓ_2-norm regularized LS (ridge-regression) problem. It admits the closed-form solution

$$\mathbf{q}[t] = \left(\lambda_t \mathbf{I}_\rho + \mathbf{L}'[t-1]\mathbf{\Omega}_t \mathbf{L}[t-1]\right)^{-1} \mathbf{L}'[t-1]\mathcal{P}_{\omega_t}(\mathbf{y}_t) \quad (4)$$

where diagonal matrix $\mathbf{\Omega}_t \in \{0,1\}^{P\times P}$ is such that $[\mathbf{\Omega}_t]_{p,p} = 1$ if $p \in \omega_t$, and is zero elsewhere. In the second step of the AM scheme, the updated subspace matrix $\mathbf{L}[t]$ is obtained by minimizing (P3) with respect to $\mathbf{L}$, while the optimization variables $\{\mathbf{q}_\tau\}_{\tau=1}^t$ are fixed and take the values $\{\mathbf{q}[\tau]\}_{\tau=1}^t$, namely

$$\mathbf{L}[t] = \arg\min_{\mathbf{L}} \left[\frac{\lambda_t}{2}\|\mathbf{L}\|_F^2 + \sum_{\tau=1}^t \theta^{t-\tau}\frac{1}{2}\left\|\mathcal{P}_{\omega_\tau}\left(\mathbf{y}_\tau - \mathbf{L}\mathbf{q}[\tau]\right)\right\|_2^2\right]. \quad (5)$$

Notice that (5) decouples over the rows of $\mathbf{L}$ which are obtained in parallel via

$$\mathbf{l}_p[t] = \arg\min_{\mathbf{l}} \left[\frac{\lambda_t}{2}\|\mathbf{l}\|_2^2 + \sum_{\tau=1}^t \theta^{t-\tau}\omega_{p,\tau}\left(y_{p,\tau} - \mathbf{l}'\mathbf{q}[\tau]\right)^2\right], \quad (6)$$

for $p = 1,\ldots,P$, where $\omega_{p,\tau}$ denotes the p-th diagonal entry of $\mathbf{\Omega}_\tau$. For $\theta = 1$ and fixed $\lambda_t = \lambda, \forall t$, subproblems (6) can be efficiently solved using recursive LS (RLS) [37]. Upon defining $\mathbf{s}_p[t] := \sum_{\tau=1}^t \theta^{t-\tau}\omega_{p,\tau}y_{p,\tau}\mathbf{q}[\tau]$, $\mathbf{H}_p[t] := \sum_{\tau=1}^t \theta^{t-\tau}\omega_{p,\tau}\mathbf{q}[\tau]\mathbf{q}'[\tau] + \lambda_t\mathbf{I}_\rho$, and $\mathbf{M}_p[t] := \mathbf{H}_p^{-1}[t]$, one updates

$$\mathbf{s}_p[t] = \mathbf{s}_p[t-1] + \omega_{p,t}y_{p,t}\mathbf{q}[t]$$
$$\mathbf{M}_p[t] = \mathbf{M}_p[t-1] - \omega_{p,t}\frac{\mathbf{M}_p[t-1]\mathbf{q}[t]\mathbf{q}'[t]\mathbf{M}_p[t-1]}{1+\mathbf{q}'[t]\mathbf{M}_p[t-1]\mathbf{q}[t]}$$

and forms $\mathbf{l}_p[t] = \mathbf{M}_p[t]\mathbf{s}_p[t]$, for $p = 1,\ldots,P$.

However, for $0 < \theta < 1$ the regularization term $(\lambda_t/2)\|\mathbf{l}\|_2^2$ in (6) makes it impossible to express $\mathbf{H}_p[t]$ in terms of $\mathbf{H}_p[t-1]$ plus a rank-one correction. Hence, one cannot resort to the matrix inversion lemma and update $\mathbf{M}_p[t]$ with quadratic complexity only. Based on direct inversion of each $\mathbf{H}_p[t]$, the alternating recursive LS algorithm for subspace tracking from incomplete data is tabulated under Algorithm 1.

Algorithm 1: Alternating LS for subspace tracking from incomplete observations

input $\{\mathcal{P}_{\omega_\tau}(\mathbf{y}_\tau), \omega_\tau\}_{\tau=1}^\infty$, $\{\lambda_\tau\}_{\tau=1}^\infty$, and θ.

initialize $\mathbf{G}_p[0] = \mathbf{0}_{\rho\times\rho}$, $\mathbf{s}_p[0] = \mathbf{0}_\rho, p = 1,\ldots,P$, and $\mathbf{L}[0]$ at random.

for $t = 1, 2, \ldots$ **do**

$\mathbf{D}[t] = (\lambda_t\mathbf{I}_\rho + \mathbf{L}'[t-1]\mathbf{\Omega}_t\mathbf{L}[t-1])^{-1}\mathbf{L}'[t-1]$.

$\mathbf{q}[t] = \mathbf{D}[t]\mathcal{P}_{\omega_t}(\mathbf{y}_t)$.

$\mathbf{G}_p[t] = \theta\mathbf{G}_p[t-1] + \omega_{p,t}\mathbf{q}[t]\mathbf{q}[t]', p = 1,\ldots,P$.

$\mathbf{s}_p[t] = \theta\mathbf{s}_p[t-1] + \omega_{p,t}y_{p,t}\mathbf{q}[t], p = 1,\ldots,P$.

$\mathbf{l}_p[t] = (\mathbf{G}_p[t] + \lambda_t\mathbf{I}_\rho)^{-1}\mathbf{s}_p[t], p = 1,\ldots,P$.

return $\hat{\mathbf{x}}_t := \mathbf{L}[t]\mathbf{q}[t]$.

end for

Remark 1 (Computational Cost): Careful inspection of Algorithm 1 reveals that the main computational burden stems from $\rho\times\rho$ inversions to update the subspace matrix $\mathbf{L}[t]$. The per iteration complexity for performing the inversions is $\mathcal{O}(P\rho^3)$ (which could be further reduced if one leverages also the symmetry of $\mathbf{G}_p[t]$), while the cost for the rest of operations including multiplication and additions is $\mathcal{O}(P\rho^2)$. The overall cost of the algorithm per iteration can thus be safely estimated as $\mathcal{O}(P\rho^3)$, which can be affordable since ρ is typically small (cf. the low rank assumption). In addition, for the infinite memory case $\theta = 1$ where the RLS update is employed, the overall cost is further reduced to $\mathcal{O}(|\omega_t|\rho^2)$.

Remark 2 (Tuning λ_t): To tune λ_t one can resort to the heuristic rules proposed in [12], which apply under the following assumptions: i) $v_{p,t} \sim \mathcal{N}(0,\sigma^2)$; ii) elements of Ω_t are independently sampled with probability π; and, iii) P and t are large enough. Accordingly, one can pick $\lambda_t = (\sqrt{P} + \sqrt{t_e})\sqrt{\pi}\sigma$, where $t_e := \sum_{\tau=1}^t \theta^{t-\tau}$ is the effective time window. Note that λ_t naturally increases with time when $\theta = 1$, whereas for $\theta < 1$ a fixed value $\lambda_t = \lambda$ is well justified since the data window is effectively finite.

B. Low-Complexity Stochastic-Gradient Subspace Updates

Towards reducing Algorithm's 1 computational complexity in updating the subspace $\mathbf{L}[t]$, this section aims at developing lightweight algorithms which better suit the 'Big Data' landscape. To this end, the basic AM framework in Section III-A will be retained, and the update for $\mathbf{q}[t]$ will be identical [cf. (4)]. However, instead of exactly solving an unconstrained quadratic program per iteration to obtain $\mathbf{L}[t]$ [cf. (5)], the subspace estimates will be obtained via stochastic-gradient descent (SGD) iterations. As will be shown later on, these updates can be traced to inexact solutions of a certain quadratic program different from (5).

For $\theta = 1$, it is shown in Section IV-A that Algorithm 1's subspace estimate $\mathbf{L}[t]$ is obtained by minimizing the empirical cost function $\hat{C}_t(\mathbf{L}) = (1/t)\sum_{\tau=1}^t f_\tau(\mathbf{L})$, where

$$f_t(\mathbf{L}) := \frac{1}{2}\|\mathcal{P}_{\omega_\tau}(\mathbf{y}_\tau - \mathbf{L}\mathbf{q}[\tau])\|_2^2 + \frac{\lambda}{2t}\|\mathbf{L}\|_F^2 + \frac{\lambda}{2}\|\mathbf{q}[\tau]\|_2^2,\ \tau = 1,\ldots \quad (7)$$

By the law of large numbers, if data $\{\mathcal{P}_{\omega_t}(\mathbf{y}_t)\}_{t=1}^\infty$ are stationary, solving $\min_{\mathbf{L}}\lim_{t\to\infty}\hat{C}_t(\mathbf{L})$ yields the desired minimizer of the *expected* cost $\mathbb{E}[C_t(\mathbf{L})]$, where the expectation is taken with respect to the unknown probability distribution of the data. A standard approach to achieve this same goal—typically with reduced computational complexity—is to drop the expectation (or the sample averaging operator for that matter), and update the subspace via SGD; see e.g., [37]

$$\mathbf{L}[t] = \mathbf{L}[t-1] - (\mu[t])^{-1}\nabla f_t(\mathbf{L}[t-1]) \quad (8)$$

where $(\mu[t])^{-1}$ is the step size, and $\nabla f_t(\mathbf{L}) = -\mathcal{P}_{\omega_t}(\mathbf{y}_t - \mathbf{L}\mathbf{q}[t])\mathbf{q}'[t] + (\lambda/t)\mathbf{L}$. The subspace update $\mathbf{L}[t]$ is nothing but the minimizer of a second-order approximation $Q_{\mu[t],t}(\mathbf{L}, \mathbf{L}[t-1])$ of $f_t(\mathbf{L})$ around the previous subspace $\mathbf{L}[t-1]$, where

$$Q_{\mu,t}(\mathbf{L}_1,\mathbf{L}_2) := f_t(\mathbf{L}_2) + \langle\mathbf{L}_1 - \mathbf{L}_2, \nabla f_t(\mathbf{L}_2)\rangle + \frac{\mu}{2}\|\mathbf{L}_1 - \mathbf{L}_2\|_f^2.$$

To tune the step size, the backtracking rule is adopted, whereby the non-increasing step size sequence $\{(\mu[t])^{-1}\}$ decreases geometrically at certain iterations to guarantee the quadratic function $Q_{\mu[t],t}(\mathbf{L},\mathbf{L}[t-1])$ majorizes $f_t(\mathbf{L})$ at the new update $\mathbf{L}[t]$. Other choices of the step size are discussed in Section IV. It is observed that different from Algorithm 1, no matrix inversions are involved in the update of the subspace $\mathbf{L}[t]$. In the context of adaptive filtering, first-order SGD algorithms such as (7) are known to converge slower than RLS. This is expected since RLS can be shown to be an instance of Newton's (second-order) optimization method [37, Ch. 4].

Algorithm 2: Online SGD for subspace tracking from incomplete observations

input $\{\mathcal{P}_{\omega_\tau}(\mathbf{y}_\tau),\omega_\tau\}_{\tau=1}^{\infty}$, $\rho,\lambda,\eta>1$.

initialize $\mathbf{L}[0]$ at random, $\mu[0]>0$, $\tilde{\mathbf{L}}[1]:=\mathbf{L}[0]$, and $k[1]:=1$.

for $t=1,2,\ldots$ **do**

$\mathbf{D}[t]=(\lambda\mathbf{I}_\rho+\mathbf{L}'[t-1]\mathbf{\Omega}_t\mathbf{L}[t-1])^{-1}\mathbf{L}'[t-1]$

$\mathbf{q}[t]=\mathbf{D}[t]\mathcal{P}_{\omega_t}(\mathbf{y}_t)$

Find the smallest nonnegative integer $i[t]$ such that with $\bar{\mu}:=\eta^{i[t]}\mu[t-1]$

$$f_t\Big(\tilde{\mathbf{L}}[t]-\Big(\frac{1}{\bar{\mu}}\Big)\nabla f_t\Big(\tilde{\mathbf{L}}[t]\Big)\Big)\leq Q_{\bar{\mu},t}\Big(\tilde{\mathbf{L}}[t]-\Big(\frac{1}{\bar{\mu}}\Big)\nabla f_t\Big(\tilde{\mathbf{L}}[t]\Big),\tilde{\mathbf{L}}[t]\Big)$$

holds, and set $\mu[t]=\eta^{i[t]}\mu[t-1]$.

$\mathbf{L}[t]=\tilde{\mathbf{L}}[t]-(1/\mu[t])\nabla f_t(\tilde{\mathbf{L}}[t])$.

