RANK MINIMIZATION FOR SUBSPACE TRACKING FROM INCOMPLETE DATA

Morteza Mardani, Gonzalo Mateos, and Georgios B. Giannakis

Dept. of ECE, University of Minnesota, Minneapolis, MN

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 pose major challenges towards achieving this goal. In this context, the fresh look advocated here permeates benefits from rank minimization to track low-dimensional subspaces from incomplete data. Leveraging the low-dimensionality of the subspace sought, a novel estimator is proposed based on an exponentially-weighted least-squares criterion regularized with the nuclear norm. After recasting the non-separable nuclear norm into a form amenable to online optimization, a real-time algorithm is developed and its convergence established under simplifying technical assumptions. The novel subspace tracker can asymptotically offer the well-documented performance guarantees of the *batch* nuclear-norm regularized estimator. Simulated tests with real Internet data confirm the efficacy of the proposed algorithm in tracking the traffic subspace, and its superior performance relative to state-of-the-art alternatives.

Index Terms— Low rank, online algorithm, matrix completion.

1. INTRODUCTION

Nowadays ubiquitous e-commerce sites, the Web, and Internet-friendly portable devices generate massive volumes of data. The undeniable 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 problem of extracting the most informative, yet low-dimensional structure from high-dimensional datasets is of paramount importance [6]. The sheer volume of data and the fact that 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. In addition, large-scale datasets are often incomplete, and prone to corrupt measurements as well as communication errors. In this context, consider the following streaming data model with incomplete observations at time t

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

where $\mathbf{x}_t \in \mathbb{R}^P$ is the signal of interest, and $\mathbf{v}_t$ accounts for the noise. The set $\omega_t \subset [1, 2, \ldots, P]$ contains the indices of available observations, and 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. Suppose that the signal sequence $\{\mathbf{x}_t\}_{t=1}^{\infty}$ lives in a *low-dimensional* ($\ll P$) linear subspace $\mathcal{L}_t$. Given the incomplete observations $\{\mathcal{P}_{\omega_\tau}(\mathbf{y}_\tau), \omega_\tau\}_{\tau=1}^{t}$, this paper deals with online (adaptive) estimation of $\mathcal{L}_t$, and reconstruction of the signal $\mathbf{x}_t$ as a byproduct.

Work was supported by an AFOSR MURI grant no. FA 9550-10-1-0567

Relation to prior work. Subspace tracking has a long history in signal processing. An early noteworthy representative is the projection approximation subspace tracking (PAST) algorithm [14]. Recently, an algorithm (termed GROUSE) for tracking subspaces from incomplete observations was put forth in [1], based on incremental gradient descent iterations on the Grassmannian manifold of subspaces. PETRELS is a second-order recursive least-squares (RLS)-type algorithm, that extends the seminal PAST iterations to handle missing data [4]. As noted in [5], 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 [5]. Lack of regularization in PETRELS can lead to unstable behaviors, especially when the amount of missing data is large. Relative to all aforementioned works, the algorithmic framework of this paper permeates benefits from *rank minimization* to low-dimensional subspace tracking (Section 3), and offers theoretical performance guarantees (Section 4).

Contributions. Leveraging the low dimensionality of the underlying subspace, a novel estimator is proposed based on an exponentially-weighted least-squares (LS) criterion regularized with the nuclear norm of the unknown signal of interest. Upon recasting the non-separable nuclear norm into a form amenable to online optimization, a real-time algorithm for susbspace tracking is developed and its convergence is established under simplifying technical assumptions. Interestingly, under mild assumptions the proposed online algorithm attains the global optimum of the batch nuclear-norm regularized problem, whose quantifiable performance has well-appreciated merits [2, 3]. As a byproduct, the proposed online algorithm offers a viable approach to solving large-scale matrix completion problems. Simulated tests with Internet traffic data corroborate the effectiveness of the proposed algorithm for traffic estimation, and its superior performance relative to state-of-the-art alternatives [1, 4].