$k[t+1]=\frac{1+\sqrt{1+4k^2[t]}}{2}$.

$\tilde{\mathbf{L}}[t+1]=\mathbf{L}[t]+(\frac{k[t]-1}{k[t+1]})(\mathbf{L}[t]-\mathbf{L}[t-1])$.

end for

return $\hat{\mathbf{x}}[t]:=\mathbf{L}[t]\mathbf{q}[t]$.

Building on the increasingly popular *accelerated* gradient methods for batch smooth optimization [7], [31], the idea here is to speed-up the learning rate of the estimated subspace (8), without paying a penalty in terms of computational complexity per iteration. The critical difference between standard gradient algorithms and the so-termed Nesterov's variant, is that the accelerated updates take the form $\mathbf{L}[t]=\tilde{\mathbf{L}}[t]-(\mu[t])^{-1}\nabla f_t(\tilde{\mathbf{L}}[t])$, which relies on a judicious linear combination $\tilde{\mathbf{L}}[t-1]$ of the previous pair of iterates $\{\mathbf{L}[t-1],\mathbf{L}[t-2]\}$. Specifically, the choice $\tilde{\mathbf{L}}[t]=\mathbf{L}[t-1]+\frac{k[t-1]-1}{k[t]}(\mathbf{L}[t-1]-\mathbf{L}[t-2])$, where $k[t]=[1+\sqrt{4k^2[t-1]+1}]/2$, has been shown to significantly accelerate batch gradient algorithms resulting in convergence rate no worse than $\mathcal{O}(1/k^2)$; see e.g., [7] and references therein. Using this acceleration technique in conjunction with a backtracking stepsize rule [9], a fast online SGD algorithm for imputing missing entries is tabulated under Algorithm 2. Clearly, a standard (non accelerated) SGD algorithm with backtracking step size rule is subsumed as a special case, when $k[t]=1$, $t=1,2,\ldots$. In this case, complexity is $\mathcal{O}(|\omega_t|\rho^2)$ mainly due to update of $\mathbf{q}_t$, while the accelerated algorithm incurs an additional cost $O(P\rho)$ for the subspace extrapolation step.

IV. Performance Guarantees

This section studies the performance of the proposed first- and second-order online algorithms for the infinite memory special case; that is $\theta=1$. In the sequel, to make the analysis tractable the following assumptions are adopted:

(A1) Processes $\{\omega_t\}_{t=1}^{\infty}$ and $\{\mathcal{P}_{\omega_t}(\mathbf{y}_t)\}_{t=1}^{\infty}$ are independent and identically distributed (i.i.d.);

(A2) Sequence $\{\mathcal{P}_{\omega_t}(\mathbf{y}_t)\}_{t=1}^{\infty}$ is uniformly bounded; and

(A3) Iterates $\{\mathbf{L}[t]\}_{t=1}^{\infty}$ lie in a compact set.

To clearly delineate the scope of the analysis, it is worth commenting on (A1)–(A3) and the factors that influence their satisfaction. Regarding (A1), the acquired data is assumed statistically independent across time as it is customary when studying the stability and performance of online (adaptive) algorithms [37]. While independence is required for tractability, (A1) may be grossly violated because the observations $\{\mathcal{P}_{\omega_t}(\mathbf{y}_t)\}$ are correlated across time (cf. the fact that $\{\mathbf{x}_t\}$ lies in a low-dimensional subspace). Still, in accordance with the adaptive filtering folklore e.g., [37], as $\theta\to 1$ or $(\mu[t])^{-1}\to 0$ the upshot of the analysis based on i.i.d. data extends accurately to the pragmatic setting whereby the observations are correlated. Uniform boundedness of $\mathcal{P}_{\omega_t}(\mathbf{y}_t)$ [cf. (A2)] is natural in practice as it is imposed by the data acquisition process. The bounded subspace requirement in (A3) is a technical assumption that simplifies the analysis, and has been corroborated via extensive computer simulations.

A. Convergence Analysis of the Second-Order Algorithm

Convergence of the iterates generated by Algorithm 1 (with $\theta=1$) is established first. Upon defining

$$g_t(\mathbf{L},\mathbf{q}):=\frac{1}{2}\|\mathcal{P}_{\omega_t}(\mathbf{y}_t-\mathbf{L}\mathbf{q})\|_2^2+\frac{\lambda_t}{2}\|\mathbf{q}\|_2^2$$

in addition to $\ell_t(\mathbf{L}):=\min_{\mathbf{q}}g_t(\mathbf{L},\mathbf{q})$, Algorithm 1 aims at minimizing the following *average* cost function at time t

$$C_t(\mathbf{L}):=\frac{1}{t}\sum_{\tau=1}^{t}\ell_\tau(\mathbf{L})+\frac{\lambda_t}{2t}\|\mathbf{L}\|_F^2. \tag{9}$$

Normalization (by t) ensures that the cost function does not grow unbounded as time evolves. For any finite t, (9) is essentially identical to the batch estimator in (P2) up to a scaling, which does not affect the value of the minimizer. Note that as time evolves, minimization of C_t becomes increasingly complex computationally. Hence, at time t the subspace estimate $\mathbf{L}[t]$ is obtained by minimizing the *approximate* cost function

$$\hat{C}_t(\mathbf{L})=\frac{1}{t}\sum_{\tau=1}^{t}g_\tau\left(\mathbf{L},\mathbf{q}[\tau]\right)+\frac{\lambda_t}{2t}\|\mathbf{L}\|_F^2 \tag{10}$$

in which $\mathbf{q}[t]$ is obtained based on the prior subspace estimate $\mathbf{L}[t-1]$ after solving $\mathbf{q}[t]=\arg\min_{\mathbf{q}}g_t(\mathbf{L}[t-1],\mathbf{q})$ [cf. (3)]. Obtaining $\mathbf{q}[t]$ this way resembles the projection approximation adopted in [44]. Since $\hat{C}_t(\mathbf{L})$ is a smooth convex quadratic function, the minimizer $\mathbf{L}[t]=\arg\min_{\mathbf{L}}\hat{C}_t(\mathbf{L})$ is the solution of the linear equation $\nabla\hat{C}_t(\mathbf{L}[t])=\mathbf{0}_{P\times\rho}$.

So far, it is apparent that since $g_t(\mathbf{L},\mathbf{q}[t])\geq\min_{\mathbf{q}}g_t(\mathbf{L},\mathbf{q})=\ell_t(\mathbf{L})$, the approximate cost function $\hat{C}_t(\mathbf{L}[t])$ overestimates the

target cost $C_t(\mathbf{L}[t])$, for $t = 1, 2, \ldots$. However, it is not clear whether the subspace iterates $\{\mathbf{L}[t]\}_{t=1}^{\infty}$ converge, and most importantly, how well can they optimize the target cost function C_t. The good news is that $\hat{C}_t(\mathbf{L}[t])$ asymptotically approaches $C_t(\mathbf{L}[t])$, and the subspace iterates null $\nabla C_t(\mathbf{L}[t])$ as well, both as $t \to \infty$. This result is summarized in the next proposition.

Proposition 1: Under (A1)–(A3) and $\theta = 1$ in Algorithm 1, if $\lambda_t = \lambda \forall t$ and $\lambda_{\min}[\nabla^2 \hat{C}_t(\mathbf{L})] \geq c$ for some $c > 0$, then $\lim_{t\to\infty} \nabla C_t(\mathbf{L}[t]) = \mathbf{0}_{P\times\rho}$ almost surely (a.s.), i.e., the subspace iterates $\{\mathbf{L}[t]\}_{t=1}^{\infty}$ asymptotically fall into the stationary point set of the batch problem (P2).

It is worth noting that the pattern and the amount of misses, summarized in the sampling sets $\{\omega_t\}$, play a key role towards satisfying the Hessian's positive semi-definiteness condition. In fact, random misses are desirable since the Hessian $\nabla^2 \hat{C}_t(\mathbf{L}) = \frac{\lambda}{t}\mathbf{I}_{P\rho} + \frac{1}{t}\sum_{\tau=1}^{t}(\mathbf{q}[\tau]\mathbf{q}'[\tau]) \otimes \mathbf{\Omega}_\tau$ is more likely to satisfy $\nabla^2 \hat{C}_t(\mathbf{L}) \succeq c\mathbf{I}_{P\rho}$, for some $c > 0$.

The proof of Proposition 1 is inspired by [28] which establishes convergence of an online dictionary learning algorithm using the theory of martingale sequences. Details can be found in our companion paper [30], and in a nutshell the proof procedure proceeds in the following two main steps:

(S1) Establish that the approximate cost sequence $\{\hat{C}_t(\mathbf{L}[t])\}$ asymptotically converges to the target cost sequence $\{C_t(\mathbf{L}[t])\}$. To this end, it is first proved that $\{\hat{C}_t(\mathbf{L}[t])\}_{t=1}^{\infty}$ is a quasi-martingale sequence, and hence convergent a.s. This relies on the fact that $g_t(\mathbf{L}, \mathbf{q}[t])$ is a *tight* upper bound approximation of $\ell_t(\mathbf{L})$ at the previous update $\mathbf{L}[t-1]$, namely, $g_t(\mathbf{L}, \mathbf{q}[t]) \geq \ell_t(\mathbf{L}), \forall \mathbf{L} \in \mathbb{R}^{P\times\rho}$, and $g_t(\mathbf{L}[t-1], \mathbf{q}[t]) = \ell_t(\mathbf{L}[t-1])$.

(S2) Under certain regularity assumptions on g_t, establish that convergence of the cost sequence $\{\hat{C}_t(\mathbf{L}[t]) - C_t(\mathbf{L}[t])\} \to 0$ yields convergence of the gradients $\{\nabla\hat{C}_t(\mathbf{L}[t]) - \nabla C_t(\mathbf{L}[t])\} \to 0$, which subsequently results in $\lim_{t\to\infty} \nabla C_t(\mathbf{L}[t]) = \mathbf{0}_{P\times\rho}$.

B. Convergence Analysis of the First-Order Algorithm

Convergence of the SGD iterates (without Nesterov's acceleration) is established here, by resorting to the proof techniques adopted for the second-order algorithm in Section IV-A. The basic idea is to judiciously derive an appropriate surrogate $\tilde{C}_t$ of C_t, whose minimizer coincides with the SGD update for $\mathbf{L}[t]$ in (8). The surrogate $\tilde{C}_t$ then plays the same role as $\hat{C}_t$, associated with the second-order algorithm towards the convergence analysis. Recall that $\mathbf{q}[t] = \arg\min_{\mathbf{q}\in\mathbb{R}^\rho} g_t(\mathbf{L}[t-1], \mathbf{q})$. In this direction, in the average cost $\hat{C}_t(\mathbf{L}) = \frac{1}{t}\sum_{\tau=1}^{t} f_t(\mathbf{L}, \mathbf{q}[t])$ [cf. (P3) for $\theta = 1$], with $f_t(\mathbf{L}, \mathbf{q}[t]) = g_t(\mathbf{L}, \mathbf{q}[t]) + \frac{\lambda_t}{2t}\|\mathbf{L}\|_F^2$ one can further approximate f_t using the second-order Taylor expansion at the previous subspace update $\mathbf{L}[t-1]$. This yields

$$\tilde{f}_t(\mathbf{L}, \mathbf{q}[t]) = f_t(\mathbf{L}[t-1], \mathbf{q}[t]) + \frac{\alpha_t}{2}\|\mathbf{L} - \mathbf{L}[t-1]\|_F^2 + \mathrm{tr}\left\{\nabla_{\mathbf{L}} f_t(\mathbf{L}[t-1], \mathbf{q}[t])(\mathbf{L} - \mathbf{L}[t-1])'\right\} \quad (11)$$

where $\alpha_t \geq \|\nabla^2 f_t(\mathbf{L}[t-1], \mathbf{q}[t])\| = \|(\mathbf{q}[t]\mathbf{q}'[t]) \otimes \mathbf{\Omega}_t + \lambda_t/2t\mathbf{I}_{P\rho}\|$.

It is useful to recognize that the surrogate $\tilde{f}_t$ is a tight approximation of f_t in the sense that: (i) it globally majorizes the original cost function f_t, i.e., $\tilde{f}_t(\mathbf{L}, \mathbf{q}[t]) \geq f_t(\mathbf{L}, \mathbf{q}[t]), \forall \mathbf{L} \in \mathbb{R}^{P\times\rho}$; (ii) it is locally tight, namely $\tilde{f}_t(\mathbf{L}[t-1], \mathbf{q}[t]) = f_t(\mathbf{L}[t-1], \mathbf{q}[t])$; and, (iii) its gradient is locally tight, namely $\nabla_{\mathbf{L}}\tilde{f}_t(\mathbf{L}[t-1], \mathbf{q}[t]) = \nabla_{\mathbf{L}} f_t(\mathbf{L}[t-1], \mathbf{q}[t])$. Consider now the average approximate cost

$$\tilde{C}_t(\mathbf{L}) = \frac{1}{t}\sum_{\tau=1}^{t}\tilde{f}_\tau(\mathbf{L}, \mathbf{q}[\tau]) \quad (12)$$

where due to (i) it follows that $\tilde{C}_t(\mathbf{L}) \geq \hat{C}_t(\mathbf{L}) \geq C_t(\mathbf{L})$ holds for all $\mathbf{L} \in \mathbb{R}^{P\times\rho}$. The subspace update $\mathbf{L}[t]$ is then obtained as $\mathbf{L}[t] := \arg\min_{\mathbf{L}\in\mathbb{R}^{P\times\rho}} \tilde{C}_t(\mathbf{L})$, which amounts to nulling the gradient [cf. (11) and (12)]

$$\nabla\tilde{C}_t(\mathbf{L}[t]) = \frac{1}{t}\sum_{\tau=1}^{t}\{\nabla_{\mathbf{L}} f_\tau(\mathbf{L}[\tau-1], \mathbf{q}[\tau]) + \alpha_\tau(\mathbf{L} - \mathbf{L}[\tau-1])\}.$$

After defining $\bar{\alpha}_t := \sum_{\tau=1}^{t}\alpha_\tau$, the first-order optimality condition leads to the recursion

$$\begin{aligned}\mathbf{L}[t] &= \frac{1}{\bar{\alpha}_t}\sum_{\tau=1}^{t}\alpha_\tau\left(\mathbf{L}[\tau-1] - \alpha_\tau^{-1}\nabla_{\mathbf{L}} f_\tau(\mathbf{L}[\tau-1], \mathbf{q}[\tau])\right)\\ &= \frac{1}{\bar{\alpha}_t}\underbrace{\sum_{\tau=1}^{t-1}\alpha_\tau\left(\mathbf{L}[\tau-1] - \alpha_\tau^{-1}\nabla_{\mathbf{L}} f_\tau(\mathbf{L}[\tau-1], \mathbf{q}[\tau])\right)}_{:=\bar{\alpha}_{t-1}\mathbf{L}[t-1]}\\ &\quad + \frac{\alpha_t}{\bar{\alpha}_t}\left(\mathbf{L}[t-1] - \alpha_t^{-1}\nabla_{\mathbf{L}} f_t(\mathbf{L}[t-1], \mathbf{q}[t])\right)\\ &= \mathbf{L}[t-1] - \frac{1}{\bar{\alpha}_t}\nabla_{\mathbf{L}} f_t(\mathbf{L}[t-1], \mathbf{q}[t]). \end{aligned} \quad (13)$$

Upon choosing the step size sequence $(\mu[t])^{-1} := \bar{\alpha}_t^{-1}$, the recursion in (8) readily follows.

Now it only remains to verify that the main steps of the proof outlined under (S1) and (S2) in Section IV-A, carry over for the average approximate cost $\tilde{C}_t$. Under (A1)–(A3) and thanks to the approximation tightness of $\tilde{f}_t$ as reflected through (i)–(iii), one can follow the same arguments in the proof of Proposition 1 (see also [30, Lemma 3]) to show that $\{\tilde{C}_t(\mathbf{L}[t])\}$ is a quasi-martingale sequence, and $\lim_{t\to\infty}(\tilde{C}_t(\mathbf{L}[t]) - C_t(\mathbf{L}[t])) = 0$. Moreover, assuming the sequence $\{\alpha_t\}$ is bounded and under the compactness assumption (A3), the quadratic function $\tilde{f}_t$ fulfills the required regularity conditions [30, Lemma 1] so that (S2) holds true. All in all, the SGD algorithm is convergent as formalized in the following claim.

Proposition 2: Under (A1)–(A3) and for $\lambda_t = \lambda$, $\forall t$, if $\mu[t] := \sum_{\tau=1}^{t}\alpha_\tau \geq ct$ for some constant $c > 0$ and $c' \geq \alpha_t \geq \|(\mathbf{q}[t]\mathbf{q}'[t]) \otimes \mathbf{\Omega}_t + \lambda/2t\mathbf{I}_{P\rho}\|, \forall t$ hold, the subspace iterates (8) satisfy $\lim_{t\to\infty}\nabla C_t(\mathbf{L}[t]) = \mathbf{0}_{P\times\rho}$ a.s., i.e., $\{\mathbf{L}[t]\}_{t=1}^{\infty}$ asymptotically coincides with the stationary points of the batch problem (P2).

Remark 3 (Convergence of Accelerated SGD): Paralleling the steps of the convergence proof for the SGD algorithm outline before, one may expect similar claims can be established for the *accelerated* variant tabulated under Algorithm 2. However, it is so far not clear how to construct an appropriate surrogate $\tilde{C}_t$ based on the available subspace updates $\{\mathbf{L}[t]\}$, whose minimizer coincides with the extrapolated estimates $\tilde{\mathbf{L}}[t]$. Recently, a variation of the accelerated SGD algorithm was put forth in [35], which could be applicable to the subspace tracking problem

studied in this paper. Adopting a different proof technique, the algorithm of [35] is shown convergent, and this methodology could be instrumental in formalizing the convergence of Algorithm 2 as well. However, this goes beyond the scope of this paper.

C. Optimality

Beyond convergence to stationary points of (P2), one may ponder whether the online estimator offers performance guarantees of the batch nuclear-norm regularized estimator (P1), for which stable/exact recovery results are well documented e.g., in [11], [12]. Specifically, given the learned subspace $\bar{\mathbf{L}}[t]$ and the corresponding $\bar{\mathbf{Q}}[t]$ [obtained via (3)] over a time window of size t, is $\{\hat{\mathbf{X}}[t] := \bar{\mathbf{L}}[t]\bar{\mathbf{Q}}'[t]\}$ an optimal solution of (P1) as $t \to \infty$? This in turn requires asymptotic analysis of the optimality conditions for (P1) and (P2), and a positive answer is established in the next proposition whose proof is deferred to the Appendix. Additionally, numerical tests in Section VI indicate that Algorithm 1 attains the performance of (P1) after a modest number of iterations.

Proposition 3: Consider the subspace iterates $\{\mathbf{L}[t]\}$ generated by either Algorithm 1 (with $\theta = 1$), or Algorithm 2. If there exists a subsequence $\{\mathbf{L}[t_k], \mathbf{Q}[t_k]\}$ for which (c1) $\lim_{k\to\infty} \nabla C_{t_k}(\mathbf{L}[t_k]) = \mathbf{0}_{P\times\rho}$ a.s., and (c2) $\frac{1}{\sqrt{t_k}}\sigma_{\max}[\mathcal{P}_{\Omega_{t_k}}(\mathbf{Y}_{t_k} - \mathbf{L}[t_k]\mathbf{Q}'[t_k])] \leq \frac{\lambda_{t_k}}{\sqrt{t_k}}$ hold, then the sequence $\{\mathbf{X}[k] = \mathbf{L}[t_k]\mathbf{Q}'[t_k]\}$ satisfies the optimality conditions for (P1) [normalized by t_k] as $k \to \infty$ a.s.

Regarding condition (c1), even though it holds for a time invariant rank-controlling parameter λ as per Proposition 1, numerical tests indicate that it still holds true for the time-varying case (e.g., when λ_t is chosen as suggested in Remark 2). Under (A2) and (A3) one has $\sigma_{\max}[\mathcal{P}_{\Omega_t}(\mathbf{Y}_t - \mathbf{L}[t]\mathbf{Q}'[t])] \approx \mathcal{O}(\sqrt{t})$, which implies that the quantity on the left-hand side of (c2) cannot grow unbounded. Moreover, upon choosing $\lambda_t \approx \mathcal{O}(\sqrt{t})$ as per Remark 2 the term in the right-hand side of (c2) will not vanish, which suggests that the qualification condition can indeed be satisfied.

V. Online Tensor Decomposition and Imputation

As modern and massive datasets become increasingly complex and heterogeneous, in many situations one encounters data structures indexed by three or more variables giving rise to a tensor, instead of just two variables as in the matrix settings studied so far. A few examples of time-indexed, incomplete tensor data include [2]: (i) dynamic social networks represented through a temporal sequence of network adjacency matrices, meaning a data cube with entries indicating whether e.g., two agents coauthor a paper or exchange emails during time interval t, while it may be the case that not all pairwise interactions can be sampled; (ii) Electroencephalogram (EEG) data, where each signal from an electrode can be represented as a time-frequency matrix; thus, data from multiple channels is three-dimensional (temporal, spectral, and spatial) and may be incomplete if electrodes become loose or disconnected for a period of time; and (iii) multidimensional nuclear magnetic resonance (NMR) analysis, where missing data are encountered when sparse sampling is used in order to reduce the experimental time.

Many applications in the aforementioned domains aim at capturing the underlying latent structure of the data, which calls for high-order factorizations even in the presence of missing data [2], [6]. 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 PARAFAC model described next. In the sequel, the discussion will be focused on three-way tensors for simplicity in exposition, but extensions to higher-way arrays are possible.

A. Low-Rank Tensors and the PARAFAC Decomposition

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 $\mathbf{a}(m)\mathbf{b}(n)\mathbf{c}(t)$. Note that this comprises a generalization of the matrix case, where $\mathbf{a}\circ\mathbf{b} = \mathbf{a}\mathbf{b}'$ is a rank-one matrix. The rank of a tensor $\underline{\mathbf{X}}$ is defined as the minimum number of outer products required to synthesize $\underline{\mathbf{X}}$.

The PARAFAC model 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

$$\underline{\mathbf{X}} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r. \tag{14}$$

When the decomposition is exact, (14) is the PARAFAC decomposition of $\underline{\mathbf{X}}$. 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 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-complete. For a survey of algorithmic approaches to obtain approximate PARAFAC decompositions, the reader is referred to [24].

With reference to (14), 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 $\mathbf{X}_t(m, n) = \underline{\mathbf{X}}(m, n, t)$. The following compact matrix form of the PARAFAC decomposition in terms of slice factorizations will be used in the sequel

$$\mathbf{X}_t = \mathbf{A}\mathrm{diag}(\boldsymbol{\gamma}_t)\mathbf{B}' = \sum_{r=1}^{R} \boldsymbol{\gamma}_t(r)\mathbf{a}_r\mathbf{b}_r', \quad t = 1, \ldots, T \tag{15}$$

where $\boldsymbol{\xi}_t'$ 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. (14)], and one can likewise write $\mathbf{X}_m = \mathbf{B}\mathrm{diag}(\boldsymbol{\alpha}_m)\mathbf{C}'$, or, $\mathbf{X}_n = \mathbf{C}\mathrm{diag}(\boldsymbol{\beta}_n)\mathbf{A}'$ 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 matrices $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ are unique up to a common column permutation and scaling (meaning PARAFAC is identifiable); see e.g. [8], [25].

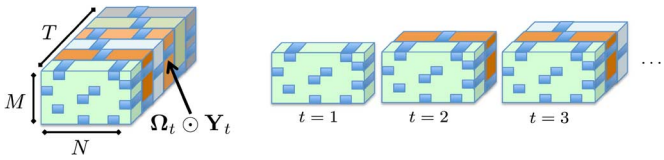

Fig. 2. Tensor data with missing entries. (Left) Batch data, and slice $\boldsymbol{\Omega}_t \odot \mathbf{Y}_t$ along the time (tube) dimension. (Right) Streaming data, where slices $\boldsymbol{\Omega}_t \odot \mathbf{Y}_t$ become available for $t = 1, 2, \ldots$.

Building on the intuition 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 (14) 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 [19], [27], [43] for *batch* tensor completion. The algorithm in the following section offers a novel approach for decomposing and imputing low-rank *streaming* tensors with *missing* data.