Notation: Operators $(\cdot)'$, $\otimes$, $\lambda_{\min}(\cdot)$, and $\sigma_{\max}(\cdot)$ will denote transposition, Kronecker product, minimum eigenvalue, and maximum singular value, respectively; $|\cdot|$ is the magnitude of a scalar and $\|\cdot\|_2$ the ℓ_2-norm of a vector. For matrix $\mathbf{X}$, $\|\mathbf{X}\|_F$ denotes the Frobenius norm, and $\mathbf{X} \succeq \mathbf{0}$ means that $\mathbf{X}$ is positive semidefinite. 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, and $\mathbf{0}_{n\times p} := \mathbf{0}_n\mathbf{0}_p'$.

2. NUCLEAR-NORM REGULARIZATION

Collect the indices of available observations up to time t in the set $\Omega_t := \cup_{\tau=1}^{t}\omega_t$, and the actual observations in the matrix $\mathcal{P}_{\Omega_t}(\mathbf{Y}_t) := [\mathcal{P}_{\omega_1}(\mathbf{y}_1), \ldots, \mathcal{P}_{\omega_t}(\mathbf{y}_t)] \in \mathbb{R}^{P\times t}$. 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 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 LS sense, as well as minimize the rank of $\mathbf{X}_t$. Unfortunately, albeit natural the rank criterion is in general NP-hard to optimize [12]. Typically, 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 surrogate to rank$(\mathbf{X}_t)$. Accordingly, one solves [3]

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

where λ_t is a (possibly time-varying) rank-controlling parameter. Albeit convex, (P1) is a non-smooth optimization problem (the nuclear norm is not differentiable at the origin). In addition, scalable algorithms to impute missing entries for streaming observations should effectively overcome the following challenges: (c1) the problem size can easily become quite large, since the number of optimization variables is Pt; (c2) existing iterative solvers for (P1) typically rely on costly SVD computations per iteration; see e.g., [2]; and (c3) different from the Frobenius-norm, (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) and (c2) are dealt with to arrive at an efficient online algorithm in Section 3.

2.1. A separable low-rank regularization

To address (c1) and reduce the computational complexity and memory storage requirements of the algorithm sought, it is henceforth assumed that the dimensionality of the underlying subspace $\mathcal{L}_t$ is bounded by a known quantity ρ. Accordingly, it is natural to require $\rho \geq \text{rank}(\hat{\mathbf{X}}_t)$. As argued in Remark 1, the smaller the value of ρ, the more efficient the algorithm becomes. Because $\text{rank}(\hat{\mathbf{X}}_t) \leq \rho$, (P1)'s search space is effectively reduced and one can factorize the 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. It is possible to interpret the columns of $\mathbf{X}$ (viewed as points in $\mathbb{R}^P$) as belonging to the low-rank subspace, spanned by the columns of $\mathbf{L}$. The rows of $\mathbf{Q}$ are thus the projections of the columns of $\mathbf{X}$ onto the subspace. Adopting this reparametrization of $\mathbf{X}$ in (P1) one arrives at a nonconvex problem where the number of variables is reduced from Pt in (P1), to $\rho(P+t)$. The savings can be significant when ρ is small, and both P and t are large.

To address (c2) [along with (c3) as it will become clear in Section 3], consider the following alternative characterization of the nuclear norm [12]

$$\|\mathbf{X}\|_* := \min_{\{\mathbf{L},\mathbf{Q}\}} \ \frac{1}{2}\left\{\|\mathbf{L}\|_F^2 + \|\mathbf{Q}\|_F^2\right\}, \quad \text{s. t.} \ \ \mathbf{X} = \mathbf{L}\mathbf{Q}'. \quad (2)$$

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 reformulation of (P1) provides an important first step towards obtaining an online algorithm:

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

As asserted in [8, Lemma 1], adopting the separable 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) [which could have considerably less variables than (P1)], 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, the next proposition shows that under relatively mild assumptions on rank$(\hat{\mathbf{X}}_t)$ and the noise variance, stationary points of (P2) qualify as global optimum solutions of (P1).