B. Algorithm for Streaming Tensor Data

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 $\mathcal{P}_{\underline{\boldsymbol{\Omega}}}(\underline{\mathbf{Y}}) = \underline{\boldsymbol{\Omega}} \odot \underline{\mathbf{Y}}$; see also Fig. 2 (left). Generalizing the nuclear-norm regularization technique in (P1) from low-rank matrix 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 [24]. Interestingly, it was argued in [6] that the Frobenius-norm regularization outlined in Section II-B offers a viable option for *batch* low-rank tensor completion under the PARAFAC model, by solving [cf. (P2) and (15)]

$$(\text{P4}) \min_{\{\mathbf{X},\mathbf{A},\mathbf{B},\mathbf{C}\}} \frac{1}{2}\|\underline{\boldsymbol{\Omega}} \odot (\underline{\mathbf{Y}} - \underline{\mathbf{X}})\|_F^2 + \frac{\lambda}{2}\left(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2\right)$$
$$\text{s. to } \mathbf{X}_t = \mathbf{A}\text{diag}(\boldsymbol{\gamma}_t)\mathbf{B}', \quad t = 1, 2 \ldots, T.$$

The regularizer in (P4) provably encourages low-rank tensor decompositions, in fact with controllable rank by tuning the parameter λ [6]. Note that similar to the matrix case there is no need for the true rank R in (P4). In fact, any upperbound $\hat{R} \geq R$ can be used for the column size of the sought matrix variables $\mathbf{A}, \mathbf{B}, \mathbf{C}$ as long as λ is tuned appropriately.

Consider now a real-time setting where the incomplete tensor slices $\boldsymbol{\Omega}_t \odot \mathbf{Y}_t$ are acquired sequentially over time $t = 1, 2, \ldots$ [i.e., streaming data as depicted in Fig. 2 (right)]. Leveraging the batch formulation (P4) one can naturally broaden the subspace tracking framework in Section III, to devise adaptive algorithms capable of factorizing tensors 'on the fly'. To this end, one can estimate the PARAFAC model factors $\{\mathbf{A}[t], \mathbf{B}[t], \mathbf{C}[t]\}$ as the minimizers of the following EWLS cost [cf. (P3)]

$$(\text{P5}) \min_{\{\mathbf{A},\mathbf{B},\mathbf{C}\}} \frac{1}{2}\sum_{\tau=1}^{t} \theta^{t-\tau}\Big[\|\boldsymbol{\Omega}_\tau \odot (\mathbf{Y}_\tau - \mathbf{A}\text{diag}(\boldsymbol{\gamma}_\tau)\mathbf{B}')\|_F^2 + \bar{\lambda}_t\left(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2\right) + \lambda_t\|\boldsymbol{\gamma}_\tau\|^2\Big].$$

Once more, the normalization $\bar{\lambda} := \lambda_t / \sum_{\tau=1}^{t}\theta^{t-\tau}$ ensures that for the infinite memory setting ($\theta = 1$) and $t = T$, (P5) coincides with the batch estimator (P4).

Paralleling the algorithmic construction steps adopted for the matrix case, upon defining the counterpart of $g_t(\mathbf{L}, \mathbf{q})$ corresponding to (P5) as

$$\bar{g}_t(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}) := \frac{1}{2}\|\boldsymbol{\Omega}_t \odot (\mathbf{Y}_t - \mathbf{A}\text{diag}(\boldsymbol{\gamma})\mathbf{B}')\|_F^2 + \frac{\lambda_t}{2}\|\boldsymbol{\gamma}\|^2 \tag{16}$$

the minimizer $\boldsymbol{\xi}_t = \arg\min_{\boldsymbol{\xi}} \bar{g}_t(\mathbf{A}, \mathbf{B}, \boldsymbol{\xi})$ is readily obtained in closed form, namely

$$\boldsymbol{\gamma}_t = \left[\lambda\mathbf{I}_R + \sum_{(m,n)\in\Omega_t} (\boldsymbol{\alpha}_m \odot \boldsymbol{\beta}_n)(\boldsymbol{\alpha}_m \odot \boldsymbol{\beta}_n)'\right]^{-1} \times \sum_{(m,n)\in\Omega_t} \underline{\mathbf{Y}}_t(m,n)(\boldsymbol{\alpha}_m \odot \boldsymbol{\beta}_n). \tag{17}$$

Accordingly, the factor matrices $\{\mathbf{A}, \mathbf{B}\}$ that can be interpreted as bases for the fiber subspace are the minimizers of the cost function

$$\bar{C}_t(\mathbf{A}, \mathbf{B}) := \sum_{\tau=1}^{t} \theta^{t-\tau}\bar{g}_\tau(\mathbf{A}, \mathbf{B}, \boldsymbol{\gamma}_\tau) + \frac{\bar{\lambda}_t}{2}\left(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2\right). \tag{18}$$

Note that $\boldsymbol{\xi}_t := \boldsymbol{\xi}_t(\mathbf{A}, \mathbf{B})$ as per (17), so minimizing $\bar{C}_t(\mathbf{A}, \mathbf{B})$ becomes increasingly complex computationally as t grows.

Remark 4 (Challenges Facing a Second-Order Algorithm): As discussed in Section III, one can approximate $\bar{g}_t(\mathbf{A}, \mathbf{B}, \boldsymbol{\xi}_t)$ with the upper bound $\bar{g}_t(\mathbf{A}, \mathbf{B}, \boldsymbol{\xi}_t(\mathbf{A}[t-1], \mathbf{B}[t-1]))$ to develop a second-order algorithm that circumvents the aforementioned increasing complexity roadblock. Unlike the matrix case however, (18) 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 is challenging. One could instead think of carrying out alternating minimizations with respect to each of the three factors per time instant t, namely updating: (i) $\boldsymbol{\xi}[t]$ first, given $\{\mathbf{A}[t-1], \mathbf{B}[t-1]\}$; (ii) then $\mathbf{B}[t]$ given $\mathbf{A}[t-1]$ and $\{\boldsymbol{\xi}[\tau]\}_{\tau=1}^{t}$; and (iii) finally $\mathbf{A}[t]$ with fixed $\mathbf{B}[t]$ and $\{\boldsymbol{\xi}[\tau]\}_{\tau=1}^{t}$. 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) SGD alternative. Mimicking the steps in Section III-B, let $\bar{f}_t(\mathbf{A}, \mathbf{B}) := \bar{g}_t(\mathbf{A}, \mathbf{B}, \boldsymbol{\xi}[t]) + \frac{\lambda_t}{2t}(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2)$ denote the t-th summand

in (18), for $t = 1, 2, \ldots$ and $\theta = 1$. The factor matrices $\mathcal{L}[t] := \{\mathbf{A}[t], \mathbf{B}[t]\}$ are obtained via the SGD iteration

$$\begin{aligned}\mathcal{L}[t] &= \arg\min_{\mathcal{L}} \bar{Q}_{\bar{\mu}[t],t}\left(\mathcal{L}, \mathcal{L}[t-1]\right) \\ &= \mathcal{L}[t-1] - \left(\bar{\mu}[t]\right)^{-1} \nabla \bar{f}_t\left(\mathcal{L}[t-1]\right) \end{aligned} \tag{19}$$

with the stepsize $(\bar{\mu}[t])^{-1}$, and $\bar{Q}_{\mu,t}(\mathcal{L}_1, \mathcal{L}_2) := \bar{f}_t(\mathcal{L}_2) + \langle \mathcal{L}_1 - \mathcal{L}_2, \nabla \bar{f}_t(\mathcal{L}_2)\rangle + \frac{\mu}{2}\|\mathcal{L}_1 - \mathcal{L}_2\|^2$. It is instructive to recognize that the quadratic surrogate $\bar{Q}_{\bar{\mu}[t],t}$ has the following properties: (i) it majorizes $\bar{f}_t(\mathcal{L})$, namely $\bar{f}_t(\mathcal{L}) \leq \bar{Q}_{\bar{\mu},t}(\mathcal{L}, \mathcal{L}[t-1]), \forall \mathcal{L}$; while it is locally tight meaning that (ii) $\bar{f}_t(\mathcal{L}[t-1]) = \bar{Q}_{\bar{\mu}[t],t}(\mathcal{L}[t-1], \mathcal{L}[t-1])$, and (iii) $\nabla \bar{f}_t(\mathcal{L}[t-1]) = \nabla \bar{Q}_{\bar{\mu}[t],t}(\mathcal{L}[t-1], \mathcal{L}[t-1])$. Accordingly, the minimizer of $\bar{Q}_{\bar{\mu}[t],t}(\mathcal{L}, \mathcal{L}[t-1])$ amounts to a correction along the negative gradient $\nabla \bar{f}_t(\mathcal{L}[t-1])$, with stepsize $(\bar{\mu}[t])^{-1}$ [cf. (19)].

Algorithm 3: Online SGD algorithm for tensor decomposition and imputation

input $\{\mathbf{Y}_t, \mathbf{\Omega}_t\}_{t=1}^{\infty}$,, $\{\bar{\mu}[t]\}_{t=1}^{\infty}$, $\hat{R}$, and λ_t.

initialize $\{\mathbf{A}[0], \mathbf{B}[0]\}$ at random, and $\bar{\mu}[0] > 0$.

for $t = 0, 1, 2, \ldots$ **do**

$\mathbf{A}'[t] := [\boldsymbol{\alpha}_1[t], \ldots, \boldsymbol{\alpha}_M[t]]$ and $\mathbf{B}'[t] := [\boldsymbol{\beta}_1[t], \ldots, \boldsymbol{\beta}_N[t]]$

$\boldsymbol{\xi}[t] = [\lambda \mathbf{I}_R + \sum_{(m,n)\in\Omega_t} (\boldsymbol{\alpha}_m[t] \odot \boldsymbol{\beta}_n[t])(\boldsymbol{\alpha}_m[t] \odot \boldsymbol{\beta}_n[t])']^{-1} \times \sum_{(m,n)\in\Omega_t} \mathbf{Y}_t(m,n)(\boldsymbol{\alpha}_m[t] \odot \boldsymbol{\beta}_n[t])$

$\mathbf{A}[t+1] = (1 - \frac{\lambda_t}{t\bar{\mu}[t]})\mathbf{A}[t] + \frac{1}{\bar{\mu}[t]}[\mathbf{\Omega}_t \odot (\mathbf{Y}_t - \mathbf{A}[t]\mathrm{diag}(\boldsymbol{\xi}[t])\mathbf{B}'[t])]\mathbf{B}[t]\mathrm{diag}(\boldsymbol{\xi}[t])$

$\mathbf{B}[t+1] = (1 - \frac{\lambda_t}{t\bar{\mu}[t]})\mathbf{B}[t] + \frac{1}{\bar{\mu}[t]}[\mathbf{\Omega}_t \odot (\mathbf{Y}_t - \mathbf{A}[t]\mathrm{diag}(\boldsymbol{\xi}[t])\mathbf{B}'[t])]'\mathbf{A}[t]\mathrm{diag}(\boldsymbol{\xi}[t])$

end for

$\hat{\mathbf{X}}[t] := \mathbf{A}[t]\mathrm{diag}(\boldsymbol{\xi}[t])\mathbf{B}'[t]$.

Putting together (17) and (19), while observing that the components of $\nabla \bar{f}_t(\mathcal{L})$ are expressible as

$$\begin{aligned}\nabla_{\mathbf{A}} \bar{f}_t(\mathbf{A}, \mathbf{B}) = &-[\mathbf{\Omega}_t \odot (\mathbf{Y}_t - \mathbf{A}\mathrm{diag}\left(\boldsymbol{\gamma}[t]\right)\mathbf{B}')]\,\mathbf{B}\mathrm{diag}\left(\boldsymbol{\gamma}[t]\right) \\ &+ \frac{\lambda_t}{t}\mathbf{A} \end{aligned} \tag{20}$$

$$\begin{aligned}\nabla_{\mathbf{B}} \bar{f}_t(\mathbf{A}, \mathbf{B}) = &-[\mathbf{\Omega}_t \odot (\mathbf{Y}_t - \mathbf{A}\mathrm{diag}\left(\boldsymbol{\gamma}[t]\right)\mathbf{B}')]'\mathbf{A}\mathrm{diag}\left(\boldsymbol{\gamma}[t]\right) \\ &+ \frac{\lambda_t}{t}\mathbf{B} \end{aligned} \tag{21}$$

one arrives at the SGD iterations tabulated under Algorithm 3. Close examination of the recursions reveals that updating $\mathbf{A}[t]$ and $\mathbf{B}[t]$ demands $\mathcal{O}(|\Omega_t|\hat{R})$ operations, while updating $\boldsymbol{\xi}[t]$ incurs a cost of $\mathcal{O}(|\Omega_t|\hat{R}^2)$. The overall complexity per iteration is thus $\mathcal{O}(|\Omega_t|\hat{R}^2)$.