Proposition 1. *[8] Let $\{\bar{\mathbf{L}}_t, \bar{\mathbf{Q}}_t\}$ be a stationary point of (P2). If $\sigma_{\max}[\mathcal{P}_{\Omega_t}(\mathbf{Y}_t - \bar{\mathbf{L}}_t\bar{\mathbf{Q}}_t')] \leq \lambda_t$, then $\hat{\mathbf{X}}_t := \bar{\mathbf{L}}_t\bar{\mathbf{Q}}_t'$ is the globally optimal solution of (P1).*

3. ONLINE MATRIX IMPUTATION

As stated in Section 1, the goal is to recursively estimate $\hat{\mathbf{x}}_t$ at time t from historical observations $\{\mathcal{P}_{\omega_\tau}(\mathbf{y}_\tau), \omega_\tau\}_{\tau=1}^t$, naturally placing more importance on recent measurements. To this end, one possible adaptive counterpart to (P2) is the exponentially-weighted LS estimator found by minimizing the empirical cost ($\mathbf{Q} := [\mathbf{q}_1, \ldots, \mathbf{q}_t]$)

$$\min_{\{\mathbf{L},\mathbf{Q}\}} \sum_{\tau=1}^t \beta^{t-\tau}\left[\frac{1}{2}\|\mathcal{P}_{\omega_\tau}(\mathbf{y}_\tau - \mathbf{L}\mathbf{q}_\tau)\|_2^2 + \frac{\bar{\lambda}_t}{2}\|\mathbf{L}\|_F^2 + \frac{\lambda_t}{2}\|\mathbf{q}_\tau\|_2^2\right] \quad (3)$$

where $\bar{\lambda}_t := \lambda_t / \sum_{\tau=1}^t \beta^{t-\tau}$, and $0 < \beta \leq 1$ is the so-termed forgetting factor. When $\beta < 1$, data in the distant past are exponentially downweighted, which facilitates tracking in nonstationary environments. In the case of infinite memory ($\beta = 1$), the formulation (3) coincides with the batch estimator (P2). This is the reason for the time-varying factor $\bar{\lambda}_t$ weighting $\|\mathbf{L}\|_F^2$.

3.1. Subspace tracking from incomplete data

Towards deriving a real-time, computationally efficient, and recursive solver of (3), 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 4.1. Per time instant t, a new datum $\{\mathcal{P}_{\omega_t}(\mathbf{y}_t), \omega_t\}$ is drawn and $\mathbf{q}_t$ is estimated via

$$\mathbf{q}[t] = \arg\min_{\mathbf{q}} \left[\frac{1}{2}\|\mathcal{P}_{\omega_t}(\mathbf{y}_t - \mathbf{L}[t-1]\mathbf{q})\|_2^2 + \frac{\lambda_t}{2}\|\mathbf{q}\|_2^2\right]. \quad (4)$$

Updating (4) entails an ℓ_2-norm regularized LS (ridge-regression) problem, that admits closed-form solution

$$\mathbf{q}[t] = \left(\lambda_t \mathbf{I}_\rho + \mathbf{L}'[t-1]\mathbf{\Omega}_t\mathbf{L}[t-1]\right)^{-1}\mathbf{L}'[t-1]\mathcal{P}_{\omega_t}(\mathbf{y}_t) \quad (5)$$

where diagonal $\mathbf{\Omega}_t \in \mathbb{R}^{P\times P}$ 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 (3) 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

$$\mathbf{L}[t] = \arg\min_{\mathbf{L}} \left[\frac{\lambda_t}{2}\|\mathbf{L}\|_F^2 + \sum_{\tau=1}^t \beta^{t-\tau}\frac{1}{2}\|\mathcal{P}_{\omega_\tau}(\mathbf{y}_\tau - \mathbf{L}\mathbf{q}[\tau])\|_2^2\right]. \quad (6)$$

Notice that (6) decouples over the rows of $\mathbf{L}$ which are obtained in parallel via

$$\mathbf{l}_p[t] = \arg\min_{\mathbf{l}} \left[\frac{\lambda_t}{2}\|\mathbf{l}\|_2^2 + \sum_{\tau=1}^t \beta^{t-\tau}\omega_{p,\tau}(y_{p,\tau} - \mathbf{l}'\mathbf{q}[\tau])^2\right], \quad (7)$$

for $p = 1, \ldots, P$, where $\omega_{p,\tau}$ denotes the p-th diagonal entry of $\mathbf{\Omega}_\tau$. For $\beta = 1$ and fixed $\lambda_t = \lambda$, $\forall t$, subproblems (7) can be efficiently solved using the RLS algorithm [13]. Upon defining $\mathbf{s}_p[t] := \sum_{\tau=1}^t \beta^{t-\tau}\omega_{p,\tau}(y_{p,\tau})\mathbf{q}[\tau]$, $\mathbf{H}_p[t] := \sum_{\tau=1}^t \beta^{t-\tau}\omega_{p,\tau}\mathbf{q}[\tau]\mathbf{q}'[\tau] + \lambda_t\mathbf{I}_\rho$, and $\mathbf{M}_p[t] := \mathbf{H}_p^{-1}[t]$, one simply 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 $0 < \beta < 1$ the regularization term $(\lambda_t/2)\|\mathbf{l}\|_2^2$ in (7) makes it impossible to express $\mathbf{H}_p[t]$ in terms of $\mathbf{H}_p[t-1]$ plus a

Algorithm 1 : Subspace tracking from incomplete observations

input $\{\mathcal{P}_{\omega_\tau}(\mathbf{y}_\tau), \omega_\tau\}_{\tau=1}^{\infty}$, $\{\lambda_\tau\}_{\tau=1}^{\infty}$, and β.
initialize $\mathbf{G}_p[0] = \mathbf{0}_{\rho\times\rho}$, $\mathbf{s}_p[0] = \mathbf{0}_\rho$, $p = 1, ..., P$, and $\mathbf{L}[0]$ at random.
for $t = 1, 2, \ldots$ **do**
$\mathbf{D}[t] = (\lambda_t \mathbf{I}_\rho + \mathbf{L}'[t-1]\mathbf{\Omega}_t\mathbf{L}[t-1])^{-1}\mathbf{L}'[t-1]$.
$\mathbf{q}[t] = \mathbf{D}[t]\mathcal{P}_{\omega_t}(\mathbf{y}_t)$.
$\mathbf{G}_p[t] = \beta\mathbf{G}_p[t-1] + \omega_{p,t}\mathbf{q}[t]\mathbf{q}[t]', \quad p = 1, \ldots, P$.
$\mathbf{s}_p[t] = \beta\mathbf{s}_p[t-1] + \omega_{p,t}y_{p,t}\mathbf{q}[t], \quad p = 1, \ldots, P$.
$\mathbf{l}_p[t] = (\mathbf{G}_p[t] + \lambda_t\mathbf{I}_\rho)^{-1}\mathbf{s}_p[t], \quad p = 1, ..., P$.
return $\hat{\mathbf{x}}_t := \mathbf{L}[t]\mathbf{q}[t]$.
end for

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 $\mathbf{H}_p[t]$, $p = 1, \ldots, P$, the overall recursive algorithm for subspace tracking from incomplete data is tabulated under Algorithm 1.

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 of multiplication as well as additions is $\mathcal{O}(P\rho^2)$. The overall cost of the algorithm per iteration can be safely estimated as $\mathcal{O}(P\rho^3)$, which is affordable since ρ is typically small (cf. the low rank assumption).