Remark 5 (Forming the Tensor Decomposition 'on-the-Fly'): In a stationary setting the low-rank tensor decomposition can be accomplished after the tensor subspace matrices are learned; that is, when the sequences $\{\mathbf{A}[t], \mathbf{B}[t]\}$ converge to the limit points, say $\{\bar{\mathbf{A}}, \bar{\mathbf{B}}\}$. The remaining factor $\bar{\mathbf{C}} := [\bar{\boldsymbol{\gamma}}_1', \ldots, \bar{\boldsymbol{\gamma}}_T']'$ is then obtained by solving $\bar{\boldsymbol{\gamma}}_t = \arg\min_{\boldsymbol{\gamma}} \bar{g}_t(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \boldsymbol{\gamma})$ for the corresponding tensor slice $\mathcal{P}_{\Omega_t}(\mathbf{X}_t)$, which yields a simple closed-form solution as in (17). This requires revisiting the past tensor slices. The factors $\{\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}\}$ then form a low-rank approximation of the entire tensor $\underline{\mathbf{X}} \in \mathbb{R}^{M\times N\times T}$. Note also that after the tensor subspace is learned say at time $t' \leq T$, e.g., from some initial training data, the projection coefficients $\bar{\boldsymbol{\gamma}}_t$ can be calculated 'on-the-fly' for $t \geq t'$; thus, Algorithm 3 offers a decomposition of the tensor containing slices t' to $t > t'$ 'on-the-fly'.

Convergence of Algorithm 3 is formalized in the next proposition, and can be established using similar arguments as in the matrix case detailed in Section IV-B. Furthermore, empirical observations in Section VI suggest that the convergence rate can be linear.

Proposition 4: Suppose slices $\{\mathbf{\Omega}_t \odot \mathbf{Y}_t\}_{t=1}^{\infty}$ and the corresponding sampling sets $\{\Omega_t\}_{t=1}^{\infty}$ are i.i.d., and $\theta = 1$ while $\lambda_t = \lambda, \forall t$. If (c1) $\{\mathcal{L}[t]\}_{t=1}^{\infty}$ live in a compact set, and (c2) the step-size sequence $\{(\bar{\mu}[t])^{-1}\}$ satisfies $\bar{\mu}[t] := \sum_{\tau=1}^{t} \tilde{\alpha}[\tau] \geq ct, \forall t$ for some $c > 0$, where (c3) $c' \geq \tilde{\alpha}[t] \geq \sigma_{\max}(\nabla^2 \bar{f}_t(\mathcal{L}[t-1])), \forall t$ for some $c' > 0$, then $\lim_{t\to\infty} \nabla C_t(\mathcal{L}[t]) = \mathbf{0}$, a.s.; i.e., the tensor subspace iterates $\{\mathcal{L}[t]\}$ asymptotically coincide with the stationary points of (P4).

Before closing this section, a remark is in order.

Remark 6 (Tucker Multilinear Rank): If the goal is to impute tensors with missing data one could alternatively leverage that the tensor exhibits low Tucker multilinear rank [19] (which is different from tensor rank). Interestingly, convex surrogates exist for *batch* multilinear rank minimization; see e.g., [19], [43]. Different from [19], [43], the nonconvex PARAFAC formulation (P4) is separable across time, thus facilitating online implementation. In addition, the PARAFAC model offers a low-rank decomposition of the tensor as a byproduct, which can be useful to unravel latent structure in the data.

VI. Numerical Tests

The convergence and effectiveness of the proposed algorithms is assessed in this section via computer simulations. Both synthetic and real data tests are carried out in the sequel.

A. Synthetic Matrix Data Tests

The signal $\mathbf{x}_t = \mathbf{U}\mathbf{w}_t$ is generated from the low-dimensional subspace $\mathbf{U} \in \mathbb{R}^{P\times r}$, with Gaussian i.i.d. entries $u_{p,i} \sim \mathcal{N}(0, 1/P)$, and projection coefficients $w_{i,t} \sim \mathcal{N}(0,1)$. The additive noise $v_{i,t} \sim \mathcal{N}(0, \sigma^2)$ is i.i.d., and to simulate the misses per time t, the sampling vector $\boldsymbol{\omega}_t \in \{0,1\}^P$ is formed, where each entry is a Bernoulli random variable, taking value one with probability (w.p.) π, and zero w.p. $1 - \pi$, which implies that $(1-\pi) \times 100\%$ entries are missing. The observations at time t are generated as $\mathcal{P}_{\omega_t}(\mathbf{y}_t) = \boldsymbol{\omega}_t \odot (\mathbf{x}_t + \mathbf{v}_t)$.

Throughout, fix $r = 5$ and $\rho = 10$, while different values of π and σ are examined. The time evolution of the average cost $C_t(\mathbf{L}[t])$ in (9) for various amounts of misses and noise strengths is depicted in Fig. 3(a) [$\theta = 1$]. For validation purposes, the optimal cost [normalized by the window size t] of the *batch* estimator (P1) is also shown. It is apparent that $C_t(\mathbf{L}[t])$ converges to the optimal objective of the nuclear-norm regularized problem (P1), corroborating that Algorithm 1 attains the performance of (P1) in the long run. This observation in addition to the low cost of Algorithm 1 [$\mathcal{O}(|\omega_t|\rho^2)$ per iteration]

TABLE I
COMPUTATIONAL COMPLEXITY PER ITERATION

GROUSE ($\kappa=\rho$)	PETRELS ($\kappa=\rho$)	Alg. 1 ($\theta\in(0,1)$)	Alg. 1 ($\theta=1$)	Alg. 2	Alg. 2 ($k[t]=1$)
$\mathcal{O}(P\rho+\|\omega_t\|\rho^2)$	$\mathcal{O}(\|\omega_t\|\rho^2)$	$\mathcal{O}(P\rho^3)$	$\mathcal{O}(\|\omega_t\|\rho^2)$	$\mathcal{O}(P\rho+\|\omega_t\|\rho^2)$	$\mathcal{O}(\|\omega_t\|\rho^2)$

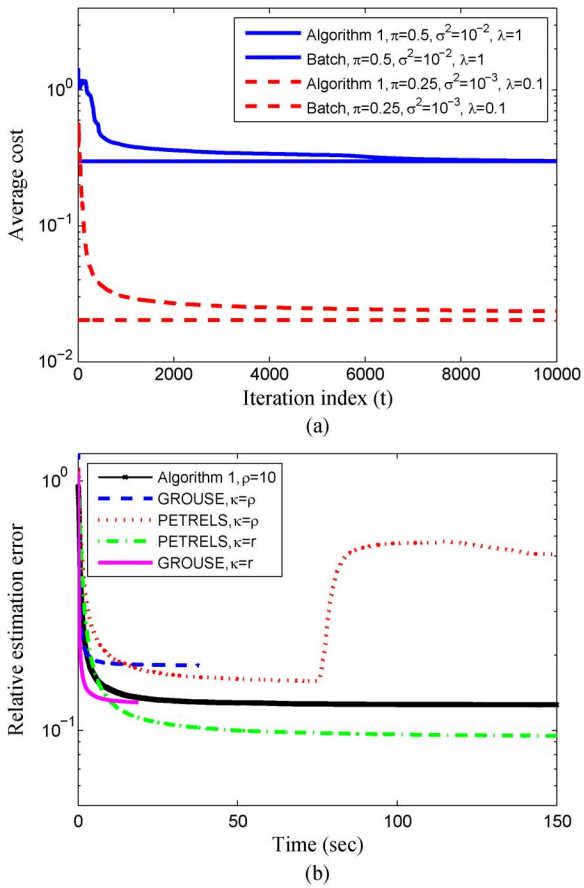

Fig. 3. Performance of Algorithm 1. (a) Evolution of the average cost $C_t(\mathbf{L}[t])$ versus the batch counterpart. (b) Relative estimation error for different schemes when $\pi=0.25$ and $\sigma^2=10^{-3}$.

suggest it as a viable alternative for solving large-scale matrix completion problems.

Next, Algorithm 1 is compared with other state-of-the-art subspace trackers, including PETRELS [13] and GROUSE [3], discussed in Section I. In essence, these algorithms need the dimension of the underlying subspace, say κ, to be known/estimated *a priori*. Fix $\lambda=0.1$, $\theta=0.99$, and introduce an abrupt subspace change at time $t=10^4$ to assess the tracking capability of the algorithms. The figure of merit depicted in Fig. 3(b) is the running-average estimation error $e_x[t]:=\frac{1}{t}\sum_{i=1}^{t}\|\hat{\mathbf{x}}_i-\mathbf{x}_i\|_2/\|\mathbf{x}_i\|_2$. It is first observed that upon choosing identical subspace dimension $\kappa=\rho$ for all three schemes, Algorithm 1 attains a better estimation accuracy, where a constant step size $(\mu[t])^{-1}=0.1$ was adopted for PETRELS and GROUSE. Albeit PETRELS performs well when the true rank is known, namely $\kappa=r$, if one overestimates the rank the algorithm exhibits erratic behaviors for large fraction 75% of missing observations. As expected, for the ideal choice of $\kappa=r$, all three schemes achieve nearly identical estimation accuracy. The smaller error exhibited by PETRELS relative to Algorithm 1 may pertain to the suboptimum selection of λ. Nonetheless, for large amount of misses both GROUSE and PETRELS are numerically unstable as the LS problems to obtain the projection coefficients $\mathbf{q}_t$ become ill-conditioned, whereas the ridge-regression type regularization terms in (P3) render Algorithm 1 numerically stable. The price paid by Algorithm 1 is however in terms of higher computational complexity per iteration, as seen in Table I which compares the complexity of various algorithms.

B. Real Matrix Data Tests

Accurate estimation of origin-to-destination (OD) flow traffic in the backbone of large-scale Internet Protocol (IP) networks is of paramount importance for proactive network security and management tasks [23]. Several experimental studies have demonstrated that OD flow traffic exhibits a low-intrinsic dimensionality, mainly due to common temporal patterns across OD flows, and periodic trends across time [26]. However, due to the massive number of OD pairs and the high volume of traffic, measuring the traffic of all possible OD flows is impossible for all practical purposes [23], [26]. Only the traffic level for a small fraction of OD flows can be measured via the NetFlow protocol [26].

Here, aggregate OD-flow traffic is collected from the operation of the Internet-2 network (Internet backbone across USA) during December 8–28, 2003 containing 121 OD pairs [1]. The measured OD flows contain spikes (anomalies), which are discarded to end up with a anomaly-free data stream $\{\mathbf{y}_t\}\in\mathbb{R}^{121}$. The detailed description of the considered dataset can be found in [30]. A subset of entries of $\mathbf{y}_t$ are then picked randomly with probability π to yield the input of Algorithm 1. The evolution of the running-average traffic estimation error $(e_x[t])$ is depicted in Fig. 4(a) for different schemes and under various amounts of missing data. Evidently, Algorithm 1 outperforms the competing alternatives when λ_t is tuned adaptively as per Remark 2 for $\sigma^2=0.1$. When only 25% of the total OD flows are sampled by Netflow, Fig. 4(b) depicts how Algorithm 1 accurately tracks three representative OD flows.

C. Synthetic Tensor Data Tests

To form the t-th 'ground truth' tensor slice $\mathbf{X}_t=\mathbf{A}\text{diag}(\boldsymbol{\xi}_t)\mathbf{B}'\in\mathbb{R}^{M\times N}$, the factors $\mathbf{A}$ and $\mathbf{B}$ are generated independently with Gaussian i.i.d. columns $\mathbf{a}_r\sim\mathcal{N}(\mathbf{0},\mathbf{I}_M)$ and $\mathbf{b}_r\sim\mathcal{N}(\mathbf{0},\mathbf{I}_N)$; likewise, the coefficients $\boldsymbol{\xi}_t\sim\mathcal{N}(0,\mathbf{I}_R)$. The sampling matrix $\boldsymbol{\Omega}_t$ also contains random Bernoulli entries taking value one w.p. π, and zero w.p. $1-\pi$. Gaussian noise is also considered with i.i.d. entries $\mathbf{V}_t(m,n)\sim\mathcal{N}(0,\sigma^2)$. Accordingly, the t-th acquired slice is formed as $\mathbf{Y}_t=\boldsymbol{\Omega}_t\odot(\mathbf{X}_t+\mathbf{V}_t)$. Fix $\sigma=10^{-3}$ and the true rank $R=5$, while different values of $M,N,\hat{R},\pi$ are examined. Performance of Algorithm 3 is tested for imputation of streaming tensor slices of relatively large size $M=N=10^3$, where a constant step size $(\bar{\mu}[t])^{-1}=10^{-2}$ is adopted. Various amounts of misses are examined, namely $1-\pi\in\{0.99,0.9,0.75\}$. Also, in accordance with the matrix completion setup select $\lambda=\sqrt{2MN\pi}\sigma$; see e.g.,

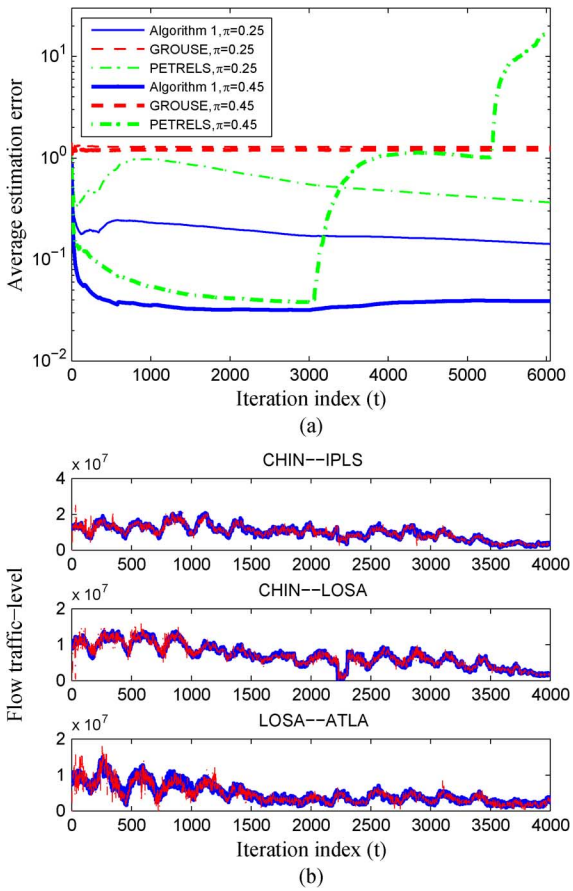

Fig. 4. Traffic estimation performance for Internet-2 data when $\kappa = \rho = 10$ and $\theta = 0.95$. (a) Average estimation error for $\pi = 0.25$ (thin line) and $\pi = 0.45$ (thick line). (b) Algorithm 1's estimated (dashed red) versus true (solid blue) OD flow traffic for 75% misses ($\pi = 0.25$).