Remark 2. [Tuning λ_t] In practice, to tune λ_t one can resort to the heuristic rules proposed in [3], which build upon the following reasonable 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 $\lambda_t = \left(\sqrt{P} + \sqrt{t_e}\right)\sqrt{\pi}\sigma$ which naturally increases as time evolves, and where $t_e := \sum_{\tau=1}^{t}\beta^{t-\tau}$ is the effective time window.

4. PERFORMANCE GUARANTEES

This section studies the performance of Algorithm 1 for the infinite memory special case i.e., when $\beta = 1$. In the sequel, to make the analysis tractable the following assumptions are made:

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

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

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

To clearly delineate the scope of the analysis, it is worth commenting on the assumptions 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 [13]. 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., [13], as $\beta \to 1$ 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 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 [9].

4.1. Convergence

The convergence of the iterates generated by Algorithm 1 is established first. Upon defining the function

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

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

$$C_t(\mathbf{L}) := \frac{1}{t}\sum_{\tau=1}^{t}\ell_\tau(\mathbf{L}) + \frac{\lambda_t}{2t}\|\mathbf{L}\|_F^2. \tag{8}$$

Normalization (by t) ensures that the cost function does not grow unbounded as time evolves. For any finite t, (8) 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

$$\hat{C}_t(\mathbf{L}) = \frac{1}{t}\sum_{\tau=1}^{t} g_\tau(\mathbf{L}, \mathbf{q}[\tau]) + \frac{\lambda_t}{2t}\|\mathbf{L}\|_F^2 \tag{9}$$

in which $\mathbf{q}[t]$ is obtained based on the prior subspace estimate $\mathbf{L}[t-1]$ after solving [cf. (4)] $\mathbf{q}[t] = \arg\min_{\mathbf{q}} g_t(\mathbf{L}[t-1], \mathbf{q})$. Obtaining $\mathbf{q}[t]$ this way resembles the projection approximation adopted in [14]. Since $\hat{C}_t(\mathbf{L})$ is a smooth convex function, the minimizer $\mathbf{L}[t] = \arg\min_{\mathbf{L}} \hat{C}_t(\mathbf{L})$ is the solution of the quadratic equation $\nabla\hat{C}_t(\mathbf{L}[t]) = \mathbf{0}_{P\times\rho}$.

So far, it is apparent that 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 $t \to \infty$. The latter result is summarized in the next proposition, whose proof is inspired by [11] and can be found in [9].

Proposition 2. *Suppose $\lambda_t = \lambda\ \forall t$, 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 coincide with the stationary points of the batch problem (P2).*

The sampling set Ω_t plays a key role towards satisfying the Hessian's positive semi-definiteness condition in Proposition 2. Intuitively, if the missing entries are somehow uniformly spread across time, the likelihood that $\nabla^2\hat{C}_t(\mathbf{L}) = \frac{\lambda}{t}\mathbf{I}_{P\rho} + \frac{1}{t}\sum_{\tau=1}^{t}(\mathbf{q}[\tau]\mathbf{q}'[\tau]) \otimes \mathbf{\Omega}_\tau \succeq c\mathbf{I}_{P\rho}$ holds is higher.

4.2. Optimality

In line with Proposition 1, one may be prompted to ponder whether the online estimator offers the performance guarantees of the nuclear-norm regularized estimator (P1), for which stable/exact recovery results are well documented e.g., in [2, 3]. Specifically, given the learned subspace $\bar{\mathbf{L}}[t]$ and the corresponding $\bar{\mathbf{Q}}[t]$ [obtained via (4)]

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) when $t \to \infty$? This in turn requires asymptotic analysis of the optimality conditions for (P1), and is established in the next proposition. Additionally, numerical tests in Section 5 corroborate that the online estimator attains the performance of (P1) after reasonable number of iterations.