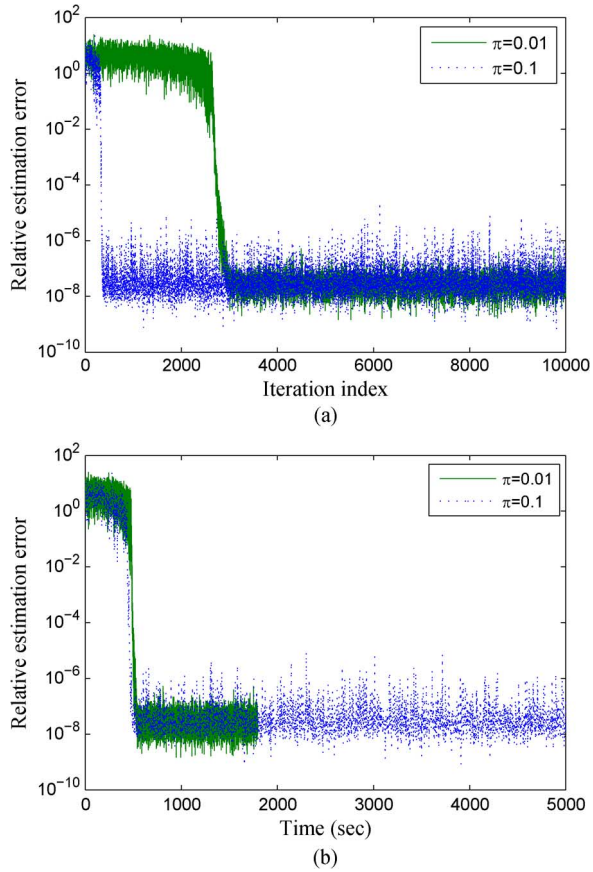

Fig. 5. Relative estimation error versus (a) iterations and (b) run-time under various amounts of misses for $M = N = 10^3$, $\sigma = 10^{-3}$ and $\hat{R} = 10$.

[12]. Fig. 5 depicts the evolution of the estimation error $e_x[t] := \|\mathbf{X}_t - \hat{\mathbf{X}}_t\|_F / \|\mathbf{X}_t\|_F$, where it is naturally seen that as more data become available the tensor subspace is learned faster. It is also apparent that after collecting sufficient amounts of data the estimation error decreases geometrically, where finally the estimate $\hat{\mathbf{X}}_t$ falls in the σ^2-neighborhood of the 'ground truth' slice $\mathbf{X}_t$. This observation suggests the linear convergence of Algorithm 3, and highlights the effectiveness of estimator (P4) in accurately reconstructing a large fraction of misses.

Here, Algorithm 3 is also adopted to decompose large-scale, dense tensors and hence find the factors $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}$. For $T = 10^4$ and for different slice sizes $M = N = 10^2$ and $M = N = 10^3$, the tensor may not even fit in main memory to apply batch solvers naively. After running Algorithm 3 instead, Table II reports the run-time under various amount of misses. One can see that smaller values of π lead to shorter run-times since one needs to carry out less computations per iteration [c.f. $\mathcal{O}(|\Omega_t|\hat{R}^2)$]. Note that the MATLAB codes for these experiments are by no means optimized, so further reduction in run-time is possible with a more meticulous implementation. Another observation is that for decomposition of low-rank tensors, it might be beneficial from a computational complexity standpoint to keep only a small subset of entries. Note that if instead of employing a higher-order decomposition one unfolds the tensor and resorts to the subspace tracking schemes developed in Section III for the sake of imputation, each basis vector entails 10^6 variables. On the other hand, using tensor models each basis (rank-one) matrix entails only 2×10^3 variables. Once again, for comparison purposes there is no alternative online scheme that imputes the missing tensor entries, and offers a PARAFAC tensor decomposition after learning the tensor subspace (see also Remark 5).

TABLE II
TENSOR DECOMPOSITION RUN-TIME (SECONDS) FOR VARIOUS SIZES AND AMOUNTS OF MISSES WHEN $\hat{R} = 10, T = 10^4$

$M = N$	$\pi = 0.01$	$\pi = 0.1$	$\pi = 0.25$
10^2	26	132	302
10^3	1.8×10^3	10^4	3×10^4

D. Real Tensor Data Tests

Two real tensor data tests are carried out next, in the context of cardiac MRI and network traffic monitoring applications.

Dynamic cardiac MRI. Cardiac MRI nowadays serves as a major imaging modality for noninvasive diagnosis of heart diseases in clinical practice [18]. However, quality of MRI images is degraded as a result of fast acquisition process which is mainly due to patient's breath-holding time. This may render some image pixels inaccurate or missing, and thus the acquired image only consists of 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.

Low-rank tensor completion is well motivated by the low-intrinsic dimensionality of cardiac MRI images [20].

Algorithm 3 is tested here using an *in vivo* MRI dataset comprising dynamic cardiac cine images of size 200×256 across 256 frames. The k-space data was acquired in full but retrospectively randomly undersampled by a factor of $1/\pi$. The 2-D k-space data per time slot were split into 20 patches of size 40×46, resulting in a moderate-sized tensor $\underline{\mathbf{X}}$ of size $40 \times 64 \times 5,120$. Batch imputation of dense tensors may be infeasible because of memory limitations. The online Algorithm 3 is however a viable alternative, which performs only $2560\pi\hat{R}^2$ operations on average per time step, and requires storing only $2560 + 64\hat{R}$ variables. The formed complex-valued tensor can be expressed as $\underline{\mathbf{X}} = \underline{\mathbf{X}}_R + i\underline{\mathbf{X}}_I$. Both the real and imaginary parts $\underline{\mathbf{X}}_R$ and $\underline{\mathbf{X}}_I$ exhibit low rank, and thus they are imputed separately in parallel.

The reconstructed image at the 146-th frame is compared against the gold standard (full acquisition) in Fig. 6 under various percentages of misses and postulated rank values. The proposed method is seen capable of accurately reconstructing the image from 25% available k-space samples, resulting in the accuracy $e_x[146] = 0.12$ when $\hat{R} = 100$. To further quantify the real-time performance Fig. 7 depicts the reconstruction error $e_x[t]$ against the run-time under different scenarios for misses and rank values. The same fixed step-size $(\bar{\mu}[t])^{-1} = 10^{-1}$ is adopted. It is seen that after around 100 seconds, the low-rank tensor subspace is learned and the reconstructed images reveal details of the cardiac snapshot.

Tracking network-traffic anomalies. In the backbone of large-scale IP networks, OD flows experience traffic volume anomalies due to e.g., equipment failures and cyberattacks which can congest the network [42]. Consider a network whose topology is represented by a directed graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{L}$ and $\mathcal{N}$ denote the set of links and nodes of cardinality $|\mathcal{L}| = L$ and $|\mathcal{N}| = N$, respectively. Upon associating the weight $w_{i,j} > 0, (i,j) \in E$ with the (i,j)-th link, G can be completely described by the weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$. For instance $w_{i,j}$ can represent the link loads as will be shown later. In the considered network, a set of OD traffic flows $\mathcal{F}$ with $|\mathcal{F}| = F$ traverses the links connecting OD pairs. Let $r_{\ell,f} \in [0,1]$ denote the fraction of f-th flow traffic at time t, say $x_{f,t}$, measured in e.g., packet counts, carried by link ℓ. The overall traffic carried by link ℓ is then the superposition of the flow rates routed through link ℓ, namely, $\sum_{f \in \mathcal{F}} r_{\ell,f} x_{\ell,f}$. It is not uncommon for some of OD flows to experience anomalies. If $o_{f,t}$ denotes the unknown traffic volume anomaly of flow f at time t, the measured link counts over link ℓ at time t are then given by

$$y_{\ell,t} = \sum_{f \in \mathcal{F}} r_{\ell,f}(x_{f,t} + o_{f,t}) + v_{\ell,t}, \ \ell \in \mathcal{L} \qquad (22)$$

where $v_{\ell,t}$ accounts for measurement errors and unmodeled dynamics. In practice, missing link counts are common due to e.g., packet losses, and thus per time only a small fraction of links (indexed by Ω_t) are measured. Note that only a small group of flows are anomalous, and the anomalies persist for short periods of time relative to the measurement horizon. This renders the anomaly vector $\mathbf{o}_t = [o_{1,t}, \ldots, o_{F,t}]' \in \mathbb{R}^F$ sparse.

In general, one can collect the partial link counts per time instant in a vector to form a (vector-valued) time-series, and subsequently apply the subspace tracking algorithms developed in e.g., [30] to unveil the anomalies in real time. Instead, to fully exploit the data structure induced by the network topology, the link counts per time t can be collected in an adjacency matrix $\mathbf{W}_t$, with $[\mathbf{W}_t]_{i,j} = y_{\ell,t}$ [edge (i,j) corresponds to link ℓ]. This matrix naturally constitutes the t-th slice of the tensor $\underline{\mathbf{Y}}$. Capitalizing on the spatiotemporal low-rank property of the nominal traffic as elaborated in Section VI-B, to discern the anomalies a low-rank ($\rho \ll R$) approximation of the incomplete tensor $\underline{\mathbf{\Omega}} \odot \underline{\mathbf{Y}}$ is obtained first in an online fashion using Algorithm 3. Anomalies are then unveiled from the residual of the approximation as elaborated next.

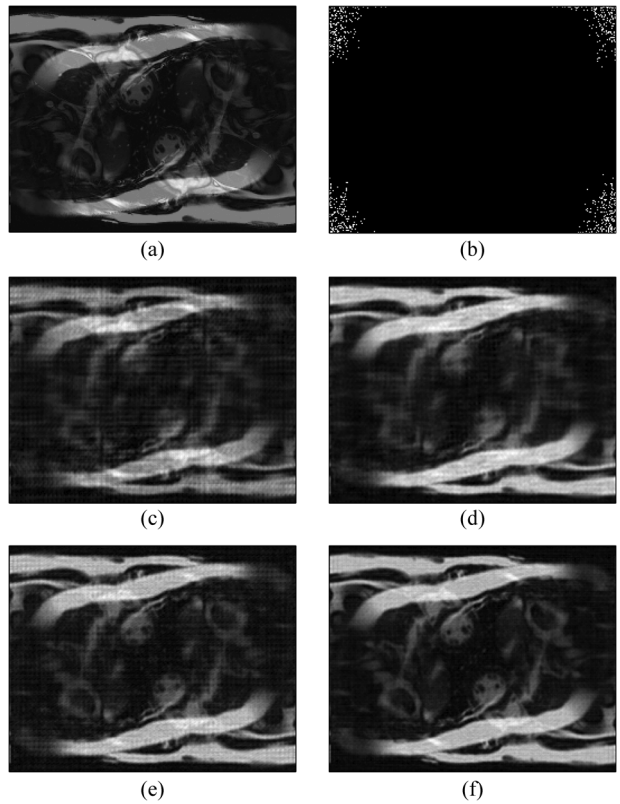

Fig. 6. Results of applying Algorithm 3 to *in vivo* MRI data. (a) Ground truth image at $t = 146$, (b) acquired k-space data undersampled randomly by a factor of 4, reconstructed image frame for $\hat{R} = 50$ with (c) 25% and (d) 40% available data; reconstructed image frame for $\hat{R} = 100$ with (e) 25% and (f) 40% available data.

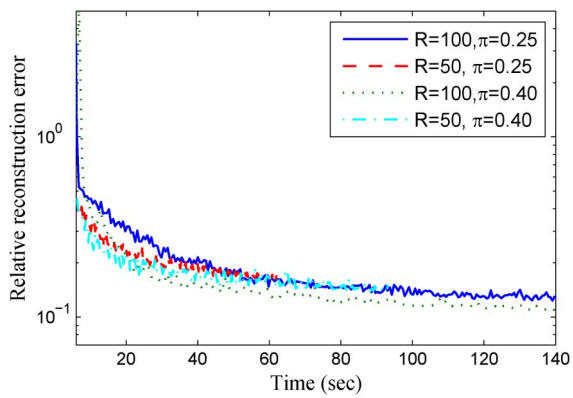

Fig. 7. Frame reconstruction error, averaged over 10 random realizations, versus run-time for various percentages of misses and rank levels.

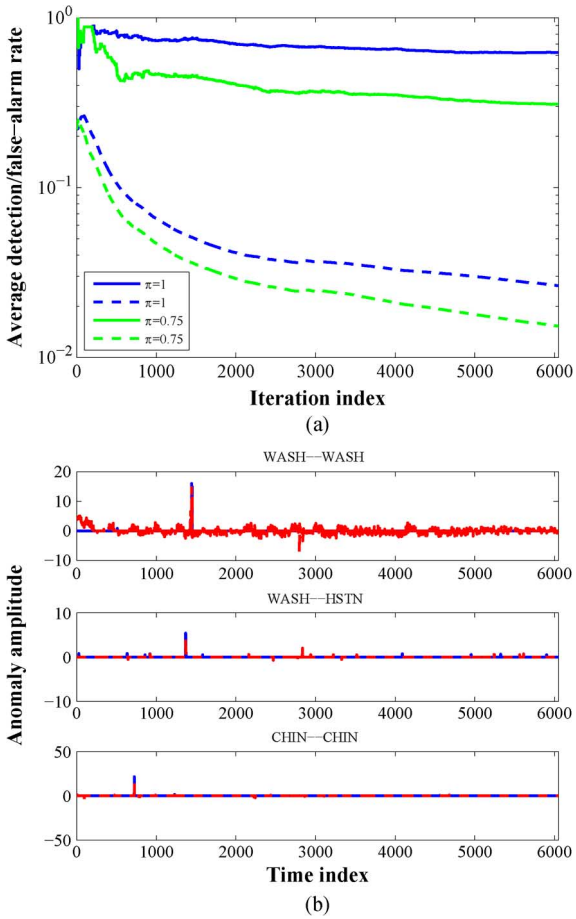

Fig. 8. Tracking Internet-2 traffic anomalies for $\hat{R} = 18$. (a) Evolution of average detection (solid) and false-alarm (dashed) rates. (b) Estimated (dashed) versus true (solid) anomalies for three representative OD flows when $\pi = 1$.

Let $\{\mathbf{A}[t], \mathbf{B}[t]\}$ denote the factors of the low-dimensional tensor subspace learned at time t, and $\hat{\mathbf{Y}}_t = \mathbf{A}[t]\text{diag}(\boldsymbol{\xi}[t])\mathbf{B}'[t] \in \mathbb{R}^{N\times N}$ the corresponding (imputed) low-rank approximation of the t-th slice. Form the residual matrix $\tilde{\mathbf{Y}}_t := \mathbf{Y}_t - \hat{\mathbf{Y}}_t$, which is (approximately) zero in the absence of anomalies. Collect the nonzero entries of $\tilde{\mathbf{Y}}_t$ into the vector $\tilde{\mathbf{y}}_t \in \mathbb{R}^L$, and the routing variables $r_{\ell,f}$ [cf. (22)] into matrix $\mathbf{R} \in \mathbb{R}^{L\times F}$. According to (22), one can postulate the linear regression model $\tilde{\mathbf{y}}_t = \mathbf{R}\mathbf{o}_t + \mathbf{v}_t$ to estimate the sparse anomaly vector $\mathbf{o}_t \in \mathbb{R}^F$ from the imputed link counts. An estimate of $\mathbf{o}_t$ can then be obtained via the least-absolute shrinkage and selection operator (LASSO)

$$\hat{\mathbf{o}}_t := \arg\min_{\mathbf{o}\in\mathbb{R}^F} \|\tilde{\mathbf{y}}_t - \mathbf{R}\mathbf{o}\|^2 + \lambda_o\|\mathbf{o}\|_1$$

where λ_o controls the sparsity in $\hat{\mathbf{o}}_t$ that is tantamount to the number of anomalies. In the absence of missing links counts, [22] has recently considered a batch tensor model of link traffic data and its Tucker decomposition to identify the anomalies.

The described tensor-based approach for network anomaly detection is tested on the Internet-2 traffic dataset described in Section VI-B, after fixing $\hat{R} = 18$. Each tensor slice $\mathbf{Y}_t \in \mathbb{R}^{11\times 11}$ contains only 41 nonzero entries corresponding to the physical links. Define the sets $S_O[t] := \{(i,j), i \in [L], j \in [t] : |o_j(i)| \geq \xi\}$ and $\bar{S}_O[t] := \{(i,j), i \in [L], j \in [t] : |o_j(i)| \leq \xi\}$ for some prescribed threshold ξ. To evaluate the detection performance, the adopted figures of merit are the running-average detection and false-alarm rates $P_{\text{D}} := |S_O \cap S_{\hat{O}}|/|S_O|$ and $P_{\text{FA}} := |\bar{S}_O \cap S_{\hat{O}}|/|\bar{S}_O|$, respectively. Fig. 8(a) depicts the time evolution of P_{D} and P_{FA} for $\pi = 1$ (fully available data), and $\pi = 0.75$. As more data becomes available, the traffic subspace is learned more accurately, and thus less false alarms are declared. For three representative OD flows, namely WASH-WASH, WASH--HSTN, and CHIN--CHIN, the true and estimated anomalies are depicted in Fig. 8(b). One can see that the significant anomalies are correctly picked in real-time by the proposed estimator. Note that the online formulation (P5) can even accommodate slowly-varying network topologies in the tensor model, which is desirable for monitoring the 'health state' of dynamic networks.

VII. Concluding Summary

This paper leverages recent advances in rank minimization for subspace tracking, and puts forth streaming algorithms for real-time, scalable decomposition of highly-incomplete multi-way Big Data arrays. For low-rank matrix data, a subspace tracker is developed based on an EWLS criterion regularized with the nuclear norm. Leveraging a separable characterization of nuclear-norm, both first- and second-order algorithms with complementary strengths are developed. In a stationary setting, the proposed algorithms asymptotically converge and provably offer the well-documented performance guarantees of the batch nuclear-norm regularized estimator. Under the same umbrella, an online algorithm is proposed for decomposing low-rank tensors with missing entries, which can accurately impute cardiac MRI images with up to 75% missing entries.

There are intriguing unanswered questions beyond the scope of this paper, but worth pursuing as future research. One such question pertains to the convergence analysis of the accelerated SGD algorithm either by following the adopted proof methodology, or, e.g., the alternative techniques used in [35]. Real-time incorporation of the spatiotemporal correlation between the unknowns by means of kernels or suitable statistical models is another important avenue to explore. Also, relaxing the qualification constraint for optimality is important for real-time applications in dynamic environments, where the learned subspace could conceivably change with time.

Appendix

Proof of Proposition 3: For the subspace sequence $\{\mathbf{L}[t]\}$ suppose that $\lim_{t\to\infty} \nabla C_t(\mathbf{L}[t]) = \mathbf{0}$. Then, due to the uniqueness of $\mathbf{q}[t] = \arg\min_{\mathbf{q}} g_t(\mathbf{L}[t], \mathbf{q})$, Danskin's Theorem [9] implies that

$$\lim_{t\to\infty} \frac{1}{t}\left(\mathcal{P}_{\Omega_t}\left(\mathbf{Y}_t - \mathbf{L}[t]\mathbf{Q}'[t]\right)\mathbf{Q}[t] - \lambda\mathbf{L}[t]\right) = \mathbf{0} \tag{23}$$

holds true almost surely, where $\mathbf{Q}[t] \in \mathbb{R}^{P\times t}$ satisfies

$$\mathbf{L}'[t]\mathcal{P}_{\Omega_t}\left(\mathbf{Y}_t - \mathbf{L}[t]\mathbf{Q}'[t]\right) - \lambda\mathbf{Q}'[t] = \mathbf{0}. \tag{24}$$

Consider now a subsequence $\{\mathbf{L}[t_k], \mathbf{Q}[t_k]\}_{k=1}^{\infty}$ which satisfies (23), (24) as well as the qualification constraint $\|\mathcal{P}_{\Omega_{t_k}}(\mathbf{Y}_{t_k} - \mathbf{L}[t_k]\mathbf{Q}'[t_k])\| \leq \lambda$. The rest of the proof then verifies that

$\{\mathbf{L}[t_k], \mathbf{Q}[t_k]\}$ asymptotically fulfills the optimality conditions for (P1). To begin with, the following equivalent formulation of (P1) is considered at time t_k, which no longer involves the non-smooth nuclear norm.

$$\text{(P5)} \quad \min_{\substack{\{\mathbf{X}\in\mathbb{R}^{P\times P}, \mathbf{W}_1\in\mathbb{R}^{P\times t_k} \\ \mathbf{W}_2\in\mathbb{R}^{t_k\times t_k}\}}} \frac{1}{2t_k}\left\|\mathcal{P}_{\Omega_{t_k}}\left(\mathbf{Y}_{t_k}-\mathbf{X}\right)\right\|_F^2 + \frac{\lambda}{2t_k}\left\{\mathrm{tr}(\mathbf{W}_1)+\mathrm{tr}(\mathbf{W}_2)\right\}$$
$$\text{s. to} \quad \mathbf{W} := \begin{pmatrix} \mathbf{W}_1 & \mathbf{X} \\ \mathbf{X}' & \mathbf{W}_2 \end{pmatrix} \succeq \mathbf{0}$$

To explore the optimality conditions for (P5), first form the Lagrangian

$$\mathcal{L}_{t_k}(\mathbf{X},\mathbf{W}_1,\mathbf{W}_2;\mathbf{M}) = \frac{1}{2t_k}\left\|\mathcal{P}_{\Omega_{t_k}}\left(\mathbf{Y}_{t_k}-\mathbf{X}\right)\right\|_F^2 + \frac{\lambda}{2t_k}\left(\mathrm{tr}\{\mathbf{W}_1\}+\mathrm{tr}\{\mathbf{W}_2\}\right) - \langle \mathbf{M},\mathbf{W}\rangle. \quad (25)$$

where $\mathbf{M}$ denotes the dual variables associated with the positive semi-definiteness constraint in (P5). For notational convenience, partition $\mathbf{M}$ into four blocks, namely $\mathbf{M}_1 := [\mathbf{M}]_{11}$, $\mathbf{M}_2 := [\mathbf{M}]_{12}$, $\mathbf{M}_3 := [\mathbf{M}]_{22}$, and $\mathbf{M}_4 := [\mathbf{M}]_{21}$, in accordance with the block structure of $\mathbf{W}$ in (P5), where $\mathbf{M}_1$ and $\mathbf{M}_3$ are $P\times P$ and $t_k\times t_k$ matrices. The optimal solution to (P1) must: (i) null the (sub)gradients

$$\nabla_{\mathbf{X}}\mathcal{L}_{t_k}(\mathbf{X},\mathbf{W}_1,\mathbf{W}_2;\mathbf{M}) = -\frac{1}{t_k}\mathcal{P}_{\Omega_{t_k}}\left(\mathbf{Y}_{t_k}-\mathbf{X}\right) - \mathbf{M}_2 - \mathbf{M}_4' \quad (26)$$

$$\nabla_{\mathbf{W}_1}\mathcal{L}_{t_k}(\mathbf{X},\mathbf{W}_1,\mathbf{W}_2;\mathbf{M}) = \frac{\lambda}{2t_k}\mathbf{I}_L - \mathbf{M}_1 \quad (27)$$

$$\nabla_{\mathbf{W}_2}\mathcal{L}_{t_k}(\mathbf{X},\mathbf{W}_1,\mathbf{W}_2;\mathbf{M}) = \frac{\lambda}{2t_k}\mathbf{I}_{t_k} - \mathbf{M}_3 \quad (28)$$

(ii) satisfy the complementary slackness condition $\langle \mathbf{M},\mathbf{W}\rangle = 0$; (iii) primal feasibility $\mathbf{W}\succeq\mathbf{0}$; and (iv) dual feasibility $\mathbf{M}\succeq\mathbf{0}$.