Proposition 3. *For the iterates generated by Algorithm 1, 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 the convergence condition c1), even though it holds for a time invariant rank-controlling parameter λ as per Proposition 2, numerical tests indicate that it also holds for the time-varying case (e.g., when λ_t is chosen as suggested in Remark 2). According to A2) and A3), $\sigma_{\max}[\mathcal{P}_{\Omega_t}(\mathbf{Y}_t - \mathbf{L}[t]\mathbf{Q}'[t])] \approx \mathcal{O}(\sqrt{t})$, 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.

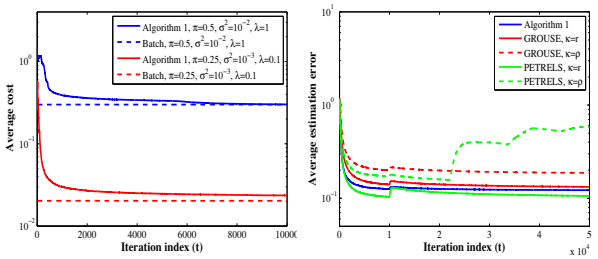
5. NUMERICAL TESTS

The convergence and effectiveness of Algorithm 1 is assessed in this section via computer simulations.

5.1. Synthetic data tests

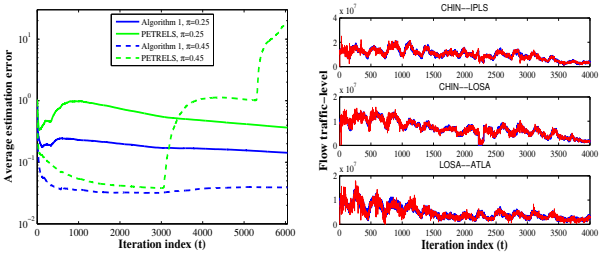
The signal $\mathbf{x}_t = \mathbf{U}\mathbf{w}_t$ is generated from the low-dimensional subspace $\mathbf{U} \in \mathbb{R}^{P\times r}$, with i.i.d. entries $u_{p,i} \sim \mathcal{N}(0, 1/P)$, and projection coefficients $w_{i,t} \sim \mathcal{N}(0, 1)$. The noise $v_{i,t} \sim \mathcal{N}(0, \sigma^2)$ is i.i.d., and the entries of $\mathbf{y}_t$ are sampled uniformly at random with probability π to form the diagonal sampling matrix $\mathbf{\Omega}_t$. The observations at time t are then generated as $\mathcal{P}_{\omega_t}(\mathbf{y}_t) = \mathbf{\Omega}_t(\mathbf{x}_t + \mathbf{v}_t)$. Fix $r = 5$ and $\rho = 10$, while different values of π and σ are examined. The evolution of the average cost $C_t(\mathbf{L}[t])$ in (8) for different percentages of missing data and noise variances is depicted in Fig. 1(a). 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 its batch counterpart (P1), which corroborates that Algorithm 1 can attain the performance of (P1). This observation together with the low-complexity of Algorithm 1's iterations [cf. Remark 1], make it a viable alternative for solving large-scale matrix completion problems.

Next, Algorithm 1 is compared with two state-of-the-arte subspace trackers, namely PETRELS [4] and GROUSE [1]. These two algorithms require and estimate of the dimensionality of the underlying subspace, which is denoted by κ. Set $\lambda = 0.1$, $\beta = 0.99$, and consider an abrupt change in the subspace at $t = 10^4$ to evaluate the tracking performance of the algorithms. The figure of merit is the average estimation error $e_t := \frac{1}{t}\sum_{i=1}^{t} \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2/\|\mathbf{x}_i\|_2$, which is depicted in Fig. 1(b). It is observed that if the subspace dimensionality is chosen as $\kappa = \rho$, Algorithm 1 attains a better estimation accuracy than PETRLES and GROUSE (a constant step size 0.1 was adopted for the latter). Even though PETRELS works well for $\kappa = r$, if one overestimates the rank PETRELS exhibiting erratic behaviors for 25% available observations. As expected, for the ideal choice of $\kappa = r$ all three schemes achieve similar estimation accuracy. The smaller error exhibited by PETRELS (relative Algorithm 1) might be due to a suboptimum choice of λ. Still, Algorithm 1 is more stable numerically when the amount of missing observations