Introduce the candidate primal variables $\mathbf{X}[k] := \mathbf{L}[t_k]\mathbf{Q}'[t_k]$, $\mathbf{W}_1[k] := \mathbf{L}[t_k]\mathbf{L}'[t_k]$ and $\mathbf{W}_2[k] := \mathbf{Q}[t_k]\mathbf{Q}'[t_k]$; and the dual variables $\mathbf{M}_1[k] := \frac{\lambda}{2t_k}\mathbf{I}_L$, $\mathbf{M}_3[k] := \frac{\lambda}{2t_k}\mathbf{I}_t$, $\mathbf{M}_2[k] := -(1/2t_k)\mathcal{P}_{\Omega_{t_k}}(\mathbf{Y}_{t_k} - \mathbf{L}[t_k]\mathbf{Q}'[t_k])$, and $\mathbf{M}_4[k] := \mathbf{M}_2'[k]$. Then, it can be readily verified that (i), (iii) and (iv) hold. Moreover, (ii) holds since

$$\begin{aligned}
&\langle \mathbf{M}[k],\mathbf{W}[k]\rangle \\
&= \langle \mathbf{M}_1[k],\mathbf{W}_1[k]\rangle + \langle \mathbf{M}_2[k],\mathbf{X}[k]\rangle \\
&\quad + \langle \mathbf{M}_2'[k],\mathbf{X}'[k]\rangle + \langle \mathbf{M}_3[k],\mathbf{W}_2[k]\rangle \\
&= \frac{\lambda}{2t_k}\langle \mathbf{I}_L, \mathbf{L}[t_k]\mathbf{L}'[t_k]\rangle + \frac{\lambda}{2t_k}\langle \mathbf{I}_{t_k}, \mathbf{Q}[t_k]\mathbf{Q}'[t_k]\rangle \\
&\quad - \frac{1}{t_k}\left\langle \mathcal{P}_{\Omega_{t_k}}\left(\mathbf{Y}_{t_k}-\mathbf{L}[t_k]\mathbf{Q}'[t_k]\right), \mathbf{L}[t_k]\mathbf{Q}'[t_k]\right\rangle \\
&= \frac{1}{2t_k}\left\langle \mathbf{L}[t_k], \lambda\mathbf{L}[t_k] - \mathcal{P}_{\Omega_{t_k}}\left(\mathbf{Y}_{t_k}-\mathbf{L}[t_k]\mathbf{Q}'[t_k]\right)\mathbf{Q}[t_k]\right\rangle \\
&\quad + \frac{1}{2t_k}\left\langle \mathbf{Q}'[t_k], \lambda\mathbf{Q}'[t_k] - \mathbf{L}'[t_k]\mathcal{P}_{\Omega_{t_k}}\left(\mathbf{Y}_{t_k}-\mathbf{L}[t_k]\mathbf{Q}'[t_k]\right)\right\rangle \\
&= \frac{1}{2t_k}\left\langle \mathbf{L}[t_k], \lambda\mathbf{L}[t_k] - \mathcal{P}_{\Omega_{t_k}}\left(\mathbf{Y}_{t_k}-\mathbf{L}[t_k]\mathbf{Q}'[t_k]\right)\mathbf{Q}[t_k]\right\rangle
\end{aligned}$$

where the last equality is due to (24). Putting pieces together, the Cauchy-Schwartz inequality implies that

$$\lim_{k\to\infty}|\langle \mathbf{M}[k],\mathbf{W}[k]\rangle| \le \sup_k \|\mathbf{L}[t_k]\|_F \times \lim_{k\to\infty}\left\|\frac{1}{2t_k}\left(\lambda\mathbf{L}[t_k] - \mathcal{P}_{\Omega_{t_k}}\left(\mathbf{Y}_{t_k}-\mathbf{L}[t_k]\mathbf{Q}'[t_k]\right)\mathbf{Q}[t_k]\right)\right\|_F = 0$$

holds almost surely due to (23), and (A3) which says $\|\mathbf{L}[t_k]\|_F$ is bounded. All in all, $\lim_{k\to\infty}\langle \mathbf{M}[k],\mathbf{W}[k]\rangle = 0$, which completes the proof. ■

REFERENCES

[1] [Online]. Available: http://internet2.edu/observatory/archive/data-collections.html

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

[3] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," presented at the Allerton Conf. Commun., Control, Comput., Monticello, IL, USA, Sep. 2010.

[4] L. Balzano, "On GROUSE and incremental SVD," presented at the 5th Workshop Comput. Adv. Multi-Sensor Adapt. Process., St. Martin, Dec. 2013.

[5] L. Balzano and S. J. Wright, "Local convergence of an algorithm for subspace identification from partial data," 2013, arXiv preprint arXiv:1306.3391 [Online]. Available: http://arxiv.org/abs/1306.3391

[6] 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.

[7] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, pp. 183–202, Jan. 2009.

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

[9] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.

[10] S. Burer and R. D. Monteiro, "Local minima and convergence in low-rank semidefinite programming," *Math. Programm.*, vol. 103, no. 3, pp. 427–444, Dec. 2005.

[11] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–722, Apr. 2009.

[12] E. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, pp. 925–936, Jun. 2009.

[13] Y. Chi, Y. C. Eldar, and R. Calderbank, "PETRELS: Subspace estimation and tracking from partial observations," presented at the IEEE Int. Conf. Acoust., Speech, Signal Process., Kyoto, Japan, Mar. 2012.

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

[15] W. Dai, O. Milenkovic, and E. Kerman, "Subspace evolution and transfer (SET) for low-rank matrix completion," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3120–3132, Jul. 2011.

[16] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 2002.

[17] J. Feng, H. Xu, and S. Yan, "Online robust PCA via stochastic optimization," presented at the Adv. Neural Inf. Process. Syst., Lake Tahoe, NV, USA, Dec. 2013.

[18] J. Finn, K. Nael, V. Deshpande, O. Ratib, and G. Laub, "Cardiac MR imaging: State of the technology," *Radiology*, vol. 241, no. 2, pp. 338–354, Nov. 2006.

[19] 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, Feb. 2011.

[20] H. Gao, "Prior rank, intensity and sparsity model (PRISM): A divide-and-conquer matrix decomposition model with low-rank coherence and sparse variation," presented at the SPIE Opt. Eng. Appl., San Diego, CA, USA, Aug. 2012.

[21] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York, NY, USA: Springer, 2009.

[22] H. Kim, S. Lee, X. Ma, and C. Wang, "Higher-order PCA for anomaly detection in large-scale networks," presented at the 3rd Workshop Comput. Adv. Multi-Sensor Adapt. Process., Aruba, Dutch Antilles, Dec. 2009.
[23] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. New York, NY, USA: Springer, 2009.
[24] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[25] J. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions with application to arithmetic complexity and statistics," *Linear Algebra Appl.*, vol. 18, no. 2, pp. 95–138, 1977.
[26] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," presented at the ACM SIGMETRICS, New York, NY, USA, Jul. 2004.
[27] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, pp. 208–220, Jan. 2013.
[28] 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.
[29] M. Mardani, G. Mateos, and G. B. Giannakis, "Decentralized sparsity regularized rank minimization: Applications and algorithms," *IEEE Trans. Signal Process.*, vol. 61, no. 21, pp. 5374–5388, Nov. 2013.
[30] 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.
[31] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $o(1/k^2)$," *Sov. Math. Doklady*, vol. 27, pp. 372–376, 1983.
[32] 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.
[33] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, Aug. 2010.
[34] M. Signoretto, R. V. Plas, B. D. Moor, and J. A. K. Suykens, "Tensor versus matrix completion: A comparison with application to spectral data," *IEEE Signal Process. Lett.*, vol. 18, pp. 403–406, Jul. 2011.
[35] K. Slavakis and G. B. Giannakis, "Online dictionary learning from big data using accelerated stochastic approximation algorithms," presented at the Int. Conf. Acoust., Speech, Signal Process., Florence, Italy, May 2014.
[36] K. Slavakis, G. B. Giannakis, and G. Mateos, "Modeling and optimization for Big Data analytics," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 18–31, Sep. 2014.
[37] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.
[38] P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Real-time online singing voice separation from monaural recordings using robust low-rank modeling," presented at the Annu. Conf. Int. Soc. Music Inf. Retrieval, Porto, Portugal, Oct. 2012.
[39] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, Dec. 2004, pp. 1329–1336.
[40] N. Srebro and A. Shraibman, "Rank, trace-norm and max-norm," in *Proc. Learn. Theory*, 2005, pp. 545–560, Springer.
[41] J. Sun, D. Tao, and C. Faloutsos, "Beyond streams and graphs: Dynamic tensor analysis," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, Philadelphia, PA, USA, Aug. 2006, pp. 374–383.
[42] M. Thottan and C. Ji, "Anomaly detection in IP networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2191–2204, Aug. 2003.
[43] R. Tomioka, T. Suzuki, K. Hayashi, and H. Kashima, "Statistical performance of convex tensor decomposition," in *Proc. Adv. Neural Inf. Process. Syst.*, Granada, Spain, Dec. 2011, pp. 972–980.
[44] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal. Process.*, vol. 43, pp. 95–107, Jan. 1995.
[45] J. F. Yang and M. Kaveh, "Adaptive eigensubspace algorithms for direction or frequency estimation and tracking," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 2, pp. 241–251, Feb. 1988.

Morteza Mardani (S'06) received his B.Sc. degree in Electrical Engineering from the Shahid Bahonar University of Kerman, Kerman, Iran, in 2006 and the M.Sc. degree in Electrical Engineering from the University of Tehran, Tehran, Iran, in 2009. Since September 2009, he has been working toward his Ph.D. degree with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis. His research interests include network science, statistical and large-scale machine learning, and biomedical imaging.

Mr. Mardani is the (co-) recipient of the Best Student Paper Award from the 13th IEEE Workshop on Signal Processing Advances in Wireless Communications in June 2012. He has also awarded a couple of Fellowships from the University of Minnesota including the ADC Fellowship offered by the Digital Technology Center for 2009–2010 and 2010–2011; and the Doctoral Dissertation Fellowship for 2014–2015.

Gonzalo Mateos (M'12) received his B.Sc. degree in Electrical Engineering from Universidad de la República (UdelaR), Montevideo, Uruguay in 2005 and the M.Sc. and Ph.D. degrees in Electrical Engineering from the University of Minnesota (UofM), Minneapolis, in 2009 and 2011.

Currently, he is an assistant professor with the Department of Electrical and Computer Engineering, University of Rochester. During 2013, he was a visiting scholar with the CS Department, Carnegie Mellon University. From 2003 to 2006, he was an assistant with the Department of Electrical Engineering, UdelaR. From 2004 to 2006, he worked as a Systems Engineer at Asea Brown Boveri (ABB), Uruguay. His research interests lie in the areas of statistical learning from Big Data, network science, wireless communications, and signal processing. His current research focuses on algorithms, analysis, and application of statistical signal processing tools to dynamic network health monitoring, social, power grid, and Big Data analytics. His doctoral work has been recognized with the 2013 UofM's Best Dissertation Award (Honorable Mention) across all Physical Sciences and Engineering areas.

Georgios B. Giannakis (F'97) received his Diploma in Electrical Engr. from the Ntl. Tech. Univ. of Athens, Greece, 1981. From 1982 to 1986 he was with the Univ. of Southern California (USC), where he received his M.Sc. in Electrical Engineering, 1983, M.Sc. in Mathematics, 1986, and Ph.D. in Electrical Engr., 1986. Since 1999 he has been a professor with the Univ. of Minnesota, where he now holds an ADC Chair in Wireless Telecommunications in the ECE Department, and serves as director of the Digital Technology Center.

His general interests span the areas of communications, networking and statistical signal processing—subjects on which he has published more than 375 journal papers, 635 conference papers, 21 book chapters, two edited books and two research monographs (h-index 112). Current research focuses on sparsity and big data analytics, wireless cognitive radios, mobile *ad hoc* networks, renewable energy, power grid, gene-regulatory, and social networks. He is the (co-) inventor of 23 patents issued, and the (co-) recipient of 8 best paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, the G. W. Taylor Award for Distinguished Research from the University of Minnesota, and the IEEE Fourier Technical Field Award (2015). He is a Fellow of EURASIP, and has served the IEEE in a number of posts, including that of a Distinguished Lecturer for the IEEE-SP Society.