(a) (b)

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



(a) (b)

Fig. 2. Performance for Internet-2 data if $\kappa = \rho = 10$ and $\beta = 0.95$. (a) Average estimation error for various amounts of missing data. (b) Estimated (red) versus true (blue) OD flow traffic for $\pi = 0.25$.

is large, thanks to the regularization terms in (3). The price paid by Algorithm 1 is in terms of higher complexity per iteration. Note that the complexity for PETRELS is $\mathcal{O}(P\kappa^2)$, and only $\mathcal{O}(P\kappa(1+\pi\kappa))$ for the first-order algorithm GROUSE.

5.2. Tracking Internet-2 network traffic

In IP networks accurate estimation of the origin-to-destination (OD) flow traffic is a task of paramount interest. Typically, the data available is a measured traffic via NetFlow [7], for a small subset of OD flows. Several studies have demonstrated that OD flow traffic exhibits a low-intrinsic dimensionality, which is mainly due to common temporal patterns across OD flows and periodic behaviors across time [7]. In this example, OD-flow traffic-levels are collected from operation of the Internet-2 network (Internet backbone across USA) [7]. The measured OD flows contain spikes (anomalies), which are removed as detailed in [9] to end up with an anomaly-free data stream $\{\mathbf{y}_t\}$. A subset of entries of $\mathbf{y}_t$ are then selected independently with probability π, to yield the input of Algorithm 1. The evolution of the average traffic estimation error (e_t) is depicted in Fig. 2(a), for different schemes and various amounts of missing data. It is observed that the estimation accuracy and subspace learning speed degrades gracefully as the NetFlow sampling rate decreases. Algorithm 1 outperforms 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. 2(b) depicts how the representative OD flows are accurately tracked using Algorithm 1.

6. REFERENCES

[1] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Proc. of Allerton Conference on Communication, Control, and Computing*, Monticello, USA, Jun. 2010.

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

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

[4] Y. Chi, Y. C. Eldar, and R. Calderbank, "PETRELS: Subspace estimation and tracking from partial observations," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, Mar. 2012.

[5] W. Dai, O. Milenkovic, and E. Kerman, "Subspace evolution and transfer (SET) for low-rank matrix completion," *IEEE Trans. Signal Process.*, vol. 59, pp. 3120–3132, Jul. 2011.

[6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning,* Springer, Second edition, 2009.

[7] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *Proc. of ACM SIGMETRICS*, NY, USA, Jul. 2004.

[8] M. Mardani, G. Mateos, and G. B. Giannakis, "In-network sparsity regularized rank minimization: Applications and algorithms," *IEEE Trans. Signal Process.*, 2012, see also arXiv:1203.1570v1 [cs.MA].

[9] M. Mardani, G. Mateos, and G. B. Giannakis, "Dynamic anomalography: Tracking network anomalies via sparsity and low-rank," *IEEE J. Sel. Tpoics in Signal Process.*, 2012, see also arXiv:1208.4043v1 [cs.NI].

[10] G. Mateos and G. B. Giannakis, "Robust PCA as bilinear decomposition with outlier-sparsity regularization," *IEEE Trans. Signal Process.*, vol. 60, pp. 5176–5190, Oct. 2012.

[11] J. Mairal, J. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," in *J. of Machine Learning Research*, vol. 11, pp. 19–60, Jan., 2010.

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

[13] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance,* Prentice Hall, 1995.

[14] M. B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, pp. 95–107, Jan. 2012.

[15] http://internet2.edu/observatory/archive/data-collections.